COMPREHENSIVE SERVICES

We offer competitive repair and calibration services, as well as easily accessible documentation and free downloadable resources.

SELL YOUR SURPLUS

We buy new, used, decommissioned, and surplus parts from every NI series. We work out the best solution to suit your individual needs.

Sell For Cash Get Credit Receive a Trade-In Deal

OBSOLETE NI HARDWARE IN STOCK & READY TO SHIP

We stock **New, New Surplus, Refurbished,** and **Reconditioned** NI Hardware.

APEX WAVES

Bridging the gap *between the manufacturer and your legacy test system.*

1-800-915-6216

www.apexwaves.com

sales@apexwaves.com

Request a Quote CLICK HERE *TSM-1012*

NI cDAQ™-9132/9133/9134/9135/9136/9137

User Manual

NI CompactDAQ Controller

Français Deutsch 日本語 한국어 简体中文

ni.com/manuals

January 2019
371800E-01

NATIONAL INSTRUMENTS™

Worldwide Technical Support and Product Information

ni.com

Worldwide Offices

Visit ni.com/niglobal to access the branch office websites, which provide up-to-date contact information, support phone numbers, email addresses, and current events.

National Instruments Corporate Headquarters

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

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.

Legal Information

Limited Warranty

This document is provided ‘as is’ and is subject to being changed, without notice, in future editions. For the latest version, refer to `ni.com/manuals`. NI reviews this document carefully for technical accuracy; however, NI MAKES NO EXPRESS OR IMPLIED WARRANTIES AS TO THE ACCURACY OF THE INFORMATION CONTAINED HEREIN AND SHALL NOT BE LIABLE FOR ANY ERRORS.

NI warrants that its hardware products will be free of defects in materials and workmanship that cause the product to fail to substantially conform to the applicable NI published specifications for one (1) year from the date of invoice.

For a period of ninety (90) days from the date of invoice, NI warrants that (i) its software products will perform substantially in accordance with the applicable documentation provided with the software and (ii) the software media will be free from defects in materials and workmanship.

If NI receives notice of a defect or non-conformance during the applicable warranty period, NI will, in its discretion: (i) repair or replace the affected product, or (ii) refund the fees paid for the affected product. Repaired or replaced Hardware will be warranted for the remainder of the original warranty period or ninety (90) days, whichever is longer. If NI elects to repair or replace the product, NI may use new or refurbished parts or products that are equivalent to new in performance and reliability and are at least functionally equivalent to the original part or product.

You must obtain an RMA number from NI before returning any product to NI. NI reserves the right to charge a fee for examining and testing Hardware not covered by the Limited Warranty.

This Limited Warranty does not apply if the defect of the product resulted from improper or inadequate maintenance, installation, repair, or calibration (performed by a party other than NI); unauthorized modification; improper environment; use of an improper hardware or software key; improper use or operation outside of the specification for the product; improper voltages; accident, abuse, or neglect; or a hazard such as lightning, flood, or other act of nature.

THE REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND THE CUSTOMER’S SOLE REMEDIES, AND SHALL APPLY EVEN IF SUCH REMEDIES FAIL OF THEIR ESSENTIAL PURPOSE.

EXCEPT AS EXPRESSLY SET FORTH HEREIN, PRODUCTS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND AND NI DISCLAIMS ALL WARRANTIES, EXPRESSED OR IMPLIED, WITH RESPECT TO THE PRODUCTS, INCLUDING ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE OR NON-INFRINGEMENT, AND ANY WARRANTIES THAT MAY ARISE FROM USAGE OF TRADE OR COURSE OF DEALING. NI DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE OF OR THE RESULTS OF THE USE OF THE PRODUCTS IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE. NI DOES NOT WARRANT THAT THE OPERATION OF THE PRODUCTS WILL BE UNINTERRUPTED OR ERROR FREE.

In the event that you and NI have a separate signed written agreement with warranty terms covering the products, then the warranty terms in the separate agreement shall control.

Copyright

End-User License Agreements and Third-Party Legal Notices

You can find end-user license agreements (EULAs) and third-party legal notices in the following locations:

- Notices are located in the `<National Instruments>\_Legal Information` and `<National Instruments>` directories.
- EULAs are located in the `<National Instruments>\Shared\MDF\Legal\license` directory.
- Review `<National Instruments>\_Legal Information.txt` for information on including legal information in installers built with NI products.

U.S. Government Restricted Rights

If you are an agency, department, or other entity of the United States Government (“Government”), the use, duplication, reproduction, release, modification, disclosure or transfer of the technical data included in this manual is governed by the Restricted Rights provisions under Federal Acquisition Regulation 52.227-14 for civilian agencies and Defense Federal Acquisition Regulation Supplement Section 252.227-7014 and 252.227-7015 for military agencies.

Trademarks

Refer to the *NI Trademarks and Logo Guidelines* at `ni.com/trademarks` for more information on National Instruments trademarks.

ARM, Keil, and µVision are trademarks or registered of ARM Ltd or its subsidiaries.

LEGO, the LEGO logo, WEDO, and MINDSTORMS are trademarks of the LEGO Group.

TETRIX by Pitsco is a trademark of Pitsco, Inc.

FIELDBUS FOUNDATION™ and FOUNDATION™ are trademarks of the Fieldbus Foundation.

Patents

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

Export Compliance Information

Refer to the *Export Compliance Information* at `ni.com/legal/export-compliance` for the National Instruments global trade compliance policy and how to obtain relevant HTS codes, ECCNs, and other import/export data.

WARNING REGARDING USE OF NATIONAL INSTRUMENTS PRODUCTS

YOU ARE ULTIMATELY RESPONSIBLE FOR VERIFYING AND VALIDATING THE SUITABILITY AND RELIABILITY OF THE PRODUCTS WHENEVER THE PRODUCTS ARE INCORPORATED IN YOUR SYSTEM OR APPLICATION, INCLUDING THE APPROPRIATE DESIGN, PROCESS, AND SAFETY LEVEL OF SUCH SYSTEM OR APPLICATION.

PRODUCTS ARE NOT DESIGNED, MANUFACTURED, OR TESTED FOR USE IN LIFE OR SAFETY CRITICAL SYSTEMS, HAZARDOUS ENVIRONMENTS OR ANY OTHER ENVIRONMENTS REQUIRING FAIL-SAFE PERFORMANCE, INCLUDING IN THE OPERATION OF NUCLEAR FACILITIES; AIRCRAFT NAVIGATION; AIR TRAFFIC CONTROL SYSTEMS; LIFE SAVING OR LIFE SUSTAINING SYSTEMS OR SUCH OTHER MEDICAL DEVICES; OR ANY OTHER APPLICATION IN WHICH THE FAILURE OF THE PRODUCT OR SERVICE COULD LEAD TO DEATH, PERSONAL INJURY, SEVERE PROPERTY DAMAGE OR ENVIRONMENTAL HARM (COLLECTIVELY, “HIGH-RISK USES”). FURTHER, PRUDENT STEPS MUST BE TAKEN TO PROTECT AGAINST FAILURES, INCLUDING PROVIDING BACK-UP AND SHUT-DOWN MECHANISMS. NI EXPRESSLY DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS OF THE PRODUCTS OR SERVICES FOR HIGH-RISK USES.

Contents

Chapter 1
Getting Started with the cDAQ Controller

Chapter 4
Digital Input/Output and PFI

Chapter 5
Counters

Chapter 6
Digital Routing and Clock Generation

1

Getting Started with the cDAQ Controller

The National Instruments four-slot CompactDAQ cDAQ-9132, cDAQ-9134, and cDAQ-9136 controllers and the eight-slot CompactDAQ cDAQ-9133, cDAQ-9135, and cDAQ-9137 controllers are available as a Windows Embedded Standard 7 (WES7) or a LabVIEW Real-Time system. NI cDAQ-9132/9133/9134/9135 controllers feature the dual-core 1.33 GHz Intel Atom processor. NI cDAQ-9136/9137 controllers feature the quad-core 1.91 GHz Intel Atom processor.

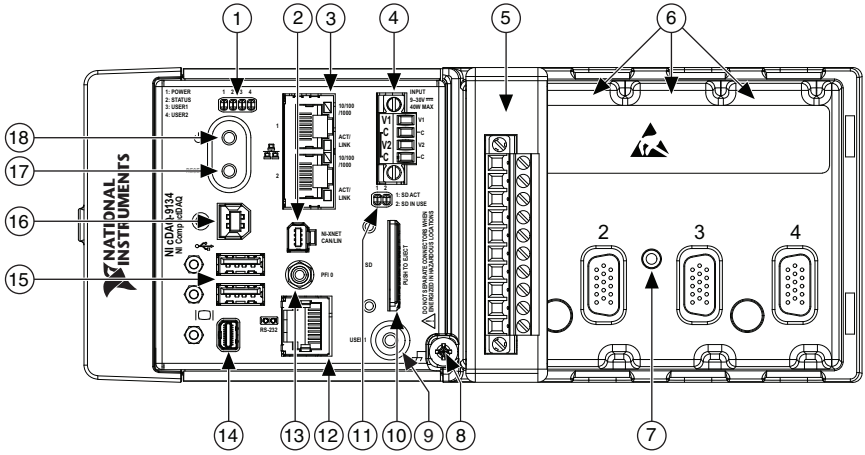
This chapter contains information about getting started with the cDAQ controller with Windows and with LabVIEW Real-Time:

- For NI cDAQ-9132/9133/9134/9135/9136/9137 for Windows, refer to the *Installing the cDAQ Controller for Windows* section
- For NI cDAQ-9132/9133/9134/9135/9136/9137 for LabVIEW Real-Time, refer to the *Installing the cDAQ Controller for LabVIEW Real-Time* section

The cDAQ controller features a number of standard interfaces and combines with C Series modules to measure a broad range of analog and digital I/O signals that can be logged to the local hard drive or an SD card. For specifications, refer to the specifications document for your cDAQ controller. For module specifications, refer to the documentation included with your C Series module(s) or go to `ni.com/manuals`.

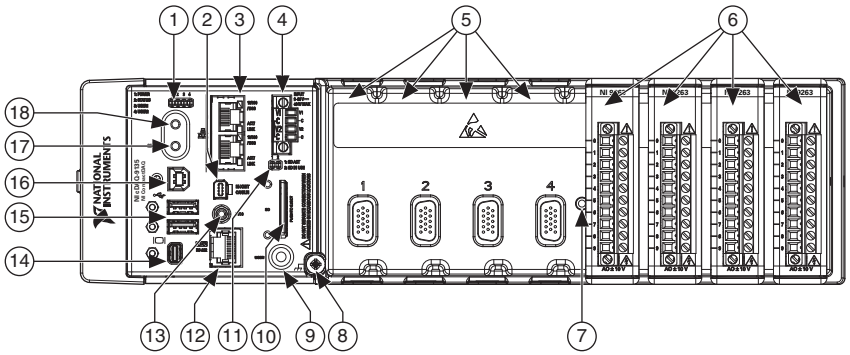
Figure 1-1 shows the NI cDAQ-9132/9134/9136 controller. Figure 1-2 shows the NI cDAQ-9133/9135/9137 controller.

Figure 1-1. NI cDAQ-9132/9134/9136 Controller



1 POWER, STATUS, USER1, and USER2 LEDs
2 NI-XNET CAN/LIN Connector (cDAQ-9134 Only)
3 RJ-45 Ethernet Ports 1 and 2, ACT/LINK and 10/100/1000 Ethernet LEDs
4 Power Connector
5 Installed C Series I/O Module
6 Module Slots
7 CMOS Reset Button
8 Chassis Grounding Screw
9 USER1 Button
10 SD Card Removable Storage and SD Card Slot Cover Mounting Holes
11 SD ACT and SD IN USE LEDs
12 RS-232 Serial Port
13 PFI 0 SMB Connector
14 Mini DisplayPort Connector
15 USB Host Ports and USB Retention Standoffs
16 USB Device Port and USB Retention Bracket
17 RESET Button
18 Power Button

Figure 1-2. NI cDAQ-9133/9135/9137 Controller



1 POWER, STATUS, USER1, and USER2 LEDs
2 NI-XNET CAN/LIN Connector (cDAQ-9135 Only)
3 RJ-45 Ethernet Ports 1 and 2, ACT/LINK and 10/100/1000 Ethernet LEDs
4 Power Connector
5 Module Slots
6 Installed C Series I/O Modules
7 CMOS Reset Button
8 Chassis Grounding Screw
9 USER1 Button
10 SD Card Removable Storage and SD Card Slot Cover Mounting Holes
11 SD ACT and SD IN USE LEDs
12 RS-232 Serial Port
13 PFI 0 SMB Connector
14 Mini DisplayPort Connector
15 USB Host Ports and USB Retention Standoffs
16 USB Device Port and USB Retention Bracket
17 RESET Button
18 Power Button

Safety Guidelines

Caution Do *not* operate the NI cDAQ-9132/9133/9134/9135/9136/9137 controller in a manner not specified in these operating instructions. Product misuse can result in a hazard. You can compromise the safety protection built into the product if the product is damaged in any way. If the product is damaged, return it to National Instruments for repair.

Note Because some C Series modules may have more stringent certification standards than the NI cDAQ-9132/9133/9134/9135/9136/9137 controller, the combined system may be limited by individual component restrictions. Refer to the specifications document for your cDAQ controller for more details.

Hot Surface This icon denotes that the component may be hot. Touching this component may result in bodily injury.

Electromagnetic Compatibility Guidelines

This product was tested and complies with the regulatory requirements and limits for electromagnetic compatibility (EMC) stated in the product specifications. These requirements and limits provide reasonable protection against harmful interference when the product is operated in the intended operational electromagnetic environment.

This product is intended for use in industrial locations. However, harmful interference may occur in some installations or when the product is connected to a peripheral device or a test object. To minimize interference with radio and television reception and prevent unacceptable performance degradation, install and use this product in strict accordance with the instructions in the product documentation.

Furthermore, any modifications to the product not expressly approved by National Instruments could void your authority to operate it under your local regulatory rules.

Caution To ensure the specified EMC performance, product installation requires either special considerations or user-installed add-on devices. Refer to the *Installing the cDAQ Controller for Windows* section or *Installing the cDAQ Controller for LabVIEW Real-Time* section for further information.

Caution To ensure the specified EMC performance, operate this product only with shielded cables and accessories. Note that the input DC power cables may be unshielded.

Caution To ensure the specified EMC performance, do not connect V2 to a DC mains supply or to any supply requiring a connecting cable longer than 3 m (10 ft). A DC mains supply is a local DC electricity supply network in the infrastructure of a site or building.

Caution To ensure the specified EMC performance, the length of any cable connected to the video and USB host ports must be no longer than 3 m (10 ft). The length of any cable connected to the RS-232 port must be no longer than 30 m (100 ft).

Caution The USB device port is intended for use in device configuration, application deployment, debug, and maintenance.

Special Guidelines for Marine Applications

Some products are Lloyd's Register (LR) Type Approved for marine (shipboard) applications. To verify Lloyd's Register certification for a product, visit `ni.com/certification` and search for the LR certificate, or look for the Lloyd's Register mark on the product label.

Caution In order to meet the EMC requirements for marine applications, install the product in a shielded enclosure with shielded and/or filtered power and input/output ports. In addition, take precautions when designing, selecting, and installing measurement probes and cables to ensure that the desired EMC performance is attained.

Hardware Symbol Definitions

The following symbols are marked on your cDAQ controller.

Caution When this symbol is marked on a product, refer to the *Safety Guidelines* section for information about precautions to take.

ESD When this symbol is marked on a product, the product could be damaged if subjected to Electrostatic Discharge (ESD) on the connector pins of any I/O port. To prevent damage, industry-standard ESD prevention measures must be employed during installation, maintenance, and operation.

EU Customers At the end of the product life cycle, all products *must* be sent to a WEEE recycling center. For more information about WEEE recycling centers, National Instruments WEEE initiatives, and compliance with WEEE Directive 2002/96/EC on Waste and Electronic Equipment, visit `ni.com/environment/weee`.

中国客户 National Instruments 符合中国电子信息产品中限制使用某些有害物质指令 (RoHS)。关于 National Instruments 中国 RoHS 合规性信息，请登录 `ni.com/environment/rohs_china`。 (For information about China RoHS compliance, go to `ni.com/environment/rohs_china`.)

Battery Directive This device contains a long-life coin cell battery. If you need to replace it, use the Return Material Authorization (RMA) process or contact an authorized National Instruments service representative. For more information about compliance with the EU Battery Directives 2006/66/EC about Batteries and Accumulators and Waste Batteries and Accumulators, visit `ni.com/environment/batterydirective`.

Unpacking

The cDAQ controller ships in an antistatic package to prevent electrostatic discharge (ESD). ESD can damage several components on the device.

Caution *Never* touch the exposed pins of connectors.

To avoid ESD damage in handling the device, take the following precautions:

- Ground yourself with a grounding strap or by touching a grounded object.
- Touch the antistatic package to a metal part of your computer chassis before removing the device from the package.

Remove the device from the package and inspect it for loose components or any other signs of damage. Notify NI if the device appears damaged in any way. Do not install a damaged device in your computer or controller.

Store the device in the antistatic package when the device is not in use.

Installing the cDAQ Controller for Windows

(NI cDAQ-9132/9133/9134/9135/9136/9137 for Windows) The NI cDAQ-9132/9133/9134/9135/9136/9137 for Windows ships with preloaded Windows Embedded Standard 7 (WES7), LabVIEW (evaluation version), and NI-DAQmx driver software. NI cDAQ-9134/9135 for Windows controllers also ship with preloaded NI-XNET software. The cDAQ controller and C Series module(s) are packaged separately.

You will need the following items to set up the NI cDAQ-9132/9133/9134/9135/9136/9137 for Windows controller:

- Power connector (packaged with the cDAQ controller)
- Ferrites (packaged with the cDAQ controller)
- Power supply
- Monitor
- Compatible mini DisplayPort cable (and adapter if necessary)
- Computer mouse and keyboard
- Number 1 and number 2 Phillips screwdrivers
- C Series module(s)

Note Table 1-1 lists the earliest supported driver version for each cDAQ controller for Windows.

Table 1-1. cDAQ Controller NI-DAQmx Software Support

cDAQ Controller	Earliest NI-DAQmx Support
NI cDAQ-9132/9134 for Windows	NI-DAQmx 14.0
NI cDAQ-9133/9135 for Windows	NI-DAQmx 14.5
NI cDAQ-9136/9137 for Windows	NI-DAQmx 15.1

The NI-DAQmx driver software preloaded onto your cDAQ controller is available for download at `ni.com/support`. The documentation for NI-DAQmx is available from **Start» All Programs»National Instruments»NI-DAQmx**.

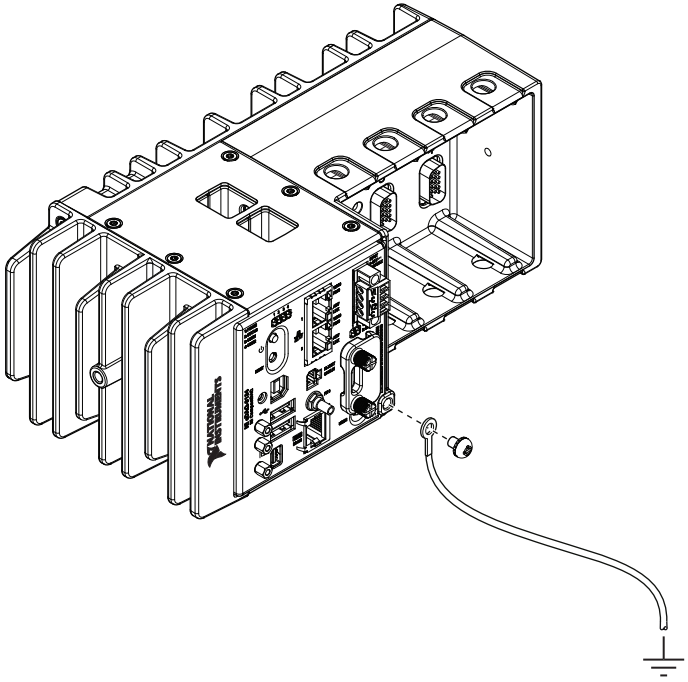
Note If you reinstall the cDAQ controller operating system, all software must also be reinstalled.

Refer to Figure 1-1 or 1-2 while completing the following assembly steps.

1. (Optional) Mount the cDAQ controller to a panel, wall, rack, or DIN rail, or attach the desktop mounting kit, as described in the *Mounting the cDAQ Controller* section.
2. Connect a video monitor to the mini DisplayPort connector with a cable (and adapter if necessary). Refer to the *Mini DisplayPort Connector* section for more information about this connector.
3. Power on the monitor.
4. Connect a computer keyboard and mouse to the USB host ports on the cDAQ controller. Use one of the ferrites around both USB cables as described in the *USB Host Ports* section.
5. Attach a ring lug to a 1.31 mm^2 (16 AWG) or larger wire. Remove the ground screw from the ground terminal on the front panel. Attach the ring lug to the ground terminal and tighten the grounding screw to 0.5 N · m (4.4 lb · in.) of torque. Attach the other end of the wire to chassis safety ground using a method appropriate for the application, as shown in Figure 1-3. Refer to the *Chassis Grounding Screw* section for more information about earth ground.

Note If you use shielded cabling to connect to a C Series module with a plastic connector, you must attach the cable shield to the chassis grounding terminal using 1.31 mm^2 (16 AWG) or larger wire. Use shorter wire for better EMC performance.

Figure 1-3. Ring Lug Attached to Ground Terminal



Note Make sure that no I/O-side power is connected to the module. If the controller is in a nonhazardous location, the controller power can be on when you install modules.

6. Align the module with a cDAQ controller slot.
7. Squeeze the latches and insert the module into the module slot, and press firmly on the connector side of the module until the latches lock the module into place.

   Repeat Steps 6 and 7 to install additional modules.
8. Wire your external power source and install the remaining ferrite as outlined in the *Wiring Power to the cDAQ Controller* section. The cDAQ controller requires an external power supply that meets the specifications listed in the specifications document for your cDAQ controller.
9. Turn on the external power supply.

   When the cDAQ controller powers on, the POWER LED lights and the controller runs a power-on self test (POST). When the POST is complete, the operating system is loaded.
10. Go through the steps on the Set Up Windows screen that opens on your monitor. Windows prepares your desktop.

11. Wire the C Series module as indicated in the C Series module documentation, available from `ni.com/manuals`.
12. Self-test your controller in Measurement & Automation Explorer (NI MAX) by double-clicking the NI MAX icon on the desktop to open MAX. Expand **Devices and Interfaces**, right-click **NI cDAQ-<*model number*>**, and select **Self-Test**. Self-test performs a brief test to determine successful controller installation.
13. Run a Test Panel in MAX by expanding **Devices and Interfaces»NI cDAQ-<*model number*>**, right-clicking your C Series module, and selecting **Test Panels** to open a test panel for the selected module.

    If the test panel displays an error message, refer to `ni.com/support`.

New users can view and use the Voltage - Continuous Input VI, available in the LabVIEW Example Finder. Experienced users can use the LabVIEW Sample Projects, Finite Measurement (NI-DAQmx) and Continuous Measurement and Logging (NI-DAQmx).

Caution Removing power without shutting down the cDAQ controller can corrupt the embedded Windows system drive. For information about how to improve robustness on the Windows system, go to `ni.com/info` and enter the Info Code `extxxx`.

Note When in use, the cDAQ controller may become warm to the touch. This is normal.

Note The network behavior is determined by the Windows network drivers. Refer to the Windows documentation for information about configuring IP settings.

Installing the cDAQ Controller for LabVIEW Real-Time

(NI cDAQ-9132/9133/9134/9135/9136/9137 for LabVIEW Real-Time) The NI cDAQ-9132/9133/9134/9135/9136/9137 for LabVIEW Real-Time features a hard drive formatted for LabVIEW Real-Time. The cDAQ controller and C Series module(s) are packaged separately.

You will need the following items to set up the NI cDAQ-9132/9133/9134/9135/9136/9137 for LabVIEW Real-Time controller:

- Power connector (packaged with the cDAQ controller)
- USB cable (packaged with the cDAQ controller)
- Ferrites (packaged with the cDAQ controller)
- Host computer running Windows (check your driver and ADE readme files for specific version compatibility)
- LabVIEW software

- LabVIEW Real-Time software
- NI-DAQmx driver (packaged with the cDAQ controller)
- **(NI cDAQ-9134/9135)** NI-XNET software (packaged with the cDAQ controller)
- Power supply
- Number 1 and number 2 Phillips screwdrivers
- C Series module(s)

Refer to Figure 1-1 or 1-2 while completing the following assembly steps.

1. Install LabVIEW on your host computer, as described in the *LabVIEW Installation Guide*.
2. Install LabVIEW Real-Time on your host computer, as described in the *LabVIEW Real-Time Module Release and Upgrade Notes*.
3. Install NI-DAQmx driver software on your host computer, as described in the *Read Me First: NI-DAQmx and DAQ Device Installation Guide*.

Note Table 1-2 lists the earliest supported driver version for each cDAQ controller for LabVIEW for Real-Time.

Table 1-2. cDAQ Controller NI-DAQmx Software Support

cDAQ Controller	Earliest NI-DAQmx Support
NI cDAQ-9132/9134 for LabVIEW Real-Time	NI-DAQmx 14.1
NI cDAQ-9133/9135 for LabVIEW Real-Time	NI-DAQmx 14.5
NI cDAQ-9136/9137 for LabVIEW Real-Time	NI-DAQmx 15.1

The NI-DAQmx driver software is included on the media shipped with your kit and is available for download at `ni.com/support`. The documentation for NI-DAQmx is available after installation from **Start»All Programs»National Instruments» NI-DAQmx**.

4. **(NI cDAQ-9134/9135)** Install NI-XNET on your host computer, as described in the *NI-XNET Hardware and Software Installation Guide*.

Note Table 1-3 lists the earliest supported driver version for each cDAQ controller for LabVIEW Real-Time.

Table 1-3. cDAQ Controller NI-XNET Software Support

cDAQ Controller	Earliest NI-XNET Support
NI cDAQ-9134 for LabVIEW Real-Time	NI-XNET 14.1
NI cDAQ-9135 for LabVIEW Real-Time	NI-XNET 14.5

5. Power on the host computer.
6. (Optional) Mount the cDAQ controller to a panel, wall, rack, or DIN rail, or attach the desktop mounting kit, as described in the *Mounting the cDAQ Controller* section.
7. Attach a ring lug to a 1.31 mm^2 (16 AWG) or larger wire. Remove the ground screw from the ground terminal on the front panel. Attach the ring lug to the ground terminal and tighten the grounding screw to 0.5 N · m (4.4 lb · in.) of torque. Attach the other end of the wire to chassis safety ground using a method appropriate for the application, as shown in Figure 1-3. Refer to the *Chassis Grounding Screw* section for more information about earth ground.

Note If you use shielded cabling to connect to a C Series module with a plastic connector, you must attach the cable shield to the chassis grounding terminal using 1.31 mm^2 (16 AWG) or larger wire. Use shorter wire for better EMC performance.

Note Make sure that no I/O-side power is connected to the module. If the controller is in a nonhazardous location, the controller power can be on when you install modules.

8. Align the module with a cDAQ controller slot.
9. Squeeze the latches and insert the module into the module slot, and press firmly on the connector side of the module until the latches lock the module into place.

   Repeat Steps 8 and 9 to install additional modules.
10. Wire your external power source and install the remaining ferrite as outlined in the *Wiring Power to the cDAQ Controller* section. The cDAQ controller requires an external power supply that meets the specifications listed in the specifications document for your cDAQ controller.
11. Turn on the external power supply.

    When the cDAQ controller powers on, the POWER LED lights and the controller runs a power-on self test (POST). When the POST is complete, the operating system is loaded.
12. Use a USB A-to-B cable (included in the shipping kit) to connect the USB device port of the cDAQ controller to a USB port on the host computer.

Caution National Instruments requires a locking USB cable, such as part number 157788-01, in order to meet the shock and vibration specifications of this product.

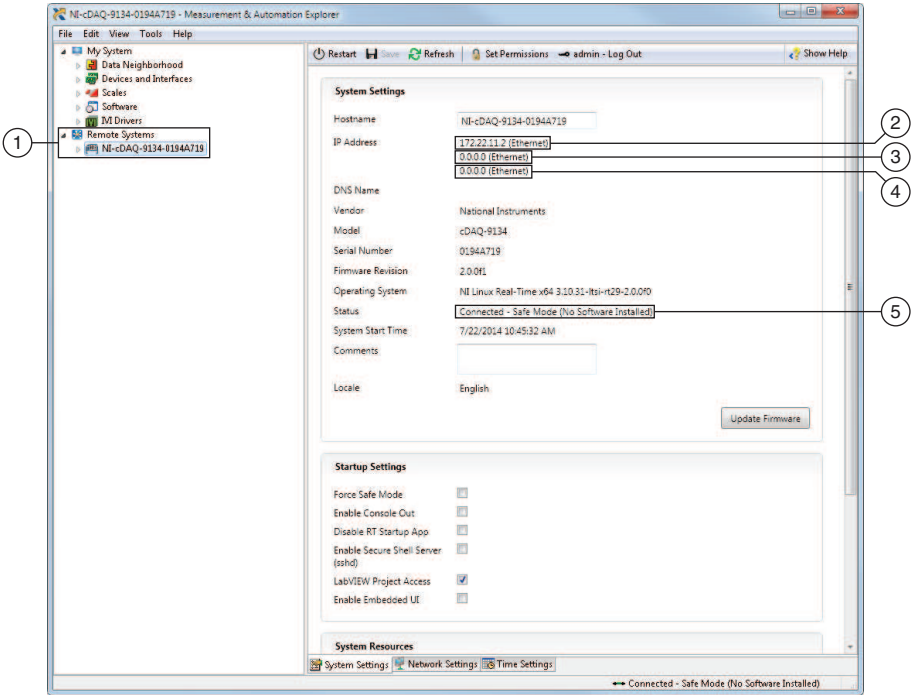
Note Alternatively, you can connect the cDAQ controller to the host computer using RJ-45 Ethernet port 1. Refer to the *Connecting to the Network through the Ethernet Port* section for more information.

13. Wire the C Series module as indicated in the C Series module documentation.

14. Launch Measurement & Automation Explorer (MAX) by double-clicking the NI MAX icon on the host computer desktop. Expand **Remote Systems** and select **NI-cDAQ<*model number*>-<*serial number*>**.

    Click the **System Settings** tab and verify that the System State reads **`Connected - Safe Mode (No Software Installed)`**.

Figure 1-4. cDAQ Controller System Settings in MAX

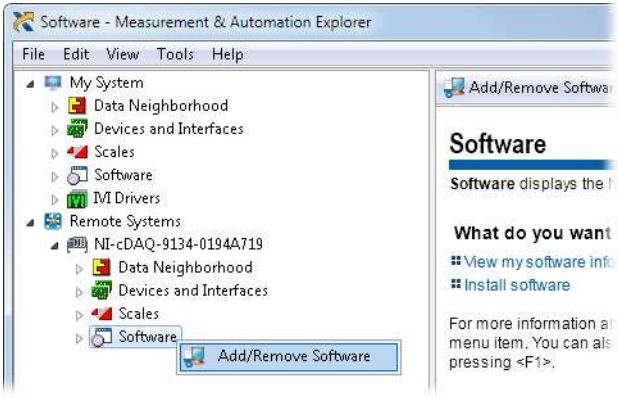


1	cDAQ Controller under Remote Systems	4	Ethernet Port 2 IP Address
2	USB Device Port IP Address	5	Controller Status
3	Ethernet Port 1 IP Address		

Note If you are connecting through the RJ-45 Ethernet port instead of the USB device port and cannot find the system in the MAX configuration tree, refer to the *Troubleshooting Network Communication in the LabVIEW Real-Time Controller* section.

15. Expand **NI-cDAQ<*model number*>-<*serial number*>**. Right-click **Software** and select **Add/Remove Software**.

Figure 1-5. Adding Software in MAX



16. Click **OK** in the Log In window. The password is blank by default.
17. In the window that opens, select **NI-DAQmx**, and then select **Install the feature**. Other required dependencies will be selected automatically.
18. **(NI cDAQ-9134/9135)** Select **NI-XNET** and then select **Install the feature**.
19. Click **Next** to confirm the requested software features.
20. Click **Next** to install the software. After the installation completes, the cDAQ controller reboots.
21. Click **Finish**.
22. In MAX, expand **Remote Systems** and select **NI-cDAQ<*model number*>-<*serial number*>**. Click the **System Settings** tab and verify that the System State reads `Connected - Running`.
23. Self-test your controller in MAX by expanding **NI-cDAQ<*model number*>-<*serial number*>»Devices and Interfaces**. Right-click **NI cDAQ-<*model number*>** and select **Self-Test**. Self-test performs a brief test to determine successful controller installation.
24. Run a Test Panel in MAX by expanding **NI-cDAQ<*model number*>-<*serial number*>» Devices and Interfaces»NI cDAQ-<*model number*>**, right-clicking your C Series module, and selecting **Test Panels** to open a test panel for the selected module.

    If the test panel displays an error message, refer to `ni.com/support`.

New users can view and use the Voltage - Continuous Input VI, available in the LabVIEW Example Finder. Experienced users can use the LabVIEW Sample Projects, LabVIEW Real-Time Control (NI-DAQmx) and LabVIEW Waveform Acquisition and Logging (NI-DAQmx).

Note You can configure network settings using MAX on a host computer. Refer to the *LabVIEW Real-Time Target Configuration* topic of the *Measurement & Automation Explorer Help* for information.

Note You can configure the cDAQ controller to launch an embedded stand-alone LabVIEW RT application each time you boot the controller. Refer to the *Building and Deploying a Stand-Alone Real-Time Application* topic of the *LabVIEW Real-Time Module Help* for more information about startup applications.

Connecting to the Network through the Ethernet Port

(NI cDAQ-9132/9133/9134/9135/9136/9137 for LabVIEW Real-Time) Use a shielded straight through Category 5 Ethernet cable to connect the cDAQ chassis to an Ethernet network.[1] Connect one end to RJ-45 Ethernet port 1 on the controller, and the other end directly to your computer or any network connection on the same subnet as your computer. Refer to the *Ethernet Cabling* section for information about the Ethernet cable.

Caution To prevent data loss and to maintain the integrity of your Ethernet installation, do *not* use a cable longer than 100 m.

The cDAQ controller attempts to initiate a DHCP network connection at powerup. If the cDAQ controller is unable to obtain an IP address, it connects to the network with a link-local IP address with the form `169.254.x.x`. The host computer communicates with the cDAQ controller over a standard Ethernet connection.

Troubleshooting Network Communication in the LabVIEW Real-Time Controller

(NI cDAQ-9132/9133/9134/9135/9136/9137 for LabVIEW Real-Time) If the cDAQ controller cannot communicate with the network, you can perform the following troubleshooting steps.

Note Windows XP users may be required to manually install the USB driver on the host computer. The USB driver is installed in the `National Instruments\CompactRIO\Staging\USBLAN` directory.

1. Use a USB A-to-B cable to connect the USB device port of the cDAQ controller shown in Figure 1-1, to a USB port on the host computer.

   When you connect the cDAQ controller to the host computer, the USB driver creates a virtual network interface card and assigns an IP address to the cDAQ controller in the format of `172.22.11.x`.

[1] You can either use a shielded straight through Category 5 Ethernet cable or an Ethernet crossover cable to connect the cDAQ chassis directly to your computer.

2. Launch MAX. Expand **Remote Systems** and select **NI-cDAQ<*model number*>-<*serial number*>**. Configure the IP and other network settings in MAX.
3. (Optional) At this point, you can reconnect the cDAQ controller to the host computer using RJ-45 Ethernet port 1. The cDAQ controller attempts to initiate a DHCP network connection at powerup. If the cDAQ controller is unable to obtain an IP address, it connects to the network with a link-local IP address with the form `169.254.x.x`. The host computer communicates with the cDAQ controller over a standard Ethernet connection.

If you are still experiencing networking issues, complete the following steps.

1. Hold the RESET button down for five seconds and then release it. The STATUS LED lights and then starts blinking three times every few seconds. The controller is now in safe mode with output from the serial port enabled. You can use a RS-232 serial port terminal to read the IP address of the controller or you can connect a monitor to the mini DisplayPort and view the IP address.
2. To set a new DHCP connection, hold the RESET button down for five seconds and then release it. The STATUS LED repeats the same behavior. The cDAQ controller attempts to establish a new DHCP connection. If it fails, it assigns itself a link-local IP address. If the DHCP connection is successful and appropriate for your application, skip to step 4.
3. Launch MAX. Expand **Remote Systems** and select **NI-cDAQ<*model number*>-<*serial number*>**. Configure the IP and other network settings in MAX.
4. Press and release the RESET button to reboot the controller.

For more information about troubleshooting network communication, refer to the *MAX Remote Systems Help* or *Finding a Network DAQ Device in MAX* topic in the *Measurement & Automation Explorer Help*.

Wiring Power to the cDAQ Controller

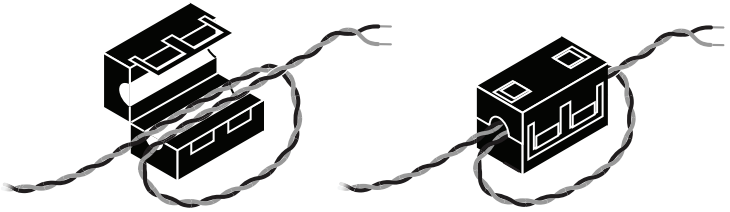
The cDAQ controller requires an external power source as described in the *Power Requirements* section of the specifications document for your cDAQ controller. Some suggested NI power supplies are listed in Table 1-10. The cDAQ controller filters and regulates the supplied power and provides power to all of the modules. The cDAQ controller has a primary power input, V1, and a secondary power input, V2. The POWER LED on the front panel identifies the power input in use. When the LED is lit green, V1 is in use; when the LED is lit yellow, V2 is in use.

Caution Do *not* connect V2 to a DC mains supply or to any supply requiring a connecting cable longer than 3 m (10 ft). A DC mains supply is a local DC electricity supply network in the infrastructure of a site or building.

Complete the following steps to connect a power source to the cDAQ controller.

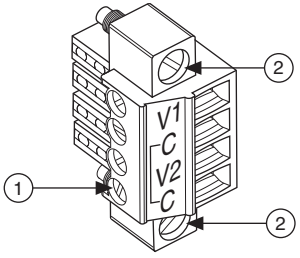
1. Make sure the power source is turned off.
2. Install the ferrite (National Instruments part number 711849-01, included in the shipping kit) across the negative and positive leads of the power source, approximately 50 to 75 mm (2 to 3 in.) from the ends of the leads near the cDAQ controller, as shown in Figure 1-6.

Figure 1-6. Installing the Ferrite on the Power Leads



3. Loosen the connector screw flanges and remove the power screw terminal connector plug from the cDAQ controller. Figure 1-7 shows the terminal screws, which secure the wires in the screw terminals, and the connector screw flanges, which secure the connector plug on the front panel.

Figure 1-7. Power Screw Terminal Connector Plug



1	Terminal Screw	2	Connector Screw Flanges

Caution Do *not* tighten or loosen the terminal screws on the power connector while the power is on.

4. Connect the positive lead of the primary power source to the V1 terminal of the power connector plug and tighten the terminal screw.
5. Connect the negative lead of the primary power source to one of the C terminals of the power screw terminal connector plug and tighten the terminal screw.
6. Optionally, you can connect the positive lead of a secondary power source to the V2 terminal and the negative lead to the other C terminal.
7. Install the power connector plug on the front panel of the cDAQ controller and tighten the connector screw flanges.
8. Turn on the external power source(s).

The cDAQ controller uses V1 if the voltage across V1 and C is 9 V or greater. If the V1-to-C voltage drops below 9 V, the cDAQ controller switches to V2. If the V2-to-C voltage is less than 9 V, operation may be interrupted.

Note If the cDAQ controller is using V1 and a secondary power source is connected to V2, there is a small leakage current on V2. The leakage current depends on the V2-to-C voltage. Refer to the *Power Requirements* section of the specifications document for your cDAQ controller for nominal values of this leakage current.

If the power source is connected to the power connector using long wiring with high DC resistance, the voltage at the power connector may be significantly lower than the specified voltage of the power source.

The C terminals are internally connected to each other but are not connected to chassis ground. You can connect the C terminals to chassis ground externally. Refer to the *Power Requirements* section of the specifications document for your cDAQ controller for information about the power supply input range. Refer to the *Safety Voltages* section of the specifications document for your cDAQ controller for information about the maximum voltage from terminal to chassis ground.

Powering Down the cDAQ Controller

There are two methods for safely powering down the cDAQ controller:

- **Power Button**—Pressing and releasing the power button, shown in Figure 1-1 or 1-2, shuts down the cDAQ controller. This default behavior for the cDAQ controller is configured in the BIOS and the Windows operating system.
- **Windows Software**—Shut down the cDAQ-9132/9133/9134/9135/9136/9137 for Windows controller through the Windows start menu.

Caution Removing power without shutting down the cDAQ controller can corrupt the embedded Windows system drive. For information about how to improve robustness on the Windows system, go to `ni.com/info` and enter the Info Code `extxxx`.

Controller Startup Options

(NI cDAQ-9132/9133/9134/9135/9136/9137 for LabVIEW Real-Time) You can configure controller startup options in MAX. Select the controller under **Remote Systems** in the MAX configuration tree and then select the **System Settings** tab. You can configure the following options under **Startup Settings**.

- **Safe Mode**—When you reboot the controller with this setting on, the controller starts without launching LabVIEW RT or any startup applications. In safe mode the controller launches only the services necessary for updating configuration and installing software.

- **Console Out**—When you reboot the controller with this setting on, the controller redirects output to the RS-232 serial port. You can use a serial-port terminal program to read the IP address and firmware version of the controller. Use a null-modem cable to connect the RS-232 serial port to a computer. Make sure that the serial-port terminal program is configured to the following settings:
  - 115,200 bits per second
  - Eight data bits
  - No parity
  - One stop bit
  - No flow control
- **Disable RT Startup App**—Rebooting the controller with this setting on prevents any LabVIEW startup applications from running.
- **Enable Secure Shell (SSH) Logins**—Rebooting the controller with this setting on starts `sshd` on the controller. Starting `sshd` enables logins over SSH, an encrypted communication protocol. For information about SSH, go to `ni.com/info` and enter the Info Code `openssh`.
- **Embedded User Interface**—You can use a single real-time VI to iteratively develop both your user interface and system logic. For more information, refer to the *Using the Embedded UI to Access RT Target VIs* topic in the *LabVIEW Help*.

Removing Modules from the cDAQ Controller

Complete the following steps to remove a C Series module from the cDAQ controller.

1. Make sure that no I/O-side power is connected to the module. If the controller is in a nonhazardous location, the controller power can be on when you remove modules.
2. Squeeze the latches on both sides of the module and pull the module out of the controller.

Mounting the cDAQ Controller

You can use the cDAQ controller on a desktop or mount it to a panel, wall, DIN rail, or rack. For accessory ordering information, refer to the pricing section of the cDAQ controller product page at `ni.com`.

Note The cDAQ controller was designed and tested in multiple mounting configurations. The varied mounting orientations or configurations can reduce the maximum allowable ambient temperature and can affect the accuracy of C Series modules in the controller. Visit `ni.com/info` and enter the Info Code `cdaqmounting` for more information about mounting and accuracy.

The following sections contain mounting method information. Before using any of these mounting methods, record the serial number from the side of the controller. You may be unable to read the serial number after you have mounted the controller.

Caution You must mount the controller horizontally on a flat, vertical, metallic surface to achieve the maximum allowable operating ambient temperature[1]. Mounting the controller in a different orientation or on a nonmetallic surface reduces the maximum allowable ambient temperature and can affect the measurement accuracy of modules in the controller. Figures 1-8 and 1-9 show the controller mounted horizontally. Refer to the *Mounting the cDAQ Controller* section for complete panel mounting instructions.

Measure the ambient temperature at each side of the controller, 63.5 mm (2.5 in.) from the side and 38.1 mm (1.5 in.) forward from the rear of the controller, as shown in Figures 1-10 and 1-11.

For more information about how different mounting configurations can cause temperature derating, go to `ni.com/info` and enter the Info Code `cdaqmounting`.

Figure 1-8. NI cDAQ-9132/9134/9136 Mounted Horizontally with Panel Mount Kit

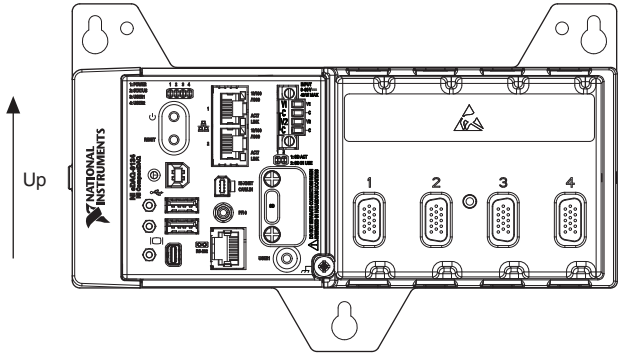
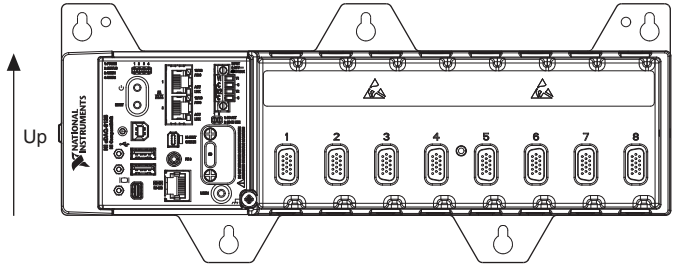


Figure 1-9. NI cDAQ-9133/9135/9137 Mounted Horizontally with Panel Mount Kit



[1] The allowable operating ambient temperature for the cDAQ-9132/9133/9136/9137 is -20 to 55 °C. The allowable operating ambient temperature for the cDAQ-9134/9135 is -40 to 70 °C.

Figure 1-10. NI cDAQ-9132/9134/9136 Temperature, Cooling, and Cabling Dimensions (NI cDAQ-9134 Shown)

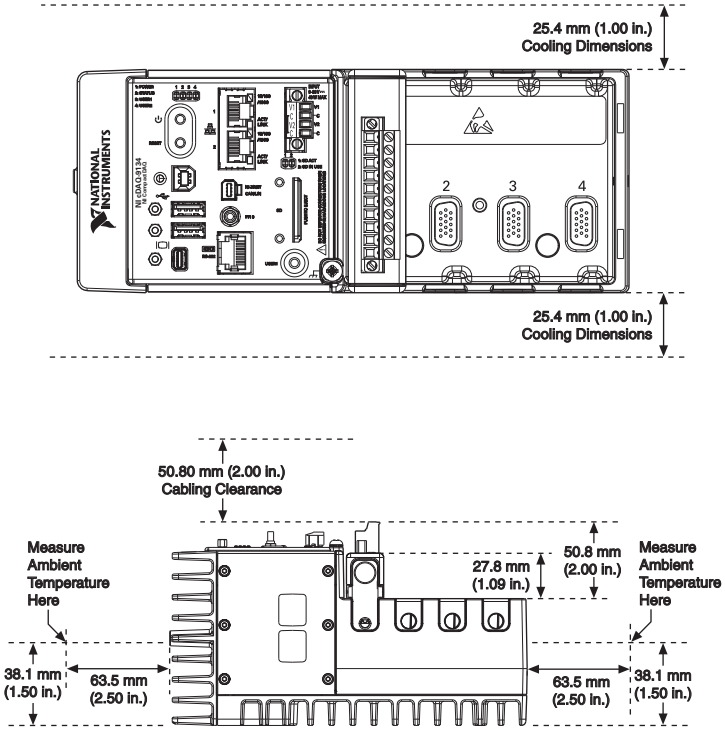
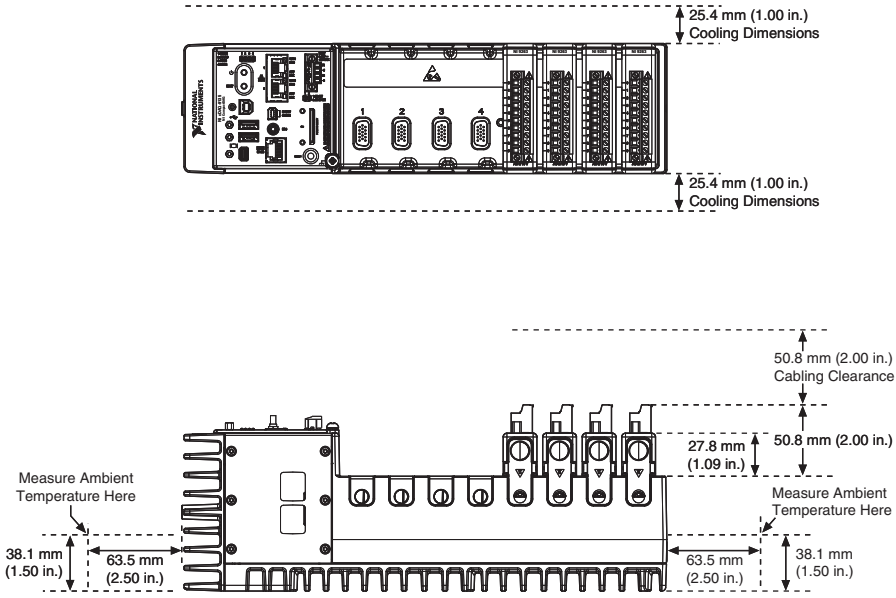


Figure 1-11. NI cDAQ-9133/9135/9137 Temperature, Cooling, and Cabling Dimensions (NI cDAQ-9135 Shown)



Caution Your installation must meet the following requirements for space and cabling clearance, as shown in Figures 1-10 and 1-11:

- Allow 25.4 mm (1.00 in.) on the top and the bottom of the controller for air circulation.
- Allow 50.8 mm (2.00 in.) in front of modules for cabling clearance for common connectors, such as the 10-terminal, detachable screw terminal connector.

Mounting the cDAQ Controller on a Panel

Directly mounting the cDAQ controller to a rigid surface is the only recommended method for applications that are subject to high shock and vibration.

You can mount the cDAQ controller without a kit by screwing through a panel into the holes on the back of the controller. Use M4 screws with a length suitable for the depth of the panel.

You can use the NI panel mount kit to mount the cDAQ controller on a flat surface. Refer to the *Cables and Accessories* section for the accessory part number for your cDAQ controller. Complete the following steps.

1. Fasten the mounting plate to the controller using a number 2 Phillips screwdriver and M4 × 10 screws[1]. National Instruments provides these screws with the panel mount kit. Tighten the screws to a maximum torque of 1.3 N · m (11.5 lb · in.).

Figure 1-12. Installing the Mounting Plate on the cDAQ Controller (cDAQ-9133/9135/9137 Shown)

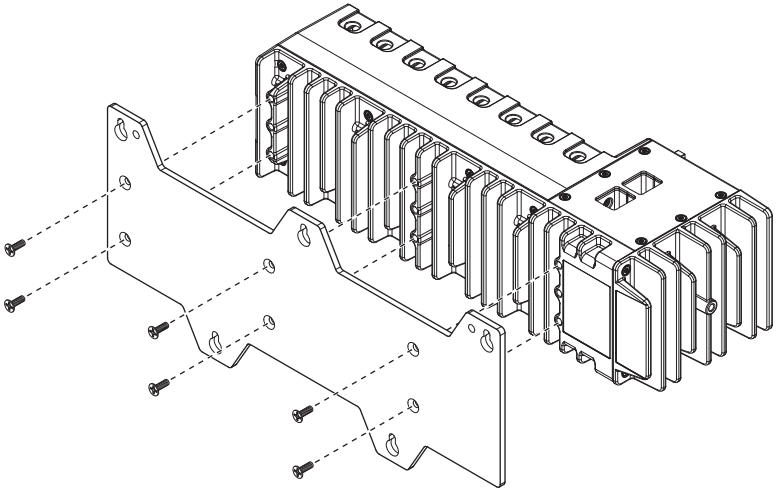
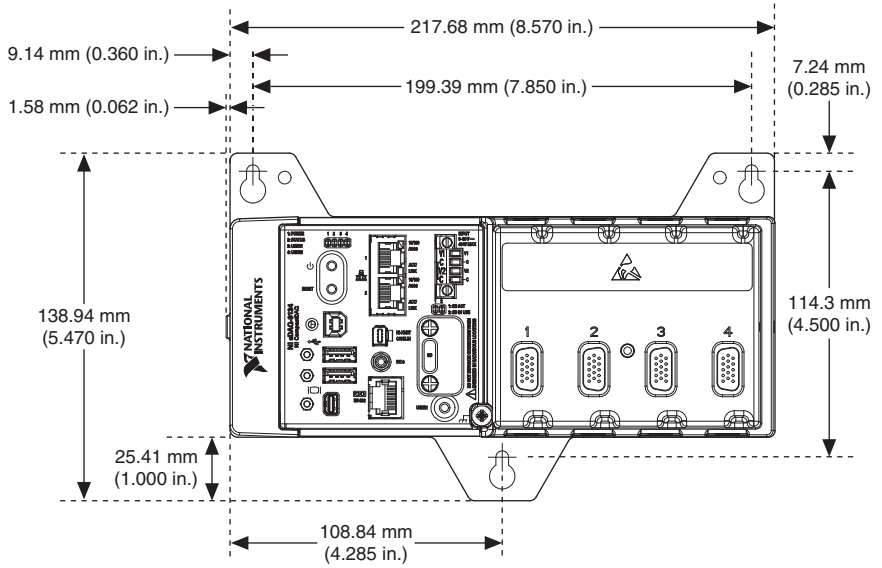
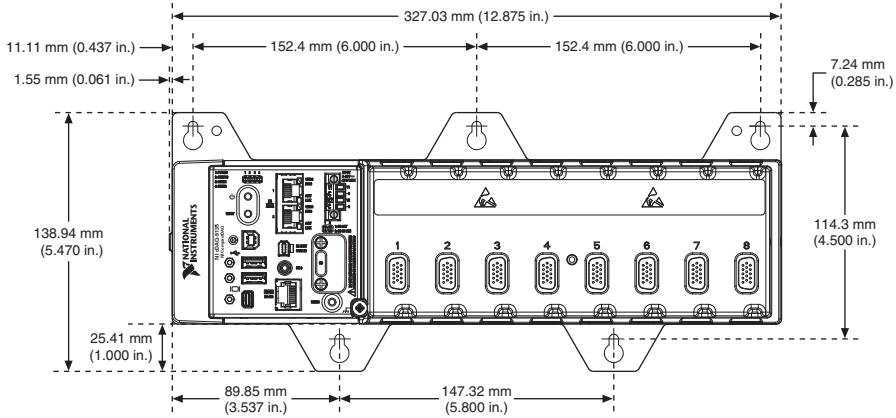


Figure 1-13. Dimensions of the cDAQ-9132/9134/9136 with Mounting Plate Installed



[1] The NI cDAQ-9132/9134/9136 controller panel mounting plate requires four screws. The NI cDAQ-9133/9135/9137 controller panel mounting plate requires six screws.

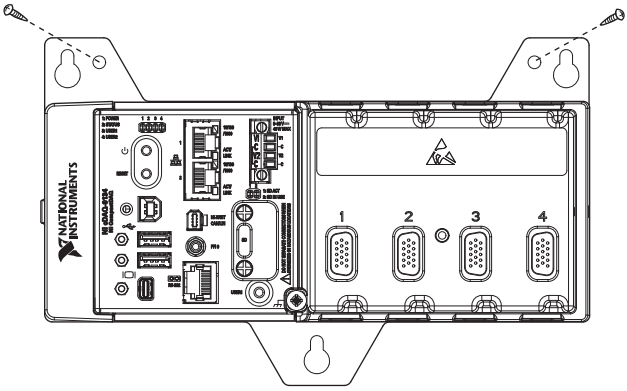
Figure 1-14. Dimensions of the cDAQ-9133/9135/9137 with Mounting Plate Installed



2. Fasten the mounting plate to the surface using the screwdriver and screws that are appropriate for the surface. The maximum screw size is M4 or number 8.

   Optionally, you can use two additional screws to attach the mounting plate to the panel or wall permanently, preventing the controller from being removed.

Figure 1-15. Permanently Attaching the Mounting Plate to the Panel or Wall (cDAQ-9134 Shown)

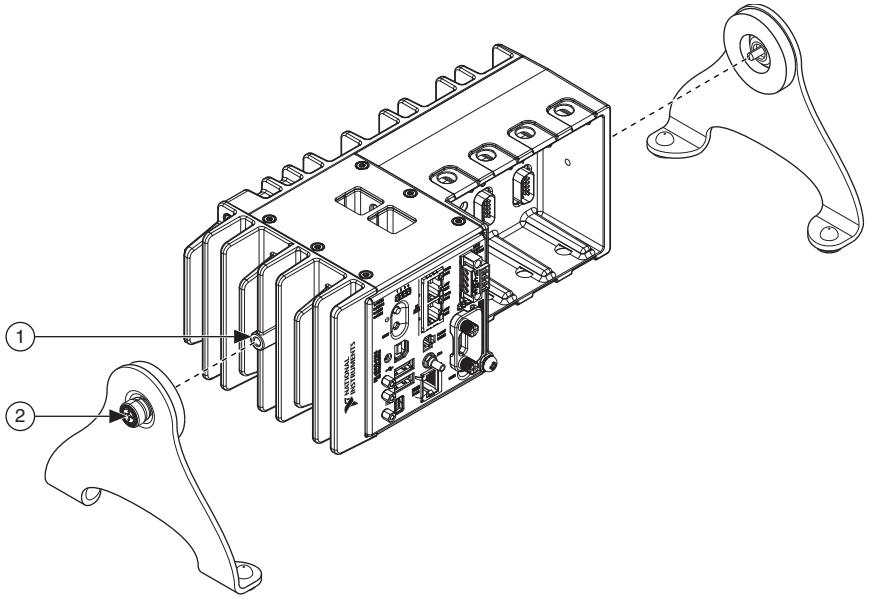


Using the cDAQ Controller on a Desktop

You can install the NI desktop mount kit to the cDAQ controller. Complete the following steps to install the NI desktop mount kit, part number 779473-01, on the cDAQ controller.

1. Align one of the end brackets with the mounting hole at one of the ends of the controller, as shown in Figure 1-16.

Figure 1-16. Connecting the End Brackets to the Controller (cDAQ-9134 Shown)



1	Mounting Holes	2	Captive Screw

2. Use a number 2 Phillips screwdriver to tighten the captive screw on the end bracket.
3. Repeat steps 1 and 2 to attach the other end bracket to the other end of the controller.

Note To achieve the highest accuracy when mounting the controller in the desktop kit, NI recommends that you operate the controller with the modules rotated forward, as shown in Figures 1-17 and 1-18. Visit `ni.com/info` and enter the Info Code `cdaqmounting` for more information about mounting and accuracy.

Figures 1-17 and 1-18 show the dimensions of a controller after the desktop mounting kit is installed.

Figure 1-17. Dimensions of the cDAQ-9132/9134/9136 with Desktop Mounting Kit Installed

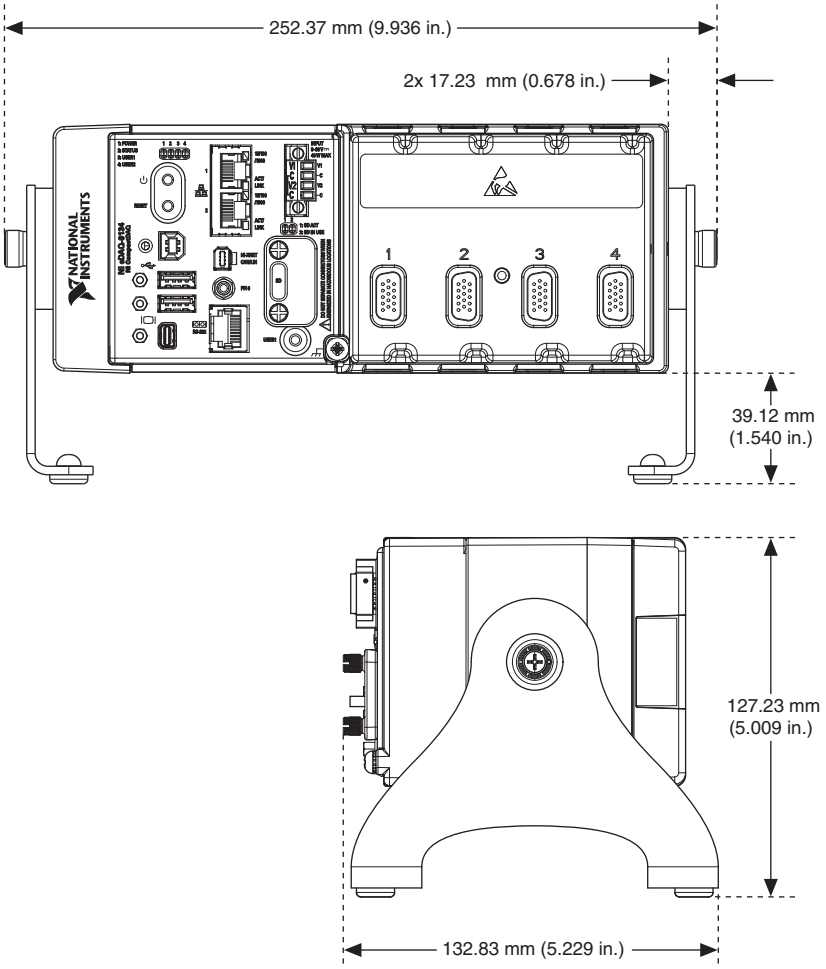
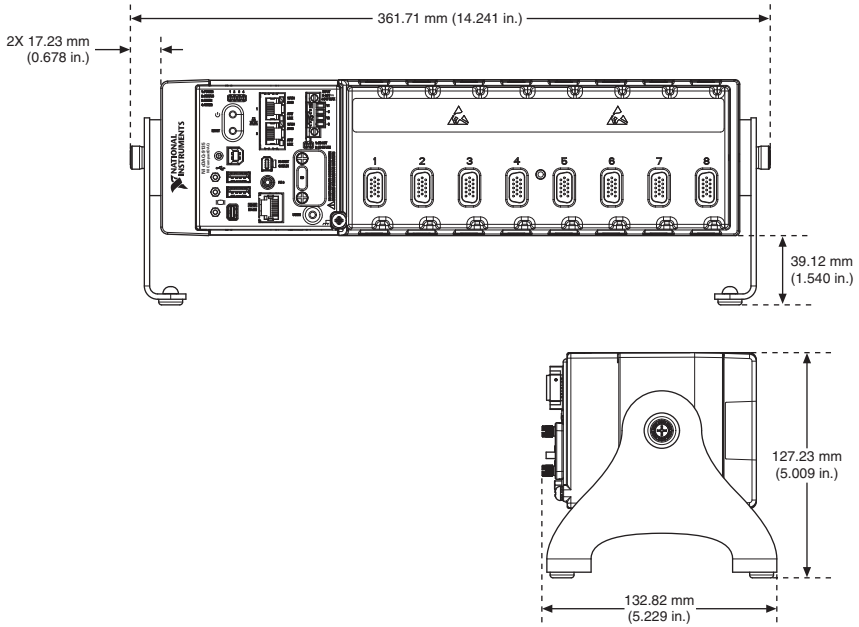


Figure 1-18. Dimensions of the cDAQ-9133/9135/9137 with Desktop Mounting Kit Installed



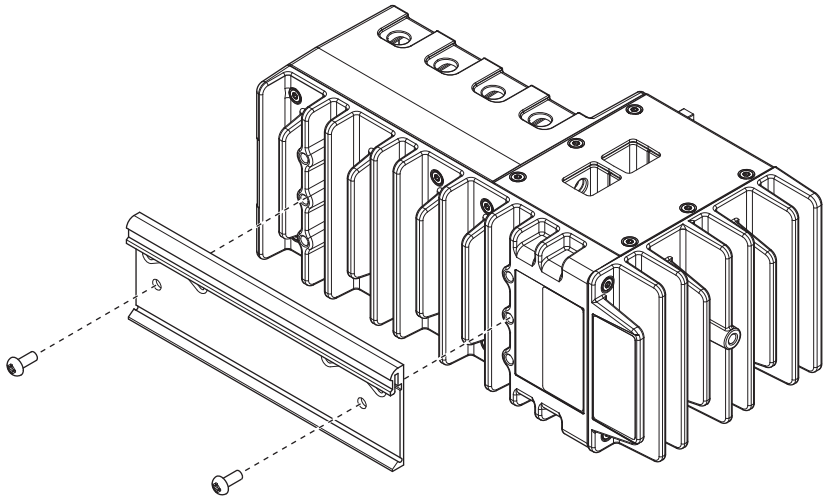
Mounting the cDAQ Controller on a DIN Rail

Use the DIN rail mounting method if you already have a DIN rail configuration or if you need to be able to remove the controller quickly. You can use the NI DIN rail mount kit to mount the controller on a DIN rail. You need one clip for mounting the controller on a standard 35 mm DIN rail. Refer to the *Cables and Accessories* section for the accessory part number for your cDAQ controller. Complete the following steps to mount the controller on a DIN rail.

1. Fasten the DIN rail clip to the controller using a number 2 Phillips screwdriver and M4 × 10 screws[1]. National Instruments provides these screws with the DIN rail mount kit. Tighten the screws to a maximum torque of 1.3 N · m (11.5 lb · in.). Make sure the DIN rail kit is installed as shown in Figure 1-19, with the larger lip of the DIN clip positioned up. When the DIN rail kit is properly installed, the cDAQ controller is centered on the DIN rail.

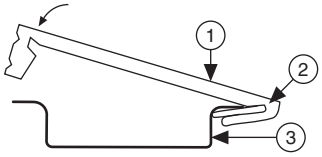
[1] The NI cDAQ-9132/9134/9136 controller DIN rail clip requires two screws.
The NI cDAQ-9133/9135/9137 controller DIN rail clip requires three screws.

Figure 1-19. Installing the DIN Rail Clip on the cDAQ Controller (cDAQ-9132/9134/9136 Shown)



2. Insert one edge of the DIN rail into the deeper opening of the DIN rail clip, as shown in Figure 1-20, and press down firmly on the controller to compress the spring until the clip locks in place on the DIN rail.

Figure 1-20. DIN Rail Clip Parts Locator Diagram



1	DIN Rail Clip	2	DIN Rail Spring	3	DIN Rail

Caution Remove the modules before removing the controller from the DIN rail.

Mounting the cDAQ Controller on a Rack

NI offers two rack mount kits, part numbers 779102-01 and 781989-01, that you can use to mount the cDAQ controller and other DIN rail–mountable equipment on a standard 19-inch rack. You must order the NI DIN rail mount kit in addition to these kits. Refer to the *Cables and Accessories* section for the accessory part number for your cDAQ controller.

Installing the Module Immobilization Accessory

The Module Immobilization accessory, part number 158533-01 (8-slot) or 158534-01 (4-slot), ensures that the C Series module latches cannot be retracted and modules cannot be removed from a system. The Module Immobilization accessory provides extra system assurance and security when shipping and installing systems, and prevents accidental removal from a system during operation.

When using the Module Immobilization accessory, NI recommends installing the accessory prior to mounting the system in any enclosure because the accessory requires tool access to the top, right, and bottom of the cDAQ controller.

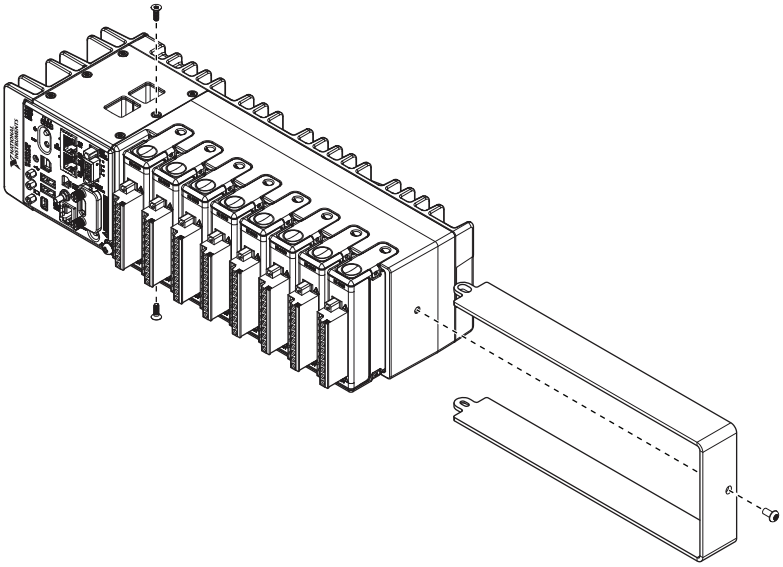
What to use:

- cDAQ controller
- C Series modules
- Module Immobilization accessory kit, 158533-01 (8-slot) or 158534-01 (4-slot)
  - Module immobilization bracket
  - Installation screws[1]
    - M4 × 0.7 button-head screw, 8 mm
    - M3 × 0.5 flat-head screws (x2), 10 mm
- Torx T10/T10H driver
- Torx T20/T20H driver

[1] The Module Immobilization accessory kit includes two sets of screws. One set is a standard set of screws that require a standard driver type, Torx T10 and T20. The other set is a tamper-resistant set of screws that require a security driver type, Torx T10H and T20H. Use the tamper-resistant set to help prevent unintended modification of the system.

Complete the following steps to install the Module Immobilization accessory.

Figure 1-21. Module Immobilization Accessory Installation



1. Ensure that all the C Series modules are installed in the cDAQ controller and the latches are locked in place.
2. Remove the center right panel screw from the top and bottom of the cDAQ controller using the Torx T10 driver.
3. Slide the bracket into place, aligning the three clearance screw holes.
4. Install the M4 × 0.7 button-head screw in the right end of the cDAQ controller using the appropriate Torx T20 driver. Tighten the screw to a maximum torque of 1.3 N · m (11.5 lb · in.).
5. Install the two M3 × 0.5 flat-head screws from the accessory kit in the top and bottom of the cDAQ controller using the appropriate Torx T10 driver. Tighten the screws to a maximum torque of 1.3 N · m (11.5 lb · in.).

Tip NI recommends using a liquid thread locker for all fasteners if the system is expected to experience vibration for an extended amount or time.

Figures 1-22 and 1-23 show the dimensions of the 4-slot and 8-slot cDAQ controllers with installed Module Immobilization accessory.

Figure 1-22. Dimensions of the cDAQ-9132/9134/9136 with Module Immobilization Accessory Installed

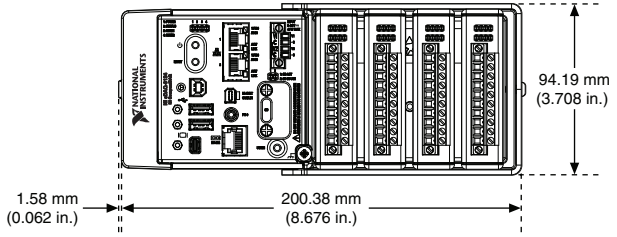
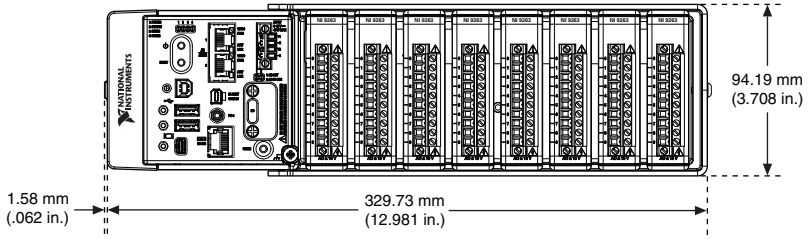


Figure 1-23. Dimensions of the cDAQ-9133/9135/9137 with Module Immobilization Accessory Installed



cDAQ Controller Features

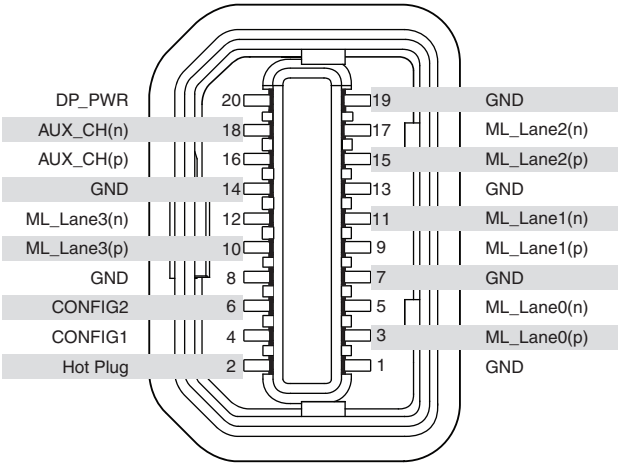
The cDAQ controller features many ports, LEDs, a RESET button, a power button, and an SD card slot. Refer to Figure 1-1 or 1-2 for the locations of these features on the cDAQ controller.

Mini DisplayPort Connector

Use the mini DisplayPort connector, shown in Figure 1-1 or 1-2, to connect a monitor to program Windows cDAQ controllers or implement a local HMI for Real-Time cDAQ controllers. You can use a single real-time VI to iteratively develop both your user interface and system logic. For more information, refer to the *Using the Embedded UI to Access RT Target VIs* topic in the *LabVIEW Help*.

Figure 1-24 lists the mini DisplayPort pins and signals. Refer to the *Cables and Accessories* section for information about supported NI cables and accessories for the cDAQ controller.

Figure 1-24. Mini DisplayPort Pinout



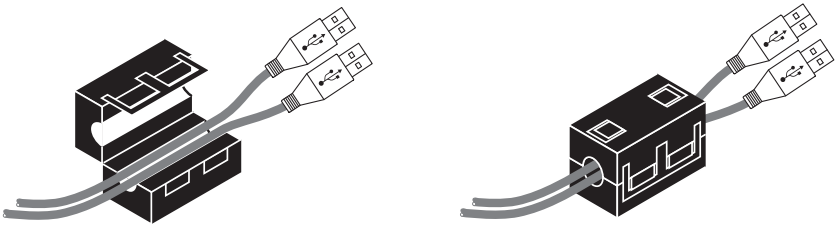
Caution Do *not* hot-swap mini DisplayPort devices while the cDAQ controller is in a hazardous location or connected to high voltages.

USB Host Ports

The cDAQ controller supports common USB mass-storage devices such as USB Flash drives and USB-to-IDE adapters formatted with FAT16 and FAT32 file systems. LabVIEW usually maps USB devices to the `U:`, `V:`, `W:`, or `X:` drive, starting with the `U:` drive if it is available. You can also use these ports to connect a computer keyboard and mouse for controller programming.

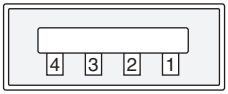
Install a noise-suppression ferrite (National Instruments part number 711849-01, included in the shipping kit) around all attached external USB cables to ensure that your device meets all EMC standards applicable to your country, as shown in Figure 1-25. The ferrite should be installed so that it is approximately 50 to 75 mm (2 to 3 in.) from the end of the cable that plugs into the USB host port. The ferrite should accommodate both USB cables depending on cable diameter.

Figure 1-25. Installing a Ferrite on Two USB Cables



Refer to Figure 1-1 or 1-2 for the location of the two USB host ports on the cDAQ controller. Refer to Table 1-4 for USB host port pin locations and signal descriptions.

Table 1-4. USB Host Port Pin Locations

Pinout	Pin	Signal Name	Signal Description
	1	VCC	Cable power (+5 V)
	2	D-	USB data-
	3	D+	USB data+
	4	GND	Ground

Caution Do *not* hot-swap USB devices while the cDAQ controller is in a hazardous location or connected to high voltages. If the cDAQ controller is not in a hazardous location, you can connect and disconnect USB devices without affecting operation.

USB Device Port

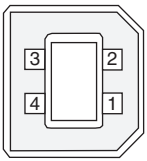
(NI cDAQ-9132/9133/9134/9135/9136/9137 for LabVIEW Real-Time) The USB device port, shown in Figure 1-1 or 1-2, is intended for use in device configuration, application deployment, debug, and maintenance. For example, you can install software or driver updates through the USB device port during field maintenance instead of interrupting communication on the RJ-45 Ethernet ports.

Caution National Instruments requires a locking USB cable, such as part number 157788-01, in order to meet the shock and vibration specifications of this product.

Note The USB device port is not supported on NI cDAQ-9132/9133/9134/9135/9136/9137 for Windows controllers.

Refer to Table 1-5 for USB device port pin locations and signal descriptions.

Table 1-5. USB Device Port Pin Locations

Pinout	Pin	Signal Name	Signal Description
	1	VCC	Cable power (+5 V)
	2	D-	USB data-
	3	D+	USB data+
	4	GND	Ground

Caution Do *not* hot-swap USB devices while the cDAQ controller is in a hazardous location or connected to high voltages. If the cDAQ controller is not in a hazardous location, you can connect and disconnect USB devices without affecting operation.

RESET Button

Pressing the RESET button, shown in Figure 1-1 or 1-2, resets the processor in the same manner as cycling power.

(NI cDAQ-9132/9133/9134/9135/9136/9137 for LabVIEW Real-Time) You can also use the RESET button to troubleshoot network connectivity. Holding the button down for 5 seconds, and then releasing it puts the controller into safe mode. For more information about using the RESET button to put the controller into safe mode and troubleshoot network issues, refer to the *Troubleshooting Network Communication in the LabVIEW Real-Time Controller* section.

Power Button

The default behavior for the cDAQ controller is to power on when power is applied to the controller and power off by pressing and releasing the power button, shown in Figure 1-1 or 1-2. The behavior of the power button can be configured in the BIOS, as described in the *Using the BIOS Setup Utility to Change Configuration Settings* section of Appendix A, *Controller Operating System and BIOS Configuration*. If the cDAQ controller becomes unresponsive, you can power it off by holding the power button down for 4 seconds. For more information about safely powering down the cDAQ controller, refer to the *Powering Down the cDAQ Controller* section.

Caution Removing power without shutting down the cDAQ controller can corrupt the embedded Windows system drive. For information about how to improve robustness on the Windows system, go to `ni.com/info` and enter the Info Code `extxxx`.

LEDs

The cDAQ controller features four LEDs—POWER, STATUS, USER1, and USER2—on its front panel as shown in Figure 1-1 or 1-2. Table 1-6 lists the LEDs and status indications.

Table 1-6. LED Indications

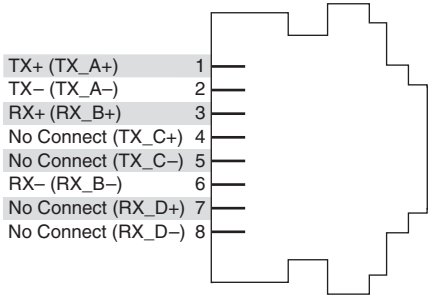
LED	LED Color	LED State	Indication
POWER	Green	Solid	The cDAQ controller is powered from the V1 input.
	Yellow	Solid	The cDAQ controller is powered from the V2 input.
	—	Off	The controller is not powered.
STATUS	Yellow	2 flashes every few seconds	The controller has detected an error in its software. This usually occurs when an attempt to upgrade the software is interrupted. Refer to the *Measurement & Automation Explorer Help* for information about installing software on the controller.
		3 flashes every few seconds	The controller is in safe mode. Refer to the *Measurement & Automation Explorer Help* for information about safe mode.
		4 flashes every few seconds	The software has crashed twice without rebooting or cycling power between crashes. This usually occurs when the controller runs out of memory. Review your RT VI and check the memory usage. Modify the VI as necessary to solve the memory usage issue.
		Continuously flashing	The controller either booted into an unsupported operating system, was interrupted during the boot process, or detected an unrecoverable software error.
		Solid	The controller is booting up.
	Red	Continuously flashing	An internal power supply has failed. Check front-panel I/O and C Series module connections for shorts. Remove any shorts and power cycle the controller. If the problem persists, contact National Instruments.
	—	Off	Normal operation.
USER1, USER2	Green/ yellow	—	USER LEDs are controlled directly from your application. You can define the USER1 and USER2 LEDs to meet the needs of your application. You can use the system hardware property node from the NI System Configuration API to write a state to the USER LEDs. **(NI cDAQ-9132/9133/9134/9135/9136/9137 for LabVIEW Real-Time)** You can also define a USER LED in LabVIEW Real-Time by using the RT LEDs VI. For more information about the RT LEDs VI, refer to the *LabVIEW Help*.

Ethernet Ports

The cDAQ controller has two tri-speed RJ-45 Ethernet ports, shown in Figure 1-1 or 1-2.

Refer to Figure 1-26 for Ethernet pin locations and signal descriptions. The Ethernet signal names are listed as Fast Ethernet signal name, RX/TX +/-, and then Gigabit Ethernet signal name, (RX/TX_*x*+/-).

Figure 1-26. Ethernet Port Pin Locations: Fast Ethernet Signals (Gigabit Ethernet Signals)



Note Both Ethernet ports perform automatic crossover configuration so you do not need to use a crossover cable to connect to a host computer.

(NI cDAQ-9132/9133/9134/9135/9136/9137 for Windows) Both Ethernet ports are enabled and configured as DHCP, to "obtain an IP address automatically," by default. The Ethernet ports can be configured in the Windows Control Panel, under the Network and Internet category. Both Ethernet ports provide Wake-on-LAN functionality. The Ethernet ports remain powered when the controller is in sleep mode.

(NI cDAQ-9132/9133/9134/9135/9136/9137 for LabVIEW Real-Time) Both Ethernet ports are enabled and configured to "obtain an IP address automatically," by default. The Ethernet ports can be configured in MAX

Ethernet LEDs

Each Ethernet port has two LEDs—ACT/LINK and 10/100/1000—described in Table 1-7.

Table 1-7. Ethernet LED Indications

LED	LED Color	LED State	Indication
ACT/LINK	—	Off	LAN link not established
	Green	Solid	LAN link established
		Flashing	Activity on LAN
10/100/1000	Yellow	Solid	1,000 Mbit/s data rate selected
	Green	Solid	100 Mbit/s data rate selected
	—	Off	10 Mbit/s data rate selected

Ethernet Cabling

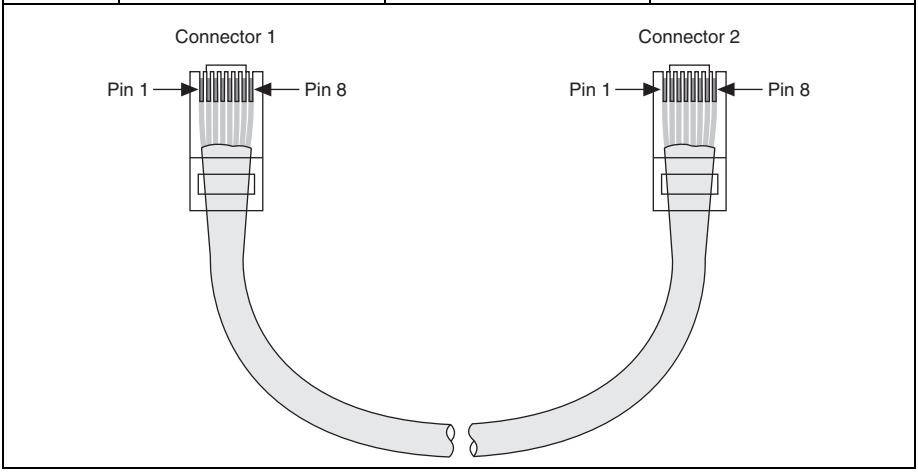
Table 1-8 shows the shielded Ethernet cable wiring connections for both straight through and crossover cables.

Table 1-8. Ethernet Cable Wiring Connections

Pin	Connector 1	Connector 2	
		Straight Through	Crossover
1	white/orange	white/orange	white/green
2	orange	orange	green
3	white/green	white/green	white/orange
4	blue	blue	blue
5	white/blue	white/blue	white/blue
6	green	green	orange

Table 1-8. Ethernet Cable Wiring Connections (Continued)

Pin	Connector 1	Connector 2	
		Straight Through	Crossover
7	white/brown	white/brown	white/brown
8	brown	brown	brown



NI-XNET CAN/LIN Connector

(NI cDAQ-9134/9135) The NI cDAQ-9134/9135 controller features an NI-XNET hardware-selectable interface port that supports CAN and LIN transceiver cables.

Users have the flexibility to choose the physical bus protocol by plugging in corresponding external transceiver cables. The NI-XNET connector supports hot-swapping of transceiver cables and can detect and identify external transceiver cable types. For information about connecting to a CAN or LIN bus, refer to your NI-XNET transceiver cable operating instructions.

Caution To maintain product performance and accuracy specifications when using the NI-XNET CAN/LIN port and NI transceiver cable(s), do not attach or mount the transceiver overmold directly to the cDAQ controller. Mount the overmolded section of the transceiver cable at least 50.8 mm (2.00 in.) away from the cDAQ controller. By following the cooling outline requirements shown in Figures 1-10 and 1-11, the NI transceiver cable will not affect the thermal performance of the system.

Caution Do *not* hot-swap NI-XNET devices while the cDAQ controller is in a hazardous location or connected to high voltages. If the cDAQ controller is not in a

hazardous location, you can connect and disconnect NI-XNET devices without affecting operation

Note **(NI cDAQ-9134/9135 for LabVIEW Real-Time)** To install or upgrade NI-XNET software on the NI cDAQ-9134/9135 for LabVIEW Real-Time controller, you must install the software on the host computer and then deploy it to the controller in MAX, as described in the *Installing the cDAQ Controller for LabVIEW Real-Time* section.

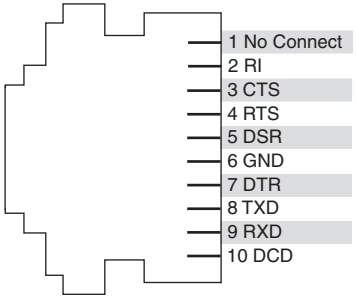
PFI 0 SMB Connector

Refer to the *PFI* section of Chapter 4, *Digital Input/Output and PFI*, for information about the SMB connector for PFI 0.

RS-232 Serial Port

The cDAQ controller has an RS-232 (RJ50) 10-position modular jack, shown in Figure 1-1 or 1-2, to which you can connect devices such as displays or input devices. Use the Serial VIs to read from and write to the serial port. Refer to the *LabVIEW Help* for information about the Serial VIs. Refer to Figure 1-27 for pin locations and signal descriptions.

Figure 1-27. RS-232 Serial Port Pin Locations



You can use the Ring Indicator (RI) on pin 2 to wake the controller from a low power state. You can drive RI with logic-level signals where a high level greater than 2.4 V signals a wake event.

Power Connector

Refer to the *Wiring Power to the cDAQ Controller* section and the specifications document for your cDAQ controller for more information about the power connector.

SD Card Removable Storage

The cDAQ controller features an SD card slot that can read from and write to NI-approved SD cards. Go to `ni.com/info` and enter Info Code `exyerk` for information about best practices for data logging performance with cDAQ controllers.

Caution You *must* use the SD card slot cover to protect the SD card in hazardous locations.

Caution Do *not* insert or remove SD cards unless power has been switched off or the area is known to be nonhazardous.

Note NI recommends that you primarily log data to the SD card when logging data.

Note Using SD cards that are not approved by NI might invalidate specifications and result in unreliable performance.

SD Card LEDs

The cDAQ controller has two LEDs that indicate SD card drive mount status and activity. Refer to Table 1-9 for descriptions of each of the LEDs.

Table 1-9. SD Card LED Indications

LED	LED Color	LED State	Description
SD ACT	Yellow	Off	There is no I/O activity on the SD card in the slot.
		Flashing	The cDAQ controller is performing I/O on the SD card in the slot. Do *not* remove the SD card while this LED is flashing.
SD IN USE	Green	Off	There is no SD card present in the slot or the cDAQ controller has unmounted the SD card from the operating system. It is safe to remove the SD card from the slot.
		Solid	The SD card in the slot is mounted in the operating system. Do *not* remove the SD card while this LED is lit.

SD Card Slot Cover

You must use the SD card slot cover to protect the SD card in hazardous locations. Do not remove an SD card while either LED is flashing or lit because file corruption may result.

Caution Do *not* insert or remove SD cards unless power has been switched off or the area is known to be nonhazardous.

Caution Removing the SD card while the IN USE LED is lit might result in incomplete or lost data.

Note Screw the slot cover closed completely. Tighten the captive screws to a maximum torque of 0.75 N · m (6.7 lb · in.) using a #1 Phillips screwdriver. Do not overtighten.

USER1 Button

The general-purpose USER1 button is user defined. You can use the system hardware property node from the NI System Configuration API to read the state of the USER1 button. For information about programming the USER1 button, go to `ni.com/info` and enter the Info Code `ex4b9n`.

Chassis Grounding Screw

For EMC compliance, the cDAQ controller *must* be connected to earth ground through the chassis ground, shown in Figures 1-1 and 1-2.

The wire should be 1.31 mm^2 (16 AWG) solid copper wire with a maximum length of 1.5 m (5 ft). Attach the wire to the earth ground of the facility's power system. For more information about earth ground connections, go to `ni.com/info` and enter the Info Code `emcground`.

Note If you use shielded cabling to connect to a C Series module with a plastic connector, you must attach the cable shield to the chassis grounding terminal using 1.31 mm^2 (16 AWG) or larger wire. Use shorter wire for better EMC performance.

CMOS Battery and CMOS Reset Button

The cDAQ controller contains a CMOS battery, a lithium cell battery that stores the system clock information when the controller is powered off. There is only a slight drain on the CMOS battery when power is applied to the cDAQ controller power connector. The rate at which the CMOS battery drains when power is disconnected depends on the ambient storage temperature. For longer battery life, store the cDAQ controller at a cooler temperature. Refer to the *CMOS Battery* section of the specifications document for your cDAQ controller for the expected battery lifetime.

The `CMOS BATTERY IS DEAD` warning appears onscreen during the power-on self test if the battery is dead. The controller still starts, but the system clock is reset to the date and time of the BIOS release. The battery is not user replaceable. If you need to replace the CMOS battery, contact National Instruments.

Resetting the System CMOS and BIOS Settings

The cDAQ controller BIOS configuration information is stored in a nonvolatile memory location that does not require a battery to preserve the settings. Additionally, the BIOS optimizes boot time by saving specific system information to memory backed up by a battery (CMOS).

Complete the following steps to reset the CMOS and reset the BIOS settings to factory default values:

1. Disconnect power from the cDAQ controller.
2. Press the CMOS reset button, shown in Figures 1-1 and 1-2, and hold it for 1 second.
3. Reconnect power to the cDAQ controller.

The `BIOS Reset Detected` warning message appears onscreen.

Note If the CMOS battery is dead, the CMOS reset button will not work.

Cables and Accessories

Table 1-10 contains information about cables and accessories available for the cDAQ controller. For a complete list of cDAQ controller accessories and ordering information, refer to the pricing section of the NI cDAQ-9132/9133/9134/9135/9136/9137 product page at `ni.com`.

Caution To ensure the specified EMC performance, operate this product only with shielded cables and accessories.

Table 1-10. Cables and Accessories

Accessory	Part Number	cDAQ Controller
NI PS-15 Power Supply* (24 VDC, 5 A, 100-120/200-240 VAC input)	781093-01	All
NI PS-10 Desktop Power Supply (24 VDC, 5 A, 100-120/200-240 VAC input)	782698-01	All
NI Retention Accessory for Mini DisplayPort	156866-01	All
SD Door Kit	783660-01	All
NI Industrial USB Extender Cable	152166-*xx*	All
NI Locking USB Cable	157788-01	All
Panel Mounting Kit	157253-01	cDAQ-9132/9134/9136
Panel Mounting Kit	157267-01	cDAQ-9133/9135/9137
NI Desktop Mounting Kit	779473-01	All
DIN Rail Mount Kit	157254-01	cDAQ-9132/9134/9136

Table 1-10. Cables and Accessories (Continued)

Accessory	Part Number	cDAQ Controller
DIN Rail Mount Kit	157268-01	cDAQ-9133/9135/9137
Module Immobilization Accessory for 4-Slot CompactRIO and CompactDAQ Controllers	158534-01	cDAQ-9132/9134/9136
Module Immobilization Accessory for 8-Slot CompactRIO and CompactDAQ Controllers	158533-01	cDAQ-9133/9135/9137
NI 9910 Sliding Rack Mount Kit	779102-01	All
NI Rack Mount Kit for cDAQ/cRIO	781989-01	All
NI Industrial USB Extender Cable	152166-*xx*	All
NI Locking USB Cable	157788-01	All
CAT-5E Ethernet Cable, shielded (2, 5, and 10 m lengths)	151733-02, 151733-05, 151733-10	All
Cable Adapters for 10-position Modular Jacks (1, 2, and 3 m lengths)	182845-01, 182845-02, 182845-03	All
SD Slot Covers (x3)	783660-01	All
SD Industrial Storage Card, 16 GB	786362-01	All
SD Industrial Storage Card, 32 GB	786363-01	All
4-Position Gold Power Supply Plugs (x5)	783529-01	All
Mini DisplayPort-to-DVI Adapter Cable	157231-0R5	All
Mini DisplayPort-to-VGA Adapter Cable	157230-0R5	All
Mini DisplayPort-to-Full DisplayPort Native Cable	157232-*xx*	All
TRC-8546 Cable	783702-02	cDAQ-9134/9135
TRC-8542 Cable	783699-02	cDAQ-9134/9135
TRC-8543 Cable	783701-02	cDAQ-9134/9135
SMB112 SMB Plug to BNC Male Cable, 50 Ω, 1 m	778827-01	All

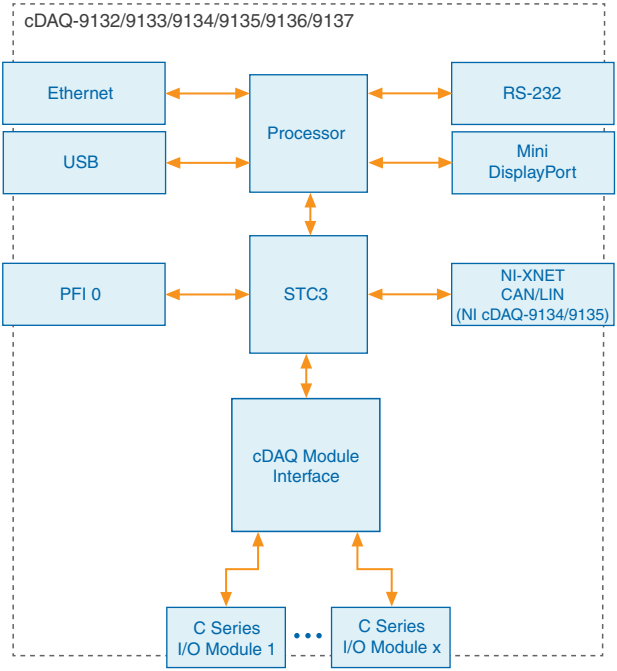
Table 1-10. Cables and Accessories (Continued)

Accessory	Part Number	cDAQ Controller
Keyboard and Mouse	779660-01	All
USB CD/DVD Drive	778492-01	All
NI TSM 1012 Touch Screen Monitor (12 in.)	783635-01	All
NI TSM 1015 Touch Screen Monitor (15 in.)	783636-01	All
NI TSM 1017 Touch Screen Monitor (17 in.)	783637-01	All
* To comply with compliance guidelines in Korea, users should use the PS-16 power supply instead of the PS-15.		

Using the cDAQ Controller

The cDAQ controller consists of four parts—C Series module(s), the cDAQ module interface, the STC3, and the processor—as shown in Figure 1-28. These components digitize signals, perform digital-to-analog conversions to generate analog output signals, measure and control digital I/O signals, and provide signal conditioning.

Figure 1-28. Block Diagram



C Series Module

National Instruments C Series modules provide built-in signal conditioning and screw terminal, spring terminal, BNC, D-SUB, or RJ-50 connectors. A wide variety of I/O types are available, allowing you to customize the cDAQ controller to meet your application needs.

C Series modules are hot-swappable and automatically detected by the cDAQ controller. I/O channels are accessible using the NI-DAQmx driver software.

Because the modules contain built-in signal conditioning for extended voltage ranges or industrial signal types, you can usually make your wiring connections directly from the C Series modules to your sensors/actuators. C Series modules can sometimes provide isolation from channel-to-earth ground and channel-to-channel.

For more information about which C Series modules are compatible with the cDAQ controller, refer to the *C Series Support in NI-DAQmx* document by going to `ni.com/info` and entering the Info Code `rdcdaq`.

Parallel versus Serial DIO Modules

Digital module capabilities are determined by the type of digital signals that the module is capable of measuring or generating.

- Serial digital modules are designed for signals that change slowly and are accessed by either software-timed or hardware-timed reads and writes.
- Parallel digital modules are for signals that change rapidly and are updated by either software-timed or hardware-timed reads and writes.

For more information about digital modules, refer to Chapter 4, *Digital Input/Output and PFI*.

cDAQ Module Interface

The cDAQ module interface manages data transfers between the STC3 and the C Series I/O modules. The interface also handles autodetection, signal routing, and synchronization.

STC3

The STC3 features independent high-speed data streams; flexible AI, AO, and DIO sample timing; triggering; PFI signals for multi-device synchronization; flexible counter/timers with hardware gating; digital waveform acquisition and generation; and static DIO.

- **AI, AO, and DIO Sample Timing**—The STC3 contains advanced AI, AO, and DIO timing engines. A wide range of timing and synchronization signals are available through the PFI lines. Refer to the following sections for more information about the configuration of these signals:
  - The *Analog Input Timing Signals* section of Chapter 2, *Analog Input*
  - The *Analog Output Timing Signals* section of Chapter 3, *Analog Output*
  - The *Digital Input Timing Signals* section of Chapter 4, *Digital Input/Output and PFI*

  - The *Digital Output Timing Signals* section of Chapter 4, *Digital Input/Output and PFI*
- **Triggering Modes**—The cDAQ controller supports different trigger modes, such as start trigger, reference trigger, and pause trigger with analog, digital, or software sources. Refer to the following sections for more information:
  - The *Analog Input Triggering Signals* section of Chapter 2, *Analog Input*
  - The *Analog Output Triggering Signals* section of Chapter 3, *Analog Output*
  - The *Digital Input Triggering Signals* section of Chapter 4, *Digital Input/Output and PFI*
  - The *Digital Output Triggering Signals* section of Chapter 4, *Digital Input/Output and PFI*
- **Independent Data Streams**—The cDAQ controller supports seven independent high-speed data streams, which allow for up to seven simultaneous hardware-timed tasks, such as analog input, analog output, buffered counter/timers, and hardware-timed digital input/output.
- **PFI Signals**—The PFI signals provide access to advanced features such as triggering, synchronization, and counter/timers. You can also enable a programmable debouncing filter on each PFI signal that, when enabled, samples the input on each rising edge of a filter clock. PFI signals are available through parallel digital input and output modules installed in up to two controller slots. Refer to the *PFI* section of Chapter 4, *Digital Input/Output and PFI*, for more information.
- **Flexible Counter/Timers**—The cDAQ controller includes four general-purpose 32-bit counter/timers that can be used to count edges, measure pulse-widths, measure periods and frequencies, and perform position measurements (encoding). In addition, the counter/timers can generate pulses, pulse trains, and square waves with adjustable frequencies. You can access the counter inputs and outputs using parallel digital modules installed in up to two slots. Refer to Chapter 5, *Counters*, for more information.

Processor and Ports

Refer to the specifications document for your cDAQ controller for information about the processor on the cDAQ controller. Refer to the *cDAQ Controller Features* section for information about using the various ports on the cDAQ controller.

2

Analog Input

To perform analog input measurements, insert a supported analog input C Series module into any slot on the cDAQ controller. The measurement specifications, such as number of channels, channel configuration, sample rate, and gain, are determined by the type of C Series module used. For more information and wiring diagrams, refer to the documentation included with your C Series modules.

The cDAQ controller has three AI timing engines, which means that three analog input tasks can be running at a time on a controller. An analog input task can include channels from multiple analog input modules. However, channels from a single module cannot be used in multiple tasks.

Multiple timing engines allow the cDAQ controller to run up to three analog input tasks simultaneously, each using independent timing and triggering configurations. The three AI timing engines are ai, te0, and te1.

Analog Input Triggering Signals

A trigger is a signal that causes an action, such as starting or stopping the acquisition of data. When you configure a trigger, you must decide how you want to produce the trigger and the action you want the trigger to cause. The cDAQ controller supports internal software triggering.

Three triggers are available: Start Trigger, Reference Trigger, and Pause Trigger. An analog or digital trigger can initiate these three trigger actions. Up to two C Series parallel digital input modules can be used in any controller slot to supply a digital trigger. To find your module triggering options, refer to the documentation included with your C Series modules. For more information about using digital modules for triggering, refer to Chapter 4, *Digital Input/Output and PFI*.

Refer to the *AI Start Trigger Signal*, *AI Reference Trigger Signal*, and *AI Pause Trigger Signal* sections for more information about the analog input trigger signals.

Analog Input Timing Signals

The cDAQ controller features the following analog input timing signals:

- *AI Sample Clock Signal**
- *AI Sample Clock Timebase Signal*
- *AI Start Trigger Signal**

- *AI Reference Trigger Signal**
- *AI Pause Trigger Signal**

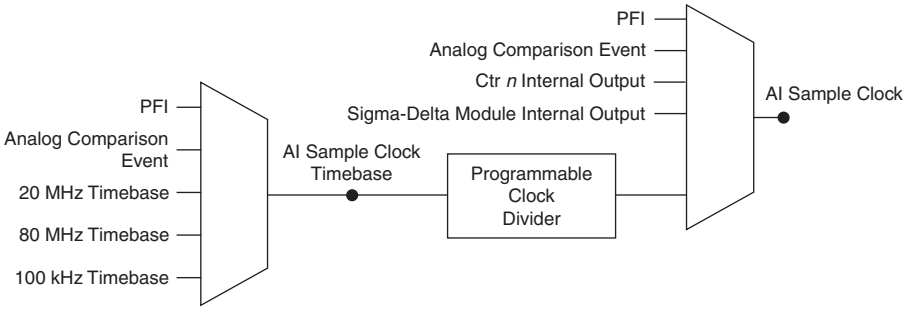
Signals with an * support digital filtering. Refer to the *PFI Filters* section of Chapter 4, *Digital Input/Output and PFI*, for more information.

Refer to the *AI Convert Clock Signal Behavior For Analog Input Modules* section for AI Convert Clock signals and the cDAQ controller.

AI Sample Clock Signal

A sample consists of one reading from each channel in the AI task. Sample Clock signals the start of a sample of all analog input channels in the task. Sample Clock can be generated from external or internal sources as shown in Figure 2-1.

Figure 2-1. AI Sample Clock Timing Options



Routing the Sample Clock to an Output Terminal

You can route Sample Clock to any output PFI terminal. Sample Clock is an active high pulse by default.

AI Sample Clock Timebase Signal

The AI Sample Clock Timebase signal is divided down to provide a source for Sample Clock. AI Sample Clock Timebase can be generated from external or internal sources. AI Sample Clock Timebase is not available as an output from the controller.

AI Convert Clock Signal Behavior For Analog Input Modules

Refer to the *Scanned Modules*, *Simultaneous Sample-and-Hold Modules*, *Sigma-Delta Modules*, and *Slow Sample Rate Modules* sections for information about the AI Convert Clock signal and C Series analog input modules.

Scanned Modules

Scanned C Series analog input modules contain a single ADC and a multiplexer to select between multiple input channels. When the cDAQ Module Interface receives a Sample Clock pulse, it begins generating a Convert Clock for each scanned module in the current task. Each Convert Clock signals the acquisition of a single channel from that module. The Convert Clock rate depends on the module being used, the number of channels used on that module, and the system Sample Clock rate.

The driver chooses the fastest conversion rate possible based on the speed of the ADC for each module and adds 10 µs of padding between each channel to allow for adequate settling time. This scheme enables the channels to approximate simultaneous sampling. If the AI Sample Clock rate is too fast to allow for 10 µs of padding, NI-DAQmx selects a conversion rate that spaces the AI Convert Clock pulses evenly throughout the sample. NI-DAQmx uses the same amount of padding for all the modules in the task. To explicitly specify the conversion rate, use the **ActiveDevs** and **AI Convert Clock Rate** properties using the **DAQmx Timing** property node or functions.

Simultaneous Sample-and-Hold Modules

Simultaneous sample-and-hold (SSH) C Series analog input modules contain multiple ADCs or circuitry that allows all the input channels to be sampled at the same time. These modules sample their inputs on every Sample Clock pulse.

Sigma-Delta Modules

Sigma-delta C Series analog input modules function much like SSH modules, but use ADCs that require a high-frequency oversample clock to produce accurate, synchronized data. Some sigma-delta modules in the cDAQ controller automatically share a single oversample clock to synchronize data from all the modules that support an external oversample clock timebase when they all share the same task. (DSA modules are an example). The cDAQ controller supports a maximum of two synchronization pulse signals configured for your system. This limits the system to two tasks with different oversample clock timebases.

The oversample clock is used as the AI Sample Clock Timebase. While most modules supply a common oversample clock frequency (12.8 MHz), some modules, such as the NI 9234, supply a different frequency. When sigma-delta modules with different oversample clock frequencies are used in an analog input task, the AI Sample Clock Timebase can use any of the available frequencies; by default, the fastest available is used. The sampling rate of all modules in the system is an integer divisor of the frequency of the AI Sample Clock Timebase.

When one or more sigma-delta modules are in an analog input task, the sigma-delta modules also provide the signal used as the AI Sample Clock. This signal is used to cause analog-to-digital conversion for other modules in the system, just as the AI Sample Clock does when a sigma-delta module is not being used.

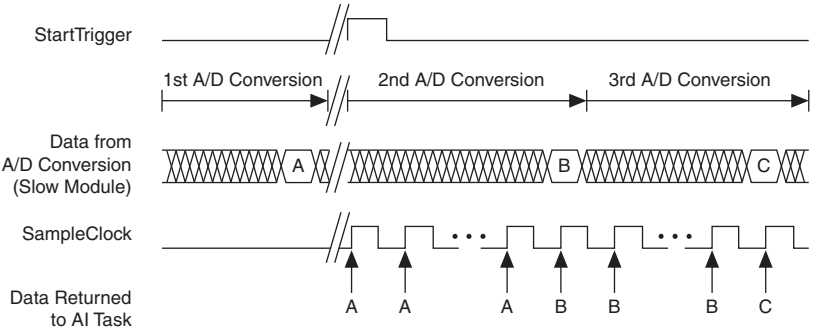
When sigma-delta modules are in an AI task, the controller automatically issues a synchronization pulse to each sigma-delta module that resets their ADCs at the same time.

Because of the filtering used in sigma-delta ADCs, these modules usually exhibit a fixed input delay relative to non-sigma-delta modules in the system. This input delay is specified in the C Series module documentation.

Slow Sample Rate Modules

Some C Series analog input modules are specifically designed for measuring signals that vary slowly, such as temperature. Because of their slow rate, it is not appropriate for these modules to constrain the AI Sample Clock to operate at or slower than their maximum rate. When using such a module in the cDAQ controller, the maximum Sample Clock rate can run faster than the maximum rate for the module. When operating at a rate faster than these slow rate modules can support, the slow rate module returns the same point repeatedly, until a new conversion completes. In a hardware-timed task, the first point is acquired when the task is committed. The second point is acquired after the start trigger as shown in Figure 2-2.

Figure 2-2. Sample Clock Timing Example



For example, if running an AI task at 1 kHz using a module with a maximum rate of 10 Hz, the slow module returns 100 samples of the first point, followed by 100 samples of the second point, etc. Other modules in the task will return 1,000 new data points per second, which is normal. When performing a single-point acquisition, no points are repeated. To avoid this behavior, use multiple AI timing engines, and assign slow sample rate modules to a task with a rate at or slower than their maximum rate.

Refer to the *C Series Support in NI-DAQmx* document by going to `ni.com/info` and entering the Info Code `rdcdaq`.

AI Start Trigger Signal

Use the Start Trigger signal to begin a measurement acquisition. A measurement acquisition consists of one or more samples. If you do not use triggers, begin a measurement with a software command. Once the acquisition begins, configure the acquisition to stop in one of the following ways:

- When a certain number of points has been sampled (in finite mode)
- After a hardware reference trigger (in finite mode)
- With a software command (in continuous mode)

An acquisition that uses a start trigger (but not a reference trigger) is sometimes referred to as a posttriggered acquisition. That is, samples are measured only after the trigger.

When you are using an internal sample clock, you can specify a default delay from the start trigger to the first sample.

Using a Digital Source

To use the Start Trigger signal with a digital source, specify a source and a rising or falling edge. Use the following signals as the source:

- Any PFI terminal
- Counter *n* Internal Output

The source also can be one of several other internal signals on your cDAQ controller. Refer to the *Device Routing in MAX* topic in the *NI-DAQmx Help* or the *LabVIEW Help* for more information.

Using an Analog Source

Some C Series modules can generate a trigger based on an analog signal. In NI-DAQmx, this is called the Analog Comparison Event. When you use an analog trigger source for Start Trigger, the acquisition begins on the first rising edge of the Analog Comparison Event signal.

Note Depending on the C Series module capabilities, you may need two modules to utilize analog triggering.

Routing AI Start Trigger to an Output Terminal

You can route the Start Trigger signal to any output PFI terminal. The output is an active high pulse.

AI Reference Trigger Signal

Use Reference Trigger to stop a measurement acquisition. To use a reference trigger, specify a buffer of finite size and a number of pretrigger samples (samples that occur before the reference trigger). The number of posttrigger samples (samples that occur after the reference trigger) desired is the buffer size minus the number of pretrigger samples.

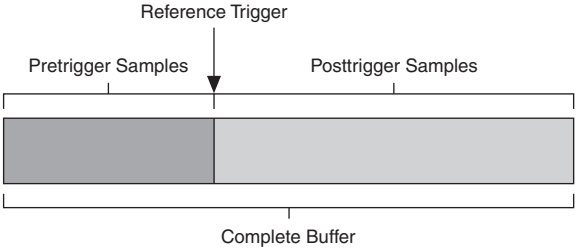
Once the acquisition begins, the cDAQ controller writes samples to the buffer. After the cDAQ controller captures the specified number of pretrigger samples, the cDAQ controller begins to look for the reference trigger condition. If the reference trigger condition occurs before the cDAQ controller captures the specified number of pretrigger samples, the controller ignores the condition.

If the buffer becomes full, the cDAQ controller continuously discards the oldest samples in the buffer to make space for the next sample. This data can be accessed (with some limitations) before the cDAQ controller discards it. Refer to the KnowledgeBase document, *Can a*

Pretriggered Acquisition be Continuous?, for more information. To access this KnowledgeBase, go to `ni.com/info` and enter the Info Code `rdcanq`.

When the reference trigger occurs, the cDAQ controller continues to write samples to the buffer until the buffer contains the number of posttrigger samples desired. Figure 2-3 shows the final buffer.

Figure 2-3. Reference Trigger Final Buffer



Using a Digital Source

To use Reference Trigger with a digital source, specify a source and a rising or falling edge. Either PFI or one of several internal signals on the cDAQ controller can provide the source. Refer to the *Device Routing in MAX* topic in the *NI-DAQmx Help* or the *LabVIEW Help* for more information.

Using an Analog Source

Some C Series modules can generate a trigger based on an analog signal. In NI-DAQmx, this is called the Analog Comparison Event.

When you use an analog trigger source, the acquisition stops on the first rising or falling edge of the Analog Comparison Event signal, depending on the trigger properties.

Note Depending on the C Series module capabilities, you may need two modules to utilize analog triggering.

Routing the Reference Trigger Signal to an Output Terminal

You can route Reference Trigger to any output PFI terminal. Reference Trigger is active high by default.

AI Pause Trigger Signal

You can use the Pause Trigger to pause and resume a measurement acquisition. The internal sample clock pauses while the external trigger signal is active and resumes when the signal is inactive. You can program the active level of the pause trigger to be high or low.

Using a Digital Source

To use the Pause Trigger, specify a source and a polarity. The source can be either from PFI or one of several other internal signals on your cDAQ controller. Refer to the *Device Routing in MAX* topic in the *NI-DAQmx Help* or the *LabVIEW Help* for more information.

Using an Analog Source

Some C Series modules can generate a trigger based on an analog signal. In NI-DAQmx, this is called the Analog Comparison Event.

When you use an analog trigger source, the internal sample clock pauses when the Analog Comparison Event signal is low and resumes when the signal goes high (or vice versa).

Note Depending on the C Series module capabilities, you may need two modules to utilize analog triggering.

Note Pause triggers are only sensitive to the level of the source, not the edge.

Getting Started with AI Applications in Software

You can use the cDAQ controller in the following analog input applications:

- Single-point acquisition
- Finite acquisition
- Continuous acquisition

For more information about programming analog input applications and triggers in software, refer to the *NI-DAQmx Help* or the *LabVIEW Help* for more information.

3

Analog Output

To generate analog output, insert an analog output C Series module in any slot on the cDAQ controller. The generation specifications, such as the number of channels, channel configuration, update rate, and output range, are determined by the type of C Series module used. For more information, refer to the documentation included with your C Series module(s).

On a single analog output C Series module, you can assign any number of channels to either a hardware-timed task or a software-timed (single-point) task. However, you cannot assign some channels to a hardware-timed task and other channels (on the same module) to a software-timed task.

Any hardware-timed task or software-timed task can have channels from multiple modules in the same controller.

Analog Output Data Generation Methods

When performing an analog output operation, you either can perform software-timed or hardware-timed generations. Hardware-timed generations must be buffered.

Software-Timed Generations

With a software-timed generation, software controls the rate at which data is generated. Software sends a separate command to the hardware to initiate each DAC conversion. In NI-DAQmx, software-timed generations are referred to as on-demand timing. Software-timed generations are also referred to as immediate or static operations. They are typically used for writing out a single value, such as a constant DC voltage.

The following considerations apply to software-timed generations:

- If any AO channel on a module is used in a hardware-timed (waveform) task, no channels on that module can be used in a software-timed task.
- You can configure software-timed generations to simultaneously update.
- Only one simultaneous update task can run at a time.
- A hardware-timed AO task and a simultaneous update AO task cannot run at the same time.

Hardware-Timed Generations

With a hardware-timed generation, a digital hardware signal controls the rate of the generation. This signal can be generated internally on the controller or provided externally.

Hardware-timed generations have several advantages over software-timed acquisitions:

- The time between samples can be much shorter.
- The timing between samples is deterministic.
- Hardware-timed acquisitions can use hardware triggering.

Hardware-timed AO operations on the cDAQ controller must be buffered.

Buffered Analog Output

A buffer is a temporary storage in computer memory for generated samples. In a buffered generation, data is moved from a host buffer to the cDAQ controller onboard FIFO before it is written to the C Series modules.

One property of buffered I/O operations is sample mode. The sample mode can be either finite or continuous:

- **Finite**—Finite sample mode generation refers to the generation of a specific, predetermined number of data samples. After the specified number of samples is written out, the generation stops.
- **Continuous**—Continuous generation refers to the generation of an unspecified number of samples. Instead of generating a set number of data samples and stopping, a continuous generation continues until you stop the operation. There are three different continuous generation modes that control how the data is written. These modes are regeneration, onboard regeneration, and non-regeneration:
  - In regeneration mode, you define a buffer in host memory. The data from the buffer is continually downloaded to the FIFO to be written out. New data can be written to the host buffer at any time without disrupting the output. There is no limitation on the number of waveform channels supported by regeneration mode.
  - With onboard regeneration, the entire buffer is downloaded to the FIFO and regenerated from there. After the data is downloaded, new data cannot be written to the FIFO. To use onboard regeneration, the entire buffer must fit within the FIFO size. The advantage of using onboard regeneration is that it does not require communication with the main host memory once the operation is started, which prevents problems that may occur due to excessive bus traffic or operating system latency. There is a limit of 16 waveform channels for onboard regeneration.
  - With non-regeneration, old data is not repeated. New data must continually be written to the buffer. If the program does not write new data to the buffer at a fast enough rate to keep up with the generation, the buffer underflows and causes an error. There is no limitation on the number of waveform channels supported by non-regeneration.

Analog Output Triggering Signals

Analog output supports two different triggering actions: AO Start Trigger and AO Pause Trigger.

An analog or digital trigger can initiate these actions. Up to two C Series parallel digital input modules can be used in any controller slot to supply a digital trigger. An analog trigger can be supplied by some C Series analog modules.

Refer to the *AO Start Trigger Signal* and *AO Pause Trigger Signal* sections for more information about the analog output trigger signals.

Analog Output Timing Signals

The cDAQ controller features the following AO (waveform generation) timing signals:

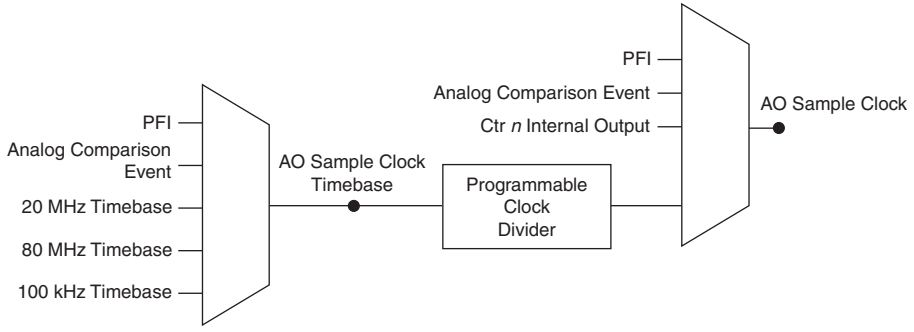
- *AO Sample Clock Signal**
- *AO Sample Clock Timebase Signal*
- *AO Start Trigger Signal**
- *AO Pause Trigger Signal**

Signals with an * support digital filtering. Refer to the *PFI Filters* section of Chapter 4, *Digital Input/Output and PFI*, for more information.

AO Sample Clock Signal

The AO sample clock (ao/SampleClock) signals when all the analog output channels in the task update. AO Sample Clock can be generated from external or internal sources as shown in Figure 3-1.

Figure 3-1. Analog Output Timing Options



Routing AO Sample Clock to an Output Terminal

You can route AO Sample Clock to any output PFI terminal. AO Sample Clock is active high by default.

AO Sample Clock Timebase Signal

The AO Sample Clock Timebase (ao/SampleClockTimebase) signal is divided down to provide a source for AO Sample Clock. AO Sample Clock Timebase can be generated from external or internal sources, and is not available as an output from the controller.

AO Start Trigger Signal

Use the AO Start Trigger (ao/StartTrigger) signal to initiate a waveform generation. If you do not use triggers, you can begin a generation with a software command. If you are using an internal sample clock, you can specify a delay from the start trigger to the first sample. For more information, refer to the *NI-DAQmx Help*.

Using a Digital Source

To use AO Start Trigger, specify a source and a rising or falling edge. The source can be one of the following signals:

- A pulse initiated by host software
- Any PFI terminal
- AI Reference Trigger
- AI Start Trigger

The source also can be one of several internal signals on the cDAQ controller. Refer to the *Device Routing in MAX* topic in the *NI-DAQmx Help* or the *LabVIEW Help* for more information.

You also can specify whether the waveform generation begins on the rising edge or falling edge of AO Start Trigger.

Using an Analog Source

Some C Series modules can generate a trigger based on an analog signal. In NI-DAQmx, this is called the Analog Comparison Event, depending on the trigger properties.

When you use an analog trigger source, the waveform generation begins on the first rising or falling edge of the Analog Comparison Event signal, depending on the trigger properties. The analog trigger circuit must be configured by a simultaneously running analog input task.

Note Depending on the C Series module capabilities, you may need two modules to utilize analog triggering.

Routing AO Start Trigger Signal to an Output Terminal

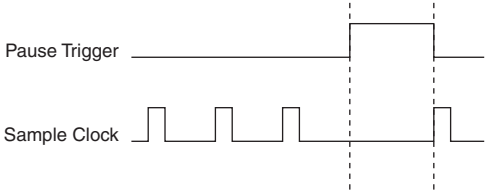
You can route AO Start Trigger to any output PFI terminal. The output is an active high pulse.

AO Pause Trigger Signal

Use the AO Pause Trigger signal (ao/PauseTrigger) to mask off samples in a DAQ sequence. When AO Pause Trigger is active, no samples occur, but AO Pause Trigger does not stop a sample that is in progress. The pause does not take effect until the beginning of the next sample.

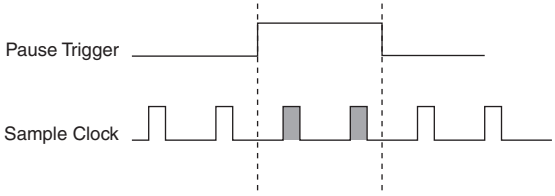
When you generate analog output signals, the generation pauses as soon as the pause trigger is asserted. If the source of the sample clock is the onboard clock, the generation resumes as soon as the pause trigger is deasserted, as shown in Figure 3-2.

Figure 3-2. AO Pause Trigger with the Onboard Clock Source



If you are using any signal other than the onboard clock as the source of the sample clock, the generation resumes as soon as the pause trigger is deasserted and another edge of the sample clock is received, as shown in Figure 3-3.

Figure 3-3. AO Pause Trigger with Other Signal Source



Using a Digital Source

To use AO Pause Trigger, specify a source and a polarity. The source can be a PFI signal or one of several other internal signals on the cDAQ controller.

You also can specify whether the samples are paused when AO Pause Trigger is at a logic high or low level. Refer to the *Device Routing in MAX* topic in the *NI-DAQmx Help* or the *LabVIEW Help* for more information.

Using an Analog Source

Some C Series modules can generate a trigger based on an analog signal. In NI-DAQmx, this is called the Analog Comparison Event, depending on the trigger properties.

When you use an analog trigger source, the samples are paused when the Analog Comparison Event signal is at a high or low level, depending on the trigger properties. The analog trigger circuit must be configured by a simultaneously running analog input task.

Note Depending on the C Series module capabilities, you may need two modules to utilize analog triggering.

Minimizing Glitches on the Output Signal

When you use a DAC to generate a waveform, you may observe glitches on the output signal. These glitches are normal. When a DAC switches from one voltage to another, it produces glitches due to released charges. The largest glitches occur when the most significant bit of the DAC code changes. You can build a lowpass deglitching filter to remove some of these glitches, depending on the frequency and nature of the output signal. Go to `ni.com/support` for more information about minimizing glitches.

Getting Started with AO Applications in Software

You can use the cDAQ controller in the following analog output applications:

- Single-point (on-demand) generation
- Finite generation
- Continuous generation
- Waveform generation

For more information about programming analog output applications and triggers in software, refer the *LabVIEW Help* or to the *NI-DAQmx Help*.

4

Digital Input/Output and PFI

This chapter describes the digital input/output (DIO) and Programmable Function Interface (PFI) functionality available on the cDAQ controller.

Digital Input/Output

To use digital I/O, insert a digital I/O C Series module into any slot on the cDAQ controller. The I/O specifications, such as number of lines, logic levels, update rate, and line direction, are determined by the type of C Series module used. For more information, refer to the documentation included with your C Series module(s).

Serial DIO versus Parallel DIO Modules

Serial digital modules have more than eight lines of digital input/output. They can be used in any controller slot and can perform the following tasks:

- Software-timed and hardware-timed digital input/output tasks

Parallel digital modules can be used in any controller slot and can perform the following tasks:

- Software-timed and hardware-timed digital input/output tasks
- Counter/timer tasks (can be used in up to two slots)
- Accessing PFI signal tasks (can be used in up to two slots)
- Filter digital input signals

Software-timed and hardware-timed digital input/output tasks have the following restrictions:

- You cannot use parallel and serial modules together on the same hardware-timed task.
- You cannot use serial modules for triggering.
- You cannot do both static and timed tasks at the same time on a single serial module.
- You can only do hardware timing in one direction at a time on a serial bidirectional module.

To determine the capability of digital modules supported by the cDAQ controller, refer to the *C Series Support in NI-DAQmx* document by going to `ni.com/info` and entering the Info Code `rdcdaq`.

Static DIO

Each of the DIO lines can be used as a static DI or DO line. You can use static DIO lines to monitor or control digital signals on some C Series modules. Each DIO line can be individually configured as a digital input (DI) or digital output (DO), if the C Series module being used allows such configuration.

All samples of static DI lines and updates of static DO lines are software-timed.

Digital Input

You can acquire digital waveforms using either parallel or serial digital modules.
The DI waveform acquisition FIFO stores the digital samples. The cDAQ controller samples the DIO lines on each rising or falling edge of the DI Sample Clock signal.

Digital Input Triggering Signals

A trigger is a signal that causes an action, such as starting or stopping the acquisition of data. When you configure a trigger, you must decide how you want to produce the trigger and the action you want the trigger to cause. The cDAQ controller supports three types of digital triggering: internal software digital triggering, external digital triggering, and internal digital triggering.

Three triggers are available: Start Trigger, Reference Trigger, and Pause Trigger. An analog or digital trigger can initiate these three trigger actions. Up to two C Series parallel digital input modules can be used in any controller slot to supply a digital trigger. To find your module triggering options, refer to the documentation included with your C Series modules. For more information about using analog modules for triggering, refer to the *Analog Input Triggering Signals* section of Chapter 2, *Analog Input*, and the *Analog Output Triggering Signals* section of Chapter 3, *Analog Output*.

Refer to the *DI Start Trigger Signal*, *DI Reference Trigger Signal*, and *DI Pause Trigger Signal* sections for more information about the digital input trigger signals.

Digital Input Timing Signals

The cDAQ controller features the following digital input timing signals:

- *DI Sample Clock Signal**
- *DI Sample Clock Timebase Signal*
- *DI Start Trigger Signal**
- *DI Reference Trigger Signal**
- *DI Pause Trigger Signal**

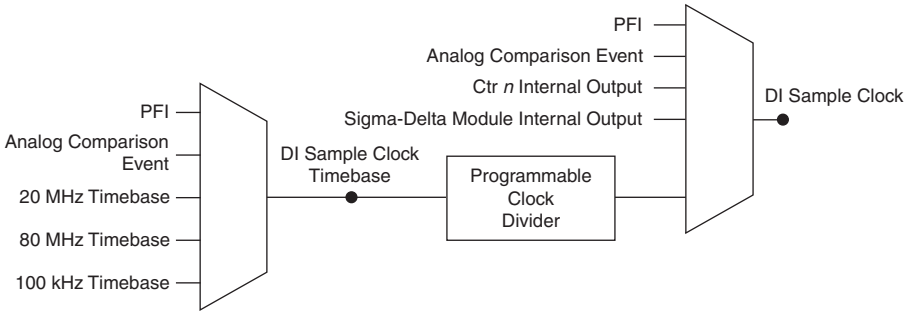
Signals with an * support digital filtering. Refer to the *PFI Filters* section for more information.

DI Sample Clock Signal

Use the DI Sample Clock (di/SampleClock) signal to sample digital I/O on any slot using parallel digital modules, and store the result in the DI waveform acquisition FIFO. If the cDAQ controller receives a DI Sample Clock signal when the FIFO is full, it reports an overflow error to the host software.

A sample consists of one reading from each channel in the DI task. DI Sample Clock signals the start of a sample of all digital input channels in the task. DI Sample Clock can be generated from external or internal sources as shown in Figure 4-1.

Figure 4-1. DI Sample Clock Timing Options



Routing DI Sample Clock to an Output Terminal

You can route DI Sample Clock to any output PFI terminal. Sample Clock is an active high pulse by default.

DI Sample Clock Timebase Signal

The DI Sample Clock Timebase (di/SampleClockTimebase) signal is divided down to provide a source for DI Sample Clock. DI Sample Clock Timebase can be generated from external or internal sources. DI Sample Clock Timebase is not available as an output from the controller.

Using an Internal Source

To use DI Sample Clock with an internal source, specify the signal source and the polarity of the signal. Use the following signals as the source:

- AI Sample Clock
- AO Sample Clock
- Counter *n* Internal Output
- Frequency Output
- DI Change Detection Output

Several other internal signals can be routed to DI Sample Clock. Refer to the *Device Routing in MAX* topic in the *NI-DAQmx Help* or the *LabVIEW Help* for more information.

Using an External Source

You can route the following signals as DI Sample Clock:

- Any PFI terminal
- Analog Comparison Event (an analog trigger)

You can sample data on the rising or falling edge of DI Sample Clock.

DI Start Trigger Signal

Use the DI Start Trigger (di/StartTrigger) signal to begin a measurement acquisition. A measurement acquisition consists of one or more samples. If you do not use triggers, begin a measurement with a software command. Once the acquisition begins, configure the acquisition to stop in one of the following ways:

- When a certain number of points has been sampled (in finite mode)
- After a hardware reference trigger (in finite mode)
- With a software command (in continuous mode)

An acquisition that uses a start trigger (but not a reference trigger) is sometimes referred to as a posttriggered acquisition. That is, samples are measured only after the trigger.

When you are using an internal sample clock, you can specify a delay from the start trigger to the first sample.

Using a Digital Source

To use DI Start Trigger with a digital source, specify a source and a rising or falling edge. Use the following signals as the source:

- Any PFI terminal
- Counter *n* Internal Output

The source also can be one of several other internal signals on the cDAQ controller. Refer to the *Device Routing in MAX* topic in the *NI-DAQmx Help* or the *LabVIEW Help* for more information.

Using an Analog Source

Some C Series modules can generate a trigger based on an analog signal. In NI-DAQmx, this is called the Analog Comparison Event. When you use an analog trigger source for DI Start Trigger, the acquisition begins on the first rising edge of the Analog Comparison Event signal.

Note Depending on the C Series module capabilities, you may need two modules to utilize analog triggering.

Routing DI Start Trigger to an Output Terminal

You can route DI Start Trigger to any output PFI terminal. The output is an active high pulse.

DI Reference Trigger Signal

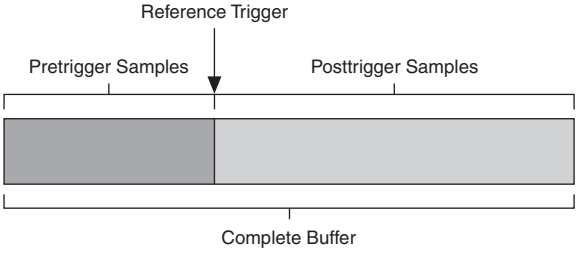
Use a reference trigger (di/ReferenceTrigger) signal to stop a measurement acquisition. To use a reference trigger, specify a buffer of finite size and a number of pretrigger samples (samples that occur before the reference trigger). The number of posttrigger samples (samples that occur after the reference trigger) desired is the buffer size minus the number of pretrigger samples.

Once the acquisition begins, the cDAQ controller writes samples to the buffer. After the cDAQ controller captures the specified number of pretrigger samples, the controller begins to look for the reference trigger condition. If the reference trigger condition occurs before the cDAQ controller captures the specified number of pretrigger samples, the controller ignores the condition.

If the buffer becomes full, the cDAQ controller continuously discards the oldest samples in the buffer to make space for the next sample. This data can be accessed (with some limitations) before the cDAQ controller discards it. Refer to the KnowledgeBase document, *Can a Pretriggered Acquisition be Continuous?*, for more information. To access this KnowledgeBase, go to `ni.com/info` and enter the Info Code `rdcanq`.

When the reference trigger occurs, the cDAQ controller continues to write samples to the buffer until the buffer contains the number of posttrigger samples desired. Figure 4-2 shows the final buffer.

Figure 4-2. Reference Trigger Final Buffer



Using a Digital Source

To use DI Reference Trigger with a digital source, specify a source and a rising or falling edge. Either PFI or one of several internal signals on the cDAQ controller can provide the source. Refer to the *Device Routing in MAX* topic in the *NI-DAQmx Help* or the *LabVIEW Help* for more information.

Using an Analog Source

Some C Series modules can generate a trigger based on an analog signal. In NI-DAQmx, this is called the Analog Comparison Event.

When you use an analog trigger source, the acquisition stops on the first rising or falling edge of the Analog Comparison Event signal, depending on the trigger properties.

Note Depending on the C Series module capabilities, you may need two modules to utilize analog triggering.

Routing DI Reference Trigger Signal to an Output Terminal

You can route DI Reference Trigger to any output PFI terminal. Reference Trigger is active high by default.

DI Pause Trigger Signal

You can use the DI Pause Trigger (di/PauseTrigger) signal to pause and resume a measurement acquisition. The internal sample clock pauses while the external trigger signal is active and resumes when the signal is inactive. You can program the active level of the pause trigger to be high or low.

Using a Digital Source

To use DI Pause Trigger, specify a source and a polarity. The source can be either from PFI or one of several other internal signals on your cDAQ controller. Refer to the *Device Routing in MAX* topic in the *NI-DAQmx Help* or the *LabVIEW Help* for more information.

Using an Analog Source

Some C Series modules can generate a trigger based on an analog signal. In NI-DAQmx, this is called the Analog Comparison Event.

When you use an analog trigger source, the internal sample clock pauses when the Analog Comparison Event signal is low and resumes when the signal goes high (or vice versa).

Note Depending on the C Series module capabilities, you may need two modules to utilize analog triggering.

Note Pause triggers are only sensitive to the level of the source, not the edge.

Digital Input Filters

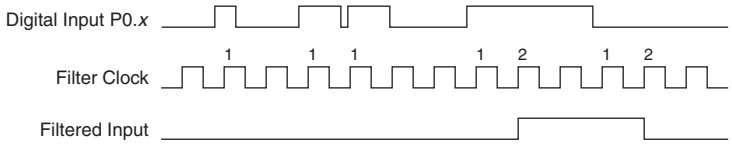
When performing a hardware-timed task, you can enable a programmable debouncing filter on the digital input lines of a parallel DIO module. All lines on a module must share the same filter configuration. When the filter is enabled, the controller samples the inputs with a user-configured Filter Clock derived from the controller timebase. This is used to determine whether a pulse is propagated to the rest of the system. However, the filter also introduces jitter onto the input signal.

In NI-DAQmx, the filter is programmed by setting the minimum pulse width, *Tp*[1], that will pass the filter, and is selectable in 25 ns increments. The appropriate Filter Clock is selected by the driver. Pulses of length less than 1/2 *Tp* will be rejected, and the filtering behavior of lengths between 1/2 *Tp* and 1 *Tp* are not defined because they depend on the phase of the Filter Clock relative to the input signal.

Figure 4-3 shows an example of low-to-high transitions of the input signal. High-to-low transitions work similarly.

Assume that an input terminal has been low for a long time. The input terminal then changes from low to high, but glitches several times. When the filter clock has sampled the signal high on consecutive rising edges, the low-to-high transition is propagated to the rest of the circuit.

Figure 4-3. Filter Example



Getting Started with DI Applications in Software

You can use the cDAQ controller in the following digital input applications:

- Single-point acquisition
- Finite acquisition
- Continuous acquisition

Change Detection Event

The Change Detection Event is the signal generated when a change on the rising or falling edge lines is detected by the change detection task.

Routing Change Detection Event to an Output Terminal

You can route ChangeDetectionEvent to any output PFI terminal.

Change Detection Acquisition

You can configure lines on parallel digital modules to detect rising or falling edges. When one or more of these lines sees the edge specified for that line, the cDAQ controller samples all the lines in the task. The rising and falling edge lines do not necessarily have to be in the task.

Change detection acquisitions can be buffered or nonbuffered:

- **Nonbuffered Change Detection Acquisition**—In a nonbuffered acquisition, data is transferred from the cDAQ controller directly to a PC buffer.

[1] *Tp* is a nominal value; the accuracy of the controller timebase and I/O distortion will affect this value.

- **Buffered Change Detection Acquisition**—A buffer is a temporary storage in computer memory for acquired samples. In a buffered acquisition, data is stored in the cDAQ controller onboard FIFO then transferred to a PC buffer. Buffered acquisitions typically allow for much faster transfer rates than nonbuffered acquisitions because data accumulates and is transferred in blocks, rather than one sample at a time.

Digital Output

To generate digital output, insert a digital output C Series module in any slot on the cDAQ controller. The generation specifications, such as the number of channels, channel configuration, update rate, and output range, are determined by the type of C Series module used. For more information, refer to the documentation included with your C Series module(s).

With parallel digital output modules (formerly known as hardware-timed modules), you can do multiple software-timed tasks on a single module, as well as mix hardware-timed and software-timed digital output tasks on a single module. On serial digital output modules, (formerly known as static digital output modules), you cannot mix hardware-timed and software-timed tasks, but you can run multiple software-timed tasks.

You may have a hardware-timed task or a software-timed task include channels from multiple modules, but a hardware-timed task may not include a mix of channels from both parallel and serial modules.

Digital Output Data Generation Methods

When performing a digital output operation, you either can perform software-timed or hardware-timed generations. Hardware-timed generations must be buffered.

Software-Timed Generations

With a software-timed generation, software controls the rate at which data is generated. Software sends a separate command to the hardware to initiate each digital generation. In NI-DAQmx, software-timed generations are referred to as on-demand timing. Software-timed generations are also referred to as immediate or static operations. They are typically used for writing out a single value.

For software-timed generations, if any DO channel on a serial digital module is used in a hardware-timed task, no channels on that module can be used in a software-timed task.

Hardware-Timed Generations

With a hardware-timed generation, a digital hardware signal controls the rate of the generation. This signal can be generated internally on the controller or provided externally.

Hardware-timed generations have several advantages over software-timed acquisitions:

- The time between samples can be much shorter.
- The timing between samples is deterministic.
- Hardware-timed acquisitions can use hardware triggering.

Hardware-timed DO operations on the cDAQ controller must be buffered.

Buffered Digital Output

A buffer is a temporary storage in computer memory for generated samples. In a buffered generation, data is moved from a host buffer to the cDAQ controller onboard FIFO before it is written to the C Series module(s).

One property of buffered I/O operations is sample mode. The sample mode can be either finite or continuous:

- **Finite**—Finite sample mode generation refers to the generation of a specific, predetermined number of data samples. After the specified number of samples is written out, the generation stops.
- **Continuous**—Continuous generation refers to the generation of an unspecified number of samples. Instead of generating a set number of data samples and stopping, a continuous generation continues until you stop the operation. There are three different continuous generation modes that control how the data is written. These modes are regeneration, onboard regeneration, and non-regeneration:
  - In regeneration mode, you define a buffer in host memory. The data from the buffer is continually downloaded to the FIFO to be written out. New data can be written to the host buffer at any time without disrupting the output.
  - With onboard regeneration, the entire buffer is downloaded to the FIFO and regenerated from there. After the data is downloaded, new data cannot be written to the FIFO. To use onboard regeneration, the entire buffer must fit within the FIFO size. The advantage of using onboard regeneration is that it does not require communication with the main host memory once the operation is started, which prevents problems that may occur due to excessive bus traffic or operating system latency.
  - With non-regeneration, old data is not repeated. New data must continually be written to the buffer. If the program does not write new data to the buffer at a fast enough rate to keep up with the generation, the buffer underflows and causes an error.

Digital Output Triggering Signals

Digital output supports two different triggering actions: DO Start Trigger and DO Pause Trigger.

A digital or analog trigger can initiate these actions. Any PFI terminal can supply a digital trigger, and some C Series analog modules can supply an analog trigger. For more information, refer to the documentation included with your C Series module(s).

Refer to the *DO Start Trigger Signal* and *DO Pause Trigger Signal* sections for more information about the digital output trigger signals.

Digital Output Timing Signals

The cDAQ controller features the following DO timing signals:

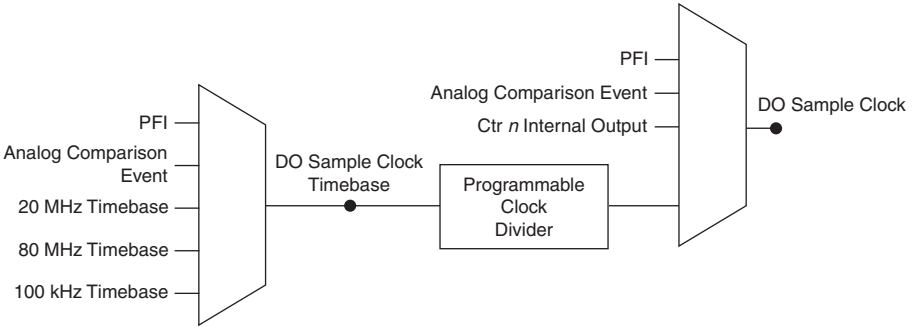
- *DO Sample Clock Signal**
- *DO Sample Clock Timebase Signal*
- *DO Start Trigger Signal**
- *DO Pause Trigger Signal**

Signals with an * support digital filtering. Refer to the *PFI Filters* section for more information.

DO Sample Clock Signal

The DO Sample Clock (do/SampleClock) signals when all the digital output channels in the task update. DO Sample Clock can be generated from external or internal sources as shown in Figure 4-4.

Figure 4-4. Digital Output Timing Options



Routing DO Sample Clock to an Output Terminal

You can route DO Sample Clock to any output PFI terminal. DO Sample Clock is active high by default.

DO Sample Clock Timebase Signal

The DO Sample Clock Timebase (do/SampleClockTimebase) signal is divided down to provide a source for DO Sample Clock. DO Sample Clock Timebase can be generated from external or internal sources, and is not available as an output from the controller.

DO Start Trigger Signal

Use the DO Start Trigger (do/StartTrigger) signal to initiate a waveform generation. If you do not use triggers, you can begin a generation with a software command. If you are using an internal sample clock, you can specify a delay from the start trigger to the first sample. For more information, refer to the *NI-DAQmx Help*.

Using a Digital Source

To use DO Start Trigger, specify a source and a rising or falling edge. The source can be one of the following signals:

- A pulse initiated by host software
- Any PFI terminal
- AI Reference Trigger
- AI Start Trigger

The source also can be one of several internal signals on the cDAQ controller. Refer to the *Device Routing in MAX* topic in the *NI-DAQmx Help* or the *LabVIEW Help* for more information.

You also can specify whether the waveform generation begins on the rising edge or falling edge of DO Start Trigger.

Using an Analog Source

Some C Series modules can generate a trigger based on an analog signal. In NI-DAQmx, this is called the Analog Comparison Event, depending on the trigger properties.

When you use an analog trigger source, the waveform generation begins on the first rising or falling edge of the Analog Comparison Event signal, depending on the trigger properties. The analog trigger circuit must be configured by a simultaneously running analog input task.

Note Depending on the C Series module capabilities, you may need two modules to utilize analog triggering.

Routing DO Start Trigger Signal to an Output Terminal

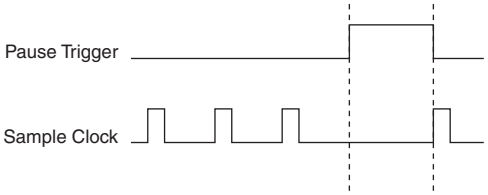
You can route DO Start Trigger to any output PFI terminal. The output is an active high pulse.

DO Pause Trigger Signal

Use the DO Pause Trigger signal (do/PauseTrigger) to mask off samples in a DAQ sequence. When DO Pause Trigger is active, no samples occur, but DO Pause Trigger does not stop a sample that is in progress. The pause does not take effect until the beginning of the next sample.

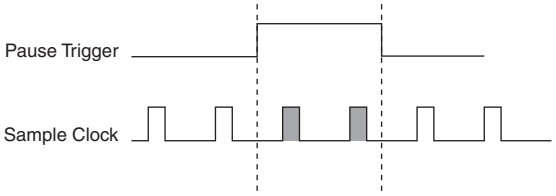
When you generate digital output signals, the generation pauses as soon as the pause trigger is asserted. If the source of the sample clock is the onboard clock, the generation resumes as soon as the pause trigger is deasserted, as shown in Figure 4-5.

Figure 4-5. DO Pause Trigger with the Onboard Clock Source



If you are using any signal other than the onboard clock as the source of the sample clock, the generation resumes as soon as the pause trigger is deasserted and another edge of the sample clock is received, as shown in Figure 4-6.

Figure 4-6. DO Pause Trigger with Other Signal Source



Using a Digital Source

To use DO Pause Trigger, specify a source and a polarity. The source can be a PFI signal or one of several other internal signals on the cDAQ controller.

You also can specify whether the samples are paused when DO Pause Trigger is at a logic high or low level. Refer to the *Device Routing in MAX* topic in the *NI-DAQmx Help* or the *LabVIEW Help* for more information.

Using an Analog Source

Some C Series modules can generate a trigger based on an analog signal. In NI-DAQmx, this is called the Analog Comparison Event, depending on the trigger properties.

When you use an analog trigger source, the samples are paused when the Analog Comparison Event signal is at a high or low level, depending on the trigger properties. The analog trigger circuit must be configured by a simultaneously running analog input task.

Note Depending on the C Series module capabilities, you may need two modules to utilize analog triggering.

Getting Started with DO Applications in Software

You can use the cDAQ controller in the following digital output applications:

- Single-point (on-demand) generation
- Finite generation
- Continuous generation

For more information about programming digital output applications and triggers in software, refer the *LabVIEW Help* or to the *NI-DAQmx Help*.

Digital Input/Output Configuration for NI 9401

When you change the configuration of lines on a NI 9401 digital module between input and output, NI-DAQmx temporarily reserves all of the lines on the module for communication to send the module a line configuration command. For this reason, you must reserve the task in advance through the DAQmx Control Task before any task has started. If another task or route is actively using the module, to avoid interfering with the other task, NI-DAQmx generates an error instead of sending the line configuration command. During the line configuration command, the output lines are maintained without glitching.

PFI

You can configure channels of a parallel digital module as Programmable Function Interface (PFI) terminals. The cDAQ controller also provides one terminal for PFI on its front panel. One or two digital modules can be used to access the cDAQ controller PFI terminal. You can use the the cDAQ controller PFI terminal to wake the controller from a low power state. You can drive a wake event with high logic-level signals. Refer to the specifications document for your cDAQ controller for DC input characteristics.

You can configure each PFI individually as the following:

- Timing input signal for AI, AO, DI, DO, or counter/timer functions
- Timing output signal from AI, AO, DI, DO, or counter/timer functions

PFI Filters

You can enable a programmable debouncing filter on each PFI signal. When the filter is enabled, the controller samples the inputs with a user-configured Filter Clock derived from the controller timebase. This is used to determine whether a pulse is propagated to the rest of the circuit. However, the filter also introduces jitter onto the PFI signal.

The following is an example of low-to-high transitions of the input signal. High-to-low transitions work similarly.

Assume that an input terminal has been low for a long time. The input terminal then changes from low to high, but glitches several times. When the Filter Clock has sampled the signal high on *N* consecutive edges, the low-to-high transition is propagated to the rest of the circuit. The value of *N* depends on the filter setting, as shown in Table 4-1.

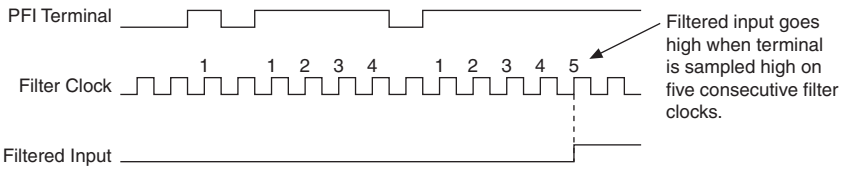
Table 4-1. Selectable PFI Filter Settings

Filter Setting	Filter Clock	Jitter	Min Pulse Width* to Pass	Max Pulse Width* to Not Pass
112.5 ns (short)	80 MHz	12.5 ns	112.5 ns	100 ns
6.4 µs (medium)	80 MHz	12.5 ns	6.4 µs	6.3875 µs
2.56 ms (high)	100 kHz	10 µs	2.56 ms	2.55 ms
Custom	User-configurable	1 Filter Clock period	T_{user}	T_{user} - (1 Filter Clock period)

* Pulse widths are nominal values; the accuracy of the controller timebase and I/O distortion will affect these values.

On power up, the filters are disabled. Figure 4-7 shows an example of a low-to-high transition on an input that has a custom filter set to $N = 5$.

Figure 4-7. PFI Filter Example

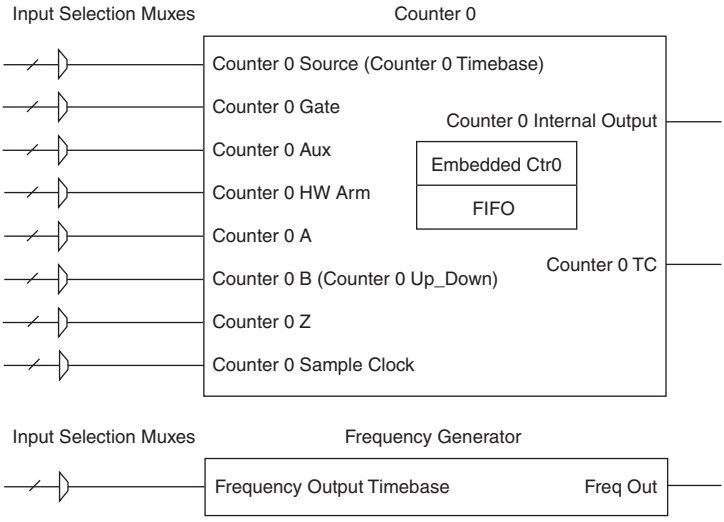


5

Counters

The cDAQ controller has four general-purpose 32-bit counter/timers and one frequency generator. The general-purpose counter/timers can be used for many measurement and pulse generation applications. Figure 5-1 shows the cDAQ controller Counter 0 and the frequency generator. All four counters on the cDAQ controller are identical.

Figure 5-1. cDAQ Controller Counter 0 and Frequency Generator



Counters have eight input signals, although in most applications only a few inputs are used.

For information about connecting counter signals, refer to the *Default Counter/Timer Routing* section.

Each counter has a FIFO that can be used for buffered acquisition and generation. Each counter also contains an embedded counter (Embedded Ctr*n*) for use in what are traditionally two-counter measurements and generations. The embedded counters cannot be programmed independent of the main counter; signals from the embedded counters are not routable.

Counter Timing Engine

Unlike analog input, analog output, digital input, and digital output, the cDAQ controller counters do not have the ability to divide down a timebase to produce an internal counter sample clock. For sample clocked operations, an external signal must be provided to supply a clock source. The source can be any of the following signals:

- AI Sample Clock
- AI Start Trigger
- AI Reference Trigger
- AO Sample Clock
- DI Sample Clock
- DI Start Trigger
- DO Sample Clock
- CTR *n* Internal Output
- Freq Out
- PFI
- Change Detection Event
- Analog Comparison Event

Not all timed counter operations require a sample clock. For example, a simple buffered pulse width measurement latches in data on each edge of a pulse. For this measurement, the measured signal determines when data is latched in. These operations are referred to as implicit timed operations. However, many of the same measurements can be clocked at an interval with a sample clock. These are referred to as sample clocked operations. Table 5-1 shows the different options for the different measurements.

Table 5-1. Counter Timing Measurements

Measurement	Implicit Timing Support	Sample Clocked Timing Support
Buffered Edge Count	No	Yes
Buffered Pulse Width	Yes	Yes
Buffered Pulse	Yes	Yes
Buffered Semi-Period	Yes	No
Buffered Frequency	Yes	Yes
Buffered Period	Yes	Yes

Table 5-1. Counter Timing Measurements (Continued)

Measurement	Implicit Timing Support	Sample Clocked Timing Support
Buffered Position	No	Yes
Buffered Two-Signal Edge Separation	Yes	Yes

Counter Input Applications

The following sections list the various counter input applications available on the cDAQ controller:

- *Counting Edges*
- *Pulse-Width Measurement*
- *Pulse Measurement*
- *Semi-Period Measurement*
- *Frequency Measurement*
- *Period Measurement*
- *Position Measurement*

Counting Edges

In edge counting applications, the counter counts edges on its Source after the counter is armed. You can configure the counter to count rising or falling edges on its Source input. You also can control the direction of counting (up or down), as described in the *Controlling the Direction of Counting* section. The counter values can be read on demand or with a sample clock.

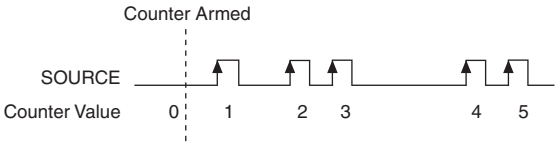
Refer to the following sections for more information about edge counting options:

- *Single Point (On-Demand) Edge Counting*
- *Buffered (Sample Clock) Edge Counting*

Single Point (On-Demand) Edge Counting

With single point (on-demand) edge counting, the counter counts the number of edges on the Source input after the counter is armed. On-demand refers to the fact that software can read the counter contents at any time without disturbing the counting process. Figure 5-2 shows an example of single point edge counting.

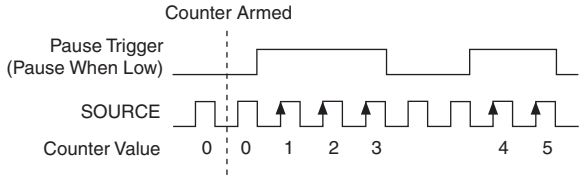
Figure 5-2. Single Point (On-Demand) Edge Counting



You also can use a pause trigger to pause (or gate) the counter. When the pause trigger is active, the counter ignores edges on its Source input. When the pause trigger is inactive, the counter counts edges normally.

You can route the pause trigger to the Gate input of the counter. You can configure the counter to pause counting when the pause trigger is high or when it is low. Figure 5-3 shows an example of on-demand edge counting with a pause trigger.

Figure 5-3. Single Point (On-Demand) Edge Counting with Pause Trigger



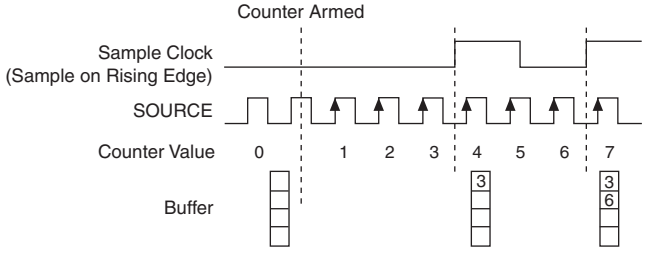
Buffered (Sample Clock) Edge Counting

With buffered edge counting (edge counting using a sample clock), the counter counts the number of edges on the Source input after the counter is armed. The value of the counter is sampled on each active edge of a sample clock and stored in the FIFO. The STC3 transfers the sampled values to host memory using a high-speed data stream.

The count values returned are the cumulative counts since the counter armed event. That is, the sample clock does not reset the counter. You can configure the counter to sample on the rising or falling edge of the sample clock.

Figure 5-4 shows an example of buffered edge counting. Notice that counting begins when the counter is armed, which occurs before the first active edge on Sample Clock.

Figure 5-4. Buffered (Sample Clock) Edge Counting



Controlling the Direction of Counting

In edge counting applications, the counter can count up or down. You can configure the counter to do the following:

- Always count up
- Always count down
- Count up when the Counter 0 B input is high; count down when it is low

For information about connecting counter signals, refer to the *Default Counter/Timer Routing* section.

Pulse-Width Measurement

In pulse-width measurements, the counter measures the width of a pulse on its Gate input signal. You can configure the counter to measure the width of high pulses or low pulses on the Gate signal.

You can route an internal or external periodic clock signal (with a known period) to the Source input of the counter. The counter counts the number of rising (or falling) edges on the Source signal while the pulse on the Gate signal is active.

You can calculate the pulse width by multiplying the period of the Source signal by the number of edges returned by the counter.

A pulse-width measurement will be accurate even if the counter is armed while a pulse train is in progress. If a counter is armed while the pulse is in the active state, it will wait for the next transition to the active state to begin the measurement.

Refer to the following sections for more information about cDAQ controller pulse-width measurement options:

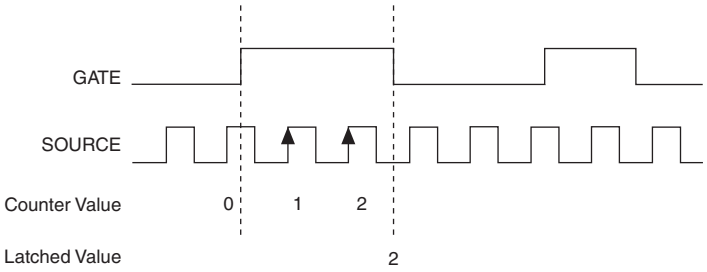
- *Single Pulse-Width Measurement*
- *Implicit Buffered Pulse-Width Measurement*
- *Sample Clocked Buffered Pulse-Width Measurement*

Single Pulse-Width Measurement

With single pulse-width measurement, the counter counts the number of edges on the Source input while the Gate input remains active. When the Gate input goes inactive, the counter stores the count in the FIFO and ignores other edges on the Gate and Source inputs. Software then reads the stored count.

Figure 5-5 shows an example of a single pulse-width measurement.

Figure 5-5. Single Pulse-Width Measurement



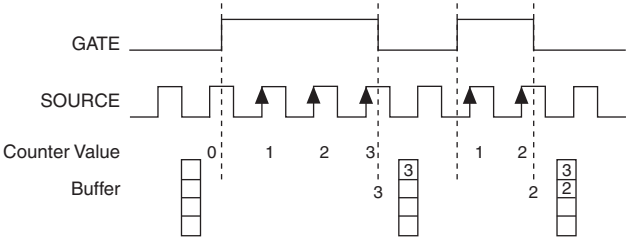
Implicit Buffered Pulse-Width Measurement

An implicit buffered pulse-width measurement is similar to single pulse-width measurement, but buffered pulse-width measurement takes measurements over multiple pulses.

The counter counts the number of edges on the Source input while the Gate input remains active. On each trailing edge of the Gate signal, the counter stores the count in the counter FIFO. The STC3 transfers the sampled values to host memory using a high-speed data stream.

Figure 5-6 shows an example of an implicit buffered pulse-width measurement.

Figure 5-6. Implicit Buffered Pulse-Width Measurement



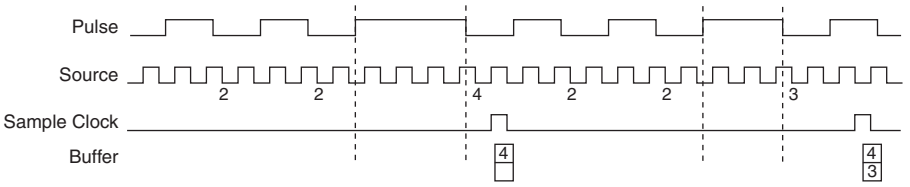
Sample Clocked Buffered Pulse-Width Measurement

A sample clocked buffered pulse-width measurement is similar to single pulse-width measurement, but buffered pulse-width measurement takes measurements over multiple pulses correlated to a sample clock.

The counter counts the number of edges on the Source input while the Gate input remains active. On each sample clock edge, the counter stores the count in the FIFO of the last pulse width to complete. The STC3 transfers the sampled values to host memory using a high-speed data stream.

Figure 5-7 shows an example of a sample clocked buffered pulse-width measurement.

Figure 5-7. Sample Clocked Buffered Pulse-Width Measurement



Note If a pulse does not occur between sample clocks, an overrun error occurs.

For information about connecting counter signals, refer to the *Default Counter/Timer Routing* section.

Pulse Measurement

In pulse measurements, the counter measures the high and low time of a pulse on its Gate input signal after the counter is armed. A pulse is defined in terms of its high and low time, high and low ticks or frequency and duty cycle. This is similar to the pulse-width measurement, except that the inactive pulse is measured as well.

You can route an internal or external periodic clock signal (with a known period) to the Source input of the counter. The counter counts the number of rising (or falling) edges occurring on the Source input between two edges of the Gate signal.

You can calculate the high and low time of the Gate input by multiplying the period of the Source signal by the number of edges returned by the counter.

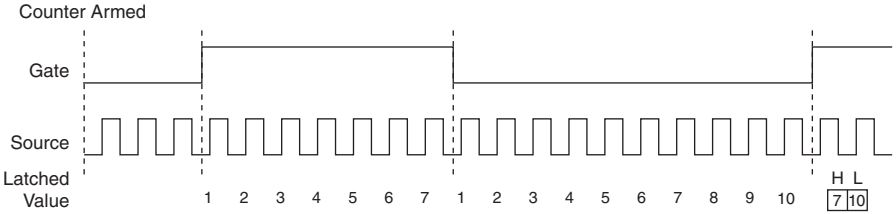
Refer to the following sections for more information about cDAQ controller pulse measurement options:

- *Single Pulse Measurement*
- *Implicit Buffered Pulse Measurement*
- *Sample Clocked Buffered Pulse Measurement*

Single Pulse Measurement

Single (on-demand) pulse measurement is equivalent to two single pulse-width measurements on the high (H) and low (L) ticks of a pulse, as shown in Figure 5-8.

Figure 5-8. Single (On-Demand) Pulse Measurement



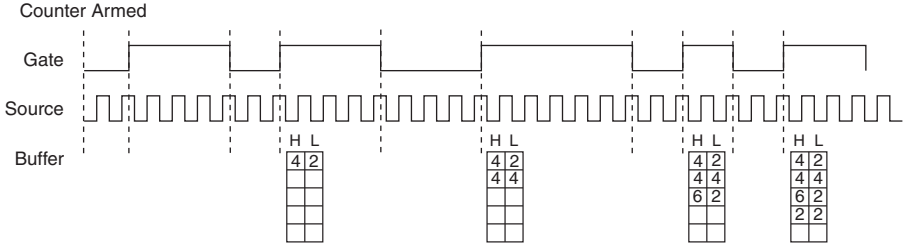
Implicit Buffered Pulse Measurement

In an implicit buffered pulse measurement, on each edge of the Gate signal, the counter stores the count in the FIFO. The STC3 transfers the sampled values to host memory using a high-speed data stream.

The counter begins counting when it is armed. The arm usually occurs between edges on the Gate input but the counting does not start until the desired edge. You can select whether to read the high pulse or low pulse first using the **StartingEdge** property in NI-DAQmx.

Figure 5-9 shows an example of an implicit buffered pulse measurement.

Figure 5-9. Implicit Buffered Pulse Measurement



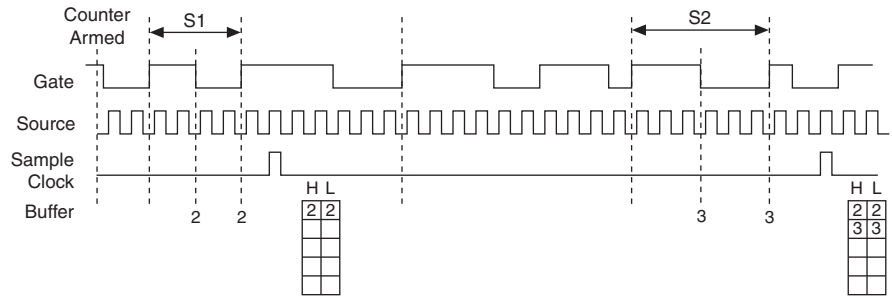
Sample Clocked Buffered Pulse Measurement

A sample clocked buffered pulse measurement is similar to single pulse measurement, but a buffered pulse measurement takes measurements over multiple pulses correlated to a sample clock.

The counter performs a pulse measurement on the Gate. On each sample clock edge, the counter stores the high and low ticks in the FIFO of the last pulse to complete. The STC3 transfers the sampled values to host memory using a high-speed data stream.

Figure 5-10 shows an example of a sample clocked buffered pulse measurement.

Figure 5-10. Sample Clocked Buffered Pulse Measurement



Note If a pulse does not occur between sample clocks, an overrun error occurs.

For information about connecting counter signals, refer to the *Default Counter/Timer Routing* section.

Semi-Period Measurement

In semi-period measurements, the counter measures a semi-period on its Gate input signal after the counter is armed. A semi-period is the time between any two consecutive edges on the Gate input.

You can route an internal or external periodic clock signal (with a known period) to the Source input of the counter. The counter counts the number of rising (or falling) edges occurring on the Source input between two edges of the Gate signal.

You can calculate the semi-period of the Gate input by multiplying the period of the Source signal by the number of edges returned by the counter.

Refer to the following sections for more information about semi-period measurement options:

- *Single Semi-Period Measurement*
- *Implicit Buffered Semi-Period Measurement*

Refer to the *Pulse versus Semi-Period Measurements* section for information about the differences between semi-period measurement and pulse measurement.

Single Semi-Period Measurement

Single semi-period measurement is equivalent to single pulse-width measurement.

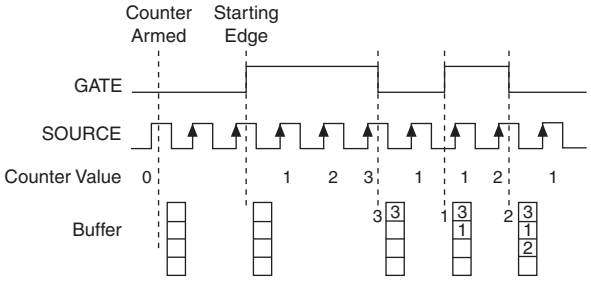
Implicit Buffered Semi-Period Measurement

In implicit buffered semi-period measurements, on each edge of the Gate signal, the counter stores the count in the FIFO. The STC3 transfers the sampled values to host memory using a high-speed data stream.

The counter begins counting when it is armed. The arm usually occurs between edges on the Gate input. You can select whether to read the first active low or active high semi period using the **CI.SemiPeriod.StartingEdge** property in NI-DAQmx.

Figure 5-11 shows an example of an implicit buffered semi-period measurement.

Figure 5-11. Implicit Buffered Semi-Period Measurement



For information about connecting counter signals, refer to the *Default Counter/Timer Routing* section.

Pulse versus Semi-Period Measurements

In hardware, pulse measurement and semi-period are the same measurement. Both measure the high and low times of a pulse. The functional difference between the two measurements is how the data is returned. In a semi-period measurement, each high or low time is considered one point of data and returned in units of seconds or ticks. In a pulse measurement, each pair of high and low times is considered one point of data and returned as a paired sample in units of frequency and duty cycle, high and low time or high and low ticks. When reading data, 10 points in a semi-period measurement will get an array of five high times and five low times. When you read 10 points in a pulse measurement, you get an array of 10 pairs of high and low times.

Also, pulse measurements support sample clock timing while semi-period measurements do not.

Frequency Measurement

You can use the counters to measure frequency in several different ways. Refer to the following sections for information about cDAQ controller frequency measurement options:

- *Low Frequency with One Counter*
- *High Frequency with Two Counters*
- *Large Range of Frequencies with Two Counters*
- *Sample Clocked Buffered Frequency Measurement*

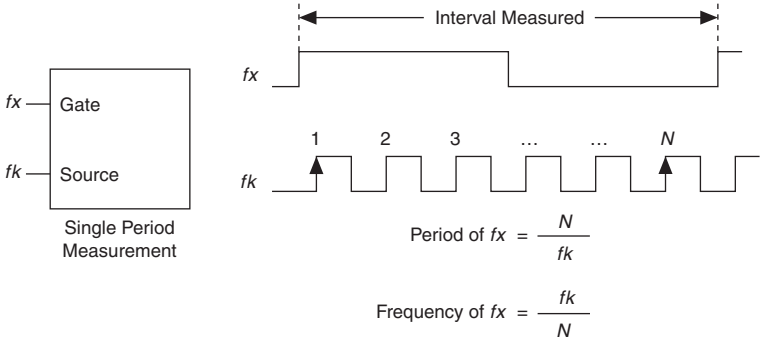
Low Frequency with One Counter

For low frequency measurements with one counter, you measure one period of your signal using a known timebase.

You can route the signal to measure (*fx*) to the Gate of a counter. You can route a known timebase (*fk*) to the Source of the counter. The known timebase can be an onboard timebase, such as 80 MHz Timebase, 20 MHz Timebase, or 100 kHz Timebase, or any other signal with a known rate.

You can configure the counter to measure one period of the gate signal. The frequency of *fx* is the inverse of the period. Figure 5-12 illustrates this method.

Figure 5-12. Low Frequency with One Counter



High Frequency with Two Counters

For high frequency measurements with two counters, you measure one pulse of a known width using your signal and derive the frequency of your signal from the result.

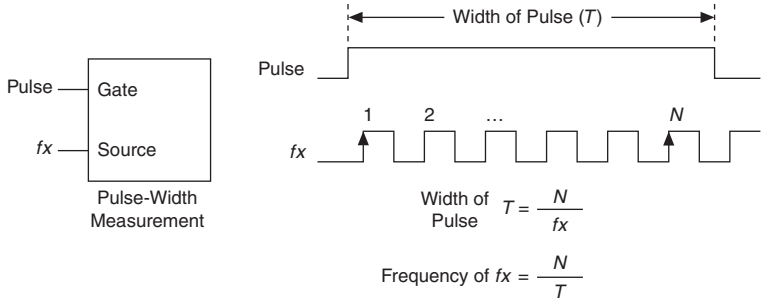
Note Counter 0 is always paired with Counter 1. Counter 2 is always paired with Counter 3.

In this method, you route a pulse of known duration (*T*) to the Gate of a counter. You can generate the pulse using a second counter. You also can generate the pulse externally and connect it to a PFI terminal. You only need to use one counter if you generate the pulse externally.

Route the signal to measure (*fx*) to the Source of the counter. Configure the counter for a single pulse-width measurement. If you measure the width of pulse *T* to be *N* periods of *fx*, the frequency of *fx* is *N*/*T*.

Figure 5-13 illustrates this method. Another option is to measure the width of a known period instead of a known pulse.

Figure 5-13. High Frequency with Two Counters



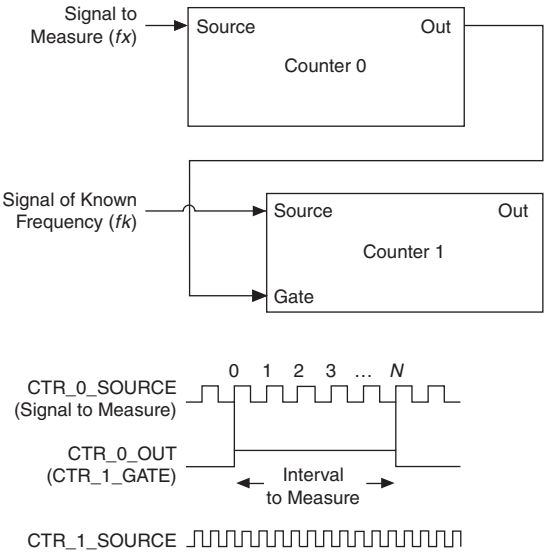
Large Range of Frequencies with Two Counters

By using two counters, you can accurately measure a signal that might be high or low frequency. This technique is called reciprocal frequency measurement. When measuring a large range of frequencies with two counters, you generate a long pulse using the signal to measure. You then measure the long pulse with a known timebase. The cDAQ controller can measure this long pulse more accurately than the faster input signal.

Note Counter 0 is always paired with Counter 1. Counter 2 is always paired with Counter 3.

You can route the signal to measure to the Source input of Counter 0, as shown in Figure 5-14. Assume this signal to measure has frequency *fx*. NI-DAQmx automatically configures Counter 0 to generate a single pulse that is the width of *N* periods of the source input signal.

Figure 5-14. Large Range of Frequencies with Two Counters



Next, route the Counter 0 Internal Output signal to the Gate input of Counter 1. You can route a signal of known frequency (*fk*) to the Counter 1 Source input. Configure Counter 1 to perform a single pulse-width measurement. Suppose the result is that the pulse width is *J* periods of the *fk* clock.

From Counter 0, the length of the pulse is *N*/*fx*. From Counter 1, the length of the same pulse is *J*/*fk*. Therefore, the frequency of *fx* is given by *fx* = *fk* * (*N*/*J*).

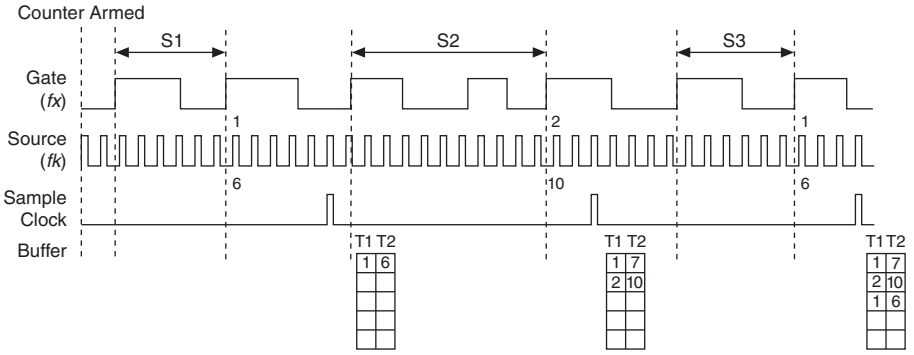
Sample Clocked Buffered Frequency Measurement

Sample clocked buffered point frequency measurements can either be a single frequency measurement or an average between sample clocks. Use **CI.Freq.EnableAveraging** to set the behavior. For buffered frequency, the default is True.

A sample clocked buffered frequency measurement with **CI.Freq.EnableAveraging** set to True uses the embedded counter and a sample clock to perform a frequency measurement. For each sample clock period, the embedded counter counts the signal to measure (*fx*) and the primary counter counts the internal time-base of a known frequency (*fk*). Suppose T1 is the number of ticks of the unknown signal counted between sample clocks and T2 is the number of ticks counted of the known timebase as shown in Figure 5-15. The frequency measured is:

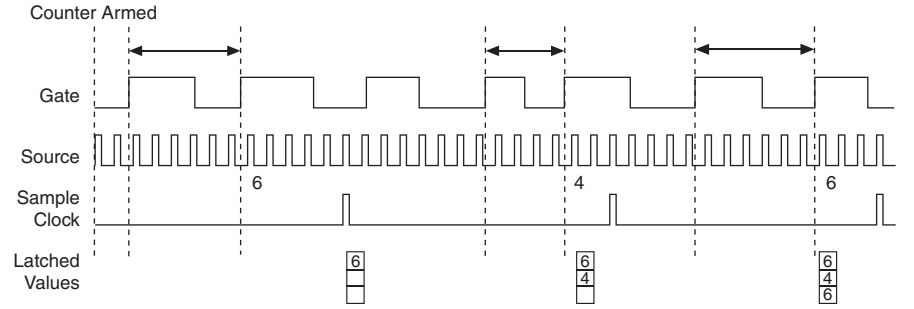
$$fx = fk * (T1/T2)$$

Figure 5-15. Sample Clocked Buffered Frequency Measurement (Averaging)



When **CI.Freq.EnableAveraging** is set to false, the frequency measurement returns the frequency of the pulse just before the sample clock. This single measurement is a single frequency measurement and is not an average between clocks as shown in Figure 5-16.

Figure 5-16. Sample Clocked Buffered Frequency Measurement (Non-Averaging)



With sample clocked frequency measurements, ensure that the frequency to measure is twice as fast as the sample clock to prevent a measurement overflow.

Choosing a Method for Measuring Frequency

The best method to measure frequency depends on several factors including the expected frequency of the signal to measure, the desired accuracy, how many counters are available, and how long the measurement can take. For all frequency measurement methods, assume the following:

fx	is the frequency to be measured if no error
fk	is the known source or gate frequency
measurement time (*T*)	is the time it takes to measure a single sample

Divide down (*N*)	is the integer to divide down measured frequency, only used in large range two counters
fs	is the sample clock rate, only used in sample clocked frequency measurements

Here is how these variables apply to each method, summarized in Table 5-2.

- **One counter**—With one counter measurements, a known timebase is used for the source frequency (*fk*). The measurement time is the period of the frequency to be measured, or 1/*fx*.
- **Two counter high frequency**—With the two counter high frequency method, the second counter provides a known measurement time. The gate frequency equals 1/*measurement time*.
- **Two counter large range**—The two counter larger range measurement is the same as a one counter measurement, but now the user has an integer divide down of the signal. An internal timebase is still used for the source frequency (*fk*), but the divide down means that the measurement time is the period of the divided down signal, or *N*/*fx* where *N* is the divide down.
- **Sample clocked**—For sample clocked frequency measurements, a known timebase is counted for the source frequency (*fk*). The measurement time is the period of the sample clock (*fs*).

Table 5-2. Frequency Measurement Methods

Variable	Sample Clocked	One Counter	Two Counter	
			High Frequency	Large Range
fk	Known timebase	Known timebase	$\frac{1}{\text{gating period}}$	Known timebase
Measurement time	$\frac{1}{fs}$	$\frac{1}{fx}$	gating period	$\frac{N}{fx}$
Max. frequency error	$fx \times \frac{fx}{fk \times \left\lfloor \frac{fx}{fs} - 1 \right\rfloor}$	$fx \times \frac{fx}{fk - fx}$	*fk*	$fx \times \frac{fx}{N \times fk - fx}$
Max. error %	$\frac{fx}{fk \times \left\lfloor \frac{fx}{fs} - 1 \right\rfloor}$	$\frac{fx}{fk - fx}$	$\frac{fk}{fx}$	$\frac{fx}{N \times fk - fx}$

Note: Accuracy equations do not take clock stability into account. Refer to the specifications document for your cDAQ controller for information about clock stability.

Which Method Is Best?

This depends on the frequency to be measured, the rate at which you want to monitor the frequency and the accuracy you desire. Take for example, measuring a 50 kHz signal. Assuming that the measurement times for the sample clocked (with averaging) and two counter frequency measurements are configured the same, Table 5-3 summarizes the results.

Table 5-3. 50 kHz Frequency Measurement Methods

Variable	Sample Clocked	One Counter	Two Counter	
			High Frequency	Large Range
fx	50,000	50,000	50,000	50,000
fk	80 M	80 M	1,000	80 M
Measurement time (*ms*)	1	.02	1	1
N	—	—	—	50
Max. frequency error (Hz)	.638	31.27	1,000	.625
Max. error %	.00128	.0625	2	.00125

From this, you can see that while the measurement time for one counter is shorter, the accuracy is best in the sample clocked and two counter large range measurements. For another example, Table 5-4 shows the results for 5 MHz.

Table 5-4. 5 MHz Frequency Measurement Methods

Variable	Sample Clocked	One Counter	Two Counter	
			High Frequency	Large Range
fx	5 M	5 M	5 M	5 M
fk	80 M	80 M	1,000	80 M
Measurement time (*ms*)	1	.0002	1	1
N	—	—	—	5,000

Table 5-4. 5 MHz Frequency Measurement Methods (Continued)

Variable	Sample Clocked	One Counter	Two Counter	
			High Frequency	Large Range
Max. Frequency error (Hz)	62.51	333 k	1,000	62.50
Max. Error %	.00125	6.67	.02	.00125

Again, the measurement time for the one counter measurement is lowest, but the accuracy is lower. Note that the accuracy and measurement time of the sample clocked and two counter large range are almost the same. The advantage of the sample clocked method is that even when the frequency to measure changes, the measurement time does not and error percentage varies little. For example, if you configured a large range two counter measurement to use a divide down of 50 for a 50 k signal, then you would get the accuracy measurement time and accuracy listed in Table 5-3. But if your signal ramped up to 5 M, then with a divide down of 50, your measurement time is 0.01 ms, but your error is now 0.125%. The error with a sample clocked frequency measurement is not as dependent on the measured frequency so at 50 k and 5 M with a measurement time of 1 ms the error percentage is still close to 0.00125%. One of the disadvantages of a sample clocked frequency measurement is that the frequency to be measured must be at least twice the sample clock rate to ensure that a full period of the frequency to be measured occurs between sample clocks.

- Low frequency measurements with one counter is a good method for many applications. However, the accuracy of the measurement decreases as the frequency increases.
- High frequency measurements with two counters is accurate for high frequency signals. However, the accuracy decreases as the frequency of the signal to measure decreases. At very low frequencies, this method may be too inaccurate for your application. Another disadvantage of this method is that it requires two counters (if you cannot provide an external signal of known width). An advantage of high frequency measurements with two counters is that the measurement completes in a known amount of time.
- Measuring a large range of frequencies with two counters measures high and low frequency signals accurately. However, it requires two counters, and it has a variable sample time and variable error % dependent on the input signal.

Table 5-5 summarizes some of the differences in methods of measuring frequency.

Table 5-5. Frequency Measurement Method Comparison

Method	Number of Counters Used	Number of Measurements Returned	Measures High Frequency Signals Accurately	Measures Low Frequency Signals Accurately
Low frequency with one counter	1	1	Poor	Good
High frequency with two counters	1 or 2	1	Good	Poor
Large range of frequencies with two counters	2	1	Good	Good
Sample clocked (averaged)	1	1	Good	Good

For information about connecting counter signals, refer to the *Default Counter/Timer Routing* section.

Period Measurement

In period measurements, the counter measures a period on its Gate input signal after the counter is armed. You can configure the counter to measure the period between two rising edges or two falling edges of the Gate input signal.

You can route an internal or external periodic clock signal (with a known period) to the Source input of the counter. The counter counts the number of rising (or falling) edges occurring on the Source input between the two active edges of the Gate signal.

You can calculate the period of the Gate input by multiplying the period of the Source signal by the number of edges returned by the counter.

Period measurements return the inverse results of frequency measurements. Refer to the *Frequency Measurement* section for more information.

Position Measurement

You can use the counters to perform position measurements with quadrature encoders or two-pulse encoders. You can measure angular position with X1, X2, and X4 angular encoders. Linear position can be measured with two-pulse encoders. You can choose to do either a single point (on-demand) position measurement or a buffered (sample clock) position measurement. You must arm a counter to begin position measurements.

Refer to the following sections for more information about the cDAQ controller position measurement options:

- *Measurements Using Quadrature Encoders*
- *Measurements Using Two Pulse Encoders*
- *Buffered (Sample Clock) Position Measurement*

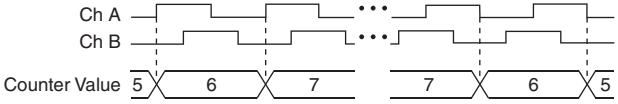
Measurements Using Quadrature Encoders

The counters can perform measurements of quadrature encoders that use X1, X2, or X4 encoding. A quadrature encoder can have up to three channels—channels A, B, and Z.

- **X1 Encoding**—When channel A leads channel B in a quadrature cycle, the counter increments. When channel B leads channel A in a quadrature cycle, the counter decrements. The amount of increments and decrements per cycle depends on the type of encoding—X1, X2, or X4.

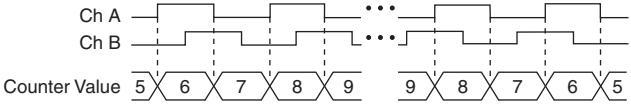
  Figure 5-17 shows a quadrature cycle and the resulting increments and decrements for X1 encoding. When channel A leads channel B, the increment occurs on the rising edge of channel A. When channel B leads channel A, the decrement occurs on the falling edge of channel A.

Figure 5-17. X1 Encoding



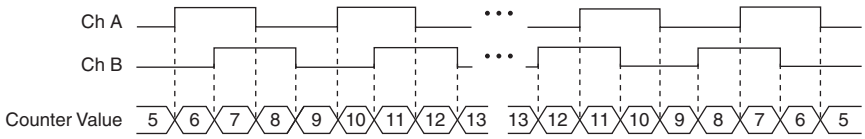
- **X2 Encoding**—The same behavior holds for X2 encoding except the counter increments or decrements on each edge of channel A, depending on which channel leads the other. Each cycle results in two increments or decrements, as shown in Figure 5-18.

Figure 5-18. X2 Encoding



- **X4 Encoding**—Similarly, the counter increments or decrements on each edge of channels A and B for X4 encoding. Whether the counter increments or decrements depends on which channel leads the other. Each cycle results in four increments or decrements, as shown in Figure 5-19.

Figure 5-19. X4 Encoding



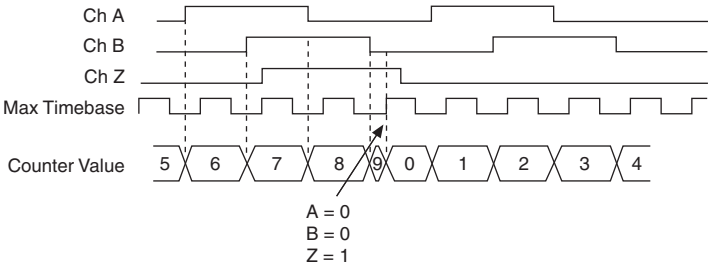
Channel Z Behavior

Some quadrature encoders have a third channel, channel Z, which is also referred to as the index channel. A high level on channel Z causes the counter to be reloaded with a specified value in a specified phase of the quadrature cycle. You can program this reload to occur in any one of the four phases in a quadrature cycle.

Channel Z behavior—when it goes high and how long it stays high—differs with quadrature encoder designs. You must refer to the documentation for your quadrature encoder to obtain timing of channel Z with respect to channels A and B. You must then ensure that channel Z is high during at least a portion of the phase you specify for reload. For instance, in Figure 5-20, channel Z is never high when channel A is high and channel B is low. Thus, the reload must occur in some other phase.

In Figure 5-20, the reload phase is when both channel A and channel B are low. The reload occurs when this phase is true and channel Z is high. Incrementing and decrementing takes priority over reloading. Thus, when the channel B goes low to enter the reload phase, the increment occurs first. The reload occurs within one maximum timebase period after the reload phase becomes true. After the reload occurs, the counter continues to count as before. The figure illustrates channel Z reload with X4 decoding.

Figure 5-20. Channel Z Reload with X4 Decoding

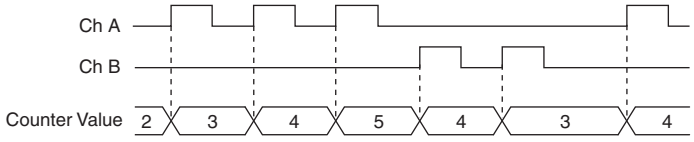


Measurements Using Two Pulse Encoders

The counter supports two pulse encoders that have two channels—channels A and B.

The counter increments on each rising edge of channel A. The counter decrements on each rising edge of channel B, as shown in Figure 5-21.

Figure 5-21. Measurements Using Two Pulse Encoders



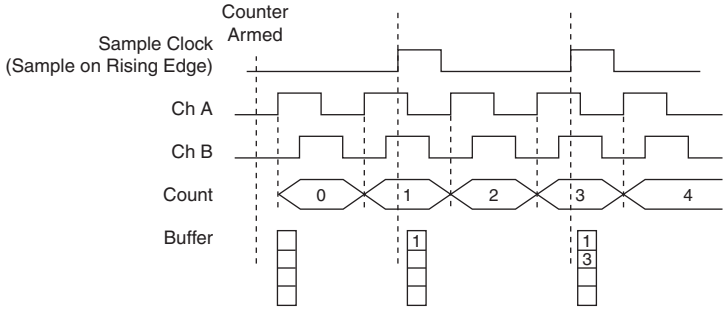
For information about connecting counter signals, refer to the *Default Counter/Timer Routing* section.

Buffered (Sample Clock) Position Measurement

With buffered position measurement (position measurement using a sample clock), the counter increments based on the encoding used after the counter is armed. The value of the counter is sampled on each active edge of a sample clock. The STC3 transfers the sampled values to host memory using a high-speed data stream. The count values returned are the cumulative counts since the counter armed event; that is, the sample clock does not reset the counter. You can route the counter sample clock to the Gate input of the counter. You can configure the counter to sample on the rising or falling edge of the sample clock.

Figure 5-22 shows an example of a buffered X1 position measurement.

Figure 5-22. Buffered Position Measurement



Two-Signal Edge-Separation Measurement

Two-signal edge-separation measurement is similar to pulse-width measurement, except that there are two measurement signals—Aux and Gate. An active edge on the Aux input starts the counting and an active edge on the Gate input stops the counting. You must arm a counter to begin a two edge separation measurement.

After the counter has been armed and an active edge occurs on the Aux input, the counter counts the number of rising (or falling) edges on the Source. The counter ignores additional edges on the Aux input.

The counter stops counting upon receiving an active edge on the Gate input. The counter stores the count in the FIFO.

You can configure the rising or falling edge of the Aux input to be the active edge. You can configure the rising or falling edge of the Gate input to be the active edge.

Use this type of measurement to count events or measure the time that occurs between edges on two signals. This type of measurement is sometimes referred to as start/stop trigger measurement, second gate measurement, or A-to-B measurement.

Refer to the following sections for more information about the cDAQ controller edge-separation measurement options:

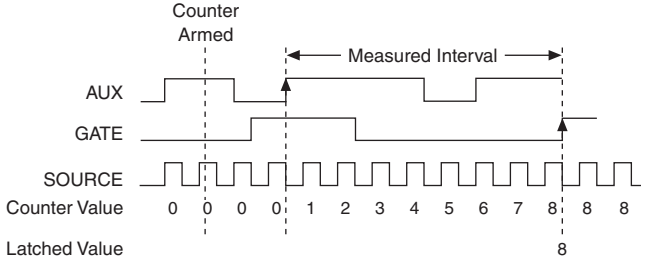
- *Single Two-Signal Edge-Separation Measurement*
- *Implicit Buffered Two-Signal Edge-Separation Measurement*
- *Sample Clocked Buffered Two-Signal Separation Measurement*

Single Two-Signal Edge-Separation Measurement

With single two-signal edge-separation measurement, the counter counts the number of rising (or falling) edges on the Source input occurring between an active edge of the Gate signal and an active edge of the Aux signal. The counter then stores the count in the FIFO and ignores other edges on its inputs. Software then reads the stored count.

Figure 5-23 shows an example of a single two-signal edge-separation measurement.

Figure 5-23. Single Two-Signal Edge-Separation Measurement



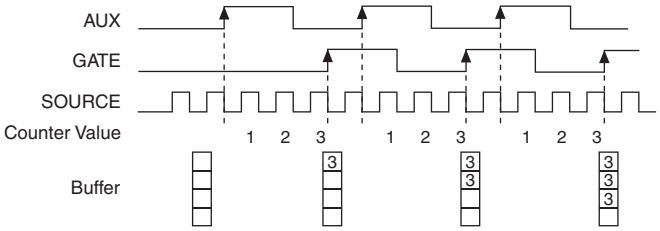
Implicit Buffered Two-Signal Edge-Separation Measurement

Implicit buffered and single two-signal edge-separation measurements are similar, but implicit buffered measurement measures multiple intervals.

The counter counts the number of rising (or falling) edges on the Source input occurring between an active edge of the Gate signal and an active edge of the Aux signal. The counter then stores the count in the FIFO. On the next active edge of the Gate signal, the counter begins another measurement. The STC3 transfers the sampled values to host memory using a high-speed data stream.

Figure 5-24 shows an example of an implicit buffered two-signal edge-separation measurement.

Figure 5-24. Implicit Buffered Two-Signal Edge-Separation Measurement

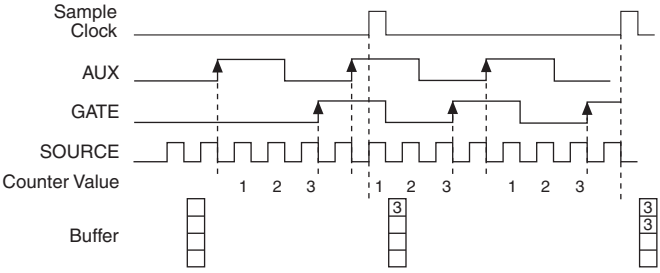


Sample Clocked Buffered Two-Signal Separation Measurement

A sample clocked buffered two-signal separation measurement is similar to single two-signal separation measurement, but buffered two-signal separation measurement takes measurements over multiple intervals correlated to a sample clock. The counter counts the number of rising (or falling) edges on the Source input occurring between an active edge of the Gate signal and an active edge of the Aux signal. The counter then stores the count in the FIFO on a sample clock edge. On the next active edge of the Gate signal, the counter begins another measurement. The STC3 transfers the sampled values to host memory using a high-speed data stream.

Figure 5-25 shows an example of a sample clocked buffered two-signal separation measurement.

Figure 5-25. Sample Clocked Buffered Two-Signal Separation Measurement



Note If an active edge on the Gate and an active edge on the Aux does not occur between sample clocks, an overrun error occurs.

For information about connecting counter signals, refer to the *Default Counter/Timer Routing* section.

Counter Output Applications

The following sections list the various counter output applications available on the cDAQ controller:

- *Simple Pulse Generation*
- *Pulse Train Generation*
- *Frequency Generation*
- *Frequency Division*
- *Pulse Generation for ETS*

Simple Pulse Generation

Refer to the following sections for more information about the cDAQ controller simple pulse generation options:

- *Single Pulse Generation*
- *Single Pulse Generation with Start Trigger*

Single Pulse Generation

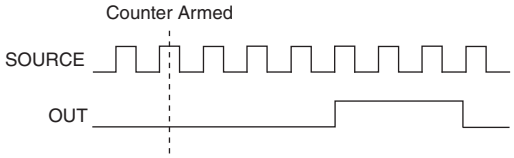
The counter can output a single pulse. The pulse appears on the Counter *n* Internal Output signal of the counter.

You can specify a delay from when the counter is armed to the beginning of the pulse. The delay is measured in terms of a number of active edges of the Source input.

You can specify a pulse width. The pulse width is also measured in terms of a number of active edges of the Source input. You also can specify the active edge of the Source input (rising or falling).

Figure 5-26 shows a generation of a pulse with a pulse delay of four and a pulse width of three (using the rising edge of Source).

Figure 5-26. Single Pulse Generation



Single Pulse Generation with Start Trigger

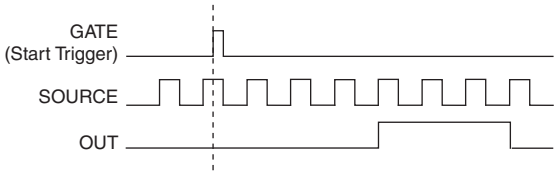
The counter can output a single pulse in response to one pulse on a hardware Start Trigger signal. The pulse appears on the Counter *n* Internal Output signal of the counter.

You can specify a delay from the Start Trigger to the beginning of the pulse. You also can specify the pulse width. The delay is measured in terms of a number of active edges of the Source input.

You can specify a pulse width. The pulse width is also measured in terms of a number of active edges of the Source input. You can also specify the active edge of the Source input (rising and falling).

Figure 5-27 shows a generation of a pulse with a pulse delay of four and a pulse width of three (using the rising edge of Source).

Figure 5-27. Single Pulse Generation with Start Trigger



Pulse Train Generation

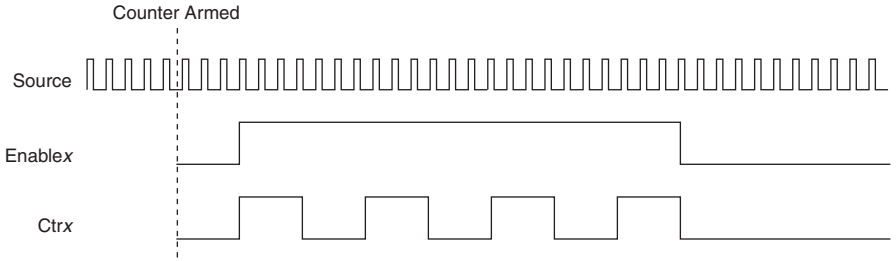
Refer to the following sections for more information about the cDAQ controller pulse train generation options:

- *Finite Pulse Train Generation*
- *Retriggerable Pulse or Pulse Train Generation*
- *Continuous Pulse Train Generation*
- *Buffered Pulse Train Generation*
- *Finite Implicit Buffered Pulse Train Generation*
- *Continuous Buffered Implicit Pulse Train Generation*
- *Finite Buffered Sample Clocked Pulse Train Generation*
- *Continuous Buffered Sample Clocked Pulse Train Generation*

Finite Pulse Train Generation

This function generates a train of pulses with programmable frequency and duty cycle for a predetermined number of pulses. With cDAQ controller counters, the primary counter generates the specified pulse train and the embedded counter counts the pulses generated by the primary counter. When the embedded counter reaches the specified tick count, it generates a trigger that stops the primary counter generation.

Figure 5-28. Finite Pulse Train Generation: Four Ticks Initial Delay, Four Pulses



Retriggerable Pulse or Pulse Train Generation

The counter can output a single pulse or multiple pulses in response to each pulse on a hardware Start Trigger signal. The generated pulses appear on the Counter *n* Internal Output signal of the counter.

You can route the Start Trigger signal to the Gate input of the counter. You can specify a delay from the Start Trigger to the beginning of each pulse. You also can specify the pulse width. The delay and pulse width are measured in terms of a number of active edges of the Source input. The initial delay can be applied to only the first trigger or to all triggers using the **CO.EnableInitalDelayOnRetrigger** property. The default for a single pulse is True, while the default for finite pulse trains is False.

The counter ignores the Gate input while a pulse generation is in progress. After the pulse generation is finished, the counter waits for another Start Trigger signal to begin another pulse generation. For retriggered pulse generation, pause triggers are not allowed since the pause trigger also uses the gate input.

Figure 5-29 shows a generation of two pulses with a pulse delay of five and a pulse width of three (using the rising edge of Source) with **CO.EnableInitalDelayOnRetrigge**r set to the default True.

Figure 5-29. Retriggerable Single Pulse Generation with Initial Delay on Retrigger

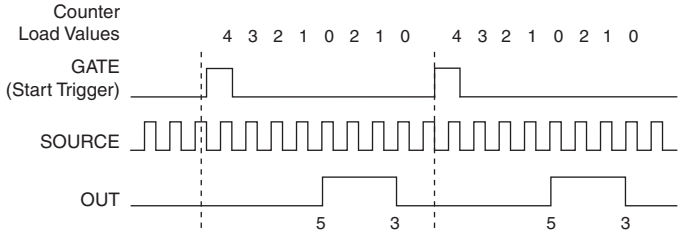
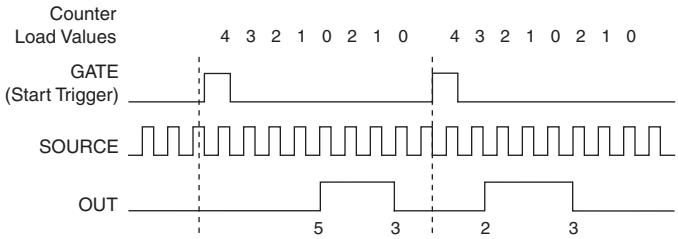


Figure 5-30 shows the same pulse train with **CO.EnableInitalDelayOnRetrigger** set to the default False.

Figure 5-30. Retriggerable Single Pulse Generation False



Note The minimum time between the trigger and the first active edge is two ticks of the source.

For information about connecting counter signals, refer to the *Default Counter/Timer Routing* section.

Continuous Pulse Train Generation

This function generates a train of pulses with programmable frequency and duty cycle. The pulses appear on the Counter *n* Internal Output signal of the counter.

You can specify a delay from when the counter is armed to the beginning of the pulse train. The delay is measured in terms of a number of active edges of the Source input.

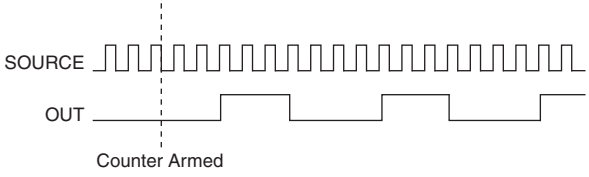
You specify the high and low pulse widths of the output signal. The pulse widths are also measured in terms of a number of active edges of the Source input. You also can specify the active edge of the Source input (rising or falling).

The counter can begin the pulse train generation as soon as the counter is armed, or in response to a hardware Start Trigger. You can route the Start Trigger to the Gate input of the counter.

You also can use the Gate input of the counter as a Pause Trigger (if it is not used as a Start Trigger). The counter pauses pulse generation when the Pause Trigger is active.

Figure 5-31 shows a continuous pulse train generation (using the rising edge of Source).

Figure 5-31. Continuous Pulse Train Generation



Continuous pulse train generation is sometimes called frequency division. If the high and low pulse widths of the output signal are *M* and *N* periods, then the frequency of the Counter *n* Internal Output signal is equal to the frequency of the Source input divided by *M* + *N*.

For information about connecting counter signals, refer to the *Default Counter/Timer Routing* section.

Buffered Pulse Train Generation

The cDAQ controller counters can use the FIFO to perform a buffered pulse train generation. This pulse train can use implicit timing or sample clock timing. When using implicit timing, the pulse idle time and active time changes with each sample you write. With sample clocked timing, each sample you write updates the idle time and active time of your generation on each sample clock edge. Idle time and active time can also be defined in terms of frequency and duty cycle or idle ticks and active ticks.

Note On buffered implicit pulse trains the pulse specifications in the DAQmx Create Counter Output Channel are ignored so that you generate the number of pulses defined in the multipoint write. On buffered sample clock pulse trains the pulse specifications in the DAQmx Create Counter Output Channel are generated after the counters starts and before the first sample clock so that you generate the number of updates defined in the multipoint write.

Finite Implicit Buffered Pulse Train Generation

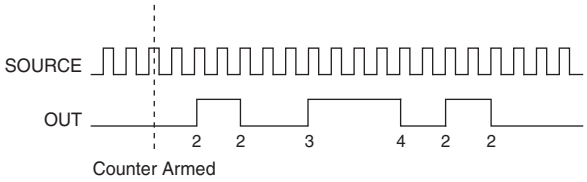
This function generates a predetermined number of pulses with variable idle and active times. Each point you write generates a single pulse. The number of pairs of idle and active times (pulse specifications) you write determines the number of pulses generated. All points are generated back to back to create a user defined pulse train.

Table 5-6 and Figure 5-32 detail a finite implicit generation of three samples.

Table 5-6. Finite Implicit Buffered Pulse Train Generation

Sample	Idle Ticks	Active Ticks
1	2	2
2	3	4
3	2	2

Figure 5-32. Finite Implicit Buffered Pulse Train Generation



Continuous Buffered Implicit Pulse Train Generation

This function generates a continuous train of pulses with variable idle and active times. Instead of generating a set number of data samples and stopping, a continuous generation continues until you stop the operation. Each point you write generates a single pulse. All points are generated back to back to create a user defined pulse train.

Finite Buffered Sample Clocked Pulse Train Generation

This function generates a predetermined number of pulse train updates. Each point you write defines pulse specifications that are updated with each sample clock. When a sample clock occurs, the current pulse (idle followed by active) finishes generation and the next pulse updates with the next sample specifications.

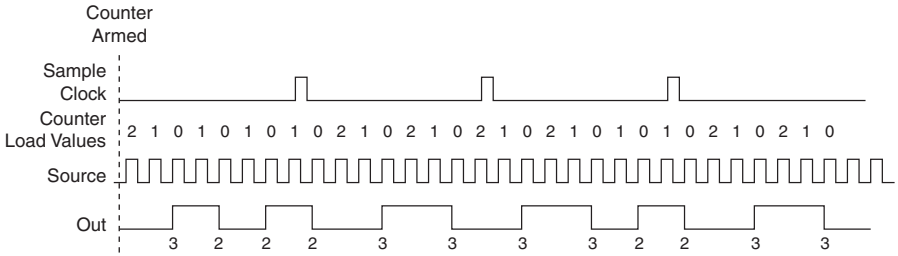
Note When the last sample is generated, the pulse train continues to generate with these specifications until the task is stopped.

Table 5-7 and Figure 5-33 detail a finite sample clocked generation of three samples where the pulse specifications from the create channel are two ticks idle, two ticks active, and three ticks initial delay.

Table 5-7. Finite Buffered Sample Clocked Pulse Train Generation

Sample	Idle Ticks	Active Ticks
1	3	3
2	2	2
3	3	3

Figure 5-33. Finite Buffered Sample Clocked Pulse Train Generation



There are several different methods of continuous generation that control what data is written. These methods are regeneration, FIFO regeneration, and non-regeneration modes.

Regeneration is the repetition of the data that is already in the buffer.

Standard regeneration is when data from the PC buffer is continually downloaded to the FIFO to be written out. New data can be written to the PC buffer at any time without disrupting the output. With FIFO regeneration, the entire buffer is downloaded to the FIFO and regenerated from there. Once the data is downloaded, new data cannot be written to the FIFO. To use FIFO regeneration, the entire buffer must fit within the FIFO size. The advantage of using FIFO regeneration is that it does not require communication with the main host memory once the operation is started, thereby preventing any problems that may occur due to excessive bus traffic.

With non-regeneration, old data is not repeated. New data must be continually written to the buffer. If the program does not write new data to the buffer at a fast enough rate to keep up with the generation, the buffer underflows and causes an error.

Continuous Buffered Sample Clocked Pulse Train Generation

This function generates a continuous train of pulses with variable idle and active times. Instead of generating a set number of data samples and stopping, a continuous generation continues until you stop the operation. Each point you write specifies pulse specifications that are updated with each sample clock. When a sample clock occurs, the current pulse finishes generation and the next pulse uses the next sample specifications.

Frequency Generation

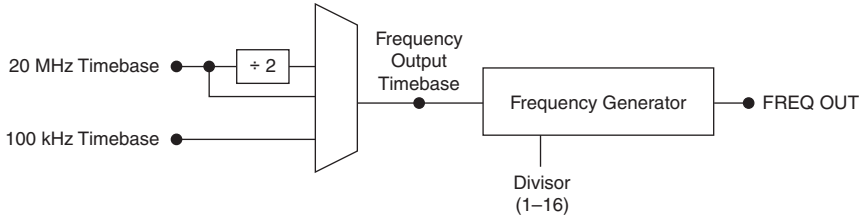
You can generate a frequency by using a counter in pulse train generation mode or by using the frequency generator circuit, as described in the *Using the Frequency Generator* section.

Using the Frequency Generator

The frequency generator can output a square wave at many different frequencies. The frequency generator is independent of the four general-purpose 32-bit counter/timer modules on the cDAQ controller.

Figure 5-34 shows a block diagram of the frequency generator.

Figure 5-34. Frequency Generator Block Diagram

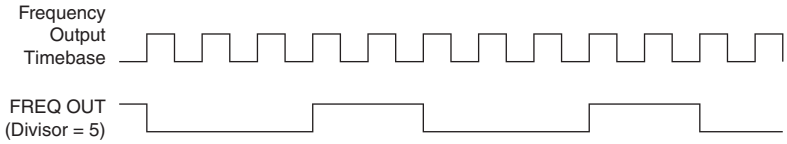


The frequency generator generates the Frequency Output signal. The Frequency Output signal is the Frequency Output Timebase divided by a number you select from 1 to 16. The Frequency Output Timebase can be either the 20 MHz Timebase, the 20 MHz Timebase divided by 2, or the 100 kHz Timebase.

The duty cycle of Frequency Output is 50% if the divider is either 1 or an even number. For an odd divider, suppose the divider is set to D. In this case, Frequency Output is low for (D + 1)/2 cycles and high for (D - 1)/2 cycles of the Frequency Output Timebase.

Figure 5-35 shows the output waveform of the frequency generator when the divider is set to 5.

Figure 5-35. Frequency Generator Output Waveform



Frequency Output can be routed out to any PFI terminal. All PFI terminals are set to high-impedance at startup. The FREQ OUT signal also can be routed to many internal timing signals.

In software, program the frequency generator as you would program one of the counters for pulse train generation.

For information about connecting counter signals, refer to the *Default Counter/Timer Routing* section.

Frequency Division

The counters can generate a signal with a frequency that is a fraction of an input signal. This function is equivalent to continuous pulse train generation. Refer to the *Continuous Pulse Train Generation* section for detailed information.

For information about connecting counter signals, refer to the *Default Counter/Timer Routing* section.

Pulse Generation for ETS

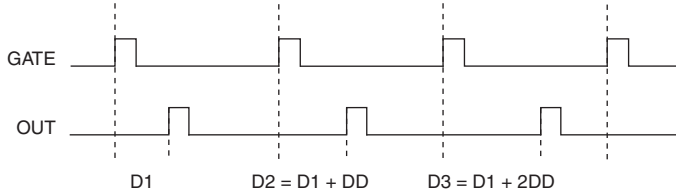
In the equivalent time sampling (ETS) application, the counter produces a pulse on the output a specified delay after an active edge on Gate. After each active edge on Gate, the counter cumulatively increments the delay between the Gate and the pulse on the output by a specified amount. Thus, the delay between the Gate and the pulse produced successively increases.

The increase in the delay value can be between 0 and 255. For instance, if you specify the increment to be 10, the delay between the active Gate edge and the pulse on the output increases by 10 every time a new pulse is generated.

Suppose you program your counter to generate pulses with a delay of 100 and pulse width of 200 each time it receives a trigger. Furthermore, suppose you specify the delay increment to be 10. On the first trigger, your pulse delay will be 100, on the second it will be 110, on the third it will be 120; the process will repeat in this manner until the counter is disarmed. The counter ignores any Gate edge that is received while the pulse triggered by the previous Gate edge is in progress.

The waveform thus produced at the counter's output can be used to provide timing for undersampling applications where a digitizing system can sample repetitive waveforms that are higher in frequency than the Nyquist frequency of the system. Figure 5-36 shows an example of pulse generation for ETS; the delay from the trigger to the pulse increases after each subsequent Gate active edge.

Figure 5-36. Pulse Generation for ETS



For information about connecting counter signals, refer to the *Default Counter/Timer Routing* section.

Counter Timing Signals

The cDAQ controller features the following counter timing signals:

- *Counter n Source Signal*
- *Counter n Gate Signal*
- *Counter n Aux Signal*
- *Counter n A Signal*
- *Counter n B Signal*
- *Counter n Z Signal*
- *Counter n Up_Down Signal*

- *Counter n HW Arm Signal*
- *Counter n Sample Clock Signal*
- *Counter n Internal Output Signal*
- *Counter n TC Signal*
- *Frequency Output Signal*

In this section, *n* refers to the cDAQ controller Counter 0, 1, 2, or 3. For example, Counter *n* Source refers to four signals—Counter 0 Source (the source input to Counter 0), Counter 1 Source (the source input to Counter 1), Counter 2 Source (the source input to Counter 2), or Counter 3 Source (the source input to Counter 3).

Note All counter timing signals can be filtered. Refer to the *PFI Filters* section of Chapter 4, *Digital Input/Output and PFI*, for more information.

Counter *n* Source Signal

The selected edge of the Counter *n* Source signal increments and decrements the counter value depending on the application the counter is performing. Table 5-8 lists how this terminal is used in various applications.

Table 5-8. Counter Applications and Counter *n* Source

Application	Purpose of Source Terminal
Pulse Generation	Counter Timebase
One Counter Time Measurements	Counter Timebase
Two Counter Time Measurements	Input Terminal
Non-Buffered Edge Counting	Input Terminal
Buffered Edge Counting	Input Terminal
Two-Edge Separation	Counter Timebase

Routing a Signal to Counter *n* Source

Each counter has independent input selectors for the Counter *n* Source signal. Any of the following signals can be routed to the Counter *n* Source input:

- 80 MHz Timebase
- 20 MHz Timebase
- 100 kHz Timebase
- Any PFI terminal
- Analog Comparison Event
- Change Detection Event

In addition, TC or Gate from a counter can be routed to a different counter source.

Some of these options may not be available in some driver software. Refer to the *Device Routing in MAX* topic in the *NI-DAQmx Help* or the *LabVIEW Help* for more information about available routing options.

Routing Counter *n* Source to an Output Terminal

You can route Counter *n* Source out to any PFI terminal.

Counter *n* Gate Signal

The Counter *n* Gate signal can perform many different operations depending on the application including starting and stopping the counter, and saving the counter contents.

Routing a Signal to Counter *n* Gate

Each counter has independent input selectors for the Counter *n* Gate signal. Any of the following signals can be routed to the Counter *n* Gate input:

- Any PFI terminal
- AI Reference Trigger
- AI Start Trigger
- AO Sample Clock
- DI Sample Clock
- DI Reference Trigger
- DO Sample Clock
- Change Detection Event
- Analog Comparison Event

In addition, a counter's Internal Output or Source can be routed to a different counter's gate.

Some of these options may not be available in some driver software. Refer to the *Device Routing in MAX* topic in the *NI-DAQmx Help* or the *LabVIEW Help* for more information about available routing options.

Routing Counter *n* Gate to an Output Terminal

You can route Counter *n* Gate out to any PFI terminal.

Counter *n* Aux Signal

The Counter *n* Aux signal indicates the first edge in a two-signal edge-separation measurement.

Routing a Signal to Counter *n* Aux

Each counter has independent input selectors for the Counter *n* Aux signal. Any of the following signals can be routed to the Counter *n* Aux input:

- Any PFI terminal
- AI Reference Trigger
- AI Start Trigger
- Analog Comparison Event
- Change Detection Event

In addition, a counter's Internal Output, Gate or Source can be routed to a different counter's Aux. A counter's own gate can also be routed to its Aux input.

Some of these options may not be available in some driver software. Refer to the *Device Routing in MAX* topic in the *NI-DAQmx Help* or the *LabVIEW Help* for more information about available routing options.

Counter *n* A, Counter *n* B, and Counter *n* Z Signals

Counter *n* B can control the direction of counting in edge counting applications. Use the A, B, and Z inputs to each counter when measuring quadrature encoders or measuring two pulse encoders.

Routing Signals to A, B, and Z Counter Inputs

Each counter has independent input selectors for each of the A, B, and Z inputs. Any of the following signals can be routed to each input:

- Any PFI terminal
- Analog Comparison Event

Routing Counter *n* Z Signal to an Output Terminal

You can route Counter *n* Z out to any PFI terminal.

Counter *n* Up_Down Signal

Counter *n* Up_Down is another name for the Counter *n* B signal.

Counter *n* HW Arm Signal

The Counter *n* HW Arm signal enables a counter to begin an input or output function.

To begin any counter input or output function, you must first enable, or arm, the counter. In some applications, such as a buffered edge count, the counter begins counting when it is armed. In other applications, such as single pulse-width measurement, the counter begins waiting for the Gate signal when it is armed. Counter output operations can use the arm signal in addition to a start trigger.

Software can arm a counter or configure counters to be armed on a hardware signal. Software calls this hardware signal the Arm Start Trigger. Internally, software routes the Arm Start Trigger to the Counter *n* HW Arm input of the counter.

Routing Signals to Counter *n* HW Arm Input

Any of the following signals can be routed to the Counter *n* HW Arm input:

- Any PFI terminal
- AI Reference Trigger
- AI Start Trigger
- Analog Comparison Event
- Change Detection Event

A counter's Internal Output can be routed to a different counter's HW Arm.

Some of these options may not be available in some driver software. Refer to the *Device Routing in MAX* topic in the *NI-DAQmx Help* or the *LabVIEW Help* for more information about available routing options.

Counter *n* Sample Clock Signal

Use the Counter *n* Sample Clock (Ctr*n*SampleClock) signal to perform sample clocked acquisitions and generations.

You can specify an internal or external source for Counter *n* Sample Clock. You also can specify whether the measurement sample begins on the rising edge or falling edge of Counter *n* Sample Clock.

If the cDAQ controller receives a Counter *n* Sample Clock when the FIFO is full, it reports an overflow error to the host software.

Using an Internal Source

To use Counter *n* Sample Clock with an internal source, specify the signal source and the polarity of the signal. The source can be any of the following signals:

- DI Sample Clock
- DO Sample Clock
- AI Sample Clock (ai/SampleClock, te0/SampleClock, te1/SampleClock)
- AI Convert Clock
- AO Sample Clock
- DI Change Detection output

Several other internal signals can be routed to Counter *n* Sample Clock through internal routes. Refer to *Device Routing in MAX* in the *NI-DAQmx Help* or the *LabVIEW Help* for more information.

Using an External Source

You can route any of the following signals as Counter *n* Sample Clock:

- Any PFI terminal
- Analog Comparison Event

You can sample data on the rising or falling edge of Counter *n* Sample Clock.

Routing Counter *n* Sample Clock to an Output Terminal

You can route Counter *n* Sample Clock out to any PFI terminal. The PFI circuitry inverts the polarity of Counter n Sample Clock before driving the PFI terminal.

Counter *n* Internal Output and Counter *n* TC Signals

The Counter *n* Internal Output signal changes in response to Counter *n* TC.

The two software-selectable output options are pulse output on TC and toggle output on TC. The output polarity is software-selectable for both options.

With pulse or pulse train generation tasks, the counter drives the pulse(s) on the Counter *n* Internal Output signal. The Counter *n* Internal Output signal can be internally routed to be a counter/timer input or an “external” source for AI, AO, DI, or DO timing signals.

Routing Counter *n* Internal Output to an Output Terminal

You can route Counter *n* Internal Output to any PFI terminal.

Frequency Output Signal

The Frequency Output (FREQ OUT) signal is the output of the frequency output generator.

Routing Frequency Output to a Terminal

You can route Frequency Output to any PFI terminal.

Default Counter/Timer Routing

Counter/timer signals are available to parallel digital I/O C Series modules. To determine the signal routing options for modules installed in your system, refer to the **Device Routes** tab in MAX.

You can use these defaults or select other sources and destinations for the counter/timer signals in NI-DAQmx. Refer to *Connecting Counter Signals* in the *NI-DAQmx Help* or the *LabVIEW Help* for more information about how to connect your signals for common counter measurements and generations. Refer to *Physical Channels* in the *NI-DAQmx Help* or the *LabVIEW Help* for a list of default PFI lines for counter functions.

Counter Triggering

Counters support three different triggering actions:

- **Arm Start Trigger**—To begin any counter input or output function, you must first enable, or arm, the counter. Software can arm a counter or configure counters to be armed on a hardware signal. Software calls this hardware signal the Arm Start Trigger. Internally, software routes the Arm Start Trigger to the Counter *n* HW Arm input of the counter.

  For counter output operations, you can use it in addition to the start and pause triggers. For counter input operations, you can use the arm start trigger to have start trigger-like behavior. The arm start trigger can be used for synchronizing multiple counter input and output tasks.

  When using an arm start trigger, the arm start trigger source is routed to the Counter *n* HW Arm signal.
- **Start Trigger**—For counter output operations, a start trigger can be configured to begin a finite or continuous pulse generation. Once a continuous generation has triggered, the pulses continue to generate until you stop the operation in software. For finite generations, the specified number of pulses is generated and the generation stops unless you use the retriggerable attribute. When you use this attribute, subsequent start triggers cause the generation to restart.

  When using a start trigger, the start trigger source is routed to the Counter *n* Gate signal input of the counter. Counter input operations can use the arm start trigger to have start trigger-like behavior.
- **Pause Trigger**—You can use pause triggers in edge counting and continuous pulse generation applications. For edge counting acquisitions, the counter stops counting edges while the external trigger signal is low and resumes when the signal goes high or vice versa. For continuous pulse generations, the counter stops generating pulses while the external trigger signal is low and resumes when the signal goes high or vice versa.

  When using a pause trigger, the pause trigger source is routed to the Counter *n* Gate signal input of the counter.

Other Counter Features

The following sections list the other counter features available on the cDAQ controller.

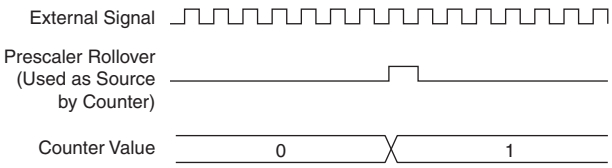
Cascading Counters

You can internally route the Counter n Internal Output and Counter n TC signals of each counter to the Gate inputs of the other counter. By cascading two counters together, you can effectively create a 64-bit counter. By cascading counters, you also can enable other applications. For example, to improve the accuracy of frequency measurements, use reciprocal frequency measurement, as described in the *Large Range of Frequencies with Two Counters* section.

Prescaling

Prescaling allows the counter to count a signal that is faster than the maximum timebase of the counter. The cDAQ controller offers 8X and 2X prescaling on each counter (prescaling can be disabled). Each prescaler consists of a small, simple counter that counts to eight (or two) and rolls over. This counter can run faster than the larger counters, which simply count the rollovers of this smaller counter. Thus, the prescaler acts as a frequency divider on the Source and puts out a frequency that is one-eighth (or one-half) of what it is accepting as shown in Figure 5-37.

Figure 5-37. Prescaling



Prescaling is intended to be used for frequency measurement where the measurement is made on a continuous, repetitive signal. The prescaling counter cannot be read; therefore, you cannot determine how many edges have occurred since the previous rollover. Prescaling can be used for event counting provided it is acceptable to have an error of up to seven (or one) ticks. Prescaling can be used when the counter Source is an external signal. Prescaling is not available if the counter Source is one of the internal timebases (80MHzTimebase, 20MHzTimebase, or 100kHzTimebase).

Synchronization Modes

The 32-bit counter counts up or down synchronously with the Source signal. The Gate signal and other counter inputs are asynchronous to the Source signal, so the cDAQ controller synchronizes these signals before presenting them to the internal counter.

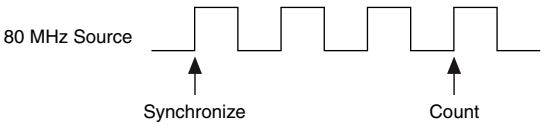
Depending on how you configure your controller, the cDAQ controller uses one of two synchronization methods:

- *80 MHz Source Mode*
- *External or Internal Source Less than 20 MHz*

80 MHz Source Mode

In 80 MHz source mode, the controller synchronizes signals on the rising edge of the source, and counts on the third rising edge of the source. Edges are pipelined so no counts are lost, as shown in Figure 5-38.

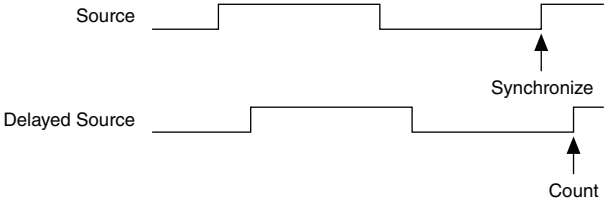
Figure 5-38. 80 MHz Source Mode



External or Internal Source Less than 20 MHz

With an external or internal source less than 20 MHz, the module generates a delayed Source signal by delaying the Source signal by several nanoseconds. The controller synchronizes signals on the rising edge of the delayed Source signal, and counts on the following rising edge of the source, as shown in Figure 5-39.

Figure 5-39. External or Internal Source Less than 20 MHz



6

Digital Routing and Clock Generation

This chapter describes the digital routing and clock routing circuitry on the cDAQ controller.

Digital Routing

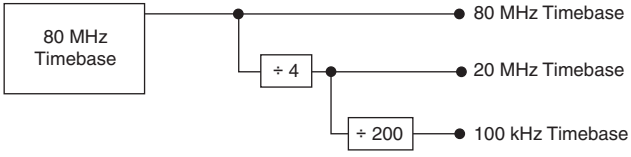
The digital routing circuitry has the following functions:

- Manages the flow of data between the bus interface and the acquisition/generation sub-systems (analog input, analog output, digital I/O, and the counters). The digital routing circuitry uses FIFOs (if present) in each sub-system to ensure efficient data movement.
- Routes timing and control signals. The acquisition/generation sub-systems use these signals to manage acquisitions and generations. These signals can come from the following sources:
  - Your C Series modules
  - User input through the PFI terminals using parallel digital C Series modules
- Routes and generates the main clock signals for the cDAQ controller. To determine the signal routing options for C Series module(s) installed in the cDAQ controller, refer to the **Device Routes** tab in MAX.

Clock Routing

Figure 6-1 shows the clock routing circuitry of the cDAQ controller.

Figure 6-1. Clock Routing Circuitry



80 MHz Timebase

You can use the 80 MHz Timebase as the Source input to the 32-bit general-purpose counter/timers.

20 MHz Timebase

The 20 MHz Timebase normally generates many of the AI and AO timing signals. It can function as the Source input to the 32-bit general-purpose counter/timers.

The 20 MHz Timebase is generated by dividing down the 80 MHz Timebase, as shown in Figure 6-1.

100 kHz Timebase

You can use the 100 kHz Timebase to generate many of the AI and AO timing signals. It can also function as the Source input to the 32-bit general-purpose counter/timers.

The 100 kHz Timebase is generated by dividing down the 20 MHz Timebase by 200, as shown in Figure 6-1.

A

Controller Operating System and BIOS Configuration

This appendix covers the following topics regarding the NI cDAQ-9132/9133/9134/9135/9136/9137 controller configurations:

- *Power-On Self Test (POST) Warning Messages*
- *Restoring the Windows Operating System*
- *Using the BIOS Setup Utility to Change Configuration Settings*
- *Resetting the BIOS Settings*

Power-On Self Test (POST) Warning Messages

The cDAQ controller POST displays warning messages for specific issues onscreen:

- **BIOS Reset Detected**—This warning is displayed when the CMOS reset button has been pushed. This warning indicates that the BIOS settings have the default values.
- **CMOS Battery Is Dead**—This warning is displayed when the CMOS battery is dead and must be replaced. The BIOS settings are preserved even when the CMOS battery is dead, but the system will boot very slowly because the BIOS cannot optimize boot time by saving specific system information to CMOS.

Note When a warning message is displayed, 10 additional seconds are added to the POST to give the user time to read the warning.

Restoring the Windows Operating System

(NI cDAQ-9132/9133/9134/9135/9136/9137 for Windows) You can restore the Windows operating system on the hard drive of the cDAQ controller from the *NI cDAQ-9132/9133/9134/9135/9136/9137 Controller OEM Re-Installation* media shipped with the cDAQ controller.

Note Restoring the operating system erases the contents of the hard drive. Back up any files you want to keep before restoring the hard drive.

Note Before restoring the operating system, you can return the cDAQ controller to factory-default condition by resetting the BIOS settings, as listed in the *Resetting the System CMOS and BIOS Settings* section of Chapter 1, *Getting Started with the cDAQ Controller*.

To use the recovery media, complete the following steps.

1. Connect an external DVD drive through a USB hub to one of the USB ports of the cDAQ controller and insert the recovery media.
2. Connect a keyboard to the other USB port on the cDAQ controller. Connect the mouse to the USB hub.
3. Power on the cDAQ system.
4. Press the <F10> key on the keyboard.
5. Select the DVD drive or recovery media.
6. Press any key to boot from the recovery media.
7. In the Windows Recovery Configuration, select your language and the hard drive configuration for Windows Embedded Standard 7 (WES7) SP1 installation.

After restoring the operating system, you can reinstall the software and drivers onto the controller in the following order:

1. LabVIEW, as described in the *LabVIEW Installation Guide*
2. NI-DAQmx
3. **(NI cDAQ-9134/9135 for Windows)** NI-XNET, as described in the *NI-XNET Hardware and Software Installation Guide*

Note For information about how to improve robustness on the Windows system, go to `ni.com/info` and enter the Info Code `extxxx`.

Using the BIOS Setup Utility to Change Configuration Settings

The cDAQ controller is shipped with configuration settings that work well for most applications. However, if your application requires different settings, you can use the BIOS setup utility to change settings. You can also use the BIOS setup utility to enable special controller functions. This section includes the following topics:

- *Launching the BIOS Setup Utility*
- *Main Setup Menu*
- *Advanced Setup Menu*
- *Security Menu*
- *Boot Setup Menu*
- *Save & Exit Menu*

Changing BIOS settings can cause incorrect controller behavior, including failure to boot. In general, do *not* change a setting unless you are absolutely sure what the setting does. Refer to the *Resetting the BIOS Settings* section for information about restoring the default configuration settings.

Launching the BIOS Setup Utility

Complete the following steps to launch the BIOS setup utility.

1. Connect a video monitor to the mini DisplayPort connector on the cDAQ controller.
2. Connect a USB keyboard to one of the USB host ports on the cDAQ controller.
3. Power on or reboot the cDAQ controller.
4. Hold down either the <F10> key or the <Del> key until the message **`Please select boot device:`** appears onscreen.
5. Use the Down Arrow key to select **`Enter Setup`** and press <Enter>. The setup utility loads after a short delay.

The Main setup menu is displayed when you first enter the BIOS setup utility. Use the keys listed in Table A-1 to navigate through the BIOS setup utility.

Table A-1. BIOS Setup Utility Keyboard Navigation

Key(s)	Function(s)
Left Arrow, Right Arrow	Move between the different setup menus. If you are in a submenu, these keys have no effect, and you must press <Esc> to leave the submenu first. (To use the arrows on the numeric keypad, you must disable Num Lock.)
Up Arrow, Down Arrow	Move between the options within a setup menu. (To use the arrows on the numeric keypad, you must disable Num Lock.)
<Enter>	Either enters a submenu or displays all available settings for a highlighted configuration option.
<Esc>	Returns to the parent menu of a submenu. At the top-level menus, this key serves as a shortcut to the Exit menu.
<+>, <->	Cycle between all available settings for a selected configuration option.
<Tab>	Selects time and date fields.
<F9>	Loads the optimal default values for all BIOS configuration settings. The optimal default values are the same as the shipping configuration default values.
<F10>	Saves settings and exits the BIOS setup utility.

Main Setup Menu

The most commonly accessed and modified BIOS settings are in the Main setup menu. The Main setup menu reports the following configuration information:

- **BIOS Version and Build Date**—These values indicate the version of the controller BIOS and the date on which the BIOS was built.
- **Embedded Firmware Version**—This value identifies the built-in hardware capabilities.
- **Processor Type, Base Processor Frequency, and Active Processor Core**—These values indicate the type of processor used in the controller, the speed of the processor, and the number of active processor cores.
- **Total Memory**—This value indicates the size of system RAM detected by the BIOS.

The Main setup menu also includes the following settings:

- **System Date**—This setting controls the date, which is stored in a battery-backed real-time clock. Most operating systems also include a way to change this setting. Use <+> and <-> in conjunction with <Enter> and <Tab> to change these values.
- **System Time**—This setting controls the time of day, which is stored in a battery-backed real-time clock. Most operating systems also include a way to change this setting. Use <+> and <-> in conjunction with <Enter> and <Tab> to change these values.

Advanced Setup Menu

This menu contains BIOS settings that normally do not require modification. If you have specific problems, such as unbootable disks or resource conflicts, you may need to examine these settings.

Caution Changing settings in this menu can result in an unstable or unbootable controller. If this happens, follow the procedures outlined in the *Resetting the BIOS Settings* section to restore BIOS settings to the factory defaults.

The Advanced setup menu includes the following settings and submenus:

- *Power/Wake Configuration Submenu*—Use this setting to access the Power/Wake Configuration submenu.
- *SATA Configuration Submenu*—Use this setting to access the SATA Configuration submenu.
- *USB Configuration Submenu*—Use this setting to access the USB Configuration submenu.

Power/Wake Configuration Submenu

Use this submenu to apply alternate configurations to the power and wake features of the chipset and controller. Normally, you do not need to modify these settings because the factory default settings provide the most compatible and optimal configuration possible.

- **Restore After Power Loss**—This setting specifies the power state that the controller should return to after DC power is lost. Valid values are **Stay Off** and **Turn On**. The default value is **Turn On**. When set to **Stay Off**, the controller returns to the soft off power state after AC power is restored. When set to **Turn On**, the controller powers on when DC power is restored.
- **Power Button Off Behavior**—This setting specifies how the system responds to the power button. Valid options are **Normal** and **Disabled**. The default value is **Normal**. If the value is **Normal**, the system responds to the power button as defined by the OS. If the value is **Disabled**, pressing the power button has no effect when the system is on. When the system is in the soft off state, pushing the power button always powers on the system. Refer to the *Power Button* section of Chapter 1, *Getting Started with the cDAQ Controller*, for more information.
- **Ring Indicator Wake**—This setting enables or disables the ability to wake a powered-off system using the Ring Indicator pin of the RS-232 serial port. The default value is **Disabled**. Refer to the *RS-232 Serial Port* section of Chapter 1, *Getting Started with the cDAQ Controller*, for more information.
- **Wake On Trigger**—This setting enables or disables the ability to wake a powered-off system using the PFI 0 connector on the front panel. The default value is **Disabled**. Refer to the *PFI 0 SMB Connector* section of Chapter 1, *Getting Started with the cDAQ Controller*, for more information.

SATA Configuration Submenu

Use this submenu to apply alternate settings to the hard disk drive (HDD) interfaces. Normally, you do not need to modify these settings because the factory default settings provide the most compatible and optimal configuration possible.

- **SATA Controller(s)**—This setting specifies whether the onboard SATA controller is enabled or disabled. The default value is **Enabled**.
- **Onboard Storage**—This item displays the onboard drive detected in the system.

USB Configuration Submenu

Use this submenu to apply alternate configurations to the USB host ports. Normally, you do not need to modify these settings because the factory default settings provide the most compatible and optimal configuration possible.

- **USB Devices**—This item lists the total number of devices detected in the system, categorized by device type.
- **Legacy USB Support**—This setting specifies whether legacy USB support is enabled. Legacy USB support refers to the ability to use a USB keyboard and mouse during system boot or in a legacy operating system such as DOS. The default value on

cDAQ-9132/9133/9134/9135/9136/9137 for LabVIEW Real-Time controllers is **Disabled**. The default value on cDAQ-9132/9133/9134/9135/9136/9137 for Windows controllers is **Enabled**.

- **Overcurrent Reporting**—This setting allows the BIOS to notify the operating system about any USB ports that source too much current. The default value is **Disabled**. Hardware overcurrent protection is always active and cannot be disabled.
- **Transfer Timeout**—This setting specifies the timeout value for Control, Bulk, and Interrupt USB transfers. The default value is **20** seconds.
- **Device Reset Timeout**—This setting specifies the number of seconds the POST waits for a USB mass storage device to start. The default value is **20** seconds.
- **Device Power-Up Delay**—This setting specifies the maximum time a device takes before enumerating. Valid options are **Auto** and **Manual**. The default value is **Auto**. When set to **Auto**, a root port is granted 100 ms, and a hub port's delay value is assigned from the hub descriptor.
- **Device Power-Up Delay in Seconds**—This setting specifies the number of seconds the POST waits for a USB device or hub to power on. This setting is only visible when the Device Power-Up Delay is set to **Manual**. The default value is **5** seconds.
- **Emulation Type**—This setting, available for each detected device if a USB mass storage device is present, specifies how the BIOS presents the USB mass storage device to the system. This option can be used to present a USB mass storage device as a floppy, Zip, hard disk, or CD-ROM drive. The default value is **Auto**, which allows the BIOS to treat small USB flash disk drives as floppy drives and larger USB flash disk drives as hard disk drives.

Security Menu

Use this menu to enable BIOS security options.

- **Administrator Password**—This setting specifies a password that must be entered to access the BIOS setup utility. If only the Administrator's password is set, then this only limits access to the BIOS setup utility and is only asked for when entering the BIOS setup utility. By default, no password is specified.
- **User Password**—This setting specifies a password that must be entered to access the BIOS setup utility or to boot the system. If only the User's password is set, then this is a power-on password and must be entered to boot or enter the BIOS setup utility. In the BIOS setup utility, the User has Administrator rights. By default, no password is specified.

Boot Setup Menu

Use this menu to configure settings related to the boot process and boot device priority.

- *Boot Settings Configuration Submenu*—Use this setting to access the Boot Settings Configuration submenu.
- **PXE Network Boot**—This setting specifies whether the PXE network boot agent is enabled. When enabled, the Intel Boot Agent will be displayed in the Boot Option Priorities menu, allowing you to boot from a PXE server on the local subnet. Note that the Intel Boot Agent device names are preceded by **IBA GE Slot 0200 v1553** in the Boot Option Priorities menu. The system must be restarted for this setting to take effect. The default value is **Disabled**.
- **Boot Option Priorities**—These settings specify the order in which the BIOS checks for bootable devices, including the local hard disk drive, removable devices such as USB flash disk drives or USB CD-ROM drives, or the PXE network boot agent. The BIOS will first attempt to boot from the device associated with **1st Boot Device**, followed by **2nd Boot Device**, and **3rd Boot Device**. If multiple boot devices are not present, the BIOS setup utility will not display all of these configuration options. To select a boot device, press <Enter> on the desired configuration option and select a boot device from the resulting menu. You can also disable certain boot devices by selecting **Disabled**.

Note Only one device of a given type is shown in this list. If more than one device of the same type exists, use the appropriate device BBS priorities submenu to re-order the priority of devices of the same type.

The following submenus are displayed if one or more bootable devices of the corresponding type is present:

- *Hard Drive BBS Priorities Submenu*—Use this setting to access the Hard Drive BBS Priorities submenu to re-order or disable bootable hard drive devices.
- *CD/DVD ROM Drive BBS Priorities Submenu*—Use this setting to access the CD/DVD ROM Drive BBS Priorities submenu to re-order or disable bootable CD/DVD ROM drive devices.
- *Floppy Drive BBS Priorities Submenu*—Use this setting to access the Floppy Drive BBS Priorities submenu to re-order or disable bootable floppy drive devices.
- *Network Device BBS Priorities Submenu*—Use this setting to access the Network Device BBS Priorities submenu to re-order or disable bootable network devices.

Boot Settings Configuration Submenu

Use this submenu to apply alternate configurations to boot settings. Normally, you do not need to modify these settings, as the factory default settings provide the most compatible and optimal configuration.

- **Setup Prompt Timeout**—This setting specifies the number of seconds the system will wait for a BIOS Setup menu keypress (the <Delete> key). The default value is **2** seconds.
- **Bootup NumLock State**—This setting specifies the power-on state of the keyboard NumLock setting. The default value is **On**.

Hard Drive BBS Priorities Submenu

Boot Option #1, Boot Option #2, Boot Option #3—These settings specify the boot priority of hard drive devices. The highest priority device is displayed on the main Boot Option Priorities list. Optionally, each device can also be disabled if the device should never be used as a boot device.

CD/DVD ROM Drive BBS Priorities Submenu

Boot Option #1, Boot Option #2, Boot Option #3—These settings specify the boot priority of CD/DVD ROM drive devices. The highest priority device is displayed on the main Boot Option Priorities list. Optionally, each device can also be disabled if the device should never be used as a boot device.

Floppy Drive BBS Priorities Submenu

Boot Option #1, Boot Option #2, Boot Option #3—These settings specify the boot priority of floppy drive devices. The highest priority device is displayed on the main Boot Option Priorities list. Optionally, each device can also be disabled if the device should never be used as a boot device.

Network Device BBS Priorities Submenu

Boot Option #1, Boot Option #2, Boot Option #3—These settings specify the boot priority of network devices. The highest priority device is displayed on the main Boot Option Priorities list. Optionally, each device can also be disabled if the device should never be used as a boot device.

Save & Exit Menu

The Save & Exit setup menu includes all available options for exiting, saving, and loading the BIOS default configuration. As an alternative to this screen, press <F9> to load optimal BIOS default settings and <F10> to save changes and exit setup.

The Exit setup menu includes the following settings:

- **Save Changes and Reset**—Any changes made to BIOS settings are stored in NVRAM. The setup utility then exits and reboots the controller. The <F10> key can also be used to select this option.
- **Discard Changes and Reset**—Any changes made to BIOS settings during this session of the BIOS setup utility are discarded. The setup utility then exits and reboots the controller. The <Esc> key can also be used to select this option.
- **Save Changes**—Changes made to BIOS settings during this session are committed to NVRAM. The setup utility remains active, allowing further changes.
- **Discard Changes**—Any changes made to BIOS settings during this session of the BIOS setup utility are discarded. The BIOS setup continues to be active.
- **Restore Factory Defaults**—This option restores all BIOS settings to the factory default. This option is useful if the controller exhibits unpredictable behavior due to an incorrect or inappropriate BIOS setting. Notice that any nondefault settings such as boot order, passwords, and so on, are also restored to their factory defaults. The <F9> key can also be used to select this option.
- **Save As User Defaults**—This option saves a copy of the current BIOS settings as the User Defaults. This option is useful for preserving custom BIOS setup configurations.
- **Restore User Defaults**—This option restores all BIOS settings to the user defaults. This option is useful for restoring previously preserved custom BIOS setup configurations.
- **Boot Override**—This option lists all possible bootable devices and allows the user to override the Boot Option Priorities list for the current boot. If no changes have been made to the BIOS setup options, the system will continue booting to the selected device without first rebooting. If BIOS setup options have been changed and saved, a reboot will be required and the boot override selection will not be valid.

Resetting the BIOS Settings

Refer to the *Resetting the System CMOS and BIOS Settings* section of Chapter 1, *Getting Started with the cDAQ Controller*, for steps to take to reset the CMOS and BIOS settings to factory default values.

B

Where to Go from Here

This appendix lists where you can find example programs for the cDAQ controller and C Series modules and relevant documentation.

Example Programs

NI-DAQmx software includes example programs to help you get started programming with the cDAQ controller and C Series modules. Modify example code and save it in an application, or use examples to develop a new application, or add example code to an existing application.

To locate NI software examples, go to `ni.com/info` and enter the Info Code `daqmxexp`. For additional examples, refer to `ni.com/examples`.

To run examples without the device installed, use an NI-DAQmx simulated device. For more information, in Measurement & Automation Explorer (MAX), select **Help»Help Topics» NI-DAQmx»MAX Help for NI-DAQmx** and search for simulated devices.

Related Documentation

Each application software package and driver includes information about writing applications for taking measurements and controlling measurement devices. The following references to documents assume you have NI-DAQmx, LabVIEW 2012 or later, and where applicable, LabVIEW Real-Time 2012 and version 8.6.1 or later of other NI application software.

cDAQ Controller Documentation

The *NI cDAQ-9132/9134/9136 for Windows Quick Start* or *NI cDAQ-9133/9135/9137 for Windows Quick Start*, packaged with your cDAQ controller preloaded with Windows Embedded Standard 7 software, describes how to set up and install the cDAQ controller and C Series modules, and how to confirm that your device is operating properly.

The *NI cDAQ-9132/9134/9136 for LabVIEW Real-Time Quick Start* or *NI cDAQ-9133/9135/9137 for LabVIEW Real-Time Quick Start*, packaged with your cDAQ controller with its hard drive formatted for LabVIEW Real-Time, describes how to set up your host computer, install your NI-DAQmx for Windows software, install the cDAQ controller and C Series modules, and how to confirm that your device is operating properly.

The *NI cDAQ-9132 Specifications, NI cDAQ-9133 Specifications, NI cDAQ-9134 Specifications*, *NI cDAQ-9135 Specifications*, *NI cDAQ-9136 Specifications*, or *NI cDAQ-9137 Specifications* list all specifications for your cDAQ controller. Go to `ni.com/manuals` and search for your cDAQ controller.

The *NI cDAQ Chassis Calibration Procedure* contains information for calibrating all National Instruments CompactDAQ chassis and controllers. Go to `ni.com/manuals` and search for your cDAQ controller.

C Series Module Documentation and Specifications

For module specifications, refer to the documentation included with your C Series module or go to `ni.com/manuals`.

NI-DAQmx

The *NI-DAQmx Readme* lists which devices, ADEs, and NI application software are supported by this version of NI-DAQmx. Select **Start»All Programs»National Instruments» NI-DAQmx»NI-DAQ Readme**.

The *NI-DAQmx Help* contains API overviews, general information about measurement concepts, key NI-DAQmx concepts, and common applications that are applicable to all programming environments. Select **Start»All Programs»National Instruments»Start» All Programs»National Instruments»NI-DAQmx»NI-DAQmx Help**.

LabVIEW

Refer to `ni.com/gettingstarted` for more information about getting started with LabVIEW.

Use the *LabVIEW Help*, available by selecting **Help»LabVIEW Help** in LabVIEW, to access information about LabVIEW programming concepts, step-by-step instructions for using LabVIEW, and reference information about LabVIEW VIs, functions, palettes, menus, and tools. Refer to the following locations on the **Contents** tab of the *LabVIEW Help* for information about NI-DAQmx:

- **VI and Function Reference»Measurement I/O VIs and Functions»DAQmx - Data Acquisition VIs and Functions**—Describes the LabVIEW NI-DAQmx VIs and functions.
- **Property and Method Reference»NI-DAQmx Properties**—Contains the property reference.
- **Taking Measurements**—Contains the conceptual and how-to information you need to acquire and analyze measurement data in LabVIEW, including common measurements, measurement fundamentals, NI-DAQmx key concepts, and device considerations.

LabVIEW Real-Time

Refer to ni.com/gettingstarted for more information about getting started with LabVIEW Real-Time.

The *Real-Time Module Concepts* book of the *LabVIEW Real-Time Module Help* includes conceptual information about real-time programming techniques, application architectures, and Real-Time Module features you can use to create real-time applications. Refer to the Real-Time Module concepts before attempting to create a deterministic real-time application.

.NET Languages without NI Application Software

With the Microsoft .NET Framework, you can use NI-DAQmx to create applications using Visual C# and Visual Basic .NET without Measurement Studio. Refer to the *NI-DAQmx Readme* for specific versions supported.

Training Courses

If you need more help getting started developing an application with NI products, NI offers training courses. To enroll in a course or obtain a detailed course outline, refer to ni.com/training.

Technical Support on the Web

For additional support, refer to ni.com/support.

Many DAQ specifications and user guides/manuals are available as PDFs. You must have Adobe Reader 7.0 or later (PDF 1.6 or later) installed to view the PDFs. Refer to the Adobe Systems Incorporated Web site at www.adobe.com to download Adobe Reader. Refer to the National Instruments Product Manuals Library at ni.com/manuals for updated documentation resources.

C

Technical Support and Professional Services

Log in to your National Instruments ni.com User Profile to get personalized access to your services. Visit the following sections of 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 self-paced online training modules at ni.com/self-paced-training. All customers automatically receive a one-year membership in the Standard Service Program (SSP) with the purchase of most software products and bundles including NI Developer Suite. NI also offers flexible extended contract options that guarantee your SSP benefits are available without interruption for as long as you need them. Visit ni.com/ssp for more information.

    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 training and certification program information. You can also 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.

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.

Index

Symbols

Numerics

A

B

C

D

E

F

G

H

P

Q

R

S

T

U

V

W

X