
cDAQ-9170/ 9173/9177 User Manual

2025-02-20



Contents

| | |
|--|----|
| Welcome to the User Manual | 6 |
| Overview | 7 |
| Driver Support | 10 |
| Components of a cDAQ-9170/9173/9177 System | 11 |
| Part Numbers for Recommended Cables and Accessories | 13 |
| Theory of Operation | 16 |
| Examples | 20 |
| Front Panel | 22 |
| LED Indicators | 22 |
| USB-C Port | 23 |
| Power Connector | 24 |
| PFI (cDAQ-9177) | 24 |
| Chassis Grounding Screw | 25 |
| Kit Contents | 26 |
| Safety Guidelines | 27 |
| Electromagnetic Compatibility Guidelines | 28 |
| Setting up the cDAQ-9170/9173/9177 | 29 |
| Unpacking | 29 |
| What You Need to Get Started | 29 |
| Installing the cDAQ Chassis | 31 |
| Wiring External Power to the cDAQ Chassis | 34 |
| Verifying Installation in HWCU/MAX | 35 |
| Verifying the Installation in Hardware Configuration Utility | 35 |
| Verifying the Installation with MAX | 36 |
| Removing Modules from the cDAQ-9170/9173/9177 | 41 |
| Mounting the cDAQ-9170/9173/9177 | 42 |
| Mounting Requirements | 43 |
| Mounting the cDAQ Chassis Directly on a Flat Surface | 44 |
| Mounting the cDAQ Chassis on a Panel with a Panel Mounting Kit | 45 |
| Alternate Mounting Configurations | 46 |
| Mounting the cDAQ Chassis on a DIN Rail | 47 |
| Mounting the Chassis on a Rack | 48 |

| | |
|--|----|
| Mounting the cDAQ Chassis on a Desktop | 48 |
| Dimensions | 49 |
| Performing Analog Input Measurements..... | 51 |
| Analog Input Triggering Signals | 51 |
| Analog Input Timing Signals | 52 |
| AI Sample Clock Signal | 52 |
| AI Sample Clock Timebase Signal | 53 |
| AI Convert Clock Signal Behavior For Analog Input Modules..... | 53 |
| AI Start Trigger Signal | 56 |
| AI Reference Trigger Signal | 57 |
| AI Pause Trigger Signal | 59 |
| Analog Input Sync Pulse..... | 59 |
| Getting Started with AI Applications in Software | 60 |
| Generating Analog Output | 61 |
| Analog Output Data Generation Methods | 61 |
| Software-Timed Generations | 61 |
| Hardware-Timed Generations..... | 62 |
| Analog Output Triggering Signals..... | 63 |
| Analog Output Timing Signals | 64 |
| AO Sample Clock Signal | 64 |
| AO Sample Clock Timebase Signal..... | 65 |
| AO Start Trigger Signal | 65 |
| AO Pause Trigger Signal | 67 |
| Analog Output Sync Pulse | 68 |
| Minimizing Glitches on the Output Signal | 68 |
| Getting Started with AO Applications in Software | 69 |
| Using the Digital Input/Output | 70 |
| Serial DIO versus Parallel DIO Modules..... | 70 |
| Static DIO | 71 |
| Digital Input..... | 71 |
| Digital Input Triggering Signals..... | 71 |
| Digital Input Timing Signals..... | 72 |
| DI Sample Clock Signal..... | 72 |
| DI Sample Clock Timebase Signal | 73 |
| DI Start Trigger Signal | 74 |
| DI Reference Trigger Signal | 75 |

| | |
|--|-----|
| DI Pause Trigger Signal | 76 |
| Digital Input Filters | 77 |
| Getting Started with DI Applications in Software | 78 |
| Change Detection Event | 78 |
| Digital Output | 79 |
| Digital Output Data Generation Methods | 80 |
| Digital Output Triggering Signals | 81 |
| Digital Output Timing Signals | 82 |
| DO Sample Clock Signal | 82 |
| DO Sample Clock Timebase Signal | 83 |
| DO Start Trigger Signal | 83 |
| DO Pause Trigger Signal | 84 |
| Getting Started with DO Applications in Software | 85 |
| Digital Input/Output Configuration for NI-9401 | 86 |
| Using PFI | 87 |
| Using Counters | 89 |
| Counter Timing Engine | 89 |
| Counter Triggering | 91 |
| Counter Input Applications | 92 |
| Counting Edges | 92 |
| Pulse-Width Measurement | 94 |
| Pulse Measurement | 96 |
| Semi-Period Measurement | 98 |
| Frequency Measurement | 100 |
| Period Measurement | 108 |
| Position Measurement | 109 |
| Channel Z Behavior | 110 |
| Two-Signal Edge-Separation Measurement | 112 |
| Counter Output Applications | 114 |
| Simple Pulse Generation | 114 |
| Pulse Train Generation | 116 |
| Frequency Generation | 121 |
| Frequency Division | 123 |
| Pulse Generation for ETS | 123 |
| Counter Timing Signals | 124 |
| Counter n Source Signal | 124 |

| | |
|--|-----|
| Counter n Gate Signal | 126 |
| Counter n Aux Signal | 126 |
| Counter n A, Counter n B, and Counter n Z Signals | 127 |
| Counter n Up_Down Signal | 128 |
| Counter n HW Arm Signal | 128 |
| Counter n Sample Clock Signal | 128 |
| Counter n Internal Output and Counter n TC Signals | 130 |
| Frequency Output Signal | 130 |
| Default Counter/Timer Routing | 130 |
| Cascading Counters | 131 |
| Prescaling | 131 |
| Synchronization Modes | 132 |
| Digital Routing | 133 |
| Clock Routing | 134 |
| Maintenance | 135 |

cDAQ-9170/9173/9177 User Manual

The cDAQ-9170/9173/9177 User Manual provides detailed descriptions of the product functionality and the step by step processes for use.

Looking for Something Else?

For information not found in the User Manual for your product, such as specifications and API reference, browse ***Related Information***.

Related information:

- [cDAQ-9183 Specifications](#)
- [Calibration Procedure](#)
- [NI-DAQmx User Manual](#)
- [NI-DAQmx LabVIEW API Reference](#)
- [NI-DAQmx Properties](#)
- [NI-DAQmx .NET Class Library Help](#)
- [Software and Driver Downloads](#)
- [NI-DAQmx Release Notes](#)
- [License Setup and Activation](#)
- [Dimensional Drawings](#)
- [Product Certifications](#)
- [Letter of Volatility](#)
- [Discussion Forums](#)
- [NI Learning Center](#)

cDAQ-9170/9173/9177 Overview

The one-slot NI cDAQ-9170, four-slot NI cDAQ-9173, and eight-slot NI cDAQ-9177 USB chassis are designed for use with C Series modules. The cDAQ chassis are capable of measuring a broad range of analog and digital I/O signals using a USB-C interface. For module specifications, refer to the documentation included with your C Series module (s) or go to ni.com/docs.

cDAQ-9170/9173/9177 Key Features

The cDAQ-9170/9173/9177 are CompactDAQ USB chassis that has the following features and capabilities.

- One, four, and eight-slot design that can accommodate up to one, four, and eight C Series I/O modules respectively
- Internal network interface with one USB port
- 32-bit general-purpose counters/timers, accessible through a hardware-timed digital C Series module, for applications involving quadrature encoders, PWM, and pulse train generation
- USB connectivity for easy integration into existing network infrastructures

System Health Monitoring

NI recommends the use of system health monitoring to ensure potential problems are detected and resolved as soon as possible. The cDAQ-9170/9173/9177 has hardware and software features your application can use to monitor system health and respond to events such as hardware faults, software failure, system crash, or any loss of communication. For more information on how to configure your system to monitor system health, visit ni.com/info and enter `systemhealth`.

Time-Based Features

The cDAQ-9170/9173/9177 chassis features automatic network-based synchronization with compatible networks and IEEE 802.1AS-capable NI Linux Real-Time controllers. The SYNC logo on the chassis front panel indicate that the chassis are capable of

hardware-based synchronization over a network.

The chassis supports IEEE 802.1AS synchronization, and all chassis timebases will be automatically synchronized. Refer to **Networking** and **Synchronization across a Network** for more information about supported topologies and other technical requirements.

Network-synchronized chassis can also take advantage of time-based synchronization features in NI-DAQmx. Certain triggers and timestamps can be specified in terms of time of day. Time-based triggers and timestamps and multichassis tasks (spanning multiple network-synchronized cDAQ chassis) can help simplify programming for large systems.

Time triggers and timestamps can be specified in Host Time or I/O Device Time, depending on the needs of your application.

- I/O Device Time—The time the cDAQ-9170/9173/9177 uses internally. This time is determined by the network configuration and is shared by all 802.1AS network-synchronized devices on your subnet.
- Host Time—The time on your Windows computer or NI Linux Real-Time controller. This is usually the current global time, and is provided by a local real-time clock or a network time protocol (NTP) server.

NI-DAQmx automatically translates from Host Time to I/O Device Time as necessary. The accuracy of this translation depends on the relationship between these times and can reduce the relative accuracy of time triggers and timestamps across multiple devices. For maximum accuracy, use an NI Linux Real-Time controller as the host in a supported topology. However, NI-DAQmx guarantees that two tasks configured to start at the same host time always start at the same I/O Device Time in all scenarios, preserving precise synchronization between chassis in this common use case. Refer to the **Timing and Triggering** topic in the **NI-DAQmx User Manual** for more information on accessing time-based features in the NI-DAQmx API.

Related reference:

- [cDAQ-9170/9173/9177 Front Panel](#)

Firmware

Firmware can be updated through the Hardware Configuration Utility, NI MAX, or the web interface to the chassis. For cDAQ firmware information and updates, visit ni.com/info and enter the Info Code `cdaqfw`.

cDAQ-9170/9173/9177 Driver Support

Determine the earliest driver version supported for your product.



Tip To optimize product performance, update to the most recent driver version.

Table 1. Earliest Driver Version Support

| Driver Name | Earliest Version Support |
|-------------|---|
| NI-DAQmx | cDAQ-9173/77 2025 Q1, cDAQ-9170 2025 Q2 |

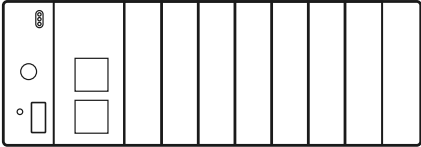

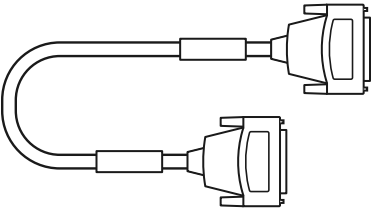
Components of a cDAQ-9170/9173/9177 System

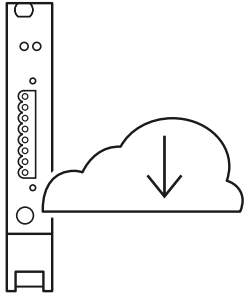

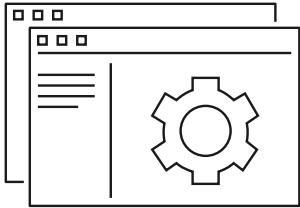
cDAQ-9170/9173/9177 is designed for use in a system that might require hardware, drivers, and software to optimize cDAQ-9170/9173/9177 for your application. Use the minimum required cDAQ-9170/9173/9177 system components as a starting point for building your system.



Note A system and the surrounding environment must meet any requirements defined in the relevant specifications documents.

Table 2. Minimum Required cDAQ-9170/9173/9177 System Components

| Component | Description and Recommendations |
|---|--|
| cDAQ Chassis  | A CompactDAQ chassis houses the cDAQ modules and supplies power for cDAQ module functions. |
| C Series Module  | Your C Series instrument. |
| Cables and Accessories  | Cables and accessories allow connectivity to/from your instrument for measurements. Refer to Part Numbers for Recommended Cables and Accessories for recommended cables and accessories and guidance. |

| Component | Description and Recommendations |
|---|---|
| <p>NI-DAQmx Driver</p>  | <p>Instrument driver software that provides functions to interact with and execute measurements using the cDAQ-9170/9173/9177 and C Series modules.</p> <div data-bbox="699 415 1468 678">  <p>Note For optimal performance, use the most current version of NI-DAQmx with the cDAQ-9170/9173/9177. You can find the NI-DAQmx driver requirements in the <i>NI-DAQmx Release Notes</i>.</p> </div> |
| <p>NI Applications</p>  | <p>NI-DCPower offers driver support for the following applications:</p> <ul style="list-style-type: none"> • LabVIEW • FlexLogger • LabWindows/CVI • C/C++ • .NET • Python |

Related reference:

- [Part Numbers for Recommended Cables and Accessories](#)

Part Numbers for Recommended Cables and Accessories

Use part numbers to purchase the cables and the accessories for optimizing the performance of cDAQ-9170/9173/9177.

Table 3. Part numbers for Recommended Power Connector Accessories

| Accessory | Description | Part number |
|--------------------------|---|-------------|
| Power Screw Terminal Kit | 2-position single supply power screw terminal (includes four screw terminal connectors for supported CompactRIO/CompactDAQ Controllers) | 780702-01 |
| NI 9971 | Quantity of 4, 2-Position C Series Module Backshell | 196375-01 |

Table 4. Part Numbers for Recommended Mounting Accessories

| Accessory | Description | Part number |
|--|--|-------------|
| NI 9904 | 4-Slot Chassis Panel Mount | 779097-01 |
| NI 9901 | Desktop Mounting Kit | 779473-01 |
| NI 9910 | Sliding Rack Mount Kit | 779102-01 |
| cRIO-908x Rack Mounting Kit | Rack Mount Kit for CompactRIO/CompactDAQ | 781989-01 |
| NI 9912 | 4-Slot CompactDAQ Legacy Chassis DIN Mount | 779019-01 |
| Rack Mount Kit for CompactRIO and CompactDAQ | Industrial Rack Mount Kit 19 in, CompactRIO/CompactDAQ Chassis DIN Mount | 786411-01 |

Table 5. Part Numbers for Recommended USB Cables

| Accessory | Description | Part number |
|---------------------------------|-------------|-------------|
| Locking USB cable, (1 m length) | | 769493-01 |

| Accessory | Description | Part number |
|---------------------------------|-------------|-------------|
| Locking USB cable, (2 m length) | | 758787-01 |

Table 6. Part Numbers for Recommended Power Supplies

| Accessory | Description | Part number |
|-----------|--|-------------|
| PS-10 | 24 VDC, 5 A, 120 W Desktop Power Supply | 782698-01 |
| PS-9 | 12 VDC, 1.25 A, 100-240 VAC Input Desktop Power Supply | 780703-01 |
| PS-14 | 24 VDC, 3.3 A, 80 W, DIN-Mountable Industrial Power Supply | 783167-01 |
| PS-15 | 24 VDC to 28 VDC, 5 A, 120 W, DIN-Mountable Industrial Power Supply | 789093-01 |
| PS-16 | 24 VDC to 28 VDC, 10 A, 240 W, DIN-Mountable Industrial Power Supply | 781094-01 |
| PS-17 | 24 VDC, 20 A, 480 W, DIN-Mountable Industrial Power Supply | 781095-01 |

Table 7. Part Numbers for Recommended Accessories

| Accessory | Description | Part number |
|---------------------------------|--|-------------|
| Module Immobilization Accessory | Module Immobilization Accessory for 1/4/8-Slot cRIO-905x/906x/907x, NI-914x, and cDAQ-917x | 158534-02 |
| C Series Carrying Handle | Carrying handle for CompactDAQ and Compact RIO | 786744-01 |



Note Refer to ***CompactDAQ System Accessory Compatibility Guide*** on ni.com for more information about supported cables and accessories for your instrument.

Related tasks:

- [Mounting the cDAQ Chassis on a Panel with a Panel Mounting Kit](#)

Related information:

- [CompactDAQ System Accessory Compatibility Guide](#)

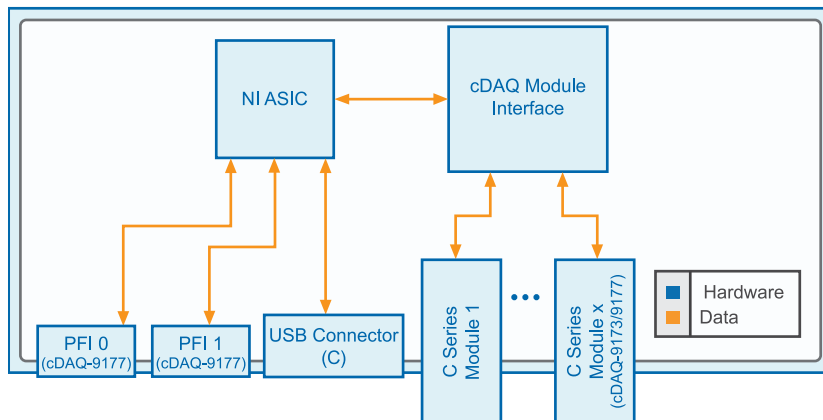
cDAQ-9170/9173/9177 Theory of Operation

The cDAQ-9170/9173/9177 USB-C chassis are designed for use with C Series modules. The cDAQ chassis are capable of measuring a broad range of analog and digital I/O signals and sensors.

The cDAQ system consists of three parts—C Series module(s), the cDAQ module interface, and the NI ASIC—as shown in Figure 2. These components digitize signals, perform D/A conversions to generate analog output signals, measure and control digital I/O signals, and provide signal conditioning.

cDAQ-9170/9173/9177 Block Diagram

Figure 2. cDAQ-9170/9173/9177 Block Diagram



cDAQ Chassis Components Overview

The cDAQ system consists of three parts: C Series module(s), the cDAQ module interface, and the NI ASIC used by the cDAQ-9170/9173/9177.

These components digitize signals, perform D/A conversions to generate analog output signals, measure and control digital I/O signals, and provide signal conditioning.

C Series Module

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

C Series modules are hot-swappable and automatically detected by the cDAQ chassis. 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. In most cases, the C Series modules provide channel-to-ground and channel-to-channel isolation.

For more information about which C Series modules are compatible with the cDAQ chassis, go to ni.com/r/cdaqenet.

Parallel versus Serial DIO Modules

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

- Serial digital C Series modules are designed for signals that change slowly and are accessed by software-timed reads and writes.
- Parallel digital C Series 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 C Series modules, refer to ***Using the Digital Input/Output***.

cDAQ Module Interface

The cDAQ module interface manages data transfers between the NI ASIC and the C Series modules. The interface also handles autodetection, signal routing, and

synchronization.

NI ASIC

The NI ASIC 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 NI ASIC 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 related topic links below for more information about the configuration of these signals.
 - ***Analog Input Timing Signals***
 - ***Analog Output Timing Signals***
 - ***Digital Input Timing Signals***
 - ***Digital Output Timing Signals***
- **Triggering Modes**—The cDAQ chassis supports different trigger modes, such as start trigger, reference trigger, and pause trigger with analog, digital, or software sources. Refer to the related topic links below for more information about the configuration of these signals.
 - ***Analog Input Triggering Signals***
 - ***Analog Output Triggering Signals***
 - ***Digital Input Triggering Signals***
 - ***Digital Output Triggering Signals***
- **Independent Data Streams**—The cDAQ-9170 chassis supports six 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.

The cDAQ-9173/9177 chassis 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 chassis slots and through the two PFI terminals provided on the cDAQ-9177 chassis. Refer to ***Using the Digital Input/Output*** for more information.

- Flexible Counter/Timers—The cDAQ chassis 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 I/O modules installed in up to two slots. Refer to the ***Using Counters*** section for more information.

Related concepts:

- [Analog Input Timing Signals](#)
- [Analog Output Timing Signals](#)
- [Digital Input Timing Signals](#)
- [Digital Output Timing Signals](#)
- [Analog Input Triggering Signals](#)
- [Analog Output Triggering Signals](#)
- [Digital Input Triggering Signals](#)
- [Digital Output Triggering Signals](#)
- [Using Counters](#)

cDAQ-9170/9173/9177 Examples

NI installs example code with your software or driver that demonstrates the functionality of cDAQ-9170/9173/9177. Use these examples to learn about the product or accelerate your own application development.

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.

Table 8. Common cDAQ-9170/9173/9177 Example code

| Example name | Description |
|--|---|
| <i>NI-DAQmx - Voltage - Continuous Input</i> | This example demonstrates how to continuously acquire voltage measurement using a DAQmx device. |
| <i>NI-DAQmx - Voltage - Continuous Output</i> | This example demonstrates how to continuously re-generate analog output data from a buffer in your computer's memory. |
| <i>NI-DAQmx - Current - Continuous Input</i> | This example demonstrates how to continuously acquire buffered current measurements using a DAQmx device. |
| <i>NI-DAQmx - Current - Continuous Output</i> | This example demonstrates how to continuously re-generate analog output data from a buffer in your computer's memory. |
| <i>NI-DAQmx Count Edges</i> | This example demonstrates how to continuously read back the number of digital edges that have been counted by a counter input. |
| <i>NI-DAQmx Digital - Change Detection</i> | This example demonstrates how to acquire a continuous amount of digital data based off of the changes of an external signal across one or many lines on either the rising or falling edges. |
| <i>NI-DAQmx Digital - SW-Timed Output</i> | This example demonstrates how to generate digital output data using software timing. |



Note The described example codes cannot be used without the appropriate modules being inserted into the chassis.

cDAQ-9170/9173/9177 Example Locations

Find the examples installed with cDAQ-9170/9173/9177.

NI-DAQmx examples are available from the software download and from ni.com/support. To run examples without the device installed, use an NI-DAQmx simulated device.

NI-DAQmx installs examples for cDAQ-9170/9173/9177 in the following software:

- LabVIEW or LabWindows/CVI
- SignalExpress
- ANSI C
- .NET

Examples installed in LabVIEW and LabWindows/CVI are available from the Windows Start menu or from the **NI Example Finder**. The NI Example Finder is a utility that organizes examples into categories. The NI Example Finder allows you to browse and to search installed examples.

Examples installed in LabVIEW NXG are available from the Learning tab in LabVIEW NXG.

Depending on the software the examples install for, you have options for accessing the installed examples.

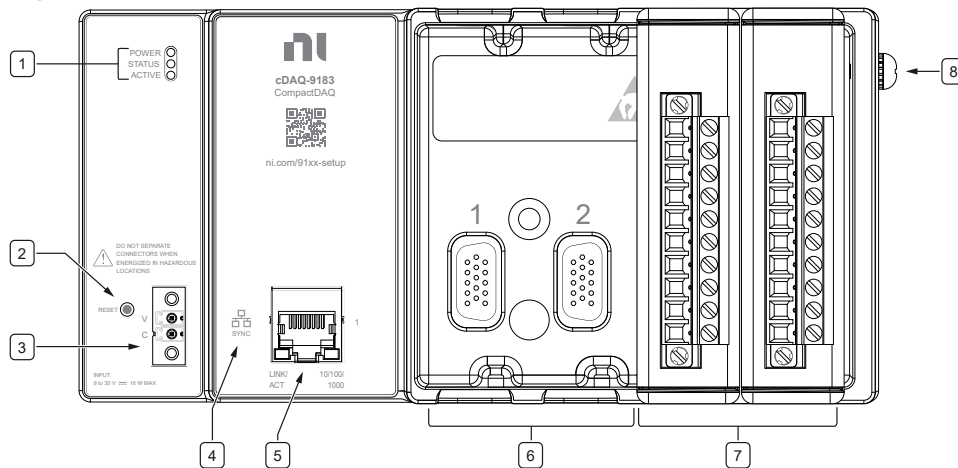
Table 9. Access options for NI-DAQmx Examples in LabVIEW and LabWindows CVI

| Location | Access Option #1 | Access Option #2 |
|--------------------|---|--|
| NI Example Finder | Choose Browse , then Task » Hardware Input and Output » DAQmx | Choose Browse , then Directory Structure » DAQmx |
| Windows Start Menu | Start » National Instruments » NI-DAQmx Examples | — |

cDAQ-9170/9173/9177 Front Panel

Refer to the front panel diagram to understand the connectors, LEDs, and other features of the cDAQ-9170/9173/9177.

Figure 3. cDAQ-9170/9173/9177 Front Panel



1. POWER, STATUS, and ACTIVE LEDs
2. Reset Button
3. Power Connector
4. SYNC Logo
5. Ethernet Port, 10/100/1000 and LINK/ACT LEDs
6. Module Slots
7. Installed C Series Modules
8. Chassis Grounding Screw

cDAQ-9170/9173/9177 LED Indicators

The Power, Status, and Active LEDs, located on the chassis front panel, indicates chassis power, status, and operation activity.

cDAQ-9170/9173/9177 Power LED

The Power LED indicates chassis power.

Table 10. Power LED State/Chassis Status

| Color | LED State | Chassis Status |
|-------|-----------|----------------|
| Green | On | Power on |
| Green | Off | Power off |

cDAQ-9170/9173/9177 Active LED

The Active LED indicates chassis operating activity.

Table 11. Active LED State/Chassis Status

| Color | LED State | Chassis Status |
|-------|-----------|--------------------------------|
| Green | On | USB traffic present |
| Amber | On | Unconfigured |
| — | Off | No USB traffic present/suspend |

cDAQ-9170/9173/9177 Ready LED

The Status LED indicates chassis status.

Table 12. Status LED Indicator Status

| Color | LED State | Chassis Status |
|-------|-----------|---|
| Green | On | Full speed (12 Mb/s) |
| Amber | On | Hi-Speed (480 Mb/s) |
| — | Off | USB connection is not established/suspend |

USB-C Port

The cDAQ-9170/9173/9177 chassis have a USB-C port. You can use a strain relieved USB-C cable to network your chassis to a computer host or a NI Linux Real-Time controller.

Related information:

- [cDAQ-9170/9173/9177 Networking](#)

USB Cabling

The cDAQ chassis features a USB-C interface. You can provide strain relief for the USB cable by using the jackscrew on a locking USB cable to securely attach the cable to the chassis.

Power Connector

The cDAQ chassis ships with the following:

- 2-position power screw terminal connector plug (for use with an external power source)
- Power supply



Note When operating the cDAQ-9170/9173/9177 in hazardous locations, you must use the power connector with an external power supply rated for hazardous locations. The power supply included in the cDAQ-9170/9173/9177 kit is intended only for desktop use. For all other applications use the included 2-position power connector plug and a power supply rated for your application power requirements. Visit [ni.com](https://www.ni.com) to find hazardous locations-certified power supplies. Refer to the specifications document for your chassis for information about the power connector.

PFI (cDAQ-9177)

You can configure channels of a parallel digital module as Programmable Function Interface (PFI) terminals. The cDAQ-9177 chassis also provides two terminals for PFI. Up to two digital modules can be used to access PFI terminals in a single chassis.

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



Note Only the 8 slot cDAQ-9177 has PFI slots.

Chassis Grounding Screw



Notice To ensure the specified EMC performance, the cDAQ chassis must be connected to the grounding electrode system of your facility using the chassis ground terminal.

The wire should be 1.31 mm^2 (16 AWG) or larger 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, refer to the ***Grounding for Test and Measurement Devices*** document by going to ni.com/info and entering 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.

Kit Contents

Items in your cDAQ chassis kit:

- cDAQ-9170/9173/9177 chassis
- Documentation
- Power supply
- USB-C Cable
- Screwdriver

Safety Guidelines



Caution Do ***not*** operate the cDAQ-9170/9173/9177 in a manner not specified in this user manual. 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.

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, when the product is connected to a peripheral device or test object, or if the product is used in residential or commercial areas. 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.



Notice To ensure the specified EMC performance, product installation requires either special considerations or user-installed add-on devices.



Notice 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.



Notice To ensure the specified EMC performance, do not connect the power connector 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.

Setting up the cDAQ-9170/9173/9177

Confirm your kit contents, install the cDAQ-9170/9173/9177 chassis, and prepare it for use.

The cDAQ chassis and C Series module(s) are packaged separately.

Unpacking

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



Notice *Never* touch the exposed pins of connectors.

To avoid ESD damage in handling the chassis, 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 chassis from the package.

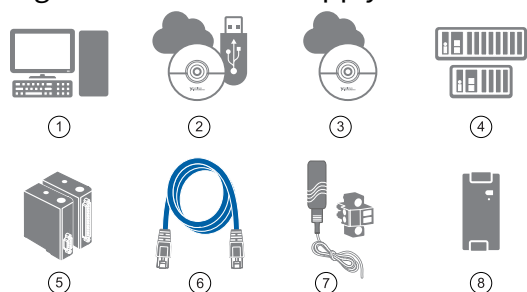
Remove the chassis from the package and inspect it for loose components or any other signs of damage. Notify NI if the device appears damaged in any way. Do **not** install a damaged chassis.

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

What You Need to Get Started

You will need the following items to set up your cDAQ chassis.

Figure 4. Installation Supply List



1. Host Computer Running Windows
2. Application Software, such as LabVIEW, if not already installed
3. NI-DAQmx Driver (packaged with the cDAQ chassis)
4. cDAQ-9170/9173/9177 Chassis
5. C Series Module(s)
6. Strain relieved USB-C cable
7. Power adapter (packaged with the cDAQ chassis)
8. External Power Supply (if using the power connector instead of the included power supply)



Note Check the NI-DAQmx driver and application software release notes for specific version compatibility with your host computer.



Note When operating the cDAQ chassis in hazardous locations, you must use the power connector with an external power supply rated for hazardous locations. The power supply included in the cDAQ chassis kit is intended only for desktop use. For all other applications use the included 2-position power connector plug and a power supply rated for your application power requirements. Visit ni.com to find hazardous locations-certified power supplies.



Note When operating the cDAQ chassis in temperatures below 0 °C, you must use the PS-15 power supply or another power supply rated for below 0 °C.

Installing the cDAQ Chassis

Complete the following steps to install the cDAQ-9170/9173/9177.

The cDAQ chassis and C Series module(s) are packaged separately. Refer to the cDAQ-9170/9173/9177 front panel diagram while completing the following assembly steps.

1. Install the application software (if applicable), as described in the installation instructions that accompany your software.
2. Download the NI-DAQmx driver from ni.com/downloads.

Package Manager downloads with the driver software to handle the installation. Refer to the Package Manager Manual for more information about installing, removing, and upgrading NI software using Package Manager.



Note Refer to **cDAQ-9170/9173/9177 Driver Support** for the earliest NI-DAQmx support version for each cDAQ USB chassis.

3. Follow the instructions in the installation prompts.

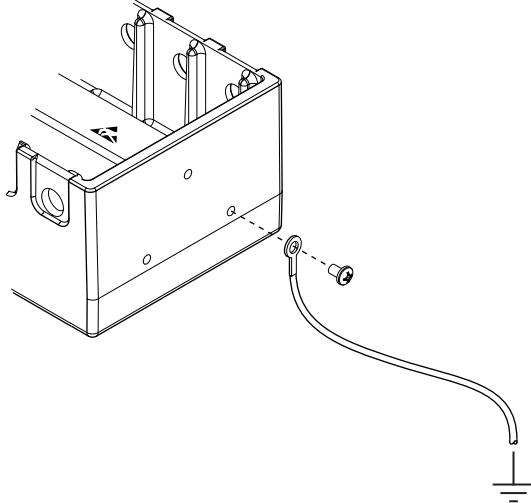
Windows users may see access and security messages during installation. Accept the prompts to complete the installation.

4. When the installer completes, select **Restart** in the dialog box that prompts you to restart, shut down, or restart later.
If you have problems installing your software, go to ni.com/support and search for the driver name.
5. (Optional) Mount the cDAQ chassis to a panel, wall, or DIN rail, or attach the desktop mounting kit, as described in the **Mounting the cDAQ-9170/9173/9177** section.
6. Attach a ring lug to a 1.31 mm^2 (16 AWG) or larger wire. Connect the ring lug to the chassis ground terminal on the cDAQ chassis using the chassis grounding screw as shown in the following figure. Attach the other end of the wire to the grounding electrode system of your facility. Refer to the **Chassis Grounding Screw** section for more information about making this connection.



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 mm² (16 AWG) or larger wire. Use shorter wire for better EMC performance.

Figure 5. Ring Lug Attached to Chassis Ground Terminal



7. Make sure that no signals are connected to the C Series module.
8. Align the C Series module with the cDAQ chassis slot.
9. Squeeze both C Series module latches, insert the module into the module slot, and press until both latches lock the module in place.
10. Wire the C Series module as indicated in the C Series module documentation.



Note Connect I/O cable shields to the chassis grounding screw, shown in the previous figure, unless otherwise specified in the C Series module documentation. Refer to the **Chassis Grounding Screw** section for more information about making this connection.

11. Connect the cDAQ chassis with the USB cable to any available USB port on your computer. If you are using a locking USB cable, use the jackscrew to securely attach the cable to the chassis.
12. Power the chassis using the included power supply or the included power connector with an external 9 V DC to 30 V DC power source. For information about wiring your external power source to the power connector, refer to the **Wiring External Power to the cDAQ Chassis** section. The cDAQ chassis requires an external power supply that meets the specifications listed in the specifications

document for your cDAQ chassis.



Note When operating the cDAQ-9170 in hazardous locations, you must use the power connector with an external power supply rated for hazardous locations. The power supply included in the cDAQ-9170 kit is intended only for desktop use. For all other applications use the included 2-position power connector plug and a power supply rated for your application power requirements. Visit ni.com to find hazardous locations-certified power supplies.

The POWER and STATUS LEDs light. The POWER LED lights as long as power is being supplied to the cDAQ chassis. The STATUS LED turns off after firmware boots. Refer to the **cDAQ-9170/9173/9177 LED Indicators** section for information about the LEDs on the cDAQ chassis.

Related concepts:

- [Mounting the cDAQ-9170/9173/9177](#)
- [Chassis Grounding Screw](#)
- [USB Cabling](#)
- [Troubleshooting Chassis Connectivity in MAX](#)

Related tasks:

- [Wiring External Power to the cDAQ Chassis](#)

Related reference:

- [cDAQ-9170/9173/9177 Front Panel](#)
- [cDAQ-9170/9173/9177 Driver Support](#)
- [cDAQ-9170/9173/9177 LED Indicators](#)

Related information:

- [Package Manager User Manual](#)
- [Connecting to a Real-Time Controller](#)

Wiring External Power to the cDAQ Chassis

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

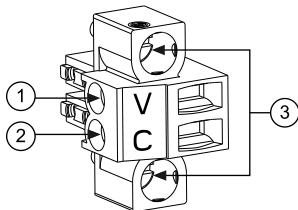
! Notice To ensure the specified EMC performance, do **not** connect the power input to a DC mains supply or to any supply requiring a connecting cable longer than 3 m (10 ft). A DC mains supply is a local DC electricity supply network in the infrastructure of a site or building.

The cDAQ chassis requires an external power source as described in the **Power Requirements** section of the specifications document for your chassis.

Some suggested NI power supplies are listed in **Part Numbers for Recommended Cables and Accessories**. The cDAQ chassis filters and regulates the supplied power and provides power to all of the modules. The green POWER LED on the front panel identifies when the power input is in use.

1. Make sure the power source is turned off.
2. If connected, loosen the connector screw flanges and remove the power screw terminal connector plug from the cDAQ chassis. The following figure 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 6. Power Screw Terminal Connector Plug



1. V (Positive) Terminal Screw
2. C (Negative) Terminal Screw
3. Connector Screw Flanges



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

3. Connect the positive lead of the power source to the V terminal of the power connector plug and tighten the terminal screw.

4. Connect the negative lead of the power source to the C terminal of the power screw terminal connector plug and tighten the terminal screw.
5. Install the power connector plug on the front panel of the cDAQ chassis and tighten the connector screw flanges.
6. Turn on the external power source.

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.

Refer to the **Power Requirements** section of the **cDAQ-9170/9173/9177 Specifications** for information about the power supply input range. Refer to the **Safety Voltages** section of the **cDAQ-9170/9173/9177 Specifications** for information about the maximum voltage from terminal to chassis ground.

Related reference:

- [Part Numbers for Recommended Cables and Accessories](#)

Related information:

- [cDAQ-9183 Specifications](#)

Verifying the Installation

Before using the cDAQ-9170/9173/9177, verify that it is installed correctly through Hardware Configuration Utility or Measurement & Automation Utility (MAX).

Verifying the Installation in Hardware Configuration Utility

NI recommends using Hardware Configuration Utility to perform and to validate initial hardware configuration.

1. Open Hardware Configuration Utility.
The cDAQ-9170/9173/9177 appears in the system pane automatically.
If the cDAQ-9170/9173/9177 is not shown in the system pane, click the + (**Add Hardware**) button, select the chassis from the Discovered devices, and click **Add**.

2. Record the name that Hardware Configuration Utility assigns to the cDAQ-9170/9173/9177 or provide a custom name.
Use this name when programming the cDAQ-9170/9173/9177.
3. Validate that your instrument is installed correctly: select the cDAQ-9170/9173/9177 module in the system pane, expand the **Troubleshooting** area of the configuration pane, and click **Self-test**.
Hardware Configuration Utility reports when it has validated the hardware setup.

What Do I Do If the cDAQ-9170/9173/9177 Does Not Appear in Hardware Configuration Utility?

1. Click the refresh button (↻) to refresh the connection between the hardware and the software.
2. If the cDAQ-9170/9173/9177 is still not listed, complete the following steps.
 - a. Power off the system.
 - b. Ensure that all hardware is correctly installed.
 - c. Restart the system.

Verifying the Installation in MAX

To configure your NI hardware, use Measurement & Automation Explorer (MAX). MAX informs other programs about the NI hardware products in the system and their hardware configuration. MAX is automatically installed with NI-DAQmx.



Note The cDAQ chassis and C Series module(s) are packaged separately.

Refer to the cDAQ-9170/9173/9177 while completing the following assembly steps.

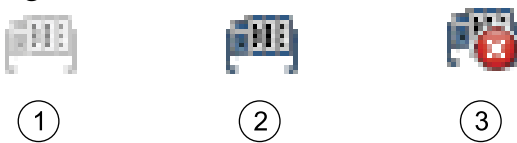
1. To add the chassis to your software configuration, open NI MAX on your Windows host computer. Expand **Devices and Interfaces** » **Network Devices**.
 - If the chassis is on your local subnet, the chassis automatically appears in the list of available devices. Right-click the cDAQ chassis and select **Add Device**.
 - If the chassis is not on your local subnet, right-click **Network Devices** and select **Find Network NI-DAQmx Devices**.
2. In the Find Network NI-DAQmx Devices dialog box that opens, do one of the following:

- Check the box that corresponds to your chassis in the Hostname column.
- If you know the chassis IP address, such as 192.168.0.2, enter it into the **Add Device Manually** field of the Find Network NI-DAQmx Devices window, and click the + button.
- Enter the hostname of the chassis. The default hostname is **cDAQ917x-*<serial number>***, where the **x** represents the last digit of your cDAQ chassis model number.

3. Click **Add Selected Devices**.

The cDAQ chassis icon changes from white to blue/grey, indicating that it is recognized and present on the network.

Figure 7. MAX Icons and States



1. Discovered, but Not Added to the Network
2. Recognized, Present, and Reserved on the Network
3. Recognized, but Disconnected from the Network, Unreserved, or Reserved by Another Host

If your chassis does not appear in Available Devices, click **Refresh List**. If the chassis still does not appear, try the following:

- If you connected the cDAQ chassis directly to your computer, ensure your network card is configured to obtain an IP address automatically, then click **Refresh List**.



Note If you connected the cDAQ chassis directly to your computer, the setup time may be longer. Wait 30 to 60 seconds after the STATUS LED turns off, then click Refresh List.

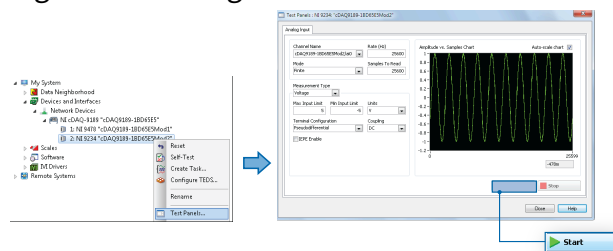
- Contact your system administrator to confirm that the network is working and that a firewall is not interfering with discovery. For additional troubleshooting resources for the cDAQ chassis, refer to ***Troubleshooting Chassis Connectivity in MAX***.

4. If the cDAQ chassis is not reserved automatically, select the chassis and click the

Reserve Chassis button. Refer to the **Reserving the Chassis in MAX** section for more information.

- Self-test your chassis in MAX by expanding **Devices and Interfaces**, right-clicking **NI cDAQ-<model number>**, and selecting **Self-Test**. Self-test performs a brief test to determine successful chassis installation. When the self-test finishes, a message indicates successful verification or if an error occurred. If an error occurs, refer to ni.com/support and search for the driver name.
- 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. Click **Start** to verify measurement functionality.

Figure 8. Running a Test Panel



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

Click **Close** to exit the test panel.



Note When in use, the cDAQ chassis may become warm to the touch. This is normal. For instructions on networking to a real-time controller, refer to the **Connecting to a Real-Time Controller** section.

Related concepts:

- [Reserving the Chassis in MAX](#)
- [Troubleshooting Chassis Connectivity in MAX](#)

Related information:

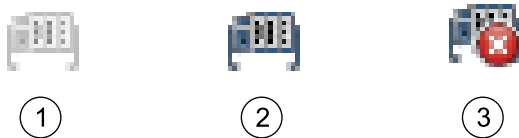
- [Connecting to a Real-Time Controller](#)

Reserving the Chassis in MAX

When the cDAQ chassis is connected to a network, multiple users can access the

chassis. To perform any DAQ functionality on the C Series modules, including reset chassis and self-test, you must reserve the cDAQ chassis in MAX. The following figure depicts the chassis state icons in MAX: an unreserved chassis or chassis reserved by another host appear with an X and reserved chassis appear as blue/grey. Only one user at a time can reserve the cDAQ chassis.

Figure 9. MAX Icons and States



1. Discovered, but Not Added to the Network
2. Recognized, Present, and Reserved on the Network
3. Recognized, but Disconnected from the Network, Unreserved, or Reserved by Another Host

If the cDAQ chassis was not reserved automatically after it was added (Add Device), you can reserve the cDAQ chassis in MAX by expanding **Devices and Interfaces** » **Network Devices**, selecting the chassis, and clicking the **Reserve Chassis** button. The Override Reservation dialog box appears when you attempt to explicitly reserve a chassis. Agreeing to override the reservation forces the cDAQ chassis to be reserved by the current user.

Troubleshooting Chassis Connectivity in MAX

If your cDAQ chassis is not discovered or becomes disconnected from the network, try the following suggestions.

If the cDAQ-9170/9173/9177 is not listed under **Device and Interfaces** » **Network Device**, complete the following steps.

1. Power off the system.
2. Ensure that all hardware is correctly installed.
3. Restart the system.

cDAQ-9170/9173/9177 Connected Directly to Your Computer is Not Discovered in MAX

For the initial configuration, the host computer must be set to obtain an IP address automatically since the cDAQ-9170/9173/9177 relies on Link-Local addressing to communicate with the host computer. The procedure for configuring your computer to obtain an IP address automatically varies depending on your operating system. Refer to your operating system documentation for instructions.

Removing Modules from the cDAQ-9170/9173/9177

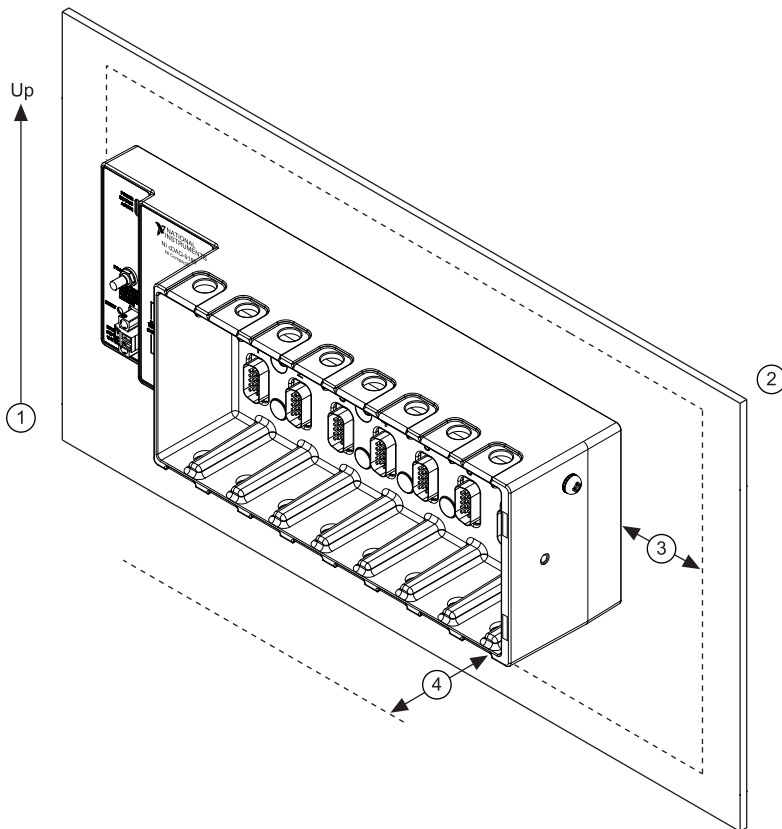
Complete the following steps to remove a C Series module from the cDAQ chassis.

1. Make sure that no I/O-side power is connected to the module. If the system is in a nonhazardous location, the chassis 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 chassis.

Mounting the cDAQ-9170/9173/9177

To ensure proper functionality during use at the maximum ambient temperature of 70 °C, you must mount the cDAQ chassis in the reference mounting configuration shown in the following image. Mounting the cDAQ chassis in the reference mounting configuration ensures that your system will operate correctly across the full operating temperature range and provide optimal C Series module accuracy. Observe the following guidelines to mount the cDAQ chassis in the reference mounting configuration.

Figure 10. cDAQ Chassis Reference Mounting Configuration (cDAQ-9189 Shown)



1. Horizontal mounting orientation
2. Panel mounting
3. Observe the cooling dimensions in the **Mounting Requirements** section
4. Allow space for cabling clearance according to the **Mounting Requirements** section



Note Panel mounting substrate options:

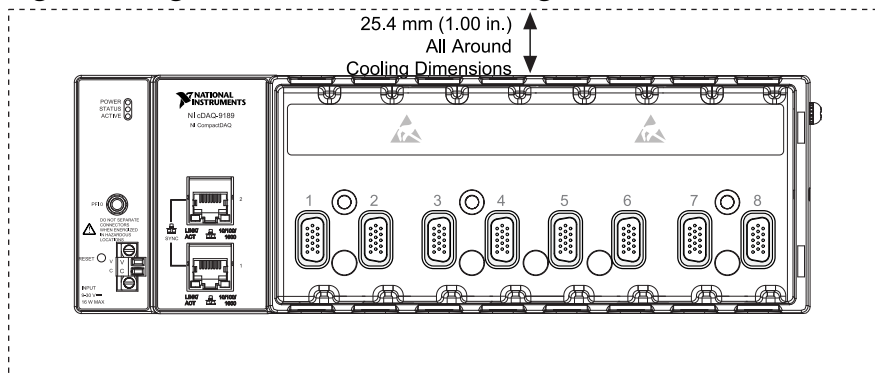
- Mount the cDAQ chassis directly to a metallic surface that is at least 1.6 mm (0.062 in.) thick and extends a minimum of 101.6 mm (4 in.) beyond all edges of the device.
- Use the NI-9904 Panel Mounting Kit to mount the cDAQ chassis to a metallic surface that is at least 1.6 mm (0.062 in.) thick and extends a minimum of 101.6 mm (4 in.) beyond all edges of the device.

Before using any of these mounting methods, record the serial number from the back of the cDAQ chassis so that you can identify the cDAQ chassis in MAX. You will be unable to read the serial number after you mount the cDAQ chassis.

cDAQ-9170/9173/9177 Mounting Requirements

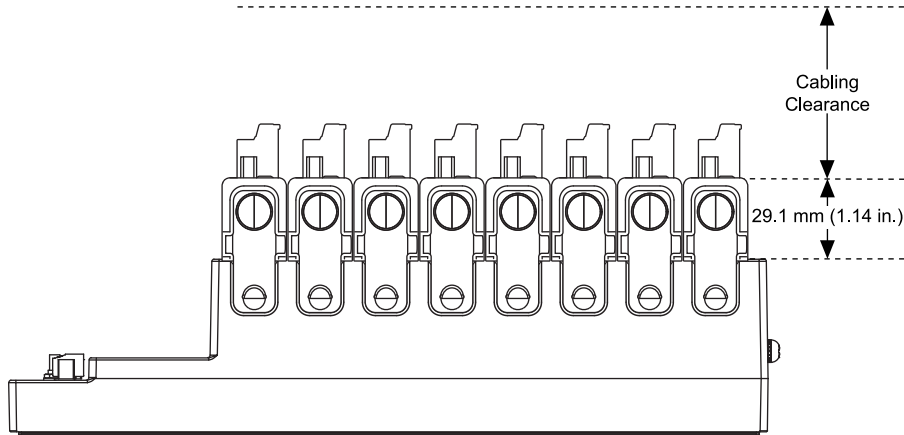
Your installation must meet the following requirements for cooling and cabling clearance. Allow 25.4 mm (1.00 in.) on all sides of the cDAQ chassis for air circulation, as shown in the following figure.

Figure 11. Figure 9. cDAQ Chassis Cooling Dimensions (cDAQ-9189 Shown)



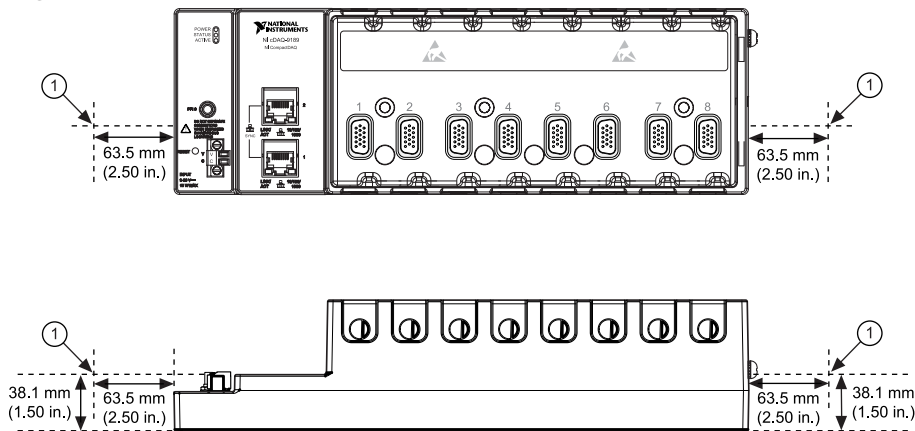
Allow the appropriate space in front of C Series modules for cabling clearance, as shown in the following figure. The different connector types on C Series modules require different cabling clearances. For a complete list of cabling clearances for C Series modules, visit ni.com/r/crioconn.

Figure 12. cDAQ Chassis Cabling Clearance (cDAQ-9189 Shown)



Measure the ambient temperature at each side of the cDAQ chassis, 63.5 mm (2.50 in.) from the side and 38.1 mm (1.50 in.) forward from the rear of the cDAQ chassis, as shown in the following figure.

Figure 13. cDAQ Chassis Ambient Temperature Location (cDAQ-9189 Shown)



1. Measure the ambient temperature here.

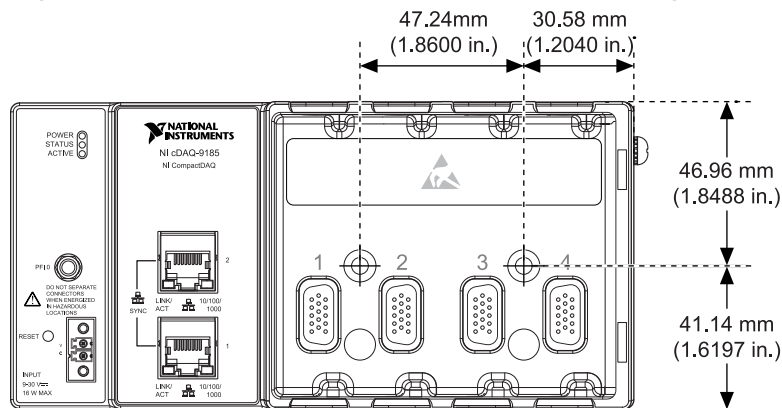
Mounting the cDAQ Chassis Directly on a Flat Surface

For environments with high shock and vibration, NI recommends mounting the cDAQ chassis directly on a flat, rigid surface using the mounting holes in the cDAQ chassis.

Complete the following steps to mount the cDAQ chassis directly on a flat surface. This mounting technique requires two M4 screws appropriate for the surface and a screwdriver.

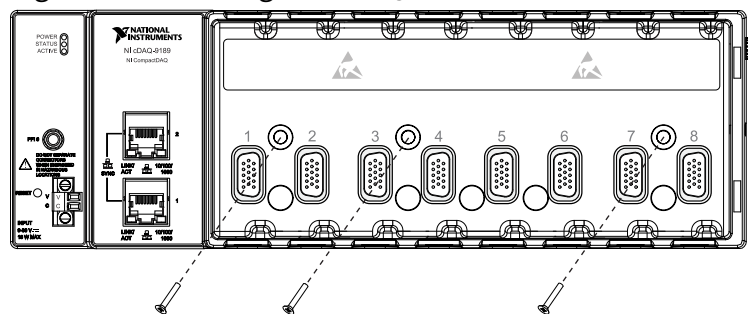
1. Prepare the surface for mounting the cDAQ chassis using the surface mounting dimensions listed in the following figure.

Figure 14. cDAQ-9170/9173/9177 Surface Mounting Dimensions (cDAQ-9185 Shown)



2. Align the cDAQ chassis on the surface.
3. Fasten the cDAQ chassis to the surface using the M4 screws appropriate for the surface. Tighten the screws to a maximum torque of $1.3 \text{ N} \cdot \text{m}$ ($11.5 \text{ lb} \cdot \text{in.}$).

Figure 15. Mounting the cDAQ Chassis on a Flat Surface (cDAQ-9189 Shown)



Mounting the cDAQ Chassis on a Panel with a Panel Mounting Kit

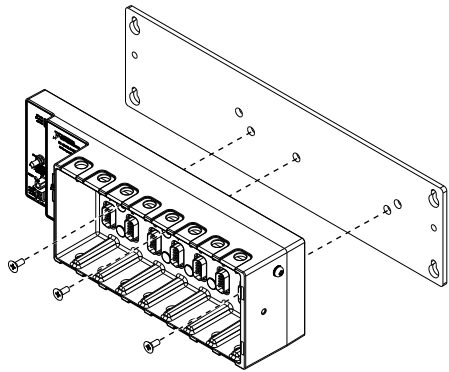
You can use a panel mounting kit to mount the cDAQ-chassis on a panel.

Complete the following steps to mount the cDAQ chassis on a panel. This mounting technique requires a Phillips #2 screwdriver and a NI-9904 panel mounting kit.

1. Align the cDAQ chassis and the panel mount plate.
2. Fasten the panel mounting plate to the cDAQ chassis using the screwdriver and two M4x25 screws, included in the kit. Tighten the screws to a maximum torque of $1.3 \text{ N} \cdot \text{m}$ ($11.5 \text{ lb} \cdot \text{in.}$).
3. Fasten the panel mounting plate to the surface using the screwdriver and screws

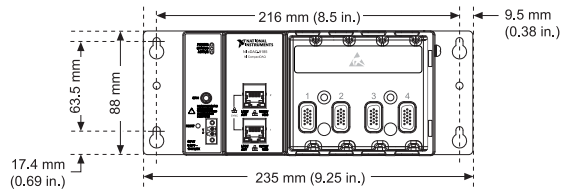
that are appropriate for the surface. The maximum screw size is M5 or number 10.

Figure 16. Mounting the Chassis on a Panel (cDAQ-9189 Shown)



The following figure shows the panel mounting dimensions for the cDAQ-9170/9173/9177.

Figure 17. cDAQ-9170/9173/9177 Panel Mounting Dimensions (cDAQ-9185 Shown)



Related reference:

- [Part Numbers for Recommended Cables and Accessories](#)

Alternate Mounting Configurations

The maximum operating temperature of 70 °C may be reduced for any mounting configuration other than the reference mounting configuration. A 10 °C (18 °F) reduction in maximum operating temperature is sufficient for most alternate mounting configurations. Follow the mounting requirements for all mounting configurations.

The published accuracy specifications, although not guaranteed for alternate mounting configurations, may be met depending on the system power and the thermal performance of the alternate mounting configuration.

Contact NI for further details regarding the impact of common alternate mounting configurations on maximum operating temperature and module accuracy.

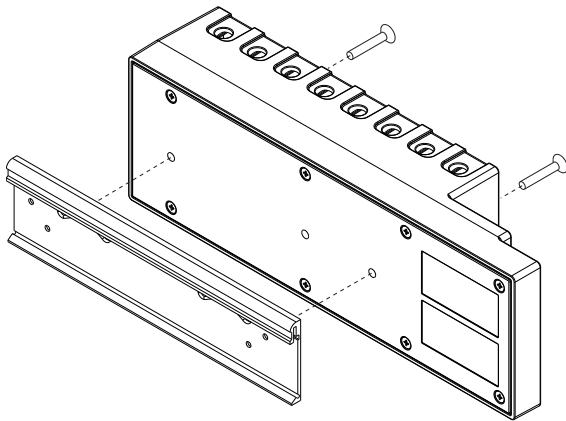
Mounting the cDAQ Chassis on a DIN Rail

You can use the NI DIN rail mounting kit to mount the cDAQ chassis on a standard 35-mm DIN rail. This mounting technique requires a Phillips #2 screwdriver and a panel mounting kit (NI 9912 DIN rail mounting kit, NI part number 779019-01).

Complete the following steps to mount the cDAQ chassis on a DIN rail.

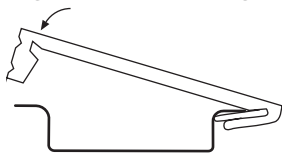
1. Align the cDAQ chassis and the DIN rail clip.
2. Fasten the DIN rail kit to the cDAQ chassis using the screwdriver and M4x25 screws. Tighten the screws to a maximum torque of $1.3 \text{ N} \cdot \text{m}$ ($11.5 \text{ lb} \cdot \text{in.}$).

Figure 18. Attaching the Chassis to a DIN Rail Clip (cDAQ-9189 Shown)



You must use the screws provided with the NI DIN rail kit because they are the correct depth and thread for the DIN rail clip.

Figure 19. Clipping the Chassis on a DIN Rail



3. Insert one edge of the DIN rail into the deeper opening of the DIN rail clip.
4. Press down firmly to compress the spring until the clip locks in place on the DIN rail.



Note Ensure that no C Series modules are in the cDAQ chassis before removing it from the DIN rail.

Mounting the Chassis on a Rack

NI offers the following rack-mount kits that you can use to mount the cDAQ chassis and other DIN rail-mountable equipment on a standard 482.6 mm (19 in.) rack:

- NI 9910 Sliding Rack-Mounting Kit, 779102-01
- NI Rack-Mounting Kit, 781989-01



Note You must order the NI 9912 DIN rail mounting kit, NI part number 779019-01, in addition to a rack-mount kit.

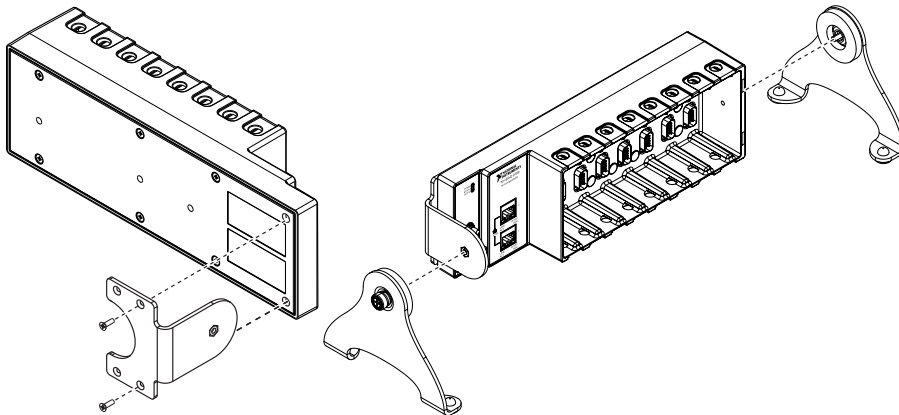
Mounting the cDAQ Chassis on a Desktop

You can use the NI 9901 desktop mounting kit, part number 779473-01, to mount the cDAQ chassis on a desktop. This mounting technique requires Phillips #1 and #2 screwdrivers and the NI 9901 desktop mounting kit, 779473-01.

Complete the following steps to mount the cDAQ chassis on a desktop.

1. Use a #1 Phillips screwdriver to remove the two screws from the back of the chassis on the front-panel side.
2. Use the screwdriver and the two M3x20 screws to attach the adapter bracket to the chassis.
3. Align one of the end brackets with the mounting hole at one of the ends of the chassis.
4. Use a #2 Phillips screwdriver to tighten the captive screw on the end bracket.

Figure 20. Attaching the Desktop Mounting Kit (cDAQ-9189 Shown)



5. Repeat steps [2](#) and [3](#) to attach the other end bracket to the other end of the

chassis.

The following figures show the desktop mounting kit dimensions.

Figure 21. Desktop Mounting Front Dimensions (cDAQ-9189 Shown)

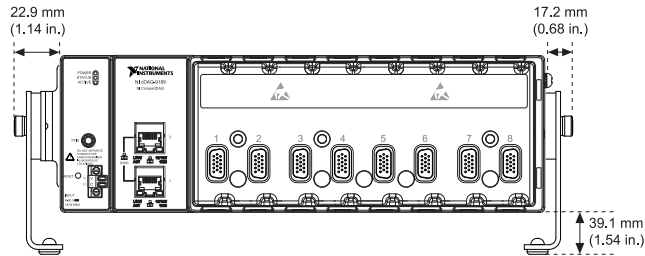
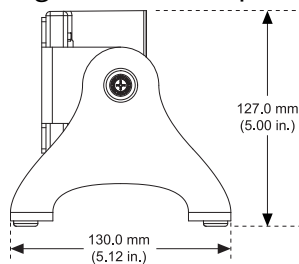


Figure 22. Desktop Mounting Side Dimensions

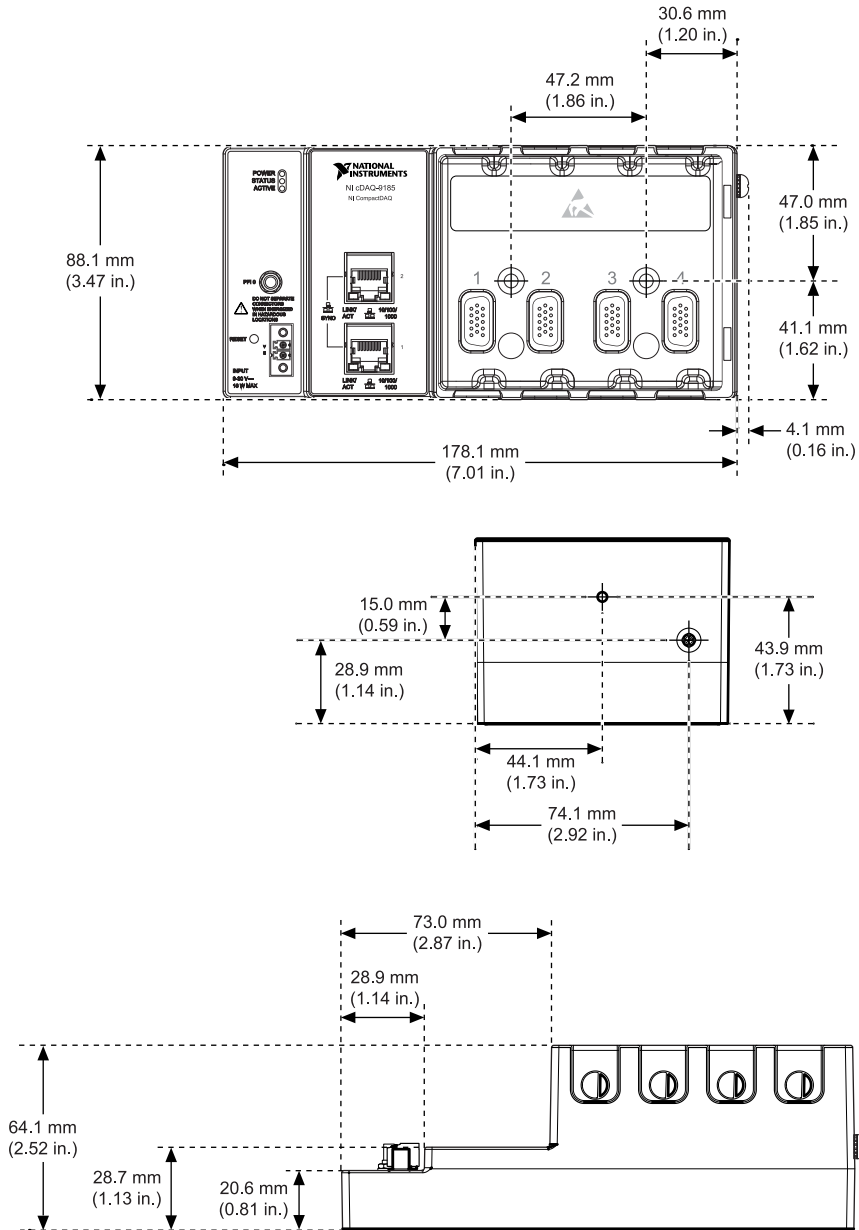


cDAQ-9170/9173/9177 Dimensions

The following figures show the front and side dimensions of the cDAQ-9170/9173/9177.

For detailed dimensional drawings and 3D models, visit ni.com/dimensions and search for cDAQ-9170/9173/9177.

Figure 23. cDAQ-9170/9173/9177 Dimensions



Performing Analog Input Measurements

To perform analog input measurements, install a supported analog input C Series module into any slot on the cDAQ chassis. 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-9173/9177 chassis has three AI timing engines, which means that three analog input tasks can be running at a time on a chassis. 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-9173/9177 chassis 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 chassis supports internal software triggering, external digital triggering, analog triggering, and internal time triggering.

Three triggers are available: Start Trigger, Reference Trigger, and Pause Trigger. An analog or digital signal can initiate these three trigger actions. Time can also initiate the Start Trigger. Up to two C Series parallel digital input modules can be used in any chassis 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 ***Using the Digital Input/Output***.

Refer to the related topic links below for more information about the analog input trigger signals.

Related concepts:

- [Using the Digital Input/Output](#)
- [AI Start Trigger Signal](#)
- [AI Reference Trigger Signal](#)
- [AI Pause Trigger Signal](#)

Analog Input Timing Signals

The cDAQ chassis 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*

Refer to the related topic links below for more information about the configuration of these signals.

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

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

Related concepts:

- [AI Sample Clock Signal](#)
- [AI Sample Clock Timebase Signal](#)
- [AI Convert Clock Signal Behavior For Analog Input Modules](#)
- [AI Start Trigger Signal](#)
- [AI Reference Trigger Signal](#)
- [AI Pause Trigger Signal](#)
- [PFI Filters](#)

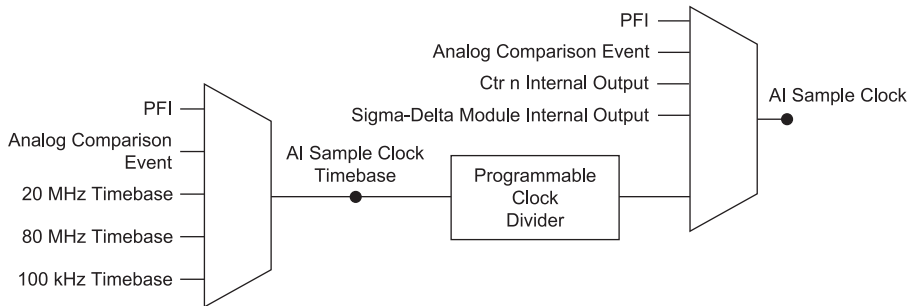
AI Sample Clock Signal

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 the following figure.

Figure 24. AI Sample Clock Timing Options



Routing the Sample Clock to an Output Terminal

You can route Sample Clock to any output PFI terminal on an installed C Series module. 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 chassis.

AI Convert Clock Signal Behavior For Analog Input Modules

Refer to the **Scanned Modules**, **Simultaneous Sample-and-Hold Modules**, **Delta-Sigma 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 A/D converter 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 A/D converter for each module and adds 10 μ s 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 μ s 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 A/D converters 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.

Delta-Sigma Modules

Delta-sigma C Series analog input modules function much like SSH modules, but use A/D converters that require a high-frequency oversample clock to produce accurate, synchronized data. Some delta-sigma modules in the cDAQ chassis 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 chassis 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 delta-sigma 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 delta-sigma modules are in an analog input task, the delta-sigma

modules also provide the signal used as the AI Sample Clock. This signal is used to cause A/D conversion for other modules in the system, just as the AI Sample Clock does when a delta-sigma module is not being used.

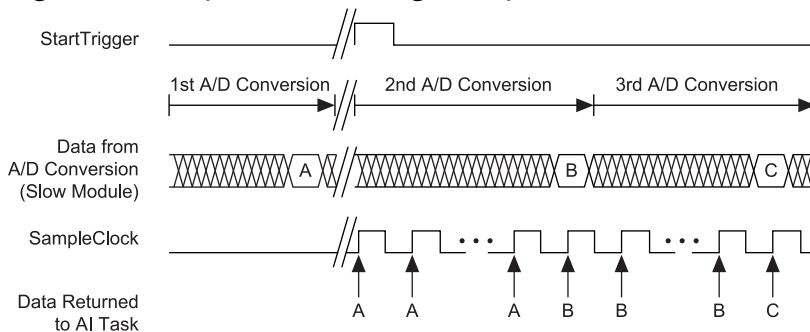
When delta-sigma modules are in an AI task, the chassis automatically issues a synchronization pulse to each delta-sigma module that resets their ADCs at the same time. You can also specify a specific time for the sync pulse to occur. Because of the filtering used in delta-sigma A/D converters, these modules usually exhibit a fixed input delay relative to non-delta-sigma modules in the system. This input delay is specified in the C Series module documentation.

When using multichassis device tasks across multiple chassis that contains channels from delta-sigma C Series modules ensure that one of those channels is the first in your channel list.

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 chassis, 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 the following figure.

Figure 25. 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.

For more information about which C Series modules are compatible with the cDAQ chassis, go to ni.com/info and enter 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 on an installed C Series module
- Counter n Internal Output

The source also can be one of several other internal signals on your cDAQ chassis. Refer to the ***Device Routing in MAX*** topic in the ***NI-DAQmx User Manual*** 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.

Using a Time Source

To use the Start Trigger signal with a time source, specify a specific time in NI-DAQmx. Refer to ***Timing and Triggering*** in the ***NI-DAQmx User Manual*** for more information on accessing time-based features in the NI-DAQmx API.

Routing AI Start Trigger to an Output Terminal

You can route the Start Trigger signal to any output PFI terminal on an installed C Series module. 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.

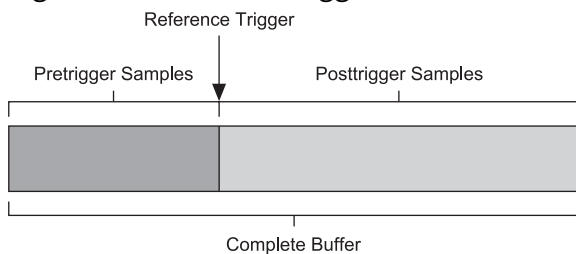
Once the acquisition begins, the cDAQ chassis writes samples to the buffer. After the cDAQ chassis captures the specified number of pretrigger samples, the cDAQ chassis begins to look for the reference trigger condition. If the reference trigger condition occurs before the cDAQ chassis captures the specified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

If the buffer becomes full, the cDAQ chassis 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 chassis discards it. Refer to the ***Can a Pretriggered Acquisition be Continuous?*** document for more information. To access this document, go to ni.com/info and enter the Info Code `rdcanq`.

When the reference trigger occurs, the cDAQ chassis continues to write samples to the buffer until the buffer contains the number of posttrigger samples desired. The following figure shows the final buffer.

Figure 26. 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 on an installed C Series module or one of several internal signals on the cDAQ chassis can provide the source. Refer to the ***Device Routing in MAX*** topic in the ***NI-DAQmx User Manual*** 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 on an installed C Series

module. 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 on an installed C Series module or one of several other internal signals on your cDAQ chassis. Refer to the ***Device Routing in MAX*** topic in the ***NI-DAQmx User Manual*** 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.

Analog Input Sync Pulse

You can use time as a trigger for sync pulses on delta-sigma C Series modules. To use a sync pulse with a time source, specify a specific time in NI-DAQmx. Refer to ***Timing***

and Triggering in the *NI-DAQmx User Manual* for more information on accessing time-based features in the NI-DAQmx API.



Note To accurately synchronize delta-sigma devices in two or more separate tasks, you must specify the same sync pulse. Otherwise, a sync pulse is initiated by software implicitly, even if time start triggers are specified for the tasks. However, for multichassis tasks, the sync pulses and start triggers are automatically synchronized.

Getting Started with AI Applications in Software

You can use the cDAQ chassis 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 User Manual* for more information.

Generating Analog Output

To generate analog output, install an analog output C Series module in any slot on the cDAQ chassis. 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 chassis.

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 chassis 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 chassis 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 chassis 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

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 chassis supports internal software triggering, external digital triggering, analog triggering, and internal time triggering.

Analog output supports two different triggering actions: AO Start Trigger and AO Pause Trigger. An analog or digital signal can initiate these actions. Time can also initiate the Start Trigger. Up to two C Series parallel digital input modules can be used in any chassis slot to supply a digital trigger. An analog trigger can be supplied by some C Series analog modules.

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

Related concepts:

- [AO Start Trigger Signal](#)

- [AO Pause Trigger Signal](#)

Analog Output Timing Signals

The cDAQ chassis 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*

Refer to the related topic links below for more information about the configuration of these signals.

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

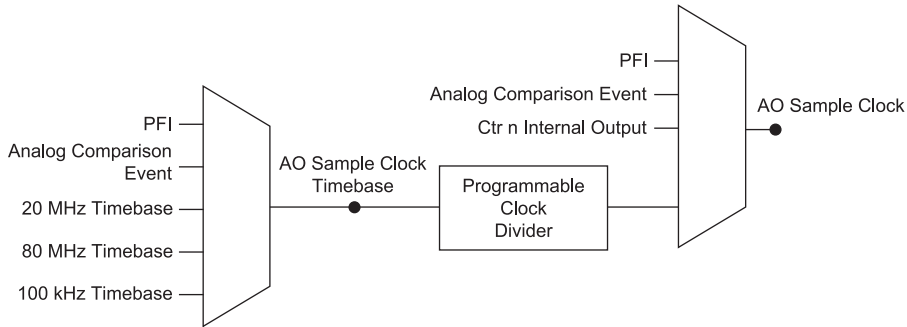
Related concepts:

- [AO Sample Clock Signal](#)
- [AO Sample Clock Timebase Signal](#)
- [AO Start Trigger Signal](#)
- [AO Pause Trigger Signal](#)
- [PFI Filters](#)

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 the following figure.

Figure 27. Analog Output Timing Options



Routing AO Sample Clock to an Output Terminal

You can route AO Sample Clock to any output PFI terminal on an installed C Series module. 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 chassis.

Using Delta-Sigma Modules

The oversample clock is used as the AO 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 delta-sigma modules with different oversample clock frequencies are used in an analog input task, the AO 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 AO Sample Clock Timebase.

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 User Manual***.

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 on an installed C Series module
- AI Reference Trigger
- AI Start Trigger

The source also can be one of several internal signals on the cDAQ chassis. Refer to the ***Device Routing in MAX*** topic in the ***NI-DAQmx User Manual*** 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.

Using a Time Source

To use AO Start Trigger signal with a time source, specify a specific time in NI-DAQmx. Refer to ***Timing and Triggering*** in the ***NI-DAQmx User Manual*** for more information on accessing time-based features in the NI-DAQmx API.

Routing AO Start Trigger Signal to an Output Terminal

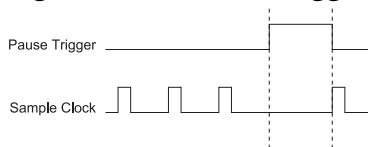
You can route AO Start Trigger to any output PFI terminal on an installed C Series module. The output is an active high pulse.

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.

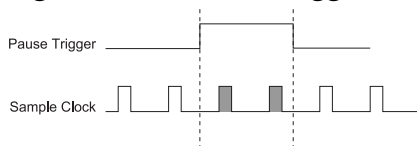
When you generate analog output signals, the generation pauses as soon as the pause trigger is asserted. If the source of the sample clock is the onboard clock, the generation resumes as soon as the pause trigger is deasserted, as shown in the following figure.

Figure 28. 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 the following figure.

Figure 29. 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 on an installed C Series module or one of several other internal signals on the cDAQ chassis.

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 User Manual*** 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.

Analog Output Sync Pulse

You can use time as a trigger for sync pulses on delta-sigma C Series modules. To use a sync pulse with a time source, specify a specific time in NI-DAQmx. Refer to ***Timing and Triggering*** in the ***NI-DAQmx User Manual*** for more information on accessing time-based features in the NI-DAQmx API.



Note To accurately synchronize delta-sigma devices in two or more separate tasks, you must specify the same sync pulse. Otherwise, a sync pulse is initiated by software implicitly, even if time start triggers are specified for the tasks. However, for multichassis tasks, the sync pulses and start triggers are automatically synchronized.

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 chassis 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 to the ***NI-DAQmx User Manual***.

Using the Digital Input/Output

To use digital input/output (DIO), install a digital C Series module into any slot on the cDAQ chassis. 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 chassis slot and can perform the following tasks:

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

Parallel digital modules can be used in any chassis 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/r/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 chassis 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 chassis supports internal software triggering, external digital triggering, analog triggering, and internal time triggering.

Three triggers are available: Start Trigger, Reference Trigger, and Pause Trigger. An analog or digital trigger can initiate these three trigger actions. Time can also initiate the Start Trigger. Up to two C Series parallel digital input modules can be used in any chassis 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 ***Analog Input Triggering Signals*** and ***Analog Output Triggering Signals***.

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

Related concepts:

- [Analog Input Triggering Signals](#)
- [Analog Output Triggering Signals](#)
- [DI Start Trigger Signal](#)
- [DI Reference Trigger Signal](#)
- [DI Pause Trigger Signal](#)

Digital Input Timing Signals

The cDAQ chassis features the following digital input timing signals.

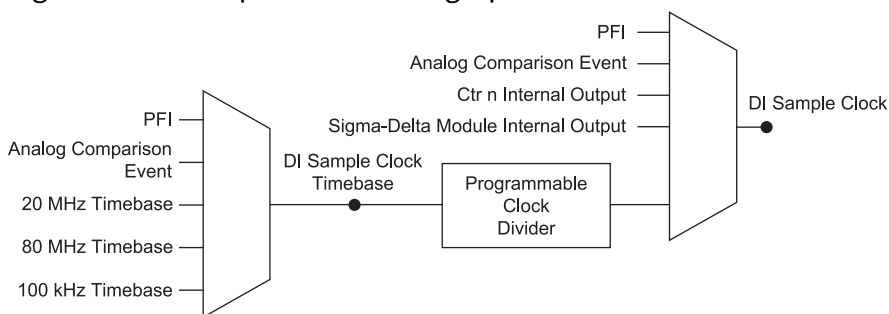
DI Sample Clock Signal, DI Start Trigger Signal, DI Reference Trigger Signal, and DI Pause Trigger Signal support digital filtering. Refer to the **PFI Filters** section for more information.

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 chassis 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 the following figure.

Figure 30. DI Sample Clock Timing Options



Routing DI Sample Clock to an Output Terminal

You can route DI Sample Clock to any output PFI terminal on an installed C Series module.

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 chassis.

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*** for more information.

Using an External Source

You can route the following signals as DI Sample Clock:

- Any PFI terminal on an installed C Series module
- Analog Comparison Event (an analog trigger)

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

Routing DI Sample Clock to an Output Terminal

You can route DI Sample Clock to any output PFI terminal on an installed C Series module. The PFI circuitry inverts the polarity of DI Sample Clock before driving the PFI terminal.

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 on an installed C Series module
- Counter n Internal Output

The source also can be one of several other internal signals on the cDAQ chassis. 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. Using a Time Source To use the Start Trigger signal with a time source, specify a specific time in NI-DAQmx. Refer to the Timestamps and Time Triggering topics in the NI-DAQmx Help for more information on accessing time-based features in the NI-DAQmx API. Routing DI Start Trigger to an Output Terminal You can route DI Start Trigger to any output PFI terminal on an installed C Series module. The output is an active high pulse.

DI Reference Trigger Signal

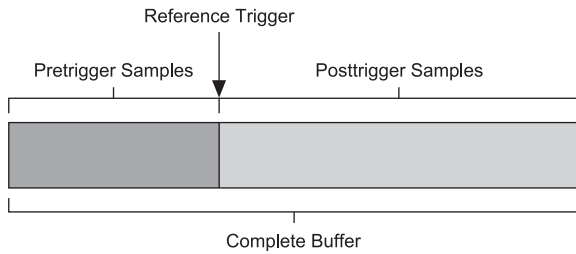
Use a reference trigger (di/ReferenceTrigger) signal to stop a measurement acquisition. To use a reference trigger, specify a buffer of finite size and a number of pretrigger samples (samples that occur before the reference trigger). The number of posttrigger samples (samples that occur after the reference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

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

If the buffer becomes full, the cDAQ chassis 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 chassis discards it. Refer to the ***Can a Pretriggered Acquisition be Continuous?*** document for more information. To access this document, go to ni.com/r/rdcanq.

When the reference trigger occurs, the cDAQ chassis continues to write samples to the buffer until the buffer contains the number of posttrigger samples desired. The following figure shows the final buffer.

Figure 31. 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 on an installed C Series module or one of several internal signals on the cDAQ chassis can provide the source. Refer to the ***Device Routing in MAX*** topic in the ***NI-DAQmx User Manual*** 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 on an installed C Series module. 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.

Some C Series modules can generate a trigger based on an analog signal. In NI-DAQmx,

this is called the Analog Comparison Event.

Using a Digital Source

To use DI Pause Trigger, specify a source and a polarity. The source can be either from PFI on an installed C Series module or one of several other internal signals on your cDAQ chassis. Refer to ***Device Routing in MAX*** topic in the ***NI-DAQmx User Manual*** for more information.

Using an Analog Source

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 chassis samples the inputs with a user-configured Filter Clock derived from the chassis 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

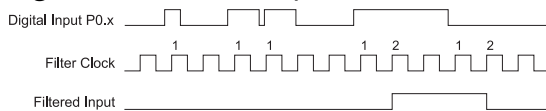
1. **Tp** is a nominal value; the accuracy of the chassis timebase and I/O distortion will affect this value.

depend on the phase of the Filter Clock relative to the input signal.

The following figure 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 32. Filter Example



Getting Started with DI Applications in Software

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

- Single-point acquisition
- Finite acquisition
- Continuous acquisition

For more information about programming digital input applications and triggers in software, refer to the ***NI-DAQmx User Manual*** for more information.

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 on an installed C Series module.

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 chassis 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 chassis directly to a PC buffer.
- **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 chassis 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, install a digital output C Series module in any slot on the cDAQ chassis. 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 chassis 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 chassis 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 chassis 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

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 chassis supports internal software triggering, external digital triggering, analog triggering, and internal time triggering.

Digital output supports two different triggering actions: DO Start Trigger and DO Pause Trigger. A digital or analog trigger can initiate these actions. Time can also initiate the

Start Trigger. Any PFI terminal on an installed C Series module 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 ***DO Start Trigger Signal*** and ***DO Pause Trigger Signal*** for more information about the digital output trigger signals.

Related concepts:

- [DO Start Trigger Signal](#)
- [DO Pause Trigger Signal](#)

Digital Output Timing Signals

The cDAQ chassis features the following DO timing signals.

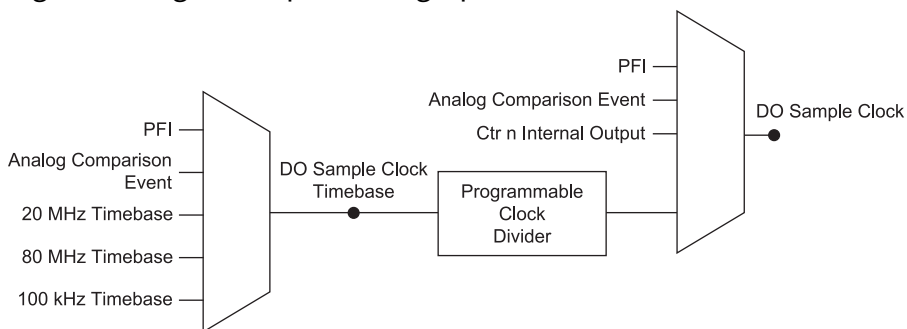
DO Sample Clock Signal, DO Start Trigger Signal, and DO Pause Trigger Signal support digital filtering. Refer to the ***PFI Filters*** section for more information.

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 the following figure.

Figure 33. Digital Output Timing Options



Routing DO Sample Clock to an Output Terminal

You can route DO Sample Clock to any output PFI terminal on an installed C Series module. 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 chassis.

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 on an installed C Series module
- AI Reference Trigger
- AI Start Trigger

The source also can be one of several internal signals on the cDAQ chassis. Refer to ***Device Routing in MAX*** in the ***NI-DAQmx User Manual*** 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.



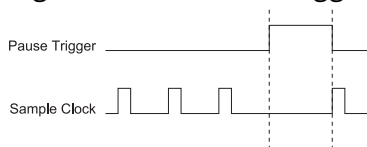
Note Depending on the C Series module capabilities, you may need two modules to utilize analog triggering. Using a Time Source To use the Start Trigger signal with a time source, specify a specific time in NI-DAQmx. Refer to the **Timing and Triggering** in the **NI-DAQmx User Manual** for more information on accessing time-based features in the NI-DAQmx API. Routing DO Start Trigger Signal to an Output Terminal You can route DO Start Trigger to any output PFI terminal on an installed C Series module. 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.

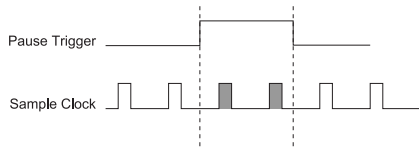
When you generate digital output signals, the generation pauses as soon as the pause trigger is asserted. If the source of the sample clock is the onboard clock, the generation resumes as soon as the pause trigger is deasserted, as shown in the following figure.

Figure 34. 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 the following figure.

Figure 35. 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 on an installed C Series module or one of several other internal signals on the cDAQ chassis.

You also can specify whether the samples are paused when DO Pause Trigger is at a logic high or low level. Refer to ***Device Routing in MAX*** in the ***NI-DAQmx User Manual*** 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 chassis 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 to the ***NI-DAQmx User Manual***.

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.

Using PFI

You can configure channels of a parallel digital module as Programmable Function Interface (PFI) terminals. Up to two digital modules can be used to access PFI terminals in a single chassis.

You can configure each module 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 chassis samples the inputs with a user-configured Filter Clock derived from the chassis 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 the following table.

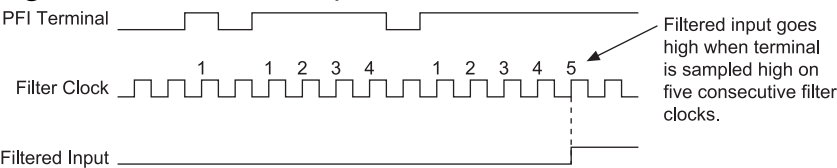
Table 13. Selectable PFI Filter Settings

| 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 μ s (medium) | 80 MHz | 12.5 ns | 6.4 μ s | 6.3875 μ s |
| 2.56 ms (high) | 100 kHz | 10 μ s | 2.56 ms | 2.55 ms |
| Custom | User-configurable | 1 Filter Clock | T _{user} | T _{user} - (1 Filter Clock |

| Filter Setting | Filter Clock | Jitter | Min Pulse Width* to Pass | Max Pulse Width* to Not Pass |
|--|--------------|--------|-----------------------------|---------------------------------|
| | | period | | period) |
| * Pulse widths are nominal values; the accuracy of the chassis timebase and I/O distortion will affect these values. | | | | |

On power up, the filters are disabled. The following figure shows an example of a low-to-high transition on an input that has a custom filter set to **N** = 5.

Figure 36. PFI Filter Example

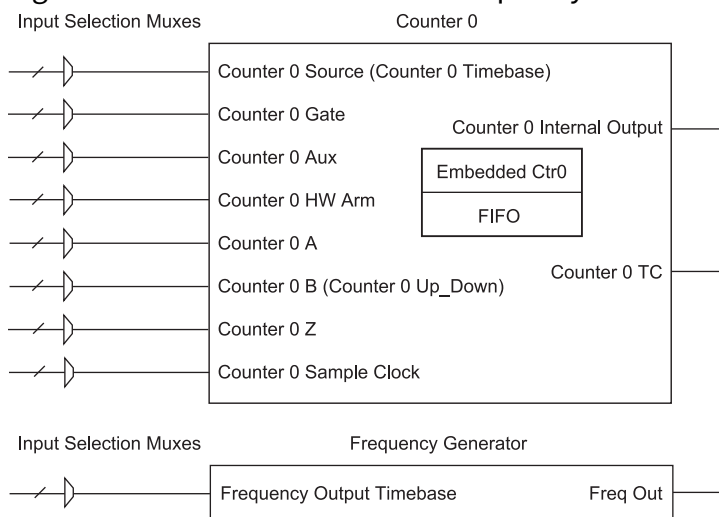


Using Counters

The cDAQ chassis 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. The following figure shows the cDAQ chassis Counter 0 and the frequency generator. All four counters on the cDAQ chassis are identical.

Figure 37. Chassis 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 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 chassis

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. The following table shows the different options for the different measurements.

Table 14. 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 |
| Buffered Position | No | Yes |

| Measurement | Implicit Timing Support | Sample Clocked Timing Support |
|-------------------------------------|-------------------------|-------------------------------|
| Buffered Two-Signal Edge Separation | Yes | Yes |

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 or a specific time. 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**—You can use the start trigger for counter output functions. You can configure a start trigger to begin a finite or continuous pulse generation. After 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. Possible triggers for counter output operations are a hardware signal or a specific time.

For counter input functions, you 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.

Counter Input Applications

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

- [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.

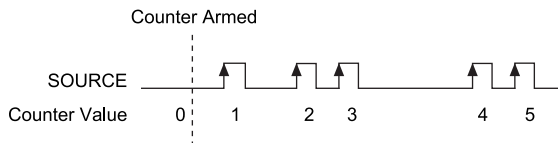
Refer to the following sections for more information about edge counting options.

- ***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. The following figure shows an example of single point edge counting.

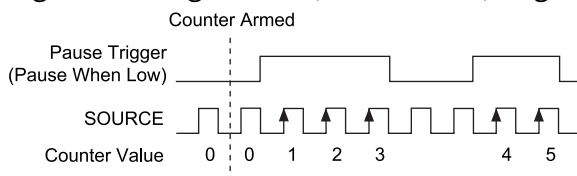
Figure 38. Single Point (On-Demand) Edge Counting



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. The following figure shows an example of on-demand edge counting with a pause trigger.

Figure 39. Single Point (On-Demand) Edge Counting with Pause Trigger



Buffered (Sample Clock) Edge Counting

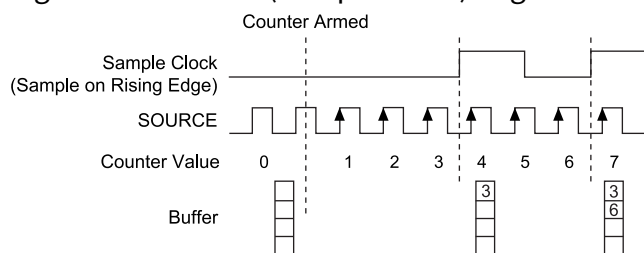
With buffered edge counting (edge counting using a sample clock), the counter counts the number of edges on the Source input after the counter is armed. The value of the counter is sampled on each active edge of a sample clock and stored in the FIFO. The NI ASIC transfers the sampled values to host memory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

The following figure 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 40. 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 chassis pulse-width measurement options.

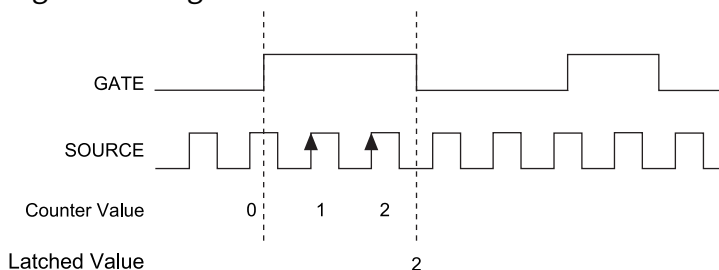
- ***Single Pulse-Width Measurement***
- ***Implicit Buffered Pulse-Width measurement***
- ***Sample Clocked Buffered Pulse-Width Measurement***

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.

The following figure shows an example of a single pulse-width measurement.

Figure 41. Single Pulse-Width Measurement



Implicit Buffered Pulse-Width Measurement

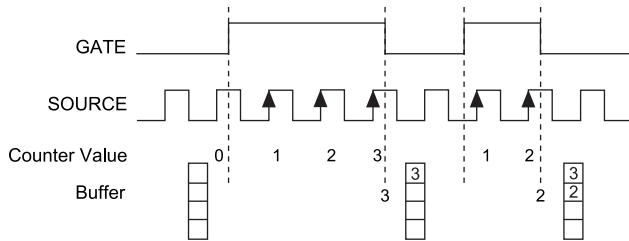
An implicit buffered pulse-width measurement is similar to single pulse-width measurement, but buffered pulse-width measurement takes measurements over multiple pulses.

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 NI ASIC transfers the sampled values to host memory using a high-speed data stream.

The following figure shows an example of an implicit buffered pulse-width

measurement.

Figure 42. Implicit Buffered Pulse-Width Measurement



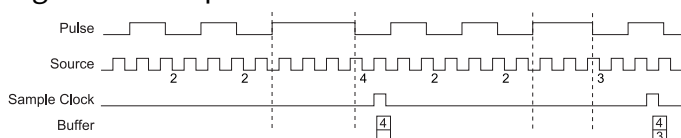
Sample Clocked Buffered Pulse-Width Measurement

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 NI ASIC transfers the sampled values to host memory using a high-speed data stream.

The following figure shows an example of a sample clocked buffered pulse-width measurement.

Figure 43. 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.

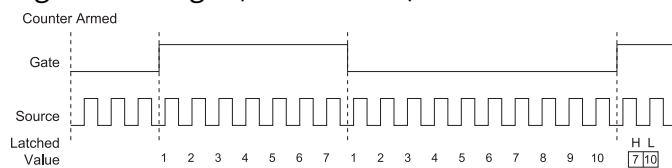
Refer to the following sections for more information about cDAQ chassis pulse measurement options.

- ***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 the following figure.

Figure 44. Single (On-Demand) Pulse Measurement



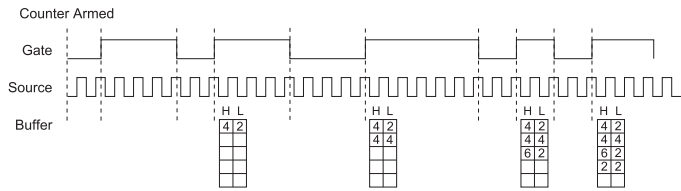
Implicit Buffered Pulse Measurement

In an implicit buffered pulse measurement, on each edge of the Gate signal, the counter stores the count in the FIFO. The NI ASIC 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.

The following figure shows an example of an implicit buffered pulse measurement.

Figure 45. Implicit Buffered Pulse Measurement



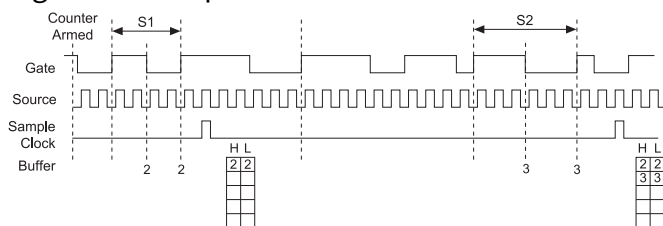
Sample Clocked Buffered Pulse Measurement

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 NI ASIC transfers the sampled values to host memory using a high-speed data stream.

The following figure shows an example of a sample clocked buffered pulse measurement.

Figure 46. 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***



Note 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.

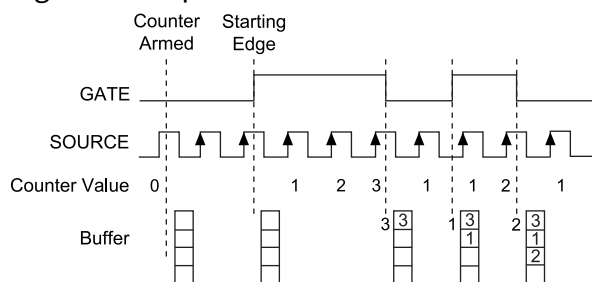
Implicit Buffered Semi-Period Measurement

In implicit buffered semi-period measurements, on each edge of the Gate signal, the counter stores the count in the FIFO. The NI ASIC 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.

The following figure shows an example of an implicit buffered semi-period measurement.

Figure 47. 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 chassis 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**

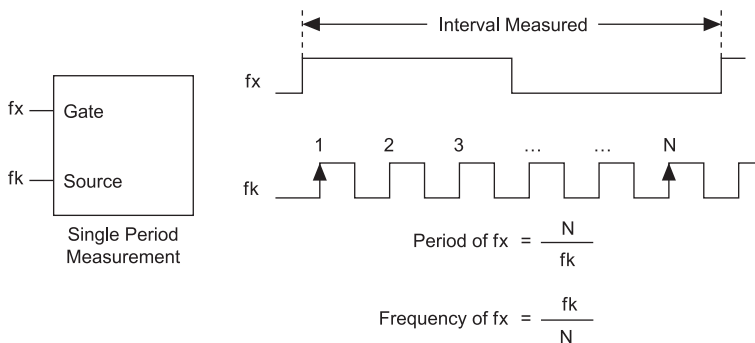
Low Frequency with One Counter

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. The following figure illustrates this method.

Figure 48. 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.

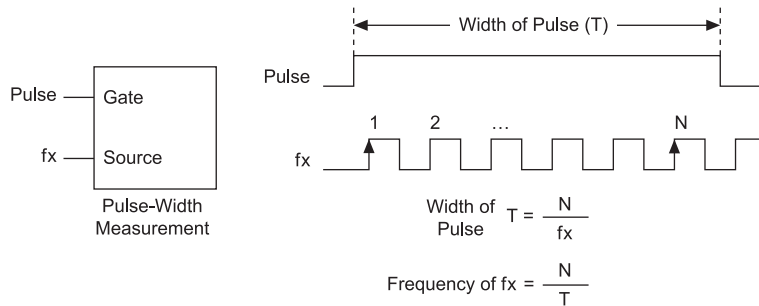
In this method, you route a pulse of known duration (***T***) to the Gate of a counter. You can generate the pulse using a second counter. You also can generate the pulse externally and connect it to a PFI terminal on an installed C Series module. You only need to use one counter if you generate the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

The following figure illustrates this method. Another option is to measure the width of a known period instead of a known pulse.

Figure 49. 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.

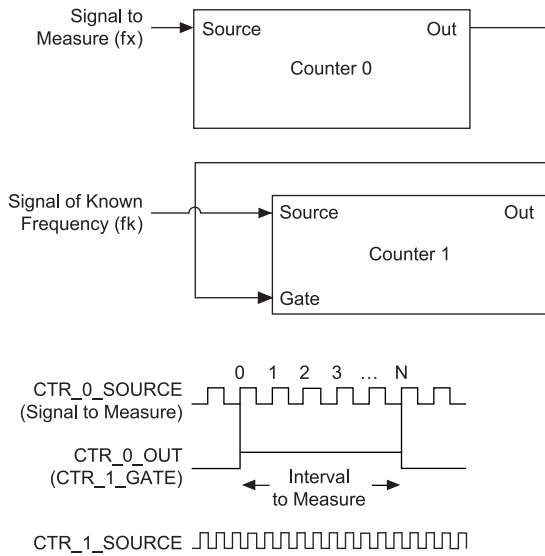
When measuring a large range of frequencies with two counters, you generate a long pulse using the signal to measure. You then measure the long pulse with a known timebase. The cDAQ chassis can measure this long pulse more accurately than the faster input signal.



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 the following figure. 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 50. 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 **$fx = fk * (N/J)$** .

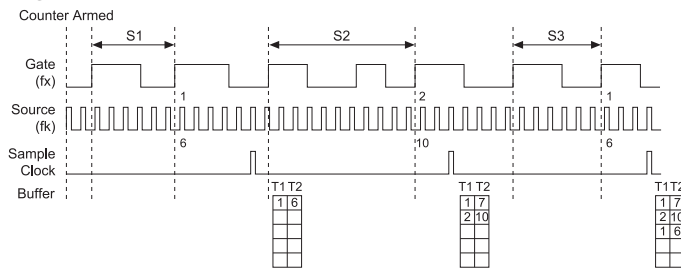
Sample Clocked Buffered Frequency Measurement

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

A sample clocked buffered frequency measurement with `Cl.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 the following figure. The frequency measured is:

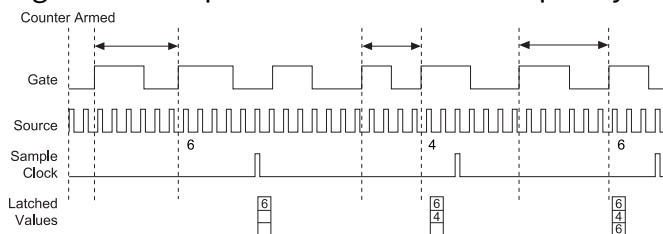
$$fx = fk * (T1/ T2)$$

Figure 51. 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 the following figure.

Figure 52. 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 the following table.

- 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 $1/f_x$.
- Two counter high frequency—With the two counter high frequency method, the second counter provides a known measurement time. The gate frequency equals $1/\text{measurement time}$.
- 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/f_x 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 15. Frequency Measurement Methods

| Variable | Sample Clocked | One Counter | Two Counters | |
|----------------------|--|------------------------------------|----------------------------------|---|
| | | | High Frequency | Large Range |
| fk | Known timebase | Known timebase | $\frac{1}{\text{gating period}}$ | Known timebase |
| Measurement time | $\frac{1}{f_s}$ | $\frac{1}{f_x}$ | gating period | $\frac{N}{f_x}$ |
| Max. frequency error | $f_x \times \frac{f_x}{f_k \times \left[\frac{f_x}{f_s} - 1 \right]}$ | $f_x \times \frac{f_x}{f_k - f_x}$ | fk | $f_x \times \frac{f_x}{N \times f_k - f_x}$ |

| Variable | Sample Clocked | One Counter | Two Counters | |
|--|---|-------------------------|-------------------|----------------------------------|
| | | | High Frequency | Large Range |
| Max. error % | $\frac{f_x}{f_k \times \left[\frac{f_x}{f_s} - 1 \right]}$ | $\frac{f_x}{f_k - f_x}$ | $\frac{f_k}{f_x}$ | $\frac{f_x}{N \times f_k - f_x}$ |
| Note: Accuracy equations do not take clock stability into account. Refer to the cDAQ-9170/9173/9177 Specifications 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, the following table summarizes the results.

Table 16. 50 kHz Frequency Measurement Methods

| Variable | Sample Clocked | One Counter | Two Counters | |
|-----------------------------|----------------|-------------|----------------|-------------|
| | | | High Frequency | Large Range |
| <i>f_x</i> | 50,000 | 50,000 | 50,000 | 50,000 |
| <i>f_k</i> | 80 M | 80 M | 1,000 | 80 M |
| Measurement time (ms) | 1 | .02 | 1 | 1 |
| <i>N</i> | — | — | — | 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, the following table shows the results for 5 MHz.

Table 17. 5 MHz Frequency Measurement Methods

| Variable | Sample Clocked | One Counter | Two Counters | |
|---------------------------|----------------|-------------|----------------|-------------|
| | | | High Frequency | Large Range |
| f_x | 5 M | 5 M | 5 M | 5 M |
| f_k | 80 M | 80 M | 1,000 | 80 M |
| Measurement time (ms) | 1 | .0002 | 1 | 1 |
| N | — | — | — | 5,000 |
| 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 the first table. 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.

The last table summarizes some of the differences in methods of measuring frequency.

Table 18. 5 MHz Frequency Measurement Methods

| Method Comparison | Sample Clocked (Averaged) | One Counter | Two Counters | |
|--|---------------------------|-------------|----------------|-------------|
| | | | High Frequency | Large Range |
| Number of counters used | 1 | 1 | 1 or 2 | 2 |
| Number of measurements returned | 1 | 1 | 1 | 1 |
| Measures high frequency signals accurately | Good | Poor | Good | Good |
| Measures low frequency signals accurately | Good | Good | Good | Poor |

For information about connecting counter signals, refer to the 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 chassis position measurement options.

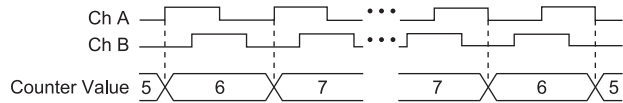
Measurements Using Quadrature Encoders

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.

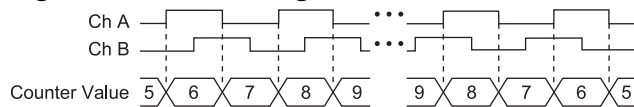
The following figure shows a quadrature cycle and the resulting increments and decrements for X1 encoding. When channel A leads channel B, the increment occurs on the rising edge of channel A. When channel B leads channel A, the decrement occurs on the falling edge of channel A.

Figure 53. X1 Encoding



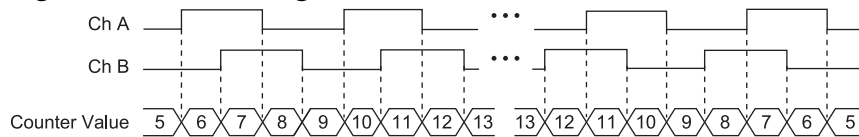
- **X2 Encoding**—The same behavior holds for X2 encoding except the counter increments or decrements on each edge of channel A, depending on which channel leads the other. Each cycle results in two increments or decrements, as shown in the following figure.

Figure 54. 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 the following figure.

Figure 55. X4 Encoding



Channel Z Behavior

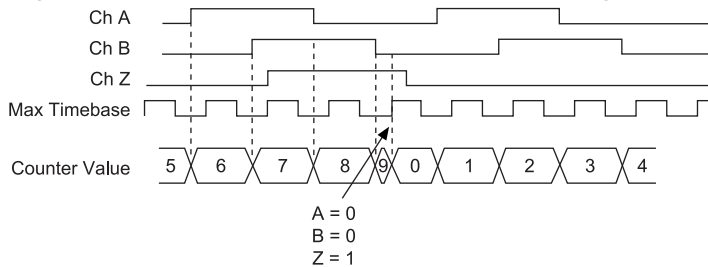
Some quadrature encoders have a third channel, channel Z, which is also referred to as the index channel. A high level on channel Z causes the counter to be reloaded with a specified value in a specified phase of the quadrature cycle. You can program this reload to occur in any one of the four phases 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 the following figure, 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 the following figure, 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 56. Channel Z Reload with X4 Decoding

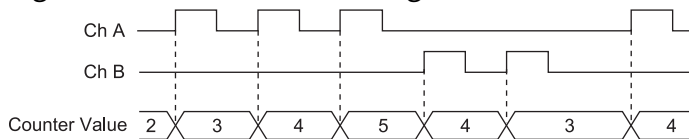


Measurements Using Two Pulse Encoders

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 the following figure.

Figure 57. Measurements Using Two Pulse Encoders



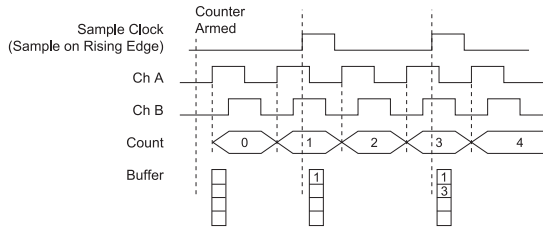
For information about connecting counter signals, refer to the section.

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 NI ASIC 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.

The following figure shows an example of a buffered X1 position measurement.

Figure 58. 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.

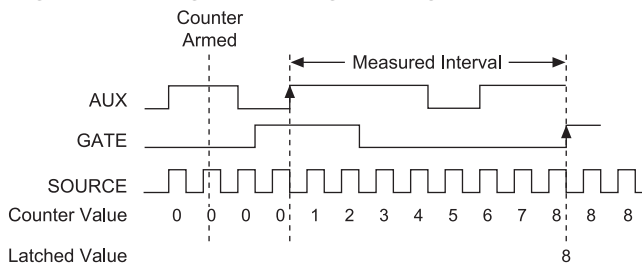
Refer to the following sections for more information about the cDAQ chassis edge-separation measurement options.

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.

The following figure shows an example of a single two-signal edge-separation measurement.

Figure 59. Single Two-Signal Edge-Separation Measurement



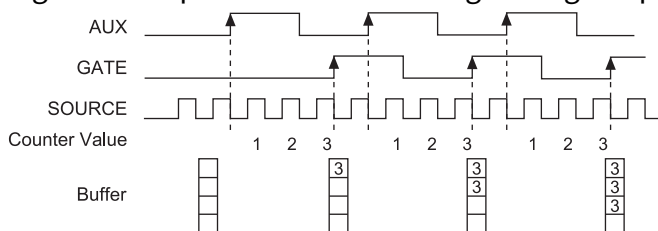
Implicit Buffered Two-Signal Edge-Separation Measurement

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 NI ASIC transfers the sampled values to host memory using a high-speed data stream.

The following figure shows an example of an implicit buffered two-signal edge-separation measurement.

Figure 60. Implicit Buffered Two-Signal Edge-Separation Measurement



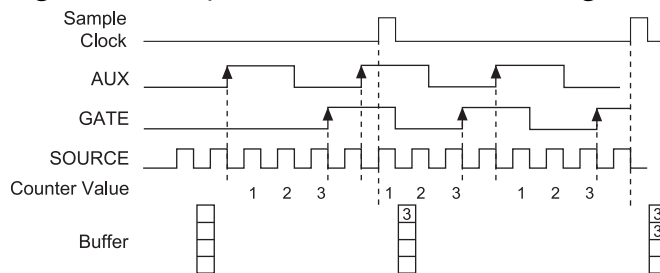
Sample Clocked Buffered Two-Signal Separation Measurement

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 NI ASIC transfers the sampled values to host memory using a high-speed data stream.

The following figure shows an example of a sample clocked buffered two-signal separation measurement.

Figure 61. 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 section.

Counter Output Applications

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

- [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 chassis simple pulse generation options.

Single Pulse Generation

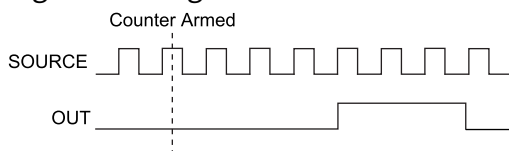
The counter can output a single pulse. The pulse appears on the Counter n Internal Output signal of the counter.

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).

The following figure 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 62. Single Pulse Generation



Single Pulse Generation with Start Trigger

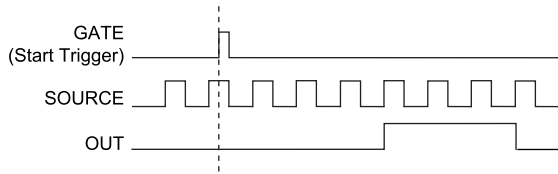
The counter can output a single pulse in response to one pulse on a hardware Start Trigger signal. The pulse appears on the Counter n Internal Output signal of the counter.

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).

The following figure 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 63. Single Pulse Generation with Start Trigger



Pulse Train Generation

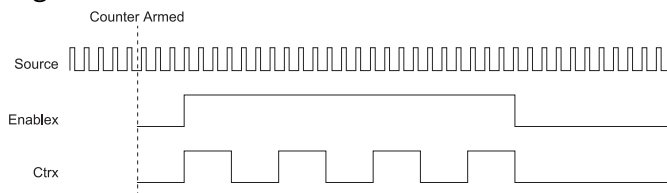
Refer to the following sections for more information about the cDAQ chassis pulse train generation options.

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 chassis 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 64. 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.

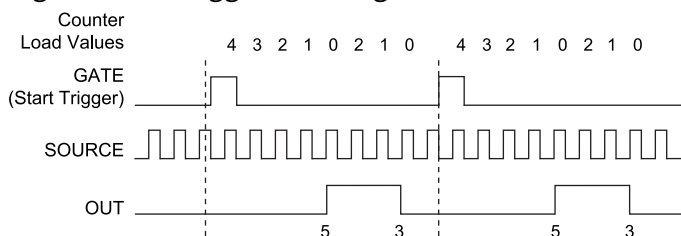
You can route the Start Trigger signal to the Gate input of the counter. You can specify a delay from the Start Trigger to the beginning of each pulse. You also can specify the pulse width. The delay and pulse width are measured in terms of a number of active edges of the Source input. The initial delay can be applied to only the first trigger or to all triggers using the **CO.EnableInitialDelayOnRetrigger** 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.

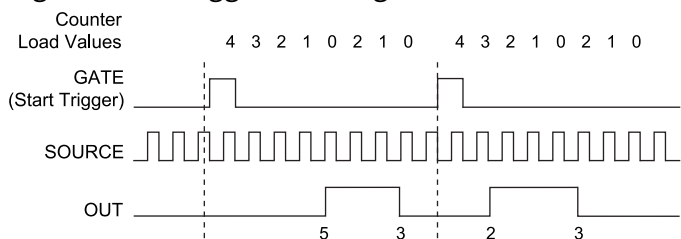
The following figure shows a generation of two pulses with a pulse delay of five and a pulse width of three (using the rising edge of Source) with **CO.EnableInitialDelayOnRetrigger** set to the default True.

Figure 65. Retriggerable Single Pulse Generation with Initial Delay on Retrigger



The following figure shows the same pulse train with **CO.EnableInitialDelayOnRetrigger** set to the default False.

Figure 66. 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 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.

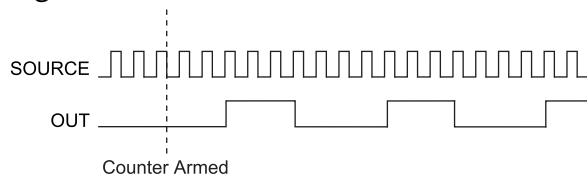
You specify the high and low pulse widths of the output signal. The pulse widths are also measured in terms of a number of active edges of the Source input. You also can specify the active edge of the Source input (rising or falling).

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.

The following figure shows a continuous pulse train generation (using the rising edge of Source).

Figure 67. 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 section.

Buffered Pulse Train Generation

The cDAQ chassis 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

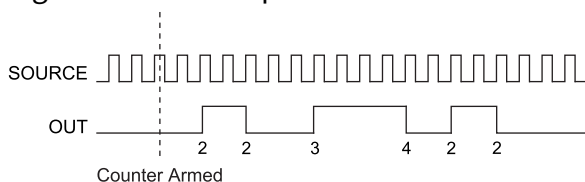
This function generates a predetermined number of pulses with variable idle and active times. Each point you write generates a single pulse. The number of pairs of idle and active times (pulse specifications) you write determines the number of pulses generated. All points are generated back to back to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

The following table and figure detail a finite implicit generation of three samples.

Table 19. Finite Implicit Buffered Pulse Train Generation

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

Figure 68. 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.



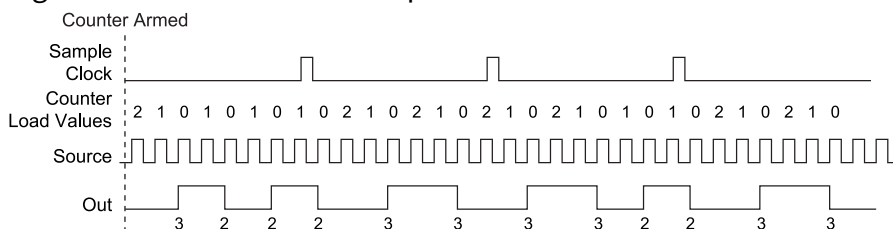
Note When the last sample is generated, the pulse train continues to generate with these specifications until the task is stopped.

The following table and figure 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 20. Finite Buffered Sample Clocked Pulse Train Generation

| Sample | Idle Ticks | Active Ticks |
|--------|------------|--------------|
| 1 | 3 | 3 |
| 2 | 2 | 2 |
| 3 | 3 | 3 |

Figure 69. 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.

Related concepts:

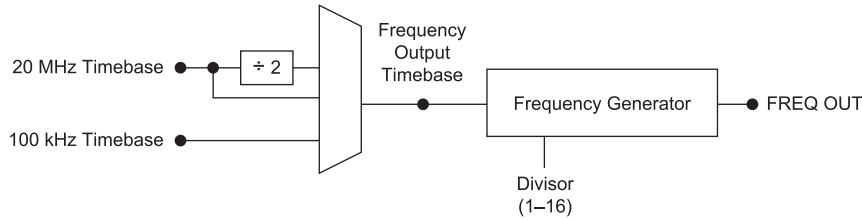
- [Using the Frequency Generator](#)

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 chassis.

The following figure shows a block diagram of the frequency generator.

Figure 70. Frequency Generator Block Diagram

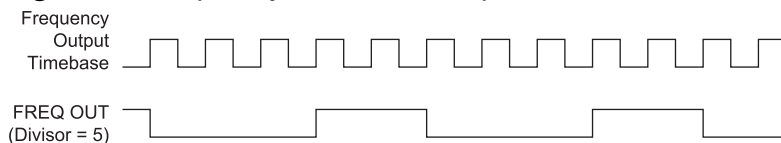


The frequency generator generates the Frequency Output signal. The Frequency Output signal is the Frequency Output Timebase divided by a number you select from 1 to 16. The Frequency Output Timebase can be either the 20 MHz Timebase, the 20 MHz Timebase divided by 2, or the 100 kHz Timebase.

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.

The following figure shows the output waveform of the frequency generator when the divider is set to 5.

Figure 71. Frequency Generator Output Waveform



Frequency Output can be routed out to any PFI terminal on an installed C Series module. All module 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 ***Default Counter/Timer Routing***.

Related concepts:

- [Frequency Generation](#)
- [Default Counter/Timer Routing](#)

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 section for detailed information.

For information about connecting counter signals, refer to the section.

Pulse Generation for ETS

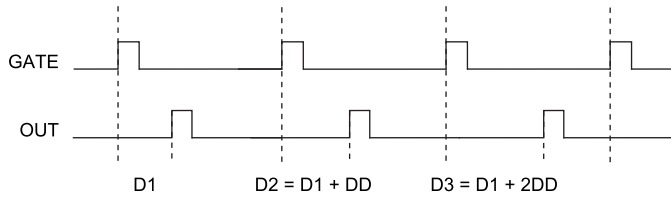
In the equivalent time sampling (ETS) application, the counter produces a pulse on the output a specified delay after an active edge on Gate. After each active edge on Gate, the counter cumulatively increments the delay between the Gate and the pulse on the output by a specified amount. Thus, the delay 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. The following figure shows an example of pulse generation for ETS; the delay from the trigger to the pulse increases after each subsequent Gate active edge.

Figure 72. Pulse Generation for ETS



For information about connecting counter signals, refer to the section.

Counter Timing Signals

The cDAQ chassis features the following counter timing signals:

- [Counter n Source Signal](#)
- [Counter n Gate Signal](#)
- [Counter n Aux Signal](#)
- [Counter n A, Counter n B, and Counter n Z Signals](#)
- [Counter n Up_Down Signal](#)
- [Counter n HW Arm Signal](#)
- [Counter n Sample Clock Signal](#)
- [Counter n Internal Output and Counter n TC Signals](#)
- [Frequency Output Signal](#)

In this section, ***n*** refers to the cDAQ chassis 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 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. The following table lists how this terminal is used in various applications.

Table 21. 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 User Manual*** 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 on an installed C Series module.

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 ***Device Routing in MAX*** in the ***NI-DAQmx User Manual*** 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 on an installed C Series module.

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 ***Device Routing in MAX*** in the ***NI-DAQmx User Manual*** 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 on an installed C Series module.

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:

- 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 **Device Routing in MAX** in the *NI-DAQmx User Manual* 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 chassis 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 User Manual*** for more information.

Using an External Source

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

- 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 on an installed C Series module.

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 on an installed C Series module.

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 on an installed C Series module.

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 User Manual** for more information about how to connect your signals for common counter measurements and generations. Refer to **Physical Channels** in the **NI-DAQmx User Manual** for a list of default PFI lines for counter functions.

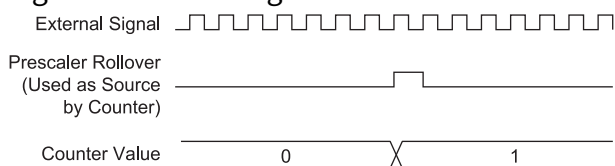
Cascading Counters

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 section.

Prescaling

Prescaling allows the counter to count a signal that is faster than the maximum timebase of the counter. The cDAQ chassis 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 the following figure.

Figure 73. 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

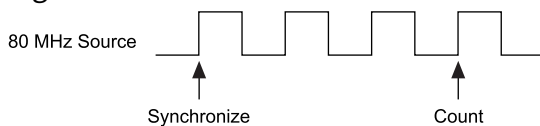
The 32-bit counter counts up or down synchronously with the Source signal. The Gate signal and other counter inputs are asynchronous to the Source signal, so the cDAQ chassis synchronizes these signals before presenting them to the internal counter.

Depending on how you configure your chassis, the cDAQ chassis uses one of two synchronization methods: 80 MHz Source Mode or External or Internal Source Less than 20 MHz.

80 MHz Source Mode

In 80 MHz source mode, the chassis 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 the following figure.

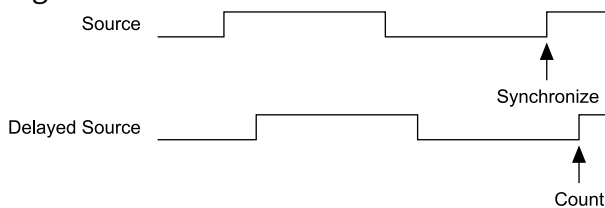
Figure 74. 80 MHz Source Mode



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 chassis synchronizes signals on the rising edge of the delayed Source signal, and counts on the following rising edge of the source, as shown in the following figure.

Figure 75. External or Internal Source Less than 20 MHz



Digital Routing

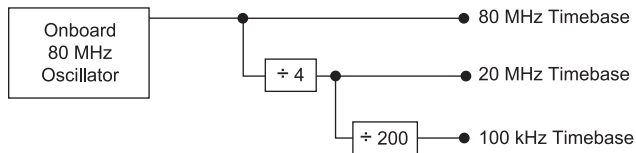
The digital routing circuitry has the following functions:

- 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 chassis. To determine the signal routing options for C Series module(s) installed in the cDAQ chassis, refer to the **Device Routes** tab in MAX.

Clock Routing

The following figure shows the clock routing circuitry of the cDAQ chassis.

Figure 76. 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. The 80 MHz Timebase can be generated from the onboard oscillator.

20 MHz and 100 kHz Timebases

The 20 MHz and 100 kHz Timebases can be used to generate many of the analog input and analog output timing signals. They can function as the Source input to the 32-bit general-purpose counter/timers. The 20 MHz and 100 kHz Timebases are generated by dividing down the 80 MHz Timebase, as shown in the **Clock Routing** topic.

Maintenance

If you need to clean the chassis, wipe it with a dry towel.