

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Visit our website - *Click HERE*

DAQ M Series

NI 6236 User Manual

Isolated Current Input/Voltage Output Devices

May 2006
371948A-01

NATIONAL INSTRUMENTS™

Worldwide Technical Support and Product Information

ni.com

National Instruments Corporate Headquarters

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

Worldwide Offices

Australia 1800 300 800, Austria 43 0 662 45 79 90 0, Belgium 32 0 2 757 00 20, Brazil 55 11 3262 3599, Canada 800 433 3488, China 86 21 6555 7838, Czech Republic 420 224 235 774, Denmark 45 45 76 26 00, Finland 385 0 9 725 725 11, France 33 0 1 48 14 24 24, Germany 49 0 89 741 31 30, India 91 80 41190000, Israel 972 0 3 6393737, Italy 39 02 413091, Japan 81 3 5472 2970, Korea 82 02 3451 3400, Lebanon 961 0 1 33 28 28, Malaysia 1800 887710, Mexico 01 800 010 0793, Netherlands 31 0 348 433 466, New Zealand 0800 553 322, Norway 47 0 66 90 76 60, Poland 48 22 3390150, Portugal 351 210 311 210, Russia 7 095 783 68 51, Singapore 1800 226 5886, Slovenia 386 3 425 4200, South Africa 27 0 11 805 8197, Spain 34 91 640 0085, Sweden 46 0 8 587 895 00, Switzerland 41 56 200 51 51, Taiwan 886 02 2377 2222, Thailand 662 278 6777, United Kingdom 44 0 1635 523545

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

© 2006 National Instruments Corporation. All rights reserved.

Important Information

Warranty

The NI 6236 is warranted against defects in materials and workmanship for a period of three years from the date of shipment, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace equipment that proves to be defective during the warranty period. This warranty includes parts and labor.

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

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

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

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

Copyright

Under the copyright laws, this publication may not be reproduced or transmitted in any form, electronic or mechanical, including photocopying, recording, storing in an information retrieval system, or translating, in whole or in part, without the prior written consent of National Instruments Corporation.

National Instruments respects the intellectual property of others, and we ask our users to do the same. NI software is protected by copyright and other intellectual property laws. Where NI software may be used to reproduce software or other materials belonging to others, you may use NI software only to reproduce materials that you may reproduce in accordance with the terms of any applicable license or other legal restriction.

Trademarks

National Instruments, NI, ni.com, and LabVIEW are trademarks of National Instruments Corporation. Refer to the *Terms of Use* section on `ni.com/legal` for more information about National Instruments trademarks.

FireWire® is the registered trademark of Apple Computer, Inc. Other product and company names mentioned herein are trademarks or trade names of their respective companies.

Members of the National Instruments Alliance Partner Program are business entities independent from National Instruments and have no agency, partnership, or joint-venture relationship with National Instruments.

Patents

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

WARNING REGARDING USE OF NATIONAL INSTRUMENTS PRODUCTS

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

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

Contents

About This Manual

Conventions xiii
Related Documentation xiv
NI-DAQ xiv
NI-DAQmx for Linux xiv
NI-DAQmx Base xv
LabVIEW xv
LabWindows™/CVI™ xvi
Measurement Studio xvi
ANSI C without NI Application Software xvi
.NET Languages without NI Application Software xvi
Device Documentation and Specifications xvii
Training Courses xvii
Technical Support on the Web xvii

Chapter 1
Getting Started

Installing NI-DAQmx 1-1
Installing Other Software 1-1
Installing the Hardware 1-1
Device Pinouts 1-1
Device Specifications 1-2
Device Accessories and Cables 1-2

Chapter 2
DAQ System Overview

DAQ Hardware 2-1
DAQ-STC2 2-2
Calibration Circuitry 2-3
Sensors and Transducers 2-3
Cables and Accessories 2-4
Custom Cabling 2-4
Programming Devices in Software 2-5

Chapter 3
Connector Information

I/O Connector Signal Descriptions 3-1
RTSI Connector Pinout 3-2

Chapter 4
Analog Input

Analog Input Circuitry 4-1
Analog Input Range 4-2
Connecting Analog Current Input Signals 4-3
Method 1 4-3
Method 2 4-4
Analog Input Ground-Reference Settings 4-5
Configuring AI Ground-Reference Settings in Software 4-7
Multichannel Scanning Considerations 4-8
Use Short High-Quality Cabling 4-8
Minimize Current Step between Adjacent Channels 4-8
Avoid Scanning Faster Than Necessary 4-9
Example 1 4-9
Example 2 4-9
Analog Input Data Acquisition Methods 4-10
Software-Timed Acquisitions 4-10
Hardware-Timed Acquisitions 4-10
Buffered 4-10
Non-Buffered 4-11
Analog Input Triggering 4-11
Field Wiring Considerations 4-11
Analog Input Timing Signals 4-12
AI Sample Clock Signal 4-15
Using an Internal Source 4-15
Using an External Source 4-15
Routing AI Sample Clock Signal to an Output Terminal 4-16
Other Timing Requirements 4-16
AI Sample Clock Timebase Signal 4-17
AI Convert Clock Signal 4-17
Using an Internal Source 4-18
Using an External Source 4-18
Routing AI Convert Clock Signal to an Output Terminal 4-18
Using a Delay from Sample Clock to Convert Clock 4-19
Other Timing Requirements 4-19
AI Convert Clock Timebase Signal 4-20
AI Hold Complete Event Signal 4-20
AI Start Trigger Signal 4-21
Using a Digital Source 4-21
Routing AI Start Trigger to an Output Terminal 4-21
AI Reference Trigger Signal 4-22
Using a Digital Source 4-23
Routing AI Reference Trigger Signal to an Output Terminal 4-23

AI Pause Trigger Signal 4-23
Using a Digital Source 4-23
Routing AI Pause Trigger Signal to an Output Terminal 4-24
Getting Started with AI Applications in Software 4-24

Chapter 5
Analog Output

Analog Output Circuitry 5-1
Minimizing Glitches on the Output Signal 5-2
Analog Output Data Generation Methods 5-2
Software-Timed Generations 5-2
Hardware-Timed Generations 5-3
Non-Buffered 5-3
Buffered 5-3
Analog Output Triggering 5-4
Connecting Analog Voltage Output Signals 5-4
Analog Output Timing Signals 5-5
AO Start Trigger Signal 5-6
Using a Digital Source 5-6
Routing AO Start Trigger Signal to an Output Terminal 5-7
AO Pause Trigger Signal 5-7
Using a Digital Source 5-8
Routing AO Pause Trigger Signal to an Output Terminal 5-8
AO Sample Clock Signal 5-8
Using an Internal Source 5-9
Using an External Source 5-9
Routing AO Sample Clock Signal to an Output Terminal 5-9
Other Timing Requirements 5-9
AO Sample Clock Timebase Signal 5-10
Getting Started with AO Applications in Software 5-11

Chapter 6
Digital Input and Output

I/O Protection 6-1
Programmable Power-Up States 6-2
Connecting Digital I/O Signals 6-2
Getting Started with DIO Applications in Software 6-3

Chapter 7
Counters

Counter Input Applications 7-2
Counting Edges 7-2
Single Point (On-Demand) Edge Counting 7-2
Buffered (Sample Clock) Edge Counting 7-3
Non-Cumulative Buffered Edge Counting 7-4
Controlling the Direction of Counting 7-4
Pulse-Width Measurement 7-5
Single Pulse-Width Measurement 7-5
Buffered Pulse-Width Measurement 7-5
Period Measurement 7-6
Single Period Measurement 7-7
Buffered Period Measurement 7-7
Semi-Period Measurement 7-8
Single Semi-Period Measurement 7-8
Buffered Semi-Period Measurement 7-9
Frequency Measurement 7-9
Method 1—Measure Low Frequency with One Counter 7-9
Method 1b—Measure Low Frequency with One Counter (Averaged) 7-10
Method 2—Measure High Frequency with Two Counters 7-11
Method 3—Measure Large Range of Frequencies Using Two Counters 7-12
Choosing a Method for Measuring Frequency 7-13
Position Measurement 7-15
Measurements Using Quadrature Encoders 7-15
Measurements Using Two Pulse Encoders 7-17
Two-Signal Edge-Separation Measurement 7-18
Single Two-Signal Edge-Separation Measurement 7-18
Buffered Two-Signal Edge-Separation Measurement 7-19
Counter Output Applications 7-20
Simple Pulse Generation 7-20
Single Pulse Generation 7-20
Single Pulse Generation with Start Trigger 7-20
Retriggerable Single Pulse Generation 7-21
Pulse Train Generation 7-22
Continuous Pulse Train Generation 7-22
Frequency Generation 7-23
Using the Frequency Generator 7-23
Frequency Division 7-24
Pulse Generation for ETS 7-24

Counter Timing Signals 7-25
Counter *n* Source Signal 7-25
Routing a Signal to Counter *n* Source 7-26
Routing Counter *n* Source to an Output Terminal 7-26
Counter *n* Gate Signal 7-27
Routing a Signal to Counter *n* Gate 7-27
Routing Counter *n* Gate to an Output Terminal 7-27
Counter *n* Aux Signal 7-27
Routing a Signal to Counter *n* Aux 7-27
Counter *n* A, Counter *n* B, and Counter *n* Z Signals 7-28
Routing Signals to A, B, and Z Counter Inputs 7-28
Routing Counter *n* Z Signal to an Output Terminal 7-28
Counter *n* Up_Down Signal 7-28
Counter *n* HW Arm Signal 7-28
Routing Signals to Counter *n* HW Arm Input 7-29
Counter *n* Internal Output and Counter *n* TC Signals 7-29
Routing Counter *n* Internal Output to an Output Terminal 7-29
Frequency Output Signal 7-29
Routing Frequency Output to a Terminal 7-29
Default Counter Terminals 7-30
Counter Triggering 7-31
Arm Start Trigger 7-31
Start Trigger 7-31
Pause Trigger 7-31
Other Counter Features 7-32
Cascading Counters 7-32
Counter Filters 7-32
Prescaling 7-33
Duplicate Count Prevention 7-34
Example Application That Works Correctly (No Duplicate Counting) 7-34
Example Application That Works Incorrectly (Duplicate Counting) 7-35
Example Application That Prevents Duplicate Count 7-35
When To Use Duplicate Count Prevention 7-36
Enabling Duplicate Count Prevention in NI-DAQmx 7-37
Synchronization Modes 7-37
80 MHz Source Mode 7-38
Other Internal Source Mode 7-38
External Source Mode 7-38

Chapter 8
PFI

Using PFI Terminals as Timing Input Signals 8-2
Exporting Timing Output Signals Using PFI Terminals 8-3
Using PFI Terminals as Static Digital Inputs and Outputs 8-3
Connecting PFI Input Signals 8-3
PFI Filters 8-4
I/O Protection 8-6
Programmable Power-Up States 8-6

Chapter 9
Isolation and Digital Isolators

Digital Isolation 9-2
Benefits of an Isolated DAQ Device 9-2
Reducing Common-Mode Noise 9-2
    Creating an AC Return Path 9-3
        Non-Isolated Systems 9-3
        Isolated Systems 9-3

Chapter 10
Digital Routing and Clock Generation

Clock Routing 10-1
    80 MHz Timebase 10-2
    20 MHz Timebase 10-2
    100 kHz Timebase 10-2
    External Reference Clock 10-2
    10 MHz Reference Clock 10-3
Synchronizing Multiple Devices 10-3
Real-Time System Integration Bus (RTSI) 10-3
    RTSI Connector Pinout 10-4
    Using RTSI as Outputs 10-5
    Using RTSI Terminals as Timing Input Signals 10-6
    RTSI Filters 10-6
PXI Clock and Trigger Signals 10-8
    PXI_CLK10 10-8
    PXI Triggers 10-8
    PXI_STAR Trigger 10-8
    PXI_STAR Filters 10-9

Chapter 11
Bus Interface

DMA Controllers 11-1
PXI Considerations 11-2
PXI Clock and Trigger Signals 11-2
PXI and PXI Express 11-2
Using PXI with CompactPCI 11-3
Data Transfer Methods 11-3
Direct Memory Access (DMA) 11-3
Interrupt Request (IRQ) 11-4
Programmed I/O 11-4
Changing Data Transfer Methods between DMA and IRQ 11-4

Chapter 12
Triggering

Triggering with a Digital Source 12-1

Appendix A
NI 6236 Device Information

NI 6236 Pinout A-1
NI 6236 Specifications A-3
NI 6236 Accessory and Cabling Options A-3

Appendix B
Troubleshooting

Analog Input B-1
Analog Output B-2
Counters B-3

Appendix C
Technical Support and Professional Services

Glossary

Index

About This Manual

The *NI 6236 User Manual* contains information about using the NI 6236 M Series data acquisition (DAQ) devices with NI-DAQ 8.1 and later. National Instruments 6236 devices feature four analog current input (AI) channels, four analog voltage output (AO) channels, two counters, six lines of digital input (DI), and four lines of digital output (DO).

Conventions

The following conventions appear in this manual:

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

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

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

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

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

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

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

Related Documentation

Each application software package and driver includes information about writing applications for taking measurements and controlling measurement devices. The following references to documents assume you have NI-DAQ 8.1 or later, and where applicable, version 7.0 or later of the NI application software.

NI-DAQ

The *DAQ Getting Started Guide* describes how to install your NI-DAQmx for Windows software, your NI-DAQmx-supported DAQ device, and how to confirm that your device is operating properly. Select **Start»All Programs»National Instruments»NI-DAQ»DAQ Getting Started Guide**.

The *NI-DAQ Readme* lists which devices are supported by this version of NI-DAQ. Select **Start»All Programs»National Instruments»NI-DAQ» NI-DAQ Readme**.

The *NI-DAQmx Help* contains general information about measurement concepts, key NI-DAQmx concepts, and common applications that are applicable to all programming environments. Select **Start»All Programs» National Instruments»NI-DAQ»NI-DAQmx Help**.

NI-DAQmx for Linux

The *DAQ Getting Started Guide* describes how to install your NI-DAQmx-supported DAQ device and confirm that your device is operating properly.

The *NI-DAQ Readme for Linux* lists supported devices and includes software installation instructions, frequently asked questions, and known issues.

The *C Function Reference Help* describes functions and attributes.

The *NI-DAQmx for Linux Configuration Guide* provides configuration instructions, templates, and instructions for using test panels.



Note All NI-DAQmx documentation for Linux is installed at `/usr/local/natinst/nidaqmx/docs`.

NI-DAQmx Base

The *NI-DAQmx Base Getting Started Guide* describes how to install your NI-DAQmx Base software, your NI-DAQmx Base-supported DAQ device, and how to confirm that your device is operating properly. Select **Start»All Programs»National Instruments»NI-DAQmx Base»Documentation»Getting Started Guide**.

The *NI-DAQmx Base Readme* lists which devices are supported by this version of NI-DAQmx Base. Select **Start»All Programs»National Instruments»NI-DAQmx Base»Documentation»Readme**.

The *NI-DAQmx Base VI Reference Help* contains VI reference and general information about measurement concepts. In LabVIEW, select **Help»NI-DAQmx Base VI Reference Help**.

The *NI-DAQmx Base C Reference Help* contains C reference and general information about measurement concepts. Select **Start»All Programs»National Instruments»NI-DAQmx Base»Documentation»C Function Reference Manual**.

LabVIEW

If you are a new user, use the *Getting Started with LabVIEW* manual to familiarize yourself with the LabVIEW graphical programming environment and the basic LabVIEW features you use to build data acquisition and instrument control applications. Open the *Getting Started with LabVIEW* manual by selecting **Start»All Programs»National Instruments»LabVIEW»LabVIEW Manuals** or by navigating to the `labview\manuals directory` and opening `LV_Getting_Started.pdf`.

Use the *LabVIEW Help*, available by selecting **Help»Search the LabVIEW Help** in LabVIEW, to access information about LabVIEW programming concepts, step-by-step instructions for using LabVIEW, and reference information about LabVIEW VIs, functions, palettes, menus, and tools. Refer to the following locations on the **Contents** tab of the *LabVIEW Help* for information about NI-DAQmx:

- **Getting Started»Getting Started with DAQ**—Includes overview information and a tutorial to learn how to take an NI-DAQmx measurement in LabVIEW using the DAQ Assistant.
- **VI and Function Reference»Measurement I/O VIs and Functions**—Describes the LabVIEW NI-DAQmx VIs and properties.

- **Taking Measurements**—Contains the conceptual and how-to information you need to acquire and analyze measurement data in LabVIEW, including common measurements, measurement fundamentals, NI-DAQmx key concepts, and device considerations.

LabWindows™/CVI™

The **Data Acquisition** book of the *LabWindows/CVI Help* contains measurement concepts for NI-DAQmx. This book also contains *Taking an NI-DAQmx Measurement in LabWindows/CVI*, which includes step-by-step instructions about creating a measurement task using the DAQ Assistant. In LabWindows/CVI, select **Help»Contents**, then select **Using LabWindows/CVI»Data Acquisition**.

The **NI-DAQmx Library** book of the *LabWindows/CVI Help* contains API overviews and function reference for NI-DAQmx. Select **Library Reference»NI-DAQmx Library** in the *LabWindows/CVI Help*.

Measurement Studio

The *NI Measurement Studio Help* contains function reference, measurement concepts, and a walkthrough for using the Measurement Studio NI-DAQmx .NET and Visual C++ class libraries. This help collection is integrated into the Microsoft Visual Studio .NET documentation. In Visual Studio .NET, select **Help»Contents**.

Note You must have Visual Studio .NET installed to view the *NI Measurement Studio Help*.

ANSI C without NI Application Software

The *NI-DAQmx Help* contains API overviews and general information about measurement concepts. Select **Start»All Programs»National Instruments»NI-DAQmx Help**.

.NET Languages without NI Application Software

The *NI Measurement Studio Help* contains function reference and measurement concepts for using the Measurement Studio NI-DAQmx .NET and Visual C++ class libraries. This help collection is integrated into the Visual Studio .NET documentation. In Visual Studio .NET, select **Help»Contents**.

Note You must have Visual Studio .NET installed to view the *NI Measurement Studio Help*.

Device Documentation and Specifications

The *NI 6236 Specifications* contains all specifications for NI 6236 M Series devices.

NI-DAQ 7.0 and later includes the Device Document Browser, which contains online documentation for supported DAQ, SCXI, and switch devices, such as help files describing device pinouts, features, and operation, and PDF files of the printed device documents. You can find, view, and/or print the documents for each device using the Device Document Browser at any time by inserting the CD. After installing the Device Document Browser, device documents are accessible from **Start»All Programs»National Instruments»NI-DAQ»Browse Device Documentation**.

Training Courses

If you need more help getting started developing an application with NI products, NI offers training courses. To enroll in a course or obtain a detailed course outline, refer to `ni.com/training`.

Technical Support on the Web

For additional support, refer to `ni.com/support` or `zone.ni.com`.

Note You can download these documents at `ni.com/manuals`.

DAQ specifications and some DAQ manuals are available as PDFs. You must have Adobe Acrobat Reader with Search and Accessibility 5.0.5 or later installed to view the PDFs. Refer to the Adobe Systems Incorporated Web site at `www.adobe.com` to download Acrobat Reader. Refer to the National Instruments Product Manuals Library at `ni.com/manuals` for updated documentation resources.

1 Getting Started

M Series NI 6236 devices feature four analog current input (AI) channels, four analog voltage output (AO) channels, two counters, six lines of digital input (DI), and four lines of digital output (DO). If you have not already installed your device, refer to the *DAQ Getting Started Guide*. For NI 6236 device specifications, refer to the *NI 6236 Specifications* on `ni.com/manuals`.

Before installing your DAQ device, you must install the software you plan to use with the device.

Installing NI-DAQmx

The *DAQ Getting Started Guide*, which you can download at `ni.com/manuals`, offers NI-DAQmx users step-by-step instructions for installing software and hardware, configuring channels and tasks, and getting started developing an application.

Installing Other Software

If you are using other software, refer to the installation instructions that accompany your software.

Installing the Hardware

The *DAQ Getting Started Guide* contains non-software-specific information about how to install PCI and PXI devices, as well as accessories and cables.

Device Pinouts

Refer to Appendix A, *NI 6236 Device Information*, for the NI 6236 device pinout.

Device Specifications

Refer to the *NI 6236 Specifications*, available on the NI-DAQ Device Document Browser or `ni.com/manuals`, for more detailed information about NI 6236 devices.

Device Accessories and Cables

NI offers a variety of accessories and cables to use with your DAQ device. Refer to Appendix A, *NI 6236 Device Information*, or `ni.com` for more information.

2

DAQ System Overview

Figure 2-1 shows a typical DAQ system, which includes sensors, transducers, cables that connect the various devices to the accessories, the M Series device, programming software, and PC. The following sections cover the components of a typical DAQ system.

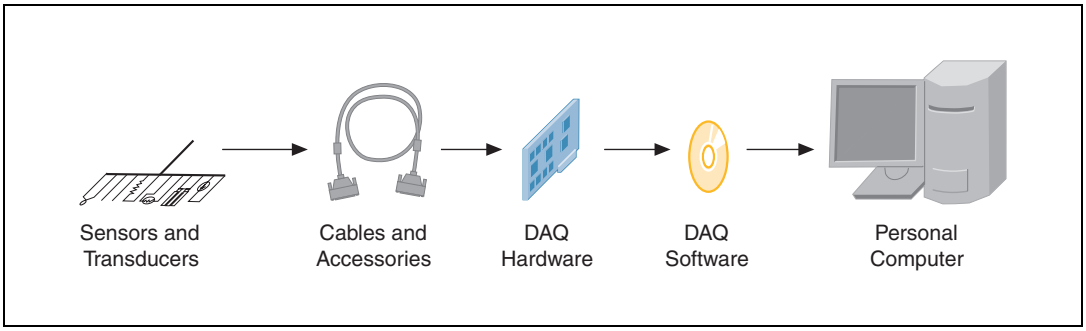


Figure 2-1. Components of a Typical DAQ System

DAQ Hardware

DAQ hardware digitizes input signals, performs D/A conversions to generate analog output signals, and measures and controls digital I/O signals. Figure 2-2 features the components of NI 6236 devices.

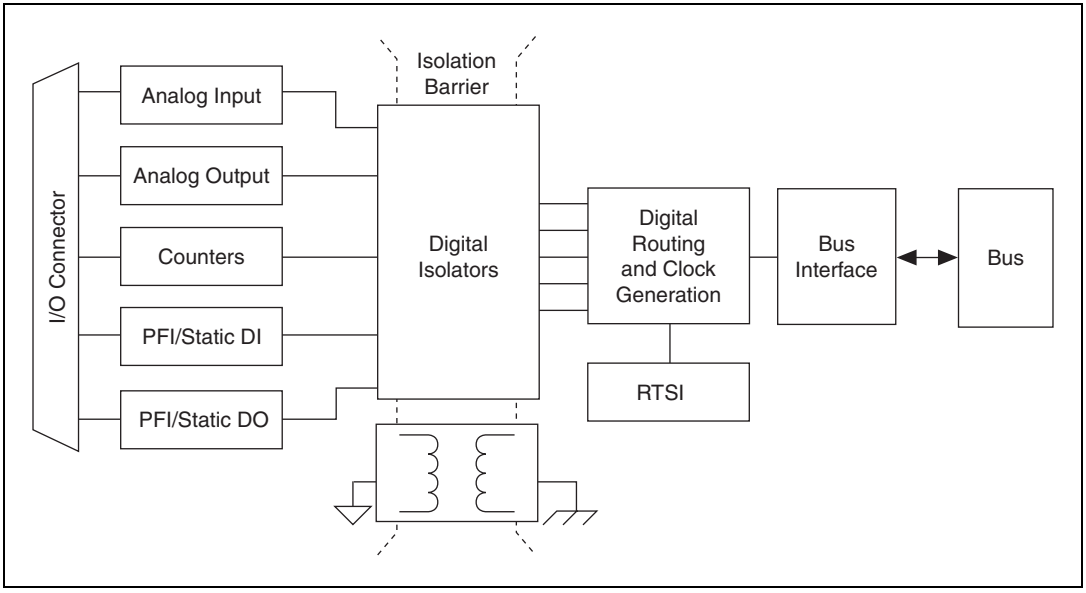


Figure 2-2. General NI 6236 Block Diagram

DAQ-STC2

The DAQ-STC2 implements a high-performance digital engine for NI 6236 data acquisition hardware. Some key features of this engine include the following:

- Flexible AI and AO sample and convert timing
- Many triggering modes
- Independent AI and AO FIFOs
- Generation and routing of RTSI signals for multi-device synchronization
- Generation and routing of internal and external timing signals
- Two flexible 32-bit counter/timer modules with hardware gating
- Static DIO signals
- PLL for clock synchronization
- PCI/PXI interface
- Independent scatter-gather DMA controllers for all acquisition and generation functions

Calibration Circuitry

The M Series analog inputs and outputs can self-calibrate to correct gain and offset errors. You can calibrate the device to minimize AI and AO errors caused by time and temperature drift at run time. No external circuitry is necessary; an internal reference ensures high accuracy and stability over time and temperature changes.

Factory-calibration constants are permanently stored in an onboard EEPROM and cannot be modified. When you self-calibrate the device, software stores new constants in a user-modifiable section of the EEPROM. To return a device to its initial factory calibration settings, software can copy the factory-calibration constants to the user-modifiable section of the EEPROM. Refer to the *NI-DAQmx Help* or the *LabVIEW 8.x Help* for more information about using calibration constants.

Sensors and Transducers

Sensors can generate electrical signals to measure physical phenomena, such as temperature, force, sound, or light. Some commonly used sensors are strain gauges, thermocouples, thermistors, angular encoders, linear encoders, and resistance temperature detectors (RTDs).

Note Current input measurement devices can only interface with sensors that output a current.

To measure signals from these various transducers, you must convert them into a form that a DAQ device can accept. For example, the output voltage of most thermocouples is very small and susceptible to noise. Therefore, you may need to amplify or filter the thermocouple output before digitizing it, or use the smallest measurement range available within the DAQ device.

For more information about sensors, refer to the following documents.

- For general information about sensors, visit `ni.com/sensors`.
- If you are using LabVIEW, refer to the *LabVIEW Help* by selecting **Help»Search the LabVIEW Help** in LabVIEW, and then navigate to the **Taking Measurements** book on the **Contents** tab.
- If you are using other application software, refer to **Common Sensors** in the *NI-DAQmx Help*, which can be accessed from **Start»All Programs»National Instruments»NI-DAQ»NI-DAQmx Help**.

Cables and Accessories

NI offers a variety of products to use with NI 6236 devices, including cables, connector blocks, and other accessories, as follows:

- Cables and cable assemblies
  - Shielded
  - Unshielded ribbon
- Screw terminal connector blocks, shielded and unshielded
- RTSI bus cables

For more specific information about these products, refer to `ni.com`.

Refer to the *Custom Cabling* section of this chapter, the *Field Wiring Considerations* section of Chapter 4, *Analog Input*, and Appendix A, *NI 6236 Device Information*, for information about how to select accessories for your M Series device.

Custom Cabling

NI offers cables and accessories for many applications. However, if you want to develop your own cable, the following kits can assist you:

- **TB-37F-37SC**—37-pin solder cup terminals, shell with strain relief
- **TB-37F-37CP**—37-pin crimp & poke terminals, shell with strain relief

Also, adhere to the following guidelines for best results:

- For AI signals, use shielded, twisted-pair wires for each AI pair of differential inputs. Connect the shield for each signal pair to the ground reference at the source.
- Route the analog lines separately from the digital lines.
- When using a cable shield, use separate shields for the analog and digital sections of the cable. Failure to do so results in noise coupling into the analog signals from transient digital signals.

For more information about the connectors used for DAQ devices, refer to the KnowledgeBase document, *Specifications and Manufacturers for Board Mating Connectors*, by going to `ni.com/info` and entering the info code `rdspmb`.

Programming Devices in Software

National Instruments measurement devices are packaged with NI-DAQ driver software, an extensive library of functions and VIs you can call from your application software, such as LabVIEW or LabWindows/CVI, to program all the features of your NI measurement devices. Driver software has an application programming interface (API), which is a library of VIs, functions, classes, attributes, and properties for creating applications for your device.

NI-DAQ includes two NI-DAQ drivers, Traditional NI-DAQ (Legacy) and NI-DAQmx. M Series devices use the NI-DAQmx driver. Each driver has its own API, hardware configuration, and software configuration. Refer to the *DAQ Getting Started Guide* for more information about the two drivers.

NI-DAQmx includes a collection of programming examples to help you get started developing an application. You can modify example code and save it in an application. You can use examples to develop a new application or add example code to an existing application.

To locate LabVIEW and LabWindows/CVI examples, open the National Instruments Example Finder.

- In LabVIEW, select **Help»Find Examples**.
- In LabWindows/CVI, select **Help»NI Example Finder**.

Measurement Studio, Visual Basic, and ANSI C examples are located in the following directories:

- NI-DAQmx examples for Measurement Studio-supported languages are in the following directories:
  - `MeasurementStudio\VCNET\Examples\NIDaq`
  - `MeasurementStudio\DotNET\Examples\NIDaq`
- NI-DAQmx examples for ANSI C are in the `NI-DAQ\Examples\DAQmx ANSI C Dev` directory

For additional examples, refer to `zone.ni.com`.

3

Connector Information

The *I/O Connector Signal Descriptions* and *RTSI Connector Pinout* sections contain information about M Series connectors. Refer to Appendix A, *NI 6236 Device Information*, for device I/O connector pinouts.

I/O Connector Signal Descriptions

Table 3-1 describes the signals found on the I/O connectors. Not all signals are available on all devices.

Table 3-1. I/O Connector Signals

Signal Name	Reference	Direction	Description
AI GND	—	—	**Analog Input Ground**—These terminals are the input bias current return point. AI GND, AO GND, and D GND are connected on the device. **Note:** AI GND, AO GND, and D GND are isolated from earth ground and chassis ground.
AI <0..3>±	AI GND	Input	**Analog Input Channels 0 to 3±**—For differential measurements, AI 0+ and AI 0– are the positive and negative inputs of differential analog input channel 0. Similarly, the following signal pairs also form differential input channels: <AI 1+, AI 1–>, <AI 2+, AI 2–>, <AI 3+, AI 3–> Also refer to the *Analog Input Ground-Reference Settings* section of Chapter 4, *Analog Input*. **Note:** AI <0..3>± are isolated from earth ground and chassis ground.
AO <0..3>	AO GND	Output	**Analog Output Channels 0 to 3**—These terminals supply the voltage output of AO channels 0 to 3. **Note:** AO <0..3> are isolated from earth ground and chassis ground.

Table 3-1. I/O Connector Signals (Continued)

Signal Name	Reference	Direction	Description
AO GND	—	—	**Analog Output Ground**—AO GND is the reference for AO <0..3>. All three ground references—AI GND, AO GND, and D GND—are connected on the device. **Note:** AI GND, AO GND, and D GND are isolated from earth ground and chassis ground.
D GND	—	—	**Digital Ground**—D GND supplies the reference for input PFI <0..5>/P0.<0..5> and output PFI <6..9>/P1.<0..3>. All three ground references—AI GND, AO GND, and D GND—are connected on the device. **Note:** AI GND, AO GND, and D GND are isolated from earth ground and chassis ground.
PFI <0..5>/P0.<0..5>	D GND	Input	**Programmable Function Interface or Static Digital Input Channels 0 to 5**—Each of these terminals can be individually configured as a PFI terminal or a digital input terminal. As an input, each PFI terminal can be used to supply an external source for AI or AO timing signals or counter/timer inputs. **Note:** PFI <0..5>/P0.<0..5> are isolated from earth ground and chassis ground.
PFI <6..9>/P1.<0..3>	D GND	Output	**Programmable Function Interface or Static Digital Output Channels 6 to 9**—Each of these terminals can be individually configured as a PFI terminal or a digital output terminal. As a PFI output, you can route many different internal AI or AO timing signals to each PFI terminal. You also can route the counter/timer outputs to each PFI terminal. **Note:** PFI <6..9>/P1.<0..3> are isolated from earth ground and chassis ground.
NC	—	—	**No connect**—Do not connect signals to these terminals.
CAL+	—	—	**External Calibration Positive Reference**—CAL+ supplies the positive reference during external calibration of the NI 6236.
CAL–	—	—	**External Calibration Negative Reference**—CAL– supplies the negative reference during external calibration of the NI 6236.

RTSI Connector Pinout

Refer to the *RTSI Connector Pinout* section of Chapter 10, *Digital Routing and Clock Generation*, for information about the RTSI connector.

4

Analog Input

Figure 4-1 shows the analog input circuitry of NI 6236 devices.

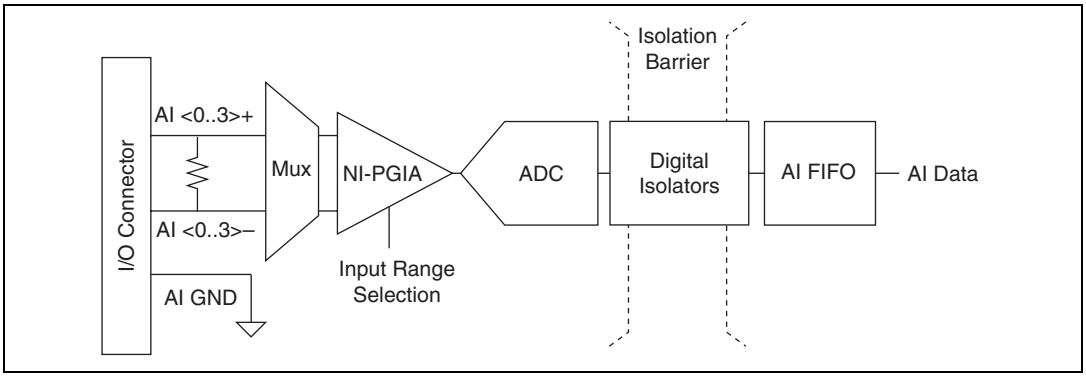


Figure 4-1. NI 6236 Analog Input Circuitry

Analog Input Circuitry

I/O Connector

You can connect analog input signals to the M Series device through the I/O connector. The proper way to connect analog input signals depends on the analog input ground-reference settings, described in the *Analog Input Ground-Reference Settings* section. Also refer to Appendix A, *NI 6236 Device Information*, for device I/O connector pinouts.

MUX

Each M Series device has one analog-to-digital converter (ADC). The multiplexers (MUX) route one AI channel at a time to the ADC through the NI-PGIA.

Instrumentation Amplifier (NI-PGIA)

The NI programmable gain instrumentation amplifier (PGIA) is a measurement and instrument class amplifier that minimizes settling times for all input ranges. The NI-PGIA can amplify or attenuate an AI signal to ensure that you use the maximum resolution of the ADC.

M Series devices use the NI-PGIA to deliver high accuracy even when sampling multiple channels with small input ranges at fast rates. M Series devices can sample channels in any order at the maximum conversion rate, and you can individually program each channel in a sample with a different input range.

A/D Converter

The analog-to-digital converter (ADC) digitizes the AI signal by converting the analog voltage into a digital number.

Isolation Barrier and Digital Isolators

The digital isolators across the isolation barrier provide a ground break between the isolated analog front end and the earth/chassis/building ground.

AI FIFO

M Series devices can perform both single and multiple A/D conversions of a fixed or infinite number of samples. A large first-in-first-out (FIFO) buffer holds data during AI acquisitions to ensure that no data is lost. M Series devices can handle multiple A/D conversion operations with DMA, interrupts, or programmed I/O.

Analog Input Range

Input range refers to the set of input voltages or currents that an analog input channel can digitize with the specified accuracy. The NI-PGIA amplifies or attenuates the AI signal depending on the input range. You can individually program the input range of each AI channel on your M Series device.

The input range affects the resolution of the M Series device for an AI channel. Resolution refers to the voltage or current of one ADC code.

For example, a 16-bit ADC converts analog current inputs into one of 65,536 (= 2^{16}) codes—that is, one of 65,536 possible digital values. These values are spread fairly evenly across the input range. So, for an input range of ±20 mA, the current of each code of a 16-bit ADC is:

$$\frac{(20\text{ mA} - (-20\text{ mA}))}{2^{16}} = 610\text{ nA}$$

M Series devices use a calibration method that requires some codes (typically about 5% of the codes) to lie outside of the specified range. This calibration method improves absolute accuracy, but it increases the nominal resolution of input ranges by about 5% over what the previous formulas would indicate.

Table 4-1 shows the input range and resolution supported by the NI 6236 devices.

Table 4-1. Input Ranges for NI 6236

Input Range	Nominal Resolution Assuming 5% Over Range
±20 mA	640 nA

Connecting Analog Current Input Signals

When making signal connections, caution must be taken with the signal voltage levels going into the device. There are two types of connections that can be made.

Method 1

Method 1, shown in Figure 4-2, connects AI + and AI – inputs at a voltage level with respect to AI GND. Verify that the voltage levels on the AI + and AI – side do not exceed the common-mode input range of ±10 V. *Common-mode input range* is the voltage input range with respect to AI GND (AI + versus AI GND or AI – versus AI GND).

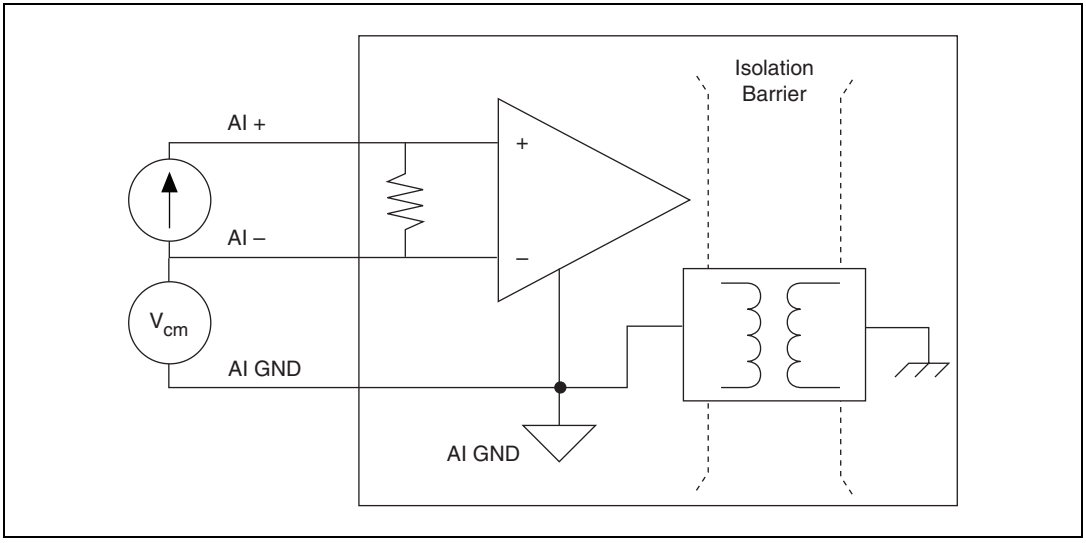


Figure 4-2. Analog Current Input Connection Method 1

Method 2

Method 2, shown in Figure 4-3, ties the AI – input to AI GND. When measuring current up to 20 mA, this type of connection ensures that the voltage level on both the positive and negative side are within the common-mode input range for NI 6236 devices.

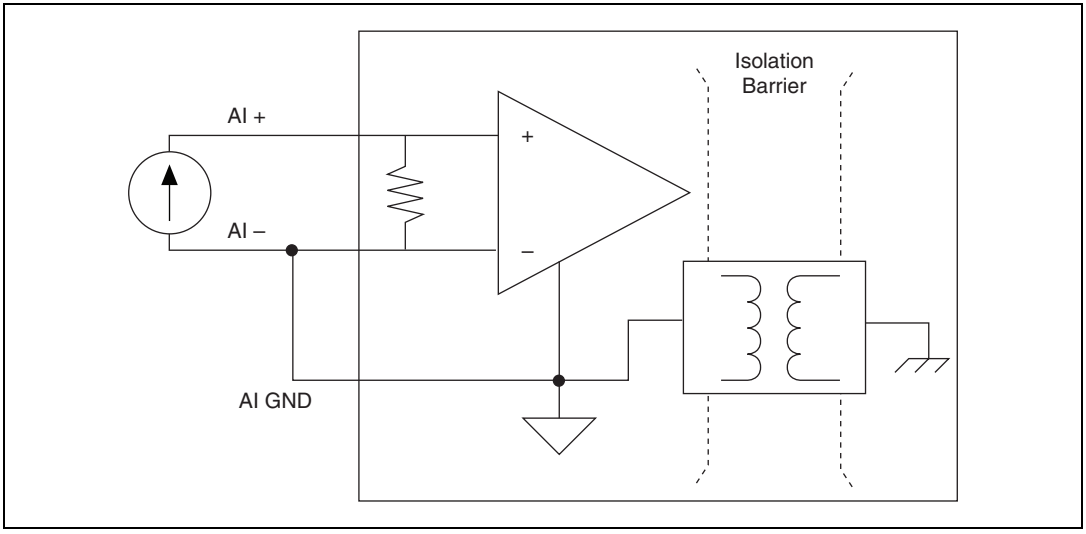


Figure 4-3. Analog Current Input Connection Method 2

Note that AI GND must always be connected to the ground-reference point of the circuit being measured. AI GND is the ground-reference point that NI 6236 devices use to reference their isolated front end.

The NI 6236 is an isolated device with isolation ratings up to 60 VDC/30 Vrms. This allows for current measurement at high voltage levels provided that the common-mode input range requirement is satisfied. For example, in Figure 4-4, the node connected to AI GND can be at 50 V above the earth ground.

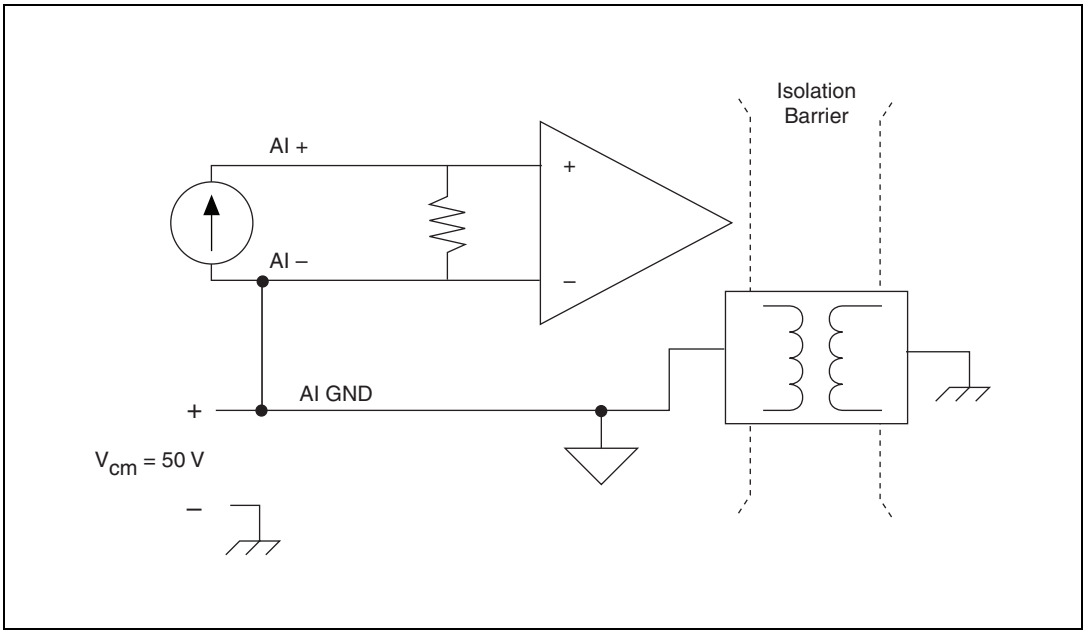


Figure 4-4. Current Measurement at High Voltage Levels

Analog Input Ground-Reference Settings

The NI 6236 device measures the voltage generated across the current input sense resistor during current input measurement. This voltage difference is then routed into the instrumentation amplifier (PGIA) differentially, as shown in Figure 4-5.

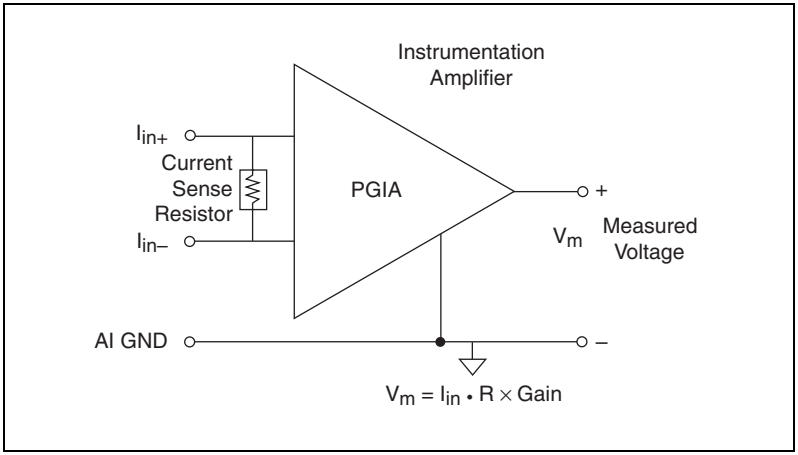


Figure 4-5. NI 6236 PGIA

Analog input ground-reference setting refers to the reference that the PGIA measures against. Differential is the only ground-reference setting for NI 6236 analog input signals, which means that the PGIA always measures the voltages between AI + and AI – generated across the input sense resistor, regardless of which analog input connection method is used. Refer to the *Connecting Analog Current Input Signals* section for allowable input connection methods.

When measuring voltages internal to the device, such as the onboard reference, the signal is measured against AI GND. This measurement method is called *referenced single-ended* (RSE). The name is derived from the fact that one end of the measurement, the positive input of the PGIA, is connected to a signal, and the negative input of the PGIA is connected to the AI GND reference.

Table 4-2 shows how signals are routed to the NI-PGIA.

Table 4-2. Signals Routed to the NI-PGIA

AI Ground-Reference Settings	Signals Routed to the Positive Input of the NI-PGIA (V_{in+})	Signals Routed to the Negative Input of the NI-PGIA (V_{in-})
DIFF	AI <0..3>+	AI <0..3>–
RSE	Internal Channels	AI GND

Note On devices that allow multiple ground-reference settings, the settings are programmed on a per-channel basis.

Caution The maximum input voltage and current ratings of AI signals with respect to AI GND (and for differential signals with respect to each other) and earth/chassis ground are listed in the *Maximum Working Voltage* section of the *NI 6236 Specifications*. Exceeding the maximum input voltage or maximum working voltage of AI signals distorts the measurement results. Exceeding the maximum input voltage or maximum working voltage rating also can damage the device and the computer. Exceeding the maximum input voltage can cause injury and harm the user. NI is *not* liable for any damage or injuries resulting from such signal connections.

AI ground-reference setting is sometimes referred to as *AI terminal configuration*.

Configuring AI Ground-Reference Settings in Software

You can program channels on an M Series device to acquire with different ground references.

To enable multimode scanning in LabVIEW, use `NI-DAQmx Create Virtual Channel.vi` of the NI-DAQmx API. You must use a new VI for each channel or group of channels configured in a different input mode. In Figure 4-6, channel 0 is configured in differential mode and aignd_vs_aignd reads the internal analog input ground-reference of the device.

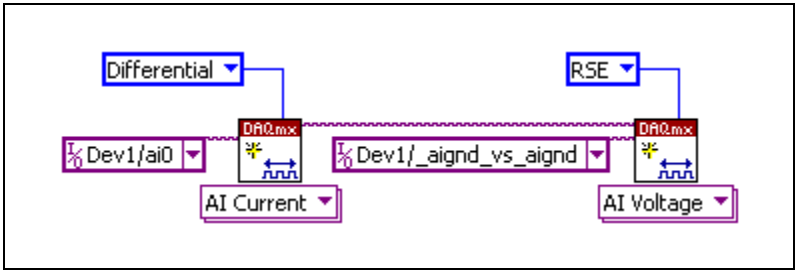


Figure 4-6. Enabling Multimode Scanning in LabVIEW

To configure the input mode of your current measurement using the DAQ Assistant, use the **Terminal Configuration** drop-down list. Refer to the *DAQ Assistant Help* for more information about the DAQ Assistant.

To configure the input mode of your current measurement using the NI-DAQmx C API, set the **terminalConfig** property. Refer to the *NI-DAQmx C Reference Help* for more information.

Multichannel Scanning Considerations

M Series devices can scan multiple channels at high rates and digitize the signals accurately. However, you should consider several issues when designing your measurement system to ensure the high accuracy of your measurements.

In multichannel scanning applications, accuracy is affected by settling time. When your M Series device switches from one AI channel to another AI channel, the device configures the NI-PGIA with the input range of the new channel. The NI-PGIA then amplifies the input signal with the gain for the new input range. Settling time refers to the time it takes the NI-PGIA to amplify the input signal to the desired accuracy before it is sampled by the ADC. The *NI 6236 Specifications* shows the device settling time.

M Series devices are designed to have fast settling times. However several factors can increase the settling time which decreases the accuracy of your measurements. To ensure fast settling times, you should do the following (in order of importance):

- Use short high-quality cabling
- Minimize current step between adjacent channels
- Avoid scanning faster than necessary

Refer to the following sections for more information about these factors.

Use Short High-Quality Cabling

Using short high-quality cables can minimize several effects that degrade accuracy including crosstalk, transmission line effects, and noise. The capacitance of the cable also can increase the settling time.

National Instruments recommends using individually shielded, twisted-pair wires that are 2 m or less to connect AI signals to the device. Refer to the *Connecting Analog Current Input Signals* section for more information.

Minimize Current Step between Adjacent Channels

When scanning between channels that have the same input range, the settling time increases with the current step between the channels. If you know the expected input range of your signals, you can group signals with similar expected ranges together in your scan list.

For example, suppose all channels in a system use a –20 to 20 mA input range. The signals on channels 0 and 2 vary between 18 and 19 mA. The signals on channels 1 and 3 vary between –18 and 0 mA. Scanning channels in the order 0, 2, 1, 3 will produce more accurate results than scanning channels in the order 0, 1, 2, 3.

Avoid Scanning Faster Than Necessary

Designing your system to scan at slower speeds gives the PGIA more time to settle to a more accurate level. Here are two examples to consider.

Example 1

Averaging many AI samples can increase the accuracy of the reading by decreasing noise effects. In general, the more points you average, the more accurate the final result will be. However, you may choose to decrease the number of points you average and slow down the scanning rate.

Suppose you want to sample 10 channels over a period of 40 ms and average the results. You could acquire 500 points from each channel at a scan rate of 125 kS/s. Another method would be to acquire 1,000 points from each channel at a scan rate of 250 kS/s. Both methods take the same amount of time. Doubling the number of samples averaged (from 500 to 1,000) decreases the effect of noise by a factor of 1.4 (the square root of 2). However, doubling the number of samples (in this example) decreases the time the PGIA has to settle from 8 µs to 4 µs. In some cases, the slower scan rate system returns more accurate results.

Example 2

If the time relationship between channels is not critical, you can sample from the same channel multiple times and scan less frequently. For example, suppose an application requires averaging 100 points from channel 0 and averaging 100 points from channel 1. You could alternate reading between channels—that is, read one point from channel 0, then one point from channel 1, and so on. You also could read all 100 points from channel 0 then read 100 points from channel 1. The second method switches between channels much less often and is affected much less by settling time.

Analog Input Data Acquisition Methods

When performing analog input measurements, you either can perform software-timed or hardware-timed acquisitions. Hardware-timed acquisitions can be buffered or non-buffered.

Software-Timed Acquisitions

With a software-timed acquisition, software controls the rate of the acquisition. Software sends a separate command to the hardware to initiate each ADC conversion. In NI-DAQmx, software-timed acquisitions are referred to as having on-demand timing. Software-timed acquisitions are also referred to as immediate or static acquisitions and are typically used for reading a single sample of data.

Hardware-Timed Acquisitions

With hardware-timed acquisitions, a digital hardware signal (ai/SampleClock) controls the rate of the acquisition. This signal can be generated internally on your device or provided externally.

Hardware-timed acquisitions 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 operations can be buffered or non-buffered. A buffer is a temporary storage in computer memory for to-be-generated samples.

Buffered

In a buffered acquisition, data is moved from the DAQ device's onboard FIFO memory to a PC buffer using DMA or interrupts before it is transferred to application memory. Buffered acquisitions typically allow for much faster transfer rates than non-buffered acquisitions because data is moved in large blocks, rather than one point at a time.

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

Finite sample mode acquisition refers to the acquisition of a specific, predetermined number of data samples. After the specified number of

samples has been written out, the generation stops. If you use a reference trigger, you must use finite sample mode.

Continuous acquisition refers to the acquisition of an unspecified number of samples. Instead of acquiring a set number of data samples and stopping, a continuous acquisition continues until you stop the operation. Continuous acquisition is also referred to as *double-buffered* or *circular-buffered acquisition*.

If data cannot be transferred across the bus fast enough, the FIFO will become full. New acquisitions will overwrite data in the FIFO before it can be transferred to host memory. The device generates an error in this case. With continuous operations, if the user program does not read data out of the PC buffer fast enough to keep up with the data transfer, the buffer could reach an overflow condition, causing an error to be generated.

Non-Buffered

In non-buffered acquisitions, data is read directly from the FIFO on the device. Typically, hardware-timed, non-buffered operations are used to read single samples with known time increments between them and good latency.

Analog Input Triggering

Analog input supports three different triggering actions:

- Start trigger
- Reference trigger
- Pause trigger

Refer to the *AI Start Trigger Signal*, *AI Reference Trigger Signal*, and *AI Pause Trigger Signal* sections for information about these triggers.

A digital trigger can initiate these actions. NI 6236 devices support digital triggering, but do not support analog triggering.

Field Wiring Considerations

Environmental noise can seriously affect the measurement accuracy of the device if you do not take proper care when running signal wires between signal sources and the device. The following recommendations apply

mainly to AI signal routing to the device, although they also apply to signal routing in general.

Minimize noise pickup and maximize measurement accuracy by using individually shielded, twisted-pair wires to connect AI signals to the device. With this type of wire, the signals attached to the positive and negative input channels are twisted together and then covered with a shield. You then connect this shield only at one point to the signal source ground. This kind of connection is required for signals traveling through areas with large magnetic fields or high electromagnetic interference.

Refer to the NI Developer Zone document, *Field Wiring and Noise Considerations for Analog Signals*, for more information. To access this document, go to `ni.com/info` and enter the info code `rdfwn3`.

Analog Input Timing Signals

In order to provide all of the timing functionality described throughout this section, NI 6236 devices have a flexible timing engine. Figure 4-7 summarizes all of the timing options provided by the analog input timing engine. Also refer to the *Clock Routing* section of Chapter 10, *Digital Routing and Clock Generation*.

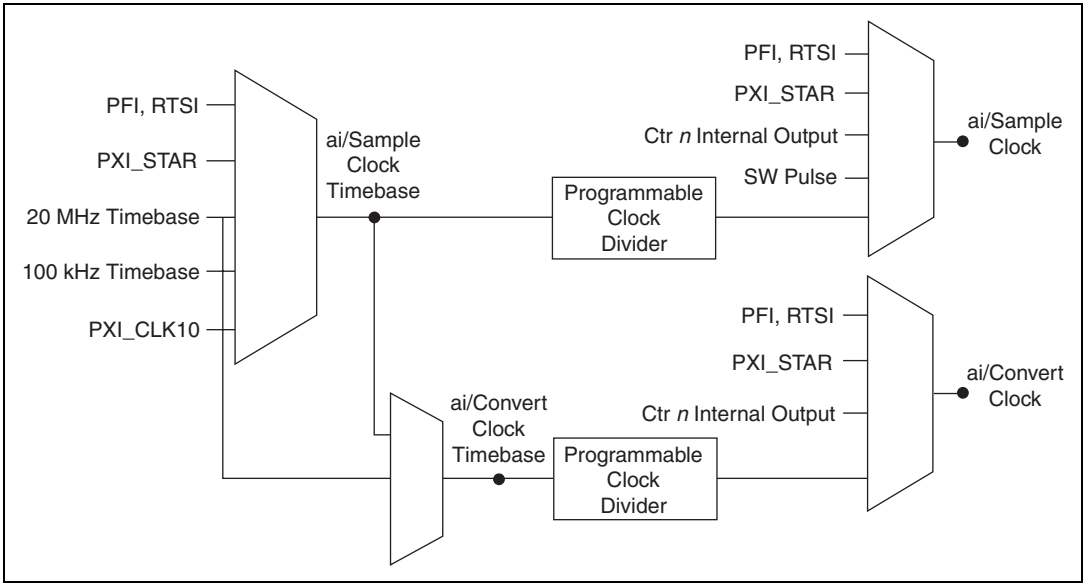


Figure 4-7. Analog Input Timing Options

M Series devices use ai/SampleClock and ai/ConvertClock to perform interval sampling. As Figure 4-8 shows, ai/SampleClock controls the sample period, which is determined by the following equation:

1/Sample Period = Sample Rate

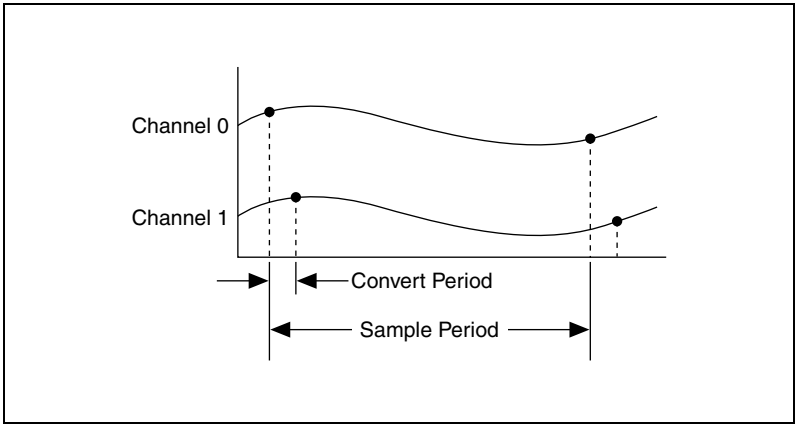


Figure 4-8. Interval Sampling

ai/ConvertClock controls the Convert Period, which is determined by the following equation:

1/Convert Period = Convert Rate

Note The sampling rate is the fastest you can acquire data on the device and still achieve accurate results. For example, if an M Series device has a sampling rate of 250 kS/s, this sampling rate is aggregate—one channel at 250 kS/s or two channels at 125 kS/s per channel illustrates the relationship.

Posttriggered data acquisition allows you to view only data that is acquired after a trigger event is received. A typical posttriggered DAQ sequence is shown in Figure 4-9. In this example, the DAQ device reads two channels five times. The sample counter is loaded with the specified number of posttrigger samples, in this example, five. The value decrements with each pulse on ai/SampleClock, until the value reaches zero and all desired samples have been acquired.

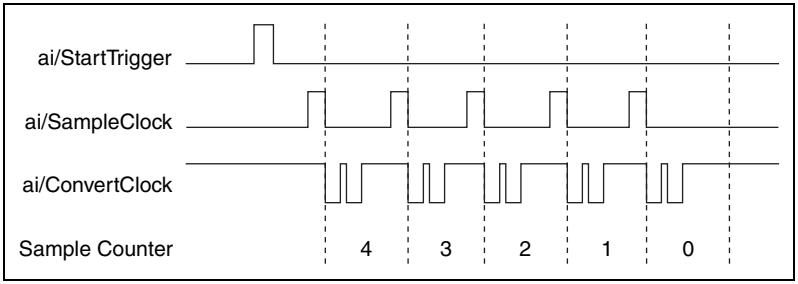


Figure 4-9. Posttriggered Data Acquisition Example

Pretriggered data acquisition allows you to view data that is acquired before the trigger of interest, in addition to data acquired after the trigger. Figure 4-10 shows a typical pretriggered DAQ sequence. ai/StartTrigger can be either a hardware or software signal. If ai/StartTrigger is set up to be a software start trigger, an output pulse appears on the ai/StartTrigger line when the acquisition begins. When the ai/StartTrigger pulse occurs, the sample counter is loaded with the number of pretriggered samples, in this example, four. The value decrements with each pulse on ai/SampleClock, until the value reaches zero. The sample counter is then loaded with the number of posttriggered samples, in this example, three.

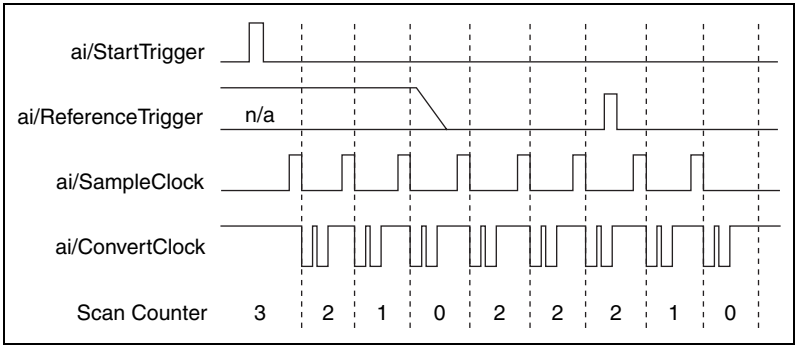


Figure 4-10. Pretriggered Data Acquisition Example

If an ai/ReferenceTrigger pulse occurs before the specified number of pretrigger samples are acquired, the trigger pulse is ignored. Otherwise, when the ai/ReferenceTrigger pulse occurs, the sample counter value decrements until the specified number of posttrigger samples have been acquired.

NI 6236 devices feature the following analog input timing signals.

- AI Sample Clock Signal
- AI Sample Clock Timebase Signal
- AI Convert Clock Signal
- AI Convert Clock Timebase Signal
- AI Hold Complete Event Signal
- AI Start Trigger Signal
- AI Reference Trigger Signal
- AI Pause Trigger Signal

AI Sample Clock Signal

Use the AI Sample Clock (ai/SampleClock) signal to initiate a set of measurements. Your M Series device samples the AI signals of every channel in the task once for every ai/SampleClock. A measurement acquisition consists of one or more samples.

You can specify an internal or external source for ai/SampleClock. You also can specify whether the measurement sample begins on the rising edge or falling edge of ai/SampleClock.

Using an Internal Source

One of the following internal signals can drive ai/SampleClock.

- Counter *n* Internal Output
- AI Sample Clock Timebase (divided down)
- A pulse initiated by host software

A programmable internal counter divides down the sample clock timebase.

Several other internal signals can be routed to ai/SampleClock through RTSI. Refer to *Device Routing in MAX* in the *NI-DAQmx Help* or the *LabVIEW 8.x Help* for more information.

Using an External Source

Use one of the following external signals as the source of ai/SampleClock:

- Input PFI <0..5>
- RTSI <0..7>
- PXI_STAR

Note Refer to the *NI 6236 Specifications* for the minimum allowable pulse width and the propagation delay of PFI <0..5>.

Routing AI Sample Clock Signal to an Output Terminal

You can route ai/SampleClock out to any output PFI <6..9> or RTSI <0..7> terminal. This pulse is always active high.

You can specify the output to have one of two behaviors. With the pulse behavior, your DAQ device briefly pulses the PFI terminal once for every occurrence of ai/SampleClock.

With level behavior, your DAQ device drives the PFI terminal high during the entire sample.

PFI <0..5> terminals are fixed inputs. PFI <6..9> terminals are fixed outputs.

Other Timing Requirements

Your DAQ device only acquires data during an acquisition. The device ignores ai/SampleClock when a measurement acquisition is not in progress. During a measurement acquisition, you can cause your DAQ device to ignore ai/SampleClock using the ai/PauseTrigger signal.

A counter on your device internally generates ai/SampleClock unless you select some external source. ai/StartTrigger starts this counter and either software or hardware can stop it once a finite acquisition completes. When using an internally generated ai/SampleClock, you also can specify a configurable delay from ai/StartTrigger to the first ai/SampleClock pulse. By default, this delay is set to two ticks of the ai/SampleClockTimebase signal. When using an externally generated ai/SampleClock, you must ensure the clock signal is consistent with respect to the timing requirements of ai/ConvertClock. Failure to do so may result in ai/SampleClock pulses that are masked off and acquisitions with erratic sampling intervals. Refer to the *AI Convert Clock Signal* section for more information about the timing requirements between ai/ConvertClock and ai/SampleClock.

Figure 4-11 shows the relationship of ai/SampleClock to ai/StartTrigger.

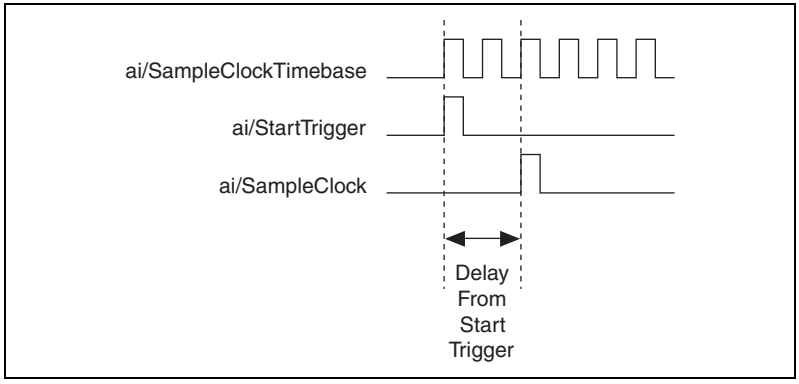


Figure 4-11. ai/SampleClock and ai/StartTrigger

AI Sample Clock Timebase Signal

You can route any of the following signals to be the AI Sample Clock Timebase (ai/SampleClockTimebase) signal:

- 20 MHz Timebase
- 100 kHz Timebase
- PXI_CLK10
- RTSI <0..7>
- Input PFI <0..5>
- PXI_STAR

ai/SampleClockTimebase is not available as an output on the I/O connector. ai/SampleClockTimebase is divided down to provide one of the possible sources for ai/SampleClock. You can configure the polarity selection for ai/SampleClockTimebase as either rising or falling edge.

AI Convert Clock Signal

Use the AI Convert Clock (ai/ConvertClock) signal to initiate a single A/D conversion on a single channel. A sample, controlled by the AI Sample Clock, consists of one or more conversions.

You can specify either an internal or external signal as the source of ai/ConvertClock. You also can specify whether the measurement sample begins on the rising edge or falling edge of ai/ConvertClock.

With NI-DAQmx, the driver will choose the fastest conversion rate possible based on the speed of the A/D converter and add 10 µs of padding between each channel to allow for adequate settling time. This scheme enables the channels to approximate simultaneous sampling and still allow for adequate settling time. If the AI Sample Clock rate is too fast to allow for this 10 µs of padding, NI-DAQmx will choose the conversion rate so that the AI Convert Clock pulses are evenly spaced throughout the sample.

To explicitly specify the conversion rate, use the **AI Convert Clock Rate DAQmx Timing** property node or function.

Caution Setting the conversion rate higher than the maximum rate specified for your device will result in errors.

Using an Internal Source

One of the following internal signals can drive ai/ConvertClock:

- AI Convert Clock Timebase (divided down)
- Counter *n* Internal Output

A programmable internal counter divides down the AI Convert Clock Timebase to generate ai/ConvertClock. The counter is started by ai/SampleClock and continues to count down to zero, produces an ai/ConvertClock, reloads itself, and repeats the process until the sample is finished. It then reloads itself in preparation for the next ai/SampleClock pulse.

Using an External Source

Use one of the following external signals as the source of ai/ConvertClock:

- Input PFI <0..5>
- RTSI <0..7>
- PXI_STAR

Note Refer to the *NI 6236 Specifications* for the minimum allowable pulse width and the propagation delay of PFI <0..5>.

Routing AI Convert Clock Signal to an Output Terminal

You can route ai/ConvertClock (as an active low signal) out to any output PFI <6..9> or RTSI <0..7> terminal.

PFI <0..5> terminals are fixed inputs. PFI <6..9> terminals are fixed outputs.

Using a Delay from Sample Clock to Convert Clock

When using an internally generated ai/ConvertClock, you also can specify a configurable delay from ai/SampleClock to the first ai/ConvertClock pulse within the sample. By default, this delay is three ticks of ai/ConvertClockTimebase.

Figure 4-12 shows the relationship of ai/SampleClock to ai/ConvertClock.

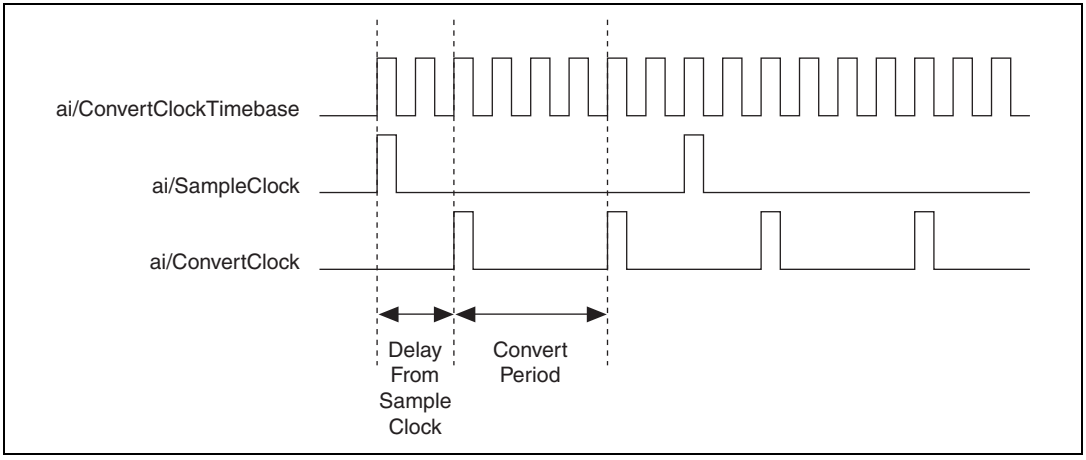


Figure 4-12. ai/SampleClock and ai/ConvertClock

Other Timing Requirements

The sample and conversion level timing of M Series devices work such that clock signals are gated off unless the proper timing requirements are met. For example, the device ignores both ai/SampleClock and ai/ConvertClock until it receives a valid ai/StartTrigger signal. After the device recognizes an ai/SampleClock pulse, it ignores subsequent ai/SampleClock pulses until it receives the correct number of ai/ConvertClock pulses.

Similarly, the device ignores all ai/ConvertClock pulses until it recognizes an ai/SampleClock pulse. After the device receives the correct number of ai/ConvertClock pulses, it ignores subsequent ai/ConvertClock pulses until it receives another ai/SampleClock. Figure 4-13 shows timing sequences for a four-channel acquisition (using AI channels 0, 1, 2, and 3) and demonstrates proper and improper sequencing of ai/SampleClock and ai/ConvertClock.

It is also possible to use a single external signal to drive both ai/SampleClock and ai/ConvertClock at the same time. In this mode, each tick of the external clock will cause a conversion on the ADC. Figure 4-13 shows this timing relationship.

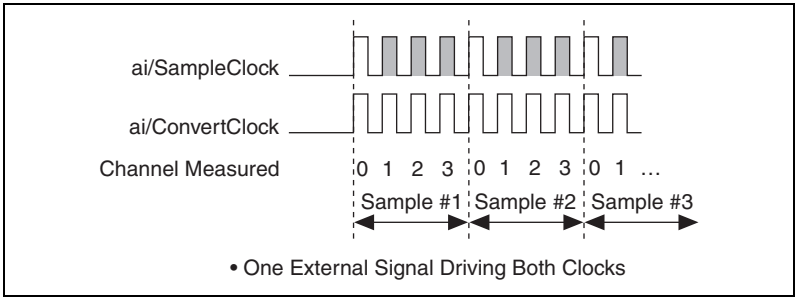


Figure 4-13. Single External Signal Driving ai/SampleClock and ai/ConvertClock Simultaneously

AI Convert Clock Timebase Signal

The AI Convert Clock Timebase (ai/ConvertClockTimebase) signal is divided down to provide one of the possible sources for ai/ConvertClock. Use one of the following signals as the source of ai/ConvertClockTimebase:

- ai/SampleClockTimebase
- 20 MHz Timebase

ai/ConvertClockTimebase is not available as an output on the I/O connector.

AI Hold Complete Event Signal

The AI Hold Complete Event (ai/HoldCompleteEvent) signal generates a pulse after each A/D conversion begins. You can route ai/HoldCompleteEvent out to any output PFI <6..9> or RTSI <0..7> terminal.

The polarity of ai/HoldCompleteEvent is software-selectable, but is typically configured so that a low-to-high leading edge can clock external AI multiplexers indicating when the input signal has been sampled and can be removed.

AI Start Trigger Signal

Use the AI Start Trigger (ai/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. After the acquisition begins, configure the acquisition to stop:

- When a certain number of points are 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.

Using a Digital Source

To use ai/StartTrigger with a digital source, specify a source and an edge. The source can be any of the following signals:

- Input PFI <0..5>
- RTSI <0..7>
- Counter *n* Internal Output
- PXI_STAR

Note Refer to the *NI 6236 Specifications* for the minimum allowable pulse width and the propagation delay of PFI <0..5>.

The source also can be one of several other internal signals on your DAQ device. Refer to *Device Routing in MAX* in the *NI-DAQmx Help* or the *LabVIEW 8.x Help* for more information.

You also can specify whether the measurement acquisition begins on the rising edge or falling edge of ai/StartTrigger.

Routing AI Start Trigger to an Output Terminal

You can route ai/StartTrigger out to any output PFI <6..9> or RTSI <0..7> terminal.

The output is an active high pulse.

The device also uses ai/StartTrigger to initiate pretriggered DAQ operations. In most pretriggered applications, a software trigger generates ai/StartTrigger. Refer to the *AI Reference Trigger Signal* section for a

complete description of the use of ai/StartTrigger and ai/ReferenceTrigger in a pretriggered DAQ operation.

AI Reference Trigger Signal

Use a reference trigger (ai/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.

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

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

When the reference trigger occurs, the DAQ device continues to write samples to the buffer until the buffer contains the number of posttrigger samples desired. Figure 4-14 shows the final buffer.

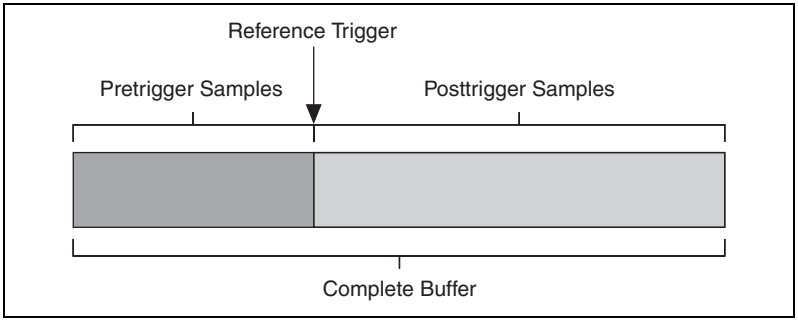


Figure 4-14. Reference Trigger Final Buffer

Using a Digital Source

To use ai/ReferenceTrigger with a digital source, specify a source and an edge. The source can be any of the following signals:

- Input PFI <0..5>
- RTSI <0..7>
- PXI_STAR

The source also can be one of several internal signals on your DAQ device. Refer to *Device Routing in MAX* in the *NI-DAQmx Help* or the *LabVIEW 8.x Help* for more information.

You also can specify whether the measurement acquisition stops on the rising edge or falling edge of ai/ReferenceTrigger.

Routing AI Reference Trigger Signal to an Output Terminal

You can route ai/ReferenceTrigger out to any output PFI <6..9> or RTSI <0..7> terminal.

AI Pause Trigger Signal

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

Using a Digital Source

To use ai/SampleClock, specify a source and a polarity. The source can be any of the following signals:

- Input PFI <0..5>
- RTSI <0..7>
- PXI_STAR

Note Refer to the *NI 6236 Specifications* for the minimum allowable pulse width and the propagation delay of PFI <0..5>.

The source also can be one of several other internal signals on your DAQ device. Refer to *Device Routing in MAX* in the *NI-DAQmx Help* or the *LabVIEW 8.x Help* for more information.

Routing AI Pause Trigger Signal to an Output Terminal

You can route ai/PauseTrigger out to RTSI <0..7>.

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

Getting Started with AI Applications in Software

You can use the M Series device in the following analog input applications.

- Single-point analog input
- Finite analog input
- Continuous analog input

You can perform these applications through DMA, interrupt, or programmed I/O data transfer mechanisms. Some of the applications also use start, reference, and pause triggers.

Note For more information about programming analog input applications and triggers in software, refer to the *NI-DAQmx Help* or the *LabVIEW 8.x Help*.

5

Analog Output

NI 6236 devices have four AO channels that are controlled by a single clock and are capable of waveform generation.

Figure 5-1 shows the analog output circuitry of NI 6236 devices.

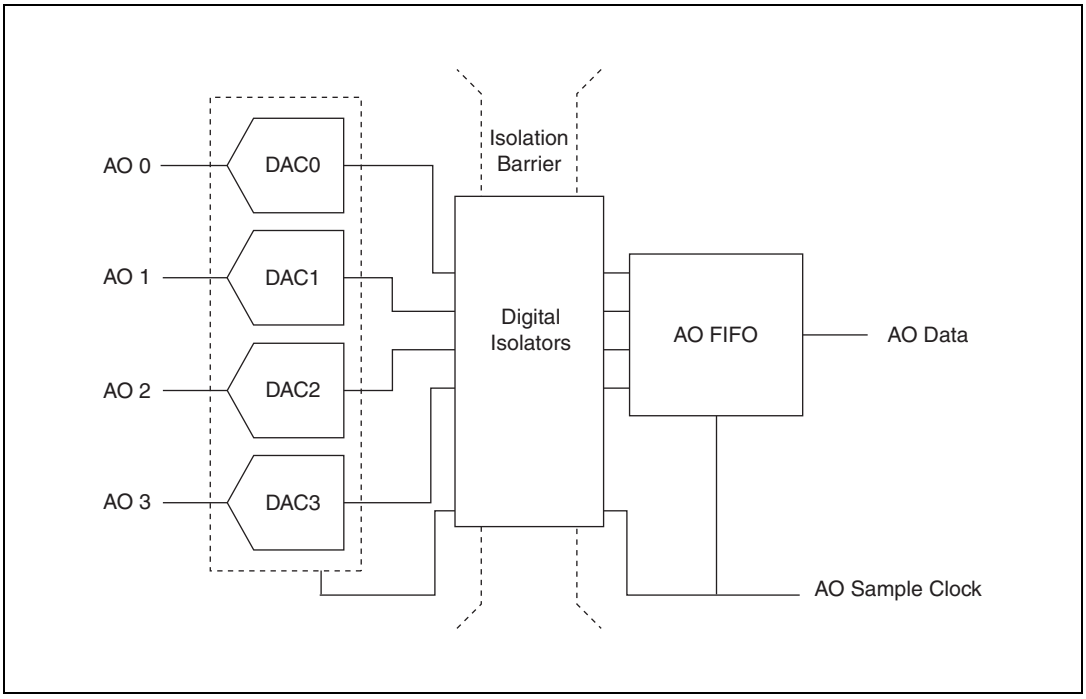


Figure 5-1. NI 6236 Analog Output Circuitry

Analog Output Circuitry

DACs

Digital-to-analog converters (DACs) convert digital codes to analog voltages.

AO FIFO

The AO FIFO enables analog output waveform generation. It is a first-in-first-out (FIFO) memory buffer between the computer and the DACs. It allows you to download the points of a waveform to your M Series device without host computer interaction.

AO Sample Clock

The AO Sample Clock signal reads a sample from the DAC FIFO and generates the AO voltage.

Isolation Barrier and Digital Isolators

The digital isolators across the isolation barrier provide a ground break between the isolated analog front end and the earth/chassis/building ground.

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 (usually worst at mid-scale transitions). 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. Visit `ni.com/support` for more information about minimizing glitches.

Analog Output Data Generation Methods

When performing an analog output operation, you either can perform software-timed or hardware-timed generations. Hardware-timed generations can be non-buffered or 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 a single value out, such as a constant DC voltage.

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 your device 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 can be deterministic.
- Hardware-timed acquisitions can use hardware triggering.

Hardware-timed operations can be buffered or non-buffered. A buffer is a temporary storage in computer memory for to-be-generated samples.

Non-Buffered

In non-buffered acquisitions, data is written directly to the DACs on the device. Typically, hardware-timed, non-buffered operations are used to write single samples with known time increments between them and good latency.

Buffered

In a buffered acquisition, data is moved from a PC buffer to the DAQ device's onboard FIFO using DMA or interrupts before it is written to the DACs one sample at a time. Buffered acquisitions typically allow for much faster transfer rates than non-buffered acquisitions because data is moved in large blocks, rather than one point at a time.

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

Finite sample mode generation refers to the generation of a specific, predetermined number of data samples. After the specified number of samples has been written out, the generation stops.

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 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. After 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 after the operation is started, thereby preventing any problems that may occur due to excessive bus traffic.

With non-regeneration, old data will not be 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 will underflow and cause an error.

Analog Output Triggering

Analog output supports two different triggering actions:

- Start trigger
- Pause trigger

A digital trigger can initiate these actions. NI 6236 devices support digital triggering, but do not support analog triggering. Refer to the *AO Start Trigger Signal* and *AO Pause Trigger Signal* sections for more information about these triggering actions.

Connecting Analog Voltage Output Signals

AO <0..3> are the voltage output signals for AO channels 0, 1, 2, and 3. AO GND is the ground reference for AO <0..3>.

Figure 5-2 shows how to make AO connections to the device.

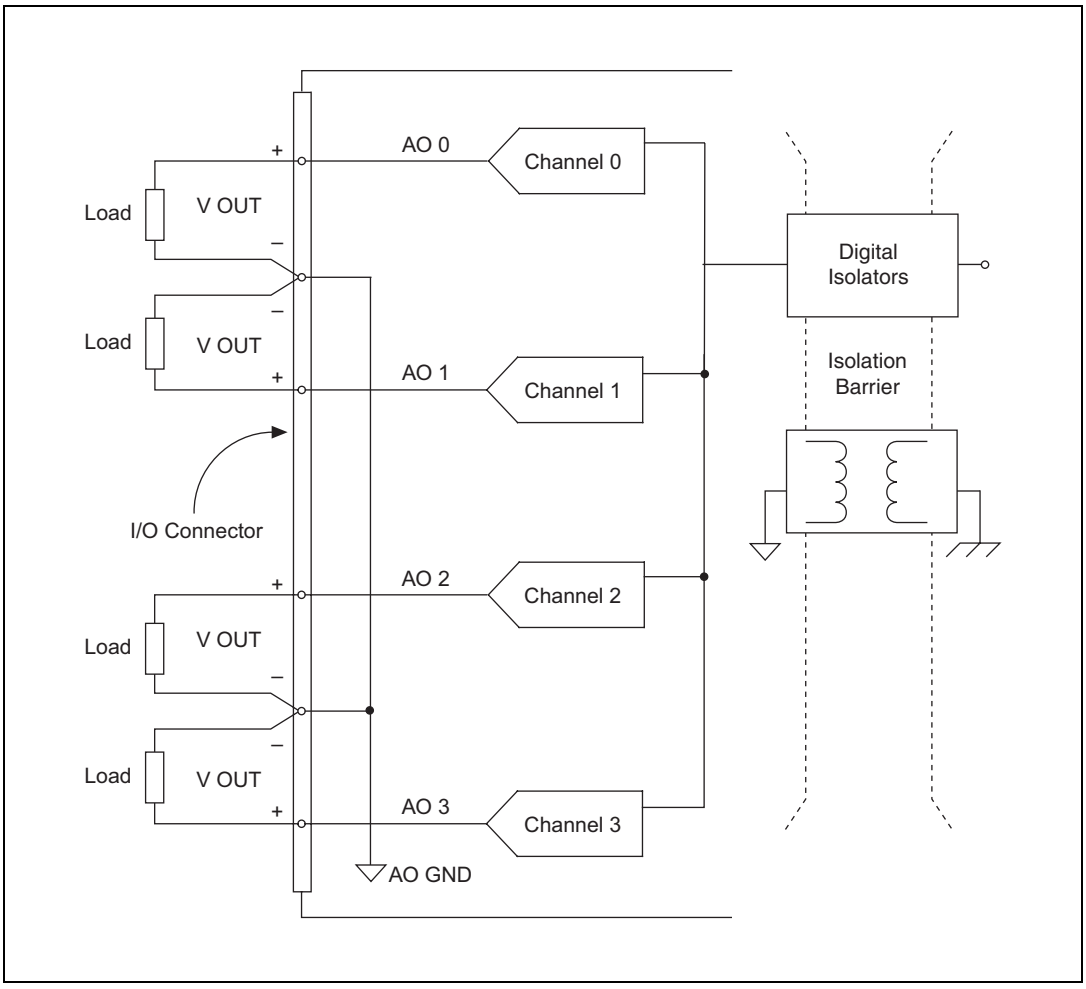


Figure 5-2. Analog Output Connections

Analog Output Timing Signals

Figure 5-3 summarizes all of the timing options provided by the analog output timing engine.

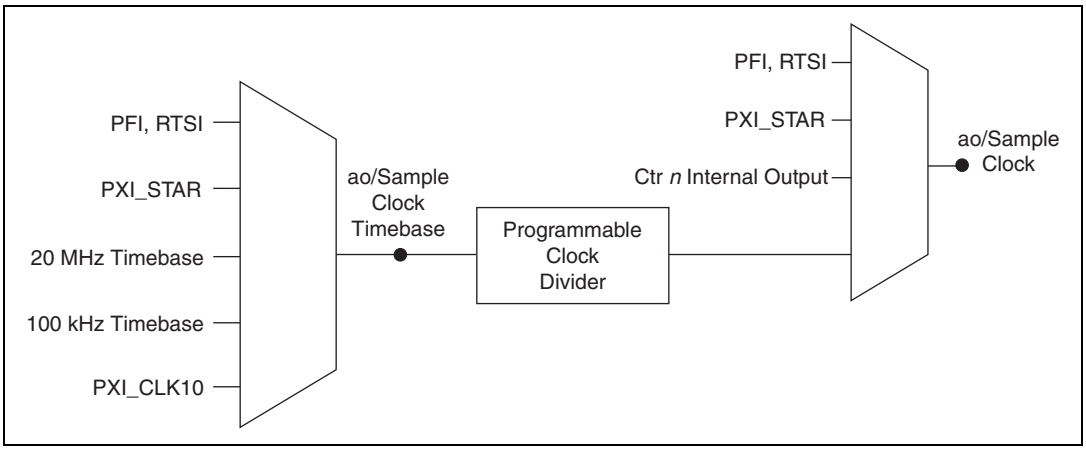


Figure 5-3. Analog Output Timing Options

NI 6236 devices feature the following AO (waveform generation) timing signals.

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

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.

Using a Digital Source

To use ao/StartTrigger, specify a source and an edge. The source can be one of the following signals:

- A pulse initiated by host software
- Input PFI <0..5>
- RTSI <0..7>
- ai/ReferenceTrigger
- ai/StartTrigger
- PXI_STAR

The source also can be one of several internal signals on your DAQ device. Refer to *Device Routing in MAX* in the *NI-DAQmx Help* or the *LabVIEW 8.x Help* for more information.

You also can specify whether the waveform generation begins on the rising edge or falling edge of ao/StartTrigger.

Routing AO Start Trigger Signal to an Output Terminal

You can route ao/StartTrigger out to any output PFI <6..9> or RTSI <0..7> terminal.

The output is an active high pulse.

PFI <0..5> terminals are fixed inputs. PFI <6..9> terminals are fixed outputs.

AO Pause Trigger Signal

Use the AO Pause Trigger signal (ao/PauseTrigger) to mask off samples in a DAQ sequence. That is, when ao/PauseTrigger is active, no samples occur.

ao/PauseTrigger 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 your sample clock is the onboard clock, the generation resumes as soon as the pause trigger is deasserted, as shown in Figure 5-4.

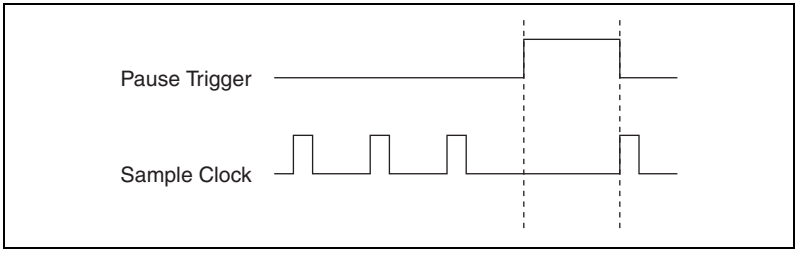


Figure 5-4. ao/PauseTrigger with the Onboard Clock Source

If you are using any signal other than the onboard clock as the source of your sample clock, the generation resumes as soon as the pause trigger is

deasserted and another edge of the sample clock is received, as shown in Figure 5-5.

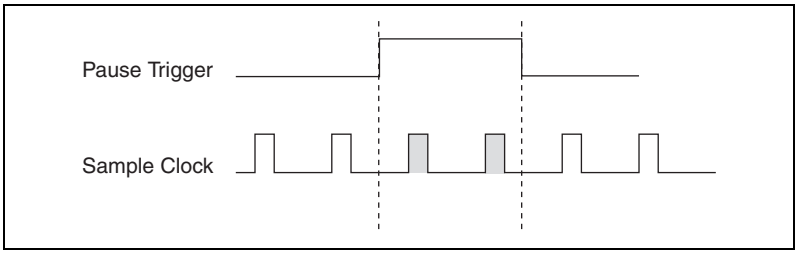


Figure 5-5. ao/PauseTrigger with Other Signal Source

Using a Digital Source

To use ao/PauseTrigger, specify a source and a polarity. The source can be one of the following signals:

- Input PFI <0..5>
- RTSI <0..7>
- PXI_STAR

The source also can be one of several other internal signals on your DAQ device. Refer to *Device Routing in MAX* in the *NI-DAQmx Help* or the *LabVIEW 8.x Help* for more information.

You also can specify whether the samples are paused when ao/PauseTrigger is at a logic high or low level.

Routing AO Pause Trigger Signal to an Output Terminal

You can route ao/PauseTrigger out to RTSI <0..7>.

AO Sample Clock Signal

Use the AO Sample Clock (ao/SampleClock) signal to initiate AO samples. Each sample updates the outputs of all of the DACs. You can specify an internal or external source for ao/SampleClock. You also can specify whether the DAC update begins on the rising edge or falling edge of ao/SampleClock.

Using an Internal Source

One of the following internal signals can drive ao/SampleClock.

- AO Sample Clock Timebase (divided down)
- Counter *n* Internal Output

A programmable internal counter divides down the AO Sample Clock Timebase signal.

Using an External Source

Use one of the following external signals as the source of ao/SampleClock:

- Input PFI <0..5>
- RTSI <0..7>
- PXI_STAR

Routing AO Sample Clock Signal to an Output Terminal

You can route ao/SampleClock (as an active low signal) out to any output PFI <6..9> or RTSI <0..7> terminal.

Other Timing Requirements

A counter on your device internally generates ao/SampleClock unless you select some external source. ao/StartTrigger starts the counter and either the software or hardware can stop it when a finite generation completes. When using an internally generated ao/SampleClock, you also can specify a configurable delay from ao/StartTrigger to the first ao/SampleClock pulse. By default, this delay is two ticks of ao/SampleClockTimebase.

Figure 5-6 shows the relationship of ao/SampleClock to ao/StartTrigger.

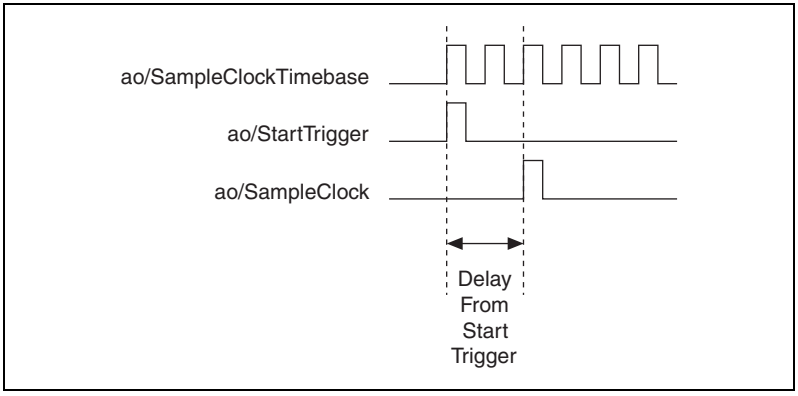


Figure 5-6. ao/SampleClock and ao/StartTrigger

AO Sample Clock Timebase Signal

The AO Sample Clock Timebase (ao/SampleClockTimebase) signal is divided down to provide a source for ao/SampleClock.

You can route any of the following signals to be the AO Sample Clock Timebase (ao/SampleClockTimebase) signal:

- 20 MHz Timebase
- 100 kHz Timebase
- PXI_CLK10
- Input PFI <0..5>
- RTSI <0..7>
- PXI_STAR

ao/SampleClockTimebase is not available as an output on the I/O connector.

You might use ao/SampleClockTimebase if you want to use an external sample clock signal, but need to divide the signal down. If you want to use an external sample clock signal, but do not need to divide the signal, then you should use ao/SampleClock rather than ao/SampleClockTimebase.

Getting Started with AO Applications in Software

You can use an M Series device in the following analog output applications.

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

You can perform these generations through programmed I/O, interrupt, or DMA data transfer mechanisms. Some of the applications also use start triggers and pause triggers.



Note For more information about programming analog output applications and triggers in software, refer to the *NI-DAQmx Help* or the *LabVIEW 8.x Help*.

6

Digital Input and Output

NI 6236 devices have six static digital input lines, P0.<0..5>. These lines also can be used as PFI inputs.

NI 6236 devices have four static digital output lines, P1.<0..3>. These lines also can be used as PFI output. By default the digital output lines are disabled (high impedance) on power up. Software can enable or disable the entire port (software cannot enable individual lines). After the port is enabled, you can individually configure each line to the following:

- Set a line to a static 0
- Set a line to a static 1
- Export a timing output signal to a line as a PFI pin

The voltage input and output levels and the current drive level of the DI and DO lines are listed in the *NI 6236 Specifications*. Refer to Chapter 8, *PFI*, for more information about PFI inputs and outputs.

I/O Protection

Each DI, DO, and PFI signal is protected against overvoltage, undervoltage, and overcurrent conditions as well as ESD events. However, you should avoid these fault conditions by following these guidelines.

- Do *not* connect any digital output line to any external signal source, ground signal, or power supply.
- Understand the current requirements of the load connected to the digital output lines. Do *not* exceed the specified current output limits of the digital outputs. NI has several signal conditioning solutions for digital applications requiring high current drive.
- Do *not* drive the digital input lines with voltages outside of its normal operating range. The DI, DO, or PFI lines have a smaller operating range than the AI signals.
- Treat the DAQ device as you would treat any static sensitive device. *Always* properly ground yourself and the equipment when handling the DAQ device or connecting to it.

Programmable Power-Up States

By default, the digital output lines (P1.<0..3>/PFI <6..9>) are disabled (high impedance) at power up. Software can configure the board to power up with the entire port enabled or disabled; you cannot enable individual lines. If the port powers up enabled, you also can configure each line individually to power up as 1 or 0.

Refer to the *NI-DAQmx Help* or the *LabVIEW 8.x Help* for more information about setting power-up states in NI-DAQmx or MAX.

Connecting Digital I/O Signals

The DI signals P0.<0..5> and DO signals P1.<0..3> are referenced to D GND. Figure 6-1 shows P0.<0..5> and P1.<0..3>. Digital input applications include receiving TTL signals and sensing external device states, such as the state of the switch shown in the figure. Digital output applications include sending TTL signals and driving external devices, such as the LED shown in the figure.

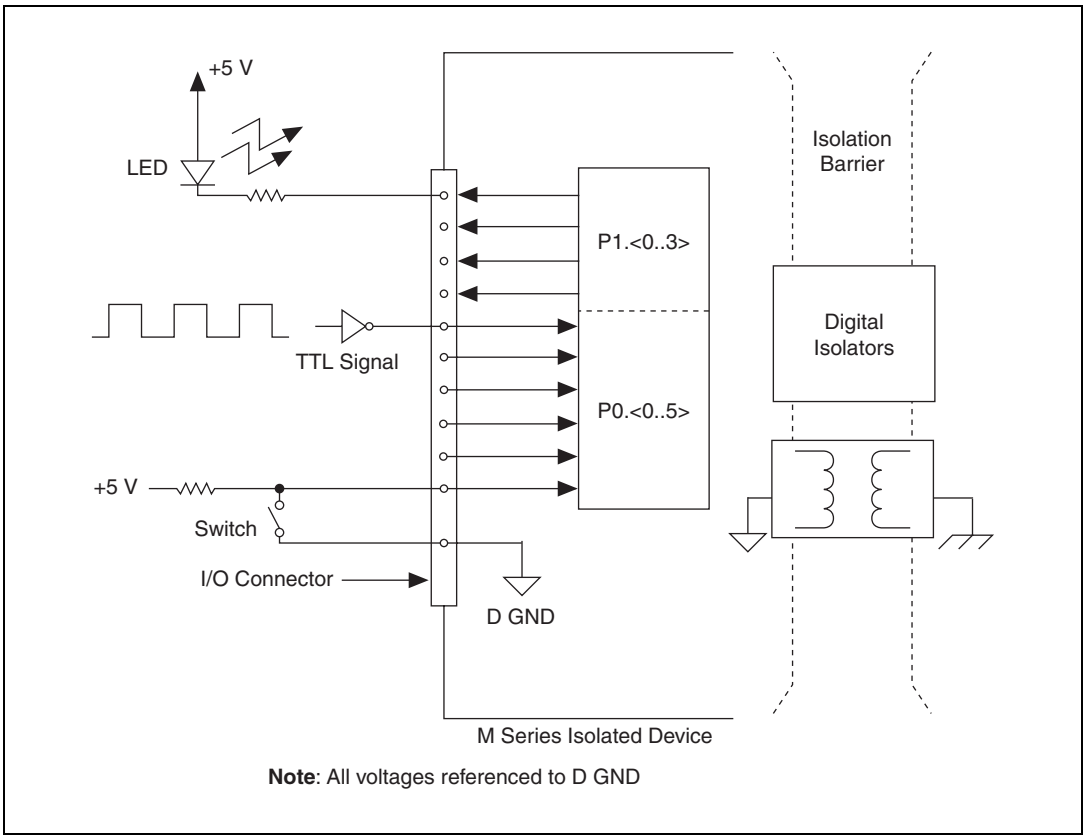


Figure 6-1. Digital I/O Connections

Caution Exceeding the maximum input voltage or maximum working voltage ratings, which are listed in the *NI 6236 Specifications*, can damage the DAQ device and the computer. NI is *not* liable for any damage resulting from such signal connections.

Getting Started with DIO Applications in Software

You can use the NI 6236 device in the following digital I/O applications:

- Static digital input
- Static digital output

Note For more information about programming digital I/O applications and triggers in software, refer to the *NI-DAQmx Help* or the *LabVIEW 8.x Help*.

7

Counters

NI 6236 devices have two general-purpose 32-bit counter/timers and one frequency generator, as shown in Figure 7-1. The general-purpose counter/timers can be used for many measurement and pulse generation applications.

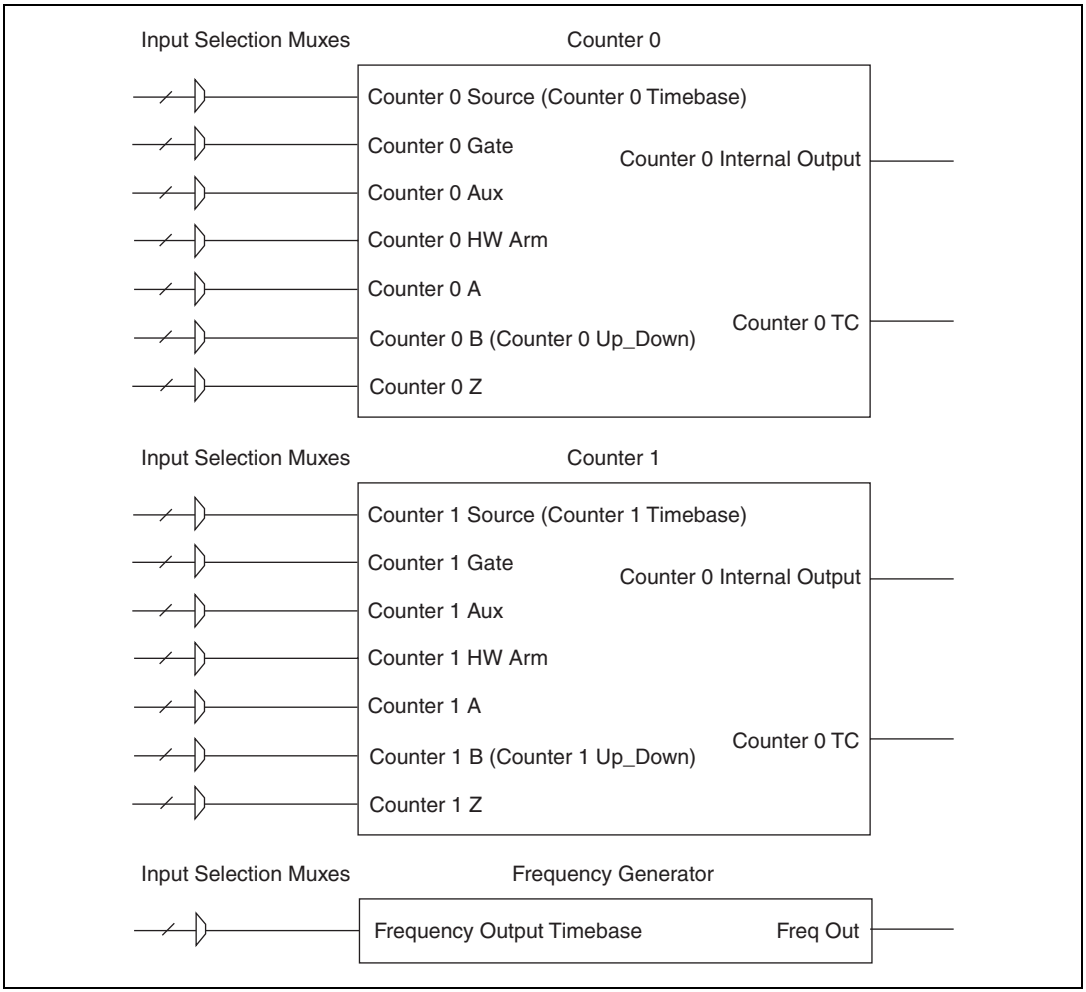


Figure 7-1. M Series Counters

The counters have seven input signals, although in most applications only a few inputs are used.

For information about connecting counter signals, refer to the *Default Counter Terminals* section.

Counter Input Applications

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).

The counter values can be read on demand or with a sample clock.

Single Point (On-Demand) Edge Counting

With single point (on-demand) edge counting, the counter counts the number of edges on the Source input after the counter is armed. *On-demand* refers to the fact that software can read the counter contents at any time without disturbing the counting process. Figure 7-2 shows an example of single point edge counting.

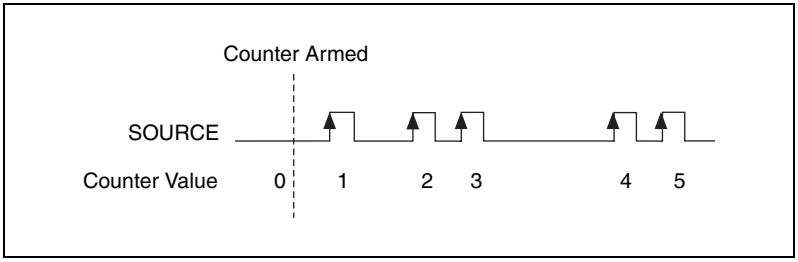


Figure 7-2. Single Point (On-Demand) Edge Counting

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

You can route the pause trigger to the Gate input of the counter. You can configure the counter to pause counting when the pause trigger is high or when it is low. Figure 7-3 shows an example of on-demand edge counting with a pause trigger.

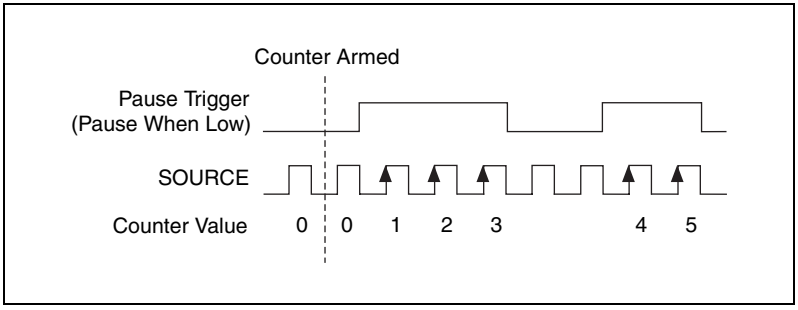


Figure 7-3. Single Point (On-Demand) Edge Counting with Pause Trigger

Buffered (Sample Clock) Edge Counting

With buffered edge counting (edge counting using a sample clock), the counter counts the number of edges on the Source input after the counter is armed. The value of the counter is sampled on each active edge of a sample clock. A DMA controller transfers the sampled values to host memory.

The count values returned are the cumulative counts since the counter armed event. That is, the sample clock does not reset the counter.

You can route the counter sample clock to the Gate input of the counter. You can configure the counter to sample on the rising or falling edge of the sample clock.

Figure 7-4 shows an example of buffered edge counting. Notice that counting begins when the counter is armed, which occurs before the first active edge on Gate.

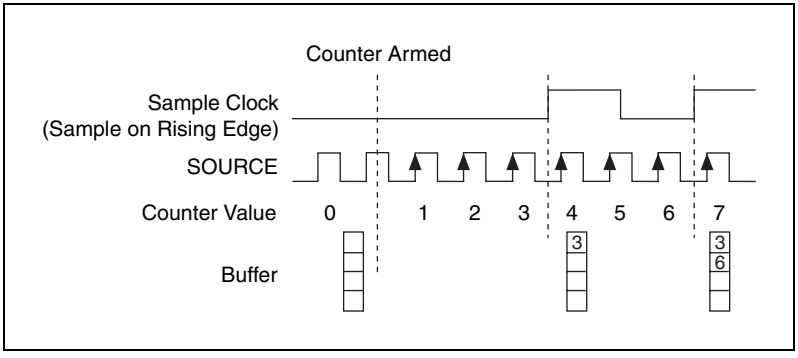


Figure 7-4. Buffered (Sample Clock) Edge Counting

Non-Cumulative Buffered Edge Counting

Non-cumulative edge counting is similar to buffered (sample clock) edge counting. However, the counter resets after each active edge of the Sample Clock. You can route the Sample Clock to the Gate input of the counter.

Figure 7-5 shows an example of non-cumulative buffered edge counting.

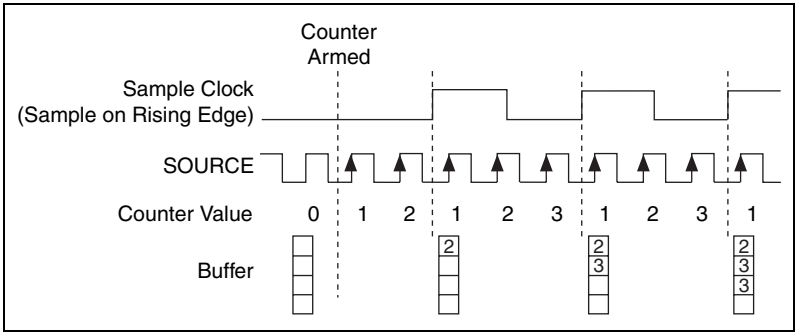


Figure 7-5. Non-Cumulative Buffered Edge Counting

Notice that the first count interval begins when the counter is armed, which occurs before the first active edge on Gate.

Notice that if you are using an external signal as the Source, at least one Source pulse should occur between each active edge of the Gate signal. This condition ensures that correct values are returned by the counter. If this condition is not met, consider using duplicate count prevention.

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 *n* B input is high; count down when it is low

For information about connecting counter signals, refer to the *Default Counter Terminals* 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.

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 a hardware save register and ignores other edges on the Gate and Source inputs. Software then can read the stored count.

Figure 7-6 shows an example of a single pulse-width measurement.

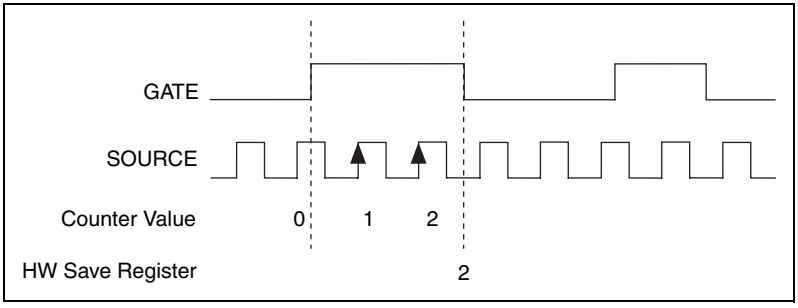


Figure 7-6. Single Pulse-Width Measurement

Buffered Pulse-Width Measurement

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 a hardware save register. A DMA controller transfers the stored values to host memory.

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

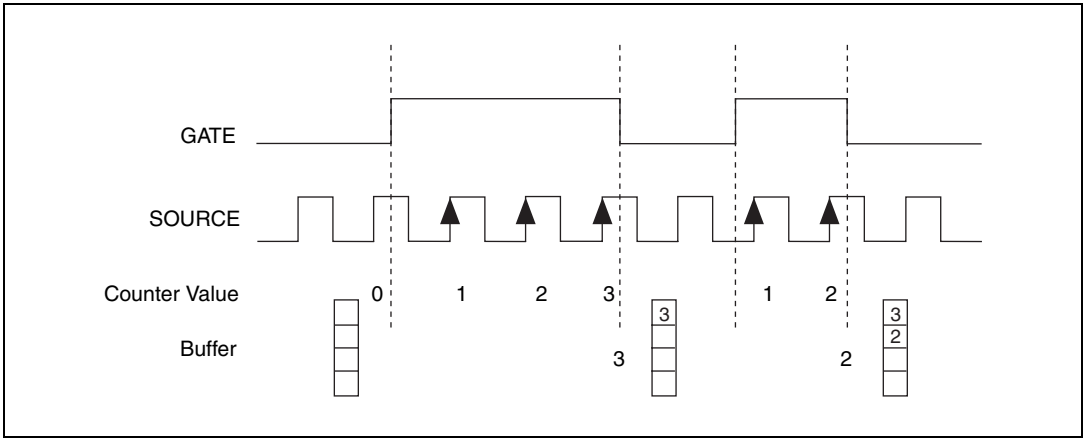


Figure 7-7. Buffered Pulse-Width Measurement

Note that if you are using an external signal as the Source, at least one Source pulse should occur between each active edge of the Gate signal. This condition ensures that correct values are returned by the counter. If this condition is not met, consider using duplicate count prevention.

For information about connecting counter signals, refer to the *Default Counter Terminals* 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.

Single Period Measurement

With single period measurement, the counter counts the number of rising (or falling) edges on the Source input occurring between two active edges of the Gate input. On the second active edge of the Gate input, the counter stores the count in a hardware save register and ignores other edges on the Gate and Source inputs. Software then can read the stored count.

Figure 7-8 shows an example of a single period measurement.

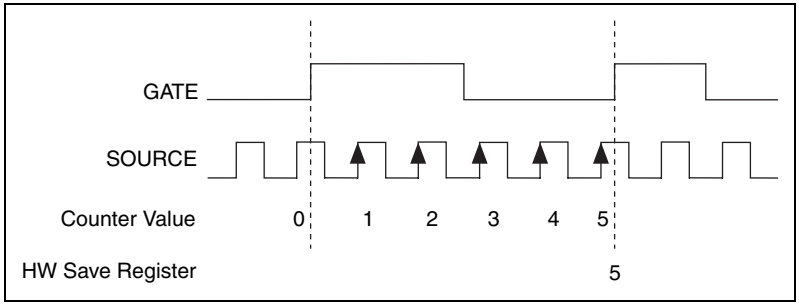


Figure 7-8. Single Period Measurement

Buffered Period Measurement

Buffered period measurement is similar to single period measurement, but buffered period measurement measures multiple periods.

The counter counts the number of rising (or falling) edges on the Source input between each pair of active edges on the Gate input. At the end of each period on the Gate signal, the counter stores the count in a hardware save register. A DMA controller transfers the stored values to host memory.

The counter begins when it is armed. The arm usually occurs in the middle of a period of the Gate input. So the first value stored in the hardware save register does not reflect a full period of the Gate input. In most applications, this first point should be discarded.

Figure 7-9 shows an example of a buffered period measurement.

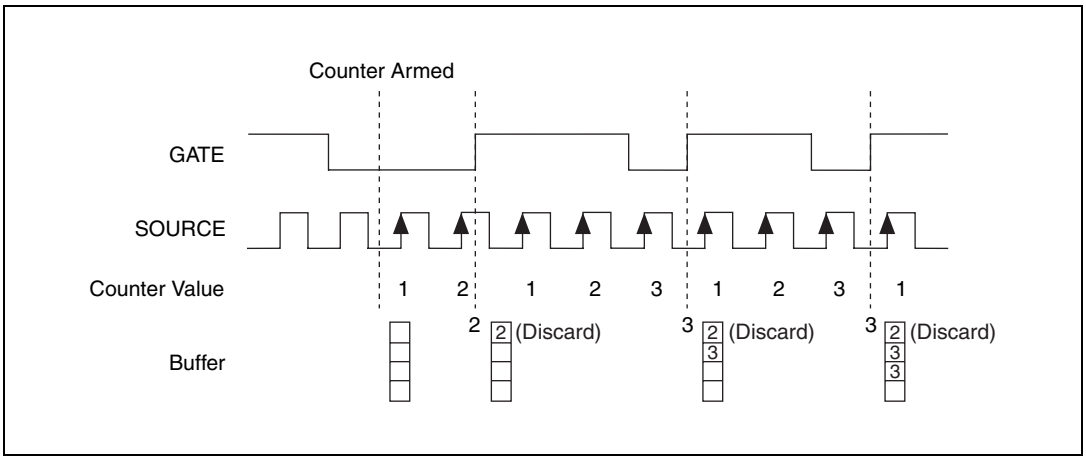


Figure 7-9. Buffered Period Measurement

Note that if you are using an external signal as the Source, at least one Source pulse should occur between each active edge of the Gate signal. This condition ensures that correct values are returned by the counter. If this condition is not met, consider using duplicate count prevention.

For information about connecting counter signals, refer to the *Default Counter Terminals* 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.

Single Semi-Period Measurement

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

Buffered Semi-Period Measurement

In buffered semi-period measurement, on each edge of the Gate signal, the counter stores the count in a hardware save register. A DMA controller transfers the stored values to host memory.

The counter begins counting when it is armed. The arm usually occurs between edges on the Gate input. So the first value stored in the hardware save register does not reflect a full semi-period of the Gate input. In most applications, this first point should be discarded.

Figure 7-10 shows an example of a buffered semi-period measurement.

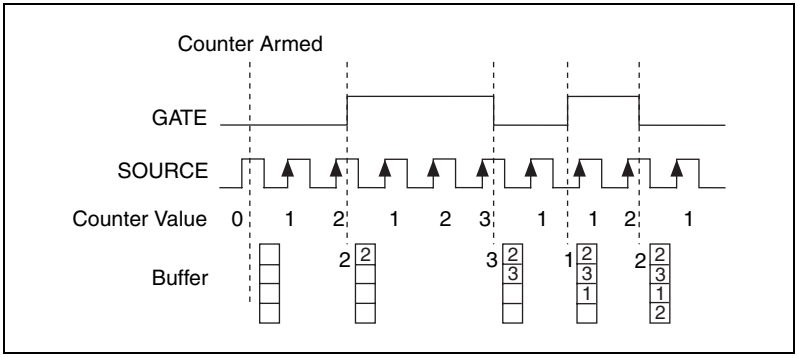


Figure 7-10. Buffered Semi-Period Measurement

Note that if you are using an external signal as the Source, at least one Source pulse should occur between each active edge of the Gate signal. This condition ensures that correct values are returned by the counter. If this condition is not met, consider using duplicate count prevention.

For information about connecting counter signals, refer to the *Default Counter Terminals* section.

Frequency Measurement

You can use the counters to measure frequency in several different ways. You can choose one of the following methods depending on your application.

Method 1—Measure Low Frequency with One Counter

In this method, you measure one period of your signal using a known timebase. This method is good for low frequency signals.

You can route the signal to measure (F1) to the Gate of a counter. You can route a known timebase (Ft) to the Source of the counter. The known timebase can be 80MHzTimebase. For signals that might be slower than 0.02 Hz, use a slower known timebase.

You can configure the counter to measure one period of the gate signal. The frequency of F1 is the inverse of the period. Figure 7-11 illustrates this method.

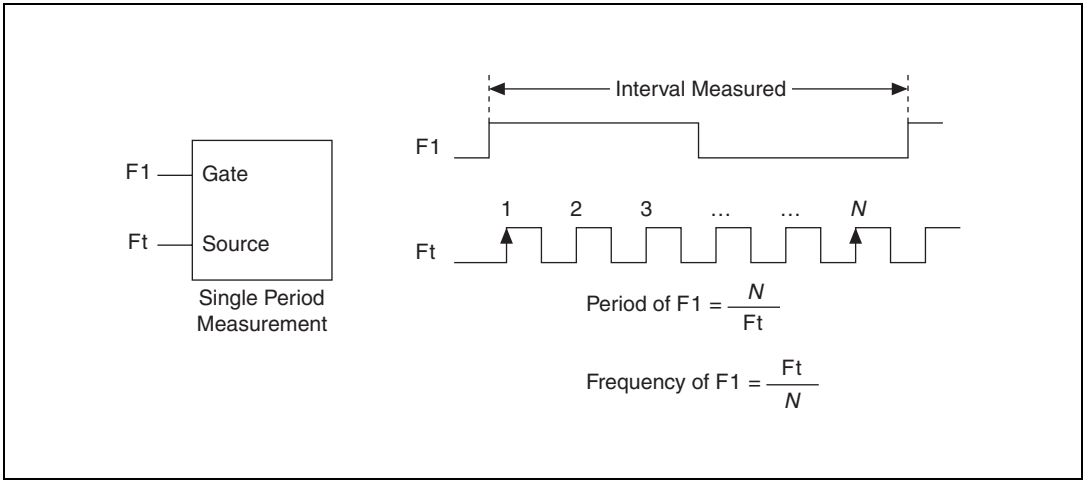


Figure 7-11. Method 1

Method 1b—Measure Low Frequency with One Counter (Averaged)

In this method, you measure several periods of your signal using a known timebase. This method is good for low to medium frequency signals.

You can route the signal to measure (F1) to the Gate of a counter. You can route a known timebase (Ft) to the Source of the counter. The known timebase can be 80MHzTimebase. For signals that might be slower than 0.02 Hz, use a slower known timebase.

You can configure the counter to make K + 1 buffered period measurements. Recall that the first period measurement in the buffer should be discarded.

Average the remaining K period measurements to determine the average period of F1. The frequency of F1 is the inverse of the average period. Figure 7-12 illustrates this method.

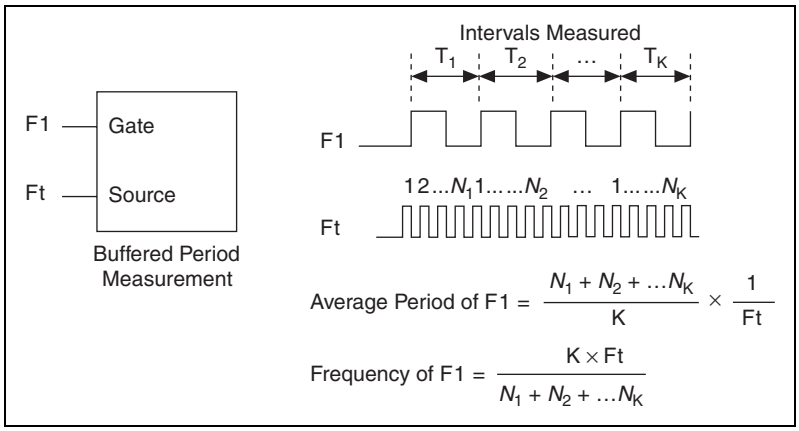


Figure 7-12. Method 1b

Method 2—Measure High Frequency with Two Counters

In this method, you measure one pulse of a known width using your signal and derive the frequency of your signal from the result. This method is good for high frequency signals.

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 or RTSI terminal. You only need to use one counter if you generate the pulse externally.

Route the signal to measure (F1) to the Source of the counter. Configure the counter for a single pulse-width measurement. Suppose you measure the width of pulse T to be *N* periods of F1. Then the frequency of F1 is *N*/T.

Figure 7-13 illustrates this method. Another option would be to measure the width of a known period instead of a known pulse.

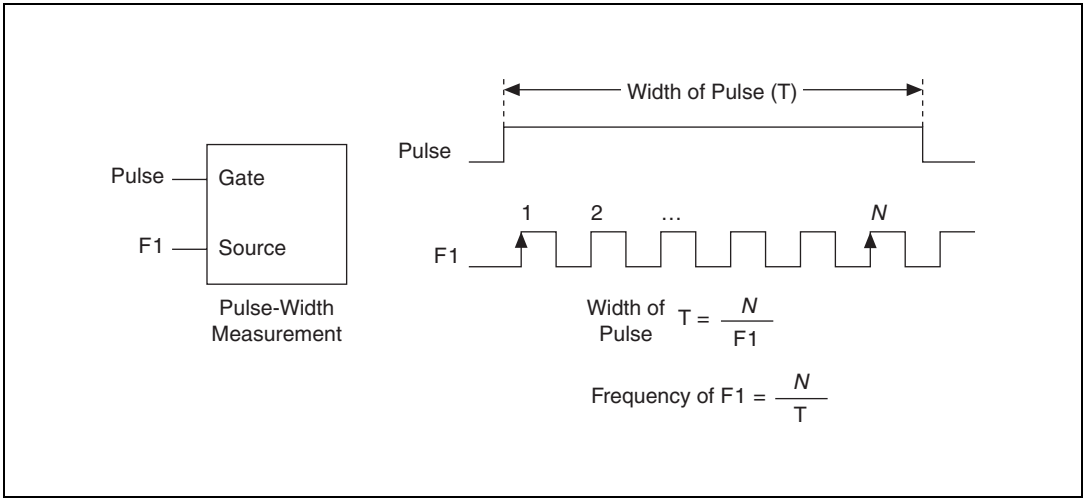


Figure 7-13. Method 2

Method 3—Measure Large Range of Frequencies Using 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*. In this method, you generate a long pulse using the signal to measure. You then measure the long pulse with a known timebase. The M Series device can measure this long pulse more accurately than the faster input signal.

You can route the signal to measure to the Source input of Counter 0, as shown in Figure 7-14. Assume this signal to measure has frequency F1. Configure Counter 0 to generate a single pulse that is the width of *N* periods of the source input signal.

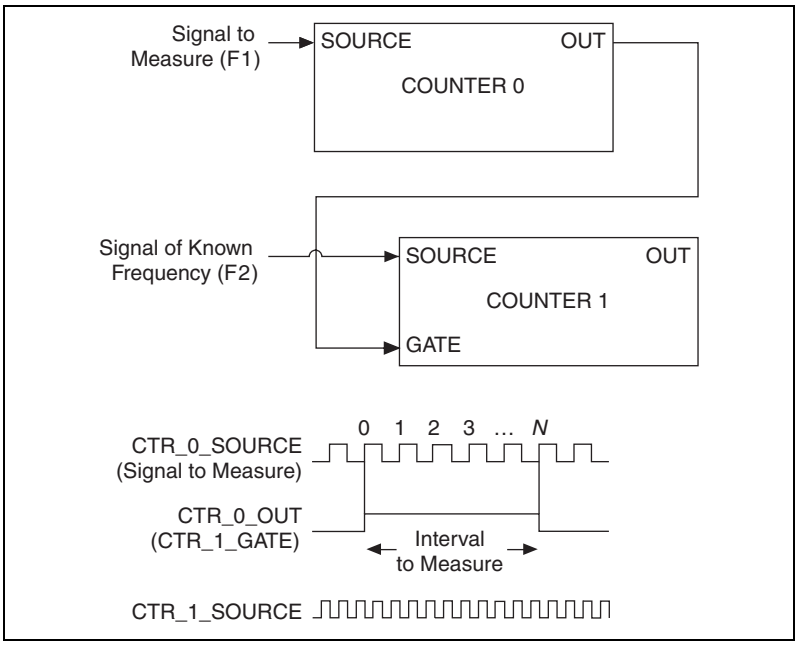


Figure 7-14. Method 3

Then route the Counter 0 Internal Output signal to the Gate input of Counter 1. You can route a signal of known frequency (F2) to the Counter 1 Source input. F2 can be 80MHzTimebase. For signals that might be slower than 0.02 Hz, use a slower known timebase. Configure Counter 1 to perform a single pulse-width measurement. Suppose the result is that the pulse width is J periods of the F2 clock.

From Counter 0, the length of the pulse is *N*/F1. From Counter 1, the length of the same pulse is J/F2. Therefore, the frequency of F1 is given by F1 = F2 * (*N*/J).

Choosing a Method for Measuring Frequency

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

- Method 1 uses only one counter. It is a good method for many applications. However, the accuracy of the measurement decreases as the frequency increases.

  Consider a frequency measurement on a 50 kHz signal using an 80 MHz Timebase. This frequency corresponds to 1600 cycles of the

80 MHz Timebase. Your measurement may return 1600 ± 1 cycles depending on the phase of the signal with respect to the timebase. As your frequency becomes larger, this error of ±1 cycle becomes more significant; Table 7-1 illustrates this point.

Table 7-1. Frequency Measurement Method 1

Task	Equation	Example 1	Example 2
Actual Frequency to Measure	F1	50 kHz	5 MHz
Timebase Frequency	Ft	80 MHz	80 MHz
Actual Number of Timebase Periods	Ft/F1	1600	16
Worst Case Measured Number of Timebase Periods	(Ft/F1) – 1	1599	15
Measured Frequency	Ft F1/(Ft – F1)	50.125 kHz	5.33 MHz
Error	[Ft F1/(Ft – F1)] – F1	125 kHz	333 kHz
Error %	[Ft/(Ft – F1)] – 1	0.06%	6.67%

- Method 1b (measuring K periods of F1) improves the accuracy of the measurement. A disadvantage of Method 1b is that you have to take K + 1 measurements. These measurements take more time and consume some of the available PCI or PXI bandwidth.
- Method 2 is accurate for high frequency signals. However, the accuracy decreases as the frequency of the signal to measure decreases. At very low frequencies, Method 2 may be too inaccurate for your application. Another disadvantage of Method 2 is that it requires two counters (if you cannot provide an external signal of known width). An advantage of Method 2 is that the measurement completes in a known amount of time.
- Method 3 measures high and low frequency signals accurately. However, it requires two counters.

Table 7-2 summarizes some of the differences in methods of measuring frequency.

Table 7-2. Frequency Measurement Method Comparison

Method	Number of Counters Used	Number of Measurements Returned	Measures High Frequency Signals Accurately	Measures Low Frequency Signals Accurately
1	1	1	Poor	Good
1b	1	Many	Fair	Good
2	1 or 2	1	Good	Poor
3	2	1	Good	Good

For information about connecting counter signals, refer to the *Default Counter Terminals* section.

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.

Measurements Using Quadrature Encoders

The counters can perform measurements of quadrature encoders that use X1, X2, or X4 encoding.

A quadrature encoder can have up to three channels—channels A, B, and Z.

X1 Encoding

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

Figure 7-15 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.

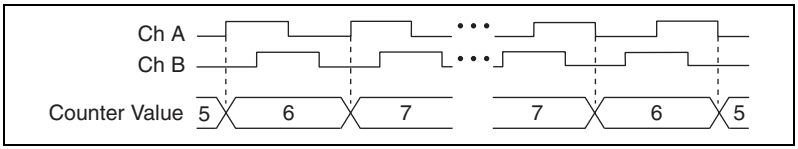


Figure 7-15. X1 Encoding

X2 Encoding

The same behavior holds for X2 encoding except the counter increments or decrements on each edge of channel A, depending on which channel leads the other. Each cycle results in two increments or decrements, as shown in Figure 7-16.

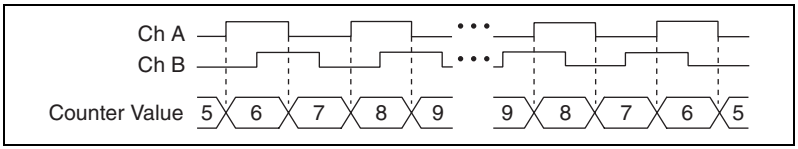


Figure 7-16. X2 Encoding

X4 Encoding

Similarly, the counter increments or decrements on each edge of channels A and B for X4 encoding. Whether the counter increments or decrements depends on which channel leads the other. Each cycle results in four increments or decrements, as shown in Figure 7-17.

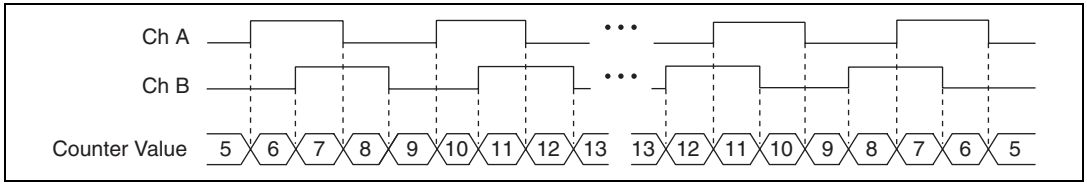


Figure 7-17. X4 Encoding

Channel Z Behavior

Some quadrature encoders have a third channel, channel Z, which is also referred to as the index channel. A high level on channel Z causes the counter to be reloaded with a specified value in a specified phase of the

quadrature cycle. You can program this reload to occur in any one of the four phases in a quadrature cycle.

Channel Z behavior—when it goes high and how long it stays high—differs with quadrature encoder designs. You must refer to the documentation for your quadrature encoder to obtain timing of channel Z with respect to channels A and B. You must then ensure that channel Z is high during at least a portion of the phase you specify for reload. For instance, in Figure 7-18, channel Z is never high when channel A is high and channel B is low. Thus, the reload must occur in some other phase.

In Figure 7-18, 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.

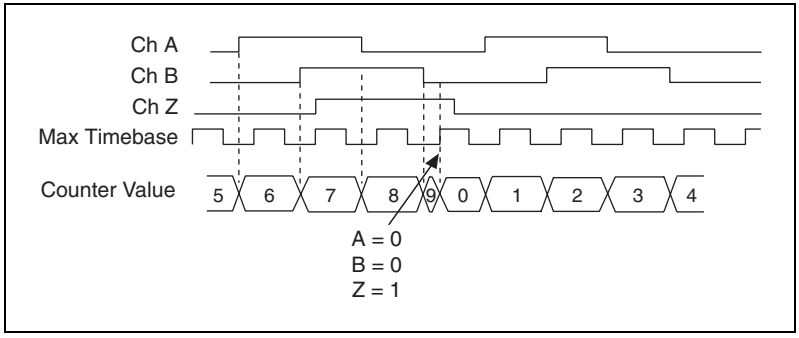


Figure 7-18. Channel Z Reload with X4 Decoding

Measurements Using Two Pulse Encoders

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

The counter increments on each rising edge of channel A. The counter decrements on each rising edge of channel B, as shown in Figure 7-19.

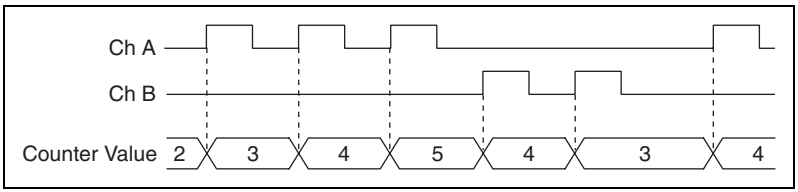


Figure 7-19. Measurements Using Two Pulse Encoders

For information about connecting counter signals, refer to the *Default Counter Terminals* section.

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 a hardware save register.

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.

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 a hardware save register and ignores other edges on its inputs. Software then can read the stored count.

Figure 7-20 shows an example of a single two-signal edge-separation measurement.

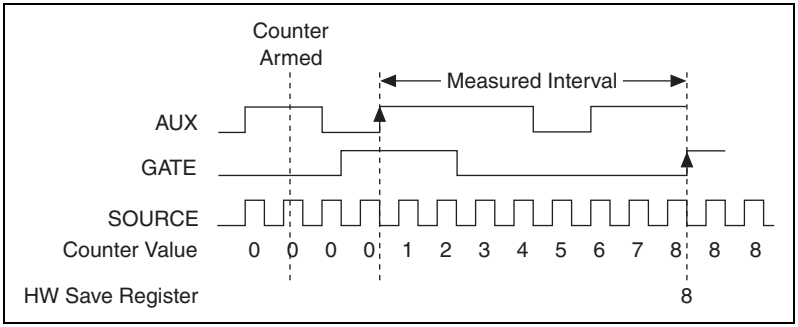


Figure 7-20. Single Two-Signal Edge-Separation Measurement

Buffered Two-Signal Edge-Separation Measurement

Buffered and single two-signal edge-separation measurements are similar, but 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 a hardware save register. On the next active edge of the Gate signal, the counter begins another measurement. A DMA controller transfers the stored values to host memory.

Figure 7-21 shows an example of a buffered two-signal edge-separation measurement.

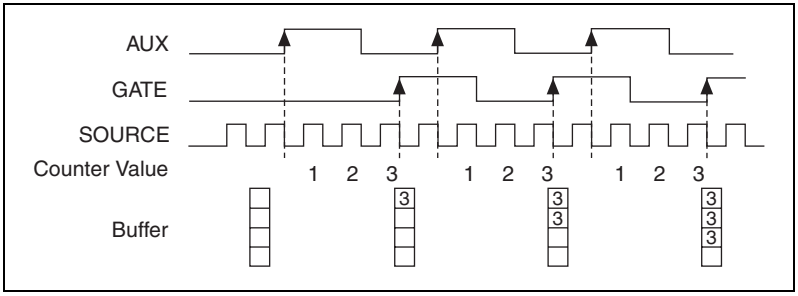


Figure 7-21. Buffered Two-Signal Edge-Separation Measurement

For information about connecting counter signals, refer to the *Default Counter Terminals* section.

Counter Output Applications

Simple Pulse Generation

Single Pulse Generation

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

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

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

Figure 7-22 shows a generation of a pulse with a pulse delay of four and a pulse width of three (using the rising edge of Source).

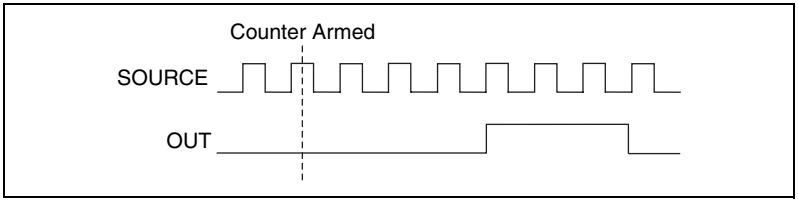


Figure 7-22. Single Pulse Generation

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 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 the 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.

After the Start Trigger signal pulses once, the counter ignores the Gate input.

Figure 7-23 shows a generation of a pulse with a pulse delay of four and a pulse width of three (using the rising edge of Source).

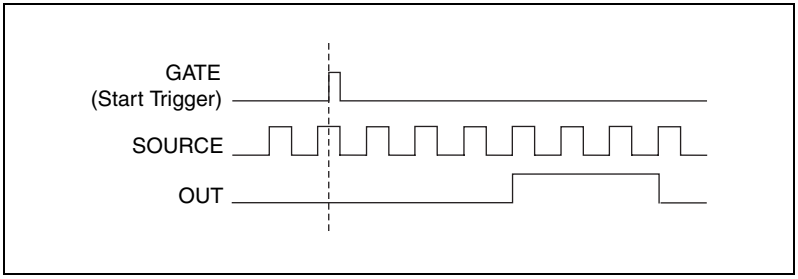


Figure 7-23. Single Pulse Generation with Start Trigger

Retriggerable Single Pulse Generation

The counter can output a single pulse in response to each pulse on a hardware Start Trigger signal. The 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 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.

Figure 7-24 shows a generation of two pulses with a pulse delay of five and a pulse width of three (using the rising edge of Source).

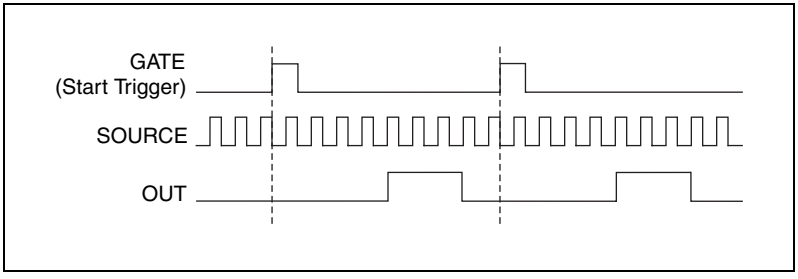


Figure 7-24. Retriggerable Single Pulse Generation

For information about connecting counter signals, refer to the *Default Counter Terminals* section.

Pulse Train Generation

Continuous Pulse Train Generation

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

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

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

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

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

Figure 7-25 shows a continuous pulse train generation (using the rising edge of Source).

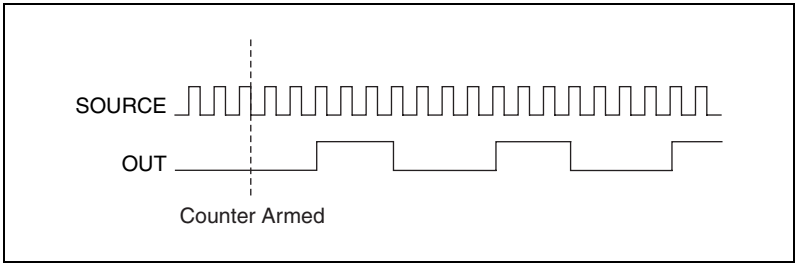


Figure 7-25. Continuous Pulse Train Generation

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

For information about connecting counter signals, refer to the *Default Counter Terminals* section.

Frequency Generation

You can generate a frequency by using a counter in pulse train generation mode or by using the frequency generator circuit.

Using the Frequency Generator

The frequency generator can output a square wave at many different frequencies. The frequency generator is independent of the two general-purpose 32-bit counter/timer modules on M Series devices.

Figure 7-26 shows a block diagram of the frequency generator.

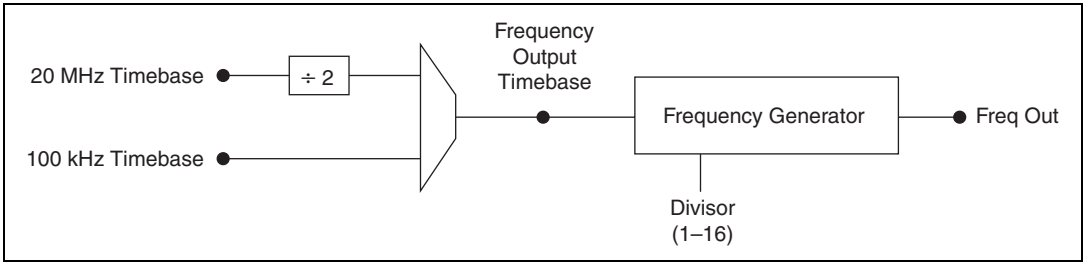


Figure 7-26. Frequency Generator Block Diagram

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 divided by 2 or the 100 kHz Timebase.

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

Figure 7-27 shows the output waveform of the frequency generator when the divider is set to 5.

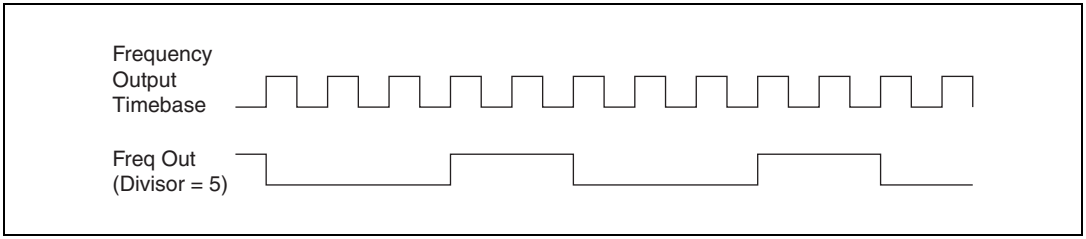


Figure 7-27. Frequency Generator Output Waveform

Frequency Output can be routed out to any output PFI <6..9> or RTSI <0..7> terminal. All PFI terminals are set to high-impedance at startup.

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

For information about connecting counter signals, refer to the *Default Counter Terminals* section.

Frequency Division

The counters can generate a signal with a frequency that is a fraction of an input signal. This function is equivalent to continuous pulse train generation.

For information about connecting counter signals, refer to the *Default Counter Terminals* section.

Pulse Generation for ETS

In this 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.

Note ETS = Equivalent Time Sampling.

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 will increase by 10 every time a new pulse is generated.

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

The waveform thus produced at the counter's output can be used to provide timing for undersampling applications where a digitizing system can sample repetitive waveforms that are higher in frequency than the Nyquist

frequency of the system. Figure 7-28 shows an example of pulse generation for ETS; the delay from the trigger to the pulse increases after each subsequent Gate active edge.

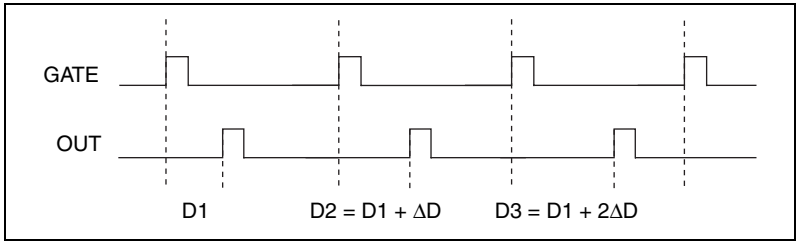


Figure 7-28. Pulse Generation for ETS

For information about connecting counter signals, refer to the *Default Counter Terminals* section.

Counter Timing Signals

M Series devices feature the following counter timing signals.

- Counter *n* Source
- Counter *n* Gate
- Counter *n* Aux
- Counter *n* A
- Counter *n* B
- Counter *n* Z
- Counter *n* Up_Down
- Counter *n* HW Arm
- Counter *n* Internal Output
- Counter *n* TC
- Frequency Output

In this section, *n* refers to either Counter 0 or 1. For example, Counter *n* Source refers to two signals—Counter 0 Source (the source input to Counter 0) and Counter 1 Source (the source input to Counter 1).

Counter *n* Source Signal

The selected edge of the Counter *n* Source signal increments and decrements the counter value depending on the application the counter is

performing. Table 7-3 lists how this terminal is used in various applications.

Table 7-3. 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
- RTSI <0..7>
- Input PFI <0..5>
- PXI_CLK10
- PXI_STAR

In addition, Counter 1 TC or Counter 1 Gate can be routed to Counter 0 Source. Counter 0 TC or Counter 0 Gate can be routed to Counter 1 Source.

Some of these options may not be available in some driver software.

Routing Counter *n* Source to an Output Terminal

You can route Counter *n* Source out to any output PFI <6..9> or RTSI <0..7> terminal. All PFIs are set to high-impedance at startup.

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.

- RTSI <0..7>
- Input PFI <0..5>
- ai/ReferenceTrigger
- ai/StartTrigger
- ai/SampleClock
- ai/ConvertClock
- ao/SampleClock
- PXI_STAR

In addition, Counter 1 Internal Output or Counter 1 Source can be routed to Counter 0 Gate. Counter 0 Internal Output or Counter 0 Source can be routed to Counter 1 Gate.

Some of these options may not be available in some driver software.

Routing Counter *n* Gate to an Output Terminal

You can route Counter *n* Gate out to any output PFI <6..9> or RTSI <0..7> terminal. All PFIs are set to high-impedance at startup.

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.

- RTSI <0..7>
- Input PFI <0..5>
- ai/ReferenceTrigger

- ai/StartTrigger
- PXI_STAR

In addition, Counter 1 Internal Output, Counter 1 Gate, Counter 1 Source, or Counter 0 Gate can be routed to Counter 0 Aux. Counter 0 Internal Output, Counter 0 Gate, Counter 0 Source, or Counter 1 Gate can be routed to Counter 1 Aux.

Some of these options may not be available in some driver software.

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.

- RTSI <0..7>
- Input PFI <0..5>
- PXI_STAR

Routing Counter *n* Z Signal to an Output Terminal

You can route Counter *n* Z out to RTSI <0..7>.

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 buffered semi-period measurement, 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.

- RTSI <0..7>
- Input PFI <0..5>
- ai/ReferenceTrigger
- ai/StartTrigger
- PXI_STAR

Counter 1 Internal Output can be routed to Counter 0 HW Arm. Counter 0 Internal Output can be routed to Counter 1 HW Arm.

Some of these options may not be available in some driver software.

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.

Routing Counter *n* Internal Output to an Output Terminal

You can route Counter *n* Internal Output to any output PFI <6..9> or RTSI <0..7> terminal. All output PFIs are set to high-impedance at startup.

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 output PFI <6..9> terminal. All PFIs are set to high-impedance at startup.

Default Counter Terminals

By default, NI-DAQmx routes the counter/timer inputs and outputs to the PFI pins, shown in Table 7-4.

Table 7-4. NI 6236 Device Default NI-DAQmx Counter/Timer Pins

Counter/Timer Signal	Default Pin Number (Name)	Port
CTR 0 SRC	13 (PFI 0)	P0.0
CTR 0 GATE	32 (PFI 1)	P0.1
CTR 0 AUX	33 (PFI 2)	P0.2
CTR 0 OUT	17 (PFI 6)	P1.0
CTR 0 A	13 (PFI 0)	P0.0
CTR 0 Z	32 (PFI 1)	P0.1
CTR 0 B	33 (PFI 2)	P0.2
CTR 1 SRC	15 (PFI 3)	P0.3
CTR 1 GATE	34 (PFI 4)	P0.4
CTR 1 AUX	35 (PFI 5)	P0.5
CTR 1 OUT	36 (PFI 7)	P1.1
CTR 1 A	15 (PFI 3)	P0.3
CTR 1 Z	34 (PFI 4)	P0.4
CTR 1 B	35 (PFI 5)	P0.5
FREQ OUT	37 (PFI 8)	P1.2

You can use these defaults or select other sources and destinations for the counter/timer signals in NI-DAQmx. Refer to *Connecting Counter Signals* in the *NI-DAQmx Help* or the *LabVIEW 8.x Help* for more information about how to connect your signals for common counter measurements and generations. M Series default PFI lines for counter functions are listed in *Physical Channels* in the *NI-DAQmx Help* or the *LabVIEW 8.x Help*.

Counter Triggering

Counters support three different triggering actions—arm start, start, and pause.

Arm Start Trigger

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

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

Start Trigger

For counter output operations, a start trigger can be configured 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.

Counter input operations can use the arm start trigger to have start trigger-like behavior.

Pause Trigger

You can use pause triggers in edge counting and continuous pulse generation applications. For edge counting acquisitions, the counter stops counting edges while the external trigger signal is low and resumes when the signal goes high or vice versa. For continuous pulse generations, the counter stops generating pulses while the external trigger signal is low and resumes when the signal goes high or vice versa.

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

Other Counter Features

Cascading Counters

You can internally route the Counter *n* Internal Output and Counter *n* TC signal 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 *Method 3—Measure Large Range of Frequencies Using Two Counters* section.

Counter Filters

You can enable a programmable debouncing filter on each PFI, RTSI, or PXI_STAR signal. When the filters are enabled, your device samples the input on each rising edge of a filter clock. M Series devices use an onboard oscillator to generate the filter clock with a 40 MHz frequency.

Note NI-DAQmx *only* supports filters on counter inputs.

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; refer to Table 7-5.

Table 7-5. Filters

Filter Setting	N (Filter Clocks Needed to Pass Signal)	Pulse Width Guaranteed to Pass Filter	Pulse Width Guaranteed to Not Pass Filter
125 ns	5	125 ns	100 ns
6.425 µs	257	6.425 µs	6.400 µs
2.55 ms	~101,800	2.55 ms	2.54 ms
Disabled	—	—	—

The filter setting for each input can be configured independently. On power up, the filters are disabled. Figure 7-29 shows an example of a low to high transition on an input that has its filter set to 125 ns (N = 5).

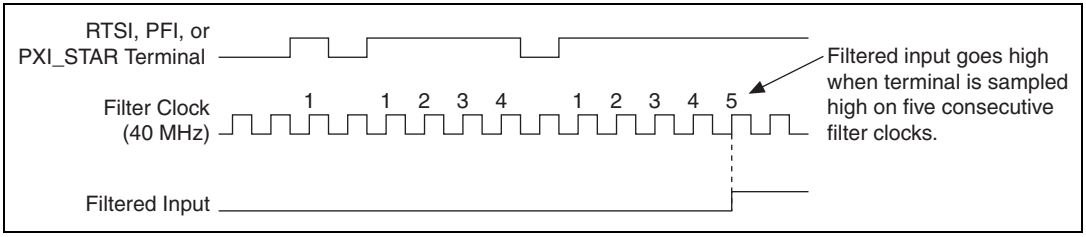


Figure 7-29. Filter Example

Enabling filters introduces jitter on the input signal. For the 125 ns and 6.425 µs filter settings, the jitter is up to 25 ns. On the 2.55 ms setting, the jitter is up to 10.025 µs.

When a PFI input is routed directly to RTSI, or a RTSI input is routed directly to PFI, the M Series device does not use the filtered version of the input signal.

Refer to the KnowledgeBase document, *Digital Filtering with M Series*, for more information about digital filters and counters. To access this KnowledgeBase, go to `ni.com/info` and enter the info code `rddfms`.

Prescaling

Prescaling allows the counter to count a signal that is faster than the maximum timebase of the counter. M Series devices offer 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.

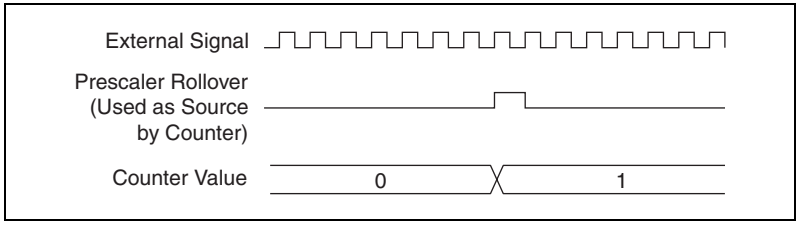


Figure 7-30. Prescaling

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). 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).

Duplicate Count Prevention

Duplicate count prevention (or synchronous counting mode) ensures that a counter returns correct data in applications that use a slow or non-periodic external source. Duplicate count prevention applies only to buffered counter applications such as measuring frequency or period. In such buffered applications, the counter should store the number of times an external Source pulses between rising edges on the Gate signal.

Example Application That Works Correctly (No Duplicate Counting)

Figure 7-31 shows an external buffered signal as the period measurement Source.

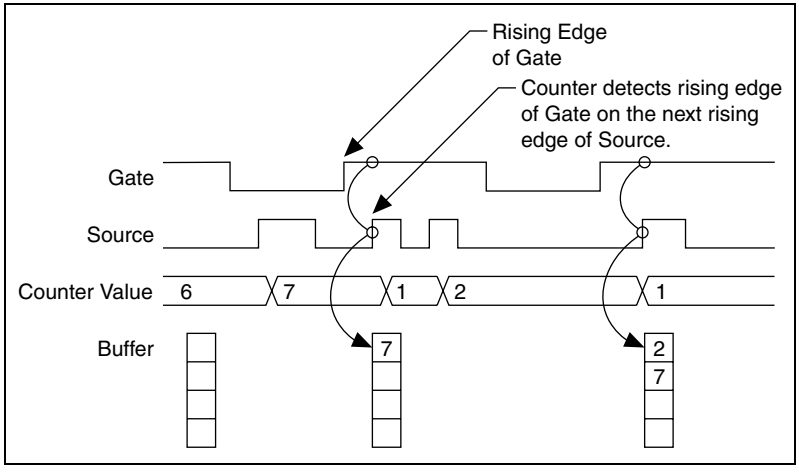


Figure 7-31. Duplicate Count Prevention Example

On the first rising edge of the Gate, the current count of 7 is stored. On the next rising edge of the Gate, the counter stores a 2 since two Source pulses occurred after the previous rising edge of Gate.

The counter synchronizes or samples the Gate signal with the Source signal, so the counter does not detect a rising edge in the Gate until the next Source pulse. In this example, the counter stores the values in the buffer on the first rising Source edge after the rising edge of Gate. The details of when exactly the counter synchronizes the Gate signal vary depending on the synchronization mode. Synchronization modes are described in the *Synchronization Modes* section.

Example Application That Works Incorrectly (Duplicate Counting)

In Figure 7-32, after the first rising edge of Gate, no Source pulses occur. So the counter does not write the correct data to the buffer.

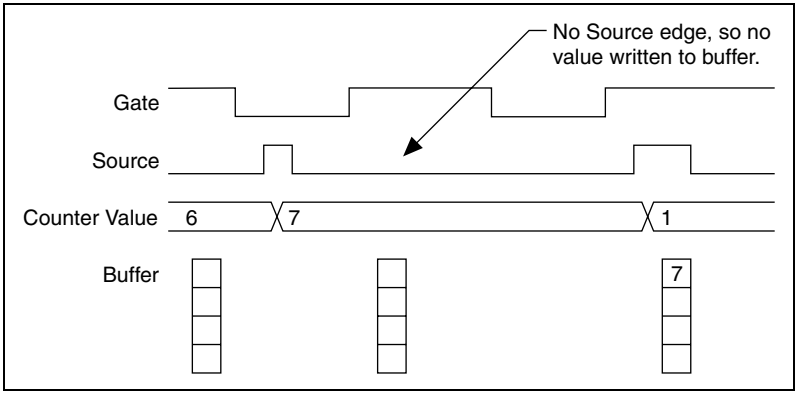


Figure 7-32. Duplicate Count Example

Example Application That Prevents Duplicate Count

With duplicate count prevention enabled, the counter synchronizes both the Source and Gate signals to the 80 MHz Timebase. By synchronizing to the timebase, the counter detects edges on the Gate even if the Source does not pulse. This enables the correct current count to be stored in the buffer even if no Source edges occur between Gate signals, as shown in Figure 7-33.

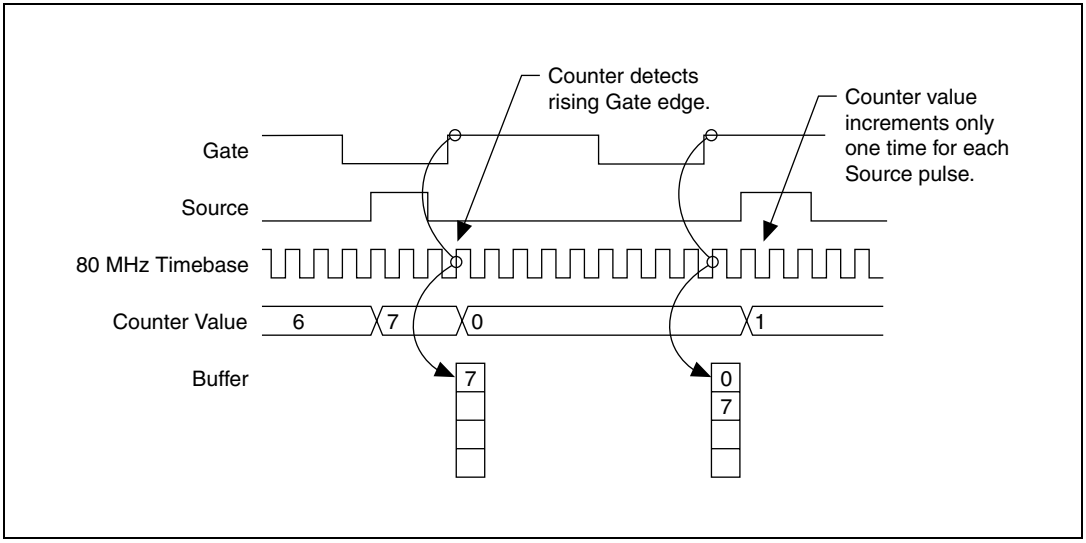


Figure 7-33. Duplicate Count Prevention Example

Even if the Source pulses are long, the counter increments only once for each Source pulse.

Normally, the counter value and Counter *n* Internal Output signals change synchronously to the Source signal. With duplicate count prevention, the counter value and Counter *n* Internal Output signals change synchronously to the 80 MHz Timebase.

Note that duplicate count prevention should only be used if the frequency of the Source signal is 20 MHz or less.

When To Use Duplicate Count Prevention

You should use duplicate count prevention if the following conditions are true.

- You are making a counter measurement
- You are using an external signal (such as PFI <0..5>) as the counter Source
- The frequency of the external source is 20 MHz or less
- You can have the counter value and output to change synchronously with the 80 MHz Timebase

In all other cases, you should *not* use duplicate count prevention.

Enabling Duplicate Count Prevention in NI-DAQmx

You can enable duplicate count prevention in NI-DAQmx by setting the **Enable Duplicate Count Prevention** attribute/property. For specific information about finding the **Enable Duplicate Count Prevention** attribute/property, refer to the help file for the API you are using.

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 M Series devices synchronize these signals before presenting them to the internal counter.

M Series devices use one of three synchronization methods:

- 80 MHz source mode
- Other internal source mode
- External source mode

In DAQmx, the device uses 80 MHz source mode if the user performs the following:

- Performs a position measurement
- Selects duplicate count prevention

Otherwise, the mode depends on the signal that drives Counter *n* Source. Table 7-6 describes the conditions for each mode.

Table 7-6. Synchronization Mode Conditions

Duplicate Count Prevention Enabled	Type of Measurement	Signal Driving Counter *n* Source	Synchronization Mode
Yes	Any	Any	80 MHz Source
No	Position Measurement	Any	80 MHz Source
No	Any	80 MHz Timebase	80 MHz Source
No	All Except Position Measurement	20 MHz Timebase, 100 kHz Timebase, or PXI_CLK10	Other Internal Source
No	All Except Position Measurement	Any Other Signal (such as PFI or RTSI)	External Source

80 MHz Source Mode

In 80 MHz source mode, the device synchronizes signals on the rising edge of the source, and counts on the following rising edge of the source, as shown in Figure 7-34.

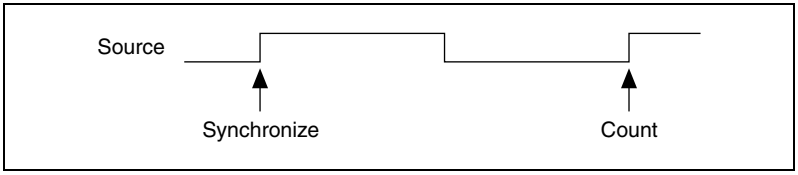


Figure 7-34. 80 MHz Source Mode

Other Internal Source Mode

In other internal source mode, the device synchronizes signals on the falling edge of the source, and counts on the following rising edge of the source, as shown in Figure 7-35.

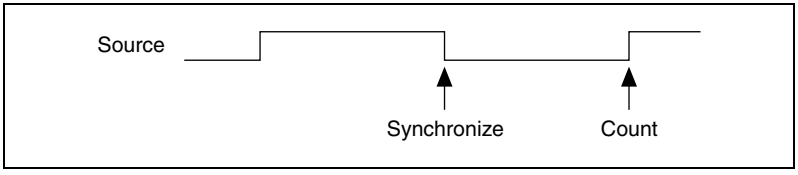


Figure 7-35. Other Internal Source Mode

External Source Mode

In external source mode, the device generates a delayed Source signal by delaying the Source signal by several nanoseconds. The device synchronizes signals on the rising edge of the delayed Source signal, and counts on the following rising edge of the source, as shown in Figure 7-36.

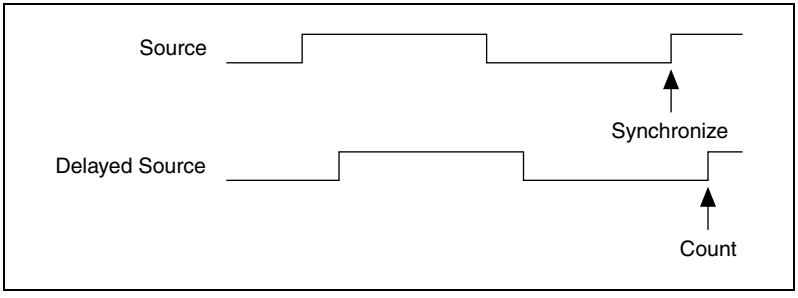


Figure 7-36. External Source Mode

8

PFI

NI 6236 devices have 10 Programmable Function Interface (PFI) signals—six input signals and four output signals.

Each PFI <0..5>/P0.<0..5> can be configured as a timing input signal for AI or counter/timer functions or a static digital input. Each PFI input also has a programmable debouncing filter.

Figure 8-1 shows the circuitry of one PFI input line. Each PFI line is similar.

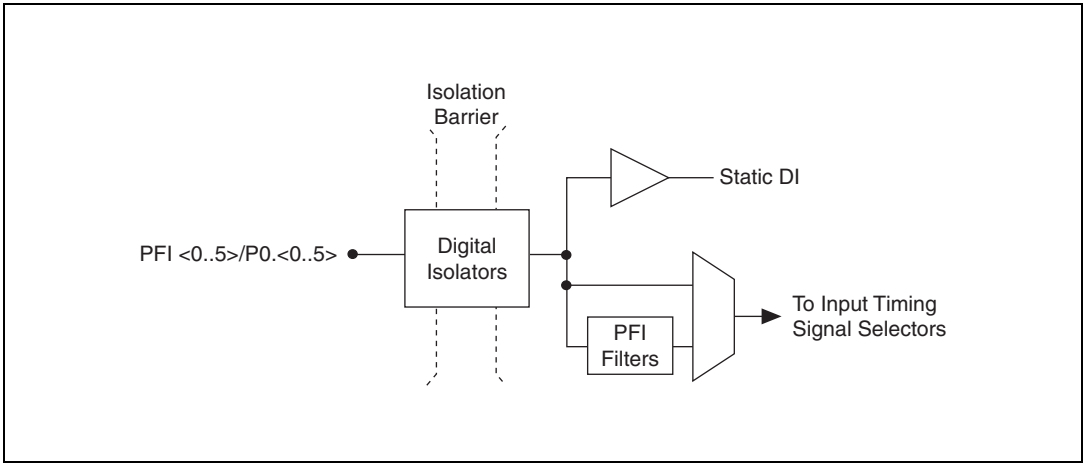


Figure 8-1. NI 6236 PFI Input Circuitry

Each PFI <6..9>/P1.<0..3> can be configured as a timing output signal from AI, AO, or counter/timer functions or a static digital output.

Figure 8-2 shows the circuitry of one PFI output line. Each PFI line is similar.

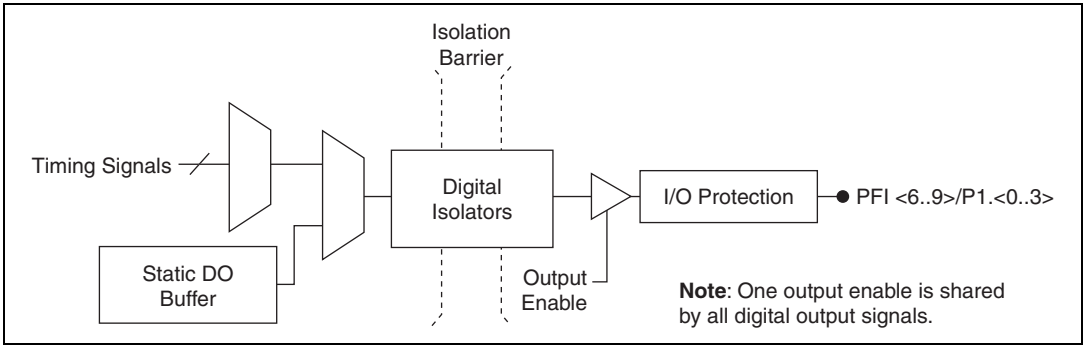


Figure 8-2. NI 6236 PFI Output Circuitry

When a terminal is used as a timing input or output signal, it is called PFI *x* (where *x* is an integer from 0 to 9). When a terminal is used as a static digital input or output, it is called P0.*x* or P1.*x*.

The voltage input and output levels and the current drive levels of the PFI signals are listed in the *NI 6236 Specifications*.

Using PFI Terminals as Timing Input Signals

Use PFI <0..5> terminals to route external timing signals to many different M Series functions. Each input PFI terminal can be routed to any of the following signals.

- AI Convert Clock
- AI Sample Clock
- AI Start Trigger
- AI Reference Trigger
- AI Pause Trigger
- AI Sample Clock Timebase
- AO Start Trigger
- AO Sample Clock
- AO Sample Clock Timebase
- AO Pause Trigger
- Counter input signals for either counter—Source, Gate, Aux, HW_Arm, A, B, Z

Most functions allow you to configure the polarity of PFI inputs and whether the input is edge or level sensitive.

Exporting Timing Output Signals Using PFI Terminals

You can route any of the following timing signals to any PFI <6..9> terminal.

- AI Convert Clock*
- AI Hold Complete Event
- AI Reference Trigger
- AI Sample Clock
- AI Start Trigger
- AO Sample Clock*
- AO Start Trigger
- Counter *n* Source
- Counter *n* Gate
- Counter *n* Internal Output
- Frequency Output
- PXI_STAR
- RTSI <0..7>

Note Signals with a * are inverted before being driven to a terminal.

Using PFI Terminals as Static Digital Inputs and Outputs

When a terminal is used as a static digital input or output, it is called P0.*x* or P1.*x*. On the I/O connector, each terminal is labeled PFI *x*/P0.*x* or PFI *x*/P1.*x*.

Connecting PFI Input Signals

All PFI input connections are referenced to D GND. Figure 8-3 shows this reference, and how to connect an external PFI 0 source and an external PFI 2 source to two PFI terminals.

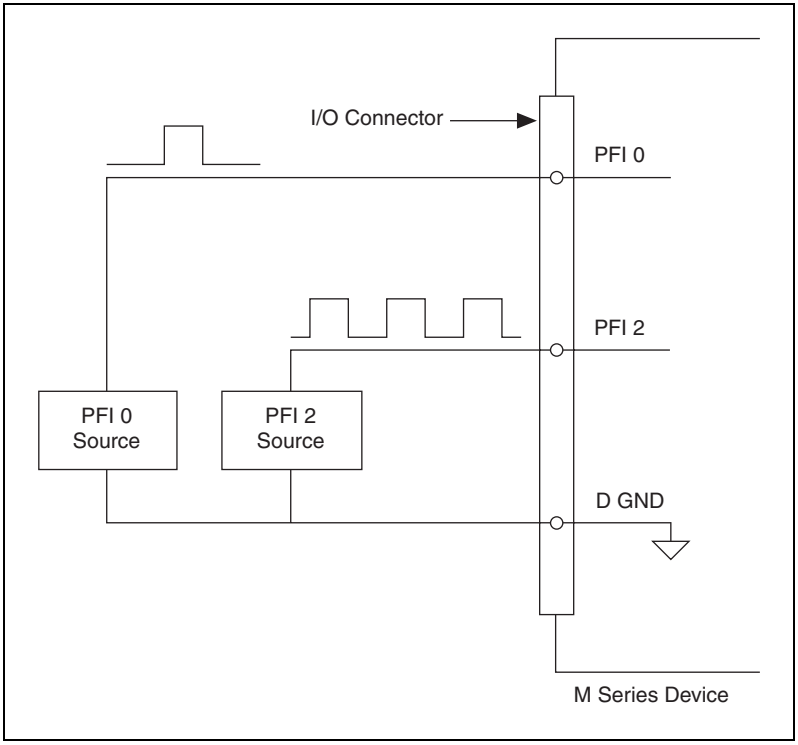


Figure 8-3. PFI Input Signals Connections

PFI Filters

You can enable a programmable debouncing filter on each PFI, RTSI, or PXI_STAR signal. When the filters are enabled, your device samples the input on each rising edge of a filter clock. M Series devices use an onboard oscillator to generate the filter clock with a 40 MHz frequency.

Note NI-DAQmx *only* supports filters on counter inputs.

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; refer to Table 8-1.

Table 8-1. Filters

Filter Setting	N (Filter Clocks Needed to Pass Signal)	Pulse Width Guaranteed to Pass Filter	Pulse Width Guaranteed to Not Pass Filter
125 ns	5	125 ns	100 ns
6.425 µs	257	6.425 µs	6.400 µs
2.55 ms	~101,800	2.55 ms	2.54 ms
Disabled	—	—	—

The filter setting for each input can be configured independently. On power up, the filters are disabled. Figure 8-4 shows an example of a low to high transition on an input that has its filter set to 125 ns (N = 5).

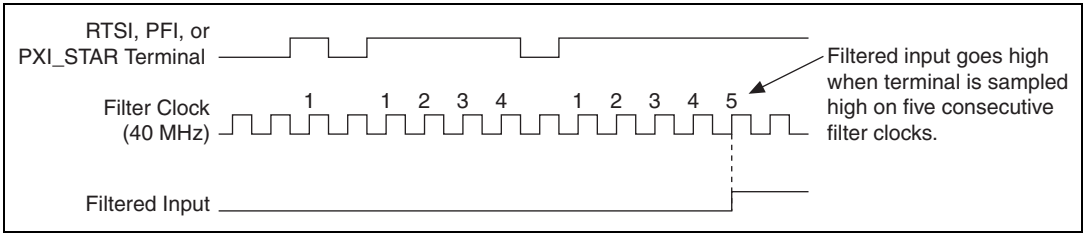


Figure 8-4. Filter Example

Enabling filters introduces jitter on the input signal. For the 125 ns and 6.425 µs filter settings, the jitter is up to 25 ns. On the 2.55 ms setting, the jitter is up to 10.025 µs.

When a PFI input is routed directly to RTSI, or a RTSI input is routed directly to PFI, the M Series device does not use the filtered version of the input signal.

Refer to the KnowledgeBase document, *Digital Filtering with M Series*, for more information about digital filters and counters. To access this KnowledgeBase, go to `ni.com/info` and enter the info code `rddfms`.

I/O Protection

Each DI, DO, and PFI signal is protected against overvoltage, undervoltage, and overcurrent conditions as well as ESD events. However, you should avoid these fault conditions by following these guidelines.

- Do *not* connect any digital output line to any external signal source, ground signal, or power supply.
- Understand the current requirements of the load connected to the digital output lines. Do *not* exceed the specified current output limits of the digital outputs. NI has several signal conditioning solutions for digital applications requiring high current drive.
- Do *not* drive the digital input lines with voltages outside of its normal operating range. The DI, DO, or PFI lines have a smaller operating range than the AI signals.
- Treat the DAQ device as you would treat any static sensitive device. *Always* properly ground yourself and the equipment when handling the DAQ device or connecting to it.

Programmable Power-Up States

By default, the digital output lines (P1.<0..3>/PFI <6..9>) are disabled (high impedance) at power up. Software can configure the board to power up with the entire port enabled or disabled; you cannot enable individual lines. If the port powers up enabled, you also can configure each line individually to power up as 1 or 0.

Refer to the *NI-DAQmx Help* or the *LabVIEW 8.x Help* for more information about setting power-up states in NI-DAQmx or MAX.

9

Isolation and Digital Isolators

NI 6236 devices are isolated data acquisition devices. As shown in Figure 9-1, the analog input, analog output, PFI/static DI, and PFI/static DO are isolated from earth/chassis ground, ⏚ , and are all referenced to the isolated ground, ⏚ . The bus interface circuitry, RTSI, digital routing, and clock generation are all referenced to a *non-isolated ground*, ⏚ .

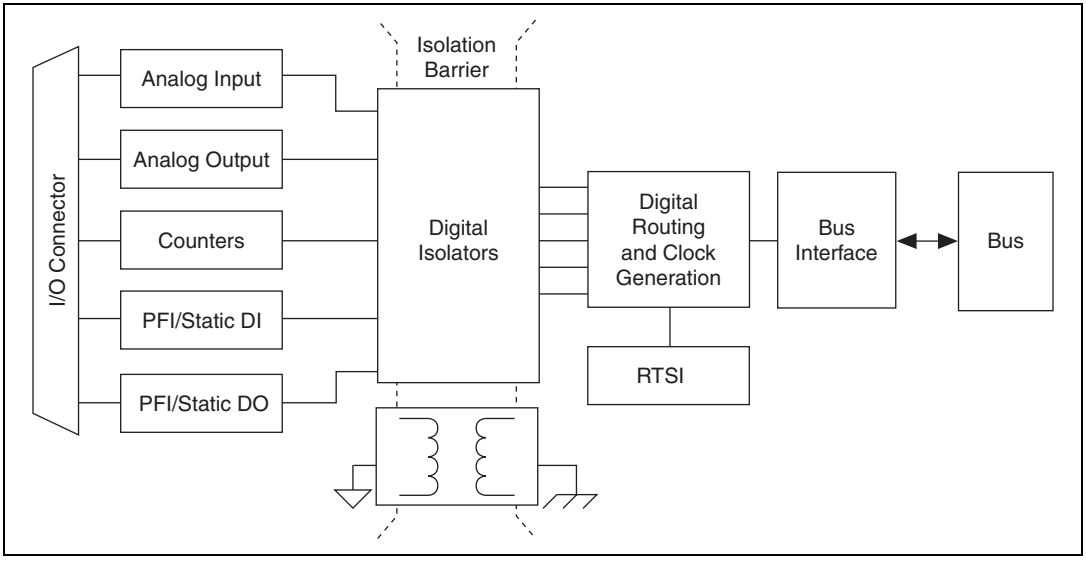


Figure 9-1. General NI 6236 Block Diagram

The non-isolated ground is connected to the chassis ground of the PC or chassis where the device is installed.

The isolated ground is *not* connected to the chassis ground of the PC or chassis. The isolated ground can be at a higher or lower voltage relative to the non-isolated ground. All analog and digital signals are made relative to the isolated ground signal.

The isolated ground is an *input* to the NI 6236 device. The user *must* connect this ground to the ground of the system being measured or controlled. Refer to the *Connecting Analog Current Input Signals* section of Chapter 4, *Analog Input*, the section of Chapter 5, *Analog Output*, the

Connecting Digital I/O Signals section of Chapter 6, *Digital Input and Output*, and the *Connecting PFI Input Signals* section of Chapter 8, *PFI*, for more information.

Digital Isolation

The NI 6236 uses digital isolators. Unlike analog isolators, digital isolators do not introduce any analog error in the measurements taken by the device. The A/D converter, used for analog input, is on the isolated side of the device. The analog inputs are digitized before they are sent across the isolation barrier. Similarly, the D/A converters, used for analog output, are on the isolated side of the device.

Benefits of an Isolated DAQ Device

With isolation, engineers can safely measure a small current in the presence of a large common-mode voltage signal. Some advantages of isolation are as follows:

- **Improved rejection**—Isolation increases the ability of the measurement system to reject common-mode voltages. *Common-mode voltage* is the signal that is present or "common" to both the positive and negative input of a measurement device, but is not part of the signal to be measured.
- **Improved accuracy**—Isolation improves measurement accuracy by physically preventing ground loops. Ground loops, a common source of error and noise, are the result of a measurement system having multiple grounds at different potentials.
- **Improved safety**—Isolation creates an insulation barrier so you can make floating measurements while protecting against large transient voltage spikes.

Reducing Common-Mode Noise

Isolated products require an isolated power supply to deliver power to the isolated side from the non-isolated side. Isolated power supplies work by switching voltages through a transformer with high-speed transistors. Switching voltages through the transformer causes charging and discharging of the parasitic capacitances and inductances in the switching power supplies that occur on every switch cycle, resulting in high-speed currents flowing through the isolated side and returning to the non-isolated side, which is earth ground.

These parasitic currents interact with parasitic and non-parasitic resistances causing voltage spikes. These voltage spikes are called *common-mode noise*, a noise source that travels in the ground and is therefore common to both the ground and any signal referenced to the ground, such as AI, AO, and digital signals. Common-mode noise appears at the harmonics of the switching power supply frequency and can corrupt measurements depending on the system setup.

To reduce common-mode noise:

- Better grounding from the front connector can reduce common-mode noise. Use low resistance cabling and connections and verify that all ground connections are kept short. Keep the number of connections to a minimum. If the device's isolated ground is being connected back to earth ground, verify that this is done in the most direct way possible.
- Reduce source impedances if possible. The parasitic currents react with these impedances.

Creating an AC Return Path

Caution Adding a capacitor will degrade the M Series device withstand voltage and isolation specifications. Withstand voltage *must* be retested by an approved testing facility after adjustments are made to the measurement system.

Non-Isolated Systems

A non-isolated measurement system is one where the device's isolated front end connects to earth ground.

In this case, further noise reduction can be achieved by adding an AC return path from the device isolated ground to earth ground. For non-isolated systems, an AC return path is only needed for high or source impedances. An AC return path can be created by connecting a capacitor between the device's isolated ground and earth ground.

Isolated Systems

A fully isolated measurement system is one where the device's isolated front end is not connected back to earth ground.

Create an AC path back to earth ground from the device's isolated ground by connecting a high voltage capacitor between the isolated board ground and earth ground. The voltage rating of the capacitor must be larger than the voltage drop between the isolated ground and earth ground.

10

Digital Routing and Clock Generation

The digital routing circuitry has the following three main 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 M Series device
  - Other devices in your system through RTSI
  - User input through the PFI terminals
  - User input through the PXI_STAR terminal
- Routes and generates the main clock signals for the M Series device.

Clock Routing

Figure 10-1 shows the clock routing circuitry of an M Series device.

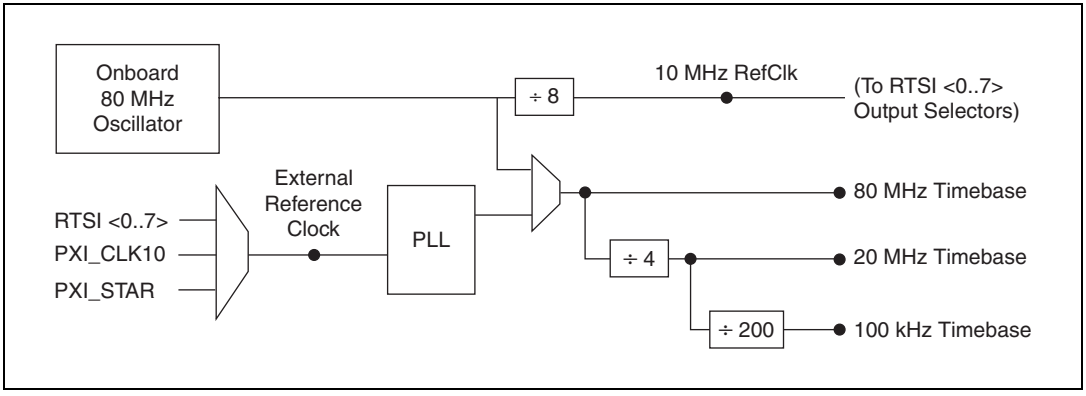


Figure 10-1. M Series Clock Routing Circuitry

80 MHz Timebase

The 80 MHz Timebase can be used as the Source input to the 32-bit general-purpose counter/timers.

The 80 MHz Timebase can be generated from either of the following.

- Onboard oscillator
- External signal (by using the external reference clock)

20 MHz Timebase

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

Your device generates 20 MHz Timebase by dividing down the 80 MHz Timebase.

100 kHz Timebase

The 100 kHz Timebase can be used to generate many of the AI and AO timing signals. The 100 kHz Timebase also can be used as the Source input to the 32-bit general-purpose counter/timers.

The 100 kHz Timebase is generated by dividing down the 20 MHz Timebase by 200.

External Reference Clock

The external reference clock can be used as a source for the internal timebases (80 MHz Timebase, 20 MHz Timebase, and 100 kHz Timebase) on an M Series device. By using the external reference clock, you can synchronize the internal timebases to an external clock.

The following signals can be routed to drive the external reference clock.

- RTSI <0..7>
- PXI_CLK10
- PXI_STAR

The external reference clock is an input to a Phase-Lock Loop (PLL). The PLL generates the internal timebases.

10 MHz Reference Clock

The 10 MHz reference clock can be used to synchronize other devices to your M Series device. The 10 MHz reference clock can be routed to the RTSI <0..7> terminals. Other devices connected to the RTSI bus can use this signal as a clock input.

The 10 MHz reference clock is generated by dividing down the onboard oscillator.

Synchronizing Multiple Devices

With the RTSI bus and the routing capabilities of M Series devices, there are several ways to synchronize multiple devices depending on your application.

To synchronize multiple devices to a common timebase, choose one device—the initiator—to generate the timebase. The initiator device routes its 10 MHz reference clock to one of the RTSI <0..7> signals.

All devices (including the initiator device) receive the 10 MHz Reference Clock from RTSI. This signal becomes the external reference clock. A PLL on each device generates the internal timebases synchronous to the external reference clock.

On PXI systems, you also can synchronize devices to PXI_CLK10. In this application the PXI chassis acts as the initiator. Each PXI module routes PXI_CLK10 to its external reference clock.

Another option in PXI systems is to use PXI_STAR. The Star Trigger controller device acts as the initiator and drives PXI_STAR with a clock signal. Each target device routes PXI_STAR to its external reference clock.

When all of the devices are using or referencing a common timebase, you can synchronize operations across them by sending a common start trigger out across the RTSI bus and setting their sample clock rates to the same value.

Real-Time System Integration Bus (RTSI)

Real-Time System Integration (RTSI) is set of bused signals among devices that allow you to do the following.

- Use a common clock (or timebase) to drive the timing engine on multiple devices
- Share trigger signals between devices

Many National Instruments DAQ, motion, vision, and CAN devices support RTSI.

In a PCI system, the RTSI bus consists of the RTSI bus interface and a ribbon cable. The bus can route timing and trigger signals between several functions on as many as five DAQ, vision, motion, or CAN devices in the computer. In a PXI system, the RTSI bus consists of the RTSI bus interface and the PXI trigger signals on the PXI backplane. This bus can route timing and trigger signals between several functions on as many as seven DAQ devices in the system.

RTSI Connector Pinout

Figure 10-2 shows the RTSI connector pinout and Table 10-1 describes the RTSI signals. The RTSI signals are referenced to earth/chassis ground; they are not isolated.

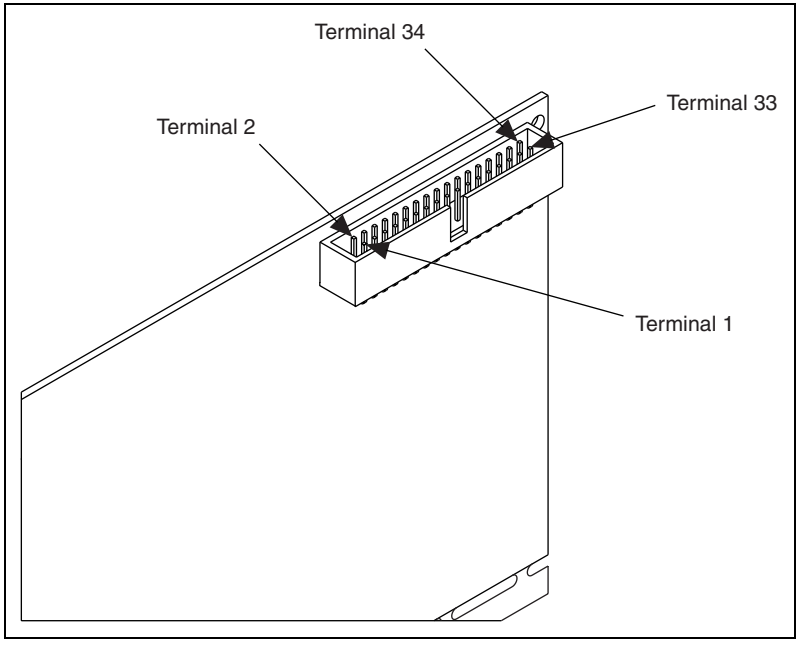


Figure 10-2. NI 6236 RTSI Pinout

Table 10-1. RTSI Signal Descriptions

RTSI Bus Signal	Terminal
RTSI 7	34
RTSI 6	32
RTSI 5	30
RTSI 4	28
RTSI 3	26
RTSI 2	24
RTSI 1	22
RTSI 0	20
Not Connected. Do not connect signals to these terminals.	1–18
GND	19, 21, 23, 25, 27, 29, 31, 33
Note: RTSI <0..7> and GND are earth/chassis ground-referenced. They are *not* isolated.	

Using RTSI as Outputs

RTSI <0..7> are bidirectional terminals. As an output, you can drive any of the following signals to any RTSI terminal.

- ai/StartTrigger
- ai/ReferenceTrigger
- ai/ConvertClock*
- ai/SampleClock
- ai/PauseTrigger
- ao/SampleClock*
- ao/StartTrigger
- ao/PauseTrigger
- 10 MHz Reference Clock
- Counter *n* Source, Gate, Z, Internal Output
- FREQ OUT
- Input PFI <0..5>

Note Signals with a * are inverted before being driven on the RTSI terminals.

Using RTSI Terminals as Timing Input Signals

You can use RTSI terminals to route external timing signals to many different M Series functions. Each RTSI terminal can be routed to any of the following signals.

- AI Convert Clock
- AI Sample Clock
- AI Start Trigger
- AI Reference Trigger
- AI Pause Trigger
- AI Sample Clock Timebase
- AO Start Trigger
- AO Sample Clock
- AO Sample Clock Timebase
- AO Pause Trigger
- Counter input signals for either counter—Source, Gate, Aux, HW_Arm, A, B, or Z

Most functions allow you to configure the polarity of PFI inputs and whether the input is edge or level sensitive.

RTSI Filters

You can enable a programmable debouncing filter on each PFI, RTSI, or PXI_STAR signal. When the filters are enabled, your device samples the input on each rising edge of a filter clock. M Series devices use an onboard oscillator to generate the filter clock with a 40 MHz frequency.

Note NI-DAQmx *only* supports filters on counter inputs.

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; refer to Table 10-2.

Table 10-2. Filters

Filter Setting	N (Filter Clocks Needed to Pass Signal)	Pulse Width Guaranteed to Pass Filter	Pulse Width Guaranteed to Not Pass Filter
125 ns	5	125 ns	100 ns
6.425 µs	257	6.425 µs	6.400 µs
2.55 ms	~101,800	2.55 ms	2.54 ms
Disabled	—	—	—

The filter setting for each input can be configured independently. On power up, the filters are disabled. Figure 10-3 shows an example of a low to high transition on an input that has its filter set to 125 ns (N = 5).

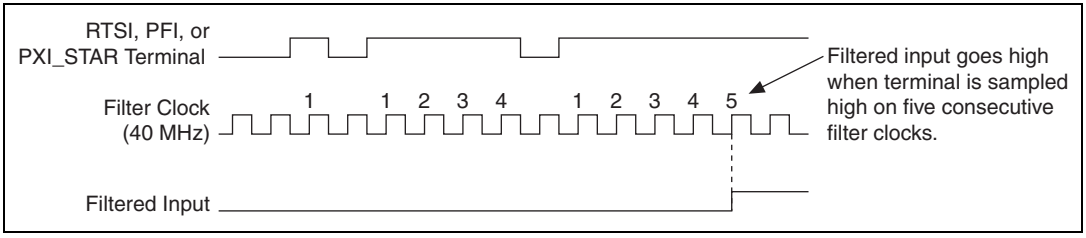


Figure 10-3. Filter Example

Enabling filters introduces jitter on the input signal. For the 125 ns and 6.425 µs filter settings, the jitter is up to 25 ns. On the 2.55 ms setting, the jitter is up to 10.025 µs.

When a PFI input is routed directly to RTSI, or a RTSI input is routed directly to PFI, the M Series device does not use the filtered version of the input signal.

Refer to the KnowledgeBase document, *Digital Filtering with M Series*, for more information about digital filters and counters. To access this KnowledgeBase, go to `ni.com/info` and enter the info code `rddfms`.

PXI Clock and Trigger Signals

Note PXI clock and trigger signals are only available on PXI devices. Other devices use RTSI.

PXI_CLK10

PXI_CLK10 is a common low-skew 10 MHz clock reference clock for synchronization of multiple modules in a PXI measurement or control system. The PXI backplane is responsible for generating PXI_CLK10 independently to each peripheral slot in a PXI chassis.

PXI Triggers

A PXI chassis provides eight bused trigger lines to each module in a system. Triggers may be passed from one module to another, allowing precisely timed responses to asynchronous external events that are being monitored or controlled. Triggers can be used to synchronize the operation of several different PXI peripheral modules.

On M Series devices, the eight PXI trigger signals are synonymous with RTSI <0..7>.

Note that in a PXI chassis with more than eight slots, the PXI trigger lines may be divided into multiple independent buses. Refer to the documentation for your chassis for details.

PXI_STAR Trigger

In a PXI system, the Star Trigger bus implements a dedicated trigger line between the first peripheral slot (adjacent to the system slot) and the other peripheral slots. The Star Trigger can be used to synchronize multiple devices or to share a common trigger signal among devices.

A Star Trigger controller can be installed in this first peripheral slot to provide trigger signals to other peripheral modules. Systems that do not require this functionality can install any standard peripheral module in this first peripheral slot.

An M Series device receives the Star Trigger signal (PXI_STAR) from a Star Trigger controller. PXI_STAR can be used as an external source for many AI, AO, and counter signals.

An M Series device is not a Star Trigger controller. An M Series device may be used in the first peripheral slot of a PXI system, but the system will not be able to use the Star Trigger feature.

PXI_STAR Filters

You can enable a programmable debouncing filter on each PFI, RTSI, or PXI_STAR signal. When the filters are enabled, your device samples the input on each rising edge of a filter clock. M Series devices use an onboard oscillator to generate the filter clock with a 40 MHz frequency.

Note NI-DAQmx *only* supports filters on counter inputs.

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; refer to Table 10-3.

Table 10-3. Filters

Filter Setting	N (Filter Clocks Needed to Pass Signal)	Pulse Width Guaranteed to Pass Filter	Pulse Width Guaranteed to Not Pass Filter
125 ns	5	125 ns	100 ns
6.425 µs	257	6.425 µs	6.400 µs
2.55 ms	~101,800	2.55 ms	2.54 ms
Disabled	—	—	—

The filter setting for each input can be configured independently. On power up, the filters are disabled. Figure 10-4 shows an example of a low to high transition on an input that has its filter set to 125 ns (N = 5).

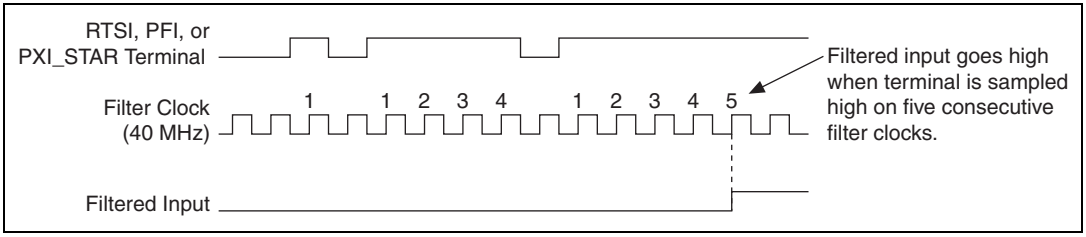


Figure 10-4. Filter Example

Enabling filters introduces jitter on the input signal. For the 125 ns and 6.425 µs filter settings, the jitter is up to 25 ns. On the 2.55 ms setting, the jitter is up to 10.025 µs.

When a PFI input is routed directly to RTSI, or a RTSI input is routed directly to PFI, the M Series device does not use the filtered version of the input signal.

Refer to the KnowledgeBase document, *Digital Filtering with M Series*, for more information about digital filters and counters. To access this KnowledgeBase, go to `ni.com/info` and enter the info code `rddfms`.

11

Bus Interface

The bus interface circuitry of NI 6236 devices efficiently moves data between host memory and the measurement and acquisition circuits. NI 6236 devices are available for the following platforms.

- PCI
- PXI

NI 6236 devices are jumperless for complete plug-and-play operation. The operating system automatically assigns the base address, interrupt levels, and other resources.

NI 6236 devices incorporate PCI-MITE technology to implement a high-performance PCI interface.

DMA Controllers

NI 6236 devices have four fully-independent DMA controllers for high-performance transfers of data blocks. One DMA controller is available for each measurement and acquisition block.

- Analog input
- Analog output
- Counter 0
- Counter 1

Each DMA controller channel contains a FIFO and independent processes for filling and emptying the FIFO. This allows the buses involved in the transfer to operate independently for maximum performance. Data is transferred simultaneously between the ports. The DMA controller supports burst transfers to and from the FIFO.

Each DMA controller acts as a PCI Master device. The DMA controllers support scatter-gather operations to and from host memory. Memory buffers may be used in linear or circular fashion.

Each DMA controller supports packing and unpacking of data through the FIFOs to connect different size devices and optimize PCI bus utilization and automatically handles unaligned memory buffers.

PXI Considerations

Note PXI clock and trigger signals are only available on PXI devices. Other devices use RTSI.

PXI Clock and Trigger Signals

Refer to the *PXI_CLK10*, *PXI Triggers*, *PXI_STAR Trigger*, and *PXI_STAR Filters* sections of Chapter 10, *Digital Routing and Clock Generation*, for more information about PXI clock and trigger signals.

PXI and PXI Express

NI PXI-6236 modules can be installed in any PXI chassis and most slots of PXI Express chassis.

PXI specifications are developed by the PXI System Alliance (`www.pxisa.org`). Using the terminology of the PXI specifications, NI PXI-6236 devices are *3U Hybrid Slot-Compatible PXI-1 Peripheral Modules*.

3U designates devices that are 100 mm tall (as opposed to the taller 6U modules).

Hybrid slot-compatible defines where the device can be installed. PXI-6236 devices can be installed in the following chassis and slots:

- **PXI chassis**—PXI-6236 devices can be installed in any peripheral slot of a PXI chassis.
- **PXI Express chassis**—PXI-6236 devices can be installed in the following PXI Express chassis slots:
  - **PXI-1 slots**—Accepts all PXI modules
  - **PXI hybrid slots**—Accepts PXI or PXI Express modules.

PXI-1 devices use PCI signaling to communicate to the host controller (as opposed to PCI Express signaling).

Peripheral devices are installed in peripheral slots and are not system controllers.

Using PXI with CompactPCI

Using PXI-compatible products with standard CompactPCI products is an important feature provided by *PXI Hardware Specification Revision 2.1*. If you use a PXI-compatible plug-in module in a standard CompactPCI chassis, you cannot use PXI-specific functions, but you can still use the basic plug-in device functions. For example, the RTSI bus on a PXI M Series device is available in a PXI chassis, but not in a CompactPCI chassis.

The CompactPCI specification permits vendors to develop sub-buses that coexist with the basic PCI interface on the CompactPCI bus. Compatible operation is not guaranteed between CompactPCI devices with different sub-buses nor between CompactPCI devices with sub-buses and PXI. The standard implementation for CompactPCI does not include these sub-buses. The PXI M Series device works in any standard CompactPCI chassis adhering to the *PICMG CompactPCI 2.0 R3.0* core specification.

PXI-specific features are implemented on the J2 connector of the CompactPCI bus. The PXI device is compatible with any CompactPCI chassis with a sub-bus that does not drive the lines used by that device. Even if the sub-bus is capable of driving these lines, the PXI device is still compatible as long as those terminals on the sub-bus are disabled by default and never enabled.

Caution Damage can result if these lines are driven by the sub-bus. NI is *not* liable for any damage resulting from improper signal connections.

Data Transfer Methods

There are three primary ways to transfer data across the PCI bus—direct memory access (DMA), interrupt request (IRQ), and programmed I/O.

Direct Memory Access (DMA)

DMA is a method to transfer data between the device and computer memory without the involvement of the CPU. This method makes DMA the fastest available data transfer method. National Instruments uses DMA hardware and software technology to achieve high throughput rates and to increase system utilization. DMA is the default method of data transfer for DAQ devices that support it.

Interrupt Request (IRQ)

IRQ transfers rely on the CPU to service data transfer requests. The device notifies the CPU when it is ready to transfer data. The data transfer speed is tightly coupled to the rate at which the CPU can service the interrupt requests. If you are using interrupts to transfer data at a rate faster than the rate the CPU can service the interrupts, your systems may start to freeze.

Programmed I/O

Programmed I/O is a data transfer mechanism where the user's program is responsible for transferring data. Each read or write call in the program initiates the transfer of data. Programmed I/O is typically used in software-timed (on-demand) operations. Refer to the *Software-Timed Generations* section of Chapter 5, *Analog Output*, for more information.

Changing Data Transfer Methods between DMA and IRQ

On PCI or PXI M Series devices, each measurement and acquisition circuit (that is, AI, AO, and so on) has a dedicated DMA channel. So in most applications, all data transfers use DMA.

However, NI-DAQmx allows you to disable DMA and use interrupts. To change your data transfer mechanism between DMA and interrupts in NI-DAQmx, use the **Data Transfer Mechanism** property node.

12

Triggering

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. NI 6236 devices support internal software triggering, as well as external digital triggering. For information about the different actions triggers can perform for each sub-system of the device, refer to the following sections:

- The *Analog Input Triggering* section of Chapter 4, *Analog Input*
- The *Analog Output Triggering* section of Chapter 5, *Analog Output*
- The *Counter Triggering* section of Chapter 7, *Counters*

Triggering with a Digital Source

Your DAQ device can generate a trigger on a digital signal. You must specify a source and an edge. The digital source can be any of the PFI, RTSI, or PXI_STAR signals.

The edge can be either the rising edge or falling edge of the digital signal. A rising edge is a transition from a low logic level to a high logic level. A falling edge is a high to low transition.

Figure 12-1 shows a falling-edge trigger.

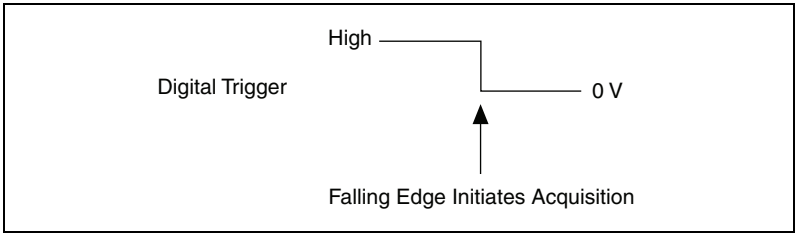


Figure 12-1. Falling-Edge Trigger

You also can program your DAQ device to perform an action in response to a trigger from a digital source. The action can affect the following.

- Analog input acquisition
- Analog output generation
- Counter behavior

A

NI 6236 Device Information

This appendix contains device pinouts, specifications, cable and accessory choices, and other information for the NI 6236 M Series devices.

To obtain documentation for devices not listed here, refer to `ni.com/manuals`.

NI 6236 Pinout

Figure A-1 shows the pinout of the NI 6236.

For a detailed description of each signal, refer to the *I/O Connector Signal Descriptions* section of Chapter 3, *Connector Information*.

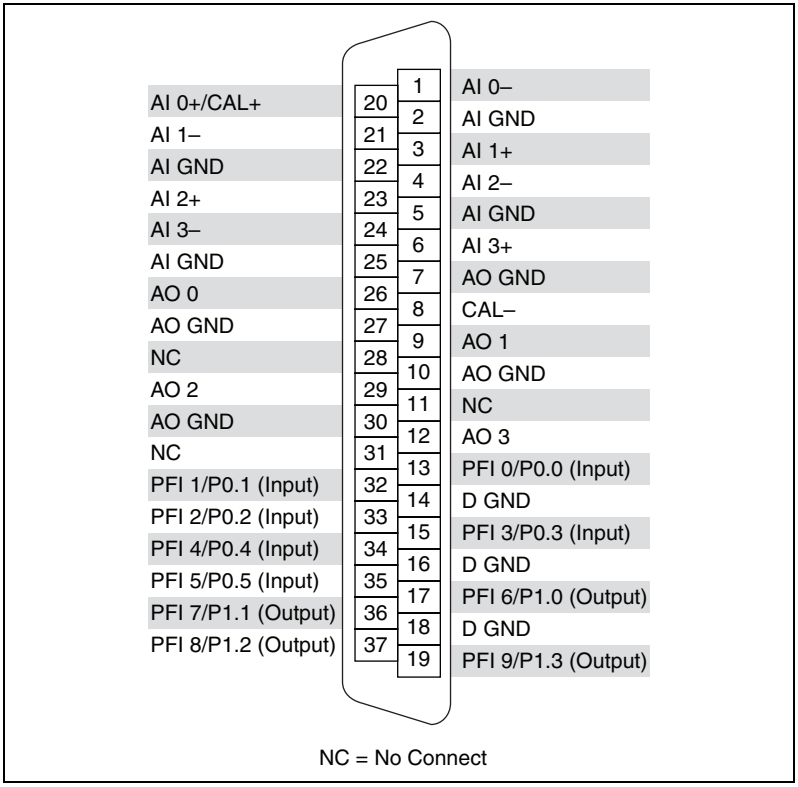


Figure A-1. NI 6236 Pinout

Table A-1. NI 6236 Device Default NI-DAQmx Counter/Timer Pins

Counter/Timer Signal	Default Pin Number (Name)	Port
CTR 0 SRC	13 (PFI 0)	P0.0
CTR 0 GATE	32 (PFI 1)	P0.1
CTR 0 AUX	33 (PFI 2)	P0.2
CTR 0 OUT	17 (PFI 6)	P1.0
CTR 0 A	13 (PFI 0)	P0.0
CTR 0 Z	32 (PFI 1)	P0.1
CTR 0 B	33 (PFI 2)	P0.2
CTR 1 SRC	15 (PFI 3)	P0.3
CTR 1 GATE	34 (PFI 4)	P0.4
CTR 1 AUX	35 (PFI 5)	P0.5

Table A-1. NI 6236 Device Default NI-DAQmx Counter/Timer Pins (Continued)

Counter/Timer Signal	Default Pin Number (Name)	Port
CTR 1 OUT	36 (PFI 7)	P1.1
CTR 1 A	15 (PFI 3)	P0.3
CTR 1 Z	34 (PFI 4)	P0.4
CTR 1 B	35 (PFI 5)	P0.5
FREQ OUT	37 (PFI 8)	P1.2

Note For more information about default NI-DAQmx counter inputs, refer to *Connecting Counter Signals* in the *NI-DAQmx Help* or the *LabVIEW 8.x Help*.

NI 6236 Specifications

Refer to the *NI 6236 Specifications*, available on the NI-DAQ Device Document Browser or `ni.com/manuals`, for more detailed information about the NI 6236 device.

NI 6236 Accessory and Cabling Options

This section describes some cable and accessory options for the NI 6236 device. Refer to `ni.com` for other accessory options including new devices.

Screw Terminal

National Instruments offers several styles of screw terminal connector blocks. Use an SH37F-37M cable to connect an NI 6236 device to a connector block, such as the following:

- **CB-37F-HVD**—37-pin DIN rail screw terminal block, UL Recognized derated to 30 Vrms, 42.4 V_{pk}, or 60 VDC
- **CB-37FH**—Horizontal DIN-mountable terminal block with 37 screw terminals
- **CB-37FV**—Vertical DIN-mountable terminal block with 37 screw terminals
- **CB-37F-LP**—Low profile terminal block with 37 screw terminals
- **TB-2621**—37-pin PXI screw terminal block, UL Recognized derated to 30 Vrms, 42.4 V_{pk}, or 60 VDC

RTSI

Use RTSI bus cables to connect timing and synchronization signals among PCI devices, such as M Series, E Series, CAN, and other measurement, vision, and motion devices. Since PXI devices use PXI backplane signals for timing and synchronization, no cables are required.

Cables

In most applications, you can use the following cables:

- **DB37M-DB37F-EP**—37-pin male-to-female twisted-pair shielded I/O cable
- **SH37F-37M-*x***—37-pin female-to-male shielded I/O cable, UL Listed derated to 30 Vrms, 42.4 V_{pk}, or 60 VDC
- **R37F-37M-1**—37-pin female-to-male ribbon I/O cable
- **SH37F-P-4**—37-pin female-to-pigtails shielded I/O cable

Custom Cabling and Connectivity

Refer to the *Custom Cabling* section of Chapter 2, *DAQ System Overview*, for more information about custom cabling solutions.

B

Troubleshooting

This section contains some common questions about M Series devices. If your questions are not answered here, refer to the National Instruments KnowledgeBase at `ni.com/kb`. It contains thousands of documents that answer frequently asked questions about NI products.

Analog Input

I am seeing crosstalk or ghost current when sampling multiple channels. What does this mean?

You may be experiencing a phenomenon called *charge injection*, which occurs when you sample a series of high-output impedance sources with a multiplexer. Multiplexers contain switches, usually made of switched capacitors. When a channel, for example AI 0, is selected in a multiplexer, those capacitors accumulate charge. When the next channel, for example AI 1, is selected, the accumulated current (or charge) leaks backward through channel 1. If the output impedance of the source connected to AI 1 is high enough, the resulting reading can somewhat affect the current in AI 0. To circumvent this problem, use a current follower that has operational amplifiers (op-amps) with unity gain for each high-impedance source before connecting to an M Series device. Otherwise, you must decrease the sample rate for each channel.

Another common cause of channel crosstalk is due to sampling among multiple channels at various gains. In this situation, the settling times can increase. For more information about charge injection and sampling channels at different gains, refer to the *Multichannel Scanning Considerations* section of Chapter 4, *Analog Input*.

I am using my device in analog input ground-reference mode and I have connected a differential input signal, but my readings are random and drift rapidly. What is wrong?

In DIFF mode, if the readings from the DAQ device are random and drift rapidly, you should check the ground-reference connections. The signal can be referenced to a level that is considered floating with reference to the device ground reference. Even if you are in DIFF mode, you must still

reference the signal to the same ground level as the device reference. There are various methods of achieving this reference while maintaining a high common-mode rejection ratio (CMRR). These methods are outlined in the *Connecting Analog Current Input Signals* section of Chapter 4, *Analog Input*.

How can I use the AI Sample Clock and AI Convert Clock signals on an M Series device to sample the AI channel(s)?

M Series devices use ai/SampleClock and ai/ConvertClock to perform interval sampling. As Figure B-1 shows, ai/SampleClock controls the sample period, which is determined by the following equation:

1/sample period = sample rate

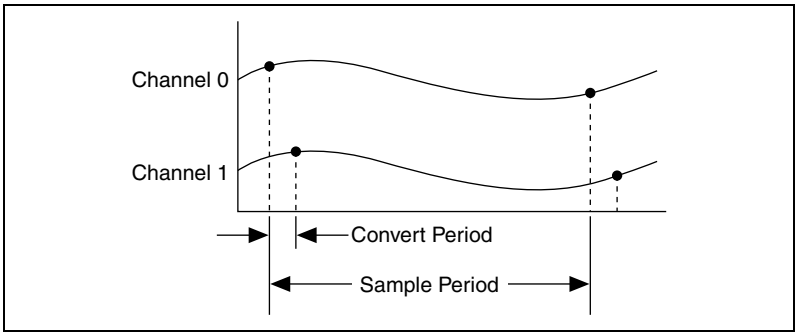


Figure B-1. ai/SampleClock and ai/ConvertClock

ai/ConvertClock controls the convert period, which is determined by the following equation:

1/convert period = convert rate

This method allows multiple channels to be sampled relatively quickly in relationship to the overall sample rate, providing a nearly simultaneous effect with a fixed delay between channels.

Analog Output

I am seeing glitches on the output signal. How can I minimize it?

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. Visit `ni.com/support` for more information about minimizing glitches.

Counters

When multiple sample clocks on my buffered counter measurement occur before consecutive edges on my source, I see weird behavior. Why?

Duplicate count prevention ensures that the counter returns correct data for counter measurement in some applications where a slow or non-periodic external source is used.

Refer to the *Duplicate Count Prevention* section of Chapter 7, *Counters*, for more information.

How do I connect counter signals to my M Series device?

The *Default Counter Terminals* section of Chapter 7, *Counters*, has information about counter signal connections.

C

Technical Support and Professional Services

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

- **Support**—Online technical support resources at ni.com/support include the following:
  - **Self-Help Resources**—For answers and solutions, visit the award-winning National Instruments Web site for software drivers and updates, a searchable KnowledgeBase, product manuals, step-by-step troubleshooting wizards, thousands of example programs, tutorials, application notes, instrument drivers, and so on.
  - **Free Technical Support**—All registered users receive free Basic Service, which includes access to hundreds of Application Engineers worldwide in the NI Developer Exchange at ni.com/exchange. National Instruments Application Engineers make sure every question receives an answer.

    For information about other technical support options in your area, visit ni.com/services or contact your local office at ni.com/contact.
- **Training and Certification**—Visit ni.com/training for self-paced training, eLearning virtual classrooms, interactive CDs, and Certification program information. You also can register for instructor-led, hands-on courses at locations around the world.
- **System Integration**—If you have time constraints, limited in-house technical resources, or other project challenges, National Instruments Alliance Partner members can help. To learn more, call your local NI office or visit ni.com/alliance.
- **Declaration of Conformity (DoC)**—A DoC is our claim of compliance with the Council of the European Communities using the manufacturer's declaration of conformity. This system affords the user protection for electronic compatibility (EMC) and product safety. You can obtain the DoC for your product by visiting ni.com/certification.

- **Calibration Certificate**—If your product supports calibration, you can obtain the calibration certificate for your product at `ni.com/calibration`.

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

Glossary

Symbols

%	Percent.
+	Positive of, or plus.
–	Negative of, or minus.
±	Plus or minus.
<	Less than.
>	Greater than.
≤	Less than or equal to.
≥	Greater than or equal to.
/	Per.
º	Degree.
Ω	Ohm.

A

A	Amperes—the unit of electric current.
A/D	Analog-to-Digital. Most often used as A/D converter.
AC	Alternating current.
accuracy	A measure of the capability of an instrument or sensor to faithfully indicate the value of the measured signal. This term is not related to resolution; however, the accuracy level can never be better than the resolution of the instrument.
ADE	Application development environment.

AI	1. Analog input. 2. Analog input channel signal.
AI GND	Analog input ground signal.
AI SENSE	Analog input sense signal.
analog	A signal whose amplitude can have a continuous range of values.
analog input signal	An input signal that varies smoothly over a continuous range of values, rather than in discrete steps.
analog output signal	An output signal that varies smoothly over a continuous range of values, rather than in discrete steps.
analog signal	A signal representing a variable that can be observed and represented continuously.
analog trigger	A trigger that occurs at a user-selected point on an incoming analog signal. Triggering can be set to occur at a specific level on either an increasing or a decreasing signal (positive or negative slope). Analog triggering can be implemented either in software or in hardware. When implemented in software (LabVIEW), all data is collected, transferred into system memory, and analyzed for the trigger condition. When analog triggering is implemented in hardware, no data is transferred to system memory until the trigger condition has occurred.
AO	Analog output.
AO 0	Analog channel 0 output signal.
AO 1	Analog channel 1 output signal.
AO 2	Analog channel 2 output signal.
AO 3	Analog channel 3 output signal.
AO GND	Analog output ground signal.
application	A software program that creates an end-user function.
arm	The process of getting an instrument ready to perform a function. For example, the trigger circuitry of a digitizer is armed, meaning that it is ready to start acquiring data when an appropriate trigger condition is met.

ASIC — Application-specific integrated circuit—A proprietary semiconductor component designed and manufactured to perform a set of specific functions for a specific customer.

asynchronous
1. Hardware—A property of an event that occurs at an arbitrary time, without synchronization to a reference clock.
2. Software—A property of a function that begins an operation and returns prior to the completion or termination of the operation.

B

b — Bit—One binary digit, either 0 or 1.

B — Byte—Eight related bits of data, an eight-bit binary number. Also used to denote the amount of memory required to store one byte of data.

block diagram — A pictorial description or representation of a program or algorithm.

BNC — Bayonet-Neill-Concelman—A type of coaxial connector used in situations requiring shielded cable for signal connections and/or controlled impedance applications.

buffer
1. Temporary storage for acquired or generated data.
2. A memory device that stores intermediate data between two devices.

bus, buses — The group of electrical conductors that interconnect individual circuitry in a computer. Typically, a bus is the expansion vehicle to which I/O or other devices are connected. Examples of PC buses are the PCI, AT(ISA), and EISA bus.

C

C — Celsius.

calibration — The process of determining the accuracy of an instrument. In a formal sense, calibration establishes the relationship of an instrument's measurement to the value provided by a standard. When that relationship is known, the instrument may then be adjusted (calibrated) for best accuracy.

calibrator	A precise, traceable signal source used to calibrate instruments.
cascading	Process of extending the counting range of a counter chip by connecting to the next higher counter.
CE	European emissions control standard.
channel	Pin or wire lead to which you apply or from which you read the analog or digital signal. For digital signals, you group channels to form ports. Ports usually consist of either four or eight digital channels.
clock	Hardware component that controls timing for reading from or writing to groups.
CMOS	Complementary metal-oxide semiconductor.
CMRR	Common-mode rejection ratio—A measure of the ability of a differential amplifier to reject interference from a common-mode signal, usually expressed in decibels (dB).
common-mode rejection	The ability of an electronic system to cancel any electronic noise pick-up that is common to both the positive and negative polarities of the input leads to the instrument front end. Common mode rejection is only a relevant specification for systems having a balanced or differential input.
common-mode signal	1. Any voltage present at the instrumentation amplifier inputs with respect to amplifier ground. 2. The signal, relative to the instrument chassis or computer's ground, of the signals from a differential input. This is often a noise signal, such as 50 or 60 Hz hum.
connector	1. A device that provides electrical connection. 2. A fixture (either male or female) attached to a cable or chassis for quickly making and breaking one or more circuits. A symbol that connects points on a flowchart.
convert rate	Reciprocal of the interchannel delay.
count	The number of events, such as zero crossings, pulses, or cycles.

counter
1. Software. A memory location used to store a count of certain occurrences.
2. Hardware. A circuit that counts events. When it refers to an instrument, it refers to a frequency counter.

counter/timer — A circuit that counts external pulses or clock pulses (timing).

D

D GND — Digital ground signal.

D-SUB connector — A serial connector.

DAC — Digital-to-Analog Converter—An electronic device, often an integrated circuit, that converts a digital number into a corresponding analog voltage or current.

In the instrumentation world, DACs can be used to generate arbitrary waveform shapes, defined by the software algorithm that computes the digital data pattern, which is fed to the DAC.

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

DAQ device — A device that acquires or generates data and can contain multiple channels and conversion devices. DAQ devices include plug-in devices, PCMCIA cards, and DAQPad devices, which connect to a computer USB or 1394 (FireWire®) port. SCXI modules are considered DAQ devices.

DAQ-STC2 — Data acquisition system timing controller chip.

data acquisition — The general concept of acquiring data, as in *begin data acquisition* or *data acquisition and control*. *See also* DAQ.

data transfer	A technique for moving digital data from one system to another. Options for data transfer are DMA, interrupt, and programmed I/O. For programmed I/O transfers, the CPU in the PC reads data from the DAQ device whenever the CPU receives a software signal to acquire a single data point. Interrupt-based data transfers occur when the DAQ device sends an interrupt to the CPU, telling the CPU to read the acquired data from the DAQ device. DMA transfers use a DMA controller, instead of the CPU, to move acquired data from the device into computer memory. Even though high-speed data transfers can occur with interrupt and programmed I/O transfers, they require the use of the CPU to transfer data. DMA transfers are able to acquire data at high speeds and keep the CPU free for performing other tasks at the same time.
dB	Decibel—The unit for expressing a logarithmic measure of the ratio of two signal levels: dB = 20log10 V1/V2, for signals in volts.
DC	Direct current—although the term speaks of current, many different types of DC measurements are made, including DC Voltage, DC current, and DC power.
device	A plug-in data acquisition product, card, or pad that can contain multiple channels and conversion devices. Plug-in products, PCMCIA cards, and devices such as the DAQPad-1200, which connects to your computer parallel port, are all examples of DAQ devices. SCXI modules are distinct from devices, with the exception of the SCXI-1200, which is a hybrid.
DIFF	Differential mode—An analog input mode consisting of two terminals, both of which are isolated from computer ground, whose difference is measured.
differential input	An input circuit that actively responds to the difference between two terminals, rather than the difference between one terminal and ground. Often associated with balanced input circuitry, but also may be used with an unbalanced source.
digital I/O	The capability of an instrument to generate and acquire digital signals. Static digital I/O refers to signals where the values are set and held, or rarely change. Dynamic digital I/O refers to digital systems where the signals are continuously changing, often at multi-MHz clock rates.

digital isolator	Provides voltage isolation between its input and output.
digital signal	A representation of information by a set of discrete values according to a prescribed law. These values are represented by numbers.
digital trigger	A TTL level signal having two discrete levels—a high and a low level.
DIO	Digital input/output.
DMA	Direct Memory Access—A method by which data can be transferred to/from computer memory from/to a device or memory on the bus while the processor does something else. DMA is the fastest method of transferring data to/from computer memory.
DMA controller chip	Performs the transfers between memory and I/O devices independently of the CPU.
driver	Software unique to the device or type of device, and includes the set of commands the device accepts.

E

edge detection	A technique that locates an edge of an analog signal, such as the edge of a square wave.
EEPROM	Electrically Erasable Programmable Read-Only Memory—ROM that can be erased with an electrical signal and reprogrammed. Some SCXI modules contain an EEPROM to store measurement-correction coefficients.
encoder	A device that converts linear or rotary displacement into digital or pulse signals. The most popular type of encoder is the optical encoder, which uses a rotating disk with alternating opaque areas, a light source, and a photodetector.
external trigger	A voltage pulse from an external source that causes a DAQ operation to begin.
EXTREF	External reference signal.

F

FIFO	First-In-First-Out memory buffer—A data buffering technique that functions like a shift register where the oldest values (first in) come out first. Many DAQ products and instruments use FIFOs to buffer digital data from an A/D converter, or to buffer the data before or after bus transmission. The first data stored is the first data sent to the acceptor. FIFOs are often used on DAQ devices to temporarily store incoming or outgoing data until that data can be retrieved or output. For example, an analog input FIFO stores the results of A/D conversions until the data can be retrieved into system memory, a process that requires the servicing of interrupts and often the programming of the DMA controller. This process can take several milliseconds in some cases. During this time, data accumulates in the FIFO for future retrieval. With a larger FIFO, longer latencies can be tolerated. In the case of analog output, a FIFO permits faster update rates, because the waveform data can be stored on the FIFO ahead of time. This again reduces the effect of latencies associated with getting the data from system memory to the DAQ device.
filter	A physical device or digital algorithm that selectively removes noise from a signal, or emphasizes certain frequency ranges and de-emphasizes others. Electronic filters include lowpass, band-pass, and highpass types. Digital filters can operate on numeric data to perform equivalent operations on digitized analog data or to enhance video images.
filtering	A type of signal conditioning that allows you to filter unwanted frequency components from the signal you are trying to measure.
floating	The condition where a common mode voltage exists, or may exist, between earth ground and the instrument or circuit of interest. Neither the high, nor the low side of a circuit is at earth potential.
floating signal sources	Signal sources with voltage signals that are not connected to an absolute reference of system ground. Also called non-referenced signal sources. Some common examples of floating signal sources are batteries, transformers, and thermocouples.
FREQ OUT	Frequency Output signal.
frequency	The number of alternating signals that occur per unit time.
ft	Feet.

function
1. A built-in execution element, comparable to an operator, function, or statement in a conventional language.
2. A set of software instructions executed by a single line of code that may have input and/or output parameters and returns a value when executed.

G

glitch — An unwanted signal excursion of short duration that is usually unavoidable.

GND — *See* ground.

ground
1. A pin.
2. An electrically neutral wire that has the same potential as the surrounding earth. Normally, a noncurrent-carrying circuit intended for safety.
3. A common reference point for an electrical system.

H

hardware — The physical components of a computer system, such as the circuit boards, plug-in devices, chassis, enclosures, peripherals, and cables.

hardware triggering — A form of triggering where you set the start time of an acquisition and gather data at a known position in time relative to a trigger signal.

Hz
1. Hertz—The SI unit for measurement of frequency. One hertz (Hz) equals one cycle per second.
2. The number of scans read or updates written per second.

I

I/O — Input/Output—The transfer of data to/from a computer system involving communications channels, operator interface devices, and/or data acquisition and control interfaces.

impedance
1. The electrical characteristic of a circuit expressed in ohms and/or capacitance/inductance.
2. Resistance.

in.	Inch or inches.
instrument driver	A set of high-level software functions that controls a specific GPIB, VXI, or RS232 programmable instrument or a specific plug-in DAQ device. Instrument drivers are available in several forms, ranging from a function callable language to a virtual instrument (VI) in LabVIEW.
instrumentation amplifier	A circuit whose output voltage with respect to ground is proportional to the difference between the voltages at its two inputs. An instrumentation amplifier normally has high-impedance differential inputs and high common-mode rejection.
interchannel delay	Amount of time that passes between sampling consecutive channels in an AI scan list. The interchannel delay must be short enough to allow sampling of all the channels in the channel list, within the sample interval. The greater the interchannel delay, the more time the PGIA is allowed to settle before the next channel is sampled. The interchannel delay is regulated by ai/ConvertClock.
interface	Connection between one or more of the following: hardware, software, and the user. For example, hardware interfaces connect two other pieces of hardware.
interrupt, interrupt request line	1. A means for a device to notify another device that an event occurred. 2. A computer signal indicating that the CPU should suspend its current task to service a designated activity.
I_{OH}	Current, output high.
I_{OL}	Current, output low.
IRQ	*See* interrupt, interrupt request line.
isolation	An electrical break between any two signals or planes up to a given voltage.
isolation barrier	An electrical break between two electrical planes providing a given or set amount of electrical isolation. Current does not flow or transfer between the two sides of the isolation barrier.

K

kHz	Kilohertz—A unit of frequency; 1 kHz = 10^3 = 1,000 Hz.
kS	1,000 samples.

L

LabVIEW	A graphical programming language.
LED	Light-Emitting Diode—A semiconductor light source.
lowpass filter	A filter that passes signals below a cutoff frequency while blocking signals above that frequency.
LSB	Least Significant Bit.

M

m	Meter.
M Series	An architecture for instrumentation-class, multichannel data acquisition devices based on the earlier E Series architecture with added new features.
measurement	The quantitative determination of a physical characteristic. In practice, measurement is the conversion of a physical quantity or observation to a domain where a human being or computer can determine the value.
measurement device	DAQ devices, such as the M Series multifunction I/O (MIO) devices, SCXI signal conditioning modules, and switch modules.
MHz	Megahertz—A unit of frequency; 1 MHz = 10^6 Hz = 1,000,000 Hz.
micro (μ)	The numerical prefix designating 10^{-6}.
MIO	Multifunction I/O—DAQ module. Designates a family of data acquisition products that have multiple analog input channels, digital I/O channels, timing, and optionally, analog output channels. An MIO product can be considered a miniature mixed signal tester, due to its broad range of signal types and flexibility. Also known as multifunction DAQ.

MITE	MXI Interface To Everything—A custom ASIC designed by National Instruments that implements the PCI bus interface. The MITE supports bus mastering for high-speed data transfers over the PCI bus.
module	A board assembly and its associated mechanical parts, front panel, optional shields, and so on. A module contains everything required to occupy one or more slots in a mainframe. SCXI and PXI devices are modules.
monotonicity	A characteristic of a DAC in which the analog output always increases as the values of the digital code input to it increase.
multichannel	Pertaining to a radio-communication system that operates on more than one channel at the same time. The individual channels might contain identical information, or they might contain different signals.
multifunction DAQ	*See* MIO.
multiplex	To assign more than one signal to a channel. *See also* mux.
mux	Multiplexer—A set of semiconductor or electromechanical switches arranged to select one of many inputs to a single output. The majority of DAQ cards have a multiplexer on the input, which permits the selection of one of many channels at a time. A switching device with multiple inputs that sequentially connects each of its inputs to its output, typically at high speeds, in order to measure several signals with a single analog input channel.

N

NI	National Instruments.
NI-DAQ	The driver software needed to use National Instruments DAQ devices and SCXI components. Some devices use Traditional NI-DAQ (Legacy); others use NI-DAQmx.
NI-DAQmx	The latest NI-DAQ driver with new VIs, functions, and development tools for controlling measurement devices. The advantages of NI-DAQmx over earlier versions of NI-DAQ include the DAQ Assistant for configuring channels and measurement tasks for your device for use in LabVIEW, LabWindows/CVI, and Measurement Studio; increased performance such as faster single-point analog I/O; and a simpler API for creating DAQ applications using fewer functions and VIs than earlier versions of NI-DAQ.

NI-PGIA	*See* instrumentation amplifier.
non-referenced signal sources	Signal sources with voltage signals that are not connected to an absolute reference or system ground. Also called floating signal sources. Some common example of non-referenced signal sources are batteries, transformers, or thermocouples.
NRSE	Non-Referenced Single-Ended mode—All measurements are made with respect to a common (NRSE) measurement system reference, but the voltage at this reference can vary with respect to the measurement system ground.

O

offset	The unwanted DC voltage due to amplifier offset voltages added to a signal.

P

PCI	Peripheral Component Interconnect—A high-performance expansion bus architecture originally developed by Intel to replace ISA and EISA. It offers a theoretical maximum transfer rate of 132 MB/s.
period	The period of a signal, most often measured from one zero crossing to the next zero crossing of the same slope. The period of a signal is the reciprocal of its frequency (in Hz). Period is designated by the symbol T.
periods	The number of periods of a signal.
PFI	Programmable Function Interface.
PGIA	Programmable Gain Instrumentation Amplifier.
physical channel	*See* channel.
Plug and Play devices	Devices that do not require DIP switches or jumpers to configure resources on the devices. Also called switchless devices port. 1. A communications connection on a computer or a remote controller; 2. A digital port, consisting of four or eight lines of digital input and/or output.
posttriggering	The technique used on a DAQ device to acquire a programmed number of samples after trigger conditions are met.

power source	An instrument that provides one or more sources of AC or DC power. Also known as power supply.
ppm	Parts per million.
pretriggering	The technique used on a DAQ device to keep a continuous buffer filled with data, so that when the trigger conditions are met, the sample includes the data leading up to the trigger condition.
pulse	A signal whose amplitude deviates from zero for a short period of time.
pulse width	The time from the rising to the falling slope of a pulse (at 50% amplitude).
PXI	A rugged, open system for modular instrumentation based on CompactPCI, with special mechanical, electrical, and software features. The PXIbus standard was originally developed by National Instruments in 1997, and is now managed by the PXIbus Systems Alliance.
PXI Express	PCI Express eXtensions for Instrumentation—The PXI implementation of PCI Express, a scalable full-simplex serial bus standard that operates at 2.5 Gbps and offers both asynchronous and isochronous data transfers.
PXI_STAR	A special set of trigger lines in the PXI backplane for high-accuracy device synchronization with minimal latencies on each PXI slot. Only devices in the PXI Star controller Slot 2 can set signal on this line. For additional information concerning PXI star signal specifications and capabilities, refer to the *PXI Specification* located at `www.pxisa.org/specs`.

Q

quadrature encoder	An encoding technique for a rotating device where two tracks of information are placed on the device, with the signals on the tracks offset by 90° from each other. This makes it possible to detect the direction of the motion.

R

range	The maximum and minimum parameters between which a sensor, instrument, or device operates with a specified set of characteristics. This may be a voltage range or a frequency range.

real time	1. Displays as it comes in; no delays. 2. A property of an event or system in which data is processed and acted upon as it is acquired instead of being accumulated and processed at a later time. 3. Pertaining to the performance of a computation during the actual time that the related physical process transpires so results of the computation can be used in guiding the physical process.
RSE	Referenced Single-Ended configuration—All measurements are made with respect to a common reference measurement system or a ground. Also called a grounded measurement system.
RTSI	Real-Time System Integration.
RTSI bus	Real-Time System Integration bus—The National Instruments timing bus that connects DAQ devices directly, by means of connectors on top of the devices, for precise synchronization of functions.

S

s	Seconds.
S	Samples.
sample counter	The clock that counts the output of the channel clock, in other words, the number of samples taken. On devices with simultaneous sampling, this counter counts the output of the scan clock and hence the number of scans.
scan	One or more analog or digital input samples. Typically, the number of input samples in a scan is equal to the number of channels in the input group. For example, one pulse from the scan clock produces one scan which acquires one new sample from every analog input channel in the group.
scan interval	Controls how often a scan is initialized; is regulated by the AI Sample Clock signal.
scan rate	Reciprocal of the scan interval.
SCC	Signal Conditioning Carriers—A compact, modular form factor for signal conditioning modules.

SCXI	Signal Conditioning eXtensions for Instrumentation—The National Instruments product line for conditioning low-level signals within an external chassis near sensors so that only high-level signals are sent to DAQ devices in the noisy PC environment.
sensor	A device that responds to a physical stimulus (heat, light, sound, pressure, motion, flow, and so on), and produces a corresponding electrical signal. Primary characteristics of sensors are sensitivity, frequency range, and linearity.
signal conditioning	1. Electronic equipment that makes transducer or other signals suitable in level and range to be transmitted over a distance, or to interface with voltage input instruments. 2. The manipulation of signals to prepare them for digitizing.
signal source	A generic term for any instrument in the family of signal generators.
signals	Signals are waveforms containing information. Although physical signals can be in the form of mechanical, electromagnetic, or other forms, they are most often converted to electronic form for measurement.
single trigger mode	When the arbitrary waveform generator goes through the staging list only once.
single-buffered	Describes a device that acquires a specified number of samples from one or more channels and returns the data when the acquisition is complete.
single-ended input	A circuit that responds to the voltage on one input terminal and ground. *See also* differential input.
single-ended output	A circuit whose output signal is present between one output terminal and ground.
software applications	The programs that run on your computer and perform a specific user-oriented function, such as accounting, program development, measurement, or data acquisition. In contrast, operating system functions basically perform the generic ‘housekeeping’ of the machine, which is independent of any specific application. Operating system functions include the saving of data (file system), handling of multiple programs at the same time (multi-tasking), network interconnection, printing, and keyboard/user interface interaction.

software triggering	A method of triggering in which you simulate an analog trigger using software. Also called *conditional retrieval*.
source impedance	A parameter of signal sources that reflects current-driving ability of voltage sources (lower is better) and the voltage-driving ability of current sources (higher is better).
synchronous	1. Hardware—A property of an event that is synchronized to a reference clock. 2. Software—A property of a function that begins an operation and returns only when the operation is complete. A synchronous process is, therefore, *locked* and no other processes can run during this time.

T

task	In NI-DAQmx, a collection of one or more channels, timing, and triggering and other properties that apply to the task itself. Conceptually, a task represents a measurement or generation you want to perform.
TC	*See* terminal count.
terminal	An object or region on a node through which data passes.
terminal count	The highest value of a counter.
t_{gh}	Gate hold time.
t_{gsu}	Gate setup time.
t_{gw}	Gate pulse width.
Timebase	The reference signals for controlling the basic accuracy of time or frequency-based measurements. For instruments, timebase refers to the accuracy of the internal clock.
t_{out}	Output delay time.
transducer	A device that responds to a physical stimulus (heat, light, sound, pressure, motion, flow, and so on), and produces a corresponding electrical signal. *See also* sensor.

trigger	1. Any event that causes or starts some form of data capture. 2. An external stimulus that initiates one or more instrument functions. Trigger stimuli include a front panel button, an external input voltage pulse, or a bus trigger command. The trigger may also be derived from attributes of the actual signal to be acquired, such as the level and slope of the signal.
t_{sc}	Source clock period.
t_{sp}	Source pulse width.
TTL	Transistor-Transistor Logic—A digital circuit composed of bipolar transistors wired in a certain manner. A typical medium-speed digital technology. Nominal TTL logic levels are 0 and 5 V.

U

USB	Universal Serial Bus—A 480 Mbit/s serial bus with up to 12-Mbps bandwidth for connecting computers to keyboards, printers, and other peripheral devices. USB 2.0 retains compatibility with the original USB specification.

V

V	Volts.
V_{cm}	Common-mode voltage.
V_g	Ground loop voltage.
V_{IH}	Volts, input high.
V_{IL}	Volts, input low.
V_{in}	Volts in.
V_m	Measured voltage.
V_{OH}	Volts, output high.
V_{OL}	Volts, output low.

V_{out}	Volts out.
V_s	Signal source voltage.
virtual channel	*See* channel.

W

waveform	1. The plot of the instantaneous amplitude of a signal as a function of time. 2. Multiple voltage readings taken at a specific sampling rate.

Index

Symbols

.NET languages documentation, *xvi*

Numerics

10 MHz reference clock, 10-3
100 kHz Timebase, 10-2
20 MHz Timebase, 10-2
80 MHz source mode, 7-38
80 MHz Timebase, 10-2

A

A/D converter, 4-2
AC return path, creating, 9-3
accessories, 2-4, A-3
    choosing for your device, 1-2
acquisitions
    circular-buffered, 4-11
    double-buffered, 4-11
    hardware-timed, 4-10
    software-timed, 4-10
AI Convert Clock signal, 4-17
AI Convert Clock Timebase signal, 4-20
AI data acquisition methods, 4-10
AI FIFO, 4-2
AI Hold Complete Event signal, 4-20
AI Pause Trigger signal, 4-23
AI Reference Trigger signal, 4-22
AI Sample Clock signal, 4-15
AI Sample Clock Timebase signal, 4-17
AI Start Trigger signal, 4-21
AI timing signals, 4-12
ai/ConvertClock, 4-17
ai/ConvertClockTimebase, 4-20
ai/HoldCompleteEvent, 4-20
ai/PauseTrigger, 4-23
ai/ReferenceTrigger, 4-22
ai/SampleClock, 4-15
ai/SampleClockTimebase, 4-17
ai/StartTrigger, 4-21
analog input
    analog-to-digital converter, 4-2
    channels, sampling with AI Sample Clock and AI Convert Clock, B-2
    charge injection, B-1
    circuitry, 4-1
    crosstalk when sampling multiple channels, B-1
    data acquisition methods, 4-10
    FIFO, 4-2
    getting started with applications in software, 4-24
    ghost current when sampling multiple channels, B-1
    ground-reference settings, 4-5
    I/O connector, 4-1
    instrumentation amplifier, 4-1
    MUX, 4-1
    NI-PGIA, 4-1
    range, 4-2
    sampling channels with AI Sample Clock and AI Convert Clock, B-2
    signals, 4-12
    timing signals, 4-12
    triggering, 4-11
    troubleshooting, B-1
analog output
    circuitry, 5-1
    connecting voltage signals, 5-4
    data generation methods, 5-2
    getting started with applications in software, 5-11
    glitches on the output signal, 5-2
    signals, 5-5

timing signals, 5-5
trigger signals, 5-4
triggering, 5-4
troubleshooting, B-2
ANSI C documentation, *xvi*
AO FIFO, 5-2
AO Pause Trigger signal, 5-7
AO Sample Clock signal, 5-8
AO Sample Clock Timebase signal, 5-10
AO Start Trigger signal, 5-6
ao/PauseTrigger, 5-7
ao/SampleClock, 5-8
ao/SampleClockTimebase, 5-10
ao/StartTrigger, 5-6
applications
counter input, 7-2
counter output, 7-20
edge counting, 7-2
arm start trigger, 7-31
avoiding scanning faster than necessary, 4-9

B

block diagram
PFI input circuitry, 8-1
PFI output circuitry, 8-2
buffered
edge counting, 7-3
non-cumulative, 7-4
hardware-timed acquisitions, 4-10
hardware-timed generations, 5-3
period measurement, 7-7
pulse-width measurement, 7-5
semi-period measurement, 7-9
two-signal edge-separation measurement, 7-19
bus
interface, 11-1
RTSI, 10-3

C

cables, 2-4, A-3
choosing for your device, 1-2
custom, 2-4
calibration certificate (NI resources), C-2
calibration circuitry, 2-3
cascading counters, 7-32
changing data transfer methods between DMA and IRQ, 11-4
channel Z behavior, 7-16
channels, sampling with AI Sample Clock and AI Convert Clock, B-2
charge injection, B-1
circular-buffered acquisition, 4-11
clock
10 MHz reference, 10-3
external reference, 10-2
generation, 10-1
PXI, and trigger signals, 10-8
routing, 10-1
common-mode noise, 9-2
CompactPCI, using with PXI, 11-3
configuring, ground-reference settings in software, 4-5
connecting
analog current input signals, 4-3
analog voltage output signals, 5-4
counter signals, B-3
digital I/O signals, 6-2
PFI input signals, 8-3
connector, A-1
information, 3-1
RTSI, 3-2, 10-4
considerations
for field wiring, 4-11
for multichannel scanning, 4-8
for PXI, 11-2
continuous pulse train generation, 7-22
controllers, DMA, 11-1

controlling counting direction, 7-2
conventions used in the manual, *xiii*
Counter *n* A signal, 7-28
Counter *n* Aux signal, 7-27
Counter *n* B signal, 7-28
Counter *n* Gate signal, 7-27
Counter *n* HW Arm signal, 7-28
Counter *n* Internal Output signal, 7-29
Counter *n* Source signal, 7-25
Counter *n* TC signal, 7-29
Counter *n* Up_Down signal, 7-28
Counter *n* Z signal, 7-28
counter output applications, 7-20
counter signals
    Counter *n* A, 7-28
    Counter *n* Aux, 7-27
    Counter *n* B, 7-28
    Counter *n* Gate, 7-27
    Counter *n* HW Arm, 7-28
    Counter *n* Internal Output, 7-29
    Counter *n* Source, 7-25
    Counter *n* TC, 7-29
    Counter *n* Up_Down, 7-28
    FREQ OUT, 7-29
    Frequency Output, 7-29
counter terminals, default, 7-30
counters
    cascading, 7-32
    duplicate count prevention, 7-34
    edge counting, 7-2
    filters, 7-32
    generation, 7-20
    input applications, 7-2
    other features, 7-32
    output applications, 7-20
    prescaling, 7-33
    pulse train generation, 7-22
    retriggerable single pulse generation, 7-21
    simple pulse generation, 7-20
    single pulse generation, 7-20
    single pulse generation with start trigger, 7-20
    synchronization modes, 7-37
    timing signals, 7-25
    triggering, 7-31
    troubleshooting, B-3
counting edges, 7-2
creating an AC return path, 9-3
crosstalk when sampling multiple channels, B-1
current, connecting analog input signals, 4-3
custom cabling, 2-4

D

DACs, 5-1
DAQ
    hardware, 2-1
    system, 2-1
DAQ-STC2, 2-2
data
    acquisition methods, 4-10
    generation methods, 5-2
data transfer methods, 11-3
    changing, 11-4
    DMA, 11-3
    IRQ, 11-4
    programmed I/O, 11-4
Declaration of Conformity (NI resources), C-1
default counter terminals, 7-30
device
    cabling, 2-4
    information, A-1
    multiple synchronization, 10-3
    specifications, 1-2
diagnostic tools (NI resources), C-1
digital
    input and output, 6-1
    isolation, 9-2

isolators, 9-1
routing, 10-1
signals, connecting, 6-2
source, triggering, 12-1
digital I/O
circuitry, 6-1
connecting signals, 6-2
getting started with applications in software, 6-3
I/O protection, 6-1
programmable power-up states, 6-2
triggering, 12-1
digital isolators, 4-2, 5-2
DMA
as a transfer method, 11-3
changing data transfer methods, 11-4
controllers, 11-1
documentation
conventions used in manual, *xiii*
NI resources, C-1
related documentation, *xiv*
double-buffered acquisition, 4-11
drivers (NI resources), C-1
duplicate count prevention, 7-34
enabling in NI-DAQmx, 7-37
example, 7-35
troubleshooting, B-3

E

edge counting, 7-2
buffered, 7-3
non-cumulative buffered, 7-4
sample clock, 7-3
edge-separation measurement
buffered two-signal, 7-19
single two-signal, 7-18
enabling duplicate count prevention in NI-DAQmx, 7-37
encoders, quadrature, 7-15
encoding
X1, 7-15
X2, 7-16
X4, 7-16
equivalent time sampling, 7-24
examples (NI resources), C-1
exporting timing output signals using PFI terminals, 8-3
external
reference clock, 10-2
source mode, 7-38

F

features, counter, 7-32
field wiring considerations, 4-11
filters
counter, 7-32
PFI, 8-4
PXI_STAR, 10-9
RTSI, 10-6
FREQ OUT signal, 7-29
frequency
division, 7-24
generation, 7-23
generator, 7-23
measurement, 7-9
Frequency Output signal, 7-29

G

generations
analog output data, 5-2
buffered hardware-timed, 5-3
clock, 10-1
continuous pulse train, 7-22
frequency, 7-23
hardware-timed, 5-3
non-buffered hardware-timed, 5-3
pulse for ETS, 7-24

pulse train, 7-22
retriggerable single pulse, 7-21
simple pulse, 7-20
single pulse, 7-20
single pulse with start trigger, 7-20
software-timed, 5-2
getting started, 1-1
AI applications in software, 4-24
AO applications in software, 5-11
DIO applications in software, 6-3
ghost voltages when sampling multiple channels, B-1
ground-reference
connections, checking, B-1
settings, analog input, 4-5

H

hardware
DAQ, 2-1
installing, 1-1
hardware-timed
acquisitions, 4-10
generations, 5-3
help, technical support, C-1

I

I/O connector, 3-1
NI 6236 pinout, A-1
I/O protection, 6-1, 8-6
input signals
using PFI terminals as, 8-2
using RTSI terminals as, 10-6
installation
hardware, 1-1
NI-DAQ, 1-1
other software, 1-1
instrument drivers (NI resources), C-1
interface, bus, 11-1
interrupt request, as a transfer method, 11-4
IRQ
as a transfer method, 11-4
changing data transfer methods, 11-4
isolated DAQ devices, 9-1
benefits, 9-2
common-mode noise, 9-2
isolation barrier, 4-2, 5-2
isolators, 9-1

K

KnowledgeBase, C-1

L

LabVIEW documentation, *xv*
LabWindows/CVI documentation, *xvi*

M

M Series
information, A-1
specifications, *xvii*
Measurement Studio documentation, *xvi*
measurements
buffered period, 7-7
buffered pulse-width, 7-5
buffered semi-period, 7-9
buffered two-signal edge-separation, 7-19
frequency, 7-9
period, 7-6
position, 7-15
pulse-width, 7-5
semi-period, 7-8
single period, 7-7
single pulse-width, 7-5
single semi-period, 7-8
single two-signal edge-separation, 7-18
two-signal edge-separation, 7-18
using quadrature encoders, 7-15
using two pulse encoders, 7-17

measuring
high frequency with two counters, 7-11
large range of frequencies using two counters, 7-12
low frequency with one counter, 7-9
averaged, 7-10
methods, data transfer, 11-3
minimizing
glitches on the output signal, 5-2
output signal glitches (troubleshooting), B-2
multichannel scanning considerations, 4-8
multiple device synchronization, 10-3

N

National Instruments support and services, C-1
NI 6236
accessory options, A-3
cabling options, A-3
information, A-1
pinout, A-1
specifications, A-3
NI support and services, C-1
NI-DAQ documentation, *xiv*
device documentation browser, *xvii*
NI-DAQmx Base documentation, *xv*
NI-DAQmx for Linux documentation, *xiv*
NI-DAQmx, enabling duplicate count prevention, 7-37
non-buffered hardware-timed
acquisitions, 4-11
generations, 5-3
non-cumulative buffered edge counting, 7-4

O

other internal source mode, 7-38
other software, 1-1
output signals
glitches (troubleshooting), B-2
minimizing glitches, 5-2
outputs, using RTSI as, 10-5
overview, 2-1

P

pause trigger, 7-31
period measurement, 7-6
buffered, 7-7
single, 7-7
PFI, 8-1
connecting input signals, 8-3
exporting timing output signals using PFI terminals, 8-3
filters, 8-4
I/O protection, 8-6
programmable power-up states, 8-6
using terminals as static digital I/Os, 8-3
using terminals as timing input signals, 8-2
pin assignments. *See* pinouts
pinouts, 1-1
device, 1-1
NI 6236, A-1
RTSI connector, 3-2, 10-4
position measurement, 7-15
power-up states, 6-2, 8-6
prescaling, 7-33
programmable
function interface, 8-1
power-up states, 6-2, 8-6
programmed I/O, 11-4
programming devices in software, 2-5
programming examples (NI resources), C-1
pulse
encoders
measurements using two, 7-17
generation for ETS, 7-24

train generation, 7-22
continuous, 7-22
pulse-width measurement, 7-5
buffered, 7-5
single, 7-5
PXI
and PXI Express, 11-2
clock, 11-2
clock and trigger signals, 10-8
considerations, 11-2
trigger signals, 11-2
triggers, 10-8
using with CompactPCI, 11-3
PXI Express chassis compatibility, 11-2
PXI_CLK10, 10-8
PXI_STAR
filters, 10-9
trigger, 10-8

Q

quadrature encoders, 7-15

R

range, analog input, 4-2
real-time system integration bus, 10-3
reciprocal frequency measurement, 7-12
reference clock
10 MHz, 10-3
external, 10-2
related documentation, *xiv*
retriggerable single pulse generation, 7-21
routing
clock, 10-1
digital, 10-1
RTSI, 10-3
connector pinout, 3-2, 10-4
filters, 10-6
using as outputs, 10-5
using terminals as timing input signals, 10-6

S

sample clock edge counting, 7-3
scanning speed, 4-9
semi-period measurement, 7-8
buffered, 7-9
single, 7-8
sensors, 2-3
settings, analog input ground-reference, 4-5
short high-quality cabling, 4-8
signal
descriptions, 3-1
routing, RTSI bus, 10-3
signals
AI Convert Clock, 4-17
AI Convert Clock Timebase, 4-20
AI Hold Complete Event, 4-20
AI Pause Trigger, 4-23
AI Reference Trigger, 4-22
AI Sample Clock, 4-15
AI Sample Clock Timebase, 4-17
AI Start Trigger, 4-21
analog input, 4-12
analog output, 5-5
AO Pause Trigger, 5-7
AO Sample Clock, 5-8
AO Sample Clock Timebase, 5-10
AO Start Trigger, 5-6
connecting analog current input, 4-3
connecting analog voltage output, 5-4
connecting counter, B-3
connecting digital I/O, 6-2
connecting PFI input, 8-3
Counter *n* A, 7-28
Counter *n* Aux, 7-27
Counter *n* B, 7-28
Counter *n* Gate, 7-27
Counter *n* HW Arm, 7-28

Counter *n* Internal Output, 7-29
Counter *n* Source, 7-25
Counter *n* TC, 7-29
Counter *n* Up_Down, 7-28
Counter *n* Z, 7-28
counter timing, 7-25
counters, 7-25
exporting timing output using PFI terminals, 8-3
FREQ OUT, 7-29
Frequency Output, 7-29
minimizing output glitches, B-2
output, minimizing glitches on, 5-2
simple pulse generation, 7-20
single
period measurement, 7-7
pulse generation, 7-20
retriggerable, 7-21
with start trigger, 7-20
pulse-width measurement, 7-5
semi-period measurement, 7-8
two-signal edge-separation measurement, 7-18
software
configuring AI ground-reference settings, 4-7
NI resources, C-1
programming devices, 2-5
timed acquisitions, 4-10
software-timed
acquisitions, 4-10
generations, 5-2
specifications, A-3
device, 1-2
start trigger, 7-31
static
digital input, 6-1
digital output, 6-1
DIO, using PFI terminals as, 8-3
support, technical, C-1
synchronization modes, 7-37
80 MHz source, 7-38
external source, 7-38
other internal source, 7-38
synchronizing multiple devices, 10-3
synchronous counting mode, 7-34

T

technical support, *xvii*, C-1
terminals, default counter, 7-30
Timebase
100 kHz, 10-2
20 MHz, 10-2
80 MHz, 10-2
timing output signals, exporting using PFI terminals, 8-3
training, *xvii*
training and certification (NI resources), C-1
transducers, 2-3
triggering, 12-1
analog input, 4-11
counter, 7-31
with a digital source, 12-1
triggers, 12-1
AI Pause Trigger signal, 4-23
AI Reference Trigger signal, 4-22
AI Start Trigger signal, 4-21
AO Pause Trigger signal, 5-7
AO Start Trigger signal, 5-6
arm start, 7-31
pause, 7-31
PXI, 10-8
PXI_STAR, 10-8
Star Trigger, 10-8
start, 7-31
troubleshooting
analog input, B-1
analog output, B-2
counters, B-3
NI resources, C-1

two-signal edge-separation
measurement, 7-18
buffered, 7-19
single, 7-18

U

using PFI terminals
as static digital I/Os, 8-3
as timing input, 8-2
to export timing output signals, 8-3
using RTSI
as outputs, 10-5
terminals as timing input signals, 10-6
using short high-quality cabling, 4-8

V

voltage, connecting analog voltage, 5-4

W

waveform generation signals, 5-5
Web resources, C-1
wiring, field, 4-11

X

X1 encoding, 7-15
X2 encoding, 7-16
X4 encoding, 7-16

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP