

# Data Acquisition and Signal Conditioning Course Manual

**Course Software Version 2011  
February 2012 Edition  
Part Number 320733P-01**

## **Copyright**

© 1995–2012 National Instruments Corporation. All rights reserved.

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**

CVI, LabVIEW, National Instruments, NI, ni.com, the National Instruments corporate logo, and the Eagle logo are trademarks of National Instruments Corporation. Refer to the *Trademark Information* at [ni.com/trademarks](http://ni.com/trademarks) for other National Instruments trademarks.

Tektronix® and Tek are registered trademarks of Tektronix, Inc. The mark LabWindows is used under a license from Microsoft Corporation. Windows is a registered trademark of Microsoft Corporation in the United States. 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/technology, refer to the appropriate location: **Help»Patents** in your software, the `patents.txt` file on your media, or the *National Instruments Patent Notice* at [ni.com/patents](http://ni.com/patents).

## **Worldwide Technical Support and Product Information**

[ni.com](http://ni.com)

## **Worldwide Offices**

Visit [ni.com/niglobal](http://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.

## **National Instruments Corporate Headquarters**

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

For further support information, refer to the *Additional Information and Resources* appendix. To comment on National Instruments documentation, refer to the National Instruments Web site at [ni.com/info](http://ni.com/info) and enter the Info Code `feedback`.

# Contents

---

## Student Guide

A. NI Certification .....	vii
B. Course Description .....	viii
C. What You Need to Get Started .....	viii
D. Course Goals .....	x
E. Course Conventions .....	xi

## Lesson 1

### Overview of a DAQ System

A. DAQ System Overview .....	1-2
B. Sensors .....	1-2
C. Signals .....	1-4
D. DAQ Hardware .....	1-10
E. Signal Conditioning .....	1-10
F. DAQ Software .....	1-11
Summary .....	1-12

## Lesson 2

### Data Acquisition Hardware and Software

A. DAQ Hardware Overview .....	2-2
B. Components of a DAQ Device .....	2-6
C. Choosing Appropriate DAQ Hardware .....	2-12
D. DAQ Software Overview .....	2-19
E. Overview of NI-DAQmx VIs .....	2-26
Summary .....	2-30

## Lesson 3

### Analog Input

A. Grounding Issues .....	3-2
B. Sampling Considerations .....	3-13
C. Single Sample Software-Timed Acquisition .....	3-20
D. DAQ Device Architectures .....	3-23
E. Finite Buffered Acquisition .....	3-26
F. Continuous Buffered Acquisition .....	3-30
G. Triggering .....	3-35
Summary .....	3-40

## Lesson 4

### Analog Output

A. Analog Output Architecture.....	4-2
B. Single Sample Generation .....	4-3
C. Finite Buffered Generation .....	4-5
D. Continuous Buffered Generation .....	4-10
E. Triggered Generation .....	4-14
Summary .....	4-15

## Lesson 5

### Digital I/O

A. Digital Overview.....	5-2
B. Digital I/O .....	5-5
C. Hardware-Timed Digital I/O .....	5-7
Summary .....	5-9

## Lesson 6

### Counters

A. Counter Signals.....	6-2
B. Edge Counting .....	6-7
C. Pulse Generation .....	6-10
D. Pulse Measurement .....	6-15
E. Frequency Measurement.....	6-18
F. Position Measurement.....	6-23
Summary .....	6-27

## Lesson 7

### Signal Conditioning

A. Overview of Signal Conditioning.....	7-2
B. Signal Conditioning Systems.....	7-2
C. Signal Conditioning for Voltage Measurements .....	7-3
D. Temperature Measurements.....	7-13
E. Strain, Pressure, Load, and Torque Measurements .....	7-20
F. Sound and Vibration Measurements.....	7-33
Summary .....	7-39

## Lesson 8

### Synchronization

A. Synchronizing Measurements.....	8-2
B. Single Device Synchronization.....	8-6
C. Multiple Device Synchronization .....	8-8
D. Counters and Synchronization .....	8-12
Summary .....	8-17

## Appendix A DAQ Theory

A. Theory of Common Sensors .....	A-2
B. Analog I/O Circuitry .....	A-7

## Appendix B Signal Processing

A. Discrete Fourier Transform (DFT) and Fast Fourier Transform (FFT).....	B-2
B. Magnitude and Phase Information .....	B-4
C. Frequency Spacing and Symmetry of the DFT/FFT .....	B-5
D. Power Spectrum .....	B-10
E. About Spectral Leakage and Smoothing Windows .....	B-14
F. Characteristics of Different Types of Window Functions .....	B-18
G. Determining Which Type of Window to Use .....	B-24
H. Filtering .....	B-27
I. Ideal Filters .....	B-27
J. Practical (Nonideal) Filters .....	B-30
K. Advantages of Digital Filters over Analog Filters .....	B-31
L. IIR and FIR Filters .....	B-32
M. Infinite Impulse Response Filters .....	B-34
N. IIR Filter Comparison .....	B-39
O. Transient Response of IIR Filters .....	B-41
P. Finite Impulse Response Filters .....	B-43
Summary .....	B-46

## Appendix C Course Slides

## Appendix D Additional Information and Resources

---

## Digital I/O

This lesson describes the digital functionality of a DAQ device, which can perform digital input and digital output.

### Topics

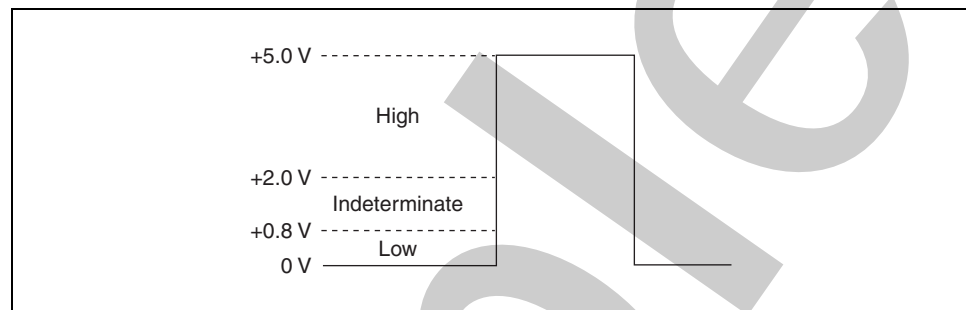
---

- A. Digital Overview
- B. Digital I/O
- C. Hardware-Timed Digital I/O

Sample

## A. Digital Overview

The digital lines on a DAQ device accept and generate TTL-compatible signals. A TTL signal has two states—logic low and logic high, as shown in Figure 5-1. Logic low signals are signals between 0 to +0.8 V. Logic high signals are signals between +2 to +5 V. Signals between +0.8 and +2.0 V are indeterminate.



**Figure 5-1.** TTL Signal States

To ensure that digital lines measure the signal correctly, make sure the voltage level of the signal is never between +0.8 and +2 V.

### Digital Terminology

Following are some of the common terms used with digital I/O operations.

- **Bit**—The smallest unit of data used in a digital operation. Bits are binary, so they can be a 1 or a 0.
- **Byte**—A binary number consisting of 8 related bits of data. Bytes are also used to denote the amount of memory required to store one byte of data.
- **Line**—An individual signal in a digital port. The difference between a bit and a line is that the bit refers to the actual data transferred, and the line refers to the hardware the bit is transferred on. However, the terms line and bit are fairly interchangeable. For example, an 8-bit port is the same as a port with eight lines.
- **Port**—A collection of digital lines. Usually the lines are grouped into an 8-bit port. Typical multifunction I/O DAQ devices have multiple 8-bit ports, each containing eight lines of bidirectional DIO signals. LabVIEW VIs often refer to a port as a digital channel.

- **Port Width**—The number of lines in a port. For example, some DAQ devices have three ports each containing 8 digital lines. Therefore, the port width is eight.
- **Mask**—Determines if a digital line is ignored. For example, if you write to a port, but you do not want to write to all the lines, you can set a mask to ignore the lines you do not want to write to.

## NI-DAQmx Digital Notation

In NI-DAQmx, use the following conventions to describe digital lines and ports. In each case, *X* corresponds to the device number of your DAQ device, *Y* corresponds to a digital port, and *A* and *B* correspond to digital lines on your DAQ device.

- **Port**—Dev*X*/Port *Y*
- **Single Line**—Dev*X*/Port *Y*/Line *A*
- **Multiple Lines**—Dev*X*/Port *Y*/Line *A*:*B*. The lines are read (or written) in ascending order, starting at Line *A* and continuing to Line *B*. To explicitly control the order in which the digital lines are read (or written), use the notation Dev*X*/Port *Y*/Line *A*, Dev*X*/Port *Y*/Line *B*. The comma separates each digital line.

## Digital Channel Data Formats

Digital channel data format deals with the type of the digital data that is read or written. You can choose which data format to use when using the DAQmx Read VI and DAQmx Write VI.

### Line Format (Boolean)

The line format represents each line within a channel as a single Boolean value (a single byte). The states of the data are limited to 1s (true) and 0s (false). Line formats are only provided for single sample reads and writes.

Use the line format when it is convenient for manipulating or displaying the digital data. A typical application is controlling or reading back relay states. For high-speed digital applications, you should generally not use the line format.

### Port Format (Integer)

The port format matches the native format of digital devices that can represent only two digital states and organize individual lines into collections known as ports. For more information, refer to the *Digital Data (Integer Format)* topic in the *DAQmx Help*.



The port format is the most efficient in terms of space, as it requires only a bit of memory per line. In addition, the port format is often the most efficient in time as it matches the native format of many devices.

The largest integer supported is 32 bits; therefore, you can read and write digital channels with no more than 32 lines when using the port format.

## Waveform

The waveform data format includes the channel name and timing information with the actual data represented in a dedicated digital format. Your ADE provides a mechanism for extracting and setting individual parts of the waveform.

The dedicated digital format represents digital data similar to logic analyzers and digital simulation tools. Each channel has no limits on the number of lines. In addition, the digital format allows for additional states beyond basic 1s and 0s. The ADE can take advantage of this format by tailoring data and graph displays for the digital data.

For input tasks, you can use the additional information for a variety of purposes. For example, you can update graphs to show the timing information and include labels with the channel names. Because there is overhead associated with including this additional information, NI-DAQmx allows you to configure the information you want to include.

For output tasks, the timing information is the primary field that is useful. A waveform generated by a library may include timing information that you can use to set up the timing for your output task.

When reading data, the waveform data includes the time when the first sample in the waveform was acquired,  $t_0$ , and the amount of time that elapsed between each sample,  $dt$ . However, there are limitations on these two values.

## B. Digital I/O

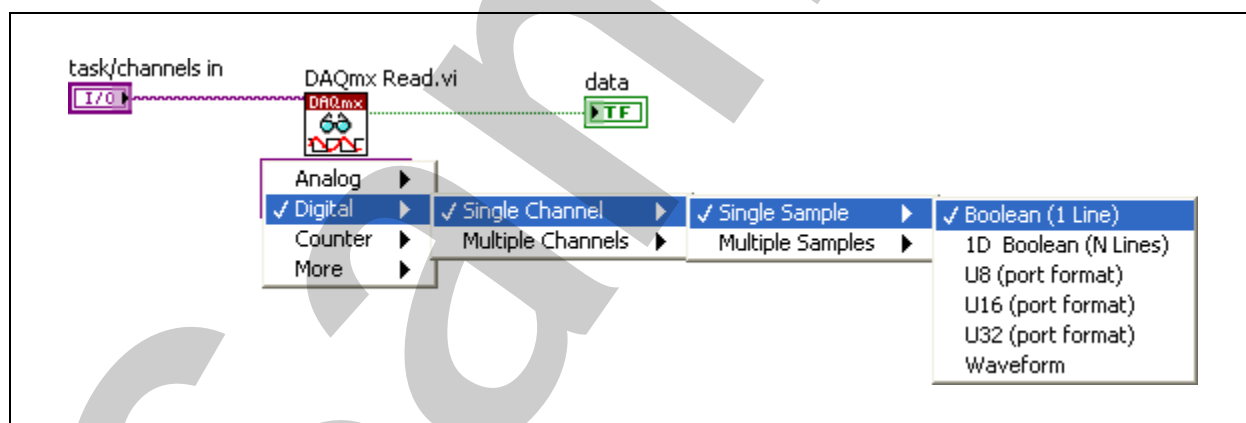
To perform digital I/O in NI-DAQmx, select a digital instance of the DAQmx Read VI or DAQmx Write VI. In addition to these VIs, use the DAQmx Timing VI and DAQmx Triggering VI to configure digital I/O tasks. To programmatically create a digital channel, use the DAQmx Create Virtual Channel VI.

### DAQmx Create Virtual Channel VI

To programmatically create a digital input or output channel, select the Digital Input or Digital Output instance of the DAQmx Create Virtual Channel VI. These instances of the VI allow you to create a channel composed of a digital port, a digital line, or a collection of digital lines. In addition, use the **line grouping** input to determine the manner in which the channel organizes the digital lines. You can select to create one channel for each line or to create one channel for all lines.

### DAQmx Read VI

The DAQmx Read VI reads samples from the task or channels you specify. The instances of this polymorphic VI specify what format of samples to return, whether to read a single sample or multiple samples at once, and whether to read from one or multiple channels. Select a digital instance from the pull-down menu to perform digital input.



**Figure 5-2.** DAQmx Read VI—Digital Instance

Select to read either a single channel or multiple channels. If the channel line grouping is set to one channel for all lines, reading a single channel returns all the values on each of the lines in the channel. If the channel line grouping is set to one channel for each line, you must read multiple channels in order to read the values on each of the digital lines specified.

Select to read either a single sample or multiple samples at once. The data type options for the return value(s) allow you to return the value in line or port format. Line format consists of a single Boolean value or an array of Boolean values (for multiple lines). An unsigned 8-bit integer (U8) or unsigned 16-bit integer (U16) or unsigned 32-bit integer (U32) are the options for returning the value in port format. When reading multiple channels, the data type options are the same, with an array dimension added to each type to account for the multiple channel reading.

If you select the U8 or U16 or U32 port format to return data, use the Number to Boolean Array function located on the **Numeric»Conversion** palette, to convert the number into an array of Boolean values.

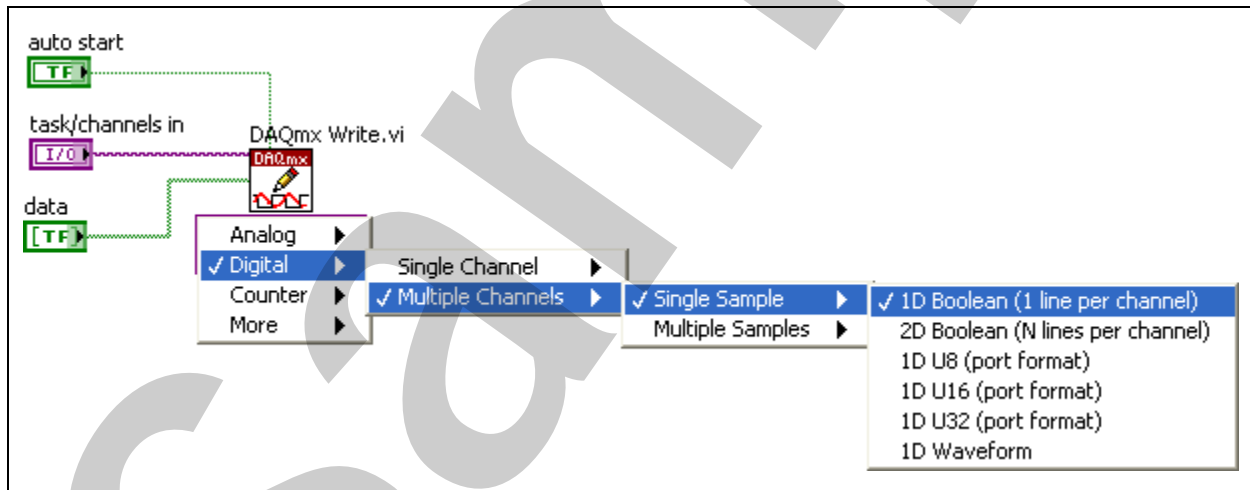
---

To practice the concepts in this section, complete Exercise 5-1.

---

## DAQmx Write VI

The DAQmx Write VI writes samples to the task or channels you specify. The instances of this polymorphic VI specify the format of the samples to write, whether to write one or multiple samples, and whether to write to one or multiple channels. Select a digital instance of the DAQmx Write VI to perform digital output.



**Figure 5-3.** DAQmx Write VI—Digital Instance

The settings for the digital instance are configured in the same manner as the digital instance of the DAQmx Read VI.

By default, the **auto start** input of the DAQmx Write VI is True when writing single samples and False when writing multiple samples. If you use the DAQmx Start Task VI and DAQmx Stop Task VI, always set the **auto**

**start** input to False. This allows you greater control of the task state model and improves the speed of your program.

## DAQmx Timing VI

The handshaking instance of the DAQmx Timing VI determines the number of digital samples to acquire or generate using digital handshaking between the device and a peripheral device. Instead of specifying a sampling rate, specify the number of digital samples to acquire or generate using digital handshaking.



**Note** Not all devices support digital handshaking. Consult your device documentation to see if handshaking is supported on your device.

## DAQmx Trigger VI

Use the DAQmx Trigger VI to configure triggering for the task. The instances of this polymorphic VI correspond to the trigger and trigger type to configure. Configure digital triggers settings the same way you configure analog input and analog output triggers. Refer to Lesson 3, *Analog Input*, of this manual for more information about configuring triggers.

---

To practice the concepts in this section, complete Exercise 5-2.

---

## C. Hardware-Timed Digital I/O

---

If your DAQ device supports hardware-timed digital I/O, you can program hardware-timed digital acquisitions and generations similar to the way you program hardware-timed analog acquisitions and generations.

### Onboard Digital I/O Sample Clock

Some DAQ devices, such as X Series devices, have a dedicated onboard sample clock for digital I/O. You can program hardware-timed acquisitions and generations the same way you program hardware-timed analog acquisitions and generations using the DAQmx Timing VI.

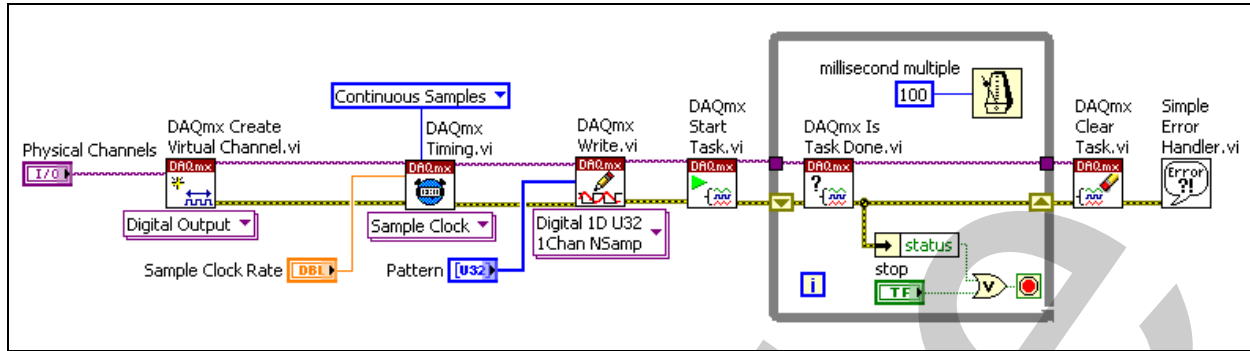


Figure 5-4. Hardware-timed Digital Generation Using DIO Sample Clock

### Correlated Digital I/O

Some DAQ devices, such as M Series devices, do not have a dedicated onboard sample clock for digital I/O, but do support correlated digital I/O. These devices can use a different clock, typically the AI or AO sample clock or an external clock, for hardware-timed acquisitions and generations. Specify which clock to use with the source input of the DAQmx Timing VI.

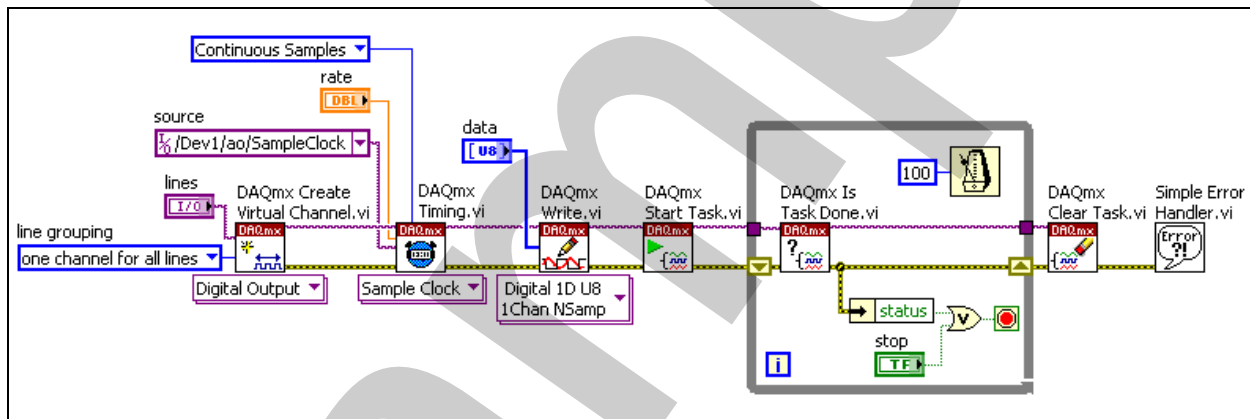


Figure 5-5. Correlated Digital Output Example VI Block Diagram

### Hardware-Timed Digital I/O Caveat

A few DAQ devices do not support hardware-timed digital I/O because they do not have a dedicated onboard digital I/O sample clock and do not support correlated digital I/O.

---

To practice the concepts in this section, complete Exercise 5-3.

---

## Summary

---

- Digital lines on a typical DAQ device can:
  - Read and write TTL-compatible signals.
  - Perform only non-timed digital I/O.
  - Be configured individually for input or output.
- Typical DAQ devices have digital lines that can read and write TTL-compatible signals.
- The specifications for a TTL signal define 0 to +0.8 V as logic low and +2 to +5 V as logic high. Between +0.8 and +2 V can be interpreted as high or low.
- Use the DAQmx Read VI and DAQmx Write VI to easily configure digital input and output operations.
- For hardware-timed digital I/O, use an onboard digital I/O sample clock or correlated digital I/O.

Sample

## Self-Review: Quiz

---

1. Digital signals are always between 0 and 5 Volts.
  - a. True
  - b. False
  
2. All DAQ devices have a dedicated onboard sample clock for digital I/O.
  - a. True
  - b. False

Sample



Sample

## Self-Review: Quiz Answers

---

1. Digital signals are always between 0 and 5 Volts.
  - a. True
  - b. False**
  
2. All DAQ devices have a dedicated onboard sample clock for digital I/O.
  - a. True
  - b. False**

Sample

## Notes

---

Sample