
Keithley KPCI-3100 Series

Using DriverLINX with Your Hardware

KEITHLEY

Information in this document is subject to change without notice. The software described in this document is furnished under a license agreement. The software may be used or copied only in accordance with the terms of the agreement.

SCIENTIFIC SOFTWARE TOOLS, INC. SHALL NOT BE LIABLE FOR ANY SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RELATED TO THE USE OF THIS PRODUCT. THIS PRODUCT IS NOT DESIGNED WITH COMPONENTS OF A LEVEL OF RELIABILITY SUITABLE FOR USE IN LIFE SUPPORT OR CRITICAL APPLICATIONS.

This document may not, in whole or in part, be copied, photocopied, reproduced, translated or reduced to any electronic medium or machine readable form without prior written consent from Scientific Software Tools, Inc.

Keithley KPCI-3100 Series: Using DriverLINX with Your Hardware
Copyright © 1999-2001 by Scientific Software Tools, Inc.
All rights reserved.

SST 28-0110-1

DriverLINX, SSTNET, and LabOBJX are registered trademarks and DriverLINX/VB is a trademark of Scientific Software Tools, Inc.
MetraByte is a trademark of Keithley Instruments, Inc.
Microsoft and Windows are registered trademarks and Visual C++ and Visual Basic are trademarks of Microsoft Corporation.
Borland is a registered trademark and Borland C++ and Delphi are trademarks of Borland International, Inc.
All other brand and product names are trademarks or registered trademarks of their respective companies.

Contents

Preface	5
Software License and Software Disclaimer of Warranty	5
About DriverLINX.....	7
About This User's Guide	7
Conventions Used in This Manual.....	9
Configuring the KPCI-3100 Series	11
Introduction.....	11
Configure DriverLINX Device Dialog	11
Device Subsystem Page.....	13
Analog Input Subsystem Page.....	15
Analog Output Subsystem Page	16
Digital Input Subsystem Page.....	17
Digital Output Subsystem Page	19
Counter/Timer Subsystem Page.....	21
Using the KPCI-3100 Series with DriverLINX	23
Introduction.....	23
KPCI-3100 Series Hardware Features	23
DriverLINX Hardware Model for KPCI-3100 Series.....	24
DriverLINX Subsystems	25
DriverLINX Modes	25
DriverLINX Operations and Events	26
Logical Channels	28
Buffers	28
Connecting Signals to the KPCI-3100 Series	29
Analog Input Subsystem Signals	29
Analog Output Subsystem Signals	30
Digital Input Subsystem Signals.....	31
Digital Output Subsystem Signals	33
Counter/Timer Subsystem Signals.....	33
Device Subsystem.....	35
Device Modes	35
Device Operations	35
Analog Input Subsystem	36
Analog Input Modes	36
Analog Input Operations.....	36
Analog Input Timing Events.....	37
Analog Input Start Events.....	43
Analog Input Stop Events	46
Analog Input Channels	50
Analog Input Buffers	56
Analog Input Data Coding.....	57

Analog Input Messages.....	58
Analog Output Subsystem.....	59
Analog Output Modes.....	59
Analog Output Operations.....	59
Analog Output Timing Events.....	60
Analog Output Start Events.....	63
Analog Output Stop Events.....	66
Analog Output Channels.....	67
Analog Output Channel Gains.....	69
Analog Output Buffers.....	70
Analog Output Data Coding.....	71
Analog Output Messages.....	72
Digital Input Subsystem.....	74
Digital Input Modes.....	74
Digital Input Operations.....	74
Digital Input Timing Events.....	75
Digital Input Start Events.....	77
Digital Input Stop Events.....	80
Digital Input Channels.....	85
Digital Input Buffers.....	88
Digital Input Messages.....	89
Digital Output Subsystem.....	90
Digital Output Modes.....	90
Digital Output Operations.....	90
Digital Output Timing Events.....	90
Digital Output Start Events.....	92
Digital Output Stop Events.....	92
Digital Output Channels.....	92
Digital Output Buffers.....	96
Digital Output Messages.....	96
Counter/Timer Subsystem.....	98
Counter/Timer Channel Usage.....	98
Clock Sources and Modes.....	99
Gate Settings.....	101
Counter/Timer Tasks.....	101
Counter/Timer Messages.....	102
Uninstalling DriverLINX	105
How do I uninstall DriverLINX?.....	105
Troubleshooting	109
Solving Problems.....	109
Solving Problems Recognizing and Installing Drivers.....	109
Solving Problems Configuring the Drivers.....	110
Solving Problems Loading Drivers.....	110
Generating a DriverLINX Configuration Report.....	113
What is in the Report?.....	113
How do I Generate the Report?.....	113
Glossary of Terms	114

Preface

Software License and Software Disclaimer of Warranty

This is a legal document which is an agreement between you, the Licensee, and Scientific Software Tools, Inc. By opening this sealed diskette package, Licensee agrees to become bound by the terms of this Agreement, which include the Software License and Software Disclaimer of Warranty.

This Agreement constitutes the complete Agreement between Licensee and Scientific Software Tools, Inc. If Licensee does not agree to the terms of this Agreement, do not open the diskette package. Promptly return the unopened diskette package and the other items (including written materials, binders or other containers, and hardware, if any) that are part of this product to Scientific Software Tools, Inc. for a full refund. No refunds will be given for products that have opened disk packages or missing components.

Licensing Agreement

Copyright. The software and documentation is owned by Scientific Software Tools, Inc. and is protected by both United States copyright laws and international treaty provisions. Scientific Software Tools, Inc. authorizes the original purchaser only (Licensee) to either (a) make one copy of the software solely for backup or archival purposes, or (b) transfer the software to a single hard disk only. The written materials accompanying the software may not be duplicated or copied for any reason.

Trade Secret. Licensee understands and agrees that the software is the proprietary and confidential property of Scientific Software Tools, Inc. and a valuable trade secret. Licensee agrees to use the software only for the intended use under this License, and shall not disclose the software or its contents to any third party.

Copy Restrictions. The Licensee may not modify or translate the program or related documentation **without the prior written consent of Scientific Software Tools, Inc.** All modifications, adaptations, and merged portions of the software constitute the software licensed to the Licensee, and the terms and conditions of this agreement apply to same. Licensee may not distribute copies, including electronic transfer of copies, of the modified, adapted or merged software or accompanying written materials to others. Licensee agrees not to reverse engineer, decompile or disassemble any part of the software.

Unauthorized copying of the software, including software that has been modified, merged, or included with other software, or of the written materials is expressly forbidden. Licensee may not rent, transfer or lease the software to any third parties. Licensee agrees to take all reasonable steps to protect Scientific Software Tools' software from theft, disclosure or use contrary to the terms of the License.

License. Scientific Software Tools, Inc. grants the Licensee only a non-exclusive right to use the serialized copy of the software on a single terminal connected to a single computer. The Licensee may not network the software or use it on more than one computer or computer terminal at the same time.

Term. This License is effective until terminated. This License will terminate automatically without notice from Scientific Software Tools, Inc. if Licensee fails to comply with any term or condition of this License. The Licensee agrees upon such termination to return or destroy the written materials and all copies of the software. The Licensee may terminate the agreement by returning or destroying the program and documentation and all copies thereof.

Limited Warranty

Scientific Software Tools, Inc. warrants that the software will perform substantially in accordance with the written materials and that the program disk, instructional manuals and reference materials are free from defects in materials and workmanship under normal use for 90 days from the date of receipt. All express or implied warranties of the software and related materials are limited to 90 days.

Except as specifically set forth herein, the software and accompanying written materials (including instructions for use) are provided "as is" without warranty of any kind. Further, Scientific Software Tools, Inc. does not warrant, guarantee, or make any representations regarding the use, or the results of the use, of the software or written materials in terms of correctness, accuracy, reliability, currentness, or otherwise. The entire risk as to the results and performance of the software is assumed by Licensee and not by Scientific Software Tools, Inc. or its distributors, agents or employees.

EXCEPT AS SET FORTH HEREIN, THERE ARE NO OTHER WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WITH RESPECT TO THE SOFTWARE, THE ACCOMPANYING WRITTEN MATERIALS, AND ANY ACCOMPANYING HARDWARE.

Remedy. Scientific Software Tools' entire liability and the Licensee's exclusive remedy shall be, at Scientific Software Tools' option, either (a) return of the price paid or (b) repair or replacement of the software or accompanying materials. In the event of a defect in material or workmanship, the item may be returned within the warranty period to Scientific Software Tools for a replacement without charge, provided the licensee previously sent in the limited warranty registration board to Scientific Software Tools, Inc., or can furnish proof of the purchase of the program. This remedy is void if failure has resulted from accident, abuse, or misapplication. Any replacement will be warranted for the remainder of the original warranty period.

NEITHER SCIENTIFIC SOFTWARE TOOLS, INC. NOR ANYONE ELSE WHO HAS BEEN INVOLVED IN THE CREATION, PRODUCTION, SALE OR DELIVERY OF THIS PRODUCT SHALL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, OR INCIDENTAL DAMAGES (INCLUDING DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION AND THE LIKE) ARISING OUT OF THE USE OF OR THE INABILITY TO USE SUCH PRODUCT EVEN IF SCIENTIFIC SOFTWARE TOOLS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, OR LIMITATIONS ON DURATION OF AN IMPLIED WARRANTY, THE ABOVE LIMITATIONS MAY NOT APPLY TO LICENSEE.

This agreement is governed by the laws of the Commonwealth of Pennsylvania.

About DriverLINX

Welcome to DriverLINX® for Microsoft® Windows™, the high-performance real-time data-acquisition device drivers for Windows application development.

DriverLINX is a language- and hardware-independent application programming interface designed to support hardware manufacturers' high-speed analog, digital, and counter/timer data-acquisition boards in Windows. DriverLINX is a multi-user and multitasking data-acquisition resource manager providing more than 100 services for foreground and background data acquisition tasks.

Included with your DriverLINX package are the following items:

- The DriverLINX API DLLs and drivers supporting your data-acquisition hardware
- Analog I/O Panel, a DriverLINX program that verifies the installation and configuration of DriverLINX for your analog input/output board and demonstrates several virtual bench-top instruments
- Source code for the sample programs
- The DriverLINX Application Programming Interface files for your compiler
- DriverLINX On-line Help System
- *DriverLINX 4.0 Installation and Configuration Guide*
- *DriverLINX Analog I/O Programming Guide*
- *DriverLINX Digital I/O Programming Guide*
- *DriverLINX Counter/Timer Programming Guide*
- *DriverLINX Technical Reference Manual*
- Supplemental Documentation on DriverLINX and your data-acquisition hardware

About This User's Guide

The purpose of this manual is to help you quickly learn how to configure and use the hardware features of Keithley's KPCI-3100 Series boards with DriverLINX.

- For help installing and configuring your hardware and DriverLINX, please see the manual that accompanied your hardware and the *DriverLINX 4.0 Installation and Configuration Guide* for your version of Windows.
- For more information on the DriverLINX API, please see the *DriverLINX Technical Reference Manual*.
- For additional help programming your board, please examine the source code examples on the Distribution Disks.

This manual contains the following chapters:

Configuring the KPCI-3100 Series

Shows how to configure the KPCI-3100 Series using the *Configure DriverLINX Device* dialog box.

Using the KPCI-3100 Series with DriverLINX

Shows how to set up DriverLINX with the *Edit Service Request* dialog box to use KPCI-3100 Series hardware features.

Uninstalling DriverLINX

Describes how to remove DriverLINX hardware drivers and other files.

Troubleshooting

Gives troubleshooting tips for installing, configuring, and loading DriverLINX drivers.

Conventions Used in This Manual

The following notational conventions are used in this manual:

- A round bullet (●) identifies itemized lists.
- Numbered lists indicate a step-by-step procedure.
- DriverLINX Application Programming Interface and Windows macro and function names are set in bold when mentioned in the text.
- **DriverLINX** indicates the exported function name of the device driver DLL while DriverLINX indicates the product as a whole.
- DriverLINX Application Programming Interface identifiers, menu items, and Dialog Box names are italicized when mentioned in the text.
- *Italics* are used for emphasis.
- Source code and data structure examples are displayed in Courier typeface and bounded by a box with a single line.

Code

- A box with a double line bound tables of information.

Table

Concept

- Important concepts and notes are printed in the left margin.

Configuring the KPCI-3100 Series

Introduction

The installation program provides general instructions for installing and configuring DriverLINX. This manual explains the steps and special features that apply to Keithley's KPCI-3100 Series boards.

Installing and configuring DriverLINX for a Keithley KPCI-3100 Series board requires three steps:

1. **Install DriverLINX.** Follow the instructions given by the installation program. The *Read Me First* instructions explain the components and drivers you can install.
2. **Configure DriverLINX.** This creates a Logical Device, which stores configuration information for your board. See "Configure DriverLINX Device Dialog" on page 11 for configuration options specific to a Keithley KPCI-3100 Series model.
3. **Install your KPCI-3100 hardware.** Follow the instructions in your hardware manual.

After configuring DriverLINX, installing your board and restarting Windows, reopen the *DriverLINX Configuration Panel* to make sure that DriverLINX loaded the Logical Device for your board. If the Logical Device is not loaded, the Event Log may have a message from the driver that explains why. You can check the Event Log using the *DriverLINX Event Viewer* on the Windows Start Menu.

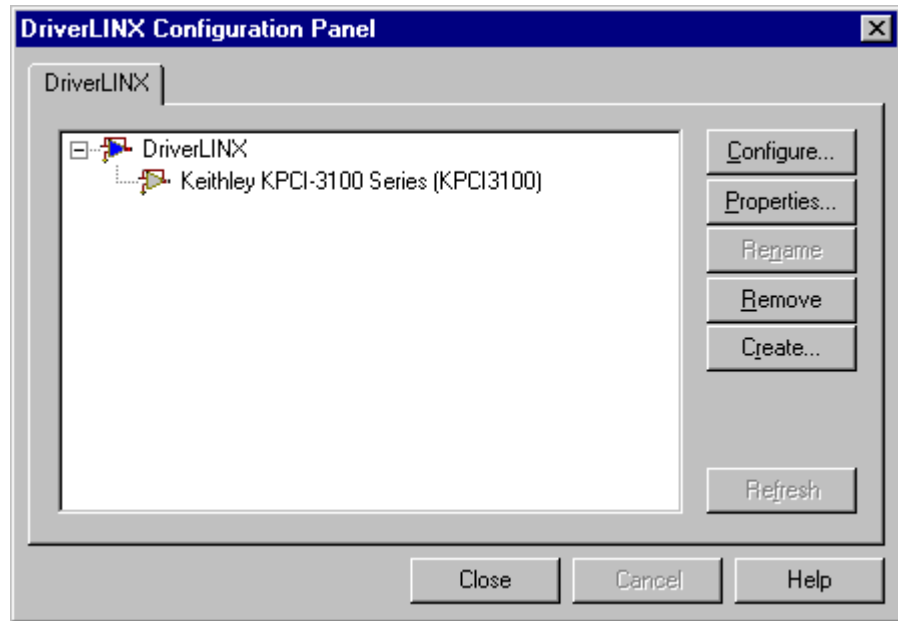
Windows NT

Under Windows NT 4.0, a Logical Device may not load because the operating system does not always configure Plug-and-Play PCI devices properly. To work around this, set your computer's BIOS to configure Plug-and-Play devices before it starts the operating system. On various computers the BIOS setting is called "Plug-and-Play Aware OS – Disabled" or "Plug & Play OS – No".

Configure DriverLINX Device Dialog

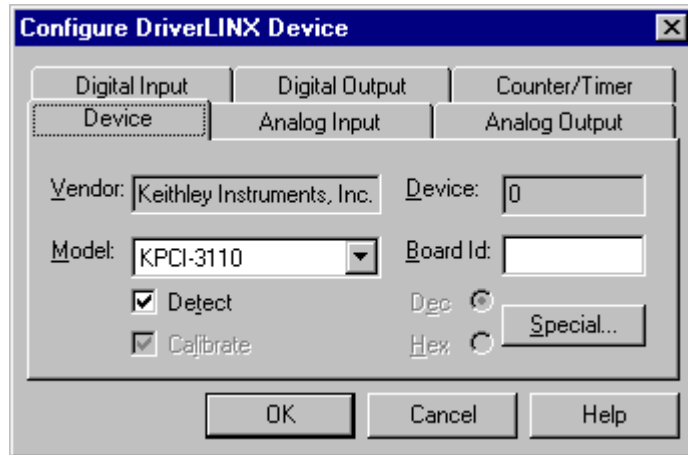
DriverLINX uses a standardized configuration protocol for all data-acquisition hardware. Configuration assigns an identifying device number to a specific KPCI-3100 Series board in your computer and allows you to enable or disable bus mastering.

The installation program automatically starts the *DriverLINX Configuration Panel*. To start it now, use the shortcut on the Windows Start Menu.



When you click the *Configure...* button on the *DriverLINX Configuration Panel*, DriverLINX displays the *Configure DriverLINX Device* dialog. The dialog has a page for each subsystem on the Keithley KPCI-3100 Series. The following sections describe your choices in configuring DriverLINX to work with your board.

Device Subsystem Page



Use the Device subsystem page to tell DriverLINX the model name of, and, optionally, the accessories connected to your KPCI-3100 Series board.

Vendor

The *Vendor* property displays “Keithley Instruments, Inc.” It is a read-only property.

Device

The *Device* property designates the Logical Device you are configuring. It is a read-only property. To change it, first save (**OK**) or quit (**Cancel**) the current configuration. Then select or create a new Logical Device using the *DriverLINX Configuration Panel*.

Model

The *Model* property selects or indicates the hardware model of the board you’re configuring.

Windows NT

Select one of the following models:

- KPCI-3101
- KPCI-3102
- KPCI-3103
- KPCI-3104
- KPCI-3110
- KPCI-3116

Windows 95/98

Windows 95 automatically determines the model of your board so DriverLINX disables Model selection.

Board Id

The KPCI-3100 Series boards do not have unique serial numbers that DriverLINX can use to match a Logical Device to a board while loading the driver. Under Windows 95/98, DriverLINX can match a Logical Device to a board based on the

model and bus slot. Under Windows NT, DriverLINX matches a Logical Device to a board based on model only.

Windows NT

DriverLINX does not use the Board Id property.

Windows 95/98

DriverLINX assigns a Board Id based on the model and bus slot.

Detect

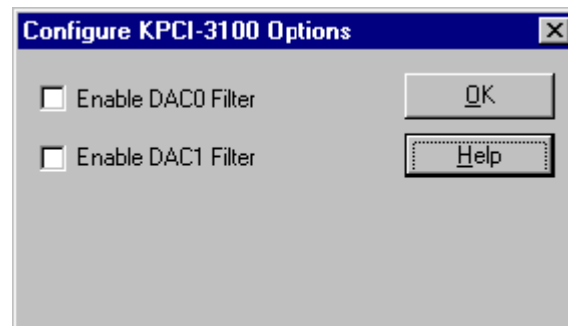
The *Detect* property enables and disables DriverLINX's hardware detection and testing algorithms. For maximum system reliability, always leave this check-box marked.

Calibrate

The *Calibrate* property enables and disables hardware auto-calibration. This option is grayed-out for the KPCI-3100 Series because it does not support automatic calibration.

Special...

The *Special...* button displays the following dialog box of KPCI-3100 Series-specific configuration options:



Note: The Filter settings apply only to model KPCI-3116.

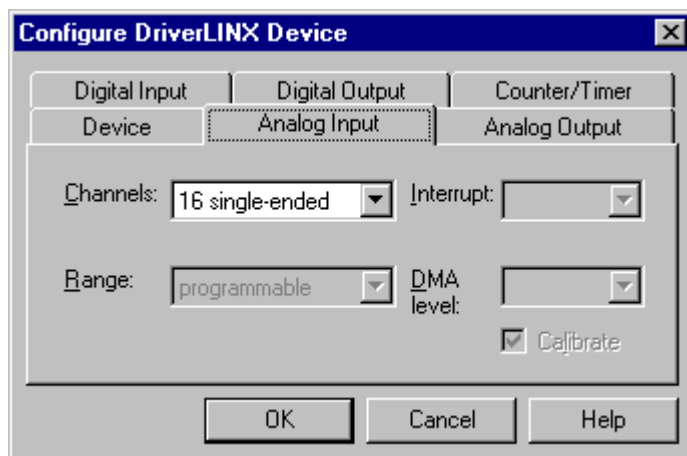
Enable DAC Filters

Model KPCI-3116 has 20 kHz filters on its analog outputs that you can configure as initially enabled or disabled. Applications can use the default configuration or change the setting programmatically.

To enable the filters as the default configuration, perform the following steps:

1. In the *Configure KPCI-3100 Options* dialog box, check the **Enable DAC0 Filter** box for Logical Channel 0
2. In the *Configure KPCI-3100 Options* dialog box, check the **Enable DAC1 Filter** box for Logical Channel 1

Analog Input Subsystem Page



Use the Analog Input subsystem page to choose between single-ended or differential analog input channels as a default configuration.

Channels

On the KPCI-3100 Series, each Analog Input channel can use single-ended or differential termination modes. When configuring the Analog Input Subsystem, you choose a default configuration for all channels. Applications can use the default configuration or specify the termination mode for each channel that it uses. This scheme supports applications that use KPCI-3100-specific features as well as those that use only generic features. For applications that do specify the termination mode, configure the subsystem for single-ended channels so that all channels are available.

For information on programming the termination mode, see “Analog Input Channels” on page 50.

Range

The analog input ranges for the KPCI-3100 Series are fully software programmable. DriverLINX grays out this property in the configuration dialog.


Interrupt

The KPCI-3100 Series does not use interrupts. DriverLINX disables this property.

DMA

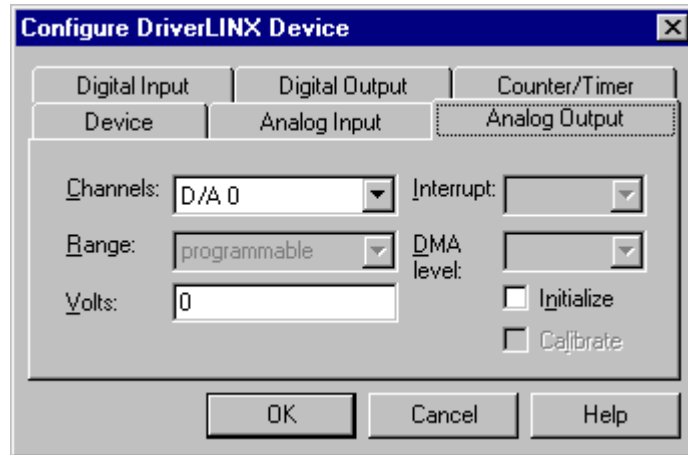
The KPCI-3100 Series does not use DMA channels for direct memory transfers. PCI devices use bus mastering for DMA.

Calibrate

The KPCI-3100 Series supports manual calibration only. You can start the calibration utility from the Windows Start menu .



Analog Output Subsystem Page



Use the Analog Output subsystem page to set or view the initial output voltages.

Channels

Lists the analog output channels on the board and selects a channel for the Volts and Initialize properties.

Range

The analog output ranges for the KPCI-3100 Series are fully software programmable. DriverLINX grays out this property in the configuration dialog.

Volts

The *Initialization Value* property specifies the analog output value DriverLINX will write to the selected Logical Channel upon hardware initialization. DriverLINX only writes this value if you enable the *Initialize* check box.

Interrupt

The KPCI-3100 Series does not use interrupts. DriverLINX disables this property.

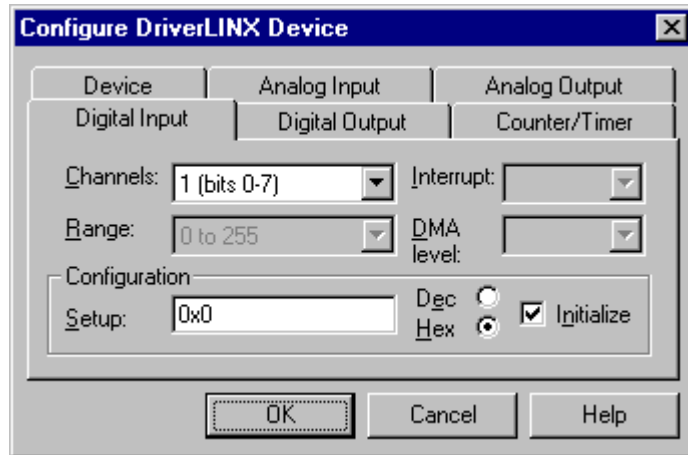
DMA

The KPCI-3100 Series does not use DMA channels for direct memory transfers. DriverLINX disables this property.

Initialize

Checking the *Initialize* check box instructs DriverLINX to use the *Volts* property for to initialize the selected analog output channel.

Digital Input Subsystem Page



Use the Digital Input subsystem page to set configurable digital channels as input or output.

Channels

The *Channels* property allows you to select a Logical Channel for configuration or viewing the channel's range.

The KPCI-3100 Series supports both fixed and configurable digital channels. DriverLINX defines the following Logical Channels for the KPCI-3100 Series Digital Input Subsystem:

Models KPCI-3101/02/03/04

Logical Channel	DriverLINX Function	KPCI-3100 Series External Connector
0	Digital Input/Output	Digital I/O Bank A 0 ... 7
1	Digital Input/Output	Digital I/O Bank B 0 ... 7
2	Digital Input/Output	Digital I/O Bank C 0 ... 6
3	External Clock	External A/D Sample Clock Input External D/A Sample Clock Input
4	External Trigger	External A/D TTL Trigger External D/A TTL Trigger

Models KPCI-3110/16

Logical Channel	DriverLINX Function	KPCI-3100 Series External Connector
0	Digital Input/Output	Digital I/O Bank A 0 ... 7
1	Digital Input/Output	Digital I/O Bank B 0 ... 7
2	External Clock	External A/D Sample Clock Input External D/A Sample Clock Input
3	External Trigger	External A/D TTL Trigger External D/A TTL Trigger

Range

The *Range* property specifies the supported digital input range for the selected Logical Channel. This is a read-only property.

Interrupt

The KPCI-3100 Series does not use interrupts. DriverLINX disables this property.

DMA

The KPCI-3100 Series does not use DMA channels for direct memory transfers. DriverLINX disables this property.

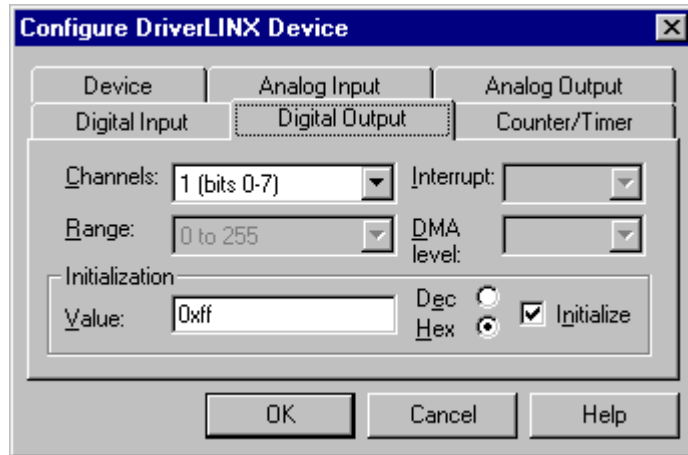
Configuration Setup

The *Configuration Setup* property specifies the hardware configuration of the digital I/O ports. Logical Channels 0 through 1 are configurable as input or output. Enter 1 to initially configure the selected channel as input or 0 to configure it as output and check the *Initialize* box.

Initialize

Checking the *Initialize* check box instructs DriverLINX to use the *Configuration Setup* property to configure the selected digital I/O channel.

Digital Output Subsystem Page



Use the Digital Output subsystem page to change the default digital output port initialization values.

Channels

The *Channels* property allows you to select a Logical Channel for initialization or viewing the channel's range.

Models KPCI-3101/02/03/04

Logical Channel	DriverLINX Function	KPCI-3100 Series External Connector
0	Digital Input/Output	Digital I/O Bank A 0 ... 7
1	Digital Input/Output	Digital I/O Bank B 0 ... 7
2	Digital Input/Output	Digital I/O Bank C 0 ... 6

Models KPCI-3110/16

Logical Channel	DriverLINX Function	KPCI-3100 Series External Connector
0	Digital Input/Output	Digital I/O Bank A 0 ... 7
1	Digital Input/Output	Digital I/O Bank B 0 ... 7

Range

The *Range* property specifies the supported digital output range for the selected Logical Channel. This is a read-only property.

Interrupt

The KPIC-3100 Series does not use interrupts. DriverLINX disables this property.

DMA

The KPCI-3100 Series does not use DMA channels for direct memory transfers. DriverLINX disables this property.

Initialization Value

The *Initialization Value* property specifies the digital output value DriverLINX will write to the selected Logical Channel upon hardware initialization. DriverLINX only writes this value if you enable the *Initialize* check box.

Initialize

Checking the *Initialize* check box instructs DriverLINX to use the *Initialization Value* property for digital output port initialization.

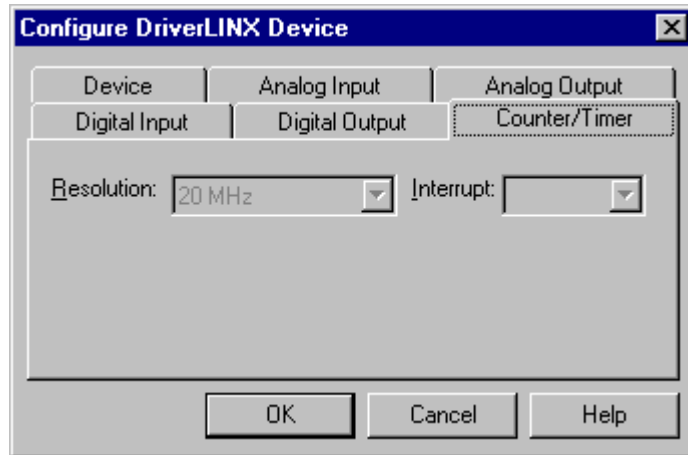
Dec

This check box converts the *Initialization Value* property to decimal.

Hex

This check box converts the *Initialization Value* property to hexadecimal.

Counter/Timer Subsystem Page



Use the Counter/Timer subsystem page to view the clock source frequency.

Resolution

The *Resolution* property specifies the clock frequency of the master oscillator. All models have a 20 MHz clock source for pacing input/output tasks and counter/timer functions.

Interrupt

The KPIC-3100 Series does not use interrupts. DriverLINX disables this property.

Using the KPCI-3100 Series with DriverLINX

Introduction

See the Analog I/O Programming Guide for an overview of DriverLINX programming.

This chapter shows you how to set up and use KPCI-3100 Series hardware features with DriverLINX.

The descriptions here use the *Edit Service Request* dialog for language and API independence. For the correct syntax with the language you're using, please see the *DriverLINX Technical Reference Manuals*. For DriverLINX examples in your programming language, please see the source code examples in the subdirectories of your DriverLINX installation directory or on the original distribution media.

KPCI-3100 Series Hardware Features

The KPCI-3100 Series offers your data-acquisition application a wide variety of useful features. Analog inputs support programmable gains, termination modes and polarities, as well as digital and analog pre-, post- and about-triggering. Analog outputs are waveform-quality (on some models). Four of the counter/timers are user-accessible. DriverLINX accesses these features through its hardware independent Applications Programming Interface (API).

The following table is a cross-reference between hardware features and the DriverLINX features that access them.

Hardware Feature	DriverLINX Feature
Calibration of each analog input and output range	DriverLINX provides a stand-alone <i>KPCI-3100 Calibration Utility</i> .
Analog triggering	<p>Analog Input:</p> <ul style="list-style-type: none"> • Analog Start Event (Post-Triggering) • Analog Stop Event (Pre- and About-Triggering) <p>Analog Output:</p> <ul style="list-style-type: none"> • Analog Start Event (Post-Triggering) <p>Digital Input:</p> <ul style="list-style-type: none"> • Analog Start Event (Post-Triggering) • Analog Stop Event (Pre- and About-Triggering)
Digital triggering	<p>Analog Input:</p> <ul style="list-style-type: none"> • Digital Start Event (Post-Triggering) <p>Analog Output:</p> <ul style="list-style-type: none"> • Digital Start Event (Post-Triggering) <p>Digital Input:</p> <ul style="list-style-type: none"> • Digital Start Event (Post-Triggering) • Digital Stop Event (Pre- and About-Triggering)
Programmable single-ended or differential termination modes	Analog Input Channel Termination Modes
Programmable gains	Analog Input Channel Gains
Waveform-quality analog outputs (models KPCI-3110/16 only)	Analog Output Subsystem AO Recycle Mode (waveform generation)
User-accessible counter/timers	Counter/Timer Subsystem

DriverLINX Hardware Model for KPCI-3100 Series

DriverLINX provides a portable, hardware-independent API for data-acquisition boards while still allowing applications to access unique or proprietary hardware features of specific products. To achieve this goal, DriverLINX maps a hardware-independent, or abstract, data-acquisition model onto KPCI-3100 Series hardware capabilities.

The following sections describe how DriverLINX implements KPCI-3100 Series hardware features as Subsystems, Modes, Operations, Events, Logical Channels, Buffers, and Messages.

DriverLINX Subsystems

The KPCI-3100 Series supports the following DriverLINX subsystems:

1. **Device**—refers to a KPCI-3100 Series board as a whole.
2. **Analog Input**—refers to the analog input channels, clocks, and control signals.
3. **Analog Output**—refers to the analog output channels, clocks, and control signals.
4. **Digital Input**—refers to the digital input ports as well as 1-bit digital control signals and external clock inputs.
5. **Digital Output**—refers to the digital output ports.
6. **Counter/Timer**—refers to the onboard counter/timer channels for pacing analog input/output or performing measuring and waveform generation functions.

DriverLINX Modes

Applications use modes in Service Requests to advise DriverLINX on their preferred hardware data transfer technique. The DriverLINX modes fall into two general classes:

- **Foreground or synchronous modes.** The calling application does not regain control until DriverLINX completes the Service Request. DriverLINX supports this mode for simple, single value I/O operations or software housekeeping functions that DriverLINX can complete without a significant delay.
- **Background or asynchronous modes.** The calling application regains control as soon as DriverLINX initiates the task. The calling application must synchronize with the data-acquisition task using status polling or DriverLINX's messages (preferred). DriverLINX supports this mode for buffered data transfers or for commands that require a significant time to complete.

DriverLINX supports four modes with the KPCI-3100 Series for its commands (Service Requests).

- **Polled Mode**—This is a foreground or synchronous operation. DriverLINX supports this mode for simple, single-value I/O operations that the data-acquisition board can complete without significant delay.
- **Interrupt Mode**—This is a background or asynchronous operation. DriverLINX transfers data between the computer's memory and the data-acquisition board using hardware interrupts and programmed I/O transfers.

- **DMA Mode**—This is a background or asynchronous operation. DriverLINX programs the data-acquisition board to transfer data between the computer’s memory and the board.
- **Other Mode**—This is a foreground or synchronous operation. DriverLINX supports this mode for initialization, configuration, calibration, data conversion, and timebase operations.

The following table summarizes the data acquisition modes that DriverLINX supports for each subsystem with the Keithley KPCI-3100 Series.

Subsystem	Polled	Interrupt	DMA	Other
Analog Input	✓	✓*	✓	✓
Analog Output	✓	✓*	✓	✓
Digital Input	✓	✓*	✓	✓
Digital Output	✓			✓
Counter/Timer	✓			✓
Device				✓

KPCI-3100 Series Supported DriverLINX Modes.

* Interrupt mode is simulated by the driver for compatibility with applications written for other hardware.

DriverLINX Operations and Events

Applications construct DriverLINX data-acquisition tasks by combining a small number of DriverLINX operations and events in many possible ways. The following table summarizes the operations and events that DriverLINX supports for the Keithley KPCI-3100 Series. Later sections for each DriverLINX subsystem will describe the operations and events in more detail.

Note: In addition to the operations shown in the table below, all subsystems allow the *MESSAGE* operation in any Mode.

Subsystem	Operation	Events		
		Timing	Start	Stop
Mode				
Analog Input				
Polled	Start	null	null, cmd	null, TC
Interrupt	Start, Stop, Status, Convert	dig, rate	cmd, ana [*] , dig	cmd, TC, ana [*] , dig
DMA	Start, Stop, Status, Convert	dig, rate	cmd, ana [*] , dig	cmd, TC, ana [*] , dig
Other	Initialize			
Analog Output (models KPCI-3102/04/10/16 only)				
Polled	Start	null	null, cmd	null, TC
Interrupt [*]	Start, Stop, Status, Convert	dig, rate	cmd, ana, dig	cmd, TC
DMA [*]	Start, Stop, Status, Convert	dig, rate	cmd, ana, dig	cmd, TC
Other	Initialize			
Digital Input				
Polled	Start	null	null, cmd	null, TC
Interrupt	Start, Stop, Status	rate	cmd, ana [*] , dig	cmd, TC, ana [*] , dig
DMA	Start, Stop, Status	rate	cmd, ana [*] , dig	cmd, TC, ana [*] , dig
Other	Initialize, Configure	DIO Setup		
Digital Output				
Polled	Start	null	null, cmd	null, TC
Other	Initialize, Configure	DIO Setup		
Counter/Timer				
Polled	Start, Stop, Status	null, rate	null, cmd	null, TC
Other	Initialize, Configure			
Device				
Other	Initialize, Capabilities			

Allowed Operations and Events for KPCI-3100 Series Subsystems and Modes.

^{*}Models KPCI-3110/16 only.

The following list explains the Event abbreviations in the preceding table:

- **null**—Null or None Event specifies when a Service Request does not require an event
- **cmd**—Command Event specifies when DriverLINX starts or stops a task on software command
- **TC**—Terminal Count Event specifies when DriverLINX processes all data buffers once
- **rate**—Rate Event specifies how DriverLINX paces or clocks data transfer
- **dig**—Digital Event specifies a trigger, clock, or other control signal to pace, start, or stop a task
- **ana**—Analog Event specifies a trigger to start or stop a task
- **DIO Setup**—DIO Setup Event assigns a digital channel to either the Digital Input or Digital Output Subsystem.

Logical Channels

DriverLINX designates the individually addressable hardware channels for each subsystem as “Logical Channels.” Generally, the zero-based Logical Channel numbering sequence closely follows the hardware manufacturer’s channel numbering scheme.

In some cases, however, DriverLINX assigns Logical Channel numbers to hardware features that users don’t commonly think of as “channels.” For instance, DriverLINX commonly models external hardware clock input lines, external hardware trigger input lines, and external interrupt inputs as 1-bit digital Logical Channels. In other cases, DriverLINX models subsystem-specific features, such as internal pacer clocks, as members of a more general-purpose set of counter/timer channels.

For a list of DriverLINX assigned Logical Channel numbers, see the notes on each supported subsystem.

Buffers

Applications usually use data buffers to exchange data between the application and the data-acquisition hardware. When using data buffers, please note the following points about DriverLINX’s data buffers:

- DriverLINX supports data-acquisition tasks with 1 to 255 data buffers per task.
- DriverLINX imposes no size limits on a single buffer, although the operating system or some hardware products may have size restrictions.
- User applications must allow DriverLINX to allocate all data buffers to guarantee application portability to different hardware and operating systems and to insure that the hardware can physically access the buffer memory.
- User applications usually don’t have concurrent or immediate access to the in-use data buffer while DriverLINX is executing a data-acquisition task.

Connecting Signals to the KPCI-3100 Series

The Keithley hardware manual describes the data and control signals for the KPCI-3100 Series and the connector pinouts for these signals. This section summarizes how DriverLINX numbers the I/O data signals and how DriverLINX uses the control connections for external clock, trigger, and gating inputs.

Analog Input Subsystem Signals

The Analog Input subsystem has 16 or 32 analog inputs, depending on the model, and an analog ground. You set the default subsystem configuration as 8 or 16 differential, or 16 or 32 single-ended channels. Applications can then use the default configuration or request a specific configuration for each channel. (See “Analog Input Channels” on page 50 for programming information.)

DriverLINX maps these connections to Logical Channels as shown in the following table:

Models KPCI-3101/02/03/04

Physical Channels	Connector Name	Logical Channels
0 – 7 Differential	Analog Input 0 ... Analog Input 7 Analog Input 0 Return ... Analog Input 7 Return	0 – 7
0 – 15 Single-ended	Analog Input 0 ... Analog Input 15 Amp Low	0 – 15

Models KPCI-3110/16

Physical Channels	Connector Name	Logical Channels
0 – 15 Differential	Analog Input 0 ... Analog Input 15 Analog Input 0 Return ... Analog Input 15 Return	0 – 15
0 – 31 Single-ended	Analog Input 0 ... Analog Input 31 Amp Low	0 – 31

How DriverLINX maps analog input hardware channels to Logical Channels.

Analog Input Pacing, Triggering and Gating Signals

Analog input tasks can use only Counter/Timer Logical Channel 4 for pacing. The clock sources can be internal or external. The KPCI-3100 Series does not support gating for Analog Input tasks.

The Analog Input subsystem can use several control signals that DriverLINX defines as external clocks and triggers as shown in the following table:

Connector Name	DriverLINX Usage
External A/D TTL Trigger	External trigger: <ul style="list-style-type: none"> Digital Start Event (Post-Triggering) Digital Stop Event (Pre- and About-Triggering) External pacer clock: <ul style="list-style-type: none"> Burst Generator: External Clocking
External A/D Sample Clock Input	Divided external pacer clock: <ul style="list-style-type: none"> Rate Generator: External Clocking
Analog Trigger	External trigger: <ul style="list-style-type: none"> Analog Start Event (Post-Triggering) Analog Stop Event (Pre- and About-Triggering)
Analog Input	External trigger: <ul style="list-style-type: none"> Analog Start Event (Post-Triggering) Analog Stop Event (Pre- and About-Triggering)

How DriverLINX uses analog input control signals.

Analog Output Subsystem Signals

The Analog Output subsystem, for models KPCI-3102, 3104, 3110 and 3116, has 2 analog differential output connections.

DriverLINX maps these connections to Logical Channels as shown in the following table:

Models KPCI-3102/04/10/16

Physical Channel	Connector Name	Logical Channel
0	Analog Output 0+, Analog Output 0 Return	0
1	Analog Output 1+, Analog Output 1 Return	1

How DriverLINX maps analog output hardware channels to Logical Channels.

Analog Output Pacing, Triggering and Gating Signals

Analog output tasks on models KPCI-3110 and 3116 can use Counter/Timer Logical Channel 5 for pacing with internal or external clock sources to produce waveform quality signals. Models KPCI-3102 and 3104 support polled transfers only. The KPCI-3100 Series does not support gating for Analog Output tasks.

The Analog Output subsystem can use several control signals that DriverLINX defines as external clocks and gates as shown in the following table:

Connector Name	DriverLINX Usage
External D/A TTL Trigger	External trigger: <ul style="list-style-type: none"> Digital Start Event (Post-Triggering) External pacer clock: <ul style="list-style-type: none"> Burst Generator: External Clocking
External D/A Sample Clock Input	Divided external pacer clock: <ul style="list-style-type: none"> Rate Generator: External Clocking
Analog Trigger	External trigger: <ul style="list-style-type: none"> Analog Start Event (Post-Triggering)
Analog Input	External trigger: <ul style="list-style-type: none"> Analog Start Event (Post-Triggering)

How DriverLINX uses analog output control signals.

Digital Input Subsystem Signals

The Digital Input Subsystem has two or three digital input/output ports, depending on the model, as well as four control inputs which DriverLINX models as two 1-bit logical digital input channels. DriverLINX maps these signals to Logical Channels as shown in the following table:

Models KPCI-3101/02/03/04

Logical Channel	DriverLINX Function	KPCI-3100 Series External Connector
0	Digital Input/Output	Digital I/O Bank A 0 ... 7
1	Digital Input/Output	Digital I/O Bank B 0 ... 7
2	Digital Input/Output	Digital I/O Bank C 0 ... 6
3	External Clock	External A/D Sample Clock Input External D/A Sample Clock Input
4	External Trigger	External A/D TTL Trigger External D/A TTL Trigger

Models KPCI-3110/16

Logical Channel	DriverLINX Function	KPCI-3100 Series External Connector
0	Digital Input/Output	Digital I/O Bank A 0 ... 7
1	Digital Input/Output	Digital I/O Bank B 0 ... 7
2	External Clock	External A/D Sample Clock Input External D/A Sample Clock Input
3	External Trigger	External A/D TTL Trigger External D/A TTL Trigger

How DriverLINX maps digital input hardware channels to Logical Channels.

Notes:

- If a channel is configured for output, reading it using the Digital Input subsystem returns the last value written.
- The External Trigger channel is not available for reading but is available for clocking and triggering.
- Applications can assign a configurable channel to either digital subsystem using a Configure operation. (See “Digital Channel Configuration” on page 74.)

Digital Input Pacing, Triggering and Gating Signals

Digital input tasks are either polled mode or DMA mode. In DMA mode, the board uses the analog input circuitry to sample digital Banks A and B, and therefore supports the same pacing and triggering options as for analog input.

The Digital Input subsystem can use several control signals that DriverLINX defines as external clocks and gates as shown in the following table:

Connector Name	DriverLINX Usage
External A/D TTL Trigger	External trigger: <ul style="list-style-type: none">• Digital Start Event (Post-Triggering)• Digital Stop Event (Pre- and About-Triggering) External pacer clock: <ul style="list-style-type: none">• Burst Generator: External Clocking
External A/D Sample Clock Input	Divided external pacer clock: <ul style="list-style-type: none">• Rate Generator: External Clocking
Analog Trigger	External trigger: <ul style="list-style-type: none">• Analog Start Event (Post-Triggering)• Analog Stop Event (Pre- and About-Triggering)
Analog Input	External trigger: <ul style="list-style-type: none">• Analog Start Event (Post-Triggering)• Analog Stop Event (Pre- and About-Triggering)

How DriverLINX uses digital input control signals.

Digital Output Subsystem Signals

The Digital Output Subsystem has two or three digital input/output ports, depending on the model. DriverLINX maps these signals to Logical Channels as shown in the following table:

Models KPCI-3101/02/03/04

Logical Channel	DriverLINX Function	KPCI-3100 Series External Connector
0	Digital Input/Output	Digital I/O Bank A 0 ... 7
1	Digital Input/Output	Digital I/O Bank B 0 ... 7
2	Digital Input/Output	Digital I/O Bank C 0 ... 6

Models KPCI-3110/16

Logical Channel	DriverLINX Function	KPCI-3100 Series External Connector
0	Digital Input/Output	Digital I/O Bank A 0 ... 7
1	Digital Input/Output	Digital I/O Bank B 0 ... 7

How DriverLINX maps digital output hardware channels to Logical Channels.

Notes:

- If a channel is configured for output, reading it (using the Digital Input subsystem) returns the last value written.
- Applications can assign a configurable channel to either digital subsystem using a Configure operation. (See “Digital Channel Configuration” on page 74.)

Digital Output Pacing, Triggering and Gating Signals

Digital output tasks can use only polled mode and therefore have no pacing, triggering or gating signals. However, the KPCI-3100 Series does support Dynamic Digital Output from the Analog Input subsystem. For more information on this feature, see “Writing Digital Output Channels with the Analog Input Subsystem” on page 54.

Counter/Timer Subsystem Signals

The Counter/Timer subsystem has a Logical Channel for each of the counter/timer chip’s six counter/timers. The first four have external clock, gate and output pins but support only counter/timer tasks. The last two have only clock inputs but do support pacing tasks on other subsystems.

Analog I/O tasks can use the board’s external trigger inputs (External A/D and D/A TTL Triggers) for pacing instead of the output of a counter/timer, and the board’s external clock inputs (External A/D or D/A Sample Clock Inputs) instead of the internal clock.

DriverLINX maps these signals as shown in the following table:

Model KPCI-3130

Logical Channel	DriverLINX Function	KPCI-3130 Series External Connector
0	External Clock (sample on DAC overload)	+/- DAC Overload Sense Lines 0 ... 7

How DriverLINX maps counter/timer hardware channels to Logical Channels.

Device Subsystem

The following sections describe how DriverLINX implements Device Subsystem features for the KPCI-3100 Series.

Device Modes

The Device Subsystem supports only DriverLINX's *Other* mode for all operations.

Device Operations

The KPCI-3100 Series Device Subsystem supports the following DriverLINX operations:

If another application is using the same data-acquisition board, DriverLINX will prevent Device Initialization from interfering with another application's data-acquisition tasks.

- **Initialize**—DriverLINX aborts all data-acquisition tasks for every subsystem controlled by the current application. DriverLINX then initializes each subsystem.
- **Capabilities**—DriverLINX provides hardware-specific and configuration information in the form of a Logical Device Descriptor database. (If you are using the DriverLINX ActiveX controls, access the Logical Device Descriptor with a DriverLINXLDD control rather than with this operation.)

Analog Input Subsystem

The following sections describe how DriverLINX implements Analog Input Subsystem features for the KPCI-3100 Series.

Analog Input Modes

The Analog Input Subsystem supports the following modes:

- **Polled**—For single-value or single-scan analog input samples.
- **Interrupt**—For buffered transfers using programmed I/O.
- **DMA**—For buffered transfers using direct memory access.
- **Other**—For subsystem initialization and data conversion.

Analog Input Operations

The KPCI-3100 Series Analog Input Subsystem supports the following DriverLINX operations:

- **Initialize**—aborts all active analog input data-acquisition tasks. However, DriverLINX prevents one application from interfering with another application's data-acquisition tasks.
- **Start**—initiates a data-acquisition task using the Mode, Timing, Start, and Stop Events, the Logical Channels, and the Buffers the application specified in the Service Request.
- **Status**—reports the buffer position of the next sample that DriverLINX will write into a buffer.
- **Stop**—terminates an analog input data-acquisition task.
- **Message**—displays a pop-up dialog box for the user containing the text for the current DriverLINX error message.

Analog Input Timing Events

Timing Events specify how the hardware paces or clocks the acquisition of analog input samples. DriverLINX uses the Timing Event to program when the KPCI-3100 Series acquires the next analog input sample.

The KPCI-3100 Series supports the following Timing Events for analog input:



- **None**—Sampling requires no pacing as DriverLINX is acquiring only a single value or scan.
- **Rate**—The KPCI-3100 Series supports fixed rate and burst mode sampling using internal and external clocks.

None or Null Timing Event

The Null Event specifies that the task does not need a clock to determine when to acquire the next sample.

Rate Timing Event

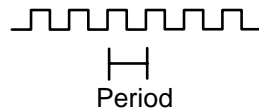
The KPCI-3100 Series supports the following Rate Events for analog input:

- **Rate Generator**—Generates a fixed rate clock with equal time intervals between tics.

- **Burst Generator**—Generates a dual frequency clock with a fixed number of tics at a high frequency separated by a time interval at a lower frequency.


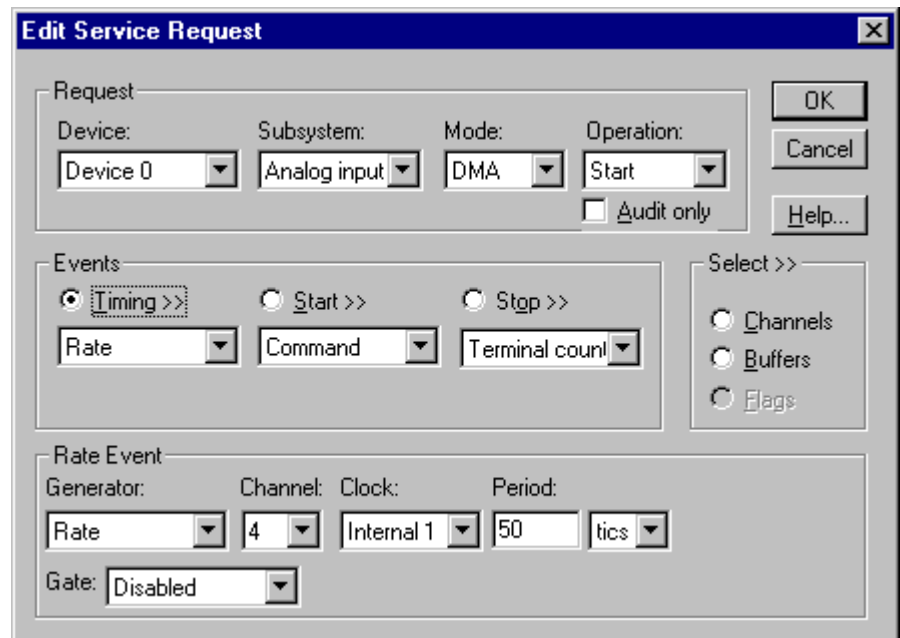
KPCI-3100 Series boards have a fixed 20 MHz master clock frequency. The sample period range depends on the model. All models limit the period to $2^{24} - 1$ tics, or 838,860 μ s (1.20 Hz). Consult your hardware manual for details.

Rate Generator: Internal Clocking

An internally clocked Rate Generator produces a fixed rate clock with equal time intervals between tics.



Use an internally clocked rate generator when you want to acquire all analog input samples at equally spaced time intervals.



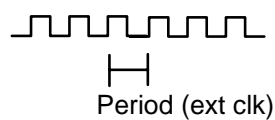
How to set up the KPCI-3100 Series for fixed rate sampling using an internal clock.

For hardware independence, specify the clock channel using the symbolic constant, `DEFAULTTIMER`, which always maps to the default Logical Channel for analog input timing.

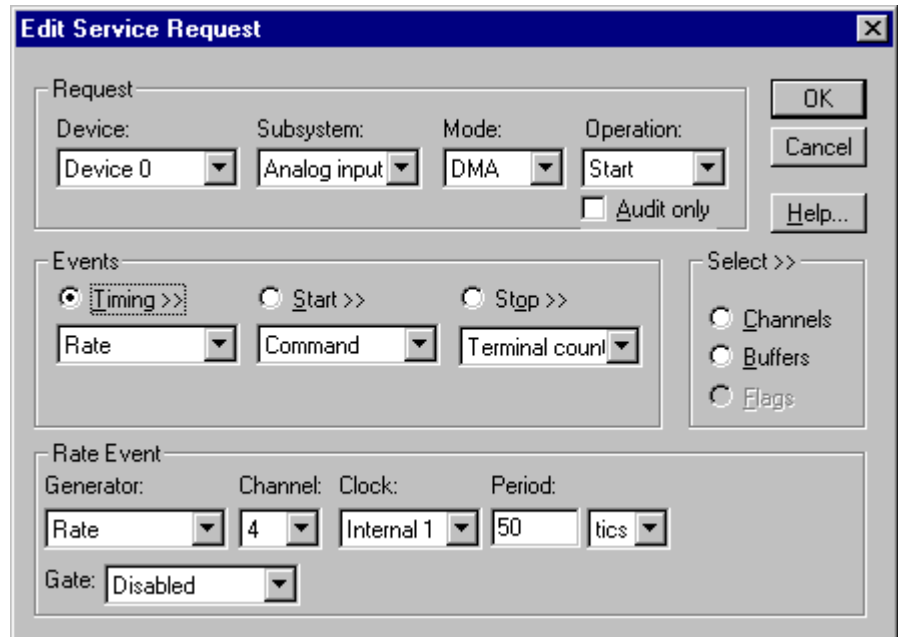
- Specify internal clocking using a **Rate Generator** on **Channel 4** with the **Internal1** Clock source.
- The *Period* property specifies the time interval between samples in tics.
- The KPCI-3100 Series does not support gating the analog input pacing clock. Therefore, only Disabled and NoConnect settings are valid for the *Gate* property.

Rate Generator: External Clocking

An externally clocked Rate Generator produces a rate clock with unknown time intervals between tics.



Use an externally clocked rate generator when you want to synchronize analog input samples with a recurrent external signal. In this mode, you need a separate external clock tic for each analog sample you want to acquire.



How to set up the KPCI-3100 Series for fixed rate sampling using an external clock.

For hardware independence, specify the clock channel using the symbolic constant, DEFAULTTIMER, which always maps to the default Logical Channel for analog input timing.

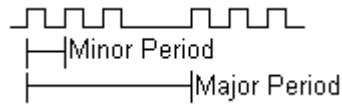
Be sure that the external clock source is TTL compatible, 0 V minimum to +5 V maximum!

- Specify external clocking using a **Rate Generator** on **Channel 4** with an **External** or **External- Clock** source.
- Specify a *Period* between the minimum and maximum external clocking period. The value does not affect the external clock frequency, but DriverLINX requires a valid hardware value in case the application requests a timebase operation and to optimize data transfer between the driver and the application.
- The KPCI-3100 Series does not support gating the analog input pacing clock. Therefore, only Disabled and NoConnect settings are valid for the *Gate* property.
- Users should connect the external clock signal to the External A/D Clock Input line.
- The frequency of the external clock must not exceed the maximum sampling rate for the board.

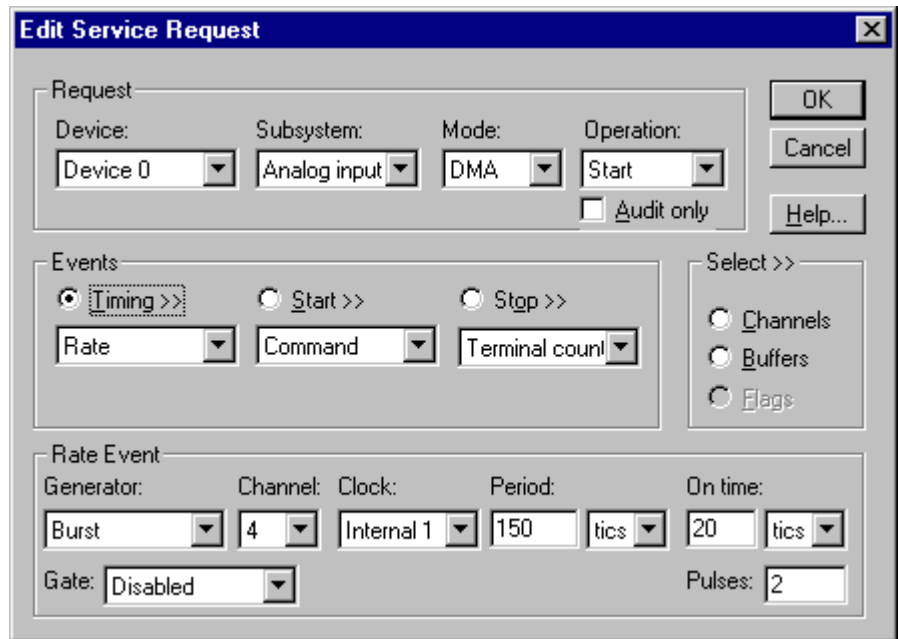
Note: The hardware transfers samples in groups of 16 on models KPCI-3101/02/03/04 and 32 on models KPCI-3110/16. Therefore, samples may be held on the board for up to 31 external clock pulses after they are taken. When acquiring a fixed number of samples, ensure the external clock continues until all desired samples are transferred from the board.

Burst Generator: Internal Clocking

An internally clocked Burst Generator produces a dual frequency clock with a fixed number of tics at a high frequency repeated at a lower frequency.



Use an internally clocked rate generator when you want to acquire analog input samples from a several channels at closely spaced time intervals and then repeat at longer intervals.



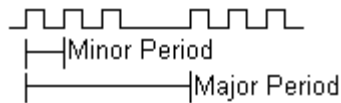
How to set up the KPCI-3100 Series for burst mode sampling using an internal clock.

For hardware independence, specify the clock channel using the symbolic constant, `DEFAULTTIMER`, which always maps to the default Logical Channel for analog input timing.

- Specify internal clocking using a **Burst Generator** on **Channel 4** with the **Internal1** Clock source.
- The *Period* property specifies the time interval between bursts in tics.
- The *On time* property specifies the time interval between samples. *Period* must be greater than or equal to $On\ time \times Pulses + 40\ tics$ (2 μs).
- The KPCI-3100 Series does not support gating the analog input pacing clock. Therefore, only Disabled and NoConnect settings are valid for the *Gate* property.
- The *Pulses* property specifies how many channels the board samples in each *Period*. *Pulses* must equal the number of channels in the channel list multiplied by the desired number of scans per *Period*. The KPCI-3100 Series can sample up to 256 scans per *Period*.

Burst Generator: External Clocking

An externally clocked Rate Generator produces a dual frequency clock with a fixed number of tics at a high, internal frequency repeated at a lower, externally controlled frequency.



Use an externally clocked burst generator when you want to synchronize a burst of analog input samples with a recurrent external signal. In this mode, you need a separate external clock tic for each burst of analog samples you want to acquire.

How to set up the KPCI-3100 Series for burst mode sampling using an external clock.

For hardware independence, specify the clock channel using the symbolic constant, `DEFAULTTIMER`, which always maps to the default Logical Channel for analog input timing.

- Specify external clocking using a **Burst Generator** on **Channel 4** with an **External, External+** or **External-** Clock source.
- Specify a *Period* between the minimum and maximum external clocking period. The value does not affect the external clock frequency, but DriverLINX requires a valid hardware value in case the application requests a timebase operation and to optimize data transfer between the driver and the application.
- The KPCI-3100 Series does not support gating the analog input pacing clock. Therefore, only Disabled and NoConnect settings are valid for the *Gate* property.
- The *On time* property specifies the time interval between samples. *Period* must be greater than or equal to $On\ time \times Pulses + 40\ tics$ (2 μs).
- The *Pulses* property specifies how many channels the board samples in each *Period*. *Pulses* must equal the number of channels in the channel list multiplied by the desired number of scans per *Period*. The KPCI-3100 Series can sample up to 256 scans per *Period*.

Be sure that the external clock source is TTL compatible, 0 V minimum to +5 V maximum!

- Users should connect the external clock signal to the External A/D TTL Trigger line.
- The frequency of the external clock must not exceed the maximum sampling rate for the board.

Note: The hardware transfers samples in groups of 16 on models KPCI-3101/02/03/04 and 32 on models KPCI-3110/16. Therefore, samples may be held on the board for up to 31 external clock pulses after they are taken. When acquiring a fixed number of samples, ensure the external clock continues until all desired samples are transferred from the board.

Analog Input Start Events

Start Events specify when the KPCI-3100 hardware starts acquiring analog input data.

The KPCI-3100 Series supports the following Start Events:

- **None**—Use this event when the DriverLINX operation does not require a Start Event.
- **Command**—DriverLINX starts the task on software command, i.e., as soon as DriverLINX finishes programming the KPCI-3100 hardware for the task.
- **Analog**—The hardware starts acquiring analog input samples when the hardware detects an analog signal satisfies the condition specified in the Start Event.
- **Digital**—The hardware starts acquiring analog input samples when the hardware detects the digital Logical Channel input satisfies the condition specified in the Start Event.

None or Null Start Event

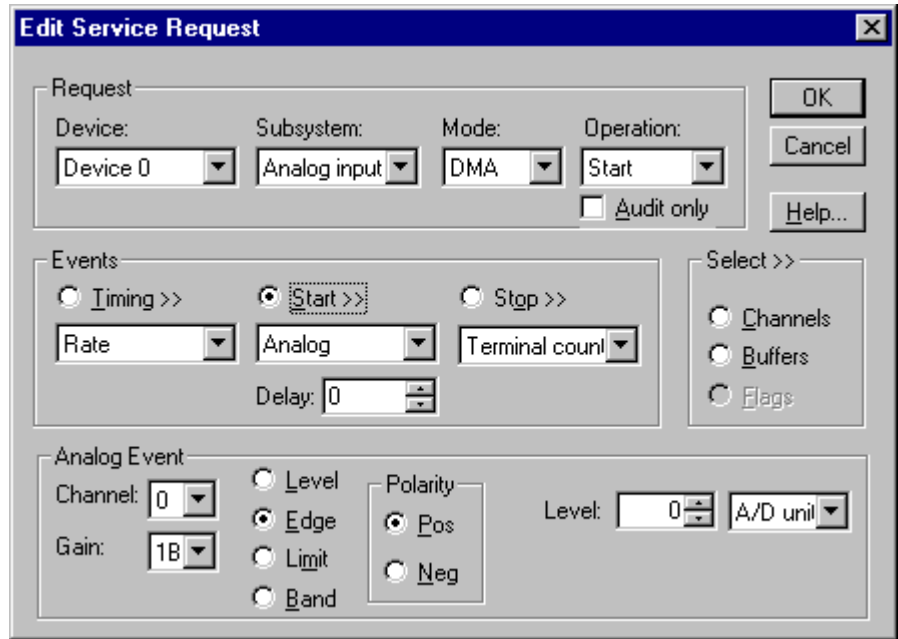
The Null Event specifies that the task does not need a Start Event to begin the task.

Command Start Event

The Command Event starts data acquisition as soon as DriverLINX has completed programming the data-acquisition hardware with the task parameters.

Analog Start Event (Post-Triggering)

The KPCI-3100 Series can acquire analog input samples *after* the hardware detects an analog trigger condition. Use post-triggering when you want to synchronize the start of data acquisition with an external signal.



How to set up the KPCI-3100 Series for post-triggered analog input.

Analog Start Events contain *Channel*, *Gain*, *Polarity* and *Limit* fields. The limits determine the type of analog event (Level, Edge, Limit, or Band). The KPCI-3100 Series supports only edge triggering. A rising trigger occurs when a sample above the limits follows a sample below the limits. Similarly, a falling trigger occurs when a sample below the limits follows a sample above the limits. As the board acquires samples, it compares data from the specified *Channel* against the identical *High* and *Low Limits*. The *Polarity* determines whether the trigger occurs on a rising or falling signal.

- Specify the *Channel* and *Gain* for a channel in the analog input subsystem. For the KPCI-3100 Series, the analog event can use either the Analog Trigger input or the first channel being sampled.

To trigger on the Analog Trigger input, specify **0** as the *Channel* and any gain plus the OEM Gain Flag as the *Gain*. (The Edit Service Request Dialog does not provide this option but it is available programmatically.) DriverLINX supports a programmable gain as a convenience. The KPCI-3100 Series actually supports only a fixed range of $\pm 10V$ with an 8-bit resolution but you can still use *Gain2Code* and *Volts2Code* to calculate the threshold settings.

To trigger on the first channel being sampled, specify the *Channel* and *Gain* as in the first entry in the channel list.

- Specify the *Polarity* (or Slope) property as **Pos** for a rising trigger or **Neg** for a falling trigger.
- Specify the *Limit* properties in hardware A/D codes as follows:

Type	High Limit	Low Limit
Edge	Threshold	Threshold

Programming Tip: Use *Volts2Code* to easily convert volts to hardware A/D codes for the threshold properties.

- Specify the *Delay* property as **0** or a positive number of samples to skip after the trigger. The *Delay* value must be multiple of the number of channels in the channel list.

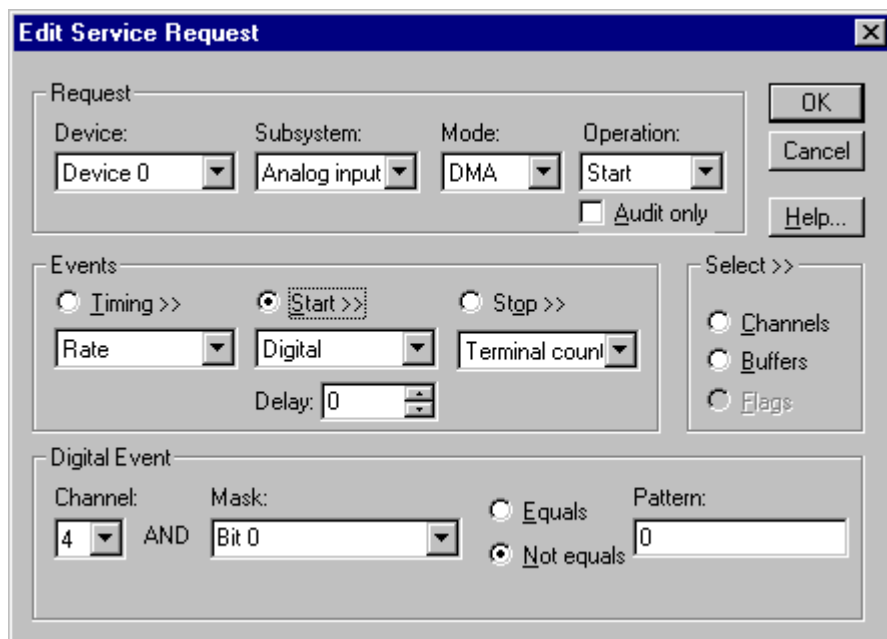
Note:

- When the hardware detects the trigger, DriverLINX sends a StartEvent message (or event). For more information on messages, see “Analog Input Messages” on page 58.

Example: For a programming example, see AIATRГ in your DrvLINX4\Source folder.

Digital Start Event (Post-Triggering)

The KPCI-3100 can acquire analog input samples *after* the hardware detects a digital trigger condition. Use post-triggering when you want to synchronize the start of data acquisition with an external signal.



How to set up the KPCI-3100 Series for post-triggered analog input.

Digital Start Events contain *mask*, *pattern*, and *match* fields. The mask is logically ANDed with the digital input data on the Logical Channel and then compared with the *pattern* for a match/mismatch.

- Specify the *Channel* as **3** for models KPCI-3110/16 or **4** for models KPCI-3101/02/03/04. For hardware-independence, you can specify the hardware external trigger channel by the symbolic constant, *DI_EXTRG*.
- Users should connect the external trigger signal to the External A/D Trigger line.

- Specify the *Mask* property as **1**, or **Bit 0**, to indicate that DriverLINX should only compare a 1-bit digital input value against the *Pattern* property.
- Specify the *Match* property as **Not equals**.
- Specify the *Pattern* property as **0** for a rising, or positive, edge trigger ($\neq 0$) or **1** for a falling, or negative, edge trigger ($\neq 1$).
- Specify the *Delay* property as **0** or a positive number of samples to skip after the trigger. The *Delay* value must be a multiple of the number of channels in the channel list.

Note:

- When the hardware detects the trigger, DriverLINX sends a StartEvent message (or event). For more information on messages, see “Analog Input Messages” on page 58.

Example: For a programming example, see AIDIGTRG in your DrvLINX4\Source folder.

Analog Input Stop Events

Stop Events specify when the hardware stops acquiring analog input data.

The KPCI-3100 Series supports the following Stop Events:

- **None**—Use this event when the DriverLINX operation does not require a Stop Event.
- **Command**—DriverLINX stops the task on software command, i.e., when the application issues a Service Request with a *Stop* operation.
- **Terminal count**—DriverLINX stops the task after the data-acquisition hardware has filled all the data buffers once.
- **Analog**—The hardware stops acquiring analog input samples when the hardware detects an analog signal satisfies the condition specified in the Start Event.
- **Digital**—The hardware stops acquiring analog input samples when the hardware detects the digital Logical Channel input satisfies the condition specified in the Stop Event.

None or Null Stop Event

The Null Event specifies that the task does not need a Stop Event to end the task.

Command Stop Event

The Command Event stops data acquisition when the user application changes the *Operation* property in the Service Request to *Stop* and resubmits the Service Request to DriverLINX.

In Stop-on-Command mode, DriverLINX continuously cycles through all the data buffers filling them with analog input data from the data-acquisition hardware.

Terminal Count Stop Event

The Terminal Count Event stops data acquisition after DriverLINX has filled all the data buffers *once* with analog input data. Use Terminal Count when you want to acquire a single scan or fixed amount of data.

Analog Stop Event (Pre- and About-Triggering)

The KPCI-3100 can acquire analog input samples *until* the hardware detects an analog trigger condition. Use pre-triggering when you want to synchronize the end of data-acquisition with an external signal. Use about-triggering when you want to delay the end of data-acquisition a fixed number of samples after the trigger.

The screenshot shows the 'Edit Service Request' dialog box with the following settings:

- Request:** Device: Device 0, Subsystem: Analog input, Mode: DMA, Operation: Start. Audit only.
- Events:** Timing >>, Start >>, Stop >>. Rate: Rate, Command: Command, Analog: Analog, Delay: 0. **Select >>**: Channels, Buffers, Flags.
- Analog Event:** Channel: 0, Gain: 1B. Level, Edge, Limit, Band. Polarity: Pos, Neg. High limit: 1.000 volts, Low limit: 0.000 volts.

How to set up the KPCI-3100 Series for pre-triggered analog input.

Analog Stop Events contain *Channel*, *Gain*, *Polarity* and *Limit* fields. The limits determine the type of analog event (Level, Edge, Limit, or Band). The KPCI-3100 Series supports only edge triggering. A rising trigger occurs when a sample above the limits follows a sample below the limits. Similarly, a falling trigger occurs when a sample below the limits follows a sample above the limits. As the board acquires samples, it compares data from the specified *Channel* against the identical *High* and *Low Limits*. The *Polarity* determines whether the trigger occurs on a rising or falling signal.

- Specify the *Channel* and *Gain* for a channel in the analog input subsystem. For the KPCI-3100 Series, the analog event can use either the Analog Trigger input or the first channel being sampled.

To trigger on the Analog Trigger input, specify **0** as the *Channel* and any gain plus the OEM Gain Flag as the *Gain*. (The Edit Service Request Dialog does not provide this option but it is available programmatically.) DriverLINX supports a programmable gain as a convenience. The KPCI-3100 Series actually supports only a fixed range of $\pm 10V$ with an 8-bit resolution but you can still use *Gain2Code* and *Volts2Code* to calculate the threshold settings.

To trigger on the channels being sampled, specify the *Channel* and *Gain* as in the first entry in the channel list. **Note:** When sampling from more than one channel, you would typically use the Analog Trigger input. This is because the hardware trigger monitors either the Analog Trigger channel or the output of the ADC, which cannot compare samples from the same channel while sampling from multiple channels.

- Specify the *Polarity* (or Slope) property as **Pos** for a rising trigger or **Neg** for a falling trigger.
- Specify the *Limit* properties in hardware A/D codes as follows:

Type	High Limit	Low Limit
Edge	Threshold	Threshold

Programming Tip: Use *Volts2Code* to easily convert volts to hardware A/D codes for the threshold properties.

- Specify the *Delay* property as **0** for pre-triggering, or a positive number of samples to obtain after the trigger for about-triggering. The *Delay* value must be a multiple of the number of the number of channels in the channel list.

The KPCI-3100 Series supports stopping immediately upon detection of the trigger, rather than at the end of the current scan. To use this feature specified the *Delay* property as **1**.

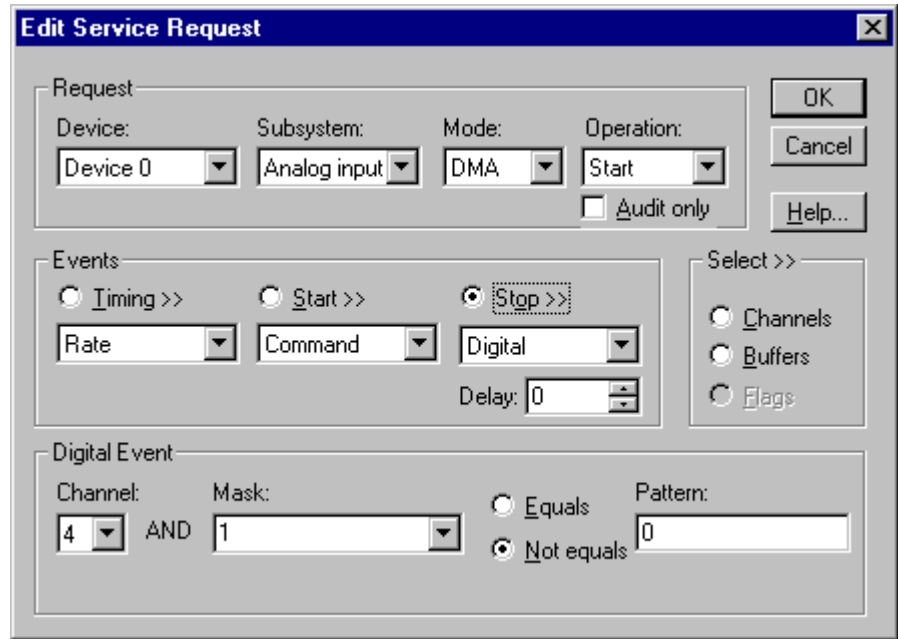
Note:

- When the hardware detects the trigger, DriverLINX sends a StopEvent message (or event) that identifies the buffer location of the last sample. For more information on messages, see “Analog Input Messages” on page 58.

Example: For a programming example, see AISTPTRG in your DrvLINX4\Source folder.

Digital Stop Event (Pre- and About-Triggering)

The KPCI-3100 can acquire analog input samples *until* the hardware detects a digital trigger condition. Use pre-triggering when you want to synchronize the end of data acquisition with an external signal.



How to set up the KPCI-3100 Series for pre-triggered analog input.

Digital Stop Events contain *mask*, *pattern*, and *match* fields. The mask is logically ANDed with the digital input data on the Logical Channel and then compared with the *pattern* for a match/mismatch.

- Specify the *Channel* as **3** for models KPCI-3110/16 or **4** for models KPCI-3101/02/03/04. For hardware-independence, you can specify the hardware external trigger channel by the symbolic constant, *DI_EXTRG*.
- Users should connect the external trigger signal to the External A/D TTL Trigger line.
- Specify the *Mask* property as **1**, or **Bit 0**, to indicate that DriverLINX should only compare a 1-bit digital input value against the *Pattern* property.
- Specify the *Match* property as **Not equals**.
- Specify the *Pattern* property as **0** for a rising, or positive, edge trigger ($\neq 0$) or **1** for a falling, or negative, edge trigger ($\neq 1$).
- Specify the *Delay* property as **0** for pre-triggering, or a positive number of samples to obtain after the trigger for about-triggering. The *Delay* value must be a multiple of the number of channels in the channel list.

The KPCI-3100 Series supports stopping immediately upon detection of the trigger, rather than at the end of the current scan. To use this feature specified the *Delay* property as **1**.

Note:

- When the hardware detects the trigger, DriverLINX sends a StopEvent message (or event) that identifies sample following the trigger. For more information, see “Analog Input Messages” on page 58.

Example: For a programming example, see AIDSTPTG in your DrvLINX4\Source folder.

Analog Input Channels

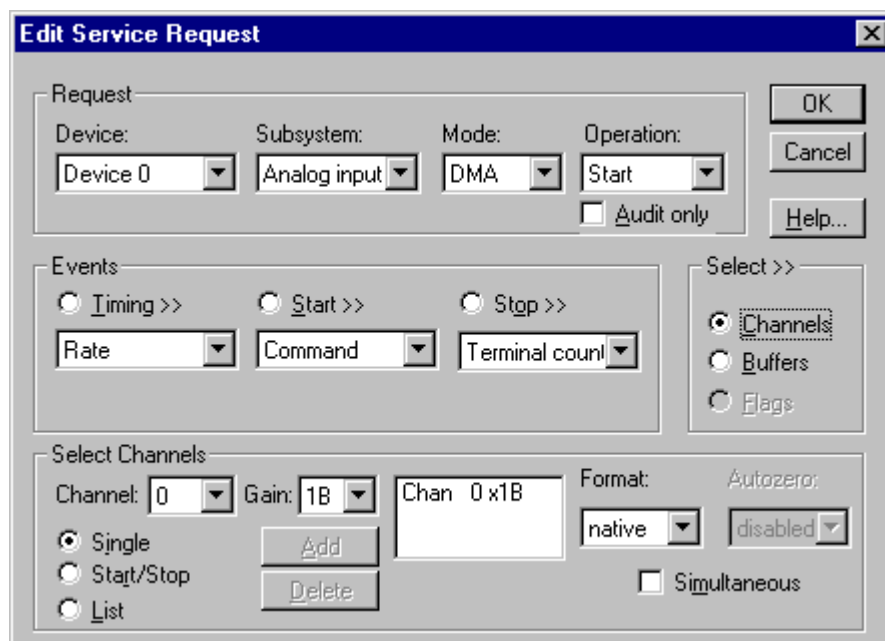
The KPCI-3100 Series models support a variety of channel gains with differential or single-ended termination modes. The Logical Device configuration sets the default termination mode for analog input channels. An application can request a particular termination mode for each channel it uses. The channel gains are also application selectable.

The KPCI-3100 Series allows applications to specify the analog channels using three techniques:

- **Start Channel**—Acquire data from a single channel.
- **Start/Stop Channel Range**—Acquire data from a consecutive range of channels.
- **Channel List**—Acquire data from a list of channels.

Single Channel Analog Input

In single channel mode, the KPCI-3100 Series acquires all data from one channel at the specified gain.



How to set up the KPCI-3100 Series for sampling on a single channel.

Multi-channel Analog Input Range

In multi-channel range mode, the KPCI-3100 Series acquires data from a consecutive range of analog channels.

- The Start Channel's gain only applies to the first channel.
- DriverLINX uses the Stop Channel's gain for all the other analog channels in the range.
- The gains may vary but they must all be either unipolar or bipolar.
- All base channels must have the same termination mode.
- If the Start Channel is greater than the Stop Channel, the channel sequence is [Start Channel,... Last Channel, 0,... Stop Channel], where Last Channel is the highest numbered channel.

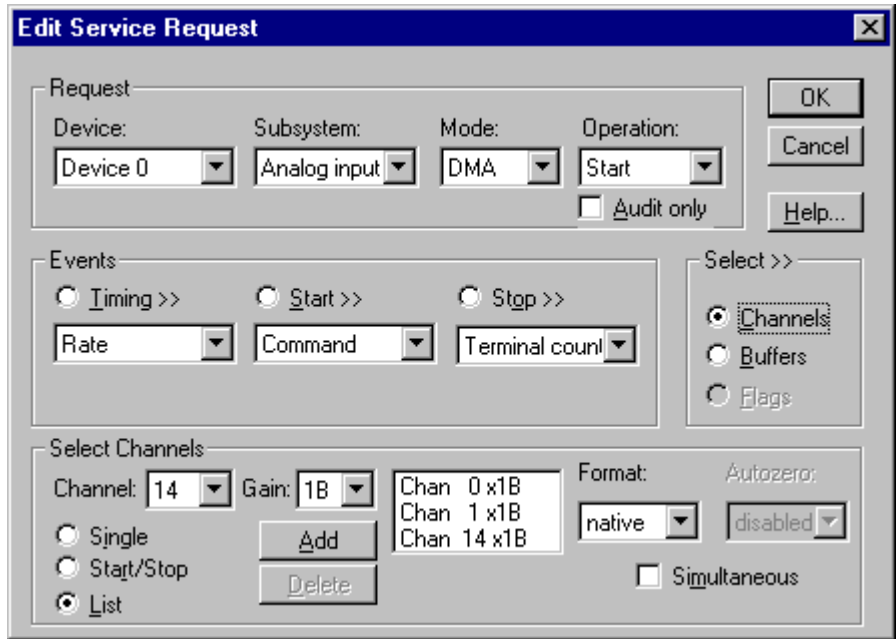
The screenshot shows the 'Edit Service Request' dialog box. It is divided into several sections: 'Request' with dropdowns for Device (Device 0), Subsystem (Analog input), Mode (DMA), and Operation (Start), and an 'Audit only' checkbox; 'Events' with radio buttons for Timing, Start, and Stop, and dropdowns for Rate, Command, and Terminal count; 'Select Channels' with a Channel dropdown (15), Gain dropdown (1B), a list of channels (Chan 0 x1B, Chan 15 x1B), Format (native), and Autozero (disabled) dropdowns, and a Simultaneous checkbox. There are also buttons for Add and Delete, and radio buttons for Single, Start/Stop, and List.

How to set up the KPCI-3100 Series for sampling on a consecutive range of channels.

Multi-channel Analog Input List

In multi-channel list mode, the KPCI-3100 Series acquires data from an arbitrary list of analog channels.

- The channel-gain list may contain up to 1024 channels in any order. The list may even include digital input channels; see “Reading Digital Input Channels with the Analog Input Subsystem” on page 53 for more information on this feature.
- The list may repeat the same channel with the same or different gains.
- The gains may vary but they must all be either unipolar or bipolar.
- All base channels must have the same termination mode.



How to set up the KPCI-3100 Series to sample from an arbitrary list of channels.

Analog Input Channel Gains

Each channel in a channel scan list has a gain code property to select the pre-amplifier gain when sampling that channel. The following tables show the correspondence between the gain multiplier, the maximum input signal range, and the gain code for each input range. Note: DriverLINX uses a negative (-) gain multiplier to signify a bipolar (\pm) range.

Gain	Range	Gain Code
-1	± 10 V	0
-2	± 5 V	1
-4	± 2.5 V	2
-8	± 1.25 V	3
1	0 - 10 V	4
2	0 - 5 V	5
4	0 - 2.5 V	6
8	0 - 1.25 V	7

Gains, Ranges, and Hardware Gain Codes for KPCI-3100 Series.

Use the DriverLINX **Gain2Code** method to easily convert between the gains in the above tables and Gain Codes. Using this method makes applications portable to different hardware models that have the same analog input ranges.

The KPCI-3100 Series can output digital data with each analog channel in the channel list. DriverLINX uses the gain code to store this digital data. See “Writing Digital Output Channels with the Analog Input Subsystem” on page 54 for more information.

Analog Input Channel Termination Modes

On the KPCI-3100 Series, Analog Input channels can use single-ended or differential termination modes. When configuring the Analog Input Subsystem, you choose a default configuration for all channels. Applications can use the default configuration or specify the termination mode for each task. This scheme supports applications that use KPCI-3100-specific features as well as those that use only generic features.

Note: Choosing *differential* as the default configuration limits the number of channels available. This number cannot be overridden. See "Analog Input Subsystem Page" on page 15.

Each channel in a channel list has a gain code property. To specify a termination mode for a channel, an application includes a termination mode flag in its gain code. On the KPCI-3100 Series, a single task cannot sample base channels in different termination modes.

The following table shows the flag value for each termination mode:

Termination mode	Flag Value
Default Configuration	CHAN_SEDIFF_DEFAULT = 0
Single-ended	CHAN_SEDIFF_SE = 2×2^{13}
Differential	CHAN_SEDIFF_DIFF = 3×2^{13}
Digital	CHAN_DIGITAL = 1×2^{13}

“Digital” signifies a digital channel rather than an analog channel termination mode. See “Reading Digital Input Channels with the Analog Input Subsystem” on page 53.

Note: The user chooses the default configuration on the Analog Input page of the DriverLINX Configuration Panel. See “Analog Input Subsystem Page” on page 15.

For example, an application that requires or knows a channel’s termination mode obtains the gain code for a single-ended channel with a bipolar gain of 4, with:

```
Gain2Code (-4) + CHAN_SEDIFF_SE
```

```
// This code will work with only drivers that allow applications to specify a termination mode.
```

An application that does not require or know a channel’s termination mode obtains the gain code for a channel with a bipolar gain of 4, with:

```
Gain2Code (-4)
```

```
// This code will work with any board that supports bipolar ranges.
```

Reading Digital Input Channels with the Analog Input Subsystem

The KPCI-3100 Series can sample digital input channels 0 and 1 along with analog input channels. To use this feature, simply add channel 0 to the channel-gain list with the gain code equal to the special value, CHAN_SEDIFF_DIG. Instead of returning an analog channel’s A/D code in the data buffer, the board returns all sixteen bits from digital input channels 0 and 1. Note that the buffer will have analog data interspersed with digital data.

Note: If you are sampling only digital input channels, you can achieve the same result by sampling with the Digital Input Subsystem. See “Digital Input Subsystem” for details.

Writing Digital Output Channels with the Analog Input Subsystem

The KPCI-3100 Series can also write digital output data while sampling each analog input channel. Models KPCI-3101/02/03/04 support seven bits of dynamic digital output. They share pins with Digital Input/Output Bank C. Models KPCI-3110/16 support only two bits of dynamic digital output.

To use this feature, merge the digital output data into the analog input channel list by shifting it left 4 bits (multiply by 16) and OR-ing it with the gain field. Then OR the symbolic constant `CHAN_OEM_FLAG` with the gain field.

CHAN_OEM_FLAG is a special value for the gain field that indicates the channel/gain entry has a hardware-specific meaning. The meaning is not portable between drivers. Its value is 2^{15} . Use it by OR-ing it with other gain settings as described for the driver.

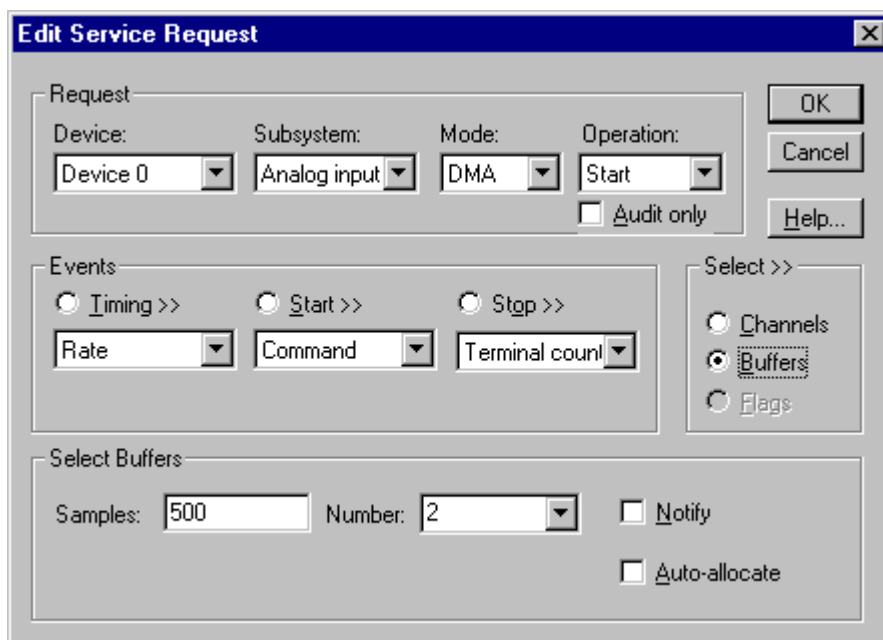
Notes:

- If you set the `CHAN_OEM_FLAG` for any channel/gain list entry, it is effectively set of all. Entries without the flag result in outputting zero while the board is sampling the corresponding channel.
- On models KPCI-3101/02/03/04, you cannot write to Logical Channel 2 using the Digital Output subsystem while writing dynamic digital output.

Analog Input Buffers

DriverLINX supports single-value, single-scan and buffered analog input.

- **For single-value input**, specify the *Number* of buffers as **0**. The buffer for a single value is the *ioValue* property.
- **For single-scan input**, specify the *Number* of buffers as **1** and the number of *Samples* equal to the number of channels.
- **For buffered input**, specify the *Number* of buffers from **1** to **255** and the number of *Samples* as desired.



How to set up the KPCI-3100 Series to store samples in buffers.

Buffer Size

For example, 500 samples/2 channels = 250 is ok, but 500 samples/3 channels = 166.67 is incorrect.

An individual DriverLINX buffer may have any size as long as the buffer length holds an integral number of channel scans (i.e., a multiple of the number of analog input channels you're acquiring). This restriction enforces the requirement that all acquired channels have the same number of samples.

Buffer Usage

DriverLINX fills buffers sequentially until the task stops. During the task, only complete buffers are available to the application. Except for tasks that stop on terminal count, the last buffer may be only partially full. If the task stops on a trigger, use the StopEvent message (or event) to determine the location of the last sample. For other cases, use a Status operation to determine the location of the last sample.

Note: The hardware transfers samples in groups of 16 on models KPCI-3101/02/03/04 and 32 on models KPCI-3110/16. Therefore, when stopping on command, the last buffer may contain up to 31 invalid samples before the stop position reported by DriverLINX.

Analog Input Data Coding

KPCI-3100 Series models return Analog Input hardware codes in right-justified binary for unipolar ranges and right-justified offset binary for bipolar ranges. DriverLINX refers to these coding schemes as the “native” format.

For any programmable gain, the KPCI-3100 models return hardware codes with the ranges in the following tables:

KPCI-3101/02/03/04/10

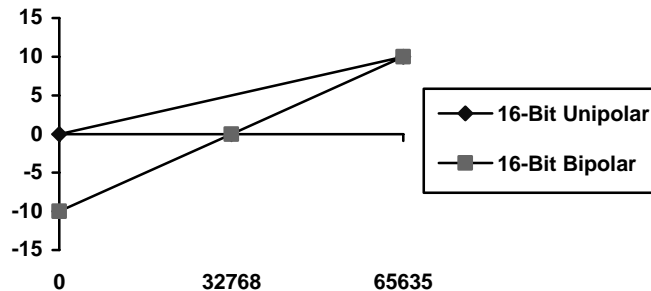
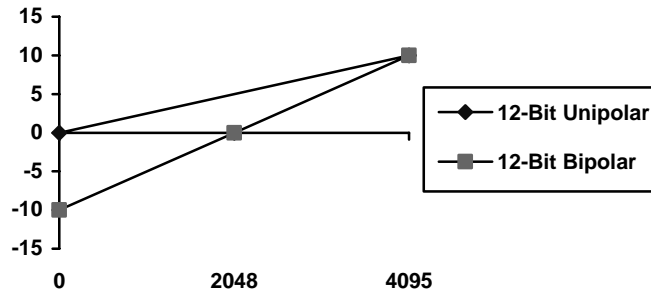
Polarity	Analog Input Resolution	Analog Input Hardware Code
Unipolar	12 bits	0 to 4095
Bipolar	12 bits	0 to 4095

Native Analog Input hardware codes for each KPCI-3100 Series polarity.

KPCI-3116

Polarity	Analog Input Resolution	Analog Input Hardware Code
Unipolar	16 bits	0 to 65535
Bipolar	16 bits	0 to 65535

Native Analog Input hardware codes for each KPCI-3100 Series polarity.



KPCI-3100 Series native Analog Input Codes versus Voltage Range for a ten-volt range.

DriverLINX refers to the default hardware analog-coding scheme as the “native” format. For computer arithmetic in a high-level language, the 16-bit two’s complement integer format is generally easier to use. Applications can use

DriverLINX's data conversion operations to transform an entire data buffer from native format to many common integer and floating-point formats.

Analog Input Messages

For analog input operations, DriverLINX can report the following messages to the application:

DriverLINX Message	Explanation
Service Start	DriverLINX has started the acquisition task.
Service Done	DriverLINX has completed the acquisition task.
Start Event	DriverLINX has detected the hardware start event.
Stop Event	DriverLINX has detected the hardware stop event.
Buffer Filled	DriverLINX has filled an analog input buffer.
Data Lost	DriverLINX has detected an analog input data overrun condition.
Critical Error	DriverLINX has encountered an unexpected hardware or software condition.

DriverLINX Event messages for analog input.

For detailed explanations of these messages see one of the following references:

- *DriverLINX Technical Reference Manual* for C/C++ users
- *DriverLINX/VB Technical Reference Manual* for VB or Delphi users

Analog Output Subsystem

The following sections describe how DriverLINX implements Analog Output Subsystem features for the KPCI-3100 Series.

Analog Output Modes

The Analog Output Subsystem supports the following modes:

- **Polled**—For single-value or single-scan analog output samples.
- **Interrupt**—For buffered transfers using programmed I/O.
- **DMA**—For buffered transfers using direct memory access.
- **Other**—For subsystem initialization and data conversion.

Note: On the KPCI-3100 Series, the driver performs all analog output data transfers in the PCI bus's slave mode. An application can specify DMA or Interrupt mode for compatibility with other boards.

Analog Output Operations

The KPCI-3100 Series Analog Output Subsystem supports the following DriverLINX operations:

- **Initialize**—aborts all active analog output data-acquisition tasks. However, DriverLINX prevents one application from interfering with another application's data-acquisition tasks.
- **Start**—initiates a data-acquisition task using the Mode, Timing, Start, and Stop Events, the Logical Channels, and the Buffers the application specified in the Service Request.
- **Status**—reports the buffer position of the next sample that DriverLINX will write from a buffer.
- **Stop**—terminates an analog output data-acquisition task.
- **Message**—displays a pop-up dialog box for the user containing the text for the current DriverLINX error message.

Analog Output Timing Events

Timing Events specify how the hardware paces or clocks the writing of analog output samples. DriverLINX uses the Timing Event to program when the KPCI-3100 Series writes the next analog output sample.

The KPCI-3100 Series supports the following Timing Events:

- **None**—Task requires no pacing as DriverLINX is writing only a single value or scan.
- **Rate**—The KPCI-3100 Series supports fixed rate writing using internal and external clocks. Also see: “Synchronizing an Analog Output Task with an Analog Input Task” on page 63.

None or Null Timing Event

The Null Event specifies that the task does not need a clock to determine when to write the next sample.

Rate Timing Event

The KPCI-3100 Series supports the following Rate Event for analog output:

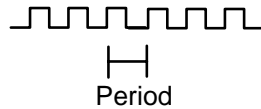
- **Rate Generator**—Generates a fixed rate clock with equal time intervals between tics.



KPCI-3100 Series boards have a fixed 20 MHz master clock frequency. The sample period range depends on the model. All models limit the period to $2^{24} - 1$ tics, or 838,860 μ s (1.20 Hz). Consult your hardware manual for details.

Rate Generator: Internal Clocking

An internally clocked Rate Generator produces a fixed rate clock with equal time intervals between tics.



Use an internally clocked rate generator when you want to write analog output samples at equally spaced time intervals.

The screenshot shows the "Edit Service Request" dialog box. It is divided into three main sections: "Request", "Events", and "Rate Event".
- In the "Request" section, "Device" is set to "Device 0", "Subsystem" is "Analog output", "Mode" is "DMA", and "Operation" is "Start". There is an "Audit only" checkbox which is unchecked.
- In the "Events" section, "Timing >>" is selected with a radio button. Below it are three dropdown menus: "Rate", "Command", and "Terminal count". To the right, there is a "Select >>" section with radio buttons for "Channels", "Buffers", and "Flags".
- In the "Rate Event" section, "Generator" is "Rate", "Channel" is "5", "Clock" is "Internal 1", and "Period" is "500 tics". The "Gate" dropdown is set to "Disabled".

How to set up the KPCI-3100 Series for fixed rate writing using an internal clock.

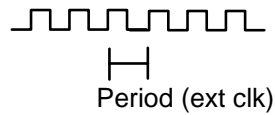
For hardware independence, specify the clock channel using the symbolic constant, `DEFAULTTIMER`, which always maps to the default Logical Channel for analog output timing.

- Specify internal clocking using a **Rate Generator** on **Channel 5** with the **Internal1 Clock** source.
- The *Period* property specifies the time interval between samples in tics.
- The KPCI-3100 Series does not support gating the analog output pacing clock. Therefore, only Disabled and NoConnect settings are valid for the *Gate* property.
- In the *Channel* list, check **Simultaneous** when sampling both analog output channels as the KPCI-3100 Series only supports writing both channels at once.

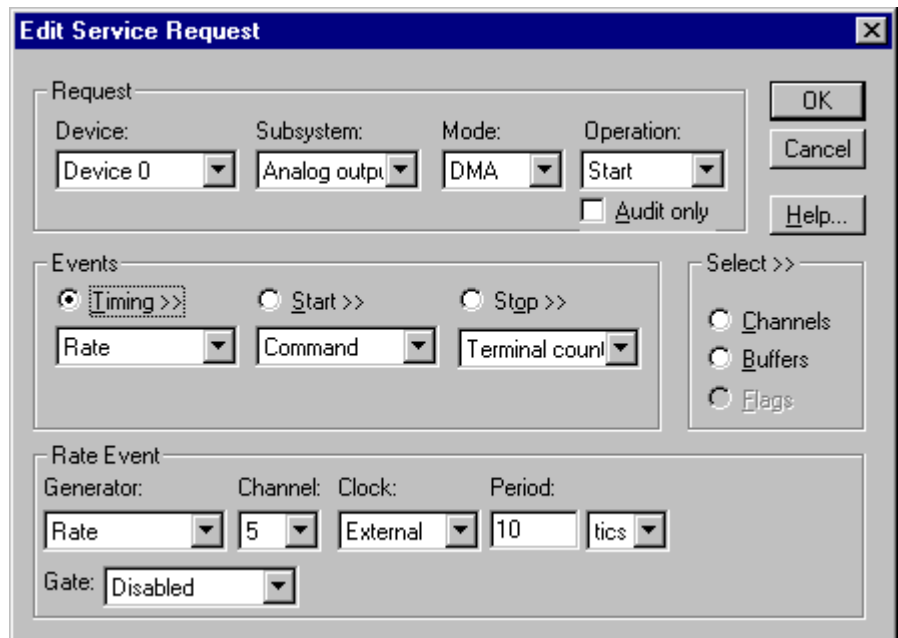
Also see: “Synchronizing an Analog Output Task with an Analog Input Task” on page 63.

Rate Generator: External Clocking

An externally clocked Rate Generator produces a rate clock with unknown time intervals between tics.



Use an externally clocked rate generator when you want to synchronize analog output samples with a recurrent external signal. In this mode, you need a separate external clock tic for each analog scan you want to write.



How to set up the KPCI-3100 Series for fixed rate writing using an external clock.

Be sure that the external clock source is TTL compatible, 0 V minimum to +5 V maximum!

- Specify external clocking using a **Rate Generator** on **Channel 5** with an **External** or **External- Clock** source. See “Counter/Timer Subsystem” on page 98 for a description of clock sources.
- Users should connect the external clock signal to the External D/A TTL Trigger line.
- Specify a *Period* between the minimum and maximum external clocking period. The value does not affect the external clock frequency, but DriverLINX requires a valid hardware value in case the application requests a timebase operation and to optimize data transfer between the driver and the application.
- The frequency of the external clock must not exceed the maximum sample rate for the board.

- The KPCI-3100 Series does not support gating the analog output pacing clock. Therefore, only Disabled and NoConnect settings are valid for the *Gate* property.
- In the *Channel* list, check **Simultaneous** when sampling both analog output channels as the KPCI-3100 Series only supports writing both channels at once.

Synchronizing an Analog Output Task with an Analog Input Task

The KPCI-3100 Series has several features that allow you to synchronize analog output with analog input:

- **Synchronized external clocking**—You can set up separate analog input and analog output tasks that use external clocking and connect the clock signal to both the External A/D Sample Clock Input and the External D/A Sample Clock Input.
- **Synchronized triggering**—You can set up separate analog input and analog output tasks that use an external trigger and connect the trigger signal to both the External A/D TTL Trigger and the External D/A TTL Trigger.

Analog Output Start Events

Start Events specify when the hardware starts acquiring analog output data.

The KPCI-3100 Series supports the following Start Events:

- **None**—Use this event when the DriverLINX operation does not require a Start Event.
- **Command**—DriverLINX starts the task on software command, i.e., as soon as DriverLINX finishes programming the KPCI-3100 hardware for the task.
- **Analog**—The hardware starts writing analog output samples when the hardware detects an analog signal satisfies the condition specified in the Start Event.
- **Digital**—The hardware starts writing analog output samples when the hardware detects the digital Logical Channel input satisfies the condition specified in the Start Event.

None or Null Start Event

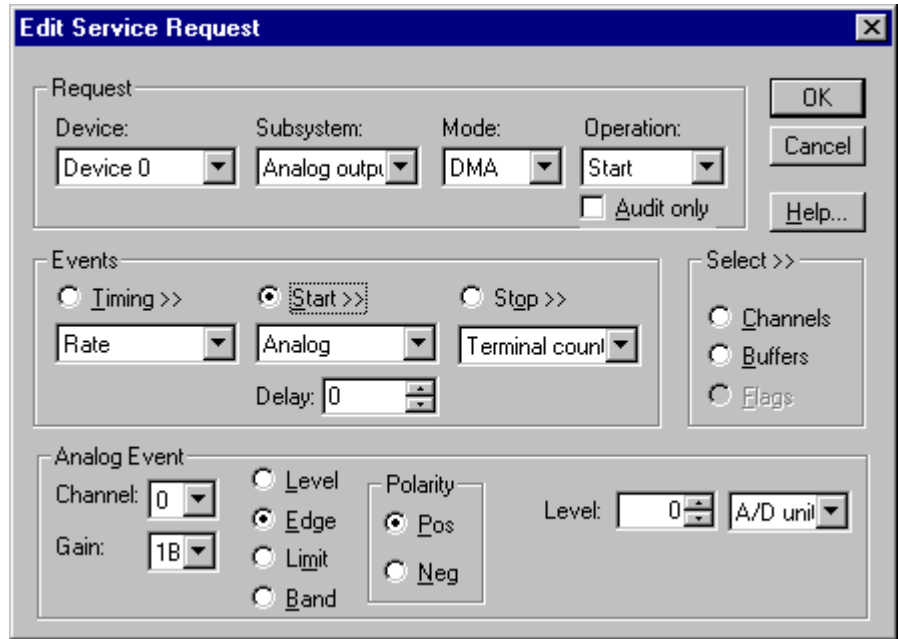
The Null Event specifies that the task does not need a Start Event to begin the task.

Command Start Event

The Command Event starts data acquisition as soon as DriverLINX has completed programming the data-acquisition hardware with the task parameters.

Analog Start Event (Post-Triggering)

The KPCI-3100 Series can write analog output samples *after* the hardware detects an analog trigger condition. Use post-triggering when you want to synchronize the start of data acquisition with an external signal.



How to set up the KPCI-3100 Series for about-triggered analog output.

Analog Start Events contain *Channel*, *Gain*, *Polarity* and *Limit* fields. The limits determine the type of analog event (Level, Edge, Limit, or Band). The KPCI-3100 Series supports only edge triggering. A rising trigger occurs when a sample above the limits follows a sample below the limits. Similarly, a falling trigger occurs when a sample below the limits follows a sample above the limits. As the board acquires samples, it compares data from the specified *Channel* against the identical *High* and *Low Limits*. The *Polarity* determines whether the trigger occurs on a rising or falling signal.

- Specify the *Channel* and *Gain* for a channel in the analog input subsystem. For the KPCI-3100 Series, the analog event can use only the Analog Trigger input.

To trigger on the Analog Trigger input, specify **0** as the *Channel* and any gain plus the OEM Gain Flag as the *Gain*. (The Edit Service Request Dialog does not provide this option but it is available programmatically.) DriverLINX supports a programmable gain as a convenience. The KPCI-3100 Series actually supports only a fixed range of $\pm 10V$ with an 8-bit resolution but you can still use *Gain2Code* and *Volts2Code* to calculate the threshold settings.

- Specify the *Polarity* (or Slope) property as **Pos** for a rising trigger or **Neg** for a falling trigger.
- Specify the *Limit* properties in hardware A/D codes as follows:

Type	High Limit	Low Limit
Edge	Threshold	Threshold

Programming Tip: Use *Volts2Code* to easily convert volts to hardware A/D codes for the threshold properties.

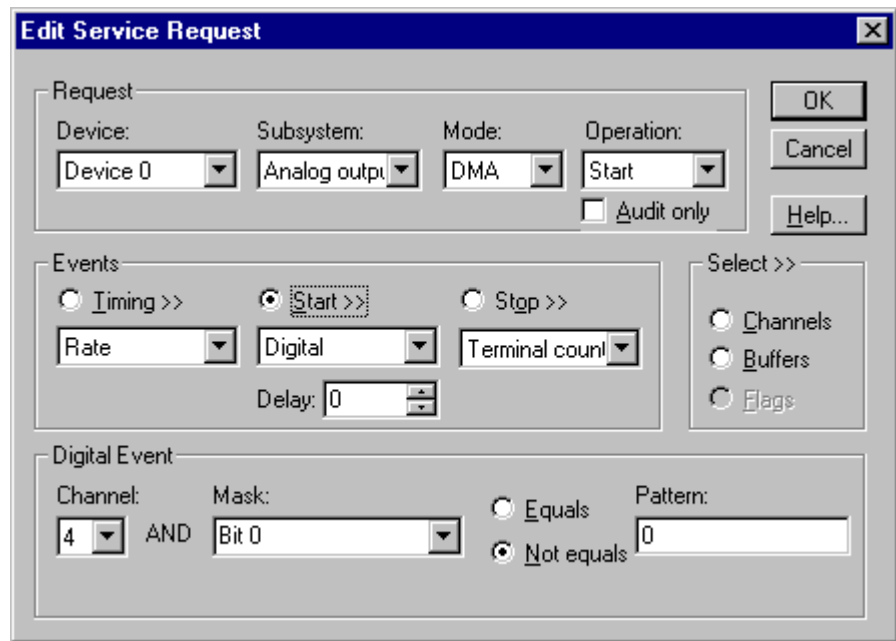
- Specify the *Delay* property as **0**.

Note:

- When the hardware detects the trigger, DriverLINX sends a StartEvent message (or event). For more information on messages, see “Analog Output Messages” on page 72.

Digital Start Event (Post-Triggering)

The KPCI-3100 can write analog output samples *after* the hardware detects a digital trigger condition. Use post-triggering when you want to synchronize the start of data acquisition with an external signal.



How to set up the KPCI-3100 Series for post-triggered analog output.

Digital Start Events contain *mask*, *pattern*, and *match* fields. The mask is logically ANDed with the digital input data on the Logical Channel and then compared with the *pattern* for a match/mismatch.

- Specify the *Channel* as **3** for models KPCI-3110/16 or **4** for models KPCI-3101/02/03/04. For hardware-independence, you can specify the hardware external trigger channel by the symbolic constant, *DI_EXTTRG*.
- Users should connect the external trigger signal to the External D/A Trigger line.
- Specify the *Mask* property as **1**, or **Bit 0**, to indicate that DriverLINX should only compare a 1-bit digital input value against the *Pattern* property.
- Specify the *Match* property as **Not equals**.
- Specify the *Pattern* property as **0** for a rising, or positive, edge trigger ($\neq 0$) or **1** for a falling, or negative, edge trigger ($\neq 1$).

- Specify the *Delay* property as **0**.

Note:

- When the hardware detects the trigger, DriverLINX sends a *StartEvent* message (or event). For more information on messages, see “Analog Output Messages” on page 72.

Example: For a programming example, see AODIGTRG in your DrvLINX4\Source folder.

Analog Output Stop Events

Stop Events specify when the hardware stops acquiring analog output data.

The KPCI-3100 Series supports the following Stop Events:

- **None**—Use this event when the DriverLINX operation does not require a Stop Event.
- **Command**—DriverLINX stops the task on software command, i.e., when the application issues a Service Request with a *Stop* operation.
- **Terminal count**—DriverLINX stops the task after the data-acquisition hardware has filled all the data buffers once.

None or Null Stop Event

The Null Event specifies that the task does not need a Stop Event to end the task.

Command Stop Event

The Command Event stops data acquisition when the user application changes the *Operation* property in the Service Request to *Stop* and resubmits the Service Request to DriverLINX.

In Stop-on-Command mode, DriverLINX continuously cycles through all the data buffers, writing analog output data to the data-acquisition hardware.

Terminal Count Stop Event

The Terminal Count Event stops data acquisition after DriverLINX has filled all the data buffers *once* with analog output data. Use Terminal Count when you want to write a single scan or fixed amount of data.

Analog Output Channels

The KPCI-3100 Series allows applications to specify the analog channels using three techniques:

- **Start Channel**—Write data to a single channel.
- **Start/Stop Channel Range**—Write data to a consecutive range of channels.
- **Channel List**—Write data to a list of channels.

Single Channel Analog Output

In single channel mode, the KPCI-3100 Series writes all data to one channel at the specified gain.

The screenshot shows the 'Edit Service Request' dialog box with the following settings:

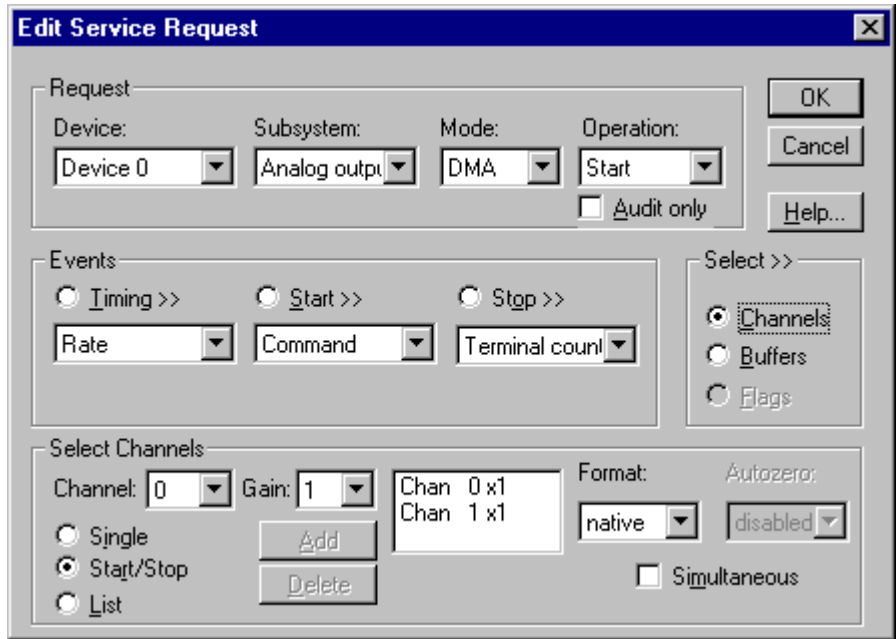
- Request:** Device: Device 0, Subsystem: Analog output, Mode: DMA, Operation: Start, Audit only:
- Events:** Timing >>, Start >>, Stop >>. Rate: [dropdown], Command: [dropdown], Terminal count: [dropdown].
- Select Channels:** Channel: 0, Gain: 1, Chan 0 x1. Single, Start/Stop, List. Add, Delete buttons. Format: native, Autozero: disabled. Simultaneous.
- Select >>:** Channels, Buffers, Flags.

How to set up the KPCI-3100 Series for writing to a single channel.

Multi-channel Analog Output Range

In multi-channel range mode, the KPCI-3100 Series writes data to a consecutive range of analog channels.

- For the KPCI-3100 Series, when writing to both analog outputs, the Start Channel must be channel 0 and the Stop Channel must be channel 1.

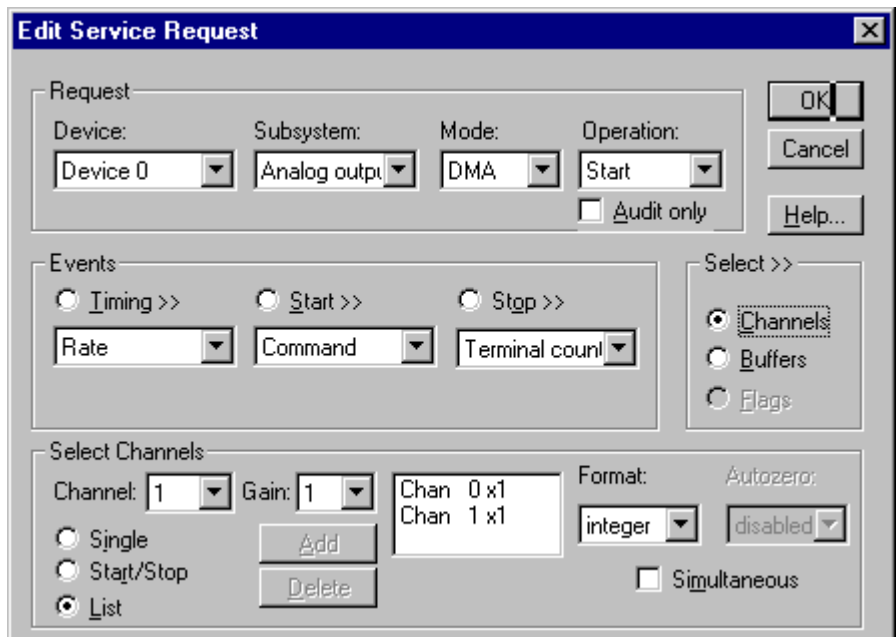


How to set up the KPCI-3100 Series for writing to a consecutive range of channels.

Multi-channel Analog Output List

In multi-channel list mode, the KPCI-3100 Series writes data to an arbitrary list of analog channels.

- For the KPCI-3100 Series, you can sample from either channel or from channel 0 followed by channel 1.



How to set up the KPCI-3100 Series to write to an arbitrary list of channels.

Analog Output Channel Gains

The KPCI-3100 Series models support a single channel gains for each digital-to-analog converter. The following table shows the correspondence between DriverLINX gains, the maximum output signal range, and the gain code for each output range. Note: DriverLINX uses a negative (-) gain value to signify a bipolar (\pm) range.

KPCI-3102/04/10/16 Models

Gain	Range	Gain Code
-1	± 10 V	0

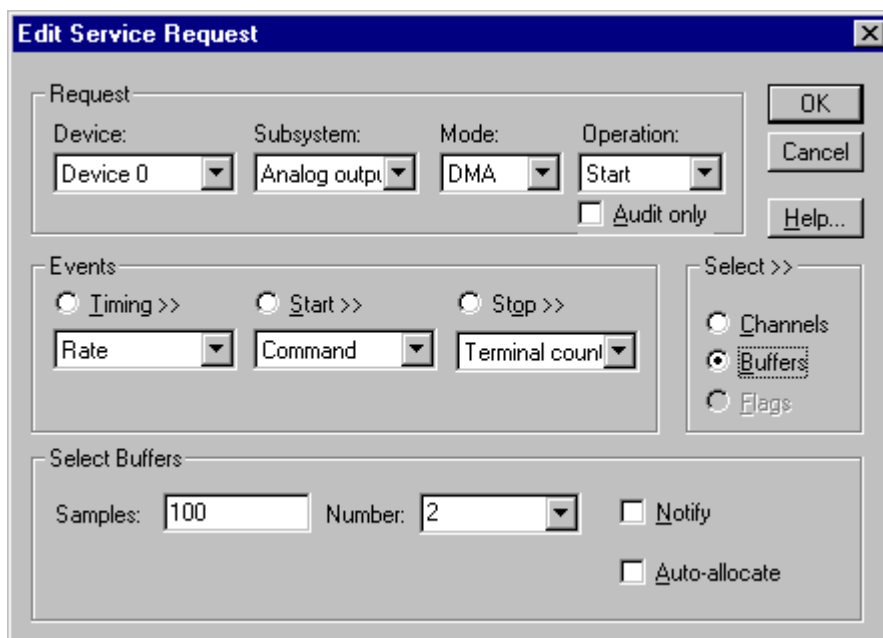
Gains, Ranges, and Gain Codes for the KPCI-3100 Series.

Use the DriverLINX **Gain2Code** method to easily convert between the gains in the above tables and hardware Gain Codes.

Analog Output Buffers

DriverLINX supports single-value, single-scan and buffered analog output.

- **For single-value output**, specify the *Number* of buffers as **0**. The buffer for a single value is the *ioValue* property.
- **For a single-scan output**, specify the *Number* of buffers as **1** and the number of *Samples* equal to the number of channels.
- **For buffered output**, specify the *Number* of buffers from **1** to **255** and the number of *Samples* as desired.



How to set up the KPCI-3100 Series to store samples in buffers.

Buffer Size

For example, 500 samples/2 channels = 250 is ok, but 500 samples/3 channels = 166.67 is incorrect.

An individual DriverLINX buffer may have any size as long as the buffer length holds an integral number of channel scans (i.e., a multiple of the number of analog output channels you're acquiring). This restriction enforces the requirement that all acquired channels have the same number of samples.

Buffer Usage

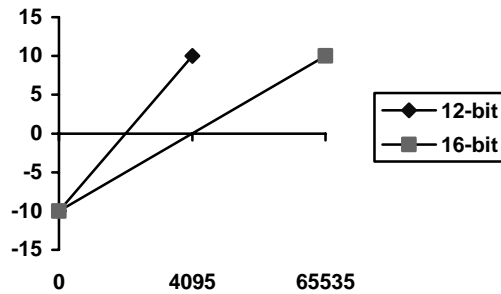
DriverLINX writes from buffers sequentially until the task stops. Except for tasks that stop on terminal count, the last buffer may be only partially used. If the task stops on a trigger, use the StopEvent message (or event) to determine the location of the last sample. For other cases, use a Status operation to determine the location of the last sample.

Analog Output Data Coding

KPCI-3100 Series uses the right-justified, offset binary format to encode analog output data, as shown in the following table. DriverLINX refers to this coding scheme as the “native” format.

Model	Analog Output Resolution	Analog Output Hardware Code
KPCI-3102/04/10	12 bits	0 to 495
KPCI-3116	16 bits	0 to 65535

Native Analog Output hardware codes for the KPCI-3100 Series.



KPCI-3100 Series native Analog Output Codes versus Voltage Range for a ten-volt range.

DriverLINX refers to the default hardware analog-coding scheme as the “native” format. For computer arithmetic in a high-level language, the 16-bit two’s complement integer format is generally easier to use. Applications can use DriverLINX’s data conversion operations to transform an entire data buffer from many common integer and floating-point formats to native format.

Analog Output Messages

For analog output operations, DriverLINX can report the following messages to the application:

DriverLINX Message	Explanation
Service Start	DriverLINX has started the acquisition task.
Service Done	DriverLINX has completed the acquisition task.
Start Event	DriverLINX has detected a hardware start event.
Buffer Filled	DriverLINX has filled an analog output buffer.
Critical Error	DriverLINX has encountered an unexpected hardware or software condition.

DriverLINX Event messages for analog output.

For detailed explanations of these messages see one of the following references:

- *DriverLINX Technical Reference Manual* for C/C++ users
- *DriverLINX/VB Technical Reference Manual* for VB or Delphi users

If your application does not need to change the output waveform during the task—that is, does not need Buffer Filled messages—you may achieve a performance benefit by using AO Recycle Mode.

AO Recycle Mode

Under certain conditions, the models KPCI-3110 and 3116 support AO Recycle Mode (also known as “waveform generation”). Recycle mode allows the board to generate a periodic waveform by recycling a buffer of up to 4096 samples. For recycle mode, DriverLINX automatically promotes AO tasks meeting the following criteria as specified in the Service Request to run from the DAC FIFO buffer in recycle mode:

Service Request	Property Value
Request mode	Interrupt or DMA
Stop event type	Command
Buffer notify flag	False
Buffer samples	2 to 2048 per channel
Number of buffers	1

The advantages of using the DAC FIFO buffer for recycle mode analog output are:

- The digital trigger operates in retrigger mode, enabling you to synchronize analog output with a recurring external signal.
- Traffic on the memory bus is significantly reduced at high output rates.
- The system response latency will be noticeably shortened.
- The analog output timing will be immune to traffic on the memory and I/O buses.

The disadvantages are:

- You cannot change the analog output values while the analog output task is running.
- There is a delay while DriverLINX loads entire wave fore into the FIFO.
- Buffer-filled notification messages cannot be posted, as the hardware does not permit detection of end-of-buffer conditions.
- The STATUS operation cannot provide the "number of buffers processed" report.

Digital Input Subsystem

The following sections describe how DriverLINX implements Digital Input Subsystem features for the KPCI-3100 Series.

Digital Input Modes

The Digital Input Subsystem supports the following modes:

- **Polled**—For single-value or scan digital input samples.
- **Interrupt**—For buffered transfers using programmed I/O.
- **DMA**—For buffered transfers using direct memory access.
- **Other**—For subsystem initialization.

Digital Input Operations

The KPCI-3100 Series Digital Input Subsystem supports the following DriverLINX operations:

- **Initialize**—aborts any active interrupt data-acquisition tasks and stops the clock. DriverLINX prevents one application from interfering with another application's data-acquisition tasks.
- **Configure**—assigns a configurable digital channel to the input subsystem.
- **Start**—initiates a data-acquisition task using the Mode, Timing, Start, and Stop Events, the Logical Channels, and the Buffers the application specified in the Service Request.
- **Status**—reports the buffer position of the next sample that DriverLINX will write into a buffer.
- **Stop**—terminates a digital input data-acquisition task.
- **Message**—displays a pop-up dialog box for the user containing the text for the current DriverLINX error message.

Digital Channel Configuration

The KPCI-3100 Series has two or three configurable digital channels, depending on the model. To use a digital channel as an input, you must first assign it to the Digital Input Subsystem using the “Configure DriverLINX Device Dialog” (page 11) or by using a Configure operation.

Special...To configure a digital channel submit a service request with following settings:

- **Subsystem**—Digital Input or Digital Output
- **Mode**—Other
- **Operation**—Configure
- **Timing Event**—DIO Setup
- **DIO Setup Channel**—Channel number

- **DIO Setup Mode**—DIO BASIC

For more information, see the following topics in the *Digital I/O Programming Guide*:

- Configuring a Digital Port at Run Time
- Determining the Digital Port Configuration

Example: For a programming example, see DIOCNFIG in your DrvLINX4\Source folder.

Digital Input Timing Events

Timing Events specify how the hardware paces or clocks the reading of Digital Input samples. DriverLINX uses the Timing Event to program when the KPCI-3100 Series reads the next digital input sample from the port.

The KPCI-3100 Series supports the following Timing Events:

- **None**—Input requires no pacing as DriverLINX is reading only a single value or scan.
- **Rate**—The KPCI-3100 Series supports fixed rate sampling using internal and external clocks.

None or Null Timing Event

The Null Event specifies that the task does not need a clock to determine when to read the next sample.

Note: Models KPCI-3101/02/03/04 require this mode when sampling Logical Channel 2.

Rate Timing Event

The KPCI-3100 Series supports the following Rate Event for digital input:

- **Rate Generator**—Generates a fixed rate clock with equal time intervals between tics.

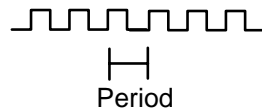


KPCI-3100 Series boards have a fixed 20 MHz master clock frequency. The sample period range depends on the model. All models limit the period to $2^{24} - 1$ tics, or 838,860 μ s (1.20 Hz). Consult your hardware manual for details.

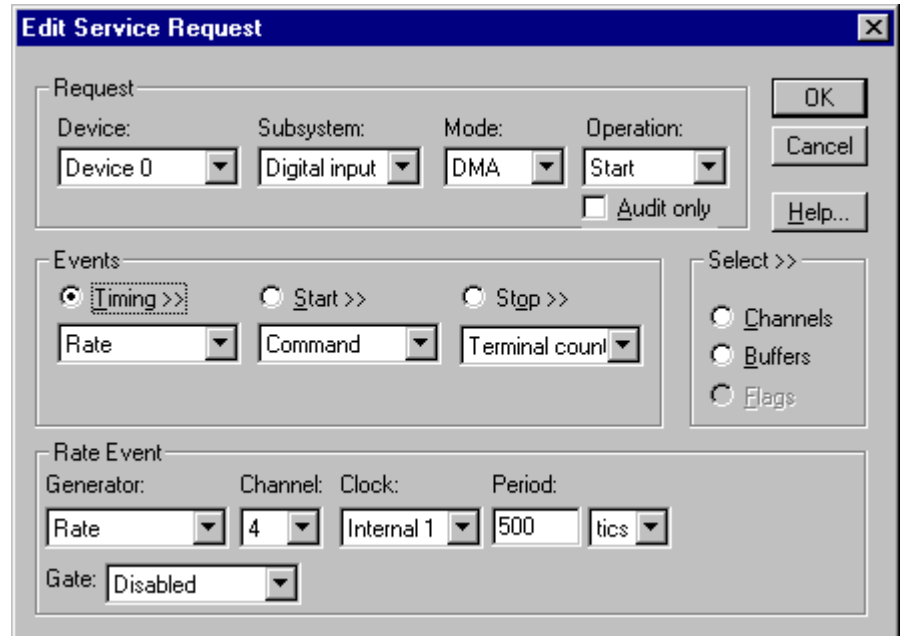
Note: The KPCI-3100 Series can sample from only Logical Channels 0 and 1 in this mode.

Rate Generator: Internal Clocking

An internally clocked Rate Generator produces a fixed rate clock with equal time intervals between tics.



Use an internally clocked rate generator when you want to acquire all digital input scans at equally spaced time intervals.



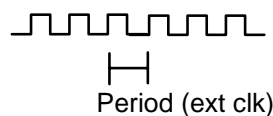
How to set up the KPCI-3100 Series for fixed rate sampling using an internal clock.

For hardware independence, specify the clock channel using the symbolic constant, DEFAULTTIMER, which always maps to the default Logical Channel for digital output timing.

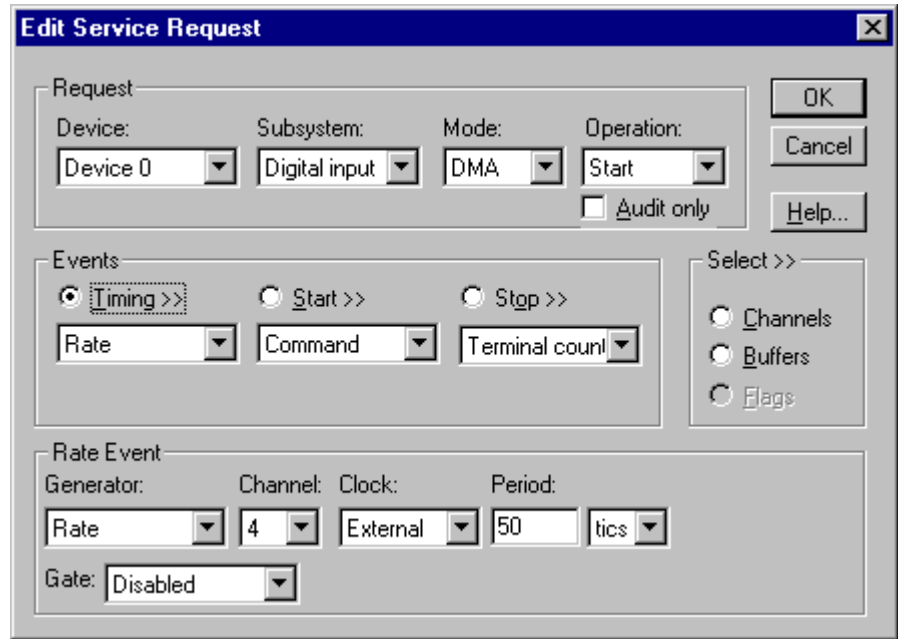
- Specify internal clocking using a **Rate Generator** on **Channel 4** with the **Internal1** Clock source.
- The *Period* property specifies the time interval between samples in tics.
- The KPCI-3100 Series does not support gating the analog input pacing clock. Therefore, only Disabled and NoConnect settings are valid for the *Gate* property.

Rate Generator: External Clocking

An externally clocked Rate Generator produces a rate clock with unknown time intervals between tics.



Use an externally clocked rate generator when you want to synchronize digital input samples with a recurrent external signal. You need a separate external clock tic for each scan you want to acquire.



How to set up the KPCI-3100 Series for fixed rate sampling using an external clock.

For hardware independence, specify the clock channel using the symbolic constant, *DEFAULTTIMER*, which always maps to the default Logical Channel for analog input timing.

Be sure that the external clock source is TTL compatible, 0 V minimum to +5 V maximum!

- Specify external clocking using a **Rate Generator** on **Channel 4** with an **External** or **External- Clock** source.
- Specify a *Period* between the minimum and maximum external clocking period. The value does not affect the external clock frequency, but DriverLINX requires a valid hardware value in case the application requests a timebase operation and to optimize data transfer between the driver and the application.
- The KPCI-3100 Series does not support gating the analog input pacing clock. Therefore, only Disabled and NoConnect settings are valid for the *Gate* property.
- Users should connect the external clock signal to the External A/D Clock Input line.
- The frequency of the external clock must not exceed the maximum sampling rate for the board.

Note: The hardware transfers samples in groups of 16 on models KPCI-3101/02/03/04 and 32 on models KPCI-3110/16. Therefore, samples may be held on the board for up to 31 external clock pulses after they are taken. When acquiring a fixed number of samples, ensure the external clock continues until all desired samples are transferred from the board.

Digital Input Start Events

Start Events specify when the KPCI-3100 Series hardware starts reading digital input data.

The KPCI-3100 Series supports the following Start Events for digital input:

- **None**—Use this event when the DriverLINX operation does not require a Start Event.

- **Command**—DriverLINX starts the task on software command, i.e., as soon as DriverLINX finishes programming the KPCI-3100 hardware for the task.
- **Analog**—The hardware starts acquiring digital input samples when the hardware detects an analog signal satisfies the condition specified in the Start Event.
- **Digital**—The hardware starts acquiring analog input samples when the hardware detects the digital Logical Channel input satisfies the condition specified in the Start Event.

None or Null Start Event

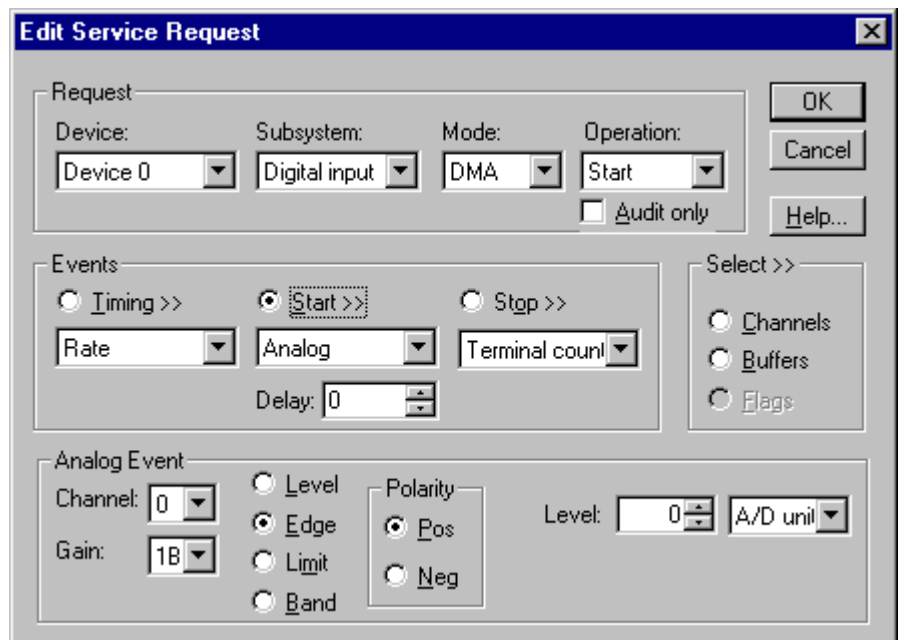
The Null Event specifies that the task does not need a Start Event to begin the task.

Command Start Event

The Command Event starts data acquisition as soon as DriverLINX has completed programming the KPCI-3100 Series hardware with the task parameters.

Analog Start Event (Post-Triggering)

The KPCI-3100 Series can acquire digital input samples *after* the hardware detects an analog trigger condition. Use post-triggering when you want to synchronize the start of data acquisition with an external signal.



How to set up the KPCI-3100 Series for about-triggered analog input.

Analog Start Events contain *Channel*, *Gain*, *Polarity* and *Limit* fields. The limits determine the type of analog event (Level, Edge, Limit, or Band). The KPCI-3100 Series supports only edge triggering. A rising trigger occurs when a sample above the limits follows a sample below the limits. Similarly, a falling trigger occurs when a sample below the limits follows a sample above the limits. As the board acquires samples, it compares data from the specified *Channel* against the identical *High* and

Low Limits. The Polarity determines whether the trigger occurs on a rising or falling signal.

- Specify the *Channel* and *Gain* for a channel in the analog input subsystem. For the KPCI-3100 Series, the analog event can use only the Analog Trigger input.

To trigger on the Analog Trigger input, specify **0** as the *Channel* and any gain plus the OEM Gain Flag as the *Gain*. (The Edit Service Request Dialog does not provide this option but it is available programmatically.) DriverLINX supports a programmable gain as a convenience. The KPCI-3100 Series actually supports only a fixed range of $\pm 10V$ with an 8-bit resolution but you can still use *Gain2Code* and *Volts2Code* to calculate the threshold settings.

Specify the *Polarity* (or Slope) property as **Pos** for a rising trigger or **Neg** for a falling trigger.

- Specify the *Limit* properties in hardware A/D codes as follows:

Type	High Limit	Low Limit
Edge	Threshold	Threshold

Programming Tip: Use *Volts2Code* to easily convert volts to hardware A/D codes for the threshold properties.

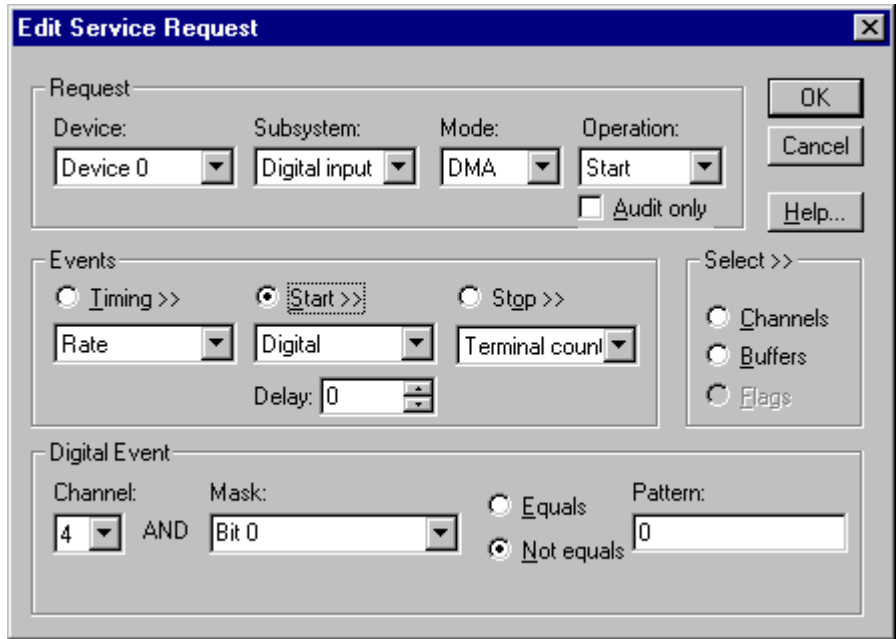
- Specify the *Delay* property as **0** or a positive number of samples to skip after the trigger. The *Delay* value must be multiple of the number of channels in the channel list.

Note:

- When the hardware detects the trigger, DriverLINX sends a StartEvent message (or event). For more information on messages, see “Digital Input Messages” on page 89.

Digital Start Event (Post-Triggering)

The KPCI-3100 can acquire digital input samples *after* the hardware detects a digital trigger condition. Use post-triggering when you want to synchronize the start of data acquisition with an external signal.



How to set up the KPCI-3100 Series for post-triggered digital input.

Digital Start Events contain *mask*, *pattern*, and *match* fields. The mask is logically ANDed with the digital input data on the Logical Channel and then compared with the *pattern* for a match/mismatch.

- Specify the *Channel* as **3** for models KPCI-3110/16 or **4** for models KPCI-3101/02/03/04. For hardware-independence, you can specify the hardware external trigger channel by the symbolic constant, *DI_EXTTRG*.
- Users should connect the external trigger signal to the External A/D Trigger line.
- Specify the *Mask* property as **1**, or **Bit 0**, to indicate that DriverLINX should only compare a 1-bit digital input value against the *Pattern* property.
- Specify the *Match* property as **Not equals**.
- Specify the *Pattern* property as **0** for a rising, or positive, edge trigger ($\neq 0$) or **1** for a falling, or negative, edge trigger ($\neq 1$).
- Specify the *Delay* property as **0** or a positive number of samples to skip after the trigger. The *Delay* value must multiple of the number of channels in the channel list.

Digital Input Stop Events

Stop Events specify when the KPCI-3100 Series hardware stops reading digital input data.

The KPCI-3100 Series supports the following Stop Events for digital input:

- **None**—Use this event when the DriverLINX operation does not require a Stop Event.

- **Command**—DriverLINX stops the task on software command, i.e., when the application issues a Service Request with a *Stop* operation.
- **Terminal count**—DriverLINX stops the task after the KPCI-3100 Series hardware has filled all the data buffers once.
- **Analog**—The hardware stops acquiring digital input samples when the hardware detects an analog signal satisfies the condition specified in the Start Event.
- **Digital**—The hardware stops acquiring analog input samples when the hardware detects the digital Logical Channel input satisfies the condition specified in the Stop Event.

None or Null Stop Event

The Null Event specifies that the task does not need a Stop Event to end the task.

Command Stop Event

The Command Event stops data acquisition when the user application changes the *Operation* property in the Service Request to *Stop* and resubmits the Service Request to DriverLINX.

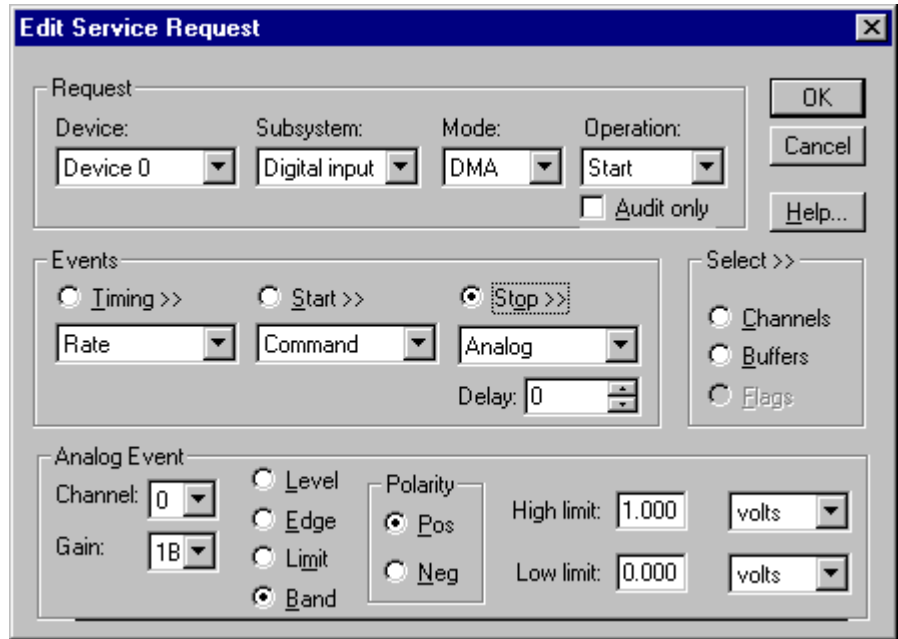
In Stop-on-Command mode, DriverLINX continuously cycles through all the data buffers, filling them with digital input data from the data-acquisition hardware.

Terminal Count Stop Event

The Terminal Count Event stops data acquisition after DriverLINX has read the digital input data into all the data buffers *once*. Use terminal count when you want to read a fixed amount of data.

Analog Stop Event (Pre- and About-Triggering)

The KPCI-3100 can acquire digital input samples *until* the hardware detects an analog trigger condition. Use pre-triggering when you want to synchronize the end of data-acquisition with an external signal. Use about-triggering when you want to delay the end of data-acquisition a fixed number of samples after the trigger.



How to set up the KPCI-3100 Series for pre-triggered digital input.

Analog Stop Events contain *Channel*, *Gain*, *Polarity* and *Limit* fields. The limits determine the type of analog event (Level, Edge, Limit, or Band). The KPCI-3100 Series supports only edge triggering. A rising trigger occurs when a sample above the limits follows a sample below the limits. Similarly, a falling trigger occurs when a sample below the limits follows a sample above the limits. As the board acquires samples, it compares data from the specified *Channel* against the identical *High* and *Low Limits*. The *Polarity* determines whether the trigger occurs on a rising or falling signal.

- Specify the *Channel* and *Gain* for a channel in the analog input subsystem. For the KPCI-3100 Series, the analog event can use only the Analog Trigger input.

To trigger on the Analog Trigger input, specify **0** as the *Channel* and any gain plus the OEM Gain Flag as the *Gain*. (The Edit Service Request Dialog does not provide this option but it is available programmatically.) DriverLINX supports a programmable gain as a convenience. The KPCI-3100 Series actually supports only a fixed range of $\pm 10V$ with an 8-bit resolution but you can still use *Gain2Code* and *Volts2Code* to calculate the threshold settings.

- Specify the *Polarity* (or Slope) property as **Pos** for a rising trigger or **Neg** for a falling trigger.
- Specify the *Limit* properties in hardware A/D codes as follows:

Type	High Limit	Low Limit
Edge	Threshold	Threshold

Programming Tip: Use *Volts2Code* to easily convert volts to hardware A/D codes for the threshold properties.

- Specify the *Delay* property as **0** for pre-triggering, or a positive number of samples to obtain after the trigger for about-triggering. The *Delay* value must be a multiple of the number of the number of channels in the channel list.

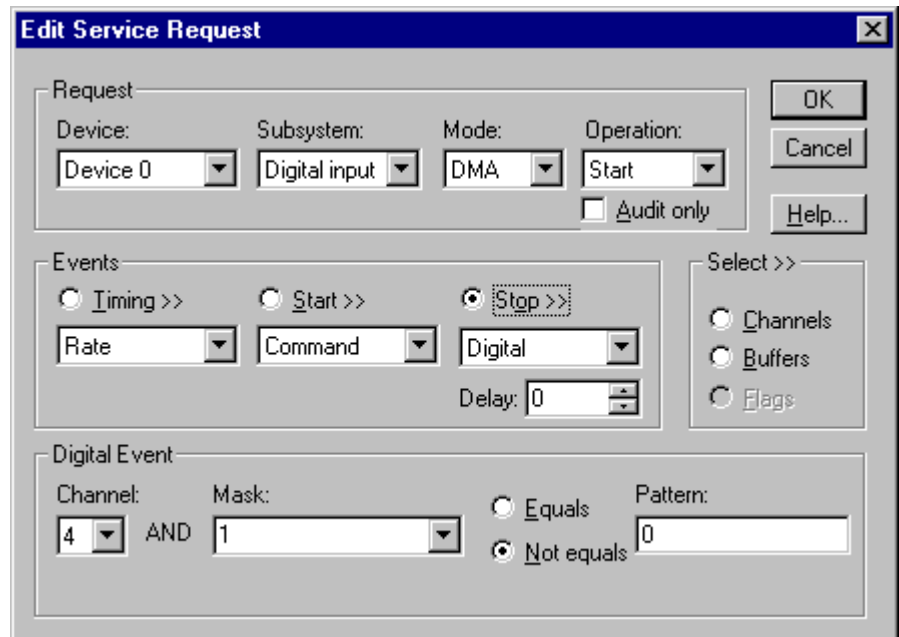
The KPCI-3100 Series supports stopping immediately upon detection of the trigger, rather than at the end of the current scan. To use this feature specified the *Delay* property as **1**.

Note:

- When the hardware detects the trigger, DriverLINX sends a StopEvent message (or event) that identifies the buffer location of the last sample. For more information on messages, see “Digital Input Messages” on page 89.

Digital Stop Event (Pre- and About-Triggering)

The KPCI-3100 can acquire digital input samples *until* the hardware detects a digital trigger condition. Use pre-triggering when you want to synchronize the end of data acquisition with an external signal.



How to set up the KPCI-3100 Series for pre-triggered digital input.

Digital Stop Events contain *mask*, *pattern*, and *match* fields. The mask is logically ANDed with the digital input data on the Logical Channel and then compared with the *pattern* for a match/mismatch.

- Specify the *Channel* as **3** for models KPCI-3110/16 or **4** for models KPCI-3101/02/03/04. For hardware-independence, you can specify the hardware external trigger channel by the symbolic constant, *DI_EXTTRG*.
- Users should connect the external trigger signal to the External A/D TTL Trigger line.

- Specify the *Mask* property as **1**, or **Bit 0**, to indicate that DriverLINX should only compare a 1-bit digital input value against the *Pattern* property.
- Specify the *Match* property as **Not equals**.
- Specify the *Pattern* property as **0** for a rising, or positive, edge trigger ($\neq 0$) or **1** for a falling, or negative, edge trigger ($\neq 1$).
- Specify the *Delay* property as **0** for pre-triggering, or a positive number of samples to obtain after the trigger for about-triggering. The *Delay* value must be a multiple of the number of channels in the channel list.

The KPCI-3100 Series supports stopping immediately upon detection of the trigger, rather than at the end of the current scan. To use this feature specified the *Delay* property as **1**.

Note:

- When the hardware detects the trigger, DriverLINX sends a StopEvent message (or event) that identifies sample following the trigger. For more information, see “Digital Input Messages” on page 89.

Digital Input Channels

The KPCI-3100 Series allows applications to specify the digital channels using three techniques:

- **Start Channel**—Acquire data from a single channel.
- **Start/Stop Channel Range**—Acquire data from a consecutive range of channels.
- **Channel List**—Acquire data from a list of channels.

Digital Input Logical Channels

The KPCI-3100 Series supports two or three configurable digital channels, depending on the model. DriverLINX defines the following Logical Channels for the KPCI-3100 Series Digital Input Subsystem:

Models KPCI-3101/02/03/04

Logical Channel	DriverLINX Function	KPCI-3100 Series External Connector
0	Digital Input/Output	Digital I/O Bank A 0 ... 7
1	Digital Input/Output	Digital I/O Bank B 0 ... 7
2	Digital Input/Output	Digital I/O Bank C 0 ... 6
3	External Clock	External A/D Sample Clock Input External D/A Sample Clock Input
4	External Trigger	External A/D TTL Trigger External D/A TTL Trigger

Models KPCI-3110/16

Logical Channel	DriverLINX Function	KPCI-3100 Series External Connector
0	Digital Input/Output	Digital I/O Bank A 0 ... 7
1	Digital Input/Output	Digital I/O Bank B 0 ... 7
2	External Clock	External A/D Sample Clock Input External D/A Sample Clock Input
3	External Trigger	External A/D TTL Trigger External D/A TTL Trigger

Single Channel Digital Input

In single channel mode, the KPCI-3100 Series acquires all data from one channel.

- To sample Logical Channel 2 on models KPCI-3101/02/03/04, you must use polled mode.

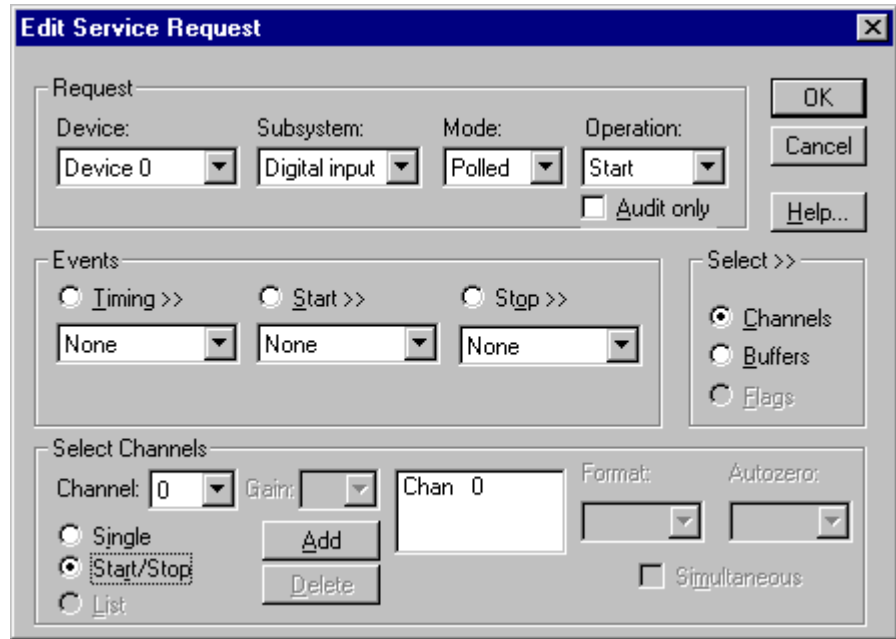
The screenshot shows the 'Edit Service Request' dialog box. The 'Request' section includes fields for Device (Device 0), Subsystem (Digital input), Mode (Polled), and Operation (Start). There is an 'Audit only' checkbox. The 'Events' section has radio buttons for Timing, Start, and Stop, each with a dropdown menu set to None. The 'Select Channels' section has a Channel dropdown set to 0, Gain, Format, and Autozero dropdowns, and radio buttons for Single, Start/Stop, and List. The 'Single' radio button is selected. There are 'Add' and 'Delete' buttons. A 'Simultaneous' checkbox is also present. On the right side, there are 'OK', 'Cancel', and 'Help...' buttons, and a 'Select >>' section with radio buttons for Channels, Buffers, and Flags, where 'Channels' is selected.

How to set up the KPCI-3100 Series for sampling on a single channel.

Multi-channel Digital Input Range

In multi-channel range mode, the KPCI-3100 Series acquires all data from a consecutive range of digital channels.

- The Stop Channel must be greater than the Start Channel.
- In DMA mode, the Start channel must be Logical Channel 0 and the Stop channel must be Logical Channel 1.
- To sample Logical Channel 2, on models KPCI-3101/02/03/04, you must use polled mode.

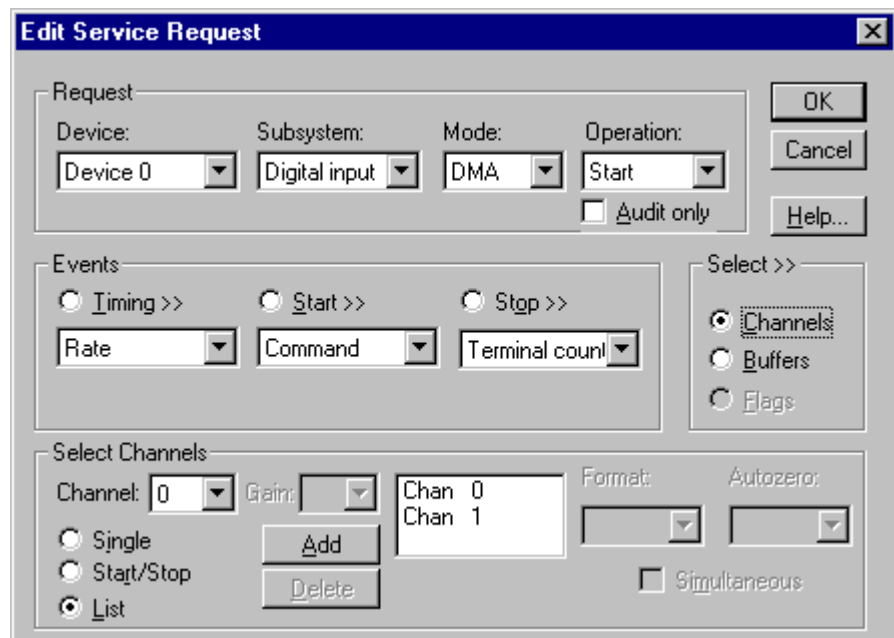


How to set up the KPCI-3100 Series for sampling on a consecutive range of channels.

Multi-channel Digital Input List

In multi-channel list mode, the KPCI-3100 Series acquires all data from an arbitrary list of digital channels.

- In Interrupt or DMA mode, the list can specify only Logical Channels 0 and 1.
- To sample Logical Channel 2, on models KPCI-3101/02/03/04, you must use polled mode.

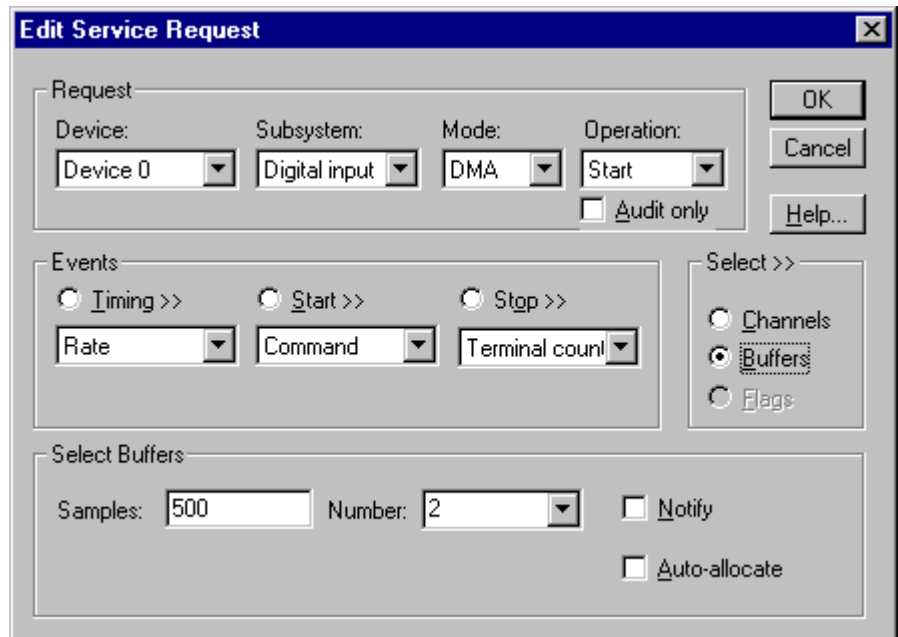


How to set up the KPCI-3100 Series to sample from an arbitrary list of channels.

Digital Input Buffers

DriverLINX supports single-value, single-scan and buffered digital input.

- **For single-value input**, specify the *Number* of buffers as **0**. The buffer for a single value is the *ioValue* property.
- **For single-scan input**, specify the *Number* of buffers as **1** and the number of *Samples* equal to the number of channels.
- **For buffered input**, specify the *Number* of buffers from **1** to **255** and the number of *Samples* as desired.



How to set up the KPCI-3100 Series to read digital samples using data buffers.

Buffer Usage

DriverLINX fills buffers sequentially until the task stops. During the task, only complete buffers are available to the application. Except for tasks that stop on terminal count, the last buffer may be only partially full. Use a Status operation to determine the location of the last sample.

Note: The hardware transfers samples in groups of 16 on models KPCI-3101/02/03/04 and 32 on models KPCI-3110/16. Therefore, when stopping on command, the last buffer may contain up to 31 invalid samples before the stop position reported by DriverLINX.

Digital Input Messages

For digital input operations, DriverLINX can report the following messages to the application:

DriverLINX Message	Explanation
Service Start	DriverLINX has started the acquisition task.
Service Done	DriverLINX has completed the acquisition task.
Start Event	DriverLINX has detected the hardware start event.
Stop Event	DriverLINX has detected the hardware stop event.
Buffer Filled	DriverLINX has filled a digital input buffer.
Critical Error	DriverLINX has encountered an unexpected hardware or software condition.

DriverLINX Event messages for digital input.

For detailed explanations of these messages see one of the following references:

- *DriverLINX Technical Reference Manual* for C/C++ users
- *DriverLINX/VB Technical Reference Manual* for VB or Delphi users

Digital Output Subsystem

The following sections describe how DriverLINX implements Digital Output Subsystem features for the KPCI-3100 Series.

Digital Output Modes

The Digital Output Subsystem supports the following modes:

- **Polled**—For single-value or scan digital output samples.
- **Other**—For subsystem initialization.

Digital Output Operations

The KPCI-3100 Series Digital Output Subsystem supports the following DriverLINX operations:

- **Initialize**—aborts any active interrupt data-acquisition tasks and stops the clock. DriverLINX prevents one application from interfering with another application's data-acquisition tasks.
- **Start**—initiates a data-acquisition task using the Mode, Timing, Start, and Stop Events, the Logical Channels, and the Buffers the application specified in the Service Request.
- **Message**—displays a pop-up dialog box for the user containing the text for the current DriverLINX error message.

Digital Output Initialization

By default, Digital Output Subsystem initialization writes zeros to all the digital I/O ports configured for output. You can specify a different initial output value using the *Configure DriverLINX Device* dialog. See “Digital Output Subsystem Page” on page 19.

Digital Channel Configuration

The KPCI-3100 Series has two or three configurable digital channels, depending on the model. To use a digital channel as an output, you must first assign it to the Digital Output Subsystem using the “Configure DriverLINX Device Dialog” (page 11) or by using a Configure operation. See “Digital Channel Configuration” on page 74 for more information on the configure operation.

Digital Output Timing Events

Timing Events specify how the hardware paces or clocks writing Digital Output samples. DriverLINX uses the Timing Event to program when the KPCI-3100 Series writes the next digital output sample from the port.

The KPCI-3100 Series supports the following Timing Events for digital output:

- **None**—Output requires no pacing as DriverLINX is writing only a single value or scan.

None or Null Timing Event

The Null Event specifies that the task does not need a clock to determine when to write the next sample.

Digital Output Start Events

Start Events specify when the KPCI-3100 Series hardware starts writing digital output data.

The KPCI-3100 Series supports the following Start Events for digital output:

- **None**—Use this event when the DriverLINX operation does not require a Start Event.
- **Command**—DriverLINX starts the task on software command, i.e., as soon as DriverLINX finishes programming the KPCI-3100 hardware for the task.

None or Null Start Event

The Null Event specifies that the task does not need a Start Event to begin the task.

Command Start Event

The Command Event starts data acquisition as soon as DriverLINX has completed programming the KPCI-3100 hardware with the task parameters.

Digital Output Stop Events

Stop Events specify when the KPCI-3100 Series hardware stops writing digital output data.

The KPCI-3100 Series supports the following Stop Events for digital output:

- **None**—Use this event when the DriverLINX operation does not require a Stop Event.
- **Terminal count**—DriverLINX stops the task after the KPCI-3100 Series hardware has written all the data buffers once.

None or Null Stop Event

The Null Event specifies that the task does not need a Stop Event to end the task.

Terminal Count Stop Event

The Terminal Count Event stops data acquisition after DriverLINX has written the digital output data from all the data buffers *once*. Use terminal count when you want to write a fixed amount of data.

Digital Output Channels

The KPCI-3100 Series allows applications to specify the digital channels using three techniques:

- **Start Channel**—Write data to a single channel.
- **Start/Stop Channel Range**—Write data to a consecutive range of channels.
- **Channel List**—Write data to a list of channels.

Digital Output Logical Channels

The Digital Output subsystem has two or three configurable digital input/output channels, depending on the model. DriverLINX maps these signals to Logical Channels as shown in the following table:

Models KPCI-3101/02/03/04

Logical Channel	DriverLINX Function	KPCI-3100 Series External Connector
0	Digital Input/Output	Digital I/O Bank A 0 ... 7
1	Digital Input/Output	Digital I/O Bank B 0 ... 7
2	Digital Input/Output	Digital I/O Bank C 0 ... 6

Models KPCI-3110/16

Logical Channel	DriverLINX Function	KPCI-3100 Series External Connector
0	Digital Input/Output	Digital I/O Bank A 0 ... 7
1	Digital Input/Output	Digital I/O Bank B 0 ... 7

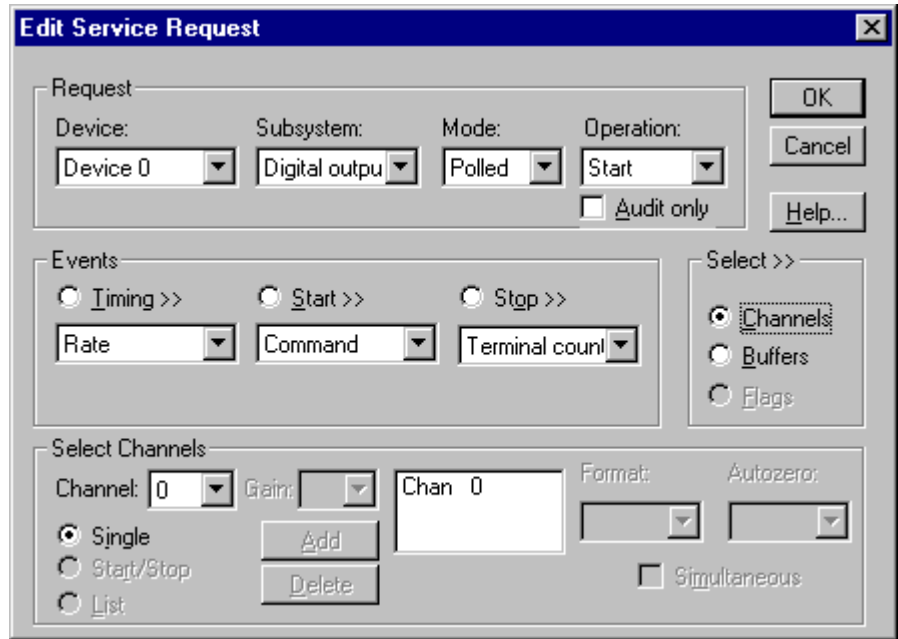
How DriverLINX maps digital output hardware channels to Logical Channels.

Notes:

- If a channel is configured for output, reading it using the Digital Input subsystem returns the last value written.
- Applications can assign a configurable channel to either subsystem using a Configure operation. (See “Digital Channel Configuration” on page 74.)
- The KPCI-3100 Series also supports Dynamic Digital Output, which allows an Analog Input task to output digital data. See “Writing Digital Output Channels with the Analog Input Subsystem” on page 54 for more information on this feature.

Single Channel Digital Output

In single channel mode, the KPCI-3100 Series writes all data from one channel.

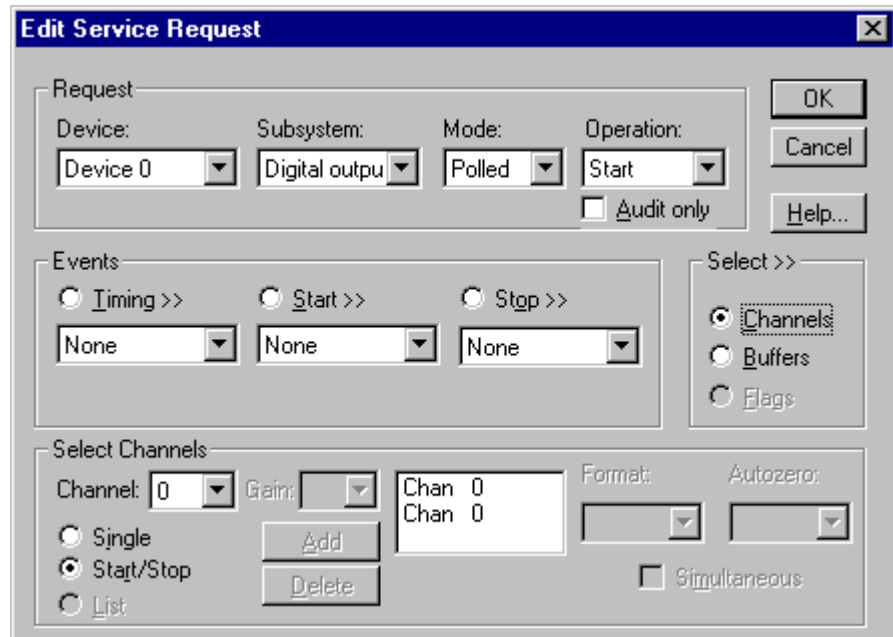


How to set up the KPCI-3100 Series for writing to a single channel.

Multi-channel Digital Output Range

In multi-channel range mode, the KPCI-3100 Series acquires all data from a consecutive range of digital channels.

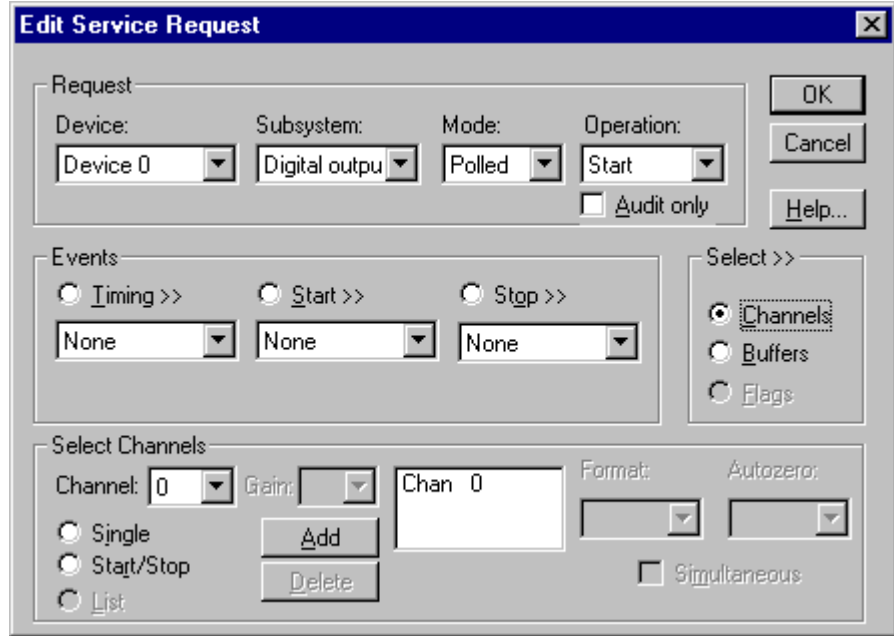
- The Stop Channel must be greater than the Start Channel.



How to set up the KPCI-3100 Series for writing to a consecutive range of channels.

Multi-channel Digital Output List

In multi-channel list mode, the KPCI-3100 Series acquires all data from an arbitrary list of digital channels.

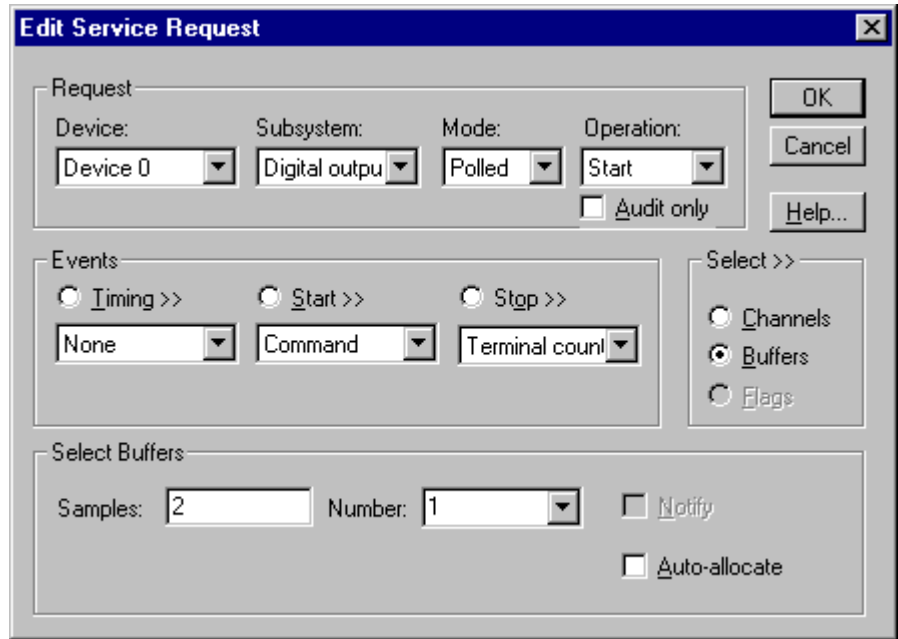


How to set up the KPCI-3100 Series to write to an arbitrary list of channels.

Digital Output Buffers

DriverLINX supports single-value, single-scan and buffered digital output.

- **For single-value output**, specify the *Number* of buffers as **0**. The buffer for a single value is the *ioValue* property.
- **For single-scan output**, specify the *Number* of buffers as **1** and the number of *Samples* equal to the number of channels.



How to set up the KPCI-3100 Series to store samples in buffers.

Digital Output Messages

For digital output operations, DriverLINX can report the following messages to the application:

DriverLINX Message	Explanation
Service Start	DriverLINX has started the acquisition task.
Service Done	DriverLINX has completed the acquisition task.
Critical Error	DriverLINX has encountered an unexpected hardware or software condition.

DriverLINX Event messages for digital output.

For detailed explanations of these messages see one of the following references:

- *DriverLINX Technical Reference Manual* for C/C++ users
- *DriverLINX/VB Technical Reference Manual* for VB or Delphi users

Counter/Timer Subsystem

The KPCI-3100 Series provides counter/timers for analog and digital input/output pacing as well as independent counter/timer tasks. The KPCI-3100 Series uses the same proprietary counter/timer chip as the KPCI-3140 Counter/Timer board. On the KPCI-3100 Series, the chips counter/timers serve as dedicated pacer clocks, scan clocks and general-purpose counter/timers.

All models have a 24-bit A/D clock to pace analog input. DriverLINX designates this clock as Logical Channel 4. The analog input subsystem also has a 24-bit scan clock. DriverLINX uses this clock in conjunction with the A/D clock when required by the task. Models KPCI-3110 and 3116 have a 24-bit D/A clock to pace analog output. DriverLINX designates this clock as Logical Channel 5.

The KPCI-3100 Series does not have a clock for digital input/output pacing. Therefore, DriverLINX performs digital output in polled mode only. For digital input, DriverLINX takes advantage of a feature that allows sampling Logical Channels 0 and 1 with the A/D clock, as well as supporting digital input on all channels in polled mode.

The KPCI-3100 Series also has external trigger and clock inputs for the analog input and output, and digital input subsystems. See “Analog Input Subsystem Signals” on page 29, “Analog Output Subsystem Signals” on page 30 or “Digital Input Subsystem Signals” on page 31 for connection details.

All models have four 16-bit general-purpose counter/timers that support a wide variety of counting and generation functions. DriverLINX designates these counters as Logical Channels 0 through 3. Each counter/timer has a clock input, a gate input and a counter output.

Model KPCI-3130

Logical Channel	DriverLINX Function	KPCI-3130 Series External Connector
0	External Clock (sample on DAC overload)	+/- DAC Overload Sense Lines 0 ... 7

An application specifies operating parameters for a counter/timer using a Rate Event for the timing event in a service request. A Rate Event has clock source, mode and gate properties. DriverLINX defines many useful generic counter/timer functions that can be performed with most counter/timer chips. Alternatively, you can use DriverLINX to program the board in specific modes of the proprietary, KPCI-3140 compatible, counter/timer chip. See the *DriverLINX Counter/Timer User's Guide* for detailed information on programming the KPCI-3140 counter/timer chip.

The following sections discuss usage of Rate Event properties for both input/output task and counter/timer tasks.

Counter/Timer Channel Usage

The following table shows which Logical Channels are available for tasks on each subsystem.

Counter/Timer Logical Channel Usage

Subsystem	Counter/Timer Channel					
	0	1	2	3	4	5
Analog Input					Default	
Analog Output						Default
Digital Input					Default	
Digital Output						
Counter/Timer	Default	Available	Available	Available	Available	Available

For portability across drivers, applications can specify the default timing channel using the symbolic constant DEFAULTTIMER, rather than a specific channel number.

Clock Sources and Modes

DriverLINX defines several internal and external clock sources and modes for Rate Events on the KPCI-3100 Series. Some apply only to pacing certain subsystems. Others are only for counter/timer tasks. The following tables lists the clock sources and the modes to which they apply for use with each subsystem:

Analog Input

Name	Source	Modes
Internal 1	Internal 20 MHz	RateGen BurstGen
External External+ External-	External A/D TTL Trigger (major period) Internal 20 MHz (minor period)	BurstGen
External External-	External A/D Clock Input	RateGen

Analog Output

Name	Source	Modes
Internal 1	Internal 20 MHz	RateGen BurstGen
External External+ External-	External D/A TTL Trigger (major period) Internal 20 MHz (minor period)	BurstGen
External External-	External D/A Clock Input	RateGen

Digital Input

Name	Source	Modes
Internal 1	Internal 20 MHz	RateGen
External External-	External A/D Clock Input	RateGen

Digital Output

The KPCI-3100 Series allows only untimed, polling of digital output channels.

Counter/Timer

Name	Source	Modes
Internal 1	Internal 20 MHz	RateGen
External	User Counter Input 0	Divider
External+	User Counter Input 1	VDCGen
TCnml [*]	User Counter Input 2	SqWave
	User Counter Input 3	Freq
		Freq32
		Count
		Count32
		Count64
		PulseWd
		PulseGen
		OneShot

^{*}Logical channels 1, 2 and 3 only

For portability across hardware with different drivers, applications can use the **Sec2Tics** function to determine the tic period of **Internal 1**.

Gate Settings

The *Gate* setting specifies how the gate signal affects the operation of the internal or external clock. The following settings are valid, depending on the subsystem:

Gate Setting	Subsystem				
	AI	AO	DI	DO	CT
Enabled selects the gate's default enabled mode. On the KPCI-3100 Series this is either High Level or Rising Edge, depending on the rate mode.					✓
Disabled allows the clock to run independently of the gate signal.	✓	✓	✓	✓	✓
No Connect normally allows the clock to run independently of the gate signal.	✓	✓	✓	✓	✓
High Level enables the clock only while the gate signal is high.					✓
Low Level enables the clock only while the gate signal is low.					✓
High Edge enables the clock after the first rising edge in the gate signal.					✓
Low Edge enables the clock after the first falling edge in the gate signal.					✓

Note: No Connect specifies that if the application is running with a board that cannot disable its gate, then the user should leave the gate unconnected so that it does not interfere with the data-acquisition task. This mode is accepted by all DriverLINX drivers.

Counter/Timer Tasks

Counter/timer tasks, as distinguished from input/output task pacing, perform various counting, measuring, and wave and pulse generation functions. Counter/timer hardware varies widely in complexity and utility. To simplify most counter/timer tasks while allowing applications to access to features of particular chips, DriverLINX allows applications to specify either function-oriented task or hardware-oriented tasks.

DriverLINX’s function-oriented tasks that the KPCI-3100 Series can perform are

- **Event Counting**
Repetitive 16, 32 or 64-bit event counting with optional level gating
Non-repetitive 16, 32 or 64-bit event counting without gating
- **Frequency Measurement**
Non-repetitive, 16 or 32-bit frequency measurement with optional level gating (these functions require external interconnections)
- **Pulse Width Measurement**
Non-repetitive measurement of high or low pulses with optional level gating
- **Pulse Generation**
Non-repetitive pulse generation
- **Frequency Generation** with optional level gating
Variable duty cycle waveform generation
Square wave generation
Periodic pulse trains

For more information on these functions, see in “Using Task-Oriented Functions” the *Counter/Timer Programming Guide*.

DriverLINX’s hardware-oriented tasks start one or more counter/timers as a group. Applications can program the KPCI-3140 Series-compatible counter/timer chip in specific hardware modes. The counter/timer chip’s modes are

- **0**—Non-retriggerable One-shot
- **1**—Retriggerable One-shot
- **2**—Continuous Increment

Applications can configure one or more counter/timers and interconnect them with the KPCI-3100 circuitry. For more information see in “Configuring a Counter/Timer Channel” the *Counter/Timer Programming Guide*.

DriverLINX then starts the group of counter/timers at the same time Application can then periodically poll for the current count or use interrupt mode to buffer counter readings. For more information see in “Status Polling a Counter/Timer” the *Counter/Timer Programming Guide*.

Counter/Timer Messages

For counter/timer operations, DriverLINX can report the following messages to the application:

DriverLINX Message	Explanation
Service Start	DriverLINX has started the counter/timer task.
Service Done	DriverLINX has completed the counter/timer task.
Critical Error	DriverLINX has encountered an unexpected hardware or software condition.

DriverLINX Event messages for counter/timer tasks.

For detailed explanations of these messages see one of the following references:

- *DriverLINX Technical Reference Manual* for C/C++ users
- *DriverLINX/VB Technical Reference Manual* for VB or Delphi users

Uninstalling DriverLINX

How do I uninstall DriverLINX?

DriverLINX consists of three separate component installations:

- DriverLINX for Keithley KPCI-3100 Series
- DriverLINX Programming Interfaces
- DriverLINX Documentation

You can uninstall the last two installations at any time without interfering with compiled applications that require DriverLINX drivers. To uninstall the latter components, run the “Add/Remove Programs” tool in the Windows Control Panel.

To uninstall DriverLINX drivers for the Keithley KPCI-3100 Series, you must

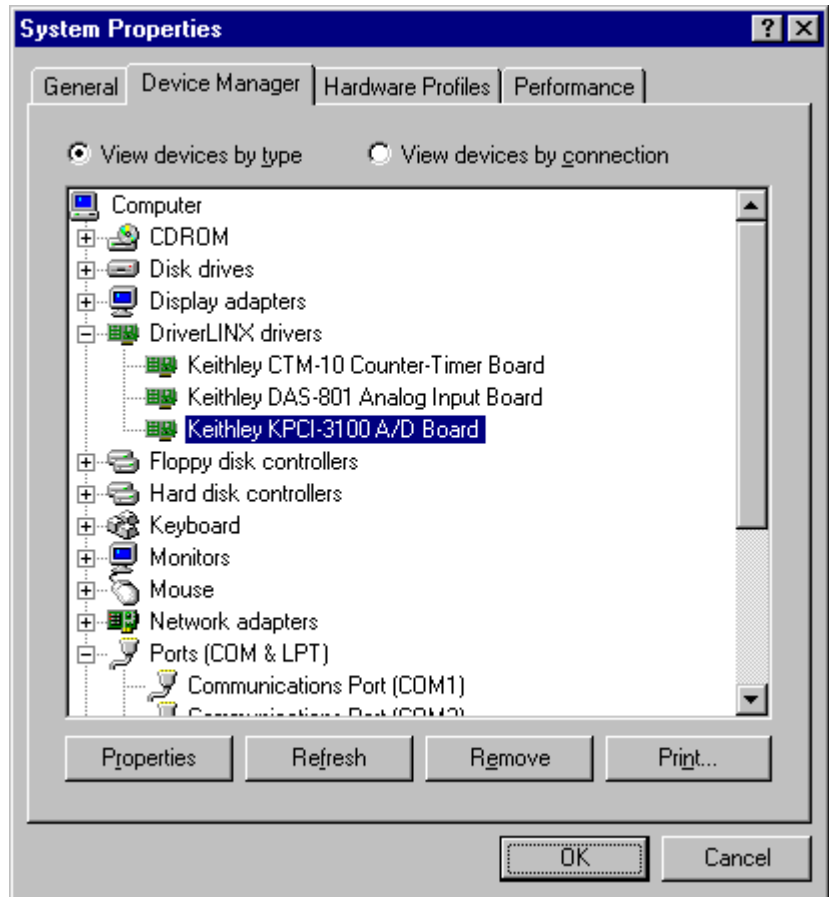
- Disable the DriverLINX driver.
- Shut down your computer to remove the hardware.
- Reboot your computer to unload the driver.
- Run the DriverLINX uninstall program.

How to Disable a DriverLINX Driver in Windows NT

1. From the Windows Start menu, select “Settings”, then “Control Panel”. Left click on the DriverLINX Configuration icon in the Control Panel.
2. Select the KPCI-3100 devices you want to disable.
3. Right click on each device and select “Disabled” on the popup menu.
4. Repeat steps 2-3 for each KPCI-3100 card that you are uninstalling.
5. Close the DriverLINX Configuration Panel.
6. When finished, shut down your computer and physically remove any installed KPCI-3100 hardware.
7. Reboot Windows.
8. To finish uninstalling, see “How to Remove DriverLINX for Keithley KPCI-3100 Series” on page 107.

How to Disable a DriverLINX Driver in Windows 95/98

1. From the Windows Start menu, select “Settings”, then “Control Panel”. Left click on the System icon in the Control Panel. Select the “Device Manager” tab in the System Properties dialog.
2. Left click the “+” icon next to “DriverLINX drivers” to display the installed Keithley KPCI-3100 devices.



3. Select the KPCI-3100 device you want to disable.
4. Click the “Remove” button.
5. In the “Confirm Device Removal” dialog, select “OK”.
6. Repeat steps 3-5 for each KPCI-3100 card or driver that you uninstalling.
7. When finished, click “Close”, shut down your computer, and physically remove any installed KPCI-3100 hardware.
8. Reboot Windows.
9. To finish uninstalling, see “How to Remove DriverLINX for Keithley KPCI-3100 Series” on page 107.

How to Remove DriverLINX for Keithley KPCI-3100 Series

1. From the Windows Start menu, select “Settings”, then “Control Panel”. Left click on the Add/Remove Programs icon in the Control Panel.
2. Select “DriverLINX for Keithley KPCI-3100 Series” in the Add/Remove Programs Properties dialog.
3. Click the “Add/Remove...” button.
4. Answer “Yes” to “Are you sure you want to remove ‘DriverLINX for Keithley KPCI-3100 Series’ and all of its components?” in the Confirm File Deletion dialog.
5. The DriverLINX uninstall program will proceed.

The uninstall program will not remove the folder, “\DrvLINX4\System”. This folder contains copies of any \Windows\System or \Windows\System32 files that the original DriverLINX installation updated.

Troubleshooting

Solving Problems

Correct operation of your KPCI-3100 hardware requires successful completion of three steps.

1. Windows recognizes KPCI-3100 hardware and installs drivers.
2. You configure the KPCI-3100 drivers using the DriverLINX Configuration Panel.
3. Windows loads the KPCI-3100 drivers into memory.

If you are having a problem installing or configuring your KPCI-3100 product, review the following notes. If these notes do not solve your problem, or your problem is not described, then contact technical support and fully describe your problem.

Solving Problems Recognizing and Installing Drivers

Windows must recognize the KPCI-3100 hardware and then install KPCI-3100 drivers for the hardware.

- For Windows to automatically recognize the KPCI-3100 hardware, you must install a KPCI-3100 product into your machine and restart the computer.
- For Windows to install KPCI-3100 drivers, you must install the KPCI-3100 driver software on the DriverLINX CD.

If you installed the KPCI-3100 hardware before you installed the DriverLINX software, Windows 95/98 may install the KPCI-3100 hardware without drivers. To fix this, you must instruct Windows to reinstall the Plug-and-Play driver. Follow the notes below for instructions.

Fixing Problems Installing Hardware Before Software on Windows 95/98

If you installed the KPCI-3100 hardware before installing the DriverLINX software, Windows 95/98 may install the hardware incorrectly.

1. Start Windows Device Manager.

2. Search for “DriverLINX drivers”. If you do not see it, go to step 4.
3. Click on the “+” icon next to “DriverLINX drivers” and search for “PCI Card”. If you find “PCI Card”, go to step 6, otherwise go to the next step.
4. Search for “Other devices”.
5. Click on the “+” icon next to “Other devices” and search for “PCI Card”.
6. Select “PCI Card”.
7. Click “Remove” in the Device Manager.
8. Highlight “Computer” in the Device Manager and click “Refresh”.
9. Windows should re-install your hardware under “DriverLINX drivers”.
10. Answer “Yes” to “Do you want to restart your computer now?”
11. After rebooting your computer, you must configure DriverLINX. See “Configuring the KPCI-3100 Series” on page 11.

Solving Problems Configuring the Drivers

Windows automatically assigns hardware resources for the KPCI-3100 Series, but you must still configure the KPCI-3100 drivers before using them. The DriverLINX configuration only requires that you assign each KPCI-3100 product a unique Logical Device number. And, on Windows NT, you must manually select the hardware model of your KPCI-3100 Series board. See “Configuring the KPCI-3100 Series” on page 11 for more information.

Solving Problems Loading Drivers

Before the KPCI-3100 drivers can load, you must

1. Install the DriverLINX software.
2. Install the KPCI-3100 hardware into your computer.
3. Configure DriverLINX.
4. Reboot your computer.

If you have not completed the above steps, please do so before proceeding.

On both Windows NT and Windows 95/98, the operating system will automatically assign hardware resources to the KPCI-3100 cards. Automatic resource assignment can fail sometimes on

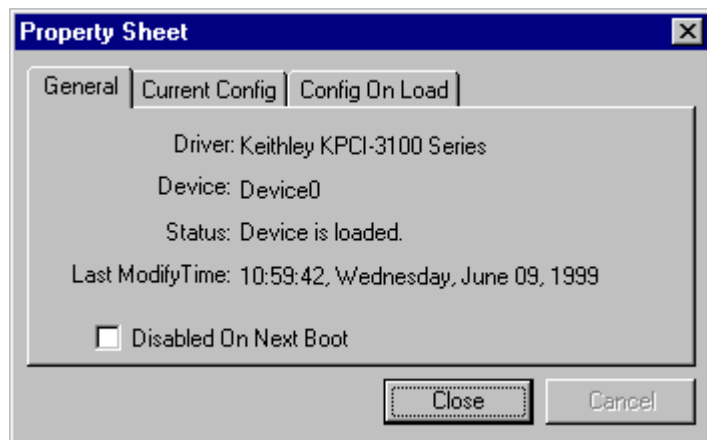
- Older PCI computers.
- Computers with ISA cards installed.
- Computers with no free hardware resources.
- Computers running Windows NT 4.0 with a BIOS setting designating the operating system as Plug-and-Play Aware.

Sorting through all possibilities can be a challenge due to the sheer number of combinations of PCI hardware designs, PC plug-in boards, and versions of Windows. The following sections will help you gather information about why a

driver may have failed to load. This information is essential for you or technical support to solve your problem.

Did the DriverLINX Driver Load?

1. Run “DriverLINX Configuration” from Windows Control Panel.
2. Select the “DriverLINX” tab.
3. Click the “+” icon next to DriverLINX to expand the list of drivers, if necessary.
4. Select “Keithley KPCI-3100 Series”. Click “+”, if necessary, to expand the list.
5. Select the line with the number of the Logical Device you configured. If the number does not exist on Windows NT or, on Windows 95/98, you see only a hardware description, you did not configure the driver. See “Configuring the KPCI-3100 Series” on page 11.
6. Click the “Properties...” button and then select the “General” tab.
7. Do you see “Status: Device Loaded”? If not, did you reboot the computer after configuring? If not, reboot now and repeat the above steps.



8. If you rebooted the computer after configuring and Windows did not load your device, see “Checking for Device Errors” on page 111.

Checking for Device Errors

When a DriverLINX kernel driver cannot load, it usually writes an explanation into the system event log. You can view this log under Windows 95/98 or Windows NT using the DriverLINX Event Viewer.

Windows 95/98 maintains additional driver information in the Device Manager. Also see “Getting More Driver Information on Windows 95/98” on page 112.

1. Run “DriverLINX Event Viewer” from the DriverLINX folder.
2. Click on the “+” icon next to “DriverLINX” in the left panel.
3. Select the abbreviation for your driver.
4. Does the first line in the right panel show a current error?
5. Double click on the error line to see more detail and an explanatory message.

6. If you cannot resolve the problem yourself, please provide this error information when contacting technical support.

Getting More Driver Information on Windows 95/98

Windows 95/98 reports additional information about device status using the Device Manager. To access this utility,

1. Right click on “My Computer” and then select “Properties”.
2. Select “Device Manager” and “View devices by type
3. Does “DriverLINX drivers” appear in the list? If not, see “Solving Problems Recognizing and Installing Drivers” on page 109.
4. Click the “+” next to “DriverLINX drivers”.
5. Does your KPCI-3100 product appear in the list? If not, see “Solving Problems Recognizing and Installing Drivers” on page 109.
6. Does the icon next to your KPCI-3100 product display an exclamation point (!)? If no, Windows has loaded your KPCI-3100 driver.
7. Select the line with the “!” and then click “Properties”.
8. The General tab will show the reason why the driver did not load.
9. The Resources tab will show if Windows detected an unresolvable hardware conflict.

Getting More Driver Information on Windows NT

On Windows NT, the only reasons that a driver does not load are

- You did not install the driver software.
- You did not correctly configure the driver.
- You changed the driver startup parameters.
- Your BIOS setting designates the operating system as Plug-and-Play aware. (Applies only to Window NT 4.0).

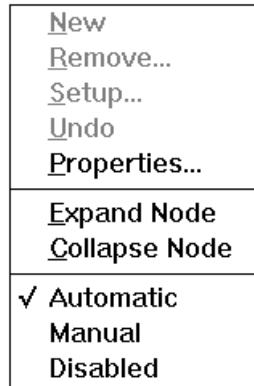
An incorrectly configured driver will report the reasons that it failed to load into the Windows Event Log. See “Checking for Device Errors” on page 111 for more information.

Under Windows NT 4.0, a Logical Device may not load because the operating system does not always configure Plug-and-Play PCI devices properly. This may result in unexpected or no Event Log entries. To work around this, set your computer’s BIOS to configure Plug-and-Play devices before it starts the operating system. On various computers the BIOS setting is called “Plug-and-Play Aware OS – Disabled” or “Plug & Play OS – No”.

On Windows NT, DriverLINX drivers load automatically during system boot. An administrator can change the startup command for any NT driver to either “manual” or “disabled”.

1. Run “DriverLINX Configuration” from Windows Control Panel.
2. Select the “DriverLINX” tab.
3. Click the “+” icon next to DriverLINX to expand the list of drivers, if necessary.

4. Select “Keithley KPCI-3100 Series”. Click “+”, if necessary, to expand the list.
5. Select the line with the number of the Logical Device that did not load.
6. Right click the mouse to see a popup menu.



7. Select “Automatic” to instruct Windows to load the driver the next time you reboot.

Generating a DriverLINX Configuration Report

Your DriverLINX installation includes a troubleshooting tool that generates a report of your DriverLINX configuration. If you call Technical Support, after reading “Solving Problems” on page 109, they may ask you to generate and e-mail this report to help you solve installation and configuration problems.

What is in the Report?

The troubleshooting tool analyzes your computer to obtain information about DriverLINX and operating system software that would assist Technical Support in troubleshooting a problem you are having. It includes information on DriverLINX files, environment variables, registry entries, hardware and the operating system.

How do I Generate the Report?

You can easily generate the report by clicking the shortcut in this topic in the help file. Once the troubleshooting tool generates the report, you will have the opportunity to review it and make deletions, if desired, before e-mailing it to Technical Support. If you do not have direct access to e-mail, you can save the report to a disk file and send a copy later. A Technical Support engineer will guide you through these steps when you are asked to send a report.

Glossary of Terms

A/D

Abbreviation for Analog-to-Digital, a process that converts a continuous analog signal into a discrete digital approximation of the analog signal.

ADC

Abbreviation for Analog-to-Digital Converter, the hardware that performs the A/D conversion.

API

Abbreviation for Application Programming Interface. An API defines the syntax of the data structures and functions of software services.

Buffer

A block of memory used to receive data from a data-acquisition device or to write data to a data-acquisition device.

Bus mastering

Bus mastering is a mode that PCI devices use to perform DMA data transfers. Bus mastering requires a compatible PCI device and motherboard. The KPCI-3100 Series and most motherboards support bus mastering.

Clocking

A periodic pulse or signal that data-acquisition hardware uses to read or write the next sample or block of samples. Also referred to as “pacing”.

D/A

Abbreviation for digital-to-analog, a process that converts a discrete digital value into a continuous analog voltage representing that value.

DAC

Abbreviation for digital-to-analog converter, the hardware that performs the D/A conversion process.

DMA

Abbreviation for Direct Memory Access, a technique where the system board can transfer data between a device and memory without using the CPU. In the PC, a standard chip on the system board controls the transfer.

Event

For DriverLINX, an event is the occurrence of a signal that clocks, starts, or stops a data-acquisition task.

In an ActiveX control, an event is a procedure in the client application called by the control.

Gating

A signal that enables and disables another signal or data-acquisition task depending on the value of the gate signal.

IRQ

Abbreviation for interrupt request. Peripheral hardware signals the CPU that it is ready to transfer data.

ISA

Abbreviation for Industry Standard Architecture. A standard for the original IBM AT bus specification that defines the bus structure, CPU and support chip architecture, and the clock frequency of the ISA bus.

ISR

Abbreviation for interrupt service routine, the software function inside a device driver that handles interrupt requests.

Logical Device

DriverLINX's designation for a specific data-acquisition board inside your computer.

Messages

In Windows and DriverLINX, a message notifies the application about the state of a process. In DriverLINX's ActiveX controls, DriverLINXSR and DriverLINXLDD, messages fire a control event.

Modes

DriverLINX data-acquisition techniques.

Operations

Allowed DriverLINX data-acquisition commands.

Pacing

A periodic pulse or signal that data-acquisition hardware uses to read or write the next sample or block of samples. Also referred to as “clocking”.

Process

Refers to the collection of data and code segments and hardware resources that the operating system assigns to one application.

Scan List

The channels sampled or written by a task, whether specified using a range or a list.

Service Request

A DriverLINX object or data structure that completely defines a data-acquisition task.

Single-scan

A task that samples once from each channel in the scan list. Such a task requires a buffer that holds exactly one scan.

Subsystem

DriverLINX subdivides a general-purpose data-acquisition device into six subsystems—Device, Analog Input, Analog Output, Digital Input, Digital Output, and Counter/Timer.

Triggering

The technique of using a pulse or signal to start or stop a data-acquisition task.

TTL

Abbreviation for transistor-transistor logic, a family of digital logic elements.

