Keithley KPCMCIA AI/AO Series

Using DriverLINX with Your Hardware

KEITHLEY

Keithley KPCMCIA AI/AO Series: Using DriverLINX with Your Hardware

SST 13-1101-1

Contents

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 card 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
- Learn DriverLINX, an interactive learning and demonstration program for DriverLINX that includes a Digital Storage Oscilloscope
- Source code for the sample programs
- The DriverLINX Application Programming Interface files for your compiler
- DriverLINX On-line Help System
- *DriverLINX Analog I/O 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 KPCMCIA AI/AO Series cards with DriverLINX.

- For help installing and configuring your hardware and DriverLINX, please see the manual that accompanied your hardware.
- 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 KPCMCIA AI/AO Series

Shows how to configure the KPCMCIA AI/AO Series using the *Configure DriverLINX Device* dialog box.

Using the KPCMCIA AI/AO Series with DriverLINX

Shows how to set up DriverLINX with the *Edit Service Request* dialog box to use KPCMCIA AI/AO Series hardware features.

Conventions Used in This Manual

The following notational conventions are used in this manual:

- Itemized lists are identified by a round bullet (•).
- 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
```

- Tables of information are bounded by a box with a double line.

Tables

Concept

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

Configuring the KPCMCIA AI/AO Series

Introduction

This manual explains the steps and special features that apply to installing and configuring Keithley's KPCMCIA AI/AO Series cards.

Installing and configuring DriverLINX for the Keithley KPCMCIA AI/AO Series cards requires three steps:

1. **To install your KPCMCIA AI/AO hardware**, read and follow the instructions in the hardware manual.
2. **To install DriverLINX**, follow the general procedure outlined in the "Read Me First" material on the installation CD.
3. **To configure DriverLINX**, use the *DriverLINX Configuration Panel*. Also see "Configure DriverLINX Device Dialog" on page 9 for configuration options specific to a Keithley KPCMCIA AI/AO Series model.

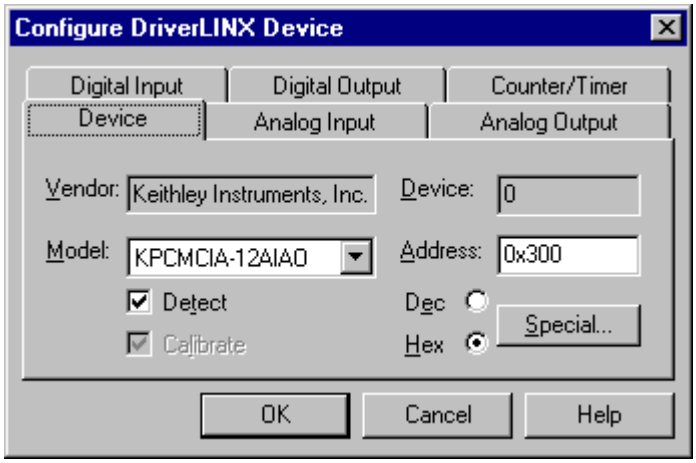
Configure DriverLINX Device Dialog

DriverLINX uses a standardized configuration protocol for all data-acquisition hardware. Even though Windows 95/98/Me automatically selects the hardware base address and interrupt channel for the KPCMCIA AI/AO Series, you must still follow the configuration process to assign a DriverLINX Logical Device number to a specific KPCMCIA AI/AO Series model in your computer.

When you activate the *Setup…* button in the *DriverLINX Configuration Panel*, DriverLINX displays the *Configure DriverLINX Device* dialog. The following sections describe your choices for configuring DriverLINX to work with a Keithley KPCMCIA AI/AO Series model.


Control Panel
File Edit View Help
Accessibility Options
Add New Hardware
Add/Remove Programs
Date/Time
Display
DriverLINX Configuration
Fonts
Internet
Keyboard
Modems
Mouse
Multimedia
Network
Passwords
PC Card (PCMCIA)
Power
Printers
Regional Settings
Sounds
System
20 object(s)

Device Subsystem Page



Note: The Configure DriverLINX Device dialog appears differently under different versions of Windows.

Use the Device subsystem page to tell DriverLINX the model name, address and, optionally, the expansion accessories connected to your KPCMCIA AI/AO Series card.

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 the hardware model of the card you're configuring.

All cards in this series support differential analog input with half the number of single-ended channels. The AIAO cards support two 12-bit analog output channels. All cards support four digital input and four digital output channels.

Windows 95/98/Me/2000

For Windows 95/98/Me/2000, *Model* is a read-only property—DriverLINX selects the next unconfigured card. To configure a different card, first save (**OK**) the current configuration. Then insert the appropriate card and select or create a new Logical Device using the *DriverLINX Configuration Panel.*

Windows NT

For Windows NT, select one of the following models:

Model	Resolution	AI Channels	AO Channels
KPCMCIA-12AI	12 bits	16	
KPCMCIA-12AIH	12 bits	16	
KPCMCIA-12AIAO	12 bits	8	2
KPCMCIA-12AIAOH	12 bits	8	2
KPCMCIA-16AI	16 bits	16	
KPCMCIA-16AIAO	16 bits	8	2

Address

Windows 95/98/Me/2000

Windows automatically selects an appropriate address and ignores this property.

Windows NT

The *Address* property selects the I/O port address for the card. The default address used by DriverLINX is 768 decimal or 0x300 hex. If you have another peripheral card at the same address, you will have to select a free range of 8 or 16 addresses (depending on the model).

Detect

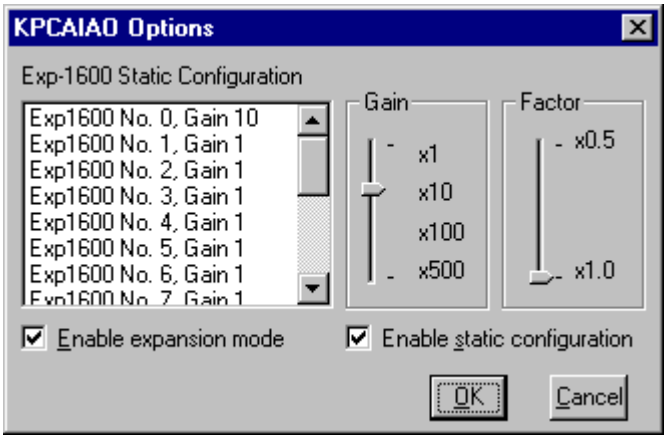
The *Detect* property enables and disables DriverLINX's hardware detection and testing algorithms. For maximum system reliability, always leave this check box marked.

Calibrate

The *Calibrate* property enables and disables hardware auto-calibration. For best accuracy, always leave this check box marked.

Special…

The *Special…* button displays the following dialog box of KPCMCIA AI/AO-specific configuration options:



This dialog allows you to configure an EXP-1600 multiplexer for use with the KPCMCIA's analog input channels.

Note: Using an EXP-1600 multiplexer requires configuring the base channels as single-ended.

- **Enable expansion mode**

Enable expansion mode to allow the KPCMCIA hardware to use an EXP-1600 multiplexer. Note: you can disable expansion mode without losing existing gain settings.

- **Enable static configuration**

Static configuration allows you to record the gain selections for each EXP-1600 attached to an analog input channel. DriverLINX can then use this information to correctly convert A/D codes to volts. With static configuration disabled, you must perform gain correction in your application. Whereas, with static configuration enabled, DriverLINX will check that you have specified a valid total gain for each channel in your service request and apply the total gain when converting data for your application.

For example, if the card's base gain settings are 1,2,4,8 and an attached multiplexer has gain of 100, then the valid selections for total gain on the corresponding channels are 100,200,400,800

- **EXP-1600 Static Configuration**

DriverLINX records gain selections for an EXP-1600 attached to each base channel. Select an EXP-1600 in the list to change its gain using the *Gain* and *Factor* sliders.

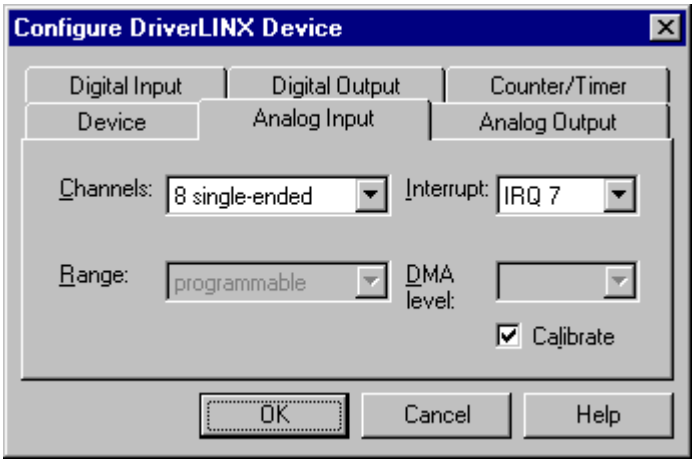
- **Gain**

For the highlighted EXP-1600 in the list, record the setting of the *Gain-set* DIP switch S4.

- **Factor**

For the highlighted EXP-1600 in the list, record the position of the *Gain-multiplier* slide switch S2.

Analog Input Subsystem Page



Use the Analog Input subsystem page to tell DriverLINX if you'll be using single-ended or differential input connections.

Channels

The *Channels* property allows you to select either single-ended or differential analog input connections.

- For 16 channel models, select either "16 single-ended" or "8 differential" channels.
- For 8 channel models, select either "8 single-ended" or "4 differential" channels.

Range

The full-scale maximum analog input range for the KPCMCIA AI/AO Series is fully software programmable. DriverLINX disables this property.

Interrupt

Windows 95/98/Me/2000

For Windows 95/98/Me/2000, the operating system automatically determines the interrupt channel for the KPCMCIA AI/AO Series card. DriverLINX disables this property.

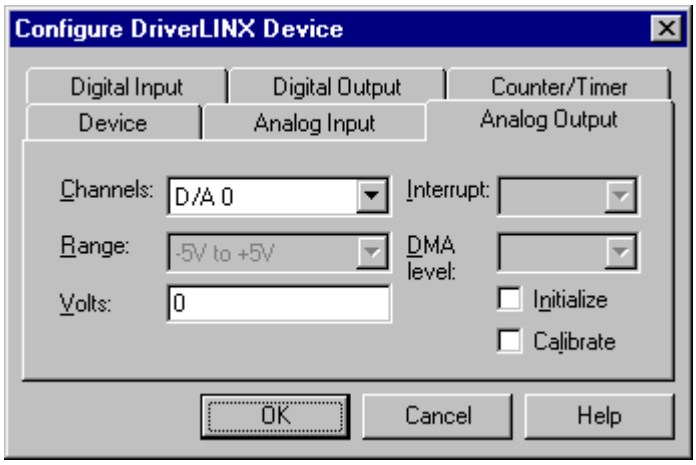
Windows NT

For Windows NT, select a free interrupt request level to support interrupt mode transfers. Valid IRQ levels are: 1 - 15.

DMA level

The KPCMCIA AI/AO Series does not use system DMA channels. DriverLINX disables this property.

Analog Output Subsystem Page



Use the Analog Output subsystem page to change the default D/A initialization voltages.

Channels

The *Channels* property allows you to select either D/A Logical Channel (0 or 1) for individual configuration of its initialization voltage.

Range

The full-scale analog output range for the KPCMCIA AI/AO Series is fixed at ±5 volts. DriverLINX disables this property.

Interrupt

The KPCMCIA AI/AO Series uses the same interrupt for analog output as for analog input. Go to the Analog Input page to set it. DriverLINX disables this property and displays it as blank.

DMA level

The KPCMCIA AI/AO Series does not use system DMA channels. DriverLINX disables this property.

Volts

The *Volts* property allows you to specify a custom output voltage for each DAC that DriverLINX uses when it initializes the hardware. DriverLINX's default initialization value is zero volts. DriverLINX ignores this property unless you also check the *Initialize* property.

Initialize

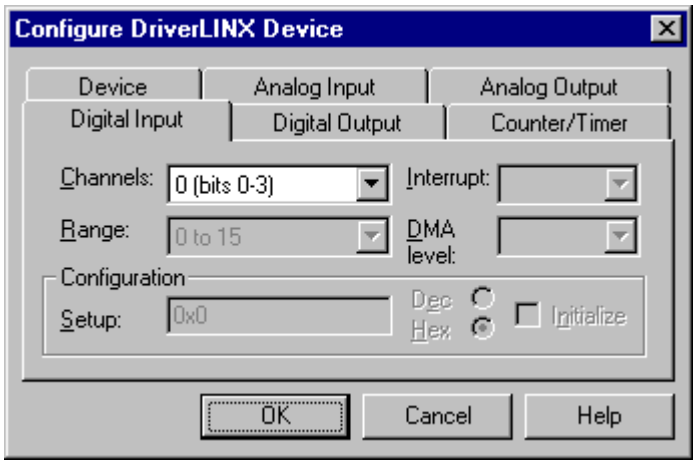
Checking the *Initialize* check box instructs DriverLINX to use the *Volts* property, rather than the default value, for analog output initialization. The KPCMCIA AI/AO Series does not support this feature.

Calibrate

The *Calibrate* property enables and disables hardware auto-calibration. DriverLINX disables this property as the KPCMCIA AI/AO Series DACs don't support auto-calibration.

Digital Input Subsystem Page

For the KPCMCIA AI/AO Series, there are no configurable options for the Digital Input subsystem.



Channels

The *Channels* property allows you to select a Logical Channel for configuration or viewing the channel's range. The KPCMCIA AI/AO Series digital input channels have fixed configurations.

DriverLINX defines the following Logical Channels for the KPCMCIA AI/AO Series digital inputs:

Logical Channel	DriverLINX Function	KPCMCIA AI/AO Series External Connector
0	Standard Digital Input	Digital input lines (DI 0 … DI 3)
1	External Clock	DI 2 / ExtClk
2	External Trigger, External Clock	DI 0 / Ext. Trigger

Range

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

Interrupt

The KPCMCIA AI/AO Series uses the same interrupt for digital input as for analog input. Go to the Analog Input page to set it. DriverLINX disables this property and displays it as blank.

DMA level

The KPCMCIA AI/AO Series does not use system DMA channels. DriverLINX disables this property and displays it as blank.

Configuration Setup

The *Configuration Setup* property specifies the hardware configuration of the digital I/O ports. The KPCMCIA AI/AO Series has a fixed digital I/O configuration. Therefore, DriverLINX disables this field.

Initialize

Checking the *Initialize* check box instructs DriverLINX to use the *Configuration Setup* property to configure the digital I/O ports. The KPCMCIA AI/AO Series has a fixed digital I/O configuration. Therefore, DriverLINX disables this field.

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. KPCMCIA AI/AO Series cards only have a single digital output channel.

Range

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

Interrupt

The KPCMCIA AI/AO Series uses the same interrupt for digital output as for analog input. Go to the Analog Input page to set it. DriverLINX disables this property and displays it as blank.

DMA level

The KPCMCIA AI/AO Series does not use system DMA channels. DriverLINX disables this property and displays it as blank.

Initialization Value

The *Initialization Value* property specifies the digital output value DriverLINX will write to the selected Logical Channel on hardware initialization. DriverLINX only writes this value if you enable the *Initialize* check box. By default, DriverLINX uses the hardware-defined initialization values if the *Initialize* check box is not checked. For the KPCMCIA AI/AO Series, the default digital output value is zero.

Initialize

Checking the *Initialize* check box instructs DriverLINX to use the *Initialization Value* property, rather than the default value, for digital output port initialization.

Dec

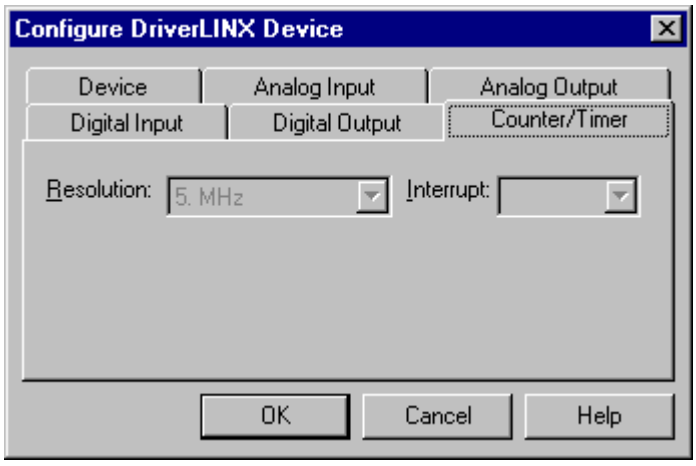
This check box converts the *Initialization Value* property to decimal.

Hex

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

Counter/Timer Subsystem Page

For the KPCMCIA AI/AO Series, there are no configurable options on the Counter/Timer subsystem page.



Resolution

The *Resolution* property specifies the clock frequency of the master oscillator. All models have a 5 MHz clock source for pacing analog input. Models 12AIAO, 12AIAOH and 16AIAO also have a 1 MHz clock source for pacing analog output.

Interrupt

The KPCMCIA AI/AO Series does not support interrupts from counter/timers. DriverLINX disables this property and displays it as blank.

Using the KPCMCIA AI/AO Series with DriverLINX

Introduction

This chapter shows you how to set up and use KPCMCIA AI/AO Series hardware features with DriverLINX. The descriptions here use the *Edit Service Request* dialogs 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.

DriverLINX Hardware Model for KPCMCIA AI/AO Series

By design 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 KPCMCIA AI/AO Series hardware capabilities.

The following sections describe how DriverLINX implements KPCMCIA AI/AO Series hardware features as Subsystems, Modes, Operations, Events, Logical Channels, Buffers, and Messages.

DriverLINX Subsystems

The KPCMCIA AI/AO Series supports all six of DriverLINX's subsystems:

1. **Device**—refers to a KPCMCIA AI/AO model as a whole.
2. **Analog Input**—refers to the analog input channels, clocks, and control signals, such as Ext. Trigger, ExtClk, etc.
3. **Analog Output**—refers to the analog output channels, clocks, and control signals.
4. **Digital Input**—refers to the 4-bit digital input port as well as 1-bit digital input (TTL) control signals, such as Ext. Trigger, etc.

5. **Digital Output**—refers to the 4-bit digital output port.
6. **Counter/Timer**—refers to the Analog Input and Analog Output subsystem-specific internal clock channels as well as a software-implemented system timer.

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 doesn't 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 three of the four modes with the KPCMCIA AI/AO 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 card 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 card using hardware interrupts and programmed I/O transfers.
- **DMA Mode**—This is a background or asynchronous operation. DriverLINX transfers data between the computer's memory and the data-acquisition card using memory bus transfers. The KPCMCIA AI/AO Series does not support this transfer mode.
- **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 KPCMCIA AI/AO Series.

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

KPCMCIA AI/AO Series Supported DriverLINX Modes

*Models KPCMCIA-12AIAO, 12AIAOH and 16AIAO only.

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 KPCMCIA AI/AO Series. Latter sections for each DriverLINX subsystem will describe the operations and events in more detail.

Note: All subsystems allow the *MESSAGE* operation and the Analog I/O subsystems allow the *CONVERT* operation, which are not shown in the table. DriverLINX allows any Mode setting for these operations.

Subsystem	Operation	Events		
Mode		Timing	Start	Stop
Analog Input				
Polled	Start	null, rate	null, cmd	null, TC
Interrupt	Start, Stop, Status	rate, dig	cmd, dig, ana*	cmd, TC, ana*
Other	Initialize			
Analog Output*				
Polled	Start	null	null, cmd	null, TC
Interrupt	Start, Stop, Status	rate	cmd	cmd, TC
Other	Initialize			
Digital Input				
Polled	Start	null	null, cmd	null, TC
Interrupt	Start, Stop, Status	rate	cmd	cmd, TC
Other	Initialize			
Digital Output				
Polled	Start	null	null, cmd	null, TC
Interrupt	Start, Stop, Status	rate	cmd	cmd, TC
Other	Initialize			
Counter/Timer				
Polled*	Start	rate	null, cmd	null, TC
Other	Initialize			
Device				
Other	Initialize, Configure, Capabilities			

*Models KPCMCIA-12AIAO, 12AIAOH and 16AIAO only.

Allowed Operations and Events for KPCMCIA AI/AO Series Subsystems and Modes

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

- **null**—Null or None Event when a Service Request doesn't require an event
- **cmd**—Command Event when DriverLINX starts or stops a task on software command
- **TC**—Terminal Count Event 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 an analog input signal to pace, start, or stop a task

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 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 how DriverLINX assigns Logical Channel numbers, see the notes for 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 observe 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 KPCMCIA AI/AO Series

The Keithley hardware manual describes the data and control signals for the KPCMCIA AI/AO Series and the connector pinouts for these signals. This section summarizes how DriverLINX logically numbers the I/O data signals and how DriverLINX uses several of these control signals for external clock, trigger, and gating inputs.

Analog Input Subsystem Signals

The Analog Input subsystem has 4, 8 or 16 analog input single-ended or differential signal connections depending on the model and configuration of your KPCMCIA AI/AO card. DriverLINX maps these signals to Logical Channels as shown in the following table:

Number of A/D Channels	Connector Name	Logical Channels
4 Differential	CH0+/CH0- – CH3+/CH3-	0–3
8 Differential	CH0+/CH0- – CH7+/CH7-	0–7
8 Single-ended	CH0 – CH7	0–7
16 Single-ended	CH0 – CH15	0–15

How DriverLINX maps analog input hardware channels to Logical Channels

The Analog Input subsystem supports a dedicated internal pacer clock that DriverLINX designates as Logical Channel 0 of the Counter/Timer subsystem.

The Analog Input subsystem also has several control signals that DriverLINX uses as external clocks, gates, and triggers as shown in the following table:

Connector Name	DriverLINX Usage
DI 0 / Ext. Trigger	External trigger / External pacer clock
DI 2 / ExtClk	External pacer clock

How DriverLINX uses analog input control signals

Analog Output Subsystem Signals

The KPCMCIA-12AIAO, 12AIAOH, and 16AIAO models have two 12-bit analog output DACs. DriverLINX maps these signals to Logical Channels as shown in the following table:

Number of D/A Channels	Connector Name	Logical Channels
2	D/A CH0 – D/A CH1	0 – 1

How DriverLINX maps analog output hardware channels to Logical Channels

The Analog Output subsystem supports a dedicated internal pacer clock that DriverLINX designates as Logical Channel 1 of the Counter/Timer subsystem.

Digital Input Subsystem Signals

The Digital Input subsystem has one 4-bit digital input port and several control inputs which DriverLINX models as 1-bit logical digital input ports. DriverLINX maps these signals to Logical Channels as shown in the following table:

Port	Connector Name	Logical Channels
4-bit digital input	DI 0 – DI 3	0
external clock	DI 2 / ExtClk	1
external trigger/ external clock	DI 0 / Ext. Trigger	2

How DriverLINX maps digital input hardware channels to Logical Channels

The Digital Input subsystem doesn't support a dedicated internal pacer clock, but DriverLINX uses a system timer clock for low-frequency digital input pacing. DriverLINX designates the system clock as Logical Channel 2 of the Counter/Timer subsystem.

Digital Output Subsystem Signals

The Digital Output subsystem has one 4-bit digital output port. DriverLINX maps these signals to Logical Channels as shown in the following table:

Port	Connector Name	Logical Channels
4-bit digital output	DO 0 – DO 3	0

How DriverLINX maps digital output hardware channels to Logical Channels

The Digital Output subsystem doesn't support a dedicated internal pacer clock, but DriverLINX uses a system timer clock for low-frequency digital output pacing. DriverLINX designates the system clock as Logical Channel 2 of the Counter/Timer subsystem.

Counter/Timer Subsystem Signals

The Counter/Timer subsystem has several internal hardware and system timers. All models in the KPCMCIA AI/AO Series have a 24-bit timer for analog input. Models KPCMCIA-12AIAO, 12AIAOH and 16AIAO also have a 16-bit timer for analog output or event counting tasks. DriverLINX provides a system timer using software for digital input/output. DriverLINX maps these timers to Logical Channels as shown in the following table:

Timer	Connector Name	Logical Channels
Analog input timer	DI 0 / Ext. Trigger, DI 2 / ExtClk, ExtGate	0
Analog output timer	DI 0 / Ext. Trigger, DI 2 / ExtClk, ExtGate	1
System timer	none	2

How DriverLINX maps counter/timer hardware channels to Logical Channels

Applications generally cannot use these timers independently from their associated analog input/output subsystem. On AIAO models only, the analog output timer can be used for simple 16-bit continuous counting, when not in use for pace analog output. See “Counter/Timer Subsystem” on page 68 for details.

Device Subsystem

The following sections describe how DriverLINX implements Device Subsystem features for the KPCMCIA AI/AO Series.

Device Modes

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

Device Operations

The KPCMCIA AI/AO Series Device Subsystem supports the following DriverLINX operations:

If another process is using the same data-acquisition card, DriverLINX will prevent Device Initialization from interfering with another process’s data-acquisition tasks.

- **Initialize**—DriverLINX aborts all data-acquisition tasks for every subsystem controlled by the current process. DriverLINX then performs an initialization for each supported subsystem.
- **Configure**—DriverLINX displays the *Configure DriverLINX Device* dialog for the current Logical Device. Please use the *DriverLINX Configuration Panel* rather than this operation to configure DriverLINX.
- **Capabilities**—DriverLINX provides hardware-specific and configuration information in the form of a Logical Device Descriptor database.

Analog Input Subsystem

The following sections describe how DriverLINX implements Analog Input Subsystem features for the KPCMCIA AI/AO 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.
- **Other**—For subsystem initialization and data conversion.

Analog Input Operations

The KPCMCIA AI/AO Series Analog Input Subsystem supports the following DriverLINX operations:

- **Initialize**—aborts any active DMA or interrupt data-acquisition tasks and stops the clock. DriverLINX prevents one application process from interfering with another process'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**—DriverLINX 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 KPCMCIA AI/AO Series acquires the next analog input sample.

The KPCMCIA AI/AO Series supports the following Timing Events:

- **None**—Sampling requires no pacing as DriverLINX is acquiring only a single value.
- **Rate**—The KPCMCIA AI/AO Series supports both fixed rate sampling and burst rate sampling using internal and external clocks.
- **Digital**—DriverLINX uses an external digital input signal to pace the acquisition of the next sample.

None or Null Event

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

Rate Event

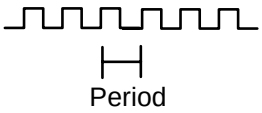
The KPCMCIA AI/AO Series supports two types of Rate Events for analog input:

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

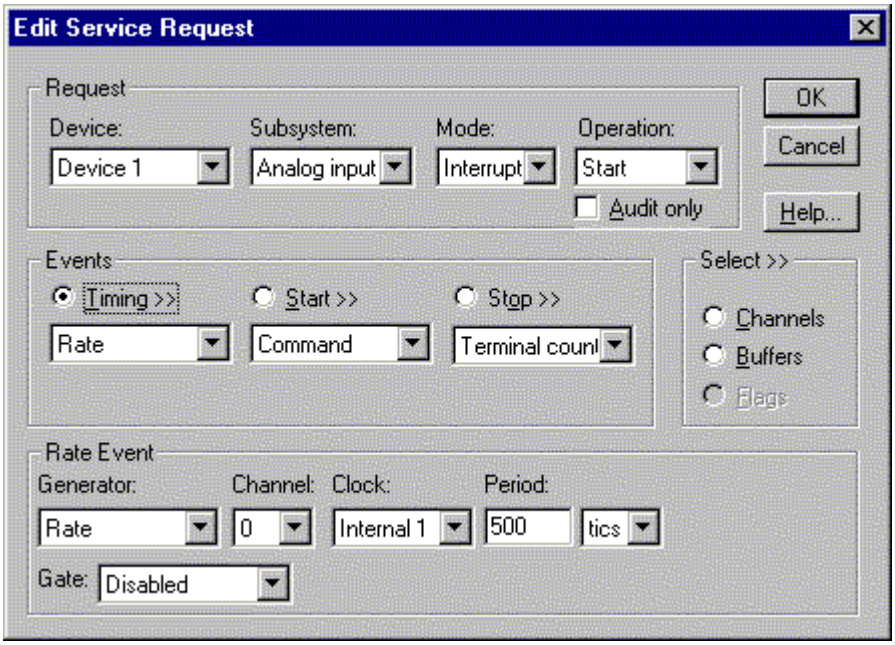
- **Burst Generator**—Generates a dual frequency clock with a fixed number of tics at a high frequency separated by a time interval at a lower frequency.

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.



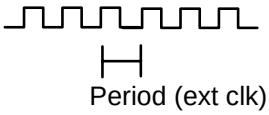
How to set up the KPCMCIA AI/AO Series for fixed rate sampling using an internal clock.

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

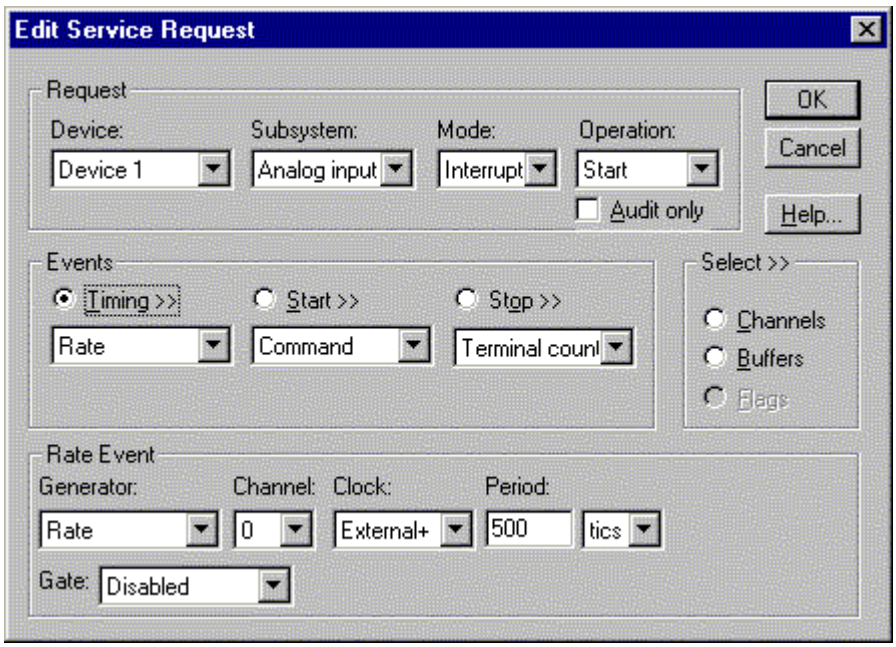
- Specify internal clocking using a **Rate** *Generator* on *Logical Channel* **0** with the **Internal 1, Internal 2** or **Internal 3** *Clock* source. See "Counter/Timer Subsystem" on page 68 for a description of clock sources.
- The *Period* property specifies the time interval between samples in tics, where an **Internal 1** tic is 0.2 µs, or 1 / 5 MHz. The minimum period is 50 tics, or 100 kHz. The maximum period is 16,777,215 tics (2^{24} -1), or 0.006 Hz.
- In the *Channels* section, check the *Simultaneous* box to sample all specified channels in each period or leave it unchecked to sample one channel per period. When checked, the hardware samples the channels simultaneously or as fast as possible. You would typically check *Simultaneous* when using a Simultaneous Sample and Hold accessory.

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'll need a separate external clock tic for each analog sample you want to acquire.



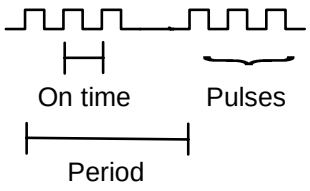
How to set up the KPCMCIA AI/AO Series for fixed rate sampling using an external clock.

***BE SURE** that the external clock source is TTL compatible, 0 V minimum to +5 V maximum!*

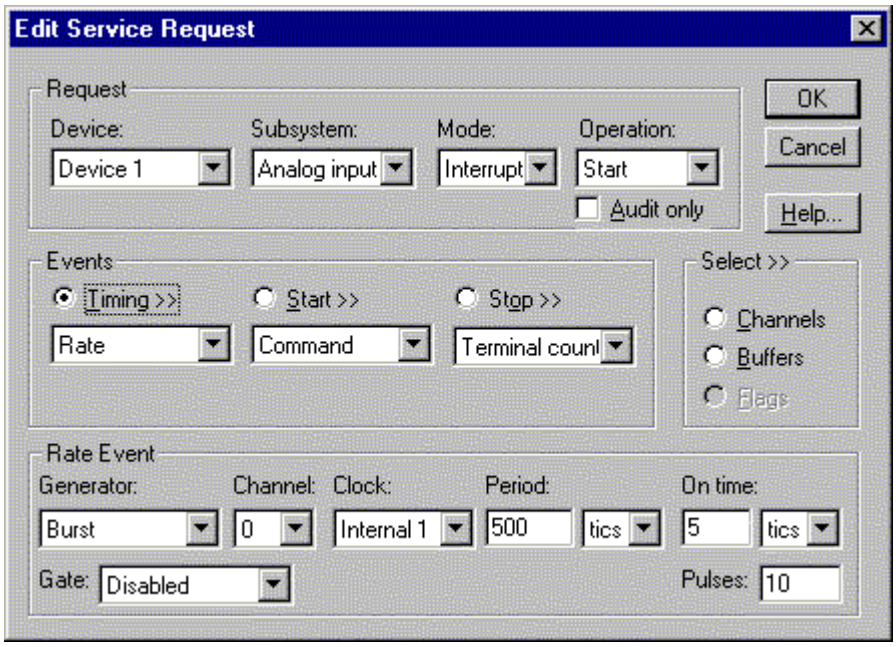
- Specify external clocking using a **Rate** *Generator* on Logical *Channel* **0** with an **External**, **External+,** or **External-** *Clock* source. See "Counter/Timer Subsystem" on page 68 for a description of clock sources.
- Users should connect the external clock signal to the DI 0 / Ext. Trigger line.
- The *Period* may be any value as long as it is ≥ 50 tics, or 10 µs. The period value doesn't 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 5 MHz.
- In the *Channels* section, check the *Simultaneous* box to sample all specified channels in each period. The hardware samples the channels simultaneously or as fast as possible. The hardware only supports sampling all specified channels per external clocking pulse. Therefore, DriverLINX assumes you want *Simultaneous* even if you do not check the box.

Burst Generator: Internal Clocking

An internally clocked Burst Generator produces a dual frequency clock with a fixed number of tics at a high frequency separated by a time interval at a lower frequency.



Use an internally clocked burst generator when you want to minimize the sampling interval between successive channels in the channel list, but you want a much longer per channel sampling interval. For many applications, burst mode sampling eliminates the need for a simultaneous sample and hold accessory.



How to set up the KPCMCIA AI/AO Series for burst rate sampling using an internal clock.

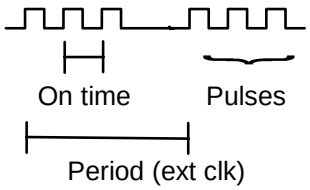
For hardware independence, specify the clock channel using the symbolic constant, DEFAULTTIMER, which always maps to the default Logical Channel for analog input timing.

- Specify internal clocking using a **Burst** *Generator* on Logical *Channel* **0** with the **Internal 1, Internal 2** or **Internal 3** *Clock* source. See "Counter/Timer Subsystem" on page 68 for a description of clock sources.
- The *Period* property (major interval) specifies the time interval between bursts in tics, where an **Internal 1** tic is 0.2 µs, or 1 / 5 MHz.
- The *On time* property (minor interval) specifies the time interval between samples in tics. The KPCMCIA AI/AO Series allows only three fixed minor periods: 10, 20 or 40 µs. If the application specifies an unsupported minor period, DriverLINX substitutes the next shortest period.
- The *Pulses* property specifies the number of clock pulses in a single burst. The KPCMCIA AI/AO Series requires that the number of pulses equals the number of channels in the channel-gain list.
- The minimum *Period* or *On time* value is 50 tics, or 100 kHz. The maximum *Period* or *On time* value is 16,777,215 tics (2^{24} -1), or 0.006 Hz. The *Period* value must be greater than the *On time* value times the number of *Pulses*.

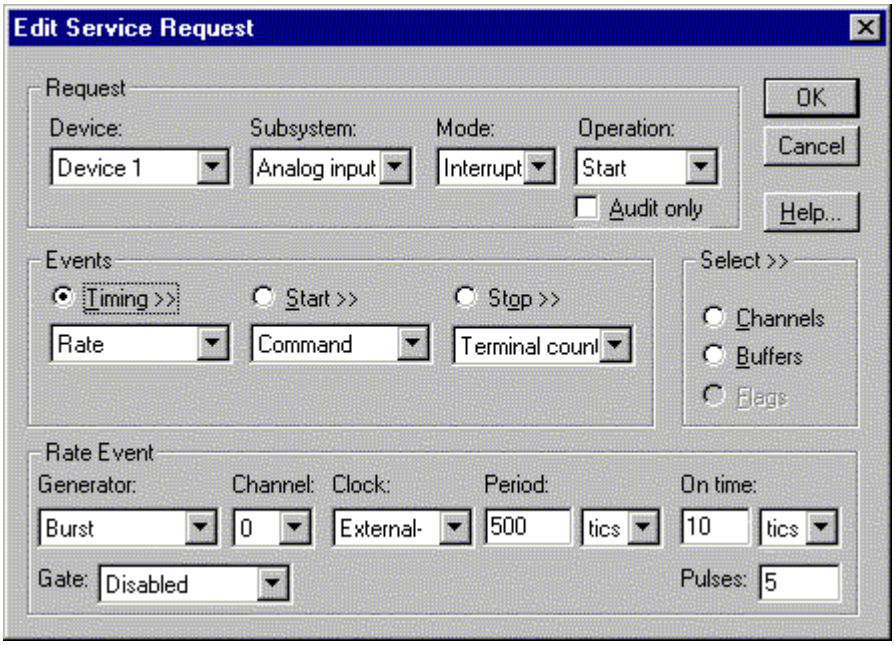
Burst Generator: External Clocking

An externally clocked Burst Generator produces a dual frequency clock with a fixed number of tics at a high frequency separated by a time interval at a lower frequency. The internal clock determines the number of, and interval between, pulses within a

burst while the external clock initiates each burst.



Use an externally clocked burst generator when you want to synchronize the scan of all channels in the channel-gain list with a recurrent external signal. In this mode you'll need a separate external clock tic for each scan of the channel list.



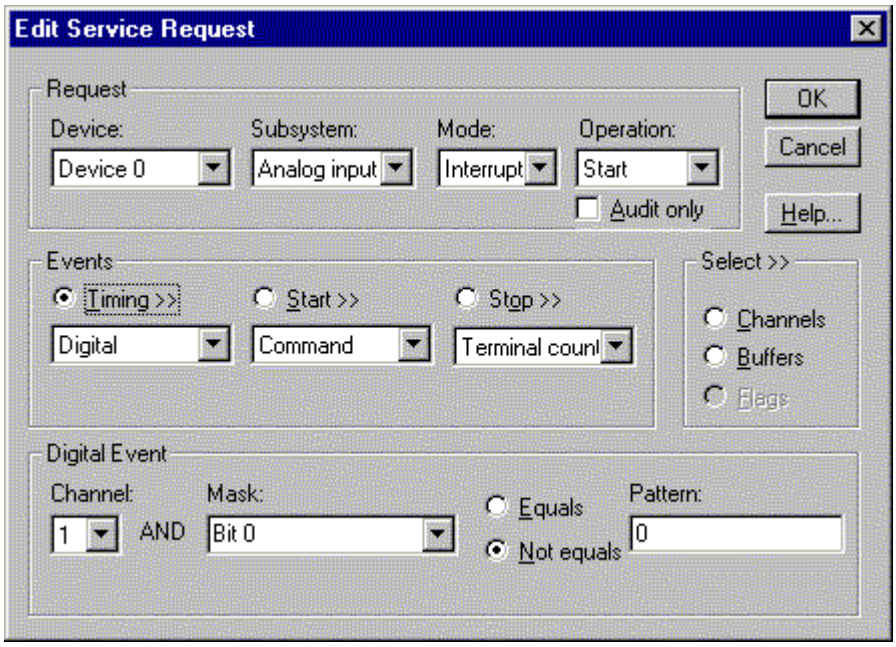
How to set up the KPCMCIA AI/AO Series for burst rate sampling using an external clock.

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

- Specify external clocking using a **Burst** *Generator* on Logical *Channel* **0** with an **External**, **External+**, or **External-** *Clock* source.
- The *Period* property (major interval) specifies the time interval between bursts in tics. It may be any value that would be valid for internal clocking. The *Period* value doesn't 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 *On time* property (minor interval) specifies the time interval between samples in tics. The KPCMCIA AI/AO Series allows only three fixed minor periods: 10, 20 or 40 µs. If the application specifies an unsupported minor period, DriverLINX substitutes the next shortest period.
- The *Pulses* property specifies the number of clock pulses in a single burst. The KPCMCIA AI/AO Series requires that the number of pulses equals the number of channels in the channel-gain list.
- Users should connect the external clock signal to the DI 2 / ExtClk.

Digital Event

DriverLINX supports Digital Events as aliases for externally clocked Rate Generators. Use this technique for compatibility with data-acquisition products that only support external clock sources.



How to set up the KPCMCIA AI/AO Series for external rate sampling using a digital event.

- Specify external clocking using Logical *Channel* **1.**

  Note: Channel 2 is an input to the card's counter/timer rather than an input to the ADC pacing circuitry. DriverLINX sets up the counter/timer to divide the input frequency by two, which is the minimum divisor. The hardware samples all specified channels every two pulses of the external clocking input.

- Users should connect the external clock signal to the DI 2 / ExtClk.
- Specify the *Mask* property as **Bit 0**, or 1, 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 clock (≠0), or **1** for a falling, or negative, edge clock (≠1).
- In the *Channels* section, check the *Simultaneous* box to sample all specified channels in each period. The hardware samples the channels simultaneously or as fast as possible. The hardware only supports sampling all specified channels per external clocking event. Therefore, DriverLINX assumes you want *Simultaneous* even if you do not check the box.

Analog Input Start Events

Start Events specify when the KPCMCIA AI/AO hardware starts acquiring analog input data.

The KPCMCIA AI/AO Series supports the following Start Events:

- **None**—Use this event when the DriverLINX operation doesn't require a Start Event.
- **Command**—DriverLINX starts the task on software command, i.e., as soon as DriverLINX finishes programming the KPCMCIA AI/AO hardware for the task.
- **Digital**—The KPCMCIA AI/AO starts acquiring analog input samples when the hardware detects the digital Logical Channel input satisfies the digital condition specified in the Start Event.
- **Analog**—The KPCMCIA AI/AO starts acquiring analog input samples when the hardware detects the analog Logical Channel input satisfies the condition specified in the Start Event. Only models 12AIAO, 12AIAOH and 16AIAO can recognize an analog event.

None or Null Event

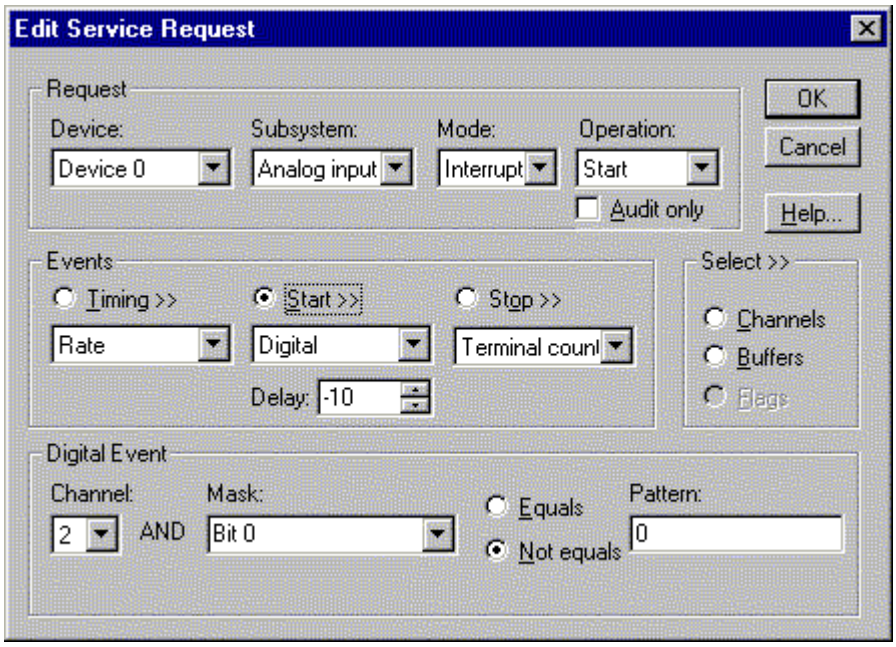
The Null Event specifies that the task does not need a Start Event to begin the task.

Command Event

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

Digital Event or About Triggering

The KPCMCIA AI/AO can acquire analog input samples ***after*** the hardware detects a digital trigger condition. In addition to acquiring any number of samples after the trigger, the KPCMCIA AI/AO can return a fixed number of samples from its FIFO buffer that it acquired ***before*** the trigger. Use about-triggering when you want to synchronize the start of data acquisition with an external signal.



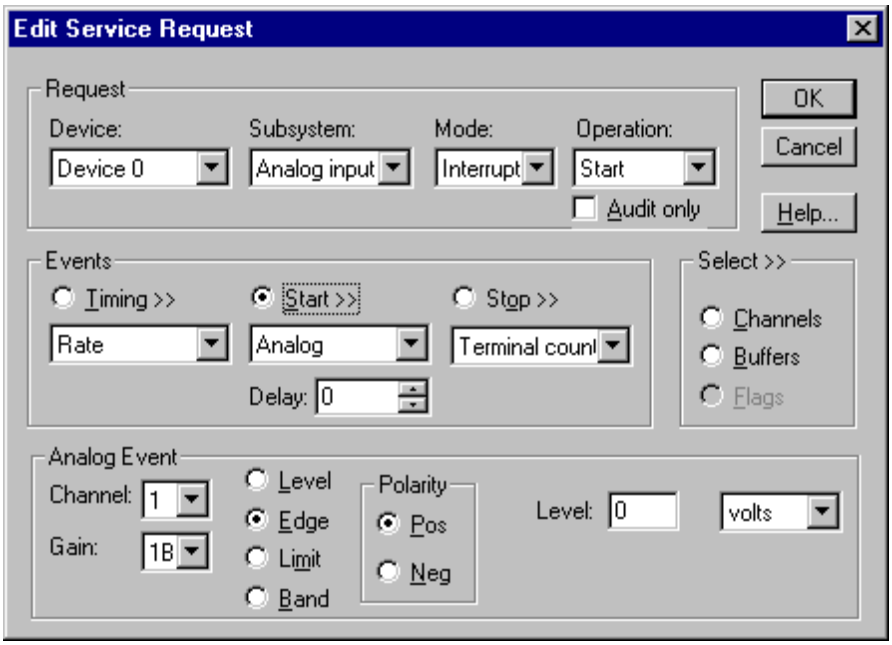
How to set up the KPCMCIA AI/AO Series for about-triggered digital input.

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

- Specify the Logical *Channel* as **2**. 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 DI 0 / Ext. Trigger line.
- Specify the *Mask* property as **Bit 0**, or 1, 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 (≠0), or **1** for a falling, or negative, edge trigger (≠1).
- Specify the *Delay* property as any integer from -2048 to $2^{32}-1$. The KPCMCIA AI/AO Series supports negative delays by returning samples from its FIFO buffer that it acquired *before* the trigger.

Analog Event or About-Triggering

The KPCMCIA AI/AO can acquire analog input samples ***after*** the hardware detects an analog trigger condition. In addition to acquiring any number of samples after the trigger, the KPCMCIA AI/AO can return a fixed number of samples from its FIFO buffer that it acquired ***before*** the trigger. Use about-triggering when you want to synchronize the start of data acquisition with an external signal. Only models 12AIAO, 12AIAOH and 16AIAO can recognize an analog event.



How to set up the KPCMCIA AI/AO 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 KPCMCIA AI/AO Series supports only edge triggering. DriverLINX samples data from the Logical Channel and compares it against the *High* and *Low Limits*. The trigger occurs when a sequence of samples is in the relationship specified by *Polarity* and *Limits*.

- Specify the *Logical Channel* from the analog input subsystem. For the KPCMCIA AI/AO Series, the analog event channel must but the first channel in the scan list. If acquiring samples before an analog trigger (either via a start event with a negative delay or a stop event), the task can only acquire data from one channel due to the nature of the trigger circuit.
- Specify the *Gain* property for the analog event channel. Use the same gain as in the scan list.
- Specify the *Polarity* (or Slope) property as **Pos** or **Neg**. Specify the *Limit* properties in hardware A/D codes as follows:

Type	High Limit	Low Limit
Edge	Threshold	Threshold

  Use the DriverLINX *Volts2Code* method to easily convert volts to hardware A/D codes for the threshold properties.
- Specify the *Delay* property as any integer from -2047 to 2047. The KPCMCIA AI/AO Series supports negative delays by returning samples from its FIFO buffer that it acquired *before* the trigger. Use caution with negative delays because they reduce the buffer capacity available for accommodating interrupt processing delays, increasing the likelihood of a data loss. Also, you can use negative delays only when acquiring a single channel.

Note: Analog triggering uses an analog-output channel for the trigger reference. Therefore, an analog-input task with analog triggering cannot be active at the same time as an analog output task.

Analog Input Stop Events

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

The KPCMCIA AI/AO Series supports the following Stop Events:

- **None**—Use this event when the DriverLINX operation doesn't 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.
- **Analog**—The data-acquisition hardware stops acquiring analog input samples when the hardware detects the analog Logical Channel input satisfies the condition specified in the Stop Event. Only models 12AIAO, 12AIAOH and 16AIAO can recognize an analog event.
- **Terminal count**—DriverLINX stops the task after the data-acquisition hardware has filled all the data buffers once.

None or Null Event

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

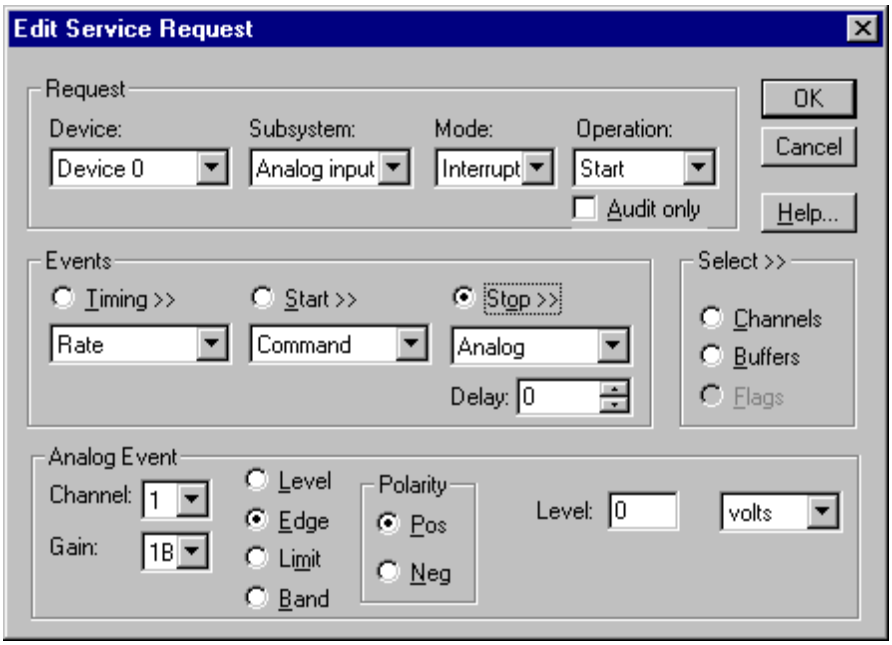
Command Event

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.

Analog Event or Pre-Triggering

The KPCMCIA AI/AO can acquire analog input samples ***until*** the hardware detects an analog trigger condition. Use pre-triggering when you want to synchronize the stopping of data acquisition with an external signal. Only models 12AIAO, 12AIAOH and 16AIAO can recognize an analog event.



How to set up the KPCMCIA AI/AO 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, Band). DriverLINX samples data from the Logical Channel and compares it against the *High* and *Low Limits*. The trigger occurs when a sequence of samples is in the relationship specified by *Polarity* and *Limits*.

- Specify the *Logical Channel* from the analog input subsystem. If acquiring samples before an analog trigger (either via a start event with a negative delay or a stop event), the task can only acquire data from one channel due to the nature of the trigger circuit.
- Specify the *Gain* property for the analog event channel.
- Specify the *Polarity* (or Slope) property as **Pos** or **Neg**.
- Specify the *Limit* properties in hardware A/D codes as follows:

Type	High Limit	Low Limit
Edge	Threshold	Threshold

  Use the DriverLINX *Volts2Code* method 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 obtain after the trigger.

Terminal Count 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 Input Channels

The KPCMCIA AI/AO 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.

The KPCMCIA AI/AO Series also supports programmable gain for each analog input channel. All models have four programmable gains and a single bipolar input range (±10V). Models KPCMCIA-12AIH and 12AIAOH have higher gain selections than the other models in the series.

The following tables show the correspondence between DriverLINX gains, the maximum input signal range, and the DriverLINX gain code for each input range:

Gain	Range (volts)	DriverLINX Gain Code
1	±10	0
2	±5	1
4	±2.5	2
8	±1.25	3

Gains, Ranges, and DriverLINX Gain Codes for Models 12AI, 12AIAO, 16AI and 16AIAO.

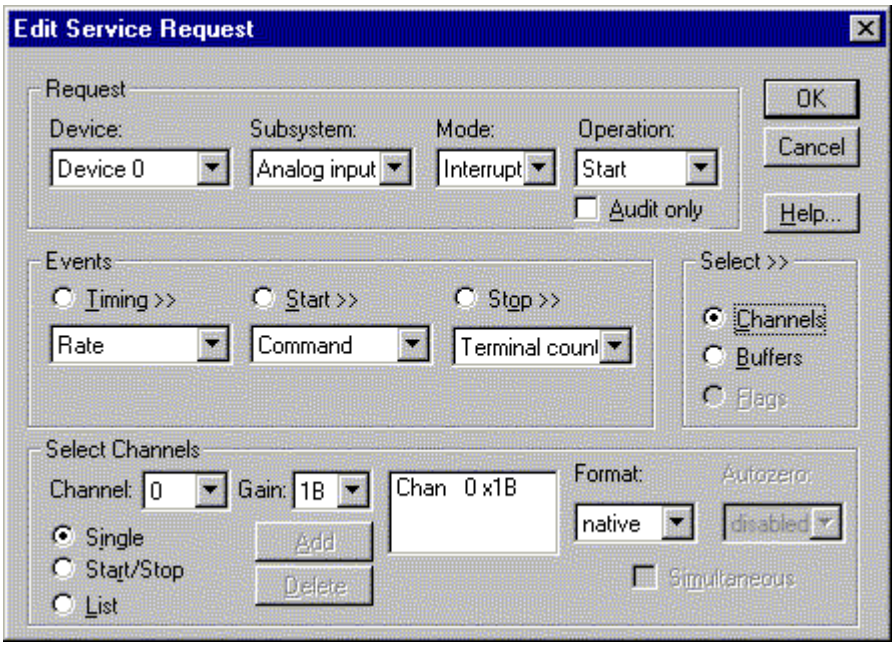
Gain	Range (volts)	DriverLINX Gain Code
1	±10	0
10	±1	1
100	±0.1	2
1000	±0.01	3

Gains, Ranges, and DriverLINX Gain Codes for Models 12AIH and 12AIAOH.

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

Single Channel Analog Input

In this mode, the KPCMCIA AI/AO Series acquires all data from one channel at the specified gain.

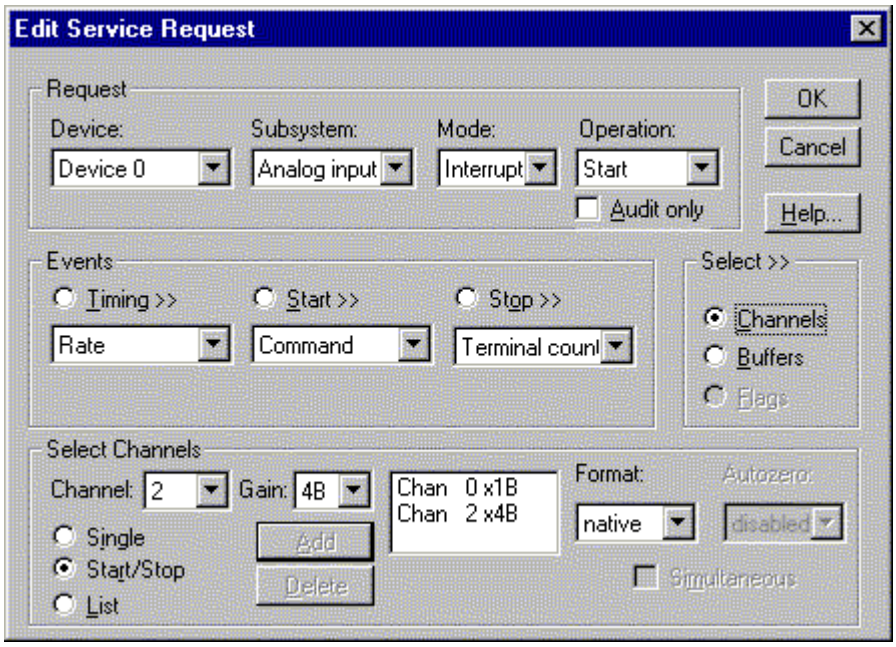


How to set up the KPCMCIA AI/AO Series for sampling on a single channel.

Multi-channel Analog Input Range

In this mode, the KPCMCIA AI/AO Series acquires all data from a consecutive range of analog channels.

- The Start Channel's gain only applies to the first Logical Channel.
- DriverLINX uses the Stop Channel's gain for all the other analog channels in the range.
- 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 Logical Channel for the KPCMCIA AI/AO model the application is using.

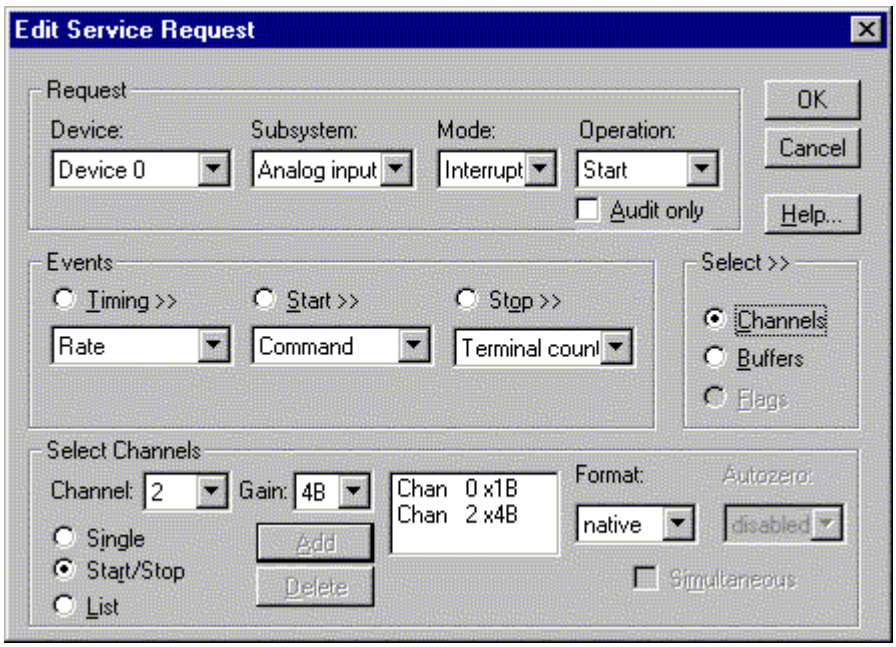


How to set up the KPCMCIA AI/AO Series for sampling on a consecutive range of channels.

Multi-channel Analog Input List

In this mode, the KPCMCIA AI/AO Series acquires all data from a random list of analog channels.

- The channel-gain list may contain up to 256 channels in any order and with any allowed gain. The list may repeat the same channel with the same or different gains.



How to set up the KPCMCIA AI/AO Series to sample on a random list of channels.

Analog Input Expansion Channels

EXP-1600 Input Multiplexers can expand the number of physical channels up to 256 differential analog input channels. The KPCMCIA hardware automatically switches the multiplexer channels, allowing you to specify expansion channels along with base channels in a channel list.

To enable DriverLINX to use multiplexer(s), configure the analog input as single-ended in the Device Configuration dialog box. Then enable expansion mode in the Special Device Settings dialog. With this configuration, DriverLINX considers the card to have the original number of base channels followed by the expansion channels.

Model	Base Channels	Expansion Channels
12AI, 12AIH and 16AI	0 – 15	16 - 271
12AIAO, 12AIAOH and 16AIAO	0 – 7	8 - 135

While a KPCMCIA AI/AO, with expansion channels enabled, has 272 or 136 logical channels, it still has at most 256 or 128 physical channels. Each mux supports 16 channels. The hardware electronically switches the 16 mux inputs to the base channel where you attached the mux.

DriverLINX uses a static addressing scheme for attaching multiplexers. In the DriverLINX scheme, attaching or removing a mux from a base channel doesn't change the logical addresses of any other channel. Also, if you remove a signal from a base channel, install a mux in its place and reattach the signal to the *first* mux channel, existing software can still read that input at its original logical address.

To determine the DriverLINX Logical Channel number for an EXP-1600 Multiplexer channel, use the following formula or refer to the following table. Note that DriverLINX uses a 0-based numbering scheme for all analog input channels.

<logical chan#> = <num base chan> + <base chan#> × <num mux chan> + <mux chan#>

<logical chan#>	Logical Channel number to use in channel lists.
<num base chan>	Number of base channels on the KPCMCIA AI/AO card. Models 12AI and 16AI have 16 base channels. Models 12AIAO and 16AIAO have 8 base channels.
<base chan#>	Base channel on the KPCMCIA AI/AO card where you attached the mux.
<num mux chan>	Number of expansion channels on the mux. Model EXP-1600 has 16 expansion channels.
<mux chan#>	Channel on the EXP-1600 where you attached the signal. Mux channels are numbered from 0 to 15.

For example, the Logical Channel address for channel 4 on a mux attached to base channel 3 on a 12AIAO is

8 + 3 × 16 + 4 = 60.

To specify multiplexer input channels 1, 2, and 3 for an EXP-1600 connected to base channel 0, add 9, 10, and 11 to the channel/gain list.

	Base Chan **Mux #**	**0** **Mux #1**	**1** **Mux #2**	**2** **Mux #3**	**3** **Mux #4**	**4** **Mux #5**	**5** **Mux #6**	**6** **Mux #7**	**7** **Mux #8**
Mux Input Chan #									
0		*8*	*24*	*40*	*56*	*72*	*88*	*104*	*120*
1		*9*	*25*	*41*	*57*	*73*	*89*	*105*	*121*
2		*10*	*26*	*42*	*58*	*74*	*90*	*106*	*122*
3		*11*	*27*	*43*	*59*	*75*	*91*	*107*	*123*
4		*12*	*28*	*44*	*60*	*76*	*92*	*108*	*124*
5		*13*	*29*	*45*	*61*	*77*	*93*	*109*	*125*
6		*14*	*30*	*46*	*62*	*78*	*94*	*110*	*126*
7		*15*	*31*	*47*	*63*	*79*	*95*	*111*	*127*
8		*16*	*32*	*48*	*64*	*80*	*96*	*112*	*128*
9		*17*	*33*	*49*	*65*	*81*	*97*	*113*	*129*
10		*18*	*34*	*50*	*66*	*82*	*98*	*114*	*130*
11		*19*	*35*	*51*	*67*	*83*	*99*	*115*	*131*
12		*20*	*36*	*52*	*68*	*84*	*100*	*116*	*132*
13		*21*	*37*	*53*	*69*	*85*	*101*	*117*	*133*
14		*22*	*38*	*54*	*70*	*86*	*102*	*118*	*134*
15		*23*	*39*	*55*	*71*	*87*	*103*	*119*	*135*

Table of logical channel numbers for eight external EXP-1600 Multiplexers and eight base channels. Models with sixteen base channels would have a larger but similar table.

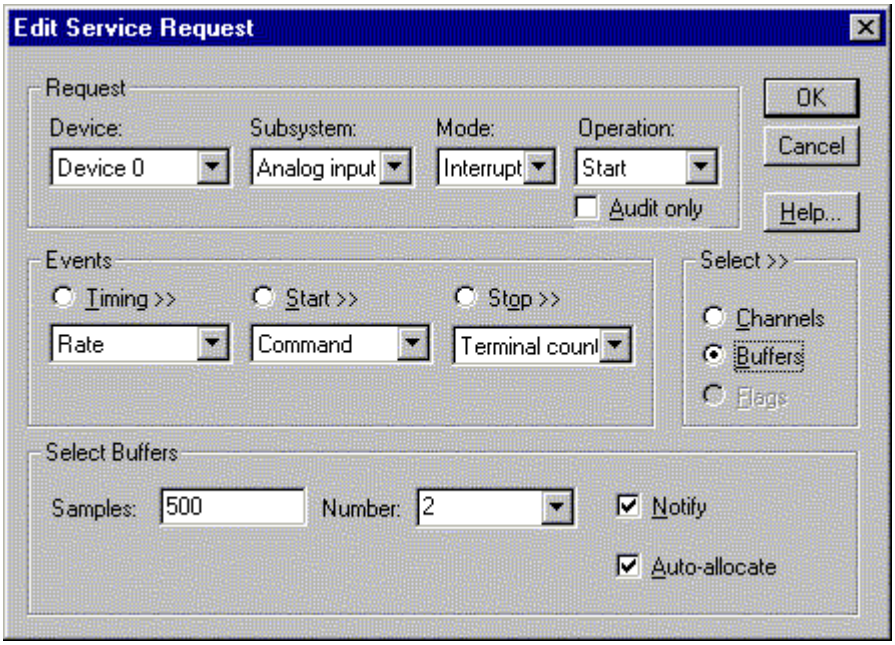
Note:

Sampling rates on expansion channels are limited by the settling time of the programmable gain amplifier on the EXP-1600, specified at (gain x 0.4) microseconds. Thus at EXP gain 10, 4 µs of settling time are needed, but at gain 500, 200 µs are required. Since the typical set-up time of KPCMCIA AI/AO Series multiplexer lines is 10 µs, any gain above 25 requires more settling time than is available, and inaccurate data will result. To correct this, make duplicate entries in the channel-gain list so that the channel is set up, sampled, and then sampled again; discard the data from the first reading and use the second, which is taken after sufficient settling time has elapsed.

Analog Input Buffers

DriverLINX supports both single-value analog input and buffered analog input.

- **For single-value input**, specify the Number of buffers as **0** and the number of *Samples* as **0**.
- **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 **256** and the number of *Samples* as desired.



How to set up the KPCMCIA AI/AO Series to store samples in buffers.

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., the number of analog input channels you're acquiring. This restriction enforces the requirement that all acquired channels have the same number of samples.

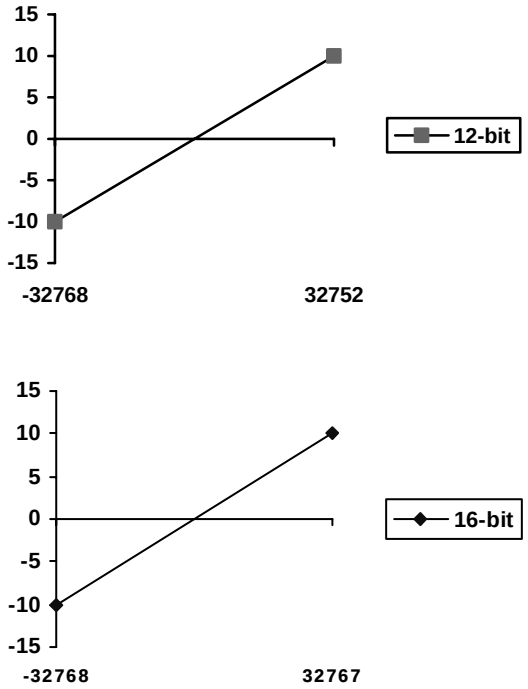
Analog Input Data Coding

All KPCMCIA AI/AO Series models return A/D hardware codes using left-justified, two's complement integers. For the 12-bit models, this means that all data values are multiplied by 16. DriverLINX refers to this coding scheme as the "native" format.

For any programmable gain, the KPCMCIA AI/AO models return hardware codes with the ranges in the following table:

A/D Resolution	Polarity	A/D Hardware Code
12 bits	Bipolar	-32768 to +32752 (-2048 × 16 to +2047 × 16)
16 bits	Bipolar	-32768 to +32767

Native A/D hardware codes for each KPCMCIA AI/AO Series resolution and polarity



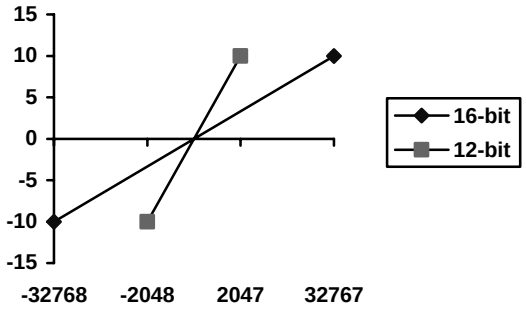
KPCMCIA AI/AO Series native A/D Codes versus Voltage Range

DriverLINX refers to the default hardware analog coding scheme as the “native” format. For computer arithmetic in a higher level language, the integer, or two’s complement, format is generally easier to use. For the 16-bit models, native and integer formats are identical.

A/D codes when converted to two’s complement integer have the following ranges and transfer functions:

A/D Resolution	Polarity	A/D Hardware Code
12 bits	Bipolar	-2048–+2047
16 bits	Bipolar	-32768–+32767

Two’s complement integer hardware codes for each KPCMCIA AI/AO Series resolution and polarity.



Converted KPCMCIA AI/AO Series integer A/D Codes versus Voltage Range

- In Polled and Interrupt modes, DriverLINX can automatically convert A/D codes to integer format, if you specify **integer** for the *Format* property in the channel list.
- Applications can use DriverLINX's data conversion operations to transform an entire data buffer from any integer format to volts.

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.
Buffer Filled	DriverLINX has filled an analog input buffer.
Start Event	DriverLINX has processed the interrupt for a hardware start event.
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

The KPCMCIA AI/AO Series generates an interrupt on a start event only for a digital event with a negative delay. DriverLINX does not send the Start Event message for any other start condition.

Analog Output Subsystem

The following sections describe how DriverLINX implements Analog Output Subsystem features for the KPCMCIA AI/AO Series.

Analog Output Modes

The Analog Output Subsystem supports the following modes:

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

Analog Output Operations

The KPCMCIA AI/AO Series Analog Output Subsystem supports the following DriverLINX operations:

- **Initialize**—aborts any active DMA or interrupt data-acquisition tasks and stops the clock. DriverLINX prevents one application process from interfering with another process'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 output data-acquisition task.
- **Message**—DriverLINX displays a pop-up dialog box for the user containing the text for the current DriverLINX error message.

Analog Output Initialization

By default, the Analog Output subsystem loads zero into both D/A channels forcing the initial output voltage to zero.

Analog Output Timing Events

Timing Events specify how the hardware paces or clocks analog output samples. DriverLINX uses the Timing Event to program when the KPCMCIA AI/AO Series writes the next analog output sample to the DACs.

The KPCMCIA AI/AO Series supports the following Timing Events:

- **None**—Output requires no pacing as DriverLINX is writing only a single value.
- **Rate**—The KPCMCIA AI/AO Series supports only fixed rate analog output using internal and external clocks.

None or Null Event

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

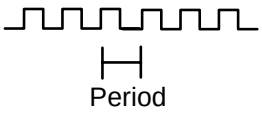
Rate Event

The KPCMCIA AI/AO Series supports one type of Rate Event for analog output:

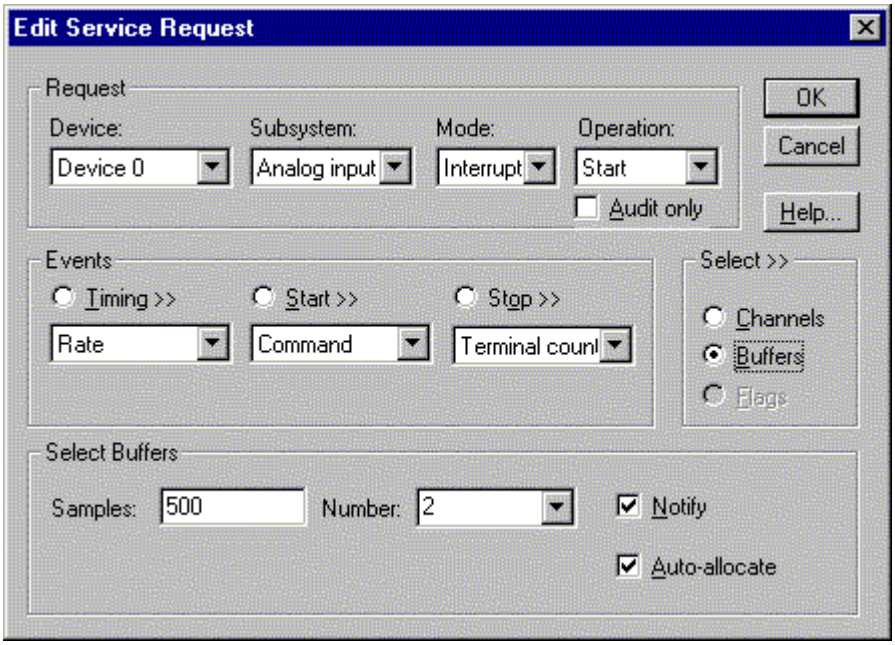
- **Rate Generator**—Generates a fixed rate clock with equal time intervals between tics.

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. Note the KPCMCIA AI/AO Series hardware always writes both analog output channels simultaneously.



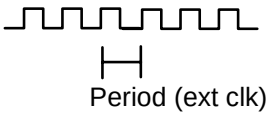
How to set up the KPCMCIA AI/AO Series for fixed rate sampling using an internal clock.

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

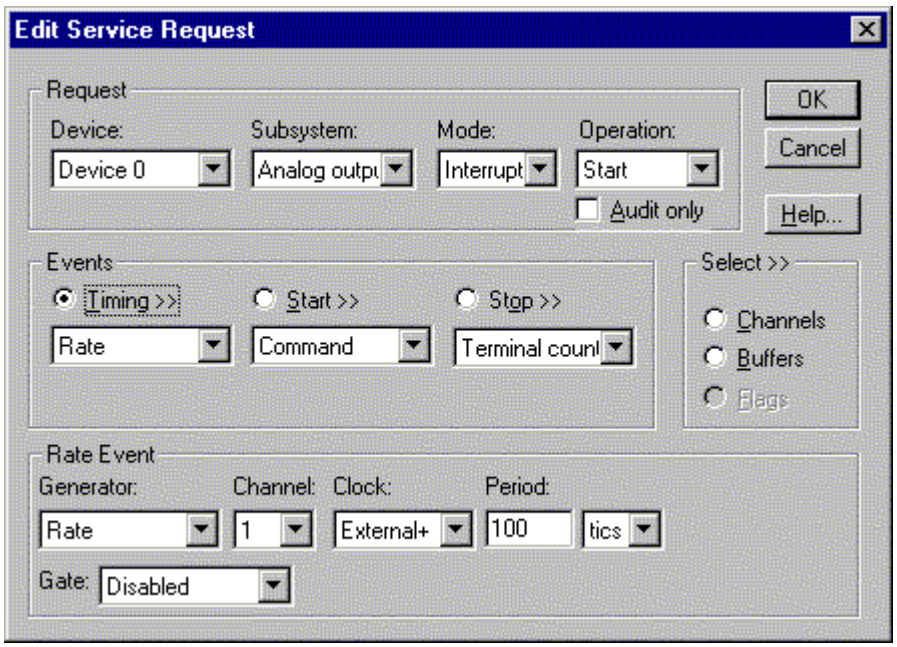
- Specify internal clocking using a **Rate** *Generator* on *Logical Channel* **1** with an **Internal 1** *Clock* source.
- The *Period* property specifies the time interval between samples in tics, where a tic is 1 µs, or 1 MHz. The minimum period is 20 tics, or 50 kHz. The maximum period is 65535 tics (2^{16} -1), or 15.26 Hz.
- In the *Channels* section, check the *Simultaneous* box to write to all specified channels in each period or leave it unchecked to write to one channel per period. When checked, the hardware writes to the channels as fast as possible.

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, the KPCMCIA AI/AO Series writes one or both DACs at each external clock tic. Note the KPCMCIA AI/AO Series hardware always writes both analog output channels simultaneously.



How to set up the KPCMCIA AI/AO Series for fixed rate sampling 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 *Logical Channel* **1** with an **External**, **External+,** or **External-** *Clock* source. External and External+ both specify sampling on the rising, or positive, edge of the external clock signal. External- specifies sampling on the falling, or negative, edge of the external clock signal.
- Users should connect the external clock signal to the DI 0 / Ext Trigger line.

- The *Period* may be any value as long as it is ≥ 20 tics, or 50 µs. The period value doesn't 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 5 MHz.
- In the *Channels* section, check the *Simultaneous* box to write to all specified channels in each period. The hardware writes to the channels as fast as possible. The hardware only supports writing all specified channels per external clocking pulse. Therefore, DriverLINX assumes you want *Simultaneous* even if you do not check the box.

Synchronous Analog Input/Output Clocking

The KPCMCIA AI/AO Series is ideal for use in "stimulus/response" type applications where a signal is applied to a "circuit" to determine its response characteristics (such as propagation delay time). The output response of the circuit must be measured simultaneously with the application of the stimulus to immediately capture the response for precise calculation of delay times and phase angles. This technique will eliminate the measurement errors caused by asynchronous analog I/O.

DriverLINX supports synchronous analog I/O using a shared pacing clock source. To synchronize analog input with analog output, set up two Service Requests as follows:

1. **Analog Output Service Request** (SR). The Timing Event should be a Rate Generator using Logical Channel 0 (AI clock). This specifies that the AO SR will use the same pacing source as the next, or currently active, AI SR. The Period must be 0.
2. **Analog Input SR with any valid Timing Event**. When the AI task starts sampling, the AO task will also sample using the same clock source (internal or external).

- Although both SRs share the same clock source, they are otherwise logically independent of each other. Your application must manage and respond to each Service Request separately.
- If the AI task terminates before the AO task, the AO task will still be logically active, but the clock stops sending timing pulses to the AO task until the next AI task starts.
- If you want to terminate the AO task when the AI task stops, either set up both SRs with equal buffer sizes and Stop Events, or issue a Stop operation request for the AO task.

Analog Output Start Events

Start Events specify when the KPCMCIA AI/AO Series hardware starts writing analog output data.

The KPCMCIA AI/AO Series supports the following Start Events for analog output:

- **None**—Use this event when the DriverLINX operation doesn't require a Start Event.
- **Command**—DriverLINX starts the task on software command, i.e., as soon as DriverLINX finishes programming the KPCMCIA AI/AO hardware for the task.

None or Null Event

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

Command Event

The Command Event starts data acquisition as soon as DriverLINX has completed programming the KPCMCIA AI/AO Series hardware with the task parameters.

Analog Output Stop Events

Stop Events specify when the KPCMCIA AI/AO Series hardware stops writing analog output data.

The KPCMCIA AI/AO Series supports the following Stop Events for analog output:

- **None**—Use this event when the DriverLINX operation doesn't 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 written all the data buffers once.

None or Null Event

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

Command 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 them to the DACs on the KPCMCIA AI/AO Series.

Terminal Count Event

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

Analog Output Channels

The KPCMCIA AI/AO 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.

Analog Output Logical Channels

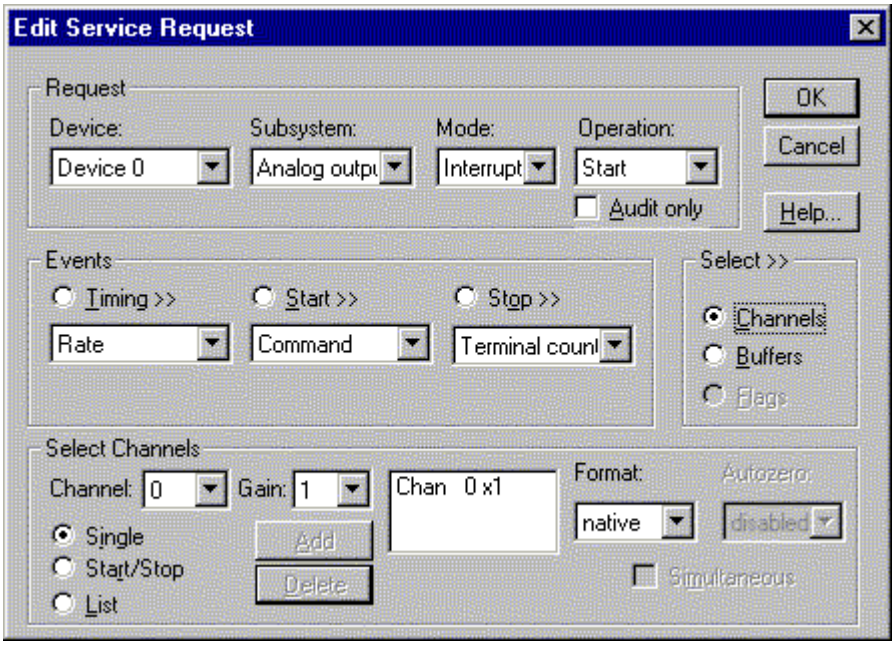
The 12AIAO, 12AIAOH and 16AIAO models have two 12/16-bit digital-to-analog converters, D/A CH0 and D/A CH1. DriverLINX maps these DACs to Logical Channels as follows:

DAC	Logical Channel
0	0
1	1

These converters have a fixed gain for each analog output channel (±5V). For software portability always set the DriverLINX hardware Gain Code to zero.

Single Channel Analog Output

In this mode, the KPCMCIA AI/AO Series acquires all data from one channel at unity gain.



How to set up the KPCMCIA AI/AO Series to write to a single DAC

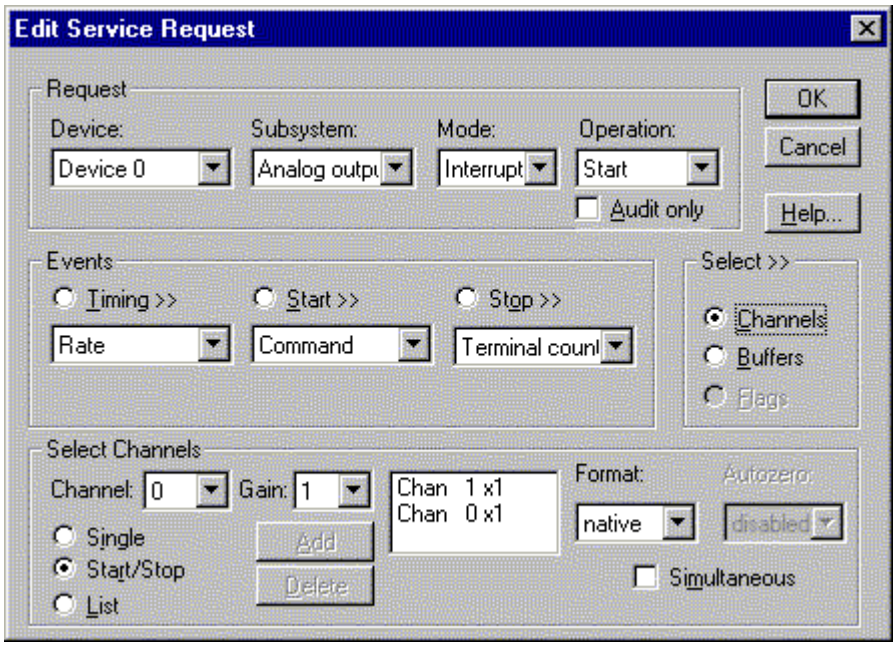
Multi-channel Analog Output Range

In this mode, the KPCMCIA AI/AO Series writes all data from a consecutive range of analog channels.

- The Start Channel's gain only applies to the first Logical Channel.
- DriverLINX uses the Stop Channel's gain for all the other analog channels in the range.
- 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 Logical Channel for the KPCMCIA AI/AO Series model the application is using.

- The KPCMCIA AI/AO Series always outputs to both DACs simultaneously.

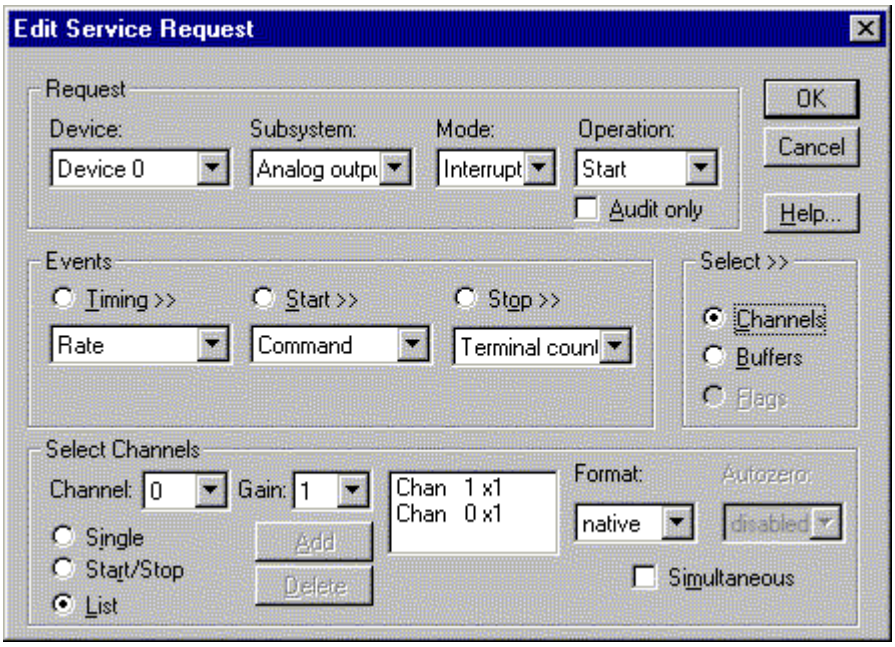


How to set up the KPCMCIA AI/AO Series using a start/stop range to write to both DACs

Multi-channel Analog Output List

In this mode, the KPCMCIA AI/AO Series acquires all data from a random list of analog channels.

- The channel-gain list may contain only two channels in any order and only with unity gain. The list may ***not*** repeat the same channel.
- The KPCMCIA AIAO always outputs to both DACs simultaneously.

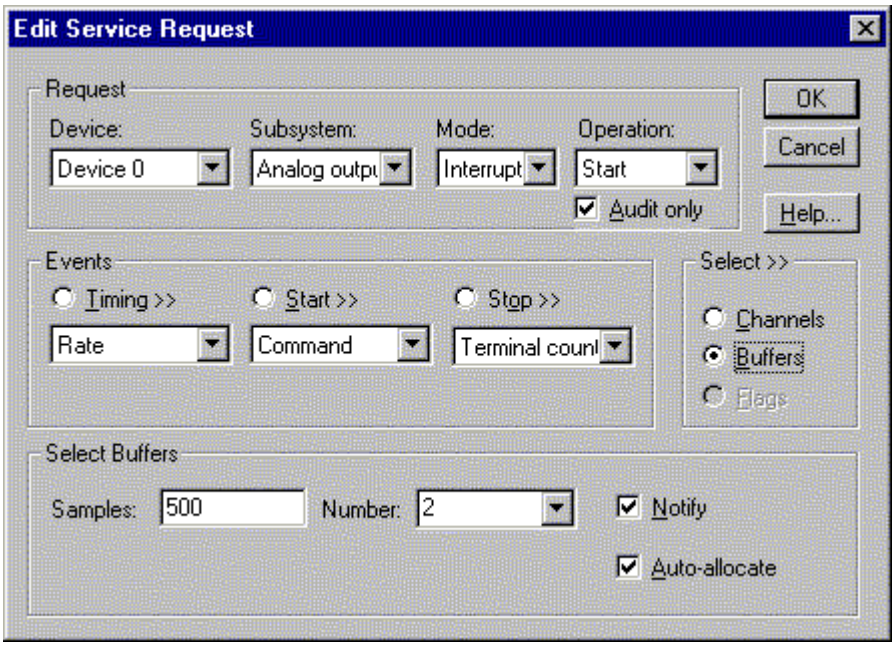


How to set up the KPCMCIA AI/AO Series using a channel list to write to both DACs

Analog Output Buffers

DriverLINX supports both single-value analog output and buffered analog output.

- **For single-value output**, specify the Number of buffers as **0** and the buffer size as **0**.
- **For buffered output**, specify the Number of buffers from **1** to **256** and the buffer size as desired.



How to set up the KPCMCIA AI/AO Series for analog output using buffers

For example, 500 samples/2 channels = 250 is ok, but 503 samples/2 channels = 251.5 is incorrect.

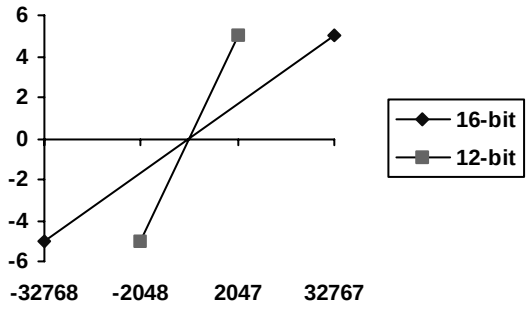
An individual DriverLINX buffer may have any size as long as the buffer length holds an integral number of channel scans, i.e., the number of analog output channels you're acquiring. This restriction enforces the requirement that all output channels have the same number of samples.

Analog Output Data Coding

Models 12AIAO and 12AIAOH encode the D/A output voltages using a 12-bit right-justified two's complement integer. Model 16AIAO encodes the D/A output voltages using a 16-bit two's complement integer. DriverLINX refers to these coding schemes as the "native" format.

D/A Resolution	Range	A/D Hardware Code
12 bits	±5 V	-2048 – +2047
16 bits	±5 V	-32768 – +32767

Two's complement integer hardware codes for the KPCMCIA AI/AO Series DACs.



KPCMCIA AI/AO Series native D/A Codes versus Voltage Range

Applications can use DriverLINX's data conversion operations to transform an entire data buffer from volts 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.
Buffer Filled	DriverLINX has finished writing from the data buffer.
Data Lost	DriverLINX has detected an analog output data underrun condition.
Critical Error	DriverLINX has encountered an unexpected hardware or software condition.

DriverLINX Event messages for analog output

Digital Input Subsystem

The following sections describe how DriverLINX implements Digital Input Subsystem features for the KPCMCIA AI/AO Series.

Digital Input Modes

The Digital Input Subsystem supports the following modes:

- **Polled**—For single value digital input samples.
- **Interrupt**—For buffered transfers using programmed I/O.
- **Other**—For subsystem initialization and data conversion.

Digital Input Operations

The KPCMCIA AI/AO 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 process from interfering with another process'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 read into a buffer.
- **Stop**—terminates a digital input data-acquisition task.
- **Message**—DriverLINX displays a pop-up dialog box for the user containing the text for the current DriverLINX error message.

Digital Port Configuration

The KPCMCIA AI/AO Series has separate, dedicated digital input and output ports and doesn't require the application to configure its digital I/O ports.

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 KPCMCIA AI/AO Series reads the next digital input sample from the port.

The KPCMCIA AI/AO Series supports the following Timing Events:

- **None**—Input requires no pacing as DriverLINX is reading only a single value.
- **Rate**—The KPCMCIA AI/AO Series supports only fixed rate digital input using an internal system clock.

None or Null Event

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

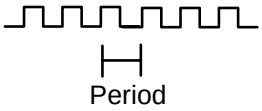
Rate Event

The KPCMCIA AI/AO Series supports one type of Rate Event for digital input:

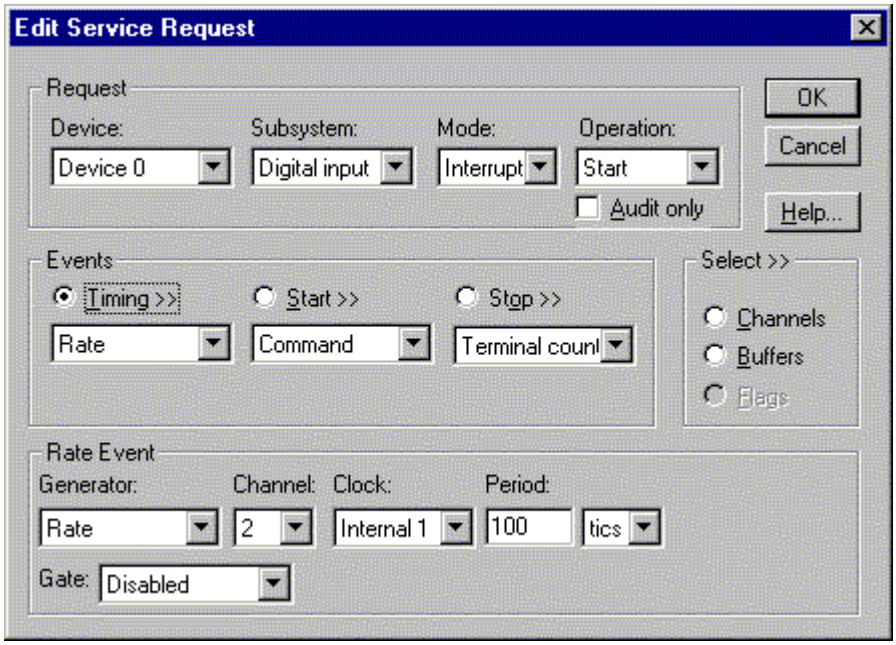
- **Rate Generator**—Generates a fixed rate clock with equal time intervals between tics.

Rate Generator: Internal Clocking

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



As the KPCMCIA AI/AO Series doesn't support a dedicated clock for digital input timing, DriverLINX uses the operating system timing clock for pacing digital input. The system timing clock doesn't have the resolution or time granularity of a dedicated clock, but it is an excellent substitute for low frequency digital input polling operations.



How to set up the KPCMCIA AI/AO 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 input timing.

- Specify internal clocking using a **Rate** *Generator* on *Logical Channel* **2** with an **Internal 1** *Clock* source.
- The *Period* property specifies the time interval between samples in tics, where a system timer tic is 1 µs, or 1 MHz. The minimum period is 100 tics, or 10 kHz. The maximum period is 4,294,967,295 tics (2^{32} -1), or 0.0002 Hz. The resolution and granularity of the system timer are operating system dependent. Under current versions of Windows, throughput is less than 1 kHz. See "Counter/Timer Subsystem" on page 68 for details on the system timer.

Digital Input Start Events

Start Events specify when the KPCMCIA AI/AO Series hardware starts reading digital input data.

The KPCMCIA AI/AO Series supports the following Start Events for digital input:

- **None**—Use this event when the DriverLINX operation doesn't require a Start Event.
- **Command**—DriverLINX starts the task on software command, i.e., as soon as DriverLINX finishes programming the KPCMCIA AI/AO hardware for the task.

None or Null Event

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

Command Event

The Command Event starts data acquisition as soon as DriverLINX has completed programming the KPCMCIA AI/AO Series hardware with the task parameters.

Digital Input Stop Events

Stop Events specify when the KPCMCIA AI/AO Series hardware stops reading digital input data.

The KPCMCIA AI/AO Series supports the following Stop Events for digital input:

- **None**—Use this event when the DriverLINX operation doesn't 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 KPCMCIA AI/AO Series hardware has filled all the data buffers once.

None or Null Event

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

Command 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, reading from the digital port on the KPCMCIA AI/AO Series.

Terminal Count 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.

Digital Input Channels

The KPCMCIA AI/AO 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.

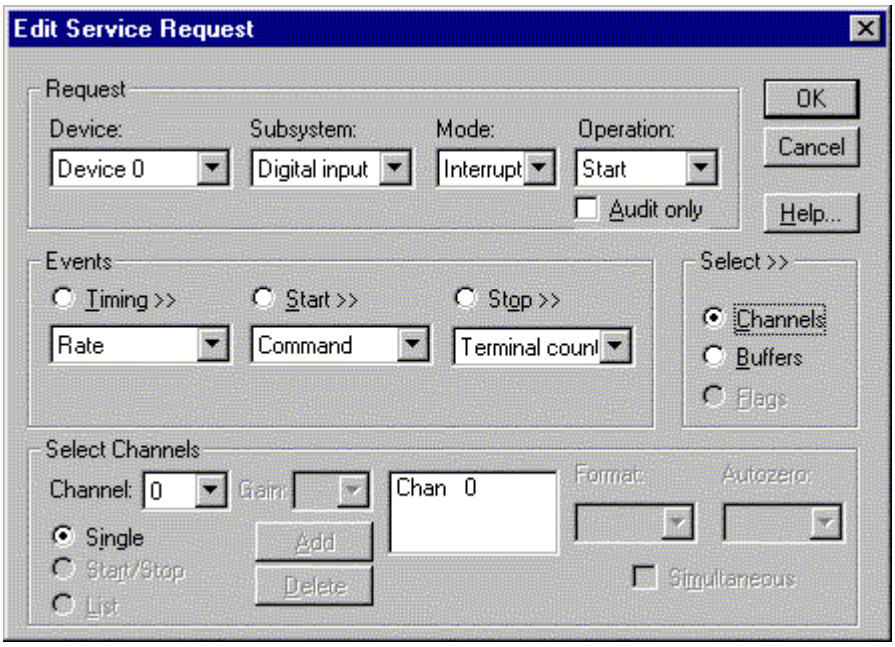
Digital Input Logical Channels

The KPCMCIA AI/AO Series has a single digital input port that DriverLINX designates as Logical Channel 0. DriverLINX defines two additional Logical Channels for the external clock and trigger signals but applications cannot directly read their values.

Logical Channel	DriverLINX Function	KPCMCIA AI/AO Series External Connector
0	Standard Digital Input	Digital input lines (DI 0 … DI 3)
1	External Clock	DI 2 / ExtClk
2	External Trigger	DI 0 / Ext. Trigger

Single Channel Digital Input

In this mode, the KPCMCIA AI/AO Series acquires all data from one channel.



How to set up the KPCMCIA AI/AO Series to read from a single channel

Multi-channel Digital Input Range

Even though the KPCMCIA AI/AO Series has only one digital input channel, DriverLINX supports specifying a channel range for compatibility with applications that use this method.

In this mode, the KPCMCIA AI/AO Series acquires all data from a consecutive range of digital channels.

- The Start and Stop Channel must specify Logical Channel 0.

Multi-channel Digital Output List

Even though the KPCMCIA AI/AO Series has only one digital input channel, DriverLINX supports specifying a channel range for compatibility with applications that use this method

In this mode, the KPCMCIA AI/AO Series acquires all data from a random list of digital channels.

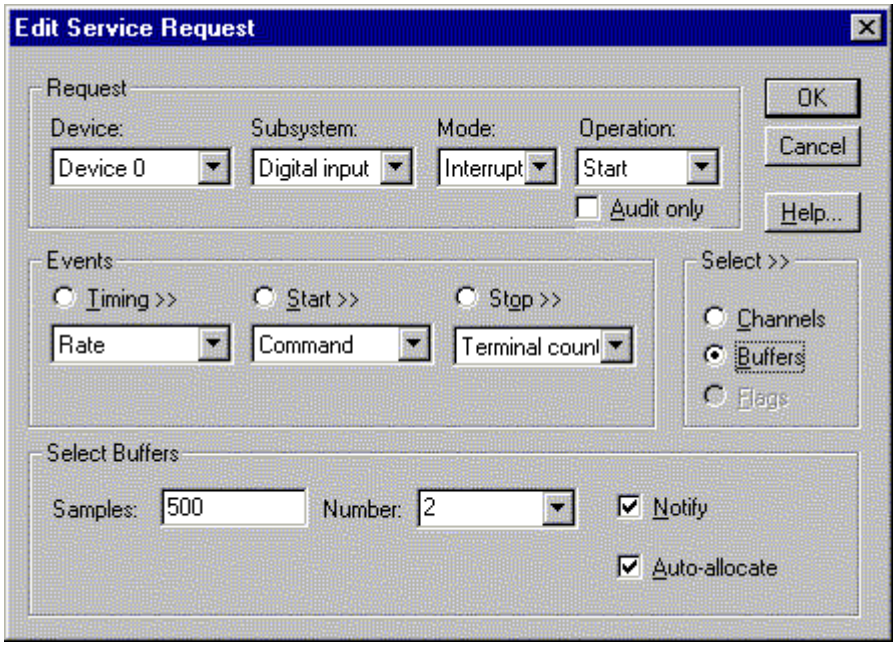
- The channel list may contain only one channel.
- As the KPCMCIA AI/AO Series only has a single digital input channel available for reading, this technique is equivalent to Single Channel Digital Input.

Digital Input Buffers

DriverLINX supports both single-value digital input and buffered digital input.

- **For single-value output**, specify the Number of buffers as **0** and the buffer size as **0**.

- **For buffered output**, specify the Number of buffers from **1** to **256** and the buffer size as desired.



How to set up the KPCMCIA AI/AO Series to read digital samples using data buffers

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

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.
Buffer Filled	DriverLINX has filled a data buffer with digital input
Data Lost	DriverLINX has detected a digital input data overrun condition.
Critical Error	DriverLINX has encountered an unexpected hardware or software condition.

DriverLINX Event message for digital input

Digital Output Subsystem

The following sections describe how DriverLINX implements Digital Output Subsystem features for the KPCMCIA AI/AO Series.

Digital Output Modes

The Digital Output Subsystem supports the following modes:

- **Polled**—For single value digital output samples.
- **Interrupt**—For buffered transfers using programmed I/O.
- **Other**—For subsystem initialization and data conversion.

Digital Output Operations

The KPCMCIA AI/AO 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 process from interfering with another process'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 a digital output data-acquisition task.
- **Message**—DriverLINX displays a pop-up dialog box for the user containing the text for the current DriverLINX error message.

Digital Output Initialization

By default, the Digital Output subsystem writes zero into the digital output port. You can specify a different initial output value using the *Configure DriverLINX Device* dialog. See "Digital Output Subsystem Page" on page 17.

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 KPCMCIA AI/AO Series writes the next digital output sample from the port.

The KPCMCIA AI/AO Series supports the following Timing Events:

- **None**—Output requires no pacing as DriverLINX is writing only a single value.
- **Rate**—The KPCMCIA AI/AO Series supports only fixed rate digital output using an internal system clock.

None or Null Event

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

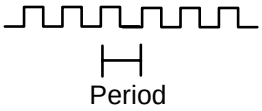
Rate Event

The KPCMCIA AI/AO Series supports one type of Rate Event for digital output:

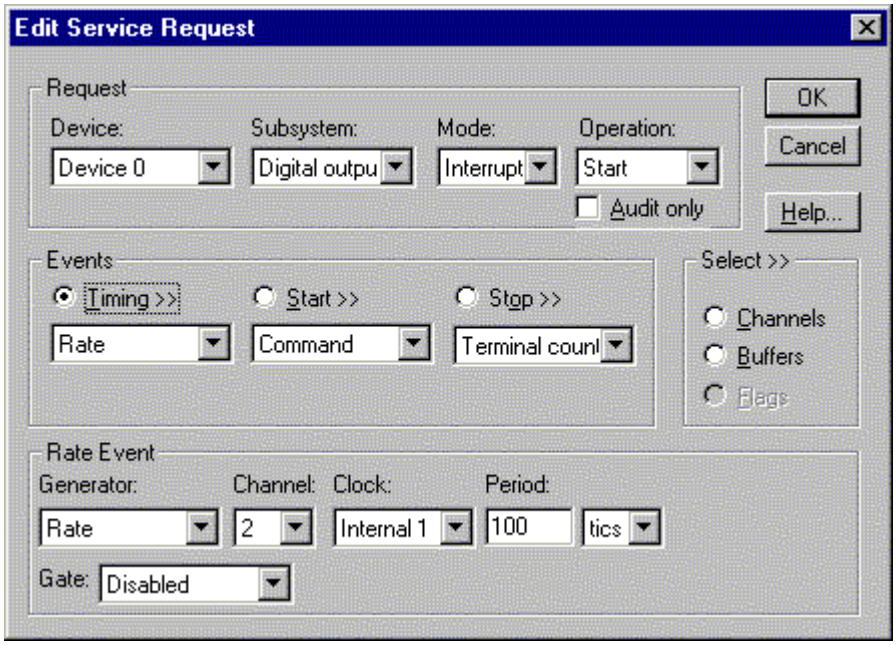
- **Rate Generator**—Generates a fixed rate clock with equal time intervals between tics.

Rate Generator: Internal Clocking

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



As the KPCMCIA AI/AO Series doesn't support a dedicated clock for digital output timing, DriverLINX uses the operating system timing clock for pacing digital output. The system timing clock doesn't have the resolution or time granularity of a dedicated clock, but it is an excellent substitute for low frequency digital output writing operations.



How to set up the KPCMCIA AI/AO 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 *Logical Channel* **2** with an **Internal 1** *Clock* source.
- The *Period* property specifies the time interval between samples in tics, where a system timer tic is 1 µs, or 1 MHz. The minimum period is 100 tics, or 10 kHz. The maximum period is 4,294,967,295 tics (2^{32} -1), or 0.0002 Hz.

As the KPCMCIA AI/AO Series doesn't have an internal hardware timer for pacing digital I/O, DriverLINX uses an internal system timer. The resolution and granularity of the system timer are operating system dependent. Typically, the system timer has a 10-msec resolution and granularity. Under current versions of Windows, throughput is less than 1 kHz.

DriverLINX implements the system timer to assist applications in creating untimed polling loops. Windows multitasking constraints cause significant jitter and lack of precision the system timer's interrupt rate. It is not suitable for applications requiring precise timing.

Digital Output Start Events

Start Events specify when the KPCMCIA AI/AO Series hardware starts writing digital output data.

The KPCMCIA AI/AO Series supports the following Start Events for digital output:

- **None**—Use this event when the DriverLINX operation doesn't require a Start Event.
- **Command**—DriverLINX starts the task on software command, i.e., as soon as DriverLINX finishes programming the KPCMCIA AI/AO hardware for the task.

None or Null Event

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

Command Event

The Command Event starts data acquisition as soon as DriverLINX has completed programming the KPCMCIA AI/AO hardware with the task parameters.

Digital Output Stop Events

Stop Events specify when the KPCMCIA AI/AO Series hardware stops writing digital output data.

The KPCMCIA AI/AO Series supports the following Stop Events for digital output:

- **None**—Use this event when the DriverLINX operation doesn't 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 KPCMCIA AI/AO Series hardware has written all the data buffers once.

None or Null Event

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

Command 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 to the digital port on the KPCMCIA AI/AO Series.

Terminal Count 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 KPCMCIA AI/AO Series allows applications to specify the digital channels using three techniques:

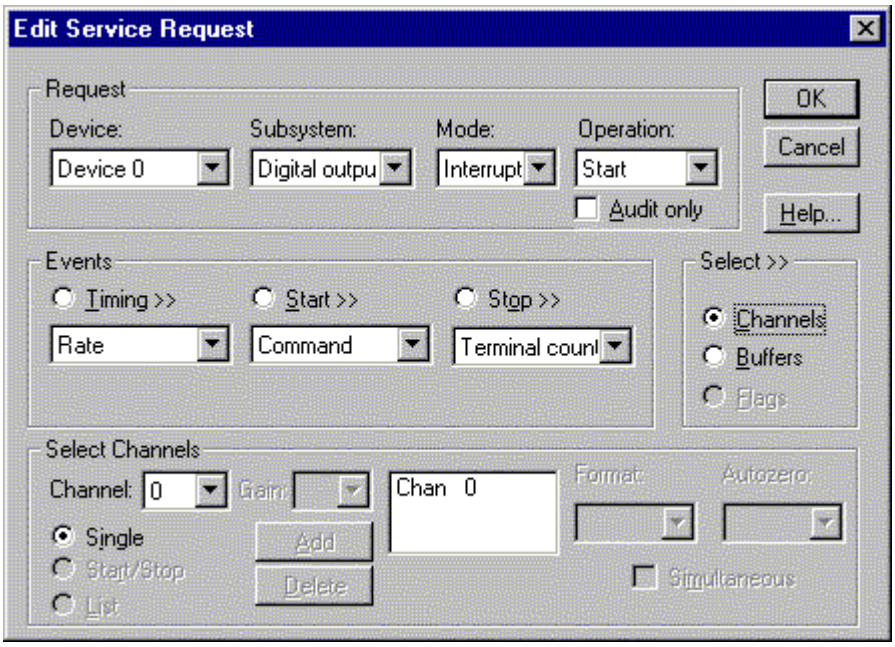
- **Start Channel**—Acquire data from a single channel.
- **Start/Stop Channel Range**—Acquire data from a consecutive range of channels.
- **Channel List**—Acquire data from a list of channels.

Digital Output Logical Channels

The KPCMCIA AI/AO Series has a single digital output port that DriverLINX designates as Logical Channel 0.

Single Channel Digital Output

In this mode, the KPCMCIA AI/AO Series writes all data from one channel.



How to set up the KPCMCIA AI/AO Series to write a single digital output channel

Multi-channel Digital Output Range

Even though the KPCMCIA AI/AO Series has only one digital output channel, DriverLINX supports specifying a channel range for compatibility with applications that use this method.

In this mode, the KPCMCIA AI/AO Series acquires all data from a consecutive range of digital channels.

- The Start and Stop Channel must specify Logical Channel 0.

Multi-channel Digital Output List

Even though the KPCMCIA AI/AO Series has only one digital output channel, DriverLINX supports specifying a channel range for compatibility with applications that use this method

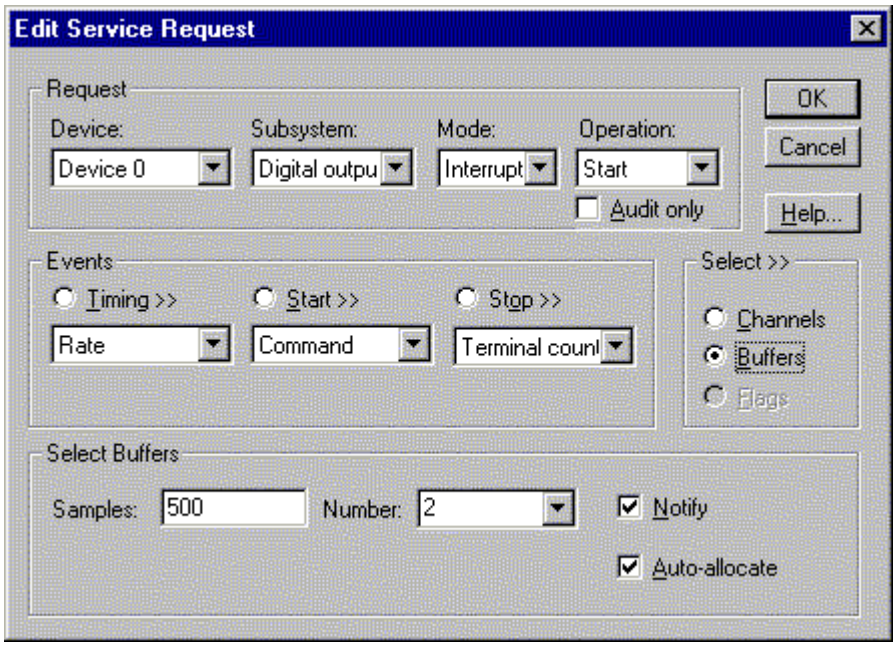
In this mode, the KPCMCIA AI/AO Series acquires all data from a random list of digital channels.

- The channel list may contain only one channel.
- As the KPCMCIA AI/AO Series only has a single digital output channel available for writing, this technique is equivalent to Single Channel Digital Output.

Digital Output Buffers

DriverLINX supports both single-value digital output and buffered digital output.

- **For single-value output**, specify the Number of buffers as **0** and the buffer size as **0**.
- **For buffered output**, specify the Number of buffers from **1** to **256** and the buffer size as desired.



How to set up the KPCMCIA AI/AO Series to write digital output using data buffers

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

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.
Buffer Filled	DriverLINX has written all data in the buffer
Data Lost	DriverLINX has detected a digital output data overrun condition.
Critical Error	DriverLINX has encountered an unexpected hardware or software condition.

DriverLINX Event messages for digital output

Counter/Timer Subsystem

The KPCMCIA AI/AO Series has counter/timers to pace analog input and output. The analog output pacer clock, on model 12AIAO, 12AIOH and 16AIAO, can be used for independent Counter/Timer Subsystem task in Count mode. The driver uses a Windows system clock to pace digital input/output. DriverLINX defines the following counter/timer channels:

All models have a hardware clock to pace analog input and a system clock to pace digital I/O. Models 12AIAO, 12AIAOH and 16AIAO also have a hardware clock to pace analog output.

The following table lists the Counter/Timer Subsystem's Logical Channels and shows their allowable clock sources, modes and gates.

Logical Channel	Clocks		Modes	Gates
	Source	Tic Period		
0 — AI Pacer	Internal 1	0.2 µs (5 MHz)	Rate Gen	Disabled
	Internal 2	0.2 µs (5 MHz)	Burst Gen	No Connect
	Internal 3	1 µs (1 MHz)		
	Internal 4	10 µs (10 MHz)		
	External			
	External+			
	External-			
1 — AO Pacer	Internal 1	1 µs (1 MHz)	Rate Gen	Enabled
	External		Count	Disabled
	External+			High Level Enabled
	External-			
2 — System	Internal 1	1 µs (1 MHz)	Rate Gen	Disabled
				No Connect

Counter/Timer Subsystem Logical Channels and Allowed Clocks, Modes and Gates

Analog Input Pacer Clock

The Analog Input Pacer Clock (Logical Channel 0) is a 24-bit counter combined with a 3-range prescaler, which supports both internal and external clock sources. The hardware limits its use to pacing analog input tasks only. It does not support independent counter/timer tasks.

Internal Clocking

The KPCMCIA AI/AO Series has a master oscillator with three prescaler outputs to pace analog input tasks. DriverLINX defines the following internal clock sources for this Logical Channel:

- **Internal 1—**specifies a clock source that allows the full range of output frequencies, from 0.006 Hz to 100 kHz. With this clock source each tic is
- **Internal 2—**specifies a clock source that allows output frequencies from 0.3 Hz to 100 kHz. With this clock source each tic is
- **Internal 3—**specifies a clock source that allows output frequencies from 0.06 Hz to 100 kHz. With this clock source each tic is
- **Internal 4—**specifies a clock source that allows output frequencies from 0.006 Hz to 50 kHz. With this clock source each tic is

The **Internal 1** clock source always allows applications to specify the full range of frequencies that the data-acquisition hardware supports, automatically using appropriate prescaler and counter settings.

Applications having a special need to control the selection of prescalers can use other internal clock sources. On the KPCMCIA AI/AO Series, **Internal 2, Internal 3** and **Internal 4** allow applications to select the counter's divide-by-2, divide-by-10 and divide-by-100 prescalers, respectively.

External Clocking

The KPCMCIA AI/AO Series allows an external clock source to pace analog input tasks. DriverLINX defines the following external clock sources for the Analog Input Pacer Clock:

- **External, External+** —specify sampling on the rising, or positive, edge of the external clock signal.
- **External-** —specifies sampling on the falling, or negative, edge of the external clock signal.

Clocking Modes

The Analog Input Pacer Clock can operate in several pacing modes. DriverLINX defines the following clock modes for this Logical Channel:

- **Rate Generator—**specifies sampling of one analog input channel in the scan list at each tic of an internal or external clock source.
- **Burst Generator (Internal Clock Source)**—specifies a dual-frequency clock where the internal clock source activates a scan of channels in the channel list, repeated at a specified major period. The internal clock paces sampling of the channels in each burst at a specified minor period. For the KPCMCIA AI/AO Series, each burst must scan all the channels in the channel list. Also, the minor period must have a frequency of 25 kHz, 50 kHz or 100 kHz. If the application specifies an unsupported frequency, DriverLINX will round up to the next highest supported frequency.

- **Burst Generator (External Clock Source)—**specifies a dual-frequency clock where an external clock source activates a scan of channels in the channel list. The internal clock paces the sampling of the channels in each burst at a specified minor period. For the KPCMCIA AI/AO Series, each burst must scan all the channels in the channel list. Also, the minor period must have a frequency of 25 kHz, 50 kHz or 100 kHz. If the application specifies an unsupported frequency, DriverLINX will round up to the next highest supported frequency.

Gating

The KPCMCIA AI/AO Series doesn't have a gate control that modulates the analog input pacer clock source. Select **Disabled** or **No Connect** for the *Gate Status* of Rate Events.

Analog Output Pacer Clock

The Analog Output Pacer Clock (Logical Channel 1) is a 16-bit counter, which supports both internal and external clock sources.

Internal Clocking

The KPCMCIA AI/AO Series has a master oscillator with a single frequency to pace analog output tasks. DriverLINX defines the following internal clock source for this Logical Channel:

- **Internal 1—**specifies a 1 MHz clock source. With this clock source each tic is 1 µs. Allowable frequencies range from 16 Hz to 100 kHz.

External Clocking

The KPCMCIA AI/AO Series allows an external clock source to pace analog output tasks. DriverLINX defines the following external clock sources for this Logical Channel:

- **External** and **External+ —**specify sampling on the rising, or positive, edge of the external clock signal.
- **External-** —specifies sampling on the falling, or negative, edge of the external clock signal.

Clocking Modes

The analog output pacer clock can operate in two modes. DriverLINX defines the following clock modes for this Logical Channel:

- **Rate Generator—**specifies to writing to all the analog output channels in the scan list at each tic of an internal or external clock source. For use with analog output tasks only.
- **Count**—specifies a 16-bit, repetitive event counter with an external input in polled. For use with counter/timer tasks only. For more

information see Event Counting in the "Counter/Timer Programming Guide."

Gating

The KPCMCIA AI/AO Series has an external gate control that modulates the Analog Output Pacer Clock source. DriverLINX defines the following gating modes for this Logical Channel:

- **Enabled—**specifies that the hardware should allow the external gate signal to modulate the clock source. For the KPCMCIA AI/AO Series, **Enabled** is the same as **High Level Enabled**.
- **Disabled—**specifies that the hardware should disable, or ignore, the gate.
- **High Level Enabled—**specifies that the hardware should allow the external gate signal to modulate the clock source. The gate is active when the signal is high and inactive when the signal is low.

System Pacer Clock

The System Pacer Clock (Logical Channel 2) is an internal, software clock source based on the Windows system timer. DriverLINX uses this internal system clock to pace digital I/O, as the KPCMCIA AI/AO Series doesn't have an internal hardware timer. The System Pacer Clock does not support independent counter/timer tasks.

DriverLINX provides the system clock for low frequency polling of digital I/O. The resolution and granularity of the system timer are operating system dependent.

Internal Clocking

The System Pacer Clock has a single frequency source to pace digital I/O tasks. DriverLINX defines the following internal clock source for this Logical Channel:

- **Internal 1—**specifies a 1 MHz clock source. With this clock source each tic is 1 µs.

Typically, the system timer has a 10-ms resolution and granularity. Under current versions of Windows, throughput is less than 1 kHz.

Clocking Modes

The System Pacer Clock can operates in only the following mode:

- **Rate Generator—**specifies sampling of one channel at each tic of an internal or external clock source.

Gating

The system timer doesn't have a gate control. Select **Disabled** or **No Connect** for the *Gate Status* of Rate Events.

Counter/Timer Interrupt

The KPCMCIA AI/AO Series supports counter/timer interrupts indirectly. You can set up an INTERRUPT mode analog input or output task with a single sample buffer. Buffer Filled messages notify your application of the counter/ timer interrupt.

Uninstalling DriverLINX

How do I uninstall DriverLINX?

DriverLINX consists of three separate component installations:

- DriverLINX for Keithley KPCMCIA
- 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 KPCMCIA, you must

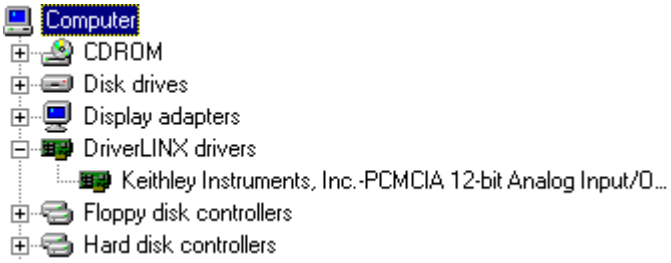
- Disable the DriverLINX driver.
- Shut down your computer to remove the hardware.
- Reboot your computer to unload the driver.
- Run the DriverLINX uninstall program.

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 KPCMCIA 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 KPCMCIA card that you are uninstalling.
5. Close the DriverLINX Configuration Panel.
6. When finished, shut down your computer and physically remove any installed KPCMCIA hardware.
7. Reboot Windows.
8. Continue with "How to disable the DriverLINX Setup Information files in Windows 95/98" on page 74.

How to Disable a DriverLINX Driver in Windows 95/98/Me/2000

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 KPCMCIA devices.

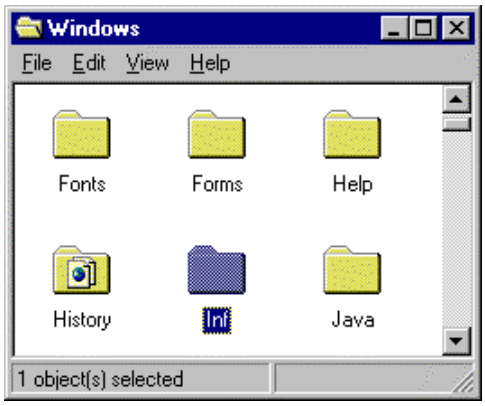


3. Select the KPCMCIA device you want to disable.
4. Click the “Remove” button.
5. In the “Confirm Device Removal” dialog, select “OK”.
6. Repeat steps 3-5 for each KPCMCIA card or driver that you uninstalling.
7. When finished, click “Close”, shut down your computer, and physically remove any installed KPCMCIA hardware.
8. Reboot Windows.
9. To finish uninstalling, see “How to Remove DriverLINX for the Keithley KPCMCIA Series” on page 77. Also, see “How to disable the DriverLINX Setup Information files in Windows 95/98” on page 74 to ensure Windows does not try to re-install the DriverLINX driver the next time you insert your card.

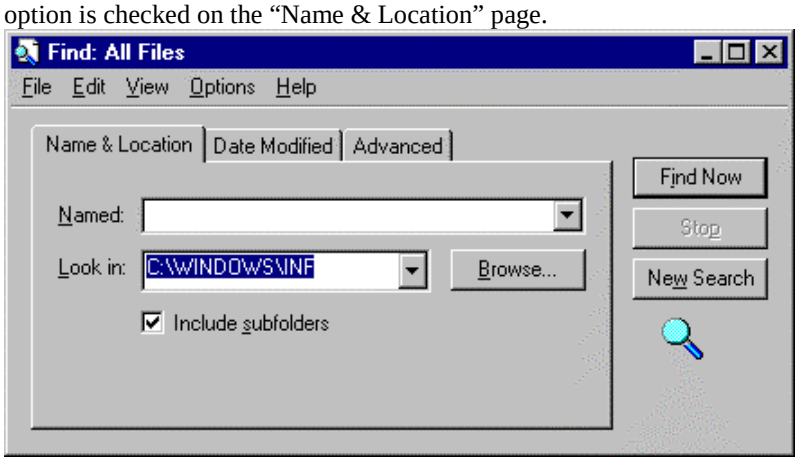
How to disable the DriverLINX Setup Information files in Windows 95/98/Me/2000

Windows saves a copy of all Setup Information files (.INF) that it has ever used to install device drivers. Deleting the .INF file will ensure that Windows does not attempt reinstall the DriverLINX driver the next time you insert the card. Unfortunately, the different versions of Windows 95 (retail, service pack 1 (Win 95A), and OEM (Win 95B)) use different naming conventions and storage locations for Setup Information files. The following protocol should work for all versions of Windows.

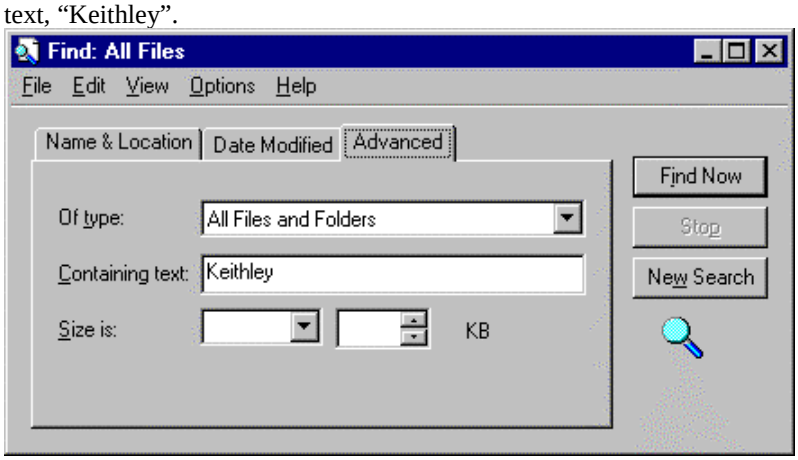
1. Using the My Computer icon, open your “Windows” folder and select the “Inf” folder.



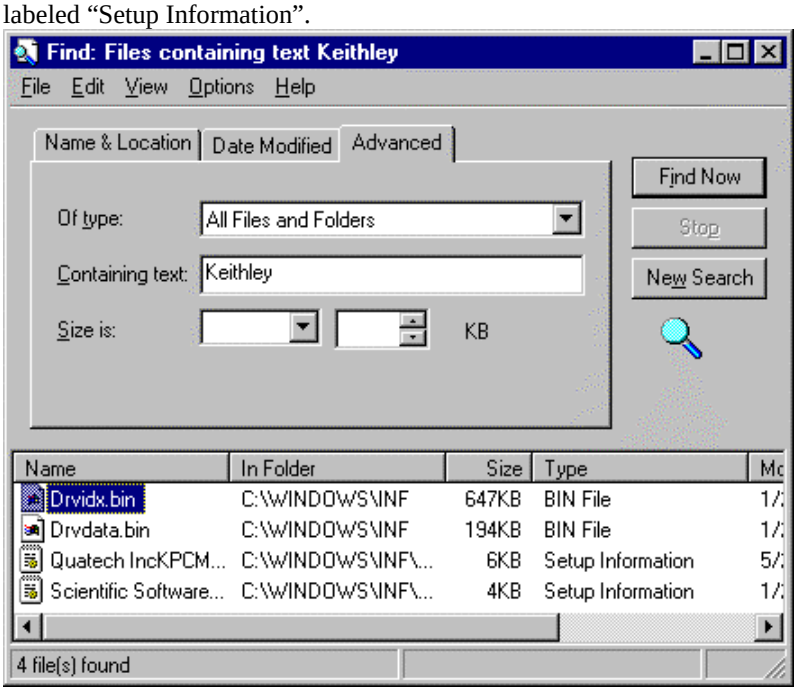
2. Right-click the “Inf” folder and select “Find…” on the pop-up menu.
3. In the “Find: All Files” dialog, make sure the “Include subfolders” option is checked on the “Name & Location” page.



4. Select the “Advanced” tab.
5. Enter “Keithley” in the edit box labeled “Containing text:”.
6. Select “Find Now” and Windows will search for all files containing the text, “Keithley”.

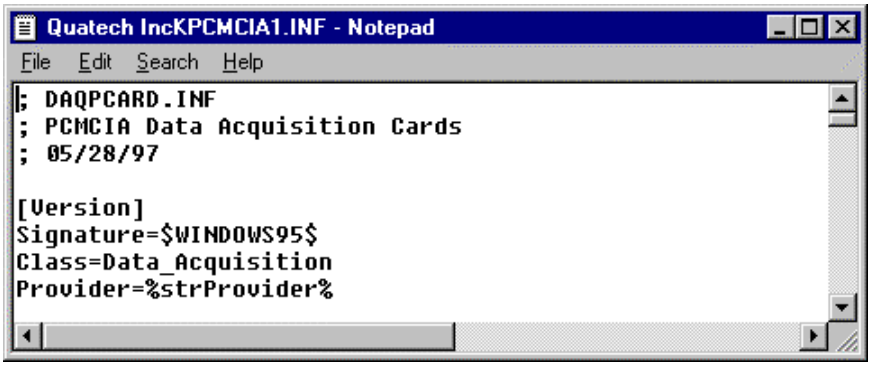


7. The search will return one or more files. Look for files with the type labeled “Setup Information”.

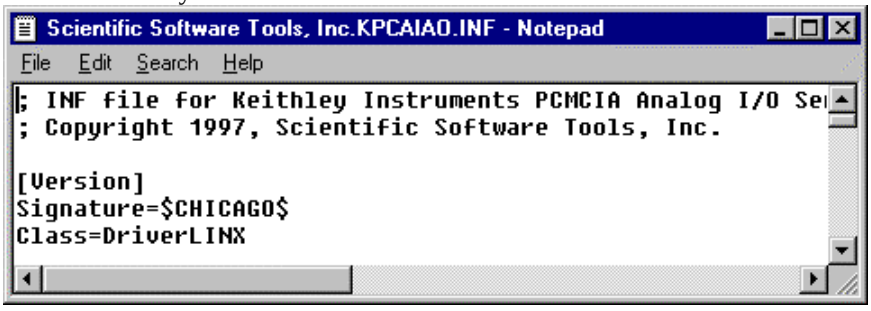


8. Open each Setup Information file by double-clicking it.

9. If the file does ***not*** contain a Scientific Software Tools copyright or the “[Version]” section does ***not*** have an entry, “Class=DriverLINX”, close it and repeat from Step 8 until you’ve examined all Setup Information files.



10. If the file contains a Scientific Software Tools copyright and the “[Version]” section has an entry, “Class=DriverLINX”, then you must determine if this Setup Information file controls the KPCMCIA hardware driver you want to uninstall.



11. Scroll through the file looking for a string name that matches the KPCMCIA card you're removing. If you don't find a match, close Notepad and return to step 8 to examine another Setup Information file.
12. If the Setup Information file controls the KPCMCIA hardware driver you're uninstalling, close Notepad. Next right-click on the file name in the "Find: Files containing text Keithley" dialog. From the pop up menu, select either "Rename" or "Delete" the file. If you rename the file, select an extension other than ".INF".
13. To finish uninstalling, see "How to Remove DriverLINX for the Keithley KPCMCIA Series" on page 77.

How to Remove DriverLINX for the Keithley KPCMCIA 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 KPCMCIA" 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 KPCMCIA 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 KPCMCIA hardware requires successful completion of four steps.

1. Windows finds free resources for the KPCMCIA card.
2. You configure the KPCMCIA drivers using the DriverLINX Configuration Panel.
3. Windows loads the KPCMCIA drivers into memory.

If you are having a problem installing or configuring your KPCMCIA 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 Installing Drivers

On Windows NT, the DriverLINX installation program reliably installs and registers the DriverLINX drivers. On Windows 95/98/Me, the DriverLINX installation program runs a wizard that guides you through installing the DriverLINX drivers. If the wizard did not complete all steps, click here to run it again.

Solving Problems Configuring the Drivers

Windows 95/98/Me/2000 assigns hardware resources for the KPCMCIA, but you must still configure the KPCMCIA drivers before using them. On Windows NT, the DriverLINX configuration requires that you select the hardware model of your KPCMCIA card and manually enter available address and interrupt resource assignments. See “Configure DriverLINX Device Dialog” on page 9 for more information.

Solving Problems Loading Drivers

Before the KPCMCIA drivers can load, you must

1. Install the DriverLINX software.
2. Install the KPCMCIA hardware into your computer.

3. Configure DriverLINX.
4. Reboot your computer.

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

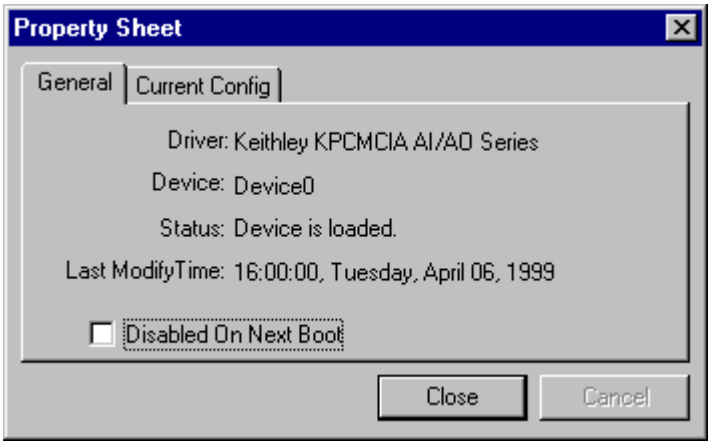
On Windows NT you must determine free hardware resources for the KPCMCIA using Windows NT Diagnostics. On Windows 95/98/Me/2000, the operating system will automatically assign hardware resources to the KPCMCIA cards. Automatic resource assignment can fail sometimes on

- Older PCI computers.
- Computers with ISA cards installed.
- Computers with no free hardware resources.

Sorting through all possibilities can be a challenge due to the sheer number of combinations of 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 KPCMCIA AI/AO 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, you did not configure the driver. See “Configure DriverLINX Device Dialog” on page 9.
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 81.

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/Me/2000 or Windows NT using the DriverLINX Event Viewer.

Windows 95/98/Me/2000 maintains additional driver information in the Device Manager. Also see "Getting More Driver Information on Windows 95/98" on page 81.

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/Me/2000

Windows 95/98/Me/2000 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 Installing Drivers" on page 79.
4. Click the "+" next to "DriverLINX drivers".
5. Does your KPCMCIA product appear in the list? If not, see "Solving Problems Installing Drivers" on page 79.
6. Does the icon next to your KPCMCIA product display an exclamation point (!)? If no, Windows has loaded your KPCMCIA 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. For the KPCMCIA Series a possible problem is "This device cannot find enough free resources that it can use. If you want to use this device, you will need to disable one of the other devices on this system. (Code 12)." In this case, you usually need to free an interrupt, such as by disabling COM2, if unused.
9. The Resources tab will show if Windows detected an unresolvable hardware conflict.

Getting More Driver Information on Windows NT

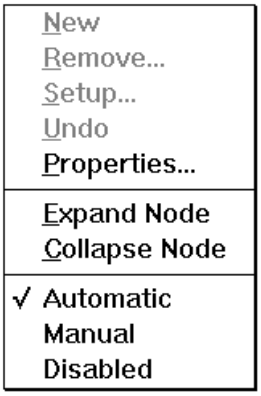
On Windows NT, the only reasons that a driver does not load are

- You did not install the driver software.
- You did not correctly configure the driver.
- You changed the driver startup parameters.

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 81 for more information.

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 KPCMCIA AI/AO 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 79, 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 this shortcut  . 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

Scan list

The list of channels sampled by a task, whether specified as a single channel, or by a range or list.

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 process.

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

A hardware technique that allows a device on the PCI bus to initiate a direct data transfer with memory or another device. The logic for controlling the transfer resides on the PCI device, not the system board. See also “DMA”.

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. See also “bus mastering”.

Event

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

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.

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”.

PCI

Abbreviation for Peripheral Component Interconnect. PCI refers to a specification for a high-speed common local bus on a system board.

PCMCIA

Abbreviation for Personal Computer Memory Card International Association. PCMCIA established a specification for miniaturized cards and sockets for installing peripheral device in portable personal computers called *PC Card*. *PCMCIA* sometimes refers to PC Card.

Process

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

Service Request

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

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.