Keithley KPCMCIA-PIO24

Using DriverLINX with Your Hardware

KEITHLEY

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

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

Keithley KPCMCIA-PIO24: Using DriverLINX with Your Hardware

SST 15-0801-1

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-PIO24 card with DriverLINX.

- 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-PIO24

Shows how to configure the KPCMCIA-PIO24 using the *Configure DriverLINX Device* dialog box.

Using the KPCMCIA-PIO24 with DriverLINX

Shows how to set up DriverLINX with the *Edit Service Request* dialog box to use KPCMCIA-PIO24 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-PIO24

Introduction

This manual explains the steps and special features that apply to Keithley's KPCMCIA-PIO24 card.

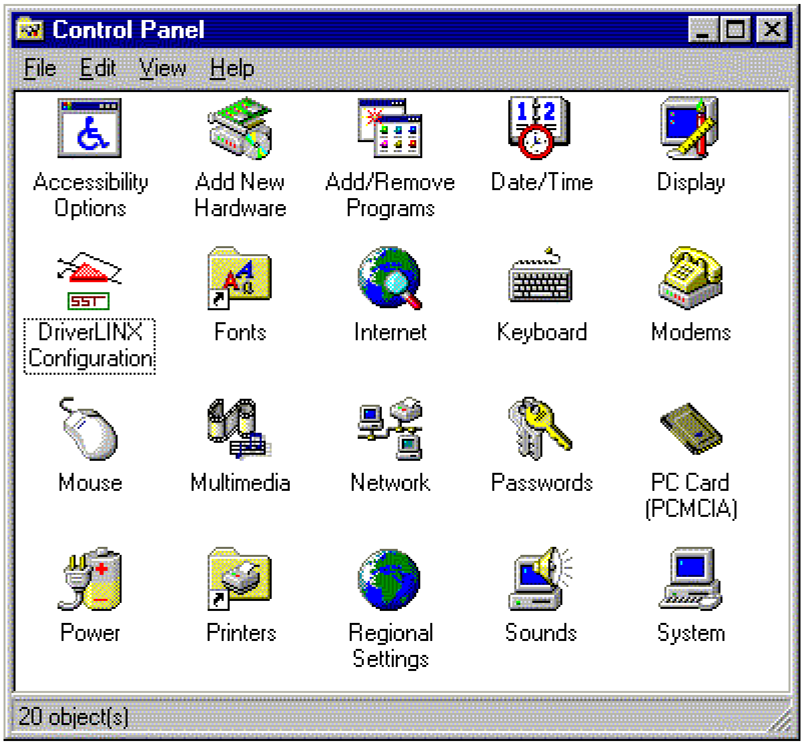
Installing and configuring DriverLINX for the Keithley KPCMCIA-PIO24 card requires three steps:

1. **To install your KPCMCIA-PIO24 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-PIO24 model.

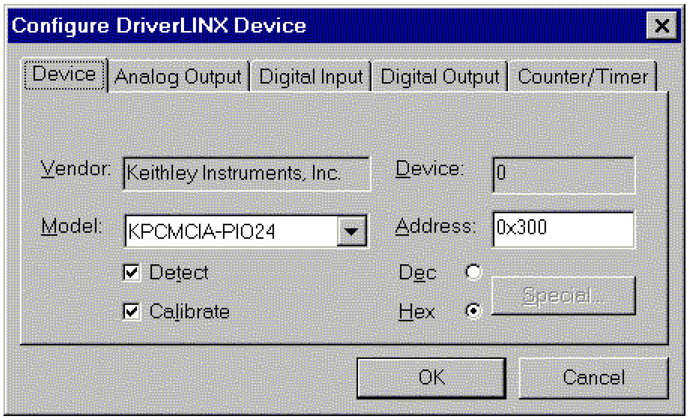
Configure DriverLINX Device Dialog

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-PIO24, you must still follow the configuration process to assign a DriverLINX Logical Device number to a specific KPCMCIA-PIO24 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-PIO24 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



Configure DriverLINX Device

Vendor: Keithley Instruments, Inc.

Device: 0

Model: KPCMCIA-PIO24

Address: 0x300

☑ Detect

☑ Calibrate

○ Dec

● Hex

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

Use the Device subsystem page to tell DriverLINX the model name and address of your KPCMCIA-PIO24 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.

This driver supports only model KPCMCIA-PIO24.

Address

Windows 95/98/Me/2000

Windows 95/98/Me/2000 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 addresses.

Detect

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

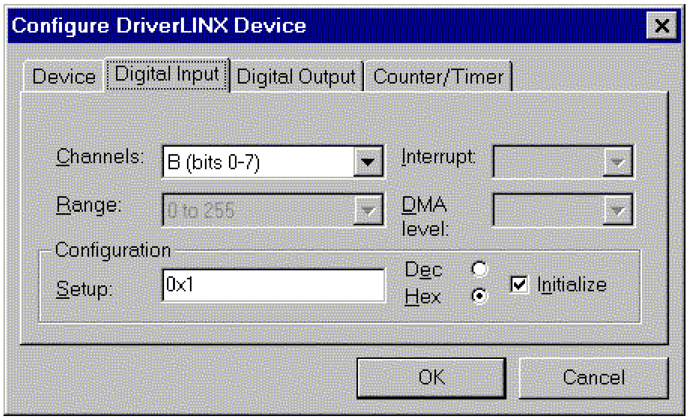
Calibrate

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

Special…

The *Special…* button displays a dialog for configuring any special, hardware-specific settings. For the KPCMCIA-PIO24, Logical Device configuration does not require any special settings so DriverLINX disables the *Special…* button.

Digital Input Subsystem Page



Channels: B (bits 0-7)

Range: 0 to 255

Interrupt:

DMA level:

Configuration

Setup: 0x1

Dec ○

Hex ●

☑ Initialize

Channels

The *Channels* property allows you to select a Logical Channel for configuration or viewing the channel's range. Only the KPCMCIA-PIO24 digital input/output channels support programmable configuration.

DriverLINX defines the following Logical Channels for the KPCMCIA-PIO24 digital inputs:

Logical Channel	DriverLINX Function	KPCMCIA-PIO24 External Connector
0	Digital Input/Output Port	Digital input/output lines PA0 - PA7
1	Digital Input/Output Port	Digital input/output lines PB0 - PB7
2	Digital Input/Output Port	Digital input/output lines PC0 - PC7
3	External Clock	INTERRUPT INPUT

Range

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

Interrupt

Windows 95/98/Me/2000

For Windows 95/98/Me/2000, the operating system automatically determines the interrupt channel for the KPCMCIA-PIO24 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-PIO24 does not use system DMA channels. DriverLINX disables this property and displays it as blank.

Configuration Setup

Use caution when configuring and connecting lines to the digital I/O ports. Connecting an input line to an output port, or vice versa, could damage the hardware.

The *Configuration Setup* property specifies the hardware configuration of the digital I/O ports. The KPCMCIA-PIO24 allows configuring each line of the 8-bit I/O ports for either input or output.

DriverLINX supports two methods for configuring a digital I/O port. The simplified method configures the whole port, while the advanced method allows you to configure groups of digital I/O lines. Select the channel to configure in the drop-down channel list and enter a *Setup* value using one of the methods below.

Simplified Digital I/O Port Configuration

For simplified configuration:

1. Select the channel number of the port to configure in the drop-down channel list.
2. Check the Initialize box.
3. Enter one of the following values for the *Setup* property to configure the entire port:
   - **1**—configures all lines in the port as input, or
   - **0**—configures all lines in the port as output.

Advanced Digital I/O Port Configuration

With advanced configuration, you can divide the port into groups of 1, 2, 4 or 8 lines and configure each group separately.

For advanced configuration:

1. Select the channel number of the port to configure in the drop-down channel list.
2. Check the Initialize box.
3. Enter a value for the *Setup* property as described below.

The following table describes each grouping. Choose whichever grouping is most convenient.

Grouping	Number of Lines in Each Group	Number of Groups in the Port
Byte	8	1
Nibble	4	2
Half-Nibble	2	4
Bit	1	8

For advanced configuration, the *Setup* property value has two variable-sized components:

- **Direction Field**—consists of a bit for each group in the port. Each bit indicates the input/output direction of the lines in the group. The least-significant bit configures the first group. High-order bits to the left configure subsequent groups.

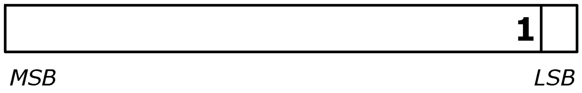
  For each bit, enter one of the following values to configure the group:

  - **1**—configures all lines in the group as input, or
  - **0**—configures all lines in the group as output.

- **Group Size Field**—consists of a binary 1 immediately to the left of the Direction field. This allows the position of the *Setup* property's leftmost 1 to indicate the start of the Direction field.

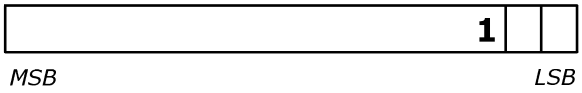
The following diagrams show the format of the *Setup* property for each grouping. All formats have the Group Size field in the high-order bits and the Direction field in the low-order bits.

Byte-Sized Groups



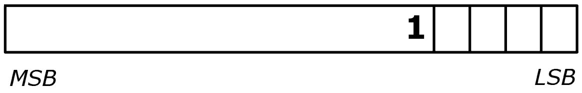
- **Direction Field**—has one bit to configure the single byte-sized group
- **Group Size Field**—has a value of 0x2 (hexadecimal) or 2 (decimal)

Nibble-Sized Groups



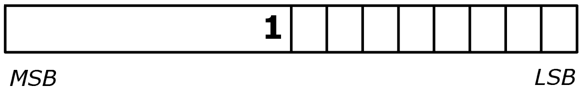
- **Direction Field**—has two bits to configure the two nibble-sized groups
- **Group Size Field**—has a value of 0x4 (hexadecimal) or 4 (decimal)

Half Nibble-Sized Groups



- **Direction Field**—has four bits to configure the four half nibble-sized groups
- **Group Size Field**—has a value of 0x10 (hexadecimal) or 16 (decimal)

Bit-Sized Groups



- **Direction Field**—has eight bits to configure the eight bit-sized groups

- **Group Size Field**—has a value of 0x100 (hexadecimal) or 512 (decimal)

Advanced Configuration Example:

This example shows how to configure the port in nibbles. It configures the first nibble as output and the second as input.

1. Choose the Nibble-Sized format and fill in the Direction field, right-to-left, as follows:

    0—to configure the first nibble as output

    1—to configure the second nibble as input

	1	**1**	**0**
MSB			*LSB*

2. Calculate the hexadecimal or decimal value:

    0x4 + 0x2 + 0x0 = 0x6

    4 + 2 + 0 = 6

3. Then, enter *0x6* (hexadecimal) or *6* (decimal) for the *Setup* property.

Initialize

Checking the *Initialize* check box instructs DriverLINX to use the *Configuration Setup* property to configure the digital I/O ports. Check *Initialize* to put the configuration setup into effect.

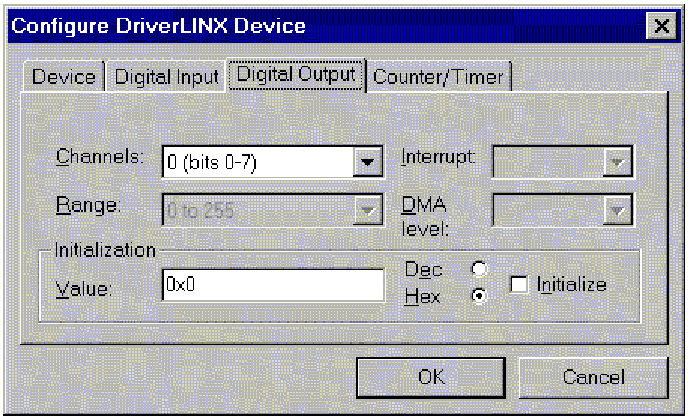
Dec

This check box converts the *Configuration Setup* property to decimal.

Hex

This check box converts the *Configuration Setup* property to hexadecimal.

Digital Output Subsystem Page



Channels: 0 (bits 0-7)

Interrupt:

Range: 0 to 255

DMA level:

Initialization

Value: 0x0

Dec ○

Hex ●

□ Initialize

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-PIO24 card has three digital input/output ports. The Digital Output subsystem shares these ports with the Digital Input subsystem. The KPCMCIA-PIO24 requires configuration of each line in the digital input/output ports as input or output. See "Digital Input Subsystem Page" on page 13.

Range

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

Interrupt

The KPCMCIA-PIO24 uses the same interrupt for digital output as for digital input. Go to the Digital Input page to set it. DriverLINX disables this property and displays it as blank.

DMA level

The KPCMCIA-PIO24 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-PIO24, 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

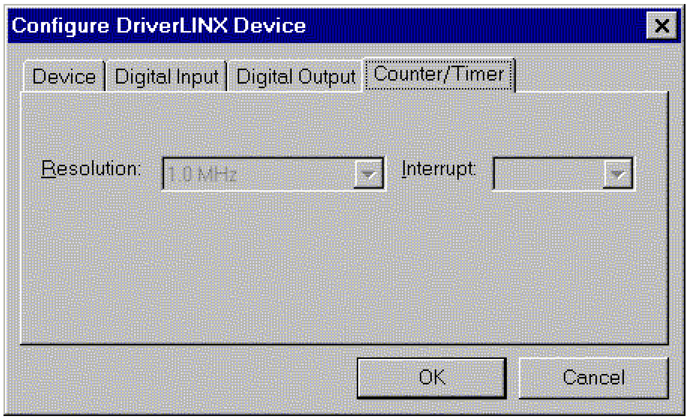
This check box converts the *Initialization Value* property to decimal.

Hex

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

Counter/Timer Subsystem Page

For the KPCMCIA-PIO24, there are no configurable options on the Counter/Timer subsystem page.



Resolution

The *Resolution* property specifies the clock frequency of the master oscillator. DriverLINX provides a System Pacer Clock which simulates a 1 MHz hardware clock for pacing input/output tasks.

Interrupt

The KPCMCIA-PIO24 does not support interrupts from counter/timers. DriverLINX disables this property and displays it as blank.

Using the KPCMCIA-PIO24 with DriverLINX

Introduction

This chapter shows you how to set up and use KPCMCIA-PIO24 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-PIO24

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-PIO24 hardware capabilities.

The following sections describe how DriverLINX implements KPCMCIA-PIO24 hardware features as Subsystems, Modes, Operations, Events, Logical Channels, Buffers, and Messages.

DriverLINX Subsystems

The KPCMCIA-PIO24 supports four of DriverLINX's six Logical Subsystems:

1. **Device**—refers to the KPCMCIA-PIO24 card as a whole.
2. **Analog Input**—refers to the analog input channels, clocks, and control signals. *The KPCMCIA-PIO24 does not support analog input.*
3. **Analog Output**—refers to the analog output channels, clocks, and control signals. *The KPCMCIA-PIO24 does not support analog output.*
4. **Digital Input**—refers to the 8-bit digital input/output ports as well as the 1-bit digital input (TTL) control signals, INTERRUPT INPUT.

5. **Digital Output**—refers to the 8-bit digital input/output ports.
6. **Counter/Timer**—refers to the clock channel for pacing input/output tasks.

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-PIO24 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-PIO24 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-PIO24.

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

KPCMCIA-PIO24 Supported DriverLINX Modes

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-PIO24. Latter sections for each DriverLINX subsystem will describe the operations and events in more detail.

Note: All subsystems allow the *MESSAGE* operation, which is not shown in the table. DriverLINX allows any Mode setting for the *MESSAGE* operation.

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

Allowed Operations and Events for KPCMCIA-PIO24 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

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-PIO24

The Keithley hardware manual describes the data and control signals for the KPCMCIA-PIO24 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 and trigger inputs.

Digital Input Subsystem Signals

The Digital Input subsystem has three 8-bit digital input/output ports and one control input which DriverLINX models as a 1-bit logical digital input port. DriverLINX maps these signals to Logical Channels as shown in the following table:

Port	Connector Name	Logical Channels
Digital Input/Output Port A	PA0 – PA7	0
Digital Input/Output Port B	PB0 – PB7	1
Digital Input/Output Port C	PC0 – PC7	2
External clock alias	INTERRUPT INPUT	3

How DriverLINX maps digital input hardware channels to Logical Channels

Note: To use a line in an 8-bit digital input/output port for input, first configure it for input on the Digital Input Subsystem Page of the *DriverLINX Configuration Panel*. See “Configuring the KPCMCIA-PIO24” on page 9.

The Digital Input subsystem shares the System Pacer Clock with the Digital Output subsystem. DriverLINX designates the System Pacer Clock as Logical Channel 0 of the Counter/Timer subsystem.

Digital Output Subsystem Signals

The Digital Output subsystem has three 8-bit digital input/output ports. DriverLINX maps these signals to Logical Channels as shown in the following table:

Port	Connector Name	Logical Channels
Digital Input/Output Port A	PA0 – PA7	0
Digital Input/Output Port B	PB0 – PB7	1
Digital Input/Output Port C	PC0 – PC7	2

How DriverLINX maps digital output hardware channels to Logical Channels

Note: To use a line in an 8-bit digital input/output port for output, first configure it for output on the Digital Input Subsystem Page of the *DriverLINX Configuration Panel*. See “Configuring the KPCMCIA-PIO24” on page 9.

The Digital Output subsystem shares the System Pacer Clock with the Digital Input subsystem. DriverLINX designates the System Pacer Clock as Logical Channel 0 of the Counter/Timer subsystem.

Counter/Timer Subsystem Signals

DriverLINX provides a system timer using software to pace digital input/output. DriverLINX maps the timer to Logical Channel 0.

Timer	Connector Name	Logical Channels
System timer	none	0

How DriverLINX maps counter/timer hardware channels to Logical Channels

Applications can’t use this timer independently from tasks on an input/output subsystem.

Device Subsystem

The following sections describe how DriverLINX implements Device Subsystem features for the KPCMCIA-PIO24.

Device Modes

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

Device Operations

The KPCMCIA-PIO24 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.

Digital Input Subsystem

The following sections describe how DriverLINX implements Digital Input Subsystem features for the KPCMCIA-PIO24.

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-PIO24 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-PIO24 has three shared digital input/output ports that require configuration before connecting signals. Therefore, DriverLINX does not allow the application to configure the KPCMCIA-PIO24's digital I/O ports.

Note: To use a line in an 8-bit digital input/output port for input, first configure it for input on the Digital Input Subsystem Page of the *DriverLINX Configuration Panel*. See "Configuring the KPCMCIA-PIO24" on page 9.

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-PIO24 reads the next digital input sample from the port.

The KPCMCIA-PIO24 supports the following Timing Events:

- **None**—Input requires no pacing as DriverLINX is reading only a single value.
- **Rate**—The KPCMCIA-PIO24 supports only fixed rate digital input using internal and external clocks.
- **Digital**—DriverLINX uses an external digital input signal to pace the acquisition of each sample.

None or Null Event

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

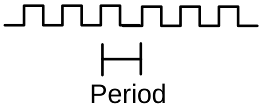
Rate Event

The KPCMCIA-PIO24 supports one type of Rate Event for digital input:

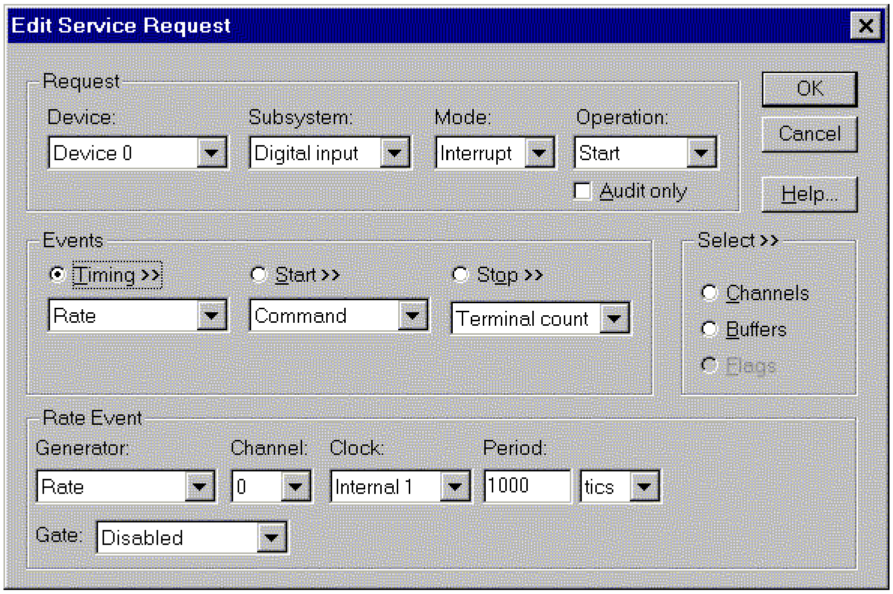
- **Rate Generator**—Generates a fixed rate clock with equal time intervals between tics.

Rate Generator: Internal Clocking

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



As the KPCMCIA-PIO24 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.



Edit Service Request

Request

Device: Device 0

Subsystem: Digital input

Mode: Interrupt

Operation: Start

□ Audit only

OK

Cancel

Help...

Events

● Timing >> Rate

○ Start >> Command

○ Stop >> Terminal count

Select >>

○ Channels

○ Buffers

○ Flags

Rate Event

Generator: Rate

Channel: 0

Clock: Internal 1

Period: 1000 tics

Gate: Disabled

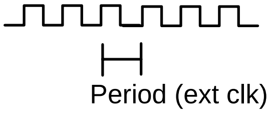
How to set up the KPCMCIA-PIO24 for fixed rate sampling using an internal clock.

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

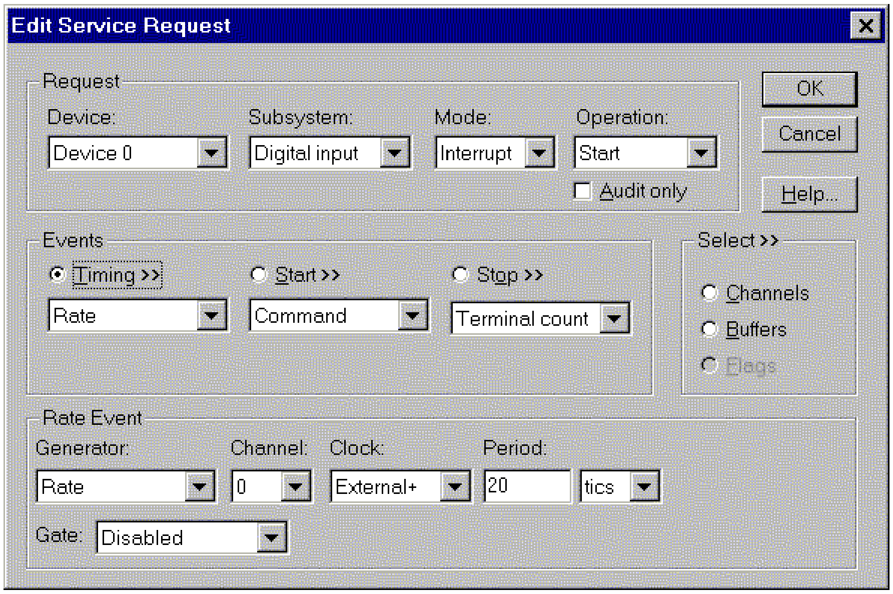
- Specify internal clocking using a **Rate** *Generator* on *Logical Channel* **0** with the **Internal 1** *Clock* source.
- The *Period* property specifies the time interval between samples in tics, where a system timer tic is 1 µs, or 1 MHz. The minimum period is 1000 tics, or 1 kHz. The maximum period 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 55 for details on the system timer.

Rate Generator: External Clocking

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



Use an externally clocked rate generator when you want to synchronize digital input samples with a recurrent external signal.



Request

Device: Device 0

Subsystem: Digital input

Mode: Interrupt

Operation: Start

□ Audit only

Events

● Timing >> Rate

○ Start >> Command

○ Stop >> Terminal count

Select >>

○ Channels

○ Buffers

○ Flags

Rate Event

Generator: Rate

Channel: 0

Clock: External+

Period: 20 tics

Gate: Disabled

How to set up the KPCMCIA-PIO24 for fixed rate sampling using an external clock.

A task cannot use the INTERRUPT INPUT line (Digital Input Logical Channel 3) for a digital start or stop event while using it as an external clock source.

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

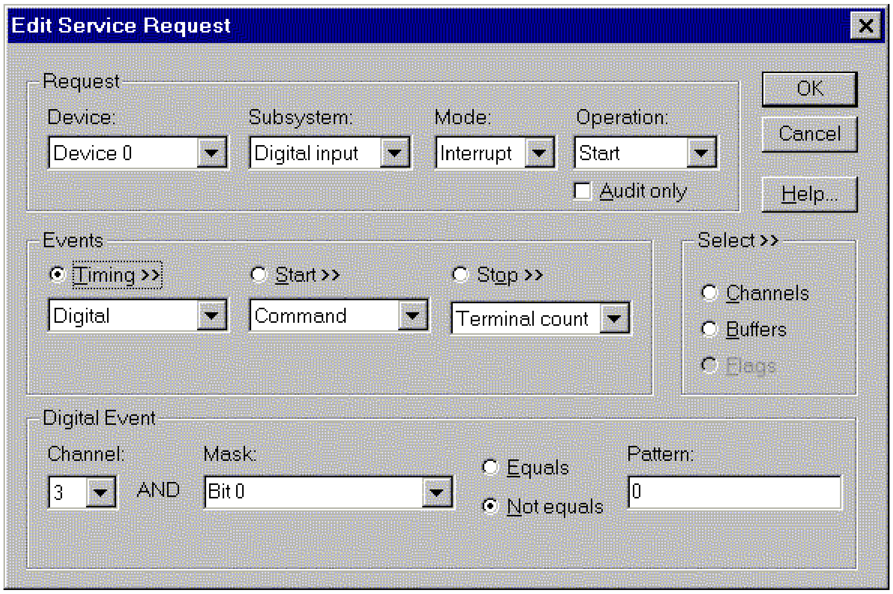
- Specify external clocking using a **Rate** *Generator* on *Logical Channel* **0** with an **External, External+** or **External–** *Clock* source. External and External+ both specify sampling on the rising, or positive, edge of the external clock signal. External**–** specifies sampling on the falling, or negative, edge of the external clock signal.

- Users should connect the external clock signal to the INTERRUPT INPUT line.

- The *Period* may be any value ≥ 20 tics, or 20 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.

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 externally clock sources.



Edit Service Request

Request

Device: Device 0

Subsystem: Digital input

Mode: Interrupt

Operation: Start

□ Audit only

Events

● Timing >> Digital

○ Start >> Command

○ Stop >> Terminal count

Select >>

○ Channels

○ Buffers

○ Flags

Digital Event

Channel: 3 AND Mask: Bit 0

○ Equals

● Not equals

Pattern: 0

How to set up the KPCMCIA-PIO24 for external rate sampling using a digital event.

Digital Timing 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.

A task cannot use a channel for a digital start or stop event while using it for a digital timing event.

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

- Specify external clocking using *Logical Channel* **2** or **3**. For hardware-independence, you can specify the hardware external trigger channel by the symbolic constant, *DI_EXTCLK*.
- Users should connect the external clock signal to an input line in port C (PC0 - PC7) for channel 2 or the INTERRUPT INPUT line for channel 3.
- Specify the *Mask* property with a single bit set 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 (≠1), or with the same bit set as in the mask for a falling, or negative, edge clock (≠0).

Digital Input Start Events

Start Events specify when the KPCMCIA-PIO24 hardware starts reading digital input data.

The KPCMCIA-PIO24 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-PIO24 hardware for the task.
- **Digital**—The KPCMCIA-PIO24 starts acquiring digital input samples when the hardware detects the digital Logical Channel input satisfies the digital condition specified in the Start Event.

None or Null Event

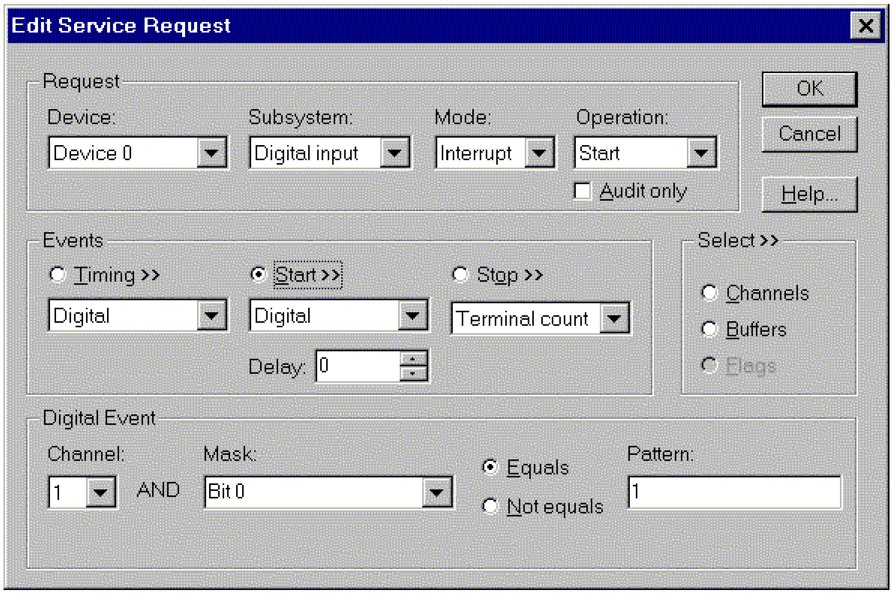
The Null Event specifies that the task does not need a Start Event to begin the task.

Command Event

The Command Event starts data acquisition as soon as DriverLINX has completed programming the KPCMCIA-PIO24 hardware with the task parameters.

Digital Event or Post-Triggering

The KPCMCIA-PIO24 can acquire digital input samples ***after*** the hardware detects a digital trigger condition. Use post-triggering when you want to synchronize the start of data acquisition with an external signal.



Request
Device: Device 0
Subsystem: Digital input
Mode: Interrupt
Operation: Start
□ Audit only

Events
○ Timing >> Digital
● Start >> Digital
Delay: 0
○ Stop >> Terminal count

Select >>
○ Channels
○ Buffers
○ Flags

Digital Event
Channel: 1 AND Mask: Bit 0
● Equals
○ Not equals
Pattern: 1

How to set up the KPCMCIA-PIO24 for post-triggered digital input.

A Digital Start Event requires an external timing source because the KPCMCIA-PIO24 does not have a hardware clock.

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.

A task cannot use a channel for a digital start or stop event while using it for a digital timing event.

- Specify the *Logical Channel* as **0, 1, 2** or **3**.
- Specify the *Mask* property to indicate that DriverLINX should only compare selected bits in the digital input value against the *Pattern* property.

- Specify the *Match* property as **Equals** to trigger on an exact match of the digital pattern to the masked digital input. Match on **Not Equals** triggers on any difference between the masked digital input and the *Pattern*.
- Specify the *Pattern* property to compare with the masked digital input value.
- Specify the *Delay* property as any number of samples from 0 to $2^{32} - 1$.

Digital Input Stop Events

Stop Events specify when the KPCMCIA-PIO24 hardware stops reading digital input data.

The KPCMCIA-PIO24 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-PIO24 hardware has filled all the data buffers once.
- **Digital—**The KPCMCIA-PIO24 stops the task when the hardware detects the digital Logical Channel input satisfies the digital condition specified in the Stop Event.

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.

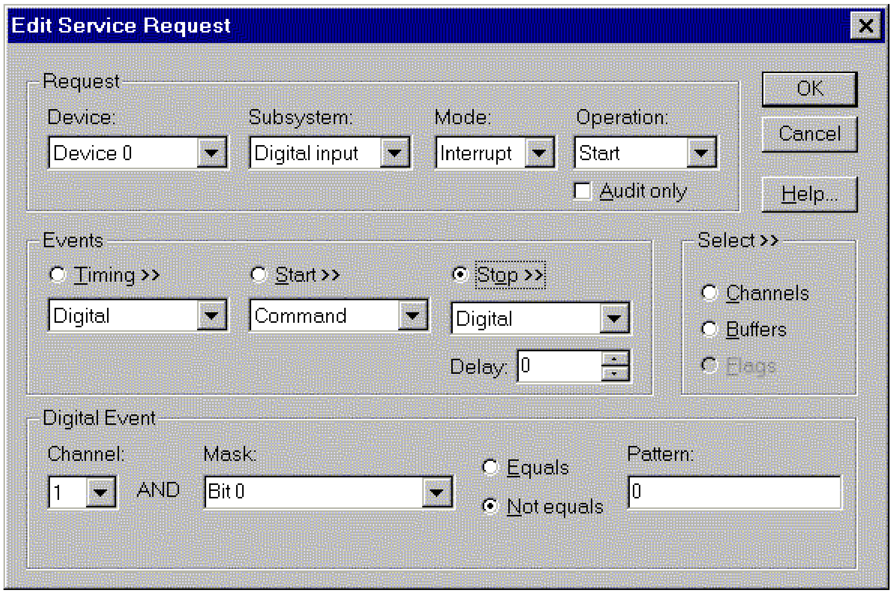
In stop-on-command mode, DriverLINX continuously cycles through all the data buffers, reading from the digital ports on the KPCMCIA-PIO24.

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 Event or Pre-Triggering

The KPCMCIA-PIO24 can acquire digital input samples ***until*** the hardware detects a digital trigger condition. Use pre-triggering when you want to synchronize the end of data acquisition with an external signal.



Edit Service Request

Request

Device: Device 0

Subsystem: Digital input

Mode: Interrupt

Operation: Start

□ Audit only

Events

○ Timing >> ○ Start >> ● Stop >>

Timing: Digital

Start: Command

Stop: Digital

Delay: 0

Select >>

○ Channels
○ Buffers
○ Flags

Digital Event

Channel: 1 AND Mask: Bit 0

○ Equals
● Not equals

Pattern: 0

How to set up the KPCMCIA-PIO24 for pre-triggered digital input.

A Digital Start Event requires an external timing source because the KPCMCIA-PIO24 does not have a hardware clock.

A task cannot use a channel for a digital start or stop event while using it for a digital timing event.

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 **0, 1, 2** or **3**.
- Specify the *Mask* property to indicate that DriverLINX should only compare selected bits in the digital input value against the *Pattern* property.
- Specify the *Match* property as **Equals** to trigger on an exact match of the digital pattern to the masked digital input. Match on **Not Equals** triggers on any difference between the masked digital input and the *Pattern*.
- Specify the *Pattern* property to compare with the masked digital input value.
- Specify the *Delay* property as any number of samples from 0 to $2^{32} - 1$.

Digital Input Channels

The KPCMCIA-PIO24 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 digital data a list of channels.

Digital Input Logical Channels

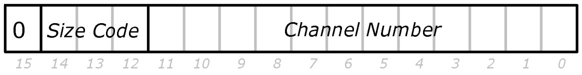
The KPCMCIA-PIO24 has three digital input/output ports that DriverLINX designates as Logical Channels 0 to 2. DriverLINX defines an additional Logical Channel for the external clock signal but applications cannot directly read its value.

Logical Channel	DriverLINX Function	KPCMCIA-PIO24 External Connector
0	Digital Input/Output Port	Digital input/output lines PA0 - PA7
1	Digital Input/Output Port	Digital input/output lines PB0 - PB7
2	Digital Input/Output Port	Digital input/output lines PC0 - PC7
3	External Clock	INTERRUPT INPUT

Extended Logical Channel Addressing

DriverLINX supports a software extension to Logical Channel addressing that allows applications to combine adjacent Logical Channels into a single channel or split a Logical Channel into smaller addressable parts. For instance, applications can address individual bits of a digital port or read and write multiple ports as a single value.

To use the Logical Channel addressing extensions, form a 16-bit Logical Channel address by combining the channel number of an addressable unit with a size field as follows:



The following table specifies the 3-bit size codes:

Size Code	Unit	Bits
0	native	varies with hardware
1	bit	1
2	half nibble	2
3	nibble	4
4	byte	8
5	word	16
6	dword	32
7	qword	64

"Native" units refer to the hardware-defined digital channel size. For most boards, this is the same as an 8-bit byte. When using extended Logical Channel addressing, DriverLINX groups digital bits in units defined by the size code and then assigns consecutive channel numbers starting from zero.

For the KPCMCIA-PIO24, the 8-bit digital input/output ports have the following channel addresses for each size code:

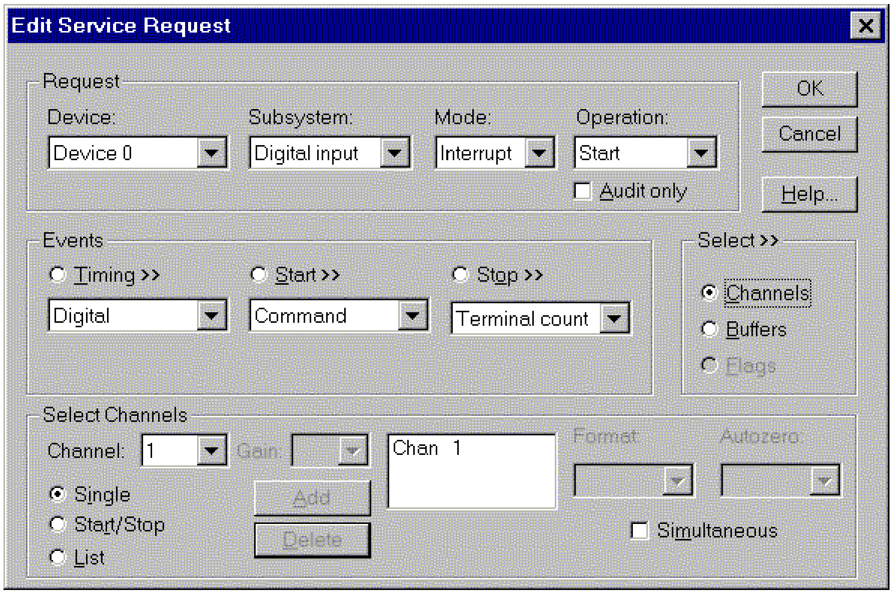
Unit	Channels	Address (dec)	Address (hex)
native	0..2	0..2	0..2
bit	0..23	4096..4119	1000..1017
half nibble	0..11	8192..8203	2000..200B
nibble	0..5	12288..12293	3000..3005
byte	0..2	16384..16386	4000..4002
word	0..0	20480..20480	5000..5000

Notes:

- DriverLINX supports only single-value transfers on Logical Channels that are smaller than the native size.
- Word addressing accesses Ports A and B only.

Single Channel Digital Input

In this mode, the KPCMCIA-PIO24 acquires all data from one channel.



Edit Service Request

Request
Device: Device 0
Subsystem: Digital input
Mode: Interrupt
Operation: Start
□ Audit only

Events
○ Timing >> Digital
○ Start >> Command
○ Stop >> Terminal count

Select >>
● Channels
○ Buffers
○ Flags

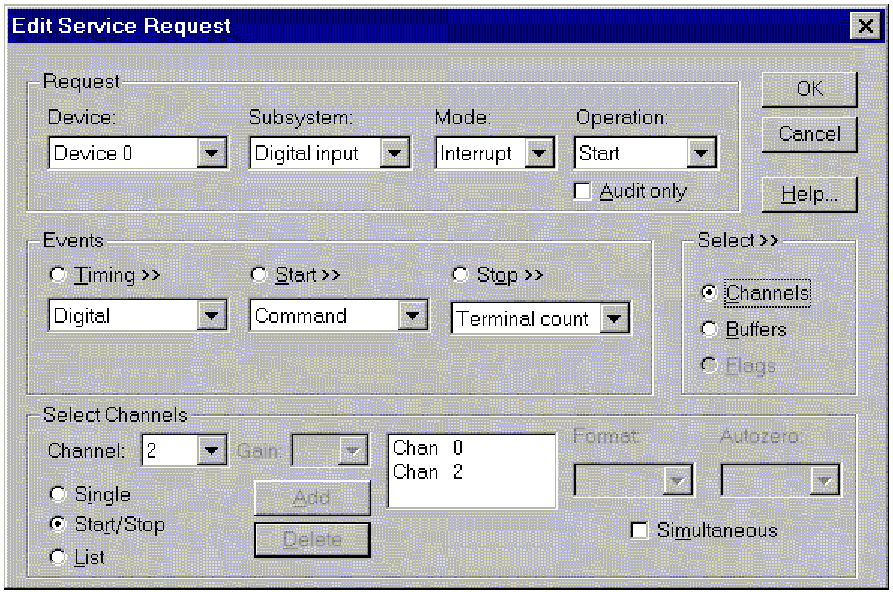
Select Channels
Channel: 1
Gain:
Chan 1
Format
Autozero
● Single
○ Start/Stop
○ List
□ Simultaneous

How to set up the KPCMCIA-PIO24 to read from a single channel

Multi-channel Digital Input Range

In this mode, the KPCMCIA-PIO24 acquires all data from a consecutive range of digital channels.

- For the KPCMCIA-PIO24, the Start Channel number must be less than or equal to the Stop Channel number.


Edit Service Request

Request

Device: Device 0

Subsystem: Digital input

Mode: Interrupt

Operation: Start

□ Audit only

Events

○ Timing >> Digital

○ Start >> Command

○ Stop >> Terminal count

Select >>

● Channels

○ Buffers

○ Flags

Select Channels

Channel: 2

Gain:

Chan 0
Chan 2

Format:

Autozero:

○ Single

● Start/Stop

○ List

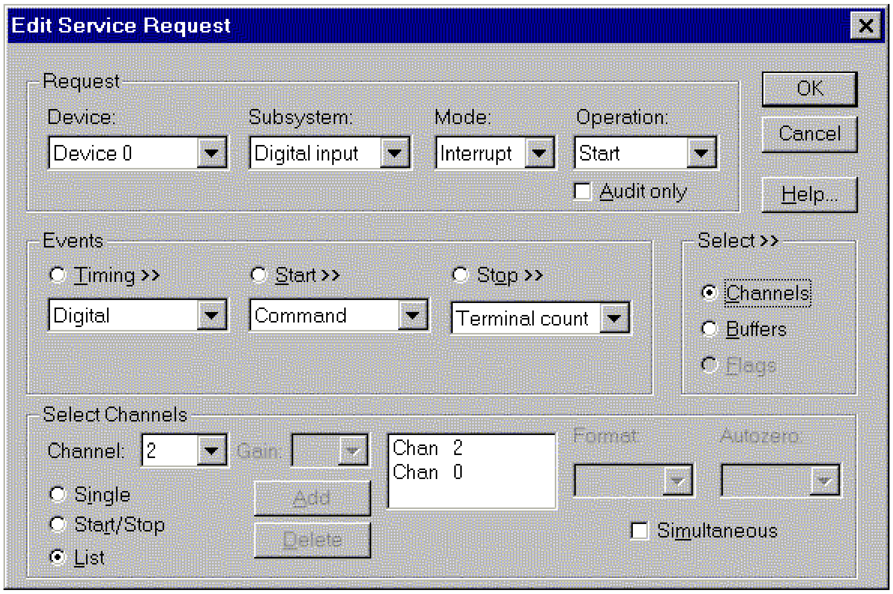
□ Simultaneous

How to set up the KPCMCIA-PIO24 to read from a range of channels

Multi-channel Digital Input List

In this mode, the KPCMCIA-PIO24 acquires all data from a random list of digital channels.

- In simultaneous mode, the list cannot repeat any Logical Channel numbers.


Edit Service Request

Request

Device: Device 0

Subsystem: Digital input

Mode: Interrupt

Operation: Start

□ Audit only

Events

○ Timing >> Digital

○ Start >> Command

○ Stop >> Terminal count

Select >>

● Channels

○ Buffers

○ Flags

Select Channels

Channel: 2

Gain:

Chan 2
Chan 0

Format:

Autozero:

○ Single

○ Start/Stop

● List

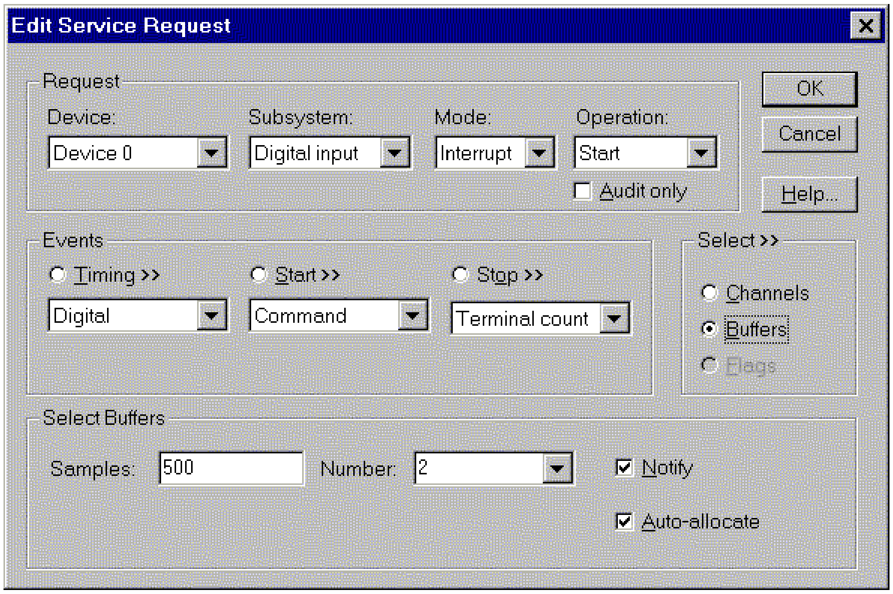
□ Simultaneous

How to set up the KPCMCIA-PIO24 to read from a range of channels

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.



Edit Service Request

Request

Device: Device 0

Subsystem: Digital input

Mode: Interrupt

Operation: Start

□ Audit only

Events

○ Timing >> Digital

○ Start >> Command

○ Stop >> Terminal count

Select >>

○ Channels

● Buffers

○ Flags

Select Buffers

Samples: 500

Number: 2

☑ Notify

☑ Auto-allocate

How to set up the KPCMCIA-PIO24 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
Start Event	DriverLINX has processed the interrupt for a hardware start event
Stop Event	DriverLINX has processed the interrupt for a hardware stop event
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-PIO24.

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-PIO24 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 an 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 Port Configuration

The KPCMCIA-PIO24 has three shared digital input/output ports that require configuration before connecting signals. Therefore, DriverLINX does not allow the application to configure the KPCMCIA-PIO24's digital I/O ports.

Note: To use a line in an 8-bit digital input/output port for output, first configure it for output on the Digital Input Subsystem Page of the *DriverLINX Configuration Panel*. See "Configuring the KPCMCIA-PIO24" on page 9.

Digital Output Initialization

By default, the Digital Output subsystem writes zero into the digital output ports. 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-PIO24 writes the next digital output sample to the ports.

The KPCMCIA-PIO24 supports the following Timing Events:

- **None**—Output requires no pacing as DriverLINX is writing only a single value.
- **Rate**—The KPCMCIA-PIO24 supports only fixed rate digital output using internal and external clocks.
- **Digital**—DriverLINX uses an external digital input signal to pace the output of each sample.

None or Null Event

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

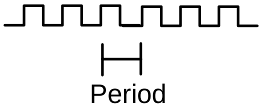
Rate Event

The KPCMCIA-PIO24 supports one type of Rate Event for digital output:

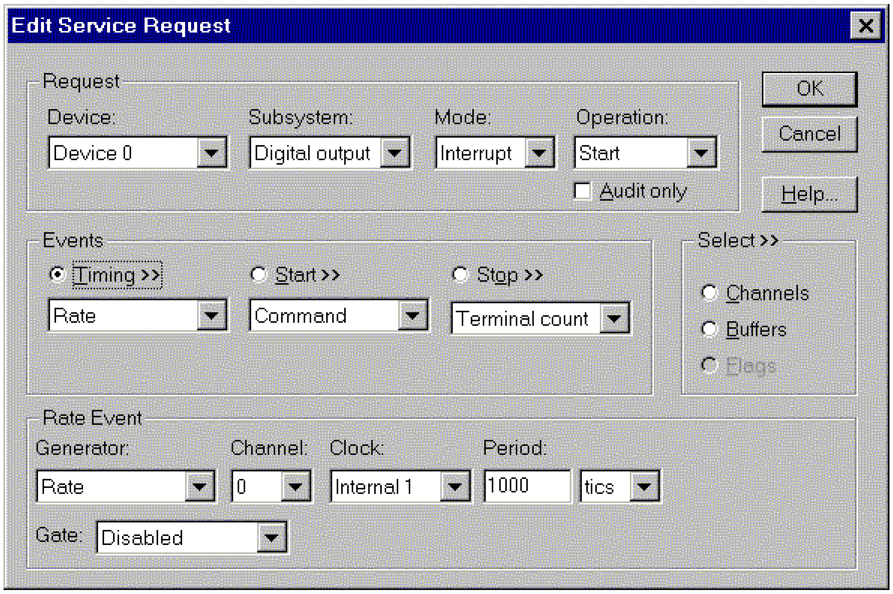
- **Rate Generator**—Generates a fixed rate clock with equal time intervals between tics.

Rate Generator: Internal Clocking

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



As the KPCMCIA-PIO24 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 polling operations.



Edit Service Request

Request
Device: Device 0
Subsystem: Digital output
Mode: Interrupt
Operation: Start
□ Audit only

Events
● Timing >> Rate
○ Start >> Command
○ Stop >> Terminal count

Select >>
○ Channels
○ Buffers
○ Flags

Rate Event
Generator: Rate
Channel: 0
Clock: Internal 1
Period: 1000 tics
Gate: Disabled

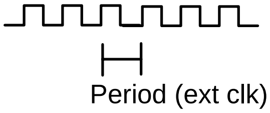
How to set up the KPCMCIA-PIO24 for fixed rate sampling using an internal clock.

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

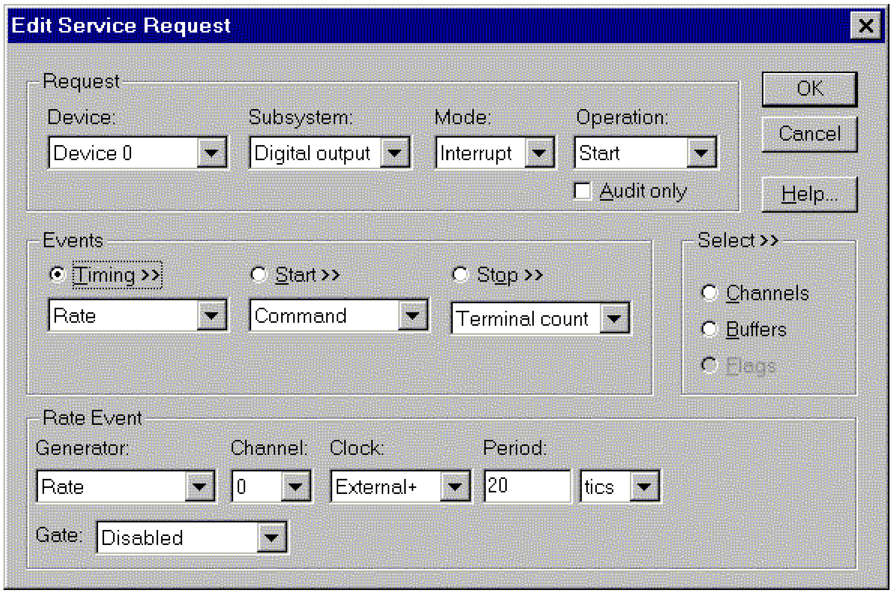
- Specify internal clocking using a **Rate** *Generator* on *Logical Channel* **0** with the **Internal 1** *Clock* source.
- The *Period* property specifies the time interval between samples in tics, where a system timer tic is 1 µs, or 1 MHz. The minimum period is 1000 tics, or 1 kHz. The maximum period 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 55 for details on the system timer.

Rate Generator: External Clocking

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



Use an externally clocked rate generator when you want to synchronize digital output samples with a recurrent external signal.



Edit Service Request

Request

Device: Device 0

Subsystem: Digital output

Mode: Interrupt

Operation: Start

□ Audit only

Events

● Timing >> Rate

○ Start >> Command

○ Stop >> Terminal count

Select >>

○ Channels

○ Buffers

○ Flags

Rate Event

Generator: Rate

Channel: 0

Clock: External+

Period: 20 tics

Gate: Disabled

How to set up the KPCMCIA-PIO24 for fixed rate sampling using an external clock.

A task cannot use the INTERRUPT INPUT line (Digital Input Logical Channel 3) for a digital start or stop event while using it as an external clock source.

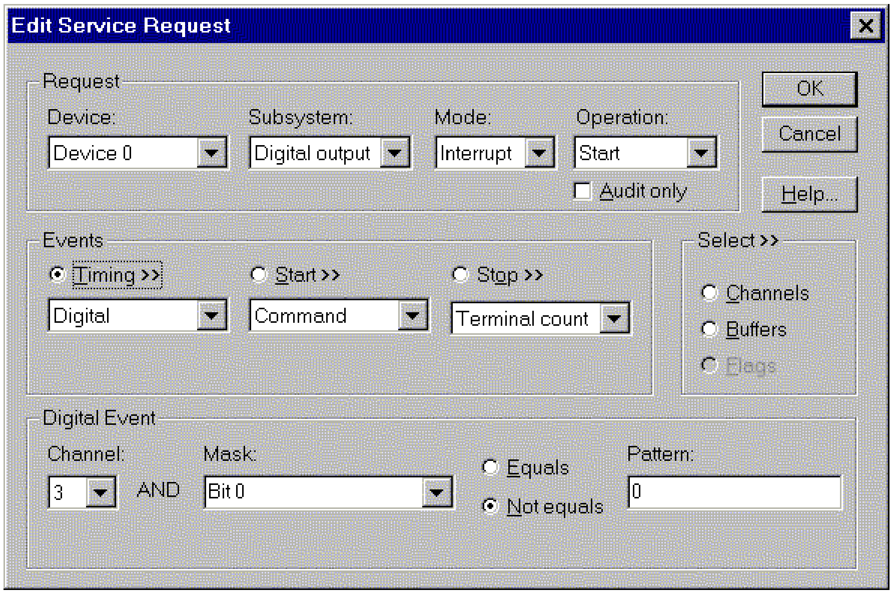
- Specify external clocking using a **Rate** *Generator* on *Logical Channel* **0** with an **External, External+** or **External–** *Clock* source. External and External+ both specify sampling on the rising, or positive, edge of the external clock signal. External**–** specifies sampling on the falling, or negative, edge of the external clock signal.

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

- Users should connect the external clock signal to the INTERRUPT INPUT line.
- The *Period* may be any value ≥ 20 tics, or 20 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.

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 externally clock sources.



Edit Service Request

Request

Device: Device 0

Subsystem: Digital output

Mode: Interrupt

Operation: Start

☐ Audit only

Events

● Timing >> ○ Start >> ○ Stop >>

Digital Command Terminal count

Select >>

○ Channels

○ Buffers

○ Flags

Digital Event

Channel: 3 AND Mask: Bit 0

○ Equals

● Not equals

Pattern: 0

How to set up the KPCMCIA-PIO24 for external rate sampling using a digital event.

Digital Timing 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.

A task cannot use a channel for a digital start or stop event while using it for a digital timing event.

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

- Specify external clocking using *Logical Channel* **2** or **3**. For hardware-independence, you can specify the hardware external trigger channel by the symbolic constant, *DI_EXTCLK*.

- Users should connect the external clock signal to an input line in port C (PC0 - PC7) for channel 2 or the INTERRUPT INPUT line for channel 3.

- Specify the *Mask* property with a single bit set 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 (≠1), or with the same bit set as in the mask for a falling, or negative, edge clock (≠0).

Digital Output Start Events

Start Events specify when the KPCMCIA-PIO24 hardware starts writing digital output data.

The KPCMCIA-PIO24 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-PIO24 hardware for the task.

- **Digital**—The KPCMCIA-PIO24 starts writing digital output samples when the hardware detects the digital Logical Channel input satisfies the digital condition specified in the Start Event.

None or Null Event

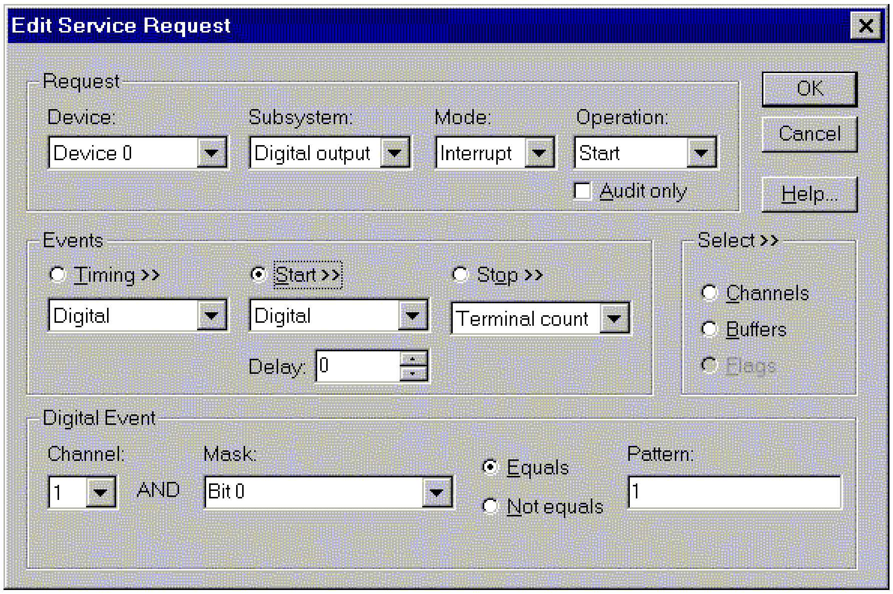
The Null Event specifies that the task does not need a Start Event to begin the task.

Command Event

The Command Event starts data acquisition as soon as DriverLINX has completed programming the KPCMCIA-PIO24 hardware with the task parameters.

Digital Event or Post-Triggering

The KPCMCIA-PIO24 can write digital output samples ***after*** the hardware detects a digital trigger condition. Use post-triggering when you want to synchronize the start of data acquisition with an external signal.



Edit Service Request

Request

Device: Device 0

Subsystem: Digital output

Mode: Interrupt

Operation: Start

□ Audit only

Events

○ Timing >> — Digital

● Start >> — Digital

○ Stop >> — Terminal count

Delay: 0

Select >>

○ Channels

○ Buffers

○ Flags

Digital Event

Channel: 1 AND Mask: Bit 0

● Equals

○ Not equals

Pattern: 1

How to set up the KPCMCIA-PIO24 for post-triggered digital output.

A Digital Start Event requires an external timing source because the KPCMCIA-PIO24 does not have a hardware clock.

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.

A task cannot use a channel for a digital start or stop event while using it for a digital timing event.

- Specify the *Logical Channel* as **0, 1, 2** or **3**.
- Specify the *Mask* property to indicate that DriverLINX should only compare selected bits in the digital input value against the *Pattern* property.
- Specify the *Match* property as **Equals** to trigger on an exact match of the digital pattern to the masked digital input. Match on **Not Equals** triggers on any difference between the masked digital input and the *Pattern*.

- Specify the *Pattern* property to compare with the masked digital input value.
- Specify the *Delay* property as any number of samples from 0 to $2^{32} - 1$.

Digital Output Stop Events

Stop Events specify when the KPCMCIA-PIO24 hardware stops writing digital output data.

The KPCMCIA-PIO24 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-PIO24 hardware has written all the data buffers once.
- **Digital**—The KPCMCIA-PIO24 stops the task when the hardware detects the digital Logical Channel input satisfies the digital condition specified in the Stop Event.

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.

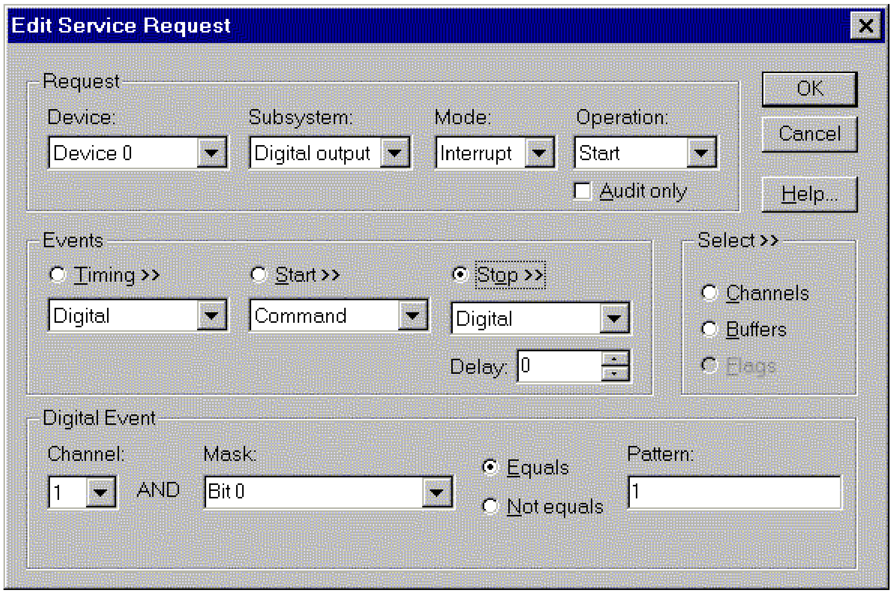
In stop on command mode, DriverLINX continuously cycles through all the data buffers writing to the digital ports on the KPCMCIA-PIO24.

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 Event or Pre-Triggering

The KPCMCIA-PIO24 can write digital output samples ***until*** the hardware detects a digital trigger condition. Use pre-triggering when you want to synchronize the end of data acquisition with an external signal.



Edit Service Request

Request

Device: Device 0

Subsystem: Digital output

Mode: Interrupt

Operation: Start

□ Audit only

Events

○ Timing >> Digital

○ Start >> Command

● Stop >> Digital

Delay: 0

Select >>

○ Channels

○ Buffers

○ Flags

Digital Event

Channel: 1 AND Mask: Bit 0

● Equals

○ Not equals

Pattern: 1

How to set up the KPCMCIA-PIO24 for pre-triggered digital output.

A Digital Start Event requires an external timing source because the KPCMCIA-PIO24 does not have a hardware clock.

A task cannot use a channel for a digital start or stop event while using it for a digital timing event.

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 **0, 1, 2** or **3**.
- Specify the *Mask* property to indicate that DriverLINX should only compare selected bits in the digital input value against the *Pattern* property.
- Specify the *Match* property as **Equals** to trigger on an exact match of the digital pattern to the masked digital input. Match on **Not Equals** triggers on any difference between the masked digital input and the *Pattern*.
- Specify the *Pattern* property to compare with the masked digital input value.
- Specify the *Delay* property as any number of samples from 0 to $2^{32} - 1$.

Digital Output Channels

The KPCMCIA-PIO24 allows applications to specify the digital channels using two techniques:

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

Digital Output Logical Channels

The KPCMCIA-PIO24 has three digital input/output ports that DriverLINX designates as Logical Channels 0 to 2.

Logical Channel	DriverLINX Function	KPCMCIA-PIO24 External Connector
0	Digital Input/Output Port	Digital input/output lines PA0 - PA7
1	Digital Input/Output Port	Digital input/output lines PB0 - PB7
2	Digital Input/Output Port	Digital input/output lines PC0 - PC7

Extended Logical Channel Addressing

DriverLINX supports a software extension to Logical Channel addressing that allows applications to combine adjacent Logical Channels into a single channel or split a Logical Channel into smaller addressable parts. For instance, applications can address individual bits of a digital port or read and write multiple ports as a single value.

To use the Logical Channel addressing extensions, form a 16-bit Logical Channel address by combining the channel number of an addressable unit with a size field as follows:

0	*Size Code*			*Channel Number*											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

The following table specifies the 3-bit size codes:

Size Code	Unit	Bits
0	native	varies with hardware
1	bit	1
2	half nibble	2
3	nibble	4
4	byte	8
5	word	16
6	dword	32
7	qword	64

"Native" units refer to the hardware-defined digital channel size. For most boards, this is the same as an 8-bit byte. When using extended Logical Channel addressing, DriverLINX groups digital bits in units defined by the size code and then assigns consecutive channel numbers starting from zero.

For the KPCMCIA-PIO24, the 8-bit digital input/output ports have the following channel addresses for each size code:

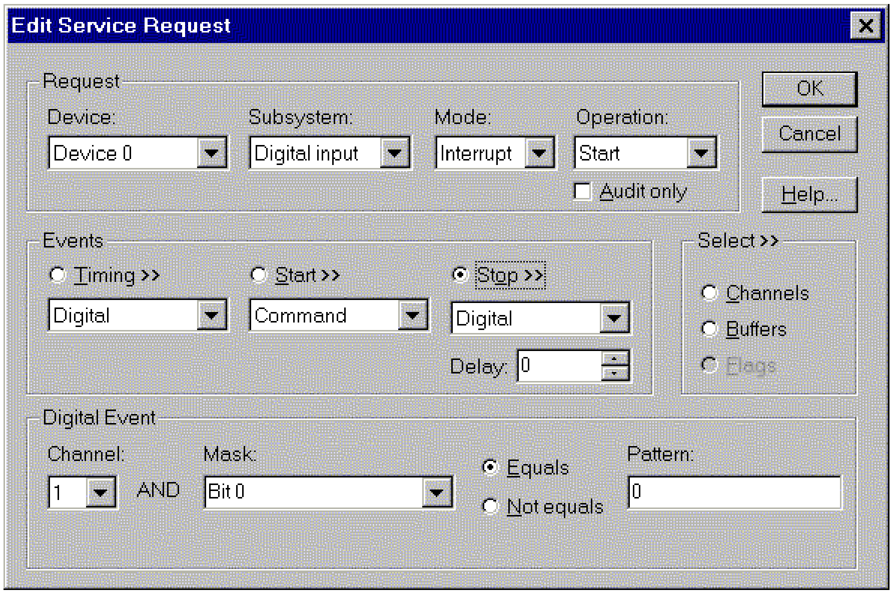
Unit	Channels	Address (dec)	Address (hex)
native	0..2	0..2	0..2
bit	0..23	4096..4119	1000..1017
half nibble	0..11	8192..8203	2000..200B
nibble	0..5	12288..12293	3000..3005
byte	0..2	16384..16386	4000..4002
word	0..0	20480..20480	5000..5000

Notes:

- DriverLINX supports only single-value transfers on Logical Channels that are smaller than the native size.
- Word addressing accesses Ports A and B only.

Single Channel Digital Output

In this mode, the KPCMCIA-PIO24 writes all data to one channel.



Edit Service Request

Request

Device: Device 0

Subsystem: Digital input

Mode: Interrupt

Operation: Start

□ Audit only

Events

○ Timing >> Digital

○ Start >> Command

● Stop >> Digital

Delay: 0

Select >>

○ Channels

○ Buffers

○ Flags

Digital Event

Channel: 1 AND Mask: Bit 0

● Equals

○ Not equals

Pattern: 0

How to set up the KPCMCIA-PIO24 to write a single digital output channel

Multi-channel Digital Output Range

In this mode, the KPCMCIA-PIO24 writes all data to a consecutive range of digital channels.

- If the Start Channel is greater than the Stop Channel, the channel sequence is:

  Start Channel

  …

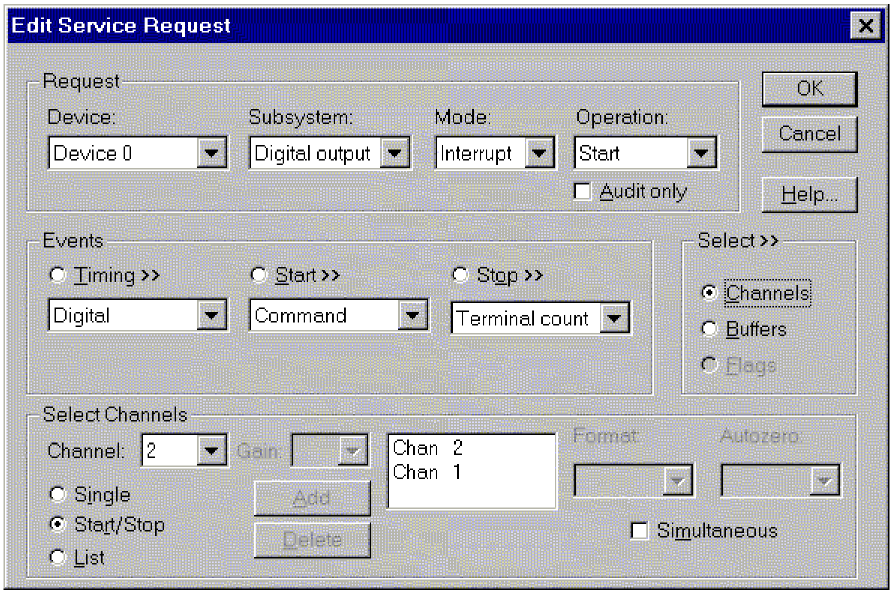
  Logical Channel n-1

Logical Channel 0

…

Stop Channel

where *n* is the number of Logical Channels in the subsystem.



Edit Service Request

Request

Device: Device 0

Subsystem: Digital output

Mode: Interrupt

Operation: Start

□ Audit only

Events

○ Timing >> Digital

○ Start >> Command

○ Stop >> Terminal count

Select >>

● Channels

○ Buffers

○ Flags

Select Channels

Channel: 2

Gain:

Chan 2
Chan 1

○ Single

● Start/Stop

○ List

Format:

Autozero:

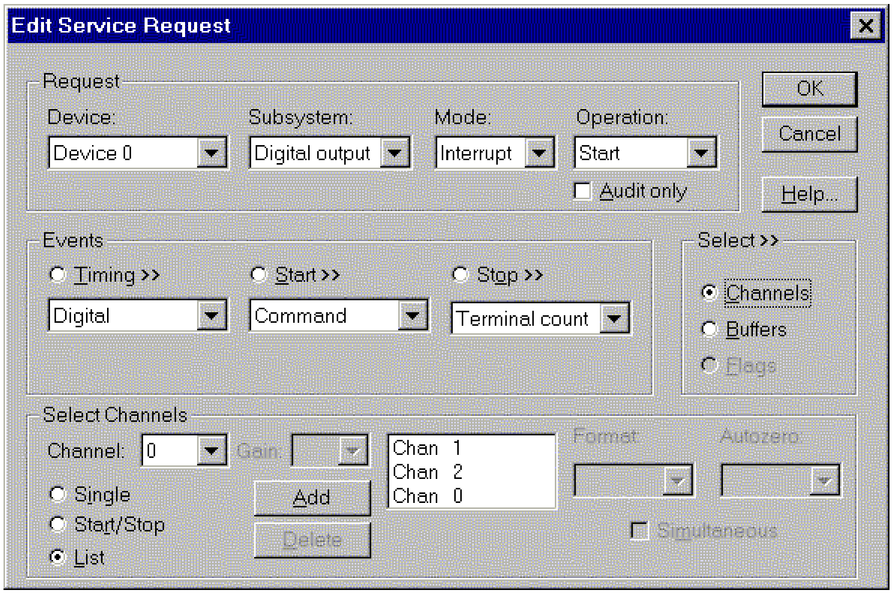
□ Simultaneous

How to set up the KPCMCIA-PIO24 to write a start/stop range of digital output channels

Multi-channel Digital Output List

In this mode, the KPCMCIA-PIO24 writes all data to a random list of digital channels.

- In simultaneous mode, the list cannot repeat any Logical Channel numbers.



Edit Service Request

Request

Device: Device 0

Subsystem: Digital output

Mode: Interrupt

Operation: Start

□ Audit only

Events

○ Timing >> Digital

○ Start >> Command

○ Stop >> Terminal count

Select >>

● Channels

○ Buffers

○ Flags

Select Channels

Channel: 0

Gain:

Chan 1
Chan 2
Chan 0

○ Single

○ Start/Stop

● List

Format:

Autozero:

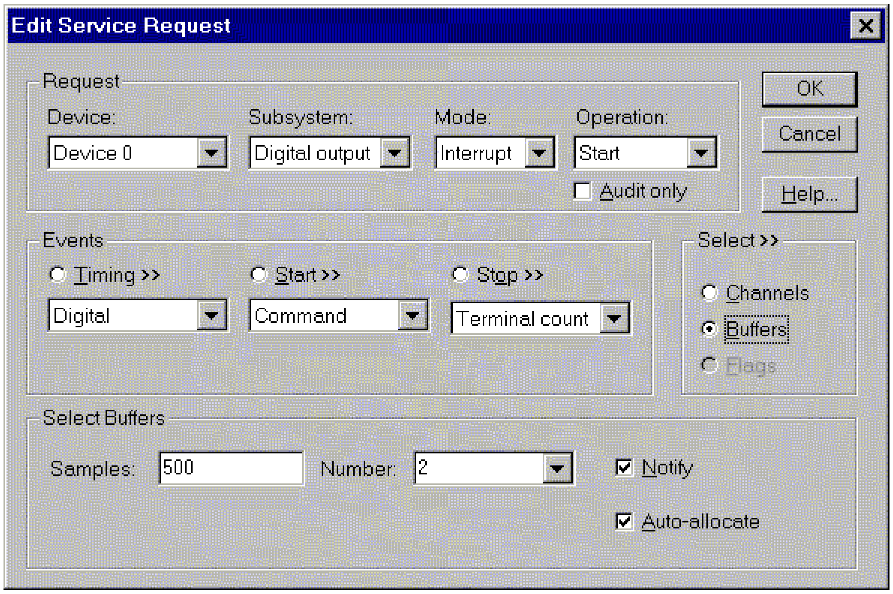
□ Simultaneous

How to set up the KPCMCIA-PIO24 to write a random list of digital output channels

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.



Request

Device: Device 0

Subsystem: Digital output

Mode: Interrupt

Operation: Start

☐ Audit only

Events

○ Timing >> Digital

○ Start >> Command

○ Stop >> Terminal count

Select >>

○ Channels

● Buffers

○ Flags

Select Buffers

Samples: 500

Number: 2

☑ Notify

☑ Auto-allocate

How to set up the KPCMCIA-PIO24 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
Start Event	DriverLINX has processed the interrupt for a hardware start event
Stop Event	DriverLINX has processed the interrupt for a hardware stop event
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 Counter/Timer Subsystem for the KPCMCIA-PIO24 has a single counter/timer, which is for internal use only. DriverLINX defines the counter/timer channel for pacing input/output subsystems but doesn't support independent counter/timer applications.

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 — System	Internal 1	1 µs (1 MHz)	Rate Gen	Disabled No Connect
	External			
	External+			
	External-			

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

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-PIO24 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 msec resolution and granularity. Under current versions of Windows, throughput is less than 1 kHz.

External Clocking

The KPCMCIA-PIO24 allows an external clock source to pace input/output tasks. DriverLINX defines the following external clock sources for the System 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 System Pacer Clock can operates in only the following mode:

- **Rate Generator—**specifies sampling of one channel or one scan at regular intervals based on 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-PIO24 supports counter/timer interrupts indirectly. You can set up an INTERRUPT mode digital 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

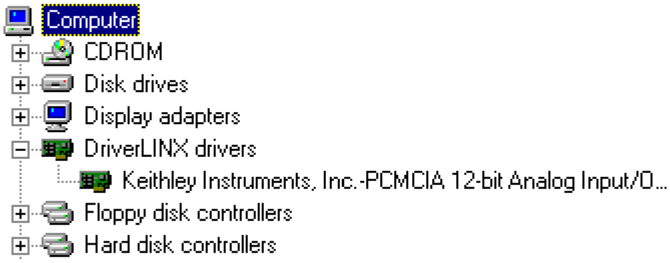
- Disable the DriverLINX driver.
- Shut down your computer to remove the hardware.
- Reboot your computer to unload the driver.
- Run the DriverLINX uninstall program.

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. To finish uninstalling, see .

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.

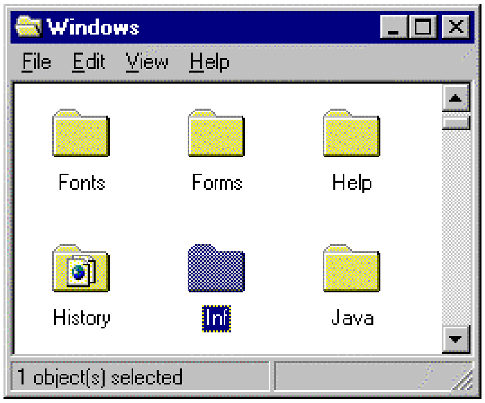


3. Select the KPCMCIA device you want to disable.
4. Click the "Remove" button.
5. In the "Confirm Device Removal" dialog, select "OK".
6. Repeat steps 3-5 for each KPCMCIA card or driver that you uninstalling.
7. When finished, click "Close", shut down your computer, and physically remove any installed KPCMCIA hardware.
8. Reboot Windows.
9. To finish uninstalling, see "How to Remove DriverLINX for the Keithley KPCMCIA Series" on page 61. Also, see "How to disable the DriverLINX Setup Information files in Windows 95/98" on page 58 to ensure Windows does not try to re-install the DriverLINX driver the next time you insert your card.

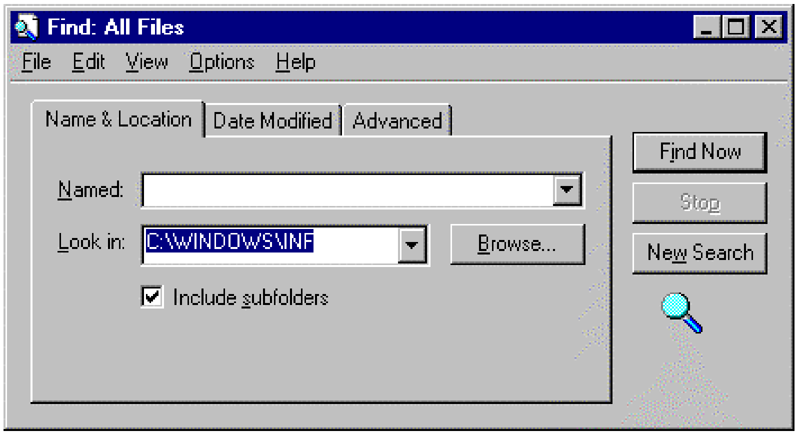
How to disable the DriverLINX Setup Information files in Windows 95/98/Me/2000

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.

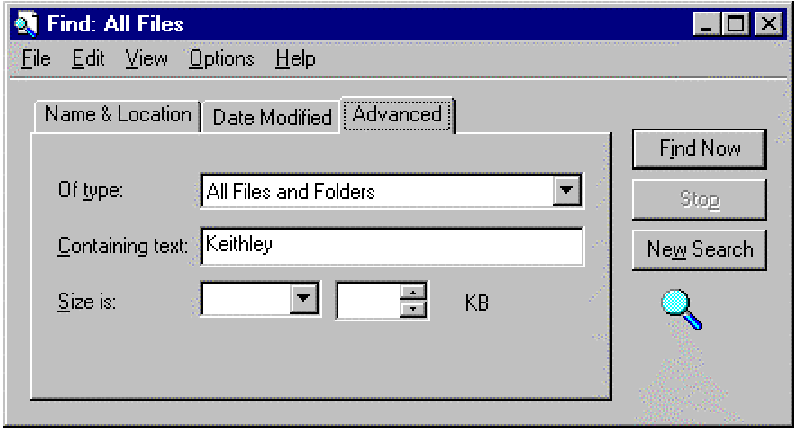


Named:

Look in: C:\WINDOWS\INF

☑ Include subfolders

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”.

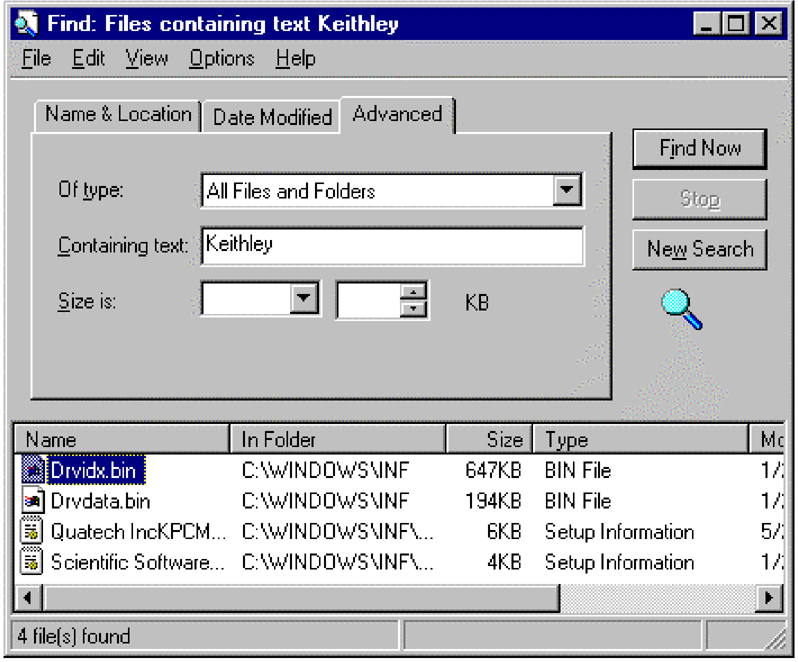


Of type: All Files and Folders

Containing text: Keithley

Size is: KB

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


Find: Files containing text Keithley
File Edit View Options Help
Name & Location
Date Modified
Advanced
Find Now
Stop
New Search

Of type: All Files and Folders

Containing text: Keithley

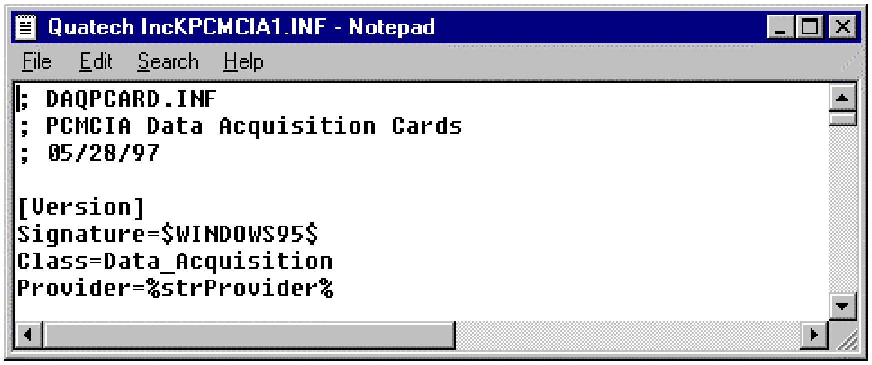
Size is: KB

Name	In Folder	Size	Type	Mo
Drvidx.bin	C:\WINDOWS\INF	647KB	BIN File	1/
Drvdata.bin	C:\WINDOWS\INF	194KB	BIN File	1/
Quatech IncKPCM...	C:\WINDOWS\INF\...	6KB	Setup Information	5/
Scientific Software...	C:\WINDOWS\INF\...	4KB	Setup Information	1/

4 file(s) found

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.

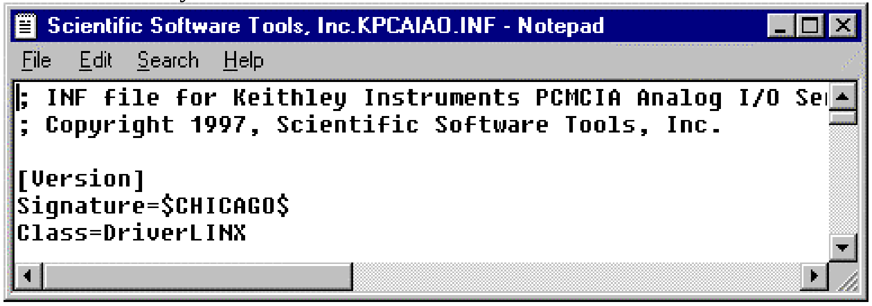


```
; DAQPCARD.INF
; PCMCIA Data Acquisition Cards
; 05/28/97

[Version]
Signature=$WINDOWS95$
Class=Data_Acquisition
Provider=%strProvider%
```

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.



```
; INF file for Keithley Instruments PCMCIA Analog I/O Se
; Copyright 1997, Scientific Software Tools, Inc.

[Version]
Signature=$CHICAGO$
Class=DriverLINX
```

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 61.

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. The DriverLINX configuration requires that you select the hardware model of your KPCMCIA card. And, on Windows NT, you must manually enter the 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.

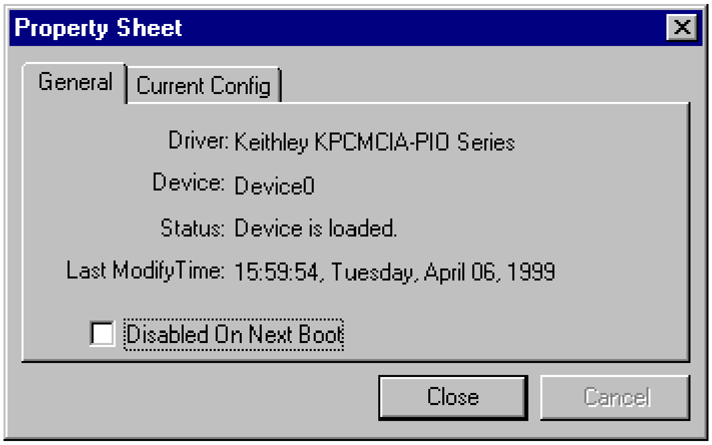
On Windows NT you must determine free hardware resources for the KPCMCIA using Windows NT Diagnostics . On Windows 95/98/Me/2000, the operating system will automatically assign hardware resources to the KPCMCIA cards. Automatic resource assignment can fail sometimes on

- 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-PIO24 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.



Driver: Keithley KPCMCIA-PIO Series

Device: Device0

Status: Device is loaded.

Last ModifyTime: 15:59:54, Tuesday, April 06, 1999

□ Disabled On Next Boot

8. If you rebooted the computer after configuring and Windows did not load your device, see “Checking for Device Errors” on page 64.

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 64.

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 62.
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 62.
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

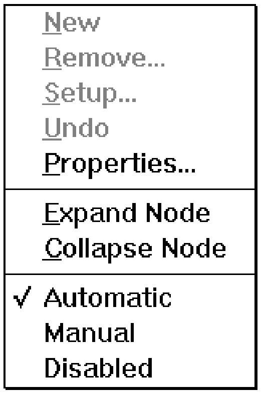
On Windows NT, the only reasons that a driver does not load are

- 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 64 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-PIO24 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 62, 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

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.