Keithley DAS-1700 Series

Using DriverLINX with Your Hardware

KEITHLEY

Keithley DAS-1700 Series: Using DriverLINX with Your Hardware

First Printing.
SST 27-0599-1

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 board to Scientific Software Tools, Inc., or can furnish proof of the purchase of the program. This remedy is void if failure has resulted from accident, abuse, or misapplication. Any replacement will be warranted for the remainder of the original warranty period.

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

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

About DriverLINX

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

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

Included with your DriverLINX package are the following items:

- The DriverLINX API DLLs and drivers supporting your data-acquisition hardware
- Analog I/O Panel, a DriverLINX program that verifies the installation and configuration of DriverLINX for your analog input/output board and demonstrates several virtual bench-top instruments
- Source code for the sample programs
- The DriverLINX Application Programming Interface files for your compiler
- DriverLINX On-line Help System
- *DriverLINX 4.0 Installation and Configuration Guide*
- *DriverLINX Analog I/O Programming Guide*
- *DriverLINX 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 DAS-1700 Series boards with DriverLINX.

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

This manual contains the following chapters:

Configuring the DAS-1700 Series

Shows how to configure the DAS-1700 Series using the *Configure DriverLINX Device* dialog box.

Using the DAS-1700 Series with DriverLINX

Shows how to set up DriverLINX with the *Edit Service Request* dialog box to use DAS-1700 Series hardware features.

Uninstalling DriverLINX

Describes how to remove DriverLINX hardware drivers and other files.

Troubleshooting

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

Conventions Used in This Manual

The following notational conventions are used in this manual:

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

```
Code
```

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

Tables

Concept

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

Configuring the DAS-1700 Series

Introduction

The installation program provides general instructions for installing and configuring DriverLINX. This manual explains the steps and special features that apply to Keithley's DAS-1700 Series boards.

Installing and configuring DriverLINX for a Keithley DAS-1700 Series board requires three steps:

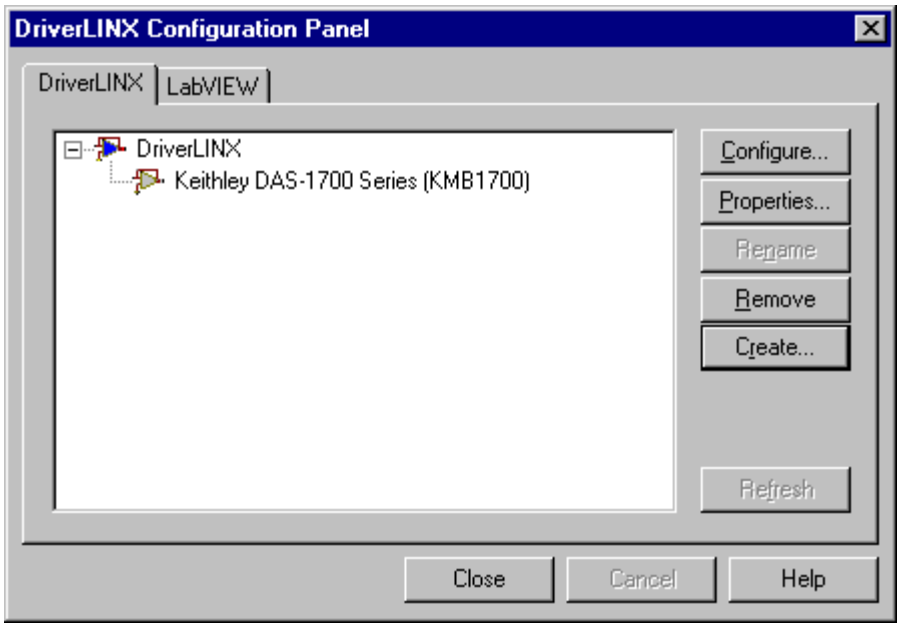
1. **Install DriverLINX.** Follow the instructions given by the installation program. The *Read Me First* instructions explain the components and drivers you can install.
2. **Configure DriverLINX.** This creates a Logical Device, which stores configuration information for your board. See "Configure DriverLINX Device Dialog" on page 11 for configuration options specific to a Keithley DAS-1700 Series model.
3. **Install your DAS-1700 hardware.** Follow the instructions in your hardware manual.

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

Configure DriverLINX Device Dialog

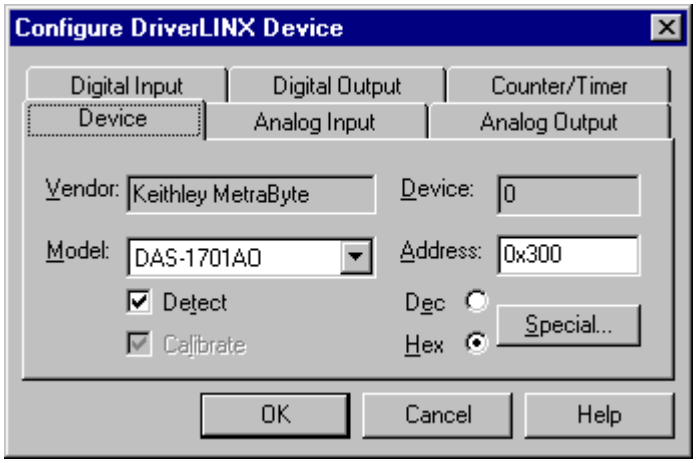
DriverLINX uses a standardized configuration protocol for all data-acquisition hardware. Configuration assigns a port address, interrupt resources and a DriverLINX Logical Device number to a specific DAS-1700 Series board in your computer.

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



When you click the *Configure…* button on the *DriverLINX Configuration Panel*, DriverLINX displays the *Configure DriverLINX Device* dialog. The dialog has a page for each subsystem on a Keithley DAS-1700 Series model. The following sections describe your choices in configuring DriverLINX to work with your board.

Device Subsystem Page



Use the Device subsystem page to tell DriverLINX the model name, address and, optionally, the expansion accessories connected to your DAS-1700 Series board.

Vendor

The *Vendor* property displays "Keithley MetraByte" It is a read-only property.

Device

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

Model

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

Windows NT

Select one of the following models:

DAS-1701ST

DAS-1702ST

DAS-1702HR

DAS-1701AO

DAS-1702AO

DAS-1701ST-DA

DAS-1702ST-DA

DAS-1702HR-DA

Windows 95/98

Under Windows 95/98, DriverLINX displays the model you chose during installation. To install a different model, cancel the configuration and run *Add New Hardware* from the Windows Control Panel.

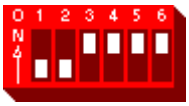
Address

The *Address* property records the I/O port address for the board. The default address used by DriverLINX is 0x300 hexadecimal or 768 decimal.

The DAS-1700 has a 6-bit DIP switch that sets its base address. Each switch corresponds to a binary digit in the address. When a switch is up, or in the ON position, its digit is 0. When it's down, its digit is 1.

The switches select the value of the 1st to the 6th digits. The 7th to the 10th digits of the address are 0. That is, there are four zeros to the right of the bits represented by the switches.

For example, the following DIP switch shows the setting for 11 0000 0000 (0x300 hexadecimal or 768 decimal).



Windows NT

Enter your board's base I/O address. Note: you need a block of sixteen free addresses. AO models use an additional block of ten addresses starting at base + 0x400.

Windows 95/98

Under Windows 95/98, *Add New Hardware* automatically selects an appropriate address. To change the setting, see "Using the Windows 95/98 Device Manager" on page 17.

Detect

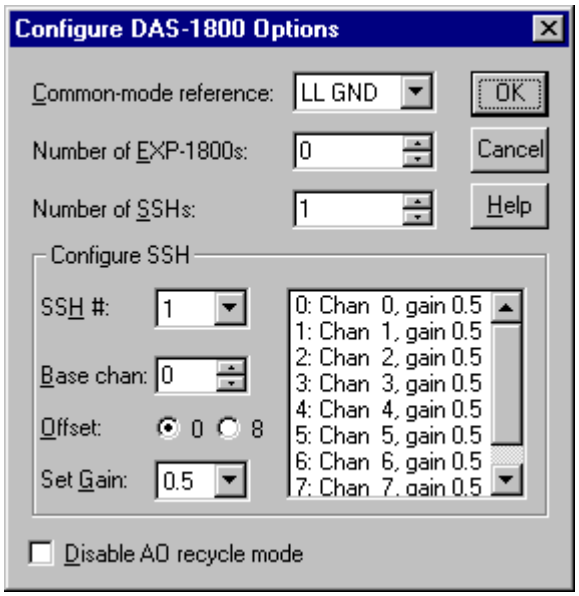
The *Detect* property enables and disables DriverLINX's hardware detection and testing algorithms. For maximum system reliability, always leave this check-box marked.

Calibrate

The *Calibrate* property enables and disables hardware auto-calibration. This option is grayed-out for the DAS-1700 Series because it does not support automatic calibration.

Special…

The *Special…* button displays the following dialog box of DAS-1700 Series-specific configuration options:



Common-mode reference

The DAS-1700 has two grounding options for single-ended analog inputs. They determine the ground reference for the ADC input amplifier.

- LL GND — references the analog ground connection.
- U_CM MD — references the user-common mode ground connection.

Use the U_CM MD setting to eliminate ground loops. See your *DAS-1700 User's Guide* for more details.

Number of EXP-1800s

You can expand the number of single-ended analog input channels connected to your DAS-1700 board by using one to sixteen EXP-1800 expansion boards. Each EXP-1800 is 1-to-16 multiplexer that replaces one onboard channel with sixteen expansion channels. Configure your DriverLINX Logical Device to use the additional channels by entering the number of EXP-1800s here. See "Analog Input Expansion Channels" on page 58 for details on accessing multiplexer channels.

Simultaneous sample and hold configuration

You can provide for simultaneous analog input sampling using external Simultaneous Sample and Hold (SSH-8) units. A DAS-1700 can have up to two SSH-8 units. In the *Configure DAS-1700 Options* dialog box, enter the number of SSH-8s connected.

For each SSH-8 unit, select its number in the *SSH #* list box and enter the following configuration parameters:

- Base channel — Each SSH-8 replaces 8 DAS-1700 channels determined by which accessory and SSH-8 connectors you use and the offset jumper settings on the SSH-8. You can connect two SSH-8 units to one accessory connector by daisy chaining them together using the connectors inside the units. See the *SSH-8 User's Guide* for details.

  Select the base channel corresponding to the connectors you are using for this SSH-8 unit:

Accessory	Base Channel	
Connector	Directly connected SSH-8	Daisy-chain connected SSH-8
STA-1800U		
J3	0	0

- Channel offset — click on an SSH-8 channel in the list and then select the channel offset that corresponds to the position of the Output Channel Jumper on the SSH-8. The list displays the number of the DAS-1700 channel that the SSH-8 channel replaces.

  Usually, you would configure all channels on a directly connected SSH-8 to the lower offset channels (0-7) and all channels on a daisy-chain connected SSH-8 to the higher offset channels (8-15).

- Channel gain — click on an SSH-8 channel in the list and then select its SSH-8 external amplifier gain. SSH-8FG models have a fixed gain of 0.5. SSH-8SG models have switches to set the gain of each channel.

For information on programming a task for simultaneous sampling, see "Rate Generator: Internal Clocking" on page 41 or "Rate Generator: External Clocking" on page 43.

Disable AO recycle mode

For the DAS-1700AO hardware, DriverLINX can automatically promote AO tasks meeting certain criteria to run from the DAC FIFO buffer in recycle mode. If the *Disable AO recycle mode* box is checked, DriverLINX will not use recycle mode. If the box is not checked, DriverLINX will use recycle mode for applicable tasks.

In recycle mode, DriverLINX automatically promotes AO tasks meeting the following criteria as specified in the Service Request to run from the DAC FIFO buffer in re-cycle mode:

Service Request Property	Value
Request mode	Interrupt
Stop event type	Command
Buffer notify flag	False
Buffer samples × Number of buffers	< 2048

The advantages of using the DAC FIFO buffer for free-run, re-cycle mode analog output are:

- The DAS-1700AO's digital trigger operates in retrigger mode, enabling you to synchronize analog output with a recurring external signal.
- Traffic on the memory bus is significantly reduced at high output rates.
- The system response latency will be noticeably shortened.
- The analog output timing will be immune to traffic on the memory and I/O buses.

The disadvantages are:

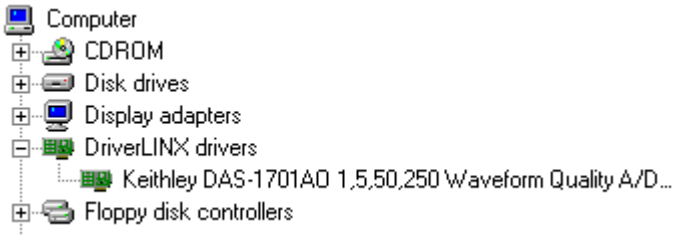
- You can't change the analog output values while the analog output task is running.
- There is a delay while DriverLINX loads the FIFO.
- Buffer-filled notification messages cannot be posted, as the hardware does not permit detection of end-of-buffer conditions.
- The *STATUS* operation cannot provide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

Using the Windows 95/98 Device Manager

Under Windows 95/98, DriverLINX uses the address and interrupt settings maintained by the Windows Device Manager.

To view or change the settings for your board using the Device Manager:

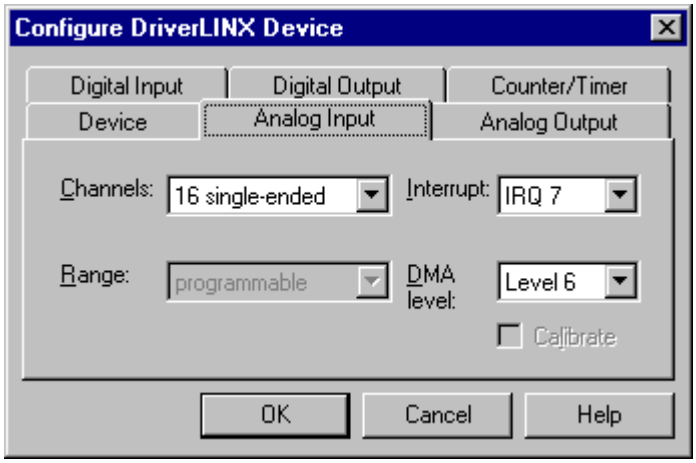
1. Start the Device Manger by right-clicking on My Computer and selecting *Properties* or click here .
2. Click the *Device Manger* tab.
3. Click the next to DriverLINX drivers, if necessary to expand the list.
4. Under *DriverLINX drivers*, select the entry for your board. (It may or may not have next to it.)



5. Click the *Properties* button.
6. On the board's property page, click the *Resources* tab.
7. To change the number and type of resources, select a different Basic Configuration under *Setting based on.*

8. To change a resource setting, select it under *Resource Type* and click the *Change Setting* button. Windows will guide you in selecting an appropriate value.
9. When you are done, click *OK* to close the board's property page.
10. The board's address switches must match the address setting you select. If necessary shut down your computer and reposition them as described in your hardware manual.
11. Restart Windows to load the Logical Device for your board using the new settings.

Analog Input Subsystem Page



Use the Analog Input subsystem page to choose between single-ended or differential analog input connections and to set or view your board's interrupt request level.

Channels

On the DAS-1700 boards, Analog Input channel configuration is software programmable for 16 single-ended or 8 differential analog inputs.

When configuring the Analog Input Subsystem, you choose a default configuration for all channels. Applications can use the default configuration or specify the connection type for each channel it uses. This scheme supports applications that use DAS-1700-specific features as well as those that use only generic features. For programming information, see "Analog Input Channels" on page 53.

Range

The analog input ranges for the DAS-1700 Series are fully software programmable. DriverLINX grays out this property in the configuration dialog.

Interrupt

Configure the board with an interrupt for full support of the board's capabilities.

Windows NT

For Windows NT, select a free interrupt request level to support interrupt or DMA mode transfers. Valid IRQ settings are: 3, 5, 7, 10, 11, 15 and None.

Windows 95/98

Under Windows 95/98, *Add New Hardware* automatically selects an appropriate interrupt. To change the setting, see "Using the Windows 95/98 Device Manager" on page 17.

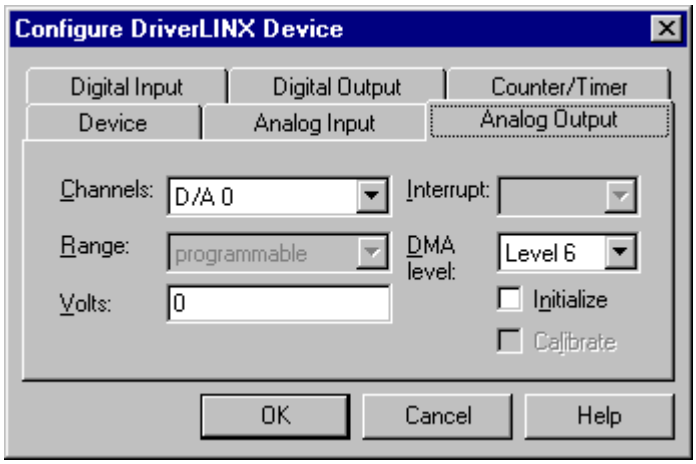
DMA

Configure the board with a DMA channel for full support of the board's capabilities.

For Windows NT, select a free DMA channel to support DMA mode transfers. Valid DMA settings are: 5, 6, 7 and None.

Under Windows 95/98, *Add New Hardware* automatically selects an appropriate DMA channel. To change the setting, see "Using the Windows 95/98 Device Manager" on page 17.

Analog Output Subsystem Page



Note: The *Configure DriverLINX Device* dialog shows the Analog Output Subsystem Page only for applicable models.

Use the Analog Output subsystem page to set or view the subsystem's DMA channel and initial output voltages.

Channels

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

Range

The analog output ranges for the DAS-1700 Series are fully software programmable. DriverLINX grays out this property in the configuration dialog.

Volts

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

Interrupt

The DAS-1700 Series uses the same interrupt for analog output as for analog input. Go to the Analog Input subsystem page to view or set it.

DMA

DMA applies only to the DAS-1701AO and DAS-1702AO models. Configure the board with a DMA channel for full support of the board's capabilities.

Windows NT

For Windows NT, select a free DMA channel to support DMA mode transfers. Valid DMA settings are: 5, 6, 7 and None.

Windows 95/98

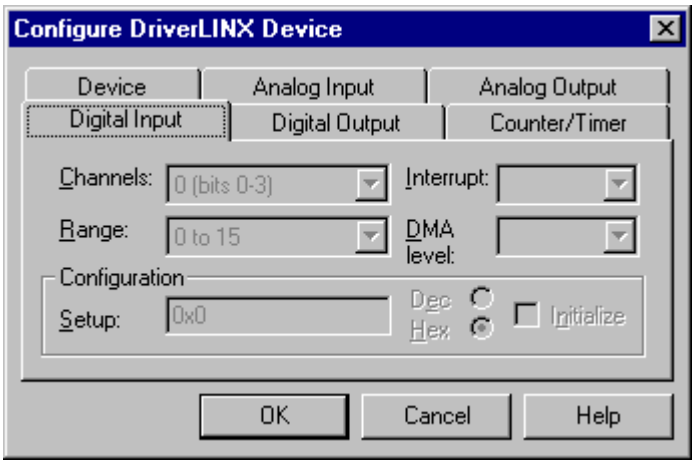
Under Windows 95/98, *Add New Hardware* automatically selects an appropriate DMA channel. To change the setting, see "Using the Windows 95/98 Device Manager" on page 17.

Initialize

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

Digital Input Subsystem Page

For the DAS-1700 Series, there are no configurable options on the Digital Input subsystem page.



Channels

The *Channels* property allows you to select a Logical Channel for configuration or viewing the channel's range. The DAS-1700 Series digital channels have fixed configurations.

DriverLINX defines the following Logical Channels for the DAS-1700 Series digital inputs:

Logical Channel	DriverLINX Function	DAS-1700 Series External Connector
0	Standard Digital Input	DI 0 … DI 3
1	External Clock	XPCLK
2	External Trigger	TGIN

Range

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

Interrupt

The DAS-1700 Series does not use interrupts for the digital input subsystem. DriverLINX disables this property and displays it as blank.

DMA

The DAS-1700 Series does not use system DMA channels. DriverLINX disables this property and displays it as blank.

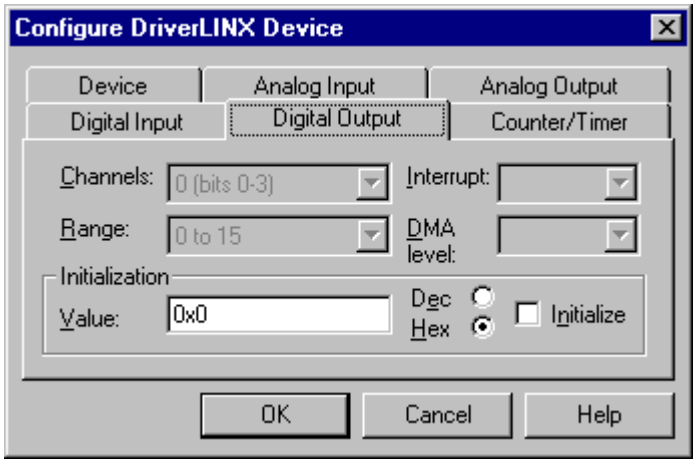
Configuration Setup

The *Configuration Setup* property specifies the hardware configuration of the digital I/O ports. The DAS-1700 Series has a fixed digital I/O configuration. Therefore, DriverLINX disables this field.

Initialize

Checking the *Initialize* check box instructs DriverLINX to use the *Configuration Setup* property to configure the digital I/O ports. The DAS-1700 Series has a fixed digital I/O configuration. Therefore, DriverLINX disables this field.

Digital Output Subsystem Page



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

Channels

The *Channels* property allows you to select a Logical Channel for initialization or viewing the channel's range. DriverLINX defines the following Logical Channels for the DAS-1700 Series digital outputs:

Logical Channel	DriverLINX Function	DAS-1700 Series External Connector
0	Standard Digital Output	DO 0 … DO 3
1	Standard Digital Output	MUX4 … MUX7, GEXT

The Analog Input subsystem uses the MUX outputs to control an EXP-1800 expansion accessory. If you are not using an EXP-1800, you can use the MUX outputs as another Digital Output channel. However, you can change its value only while the Analog Input subsystem is inactive.

Range

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

Interrupt

The DAS-1700 Series does not use interrupts for the digital output subsystem. DriverLINX disables this property and displays it as blank.

DMA

The DAS-1700 Series does not use system DMA channels for Digital Output. 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 upon 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 DAS-1700 Series, the default digital output value is zero.

Initialize

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

Dec

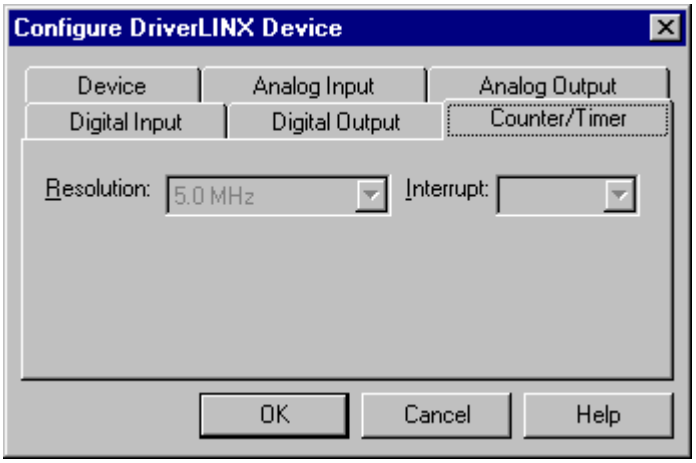
This check box converts the *Initialization Value* property to decimal.

Hex

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

Counter/Timer Subsystem Page

For the DAS-1700 Series, there are no configurable options on the Counter/Timer subsystem page.



Resolution

The *Resolution* property specifies the clock frequency of the master oscillator. All models have a 5.0 MHz clock source for pacing analog input. AO models also use the 5.0 MHz clock source for pacing analog output.

Interrupt

The DAS-1700 Series does not support interrupts from counter/timers. DriverLINX disables this property and displays it as blank.

Using the DAS-1700 Series with DriverLINX

Introduction

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

This chapter shows you how to set up and use DAS-1700 Series hardware features with DriverLINX.

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

DriverLINX Hardware Model for DAS-1700 Series

DriverLINX provides a portable, hardware-independent API for data-acquisition boards while still allowing applications to access unique or proprietary hardware features of specific products. To achieve this goal, DriverLINX maps a hardware-independent, or abstract, data-acquisition model onto DAS-1700 Series hardware capabilities.

The following sections describe how DriverLINX implements DAS-1700 Series hardware features as Subsystems, Modes, Operations, Events, Logical Channels, Buffers, and Messages.

DriverLINX Subsystems

The DAS-1700 Series supports the following DriverLINX subsystems:

1. **Device**—refers to a DAS-1700 board as a whole.
2. **Analog Input**—refers to the analog input channels, clocks, and control signals.
3. **Analog Output**—refers to the analog output channels, clocks, and control signals. *DA and AO models only.*

4. **Digital Input**—refers to the digital input port as well as 1-bit digital input (TTL) control signals, such as TGIN.
5. **Digital Output**—refers to the digital output port and control signals.
6. **Counter/Timer**—refers to the input/output subsystem-specific internal clock channels for pacing analog input/output.

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 four modes with the DAS-1700 Series for its commands (Service Requests).

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

The following table summarizes the data acquisition modes that DriverLINX supports for each subsystem with the Keithley DAS-1700 Series.

Subsystem	Polled	Interrupt	DMA	Other
Analog Input	√	√	√	√
Analog Output (DA models)	√	√		√
Analog Output (AO models)	√	√	√	√
Digital Input	√	√		√
Digital Output	√	√		√
Counter/Timer				√
Device				√

DAS-1700 Series 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 DAS-1700 Series. Later sections for each DriverLINX subsystem will describe the operations and events in more detail.

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

Subsystem	Operation	Events		
Mode		Timing	Start	Stop
Analog Input				
Polled	Start, Convert	null	null, cmd	null, TC
Interrupt	Start, Stop, Status, Convert	dig, rate	cmd, dig	cmd, TC, dig
DMA	Start, Stop, Status, Convert	dig, rate	cmd, dig	cmd, TC, dig
Other	Initialize			
Analog Output (DA and AO models only)				
Polled	Start, Convert	null	null, cmd	null, TC
Interrupt (AO models only)	Start, Stop, Status, Convert	dig, rate	cmd, dig	cmd, TC
DMA (AO models only)	Start, Stop, Status, Convert	dig, rate	cmd, dig	cmd, TC
Other	Initialize			
Digital Input				
Polled	Start	null	null, cmd	null, TC
Interrupt	Start, Stop, Status	rate	cmd	cmd, TC
Other	Initialize			
Digital Output				
Polled	Start	null	null, cmd	null, TC
Interrupt	Start, Stop, Interrupt	rate	cmd	cmd, TC
Other	Initialize			
Counter/Timer				
Other	Initialize			
Device				
Other	Initialize, Capabilities			

Allowed Operations and Events for DAS-1700 Series Subsystems and Modes.

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

- **null**—Null or None Event when a Service Request doesn't require an event
- **cmd**—Command Event when DriverLINX starts or stops a task on software command
- **TC**—Terminal Count Event when DriverLINX processes all data buffers once
- **rate**—Rate Event specifies how DriverLINX paces or clocks data transfer
- **dig**—Digital Event specifies a trigger, clock, or other control signal to pace, start, or stop a task

Logical Channels

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

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

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

Buffers

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

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

Connecting Signals to the DAS-1700 Series

The Keithley hardware manual describes the data and control signals for the DAS-1700 Series and the connector pinouts for these signals. This section summarizes how DriverLINX numbers the I/O data signals and how DriverLINX uses the control connections for external clock, trigger, and gating inputs.

Analog Input Subsystem Signals

The Analog Input subsystem has 8 differential, 32 differential, 16 single-ended or 64 single-ended analog input connections, depending on the model and configuration of your DAS-1700 board.

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

Physical Channels	Connector Name	Logical Channels
0 – 7 Differential	CH00LO, HI – CH07LO, HI	0 – 7
0 – 31 Differential	CH00LO, HI – CH31LO, HI	0 – 31
0 – 15 Single-ended	CH00LO – CH15LO, LLGND	0 – 15
0 – 15 Single-ended, User-common mode	CH00LO – CH15LO, U_CM_MD	0 – 15
0 – 63 Single-ended	CH00LO – CH63LO, AGND	0 – 63

How DriverLINX maps analog input hardware channels to Logical Channels.

Analog Input Pacing, Triggering and Gating Signals

Analog input tasks can use the input-pacing counter, which DriverLINX designates as Counter/Timer Logical Channel 0. For this counter the clock source can be internal or external.

The Analog Input subsystem uses two control signals that DriverLINX defines as an external clock and a trigger/gate as shown in the following table:

Connector Name	DriverLINX Usage
TGIN	External trigger: • Digital Start Event (Post-Triggering) • Digital Stop Event (Pre- and About-Triggering) External gate: • Rate Timing Event • Digital Start Event (Gating)

Connector Name	DriverLINX Usage
XPCLK	External pacer clock: • Rate Timing Event • Digital Timing Event

How DriverLINX uses analog input control signals.

Analog Output Subsystem Signals

The Analog Output subsystem has 2 or 4 analog differential output connections, depending on the model of your DAS-1700 board.

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

Physical Channels	Connector Name	Logical Channels
0 – 1 (HR-DA and AO models)	ODAC 0 – ODAC 1	0 – 1
0 – 3 (ST-DA models only)	ODAC 0 – ODAC 3	0 – 3

How DriverLINX maps analog output hardware channels to Logical Channels.

Analog Output Pacing, Triggering and Gating Signals

Analog output tasks can use the output-pacing counter, which DriverLINX designates as Counter/Timer Logical Channel 1. For this counter the clock source can be internal or external.

The Analog Output subsystem uses two control signals that DriverLINX defines as an external clock and a trigger as shown in the following table:

Connector Name	DriverLINX Usage
TGIN	External trigger: • Digital Start Event (Post-Triggering) External gate: • Rate Timing Event • Digital Start Event (Gating)
XPCLK	External pacer clock: • Rate Timing Event • Digital Timing Event

How DriverLINX uses analog output control signals.

Digital Input Subsystem Signals

The Digital Input subsystem has one 4 or 8-bit digital input port and two control inputs which DriverLINX models as 1-bit logical digital input ports. DriverLINX maps these signals to Logical Channels as shown in the following table:

Port	Connector Name	Logical Channel
4-bit digital input	DI 0 … DI 3	0
External clock alias	XPCLK	1
External trigger alias	TGIN	2

How DriverLINX maps digital input hardware channels to Logical Channels.

Digital Output Subsystem Signals

The Digital Output subsystem has a 4 or 8-bit digital output port and a group of control signals that you can use as an output port. DriverLINX maps these signals to Logical Channels as shown in the following tables:

Other Models

Port	Connector Name	Logical Channel
4-bit digital output	DO 0 … DO 3	0
5-bit digital output	MUX4 – MUX7, GEXT	1

How DriverLINX maps digital output hardware channels to Logical Channels of other models.

Counter/Timer Subsystem Signals

The Counter/Timer subsystem has one or two counter/timer channels for analog input/output pacing. It also has an external clock source and a gate input that the board can use with any Logical Channel. DriverLINX maps these signals as shown in the following table:

Timer	Connector Name	Logical Channel
Input Pacing Counter	XPCLK	0
Output Pacing Counter (AO models only)	XPCLK	1

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

Device Subsystem

The following sections describe how DriverLINX implements Device Subsystem features for the DAS-1700 Series.

Device Modes

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

Device Operations

The DAS-1700 Series Device Subsystem supports the following DriverLINX operations:

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

- **Initialize**—DriverLINX aborts all data-acquisition tasks for every subsystem controlled by the current application. DriverLINX then initializes each subsystem.
- **Capabilities**—DriverLINX provides hardware-specific and configuration information in the form of a Logical Device Descriptor database.

Analog Input Subsystem

The following sections describe how DriverLINX implements Analog Input Subsystem features for the DAS-1700 Series.

Analog Input Modes

The Analog Input Subsystem supports the following modes:

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

Analog Input Operations

The DAS-1700 Series Analog Input Subsystem supports the following DriverLINX operations:

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

Analog Input Pacing, Triggering and Gating Options

The DAS-1700 Series *User's Guides* describe several pacing, triggering and gating options available on DAS-1700 models. The following table summarizes these options and identifies which Service Request properties use them. Except as indicated all tasks must use Interrupt or DMA mode.

Parameter	Option	Service Request Properties
Pacing Mode		
	Periodic (paced)	Rate generator timing event
	Burst	Burst generator timing event
Clock Source		
	Software	Single-value or single-scan (Polled mode)
	Internal	Rate timing event with an internal clock source
	External +/-	Rate timing event with an external clock source Digital timing event
Trigger		
	Internal (software)	Command start event Command stop event Terminal count stop event
	Digital +/-	Digital start event Digital stop event
Trigger Mode		
	Pre-trigger	Digital stop event with 0-delay
	About-trigger	Digital stop event with positive delay
	Post-trigger	Digital start event
	Trigger-to-trigger	Digital start event Digital stop event
	Trigger-to-about-trigger	Digital start event Digital stop event with positive delay
Gate		
	Level +/-	Rate timing event Digital start event (for use with a digital timing event)

Analog Input Timing Events

Timing Events specify how the hardware paces or clocks the acquisition of analog input samples. DriverLINX uses the Timing Event to program when the DAS-1700 Series acquires the next analog input sample.

The DAS-1700 Series supports the following Timing Events:

- **None**—Sampling requires no pacing as DriverLINX is acquiring only a single value or scan.
- **Rate**—The DAS-1700 Series supports fixed rate and burst mode sampling using internal and external clocks.
- **Digital**—DriverLINX uses an external digital input signal to pace the acquisition of the next sample.

None or Null Timing Event

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

Rate Timing Event

The DAS-1700 Series supports two types of Rate Events for analog input:

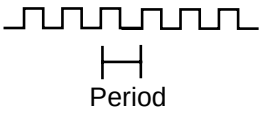
- **Rate Generator**—Generates a fixed rate clock with equal time intervals between tics.

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

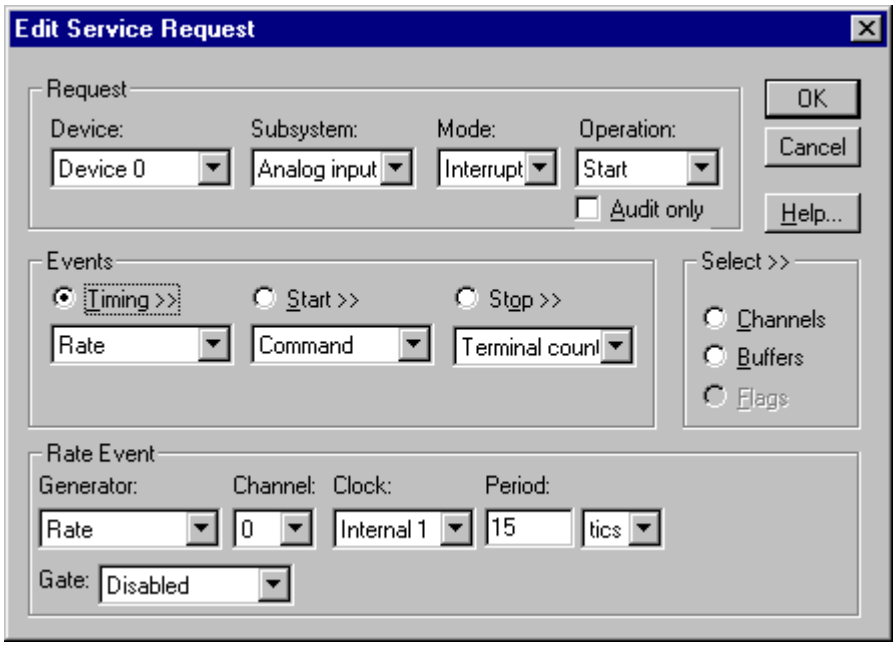
DAS-1700 boards have a 5 MHz master clock frequency, or 0.2 µs tic period. The sample period can range from 30 tics (6µs) to $2^{32}-1$ tics (833 s). This means the sample rate can range from 0.00116 Hz to 162.5kHz. However, using multiple channels or non-unity gains may reduce the maximum sample rate that the hardware can accurately acquire. Also, HR models are limited to 50 kHz. Consult your hardware manual for details.

Rate Generator: Internal Clocking

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



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



How to set up the DAS-1700 Series for fixed rate sampling using an internal clock.

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

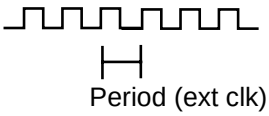
- Specify internal clocking using a **Rate** *Generator* on *Channel* **0** with the **Internal 1** *Clock* source. See "Counter/Timer Subsystem" on page 101 for a description of clock sources.
- The *Period* property specifies the time interval between samples in tics, where an **Internal 1** tic is 0.2 µs, or 5 MHz. The minimum period is 30 tics, or 162.5 kHz (100 tics or 50 kHz for HR models). The maximum period is 4294967295 tics ($2^{32} - 1$), or 0.00116 Hz.
- The *Gate* property specifies how the TGIN signal affects the operation of the internal clock. Valid settings are Enabled, Disabled, and High Level and Low Level. See "Counter/Timer Subsystem" on page 101 for a description of each *Gate* setting.

  Note: You cannot use a gated clock with a digital start or stop trigger.
- For simultaneous sampling using an SSH-8 accessory, check the *Simultaneous* box in the *Channels* section. Simultaneous means that all selected channels are sampled in each sample period.

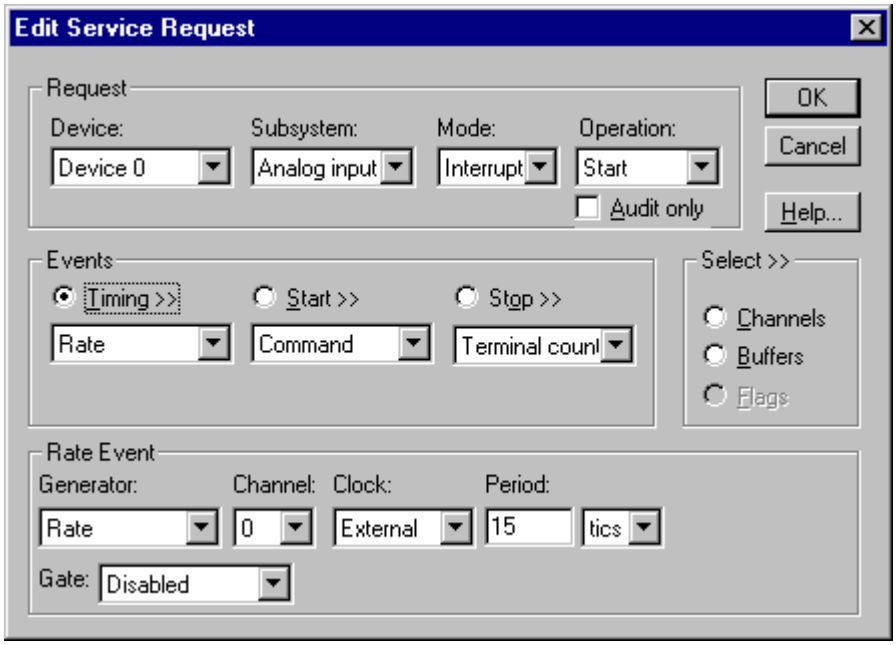
Also see: "Repeat Mode Sampling on the DAS-1702HR-DA" on page 57.

Rate Generator: External Clocking

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



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



How to set up the DAS-1700 Series for fixed rate sampling using an external clock.

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

- Specify external clocking using a **Rate** *Generator* on *Channel* **0** with an **External, External+**, or **External-** *Clock* source. See "Counter/Timer Subsystem" on page 101 for a description of clock sources. For hardware-independence, you can specify the hardware external clock channel by the symbolic constant, *DI_EXTCLK*.
- Users should connect the external clock signal to the XPCLK line.
- Specify a *Period* between the minimum and maximum external clocking period. The value doesn't affect the external clock frequency, but DriverLINX requires a valid hardware value in case the application requests a timebase operation and to optimize data transfer between the driver and the application.
- The frequency of the external clock must not exceed 162.5 kHz (50 kHz for HR models).

- The *Gate* property specifies how the TGIN signal affects the operation of the internal clock. Valid settings are Enabled, Disabled, High Level and Low Level. See “Counter/Timer Subsystem” on page 101 for a description of each *Gate* setting.

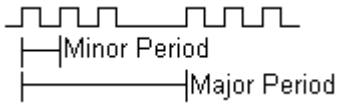
  Note: You cannot use a gated clock with a digital start or stop trigger.

- For simultaneous sampling using an SSH-8 accessory, check the *Simultaneous* box in the *Channels* section.

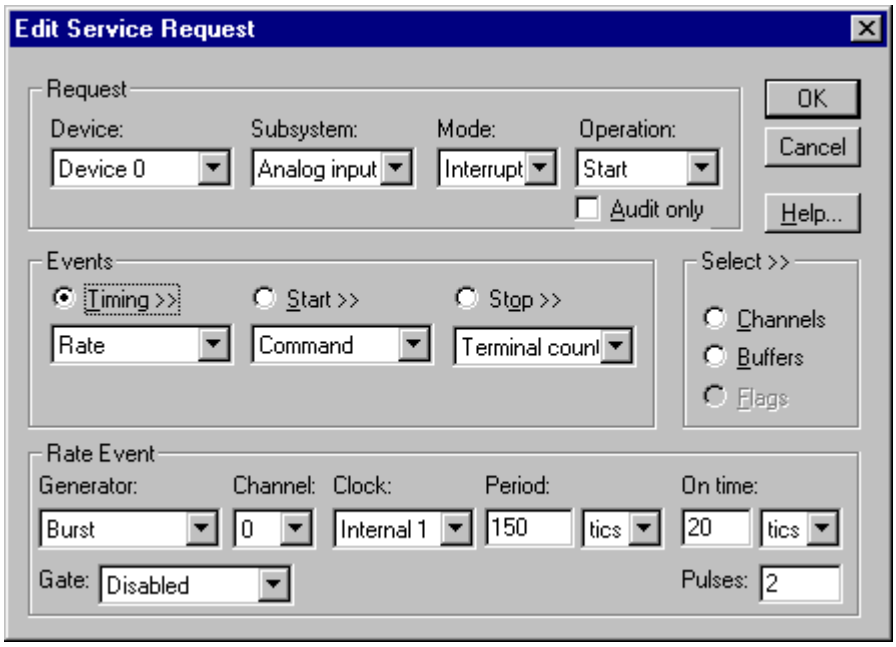
Also see: “Repeat Mode Sampling on the DAS-1702HR-DA” on page 57.

Burst Generator: Internal Clocking

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



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



How to set up the DAS-1700 Series for burst mode sampling using an internal clock.

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

- Specify internal clocking using a **Burst** *Generator* on *Channel* **0** with the **Internal 1** *Clock* source. See “Counter/Timer Subsystem” on page 101 for a description of clock sources.
- The *Period* property specifies the time interval between bursts in tics, where an **Internal 1** tic is 0.2 µs, or 5 MHz. The minimum period is 30 tics, or 162.5 kHz (100 tics or 50 kHz for HR models). The maximum period is 4294967295 tics ($2^{32}-1$), or 0.000116 Hz.
- The *On time* property specifies the time interval between samples. It must be within the range of 6µs (30 tics) minimum (20µs for DAS-1700HR) to 64µs (320 tics) maximum. Also *On time* × *Pulses* must be less than or equal to *Period*.
- The *Gate* property specifies how the TGIN signal affects the operation of the internal clock. Valid settings are Enabled, Disabled, High Level and Low Level. See “Counter/Timer Subsystem” on page 101 for a description of each *Gate* setting.

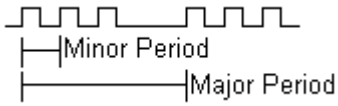
Note: You cannot use a gated clock with a digital start or stop trigger.

- The *Pulses* property specifies how many channels the board samples in each *Period*. *Pulses* must equal the number of channels in the channel list.

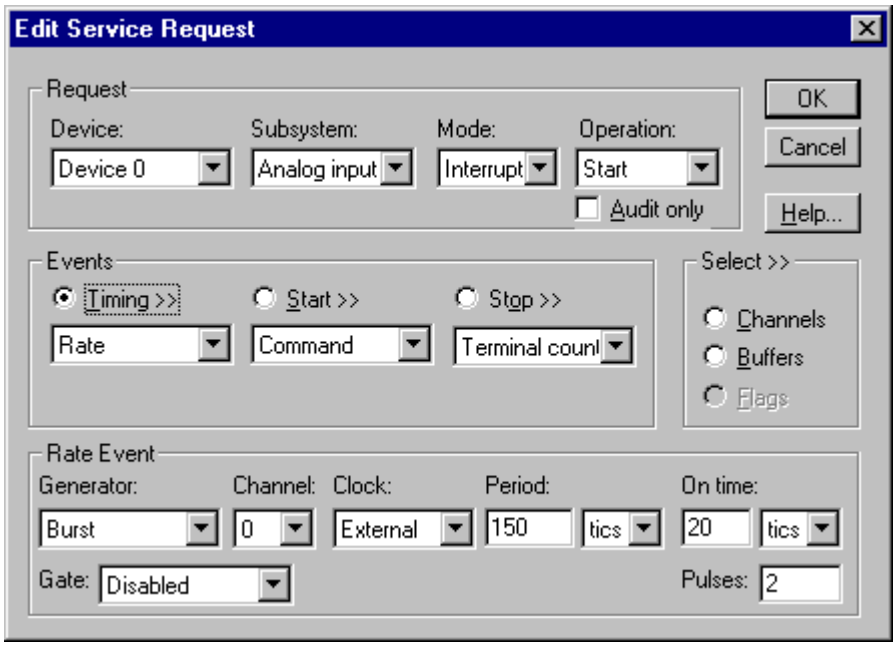
Also see: “Repeat Mode Sampling on the DAS-1702HR-DA” on page 57.

Burst Generator: External Clocking

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



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



How to set up the DAS-1700 Series for burst mode sampling using an external clock.

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

- Specify external clocking using a **Burst** *Generator* on *Channel* **0** with an **External, External+**, or **External-** *Clock* source. See “Counter/Timer Subsystem” on page 101 for a description of clock sources. For hardware-independence, you can specify the hardware external clock channel by the symbolic constant, *DI_EXTCLK*.
- Users should connect the external clock signal to the XPCLK line.
- Specify a *Period* between the minimum and maximum external clocking period. The value doesn’t affect the external clock frequency,

but DriverLINX requires a valid hardware value in case the application requests a timebase operation and to optimize data transfer between the driver and the application.

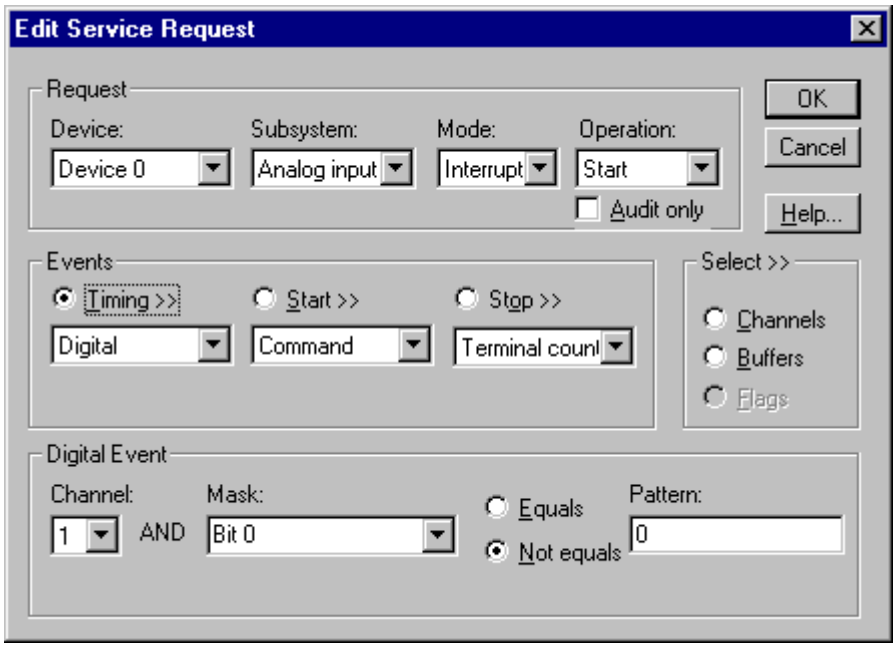
- The *On time* property specifies the time interval between samples. It must be within the range of 6µs (30 tics) minimum (20µs for DAS-1700HR) to 64µs (320 tics) maximum. Also *On time* × *Pulses* must be less than or equal to *Period*.
- The frequency of the external clock must not exceed 162.5 kHz (50 kHz for HR models).
- The *Gate* property specifies how the TGIN signal affects the operation of the internal clock. Valid settings are Enabled, Disabled, High Level and Low Level. See "Counter/Timer Subsystem" on page 101 for a description of each *Gate* setting.

  Note: You cannot use a gated clock with a digital start or stop trigger.
- The *Pulses* property specifies how many channels the board samples in each *Period*. *Pulses* must equal the number of channels in the channel list.

Also see: "Repeat Mode Sampling on the DAS-1702HR-DA" on page 57.

Digital Timing Event

DriverLINX supports Digital Events as aliases for externally clocked Rate Generators. Use this technique for compatibility with data-acquisition products that only support external clock sources, such as some digital I/O products.



How to set up the DAS-1700 Series for external rate sampling using a digital event.

- Specify external clocking using *Channel* **1.** 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 the XPCLK line.
- Specify the *Mask* property as **1,** or **Bit 0**, to indicate that DriverLINX should only compare a 1-bit digital input value against the *Pattern* property.
- Specify the *Match* property as **Not equals**.
- Specify the *Pattern* property as **0** for a rising, or positive, edge clock (≠0) or **1** for a falling, or negative, edge clock (≠1).

Analog Input Start Events

Start Events specify when the DAS-1700 hardware starts acquiring analog input data.

The DAS-1700 Series supports the following Start Events:

Note: The DAS-1700 Series does not support analog triggering.

- **None**—Use this event when the DriverLINX operation does not require a Start Event.
- **Command**—DriverLINX starts the task on software command, i.e., as soon as DriverLINX finishes programming the DAS-1700 hardware for the task.
- **Digital**—The DAS-1700 starts acquiring analog input samples when the hardware detects the digital Logical Channel input satisfies the condition specified in the Start Event.

None or Null Start Event

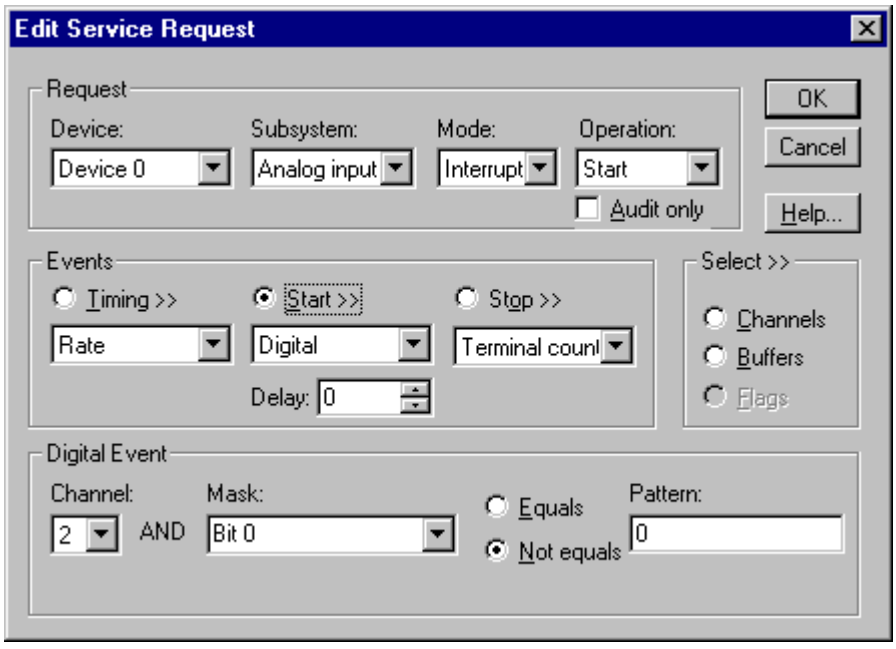
The Null Event specifies that the task does not need a Start Event to begin the task.

Command Start Event

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

Digital Start Event (Post-Triggering)

The DAS-1700 can acquire analog input samples ***after*** the hardware detects a digital trigger condition. Use post-triggering when you want to synchronize the start of data acquisition with an external signal.



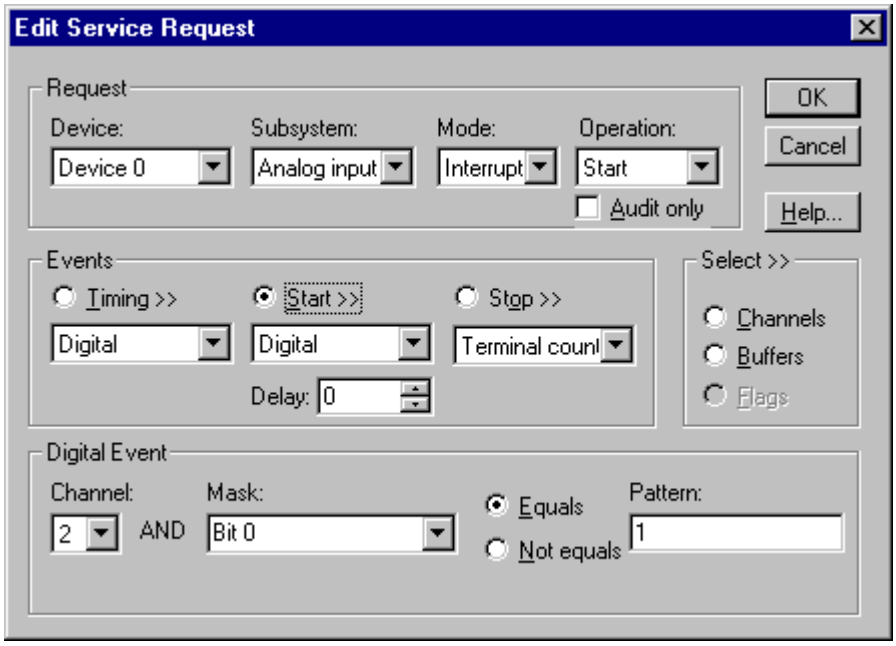
How to set up the DAS-1700 Series for post-triggered analog input.

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

- Specify the *Channel* as **2**. For hardware-independence, you can specify the hardware external trigger channel by the symbolic constant, *DI_EXTTRG*.
- Users should connect the external trigger signal to the TGIN line.
- Specify the *Mask* property as **1**, or **Bit 0**, to indicate that DriverLINX should only compare a 1-bit digital input value against the *Pattern* property.
- Specify the *Match* property as **Not equals**.
- Specify the *Pattern* property as **0** for a rising, or positive, edge trigger (≠0) or **1** for a falling, or negative, edge trigger (≠1).
- Specify the *Delay* property as 0. The DAS-1700 does not support a delay in sampling after the start trigger.
- You cannot use a digital start trigger with a gated clock.
- If both the start trigger and stop trigger are digital events, they must have identical *Pattern* settings.

Digital Start Event (Gating)

The DAS-1700 can pause and resume acquiring analog input samples based on the level of a digital input condition. Use a digital start event to set up gating when using a digital timing event. To set up gating for a rate timing event, see “Rate Timing Event” on page 40.



How to set up the DAS-1700 Series for gated analog input.

Digital Start Events contain *mask*, *pattern*, and *match* fields. The mask selects which bit in the Logical Channel to use as a gate input. The *pattern* selects whether the gate is active, or enables data-acquisition, when the gate signal is high or low.

- Specify the *Channel* as **2**. For hardware-independence, you can specify the hardware external trigger channel by the symbolic constant, *DI_EXTTRG*.
- Users should connect the external trigger signal to the TGIN line.
- Specify the *Mask* property as **1**, or **Bit 0**, to identify the gate input bit of the Logical Channel.
- Specify the *Match* property as **Equals**.
- Specify the *Pattern* property as **0** for a low-level active gate or **1** for a high-level active gate.
- Specify the *Delay* property as 0. DriverLINX does not use a delay with gating.

Analog Input Stop Events

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

The DAS-1700 Series supports the following Stop Events:

- **None**—Use this event when the DriverLINX operation doesn't require a Stop Event.
- **Command**—DriverLINX stops the task on software command, i.e., when the application issues a Service Request with a *Stop* operation.
- **Terminal count**—DriverLINX stops the task after the data-acquisition hardware has filled all the data buffers once.
- **Digital**—The DAS-1700 stops acquiring analog input samples when the hardware detects the digital Logical Channel input satisfies the condition specified in the Stop Event.

None or Null Stop Event

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

Command Stop Event

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

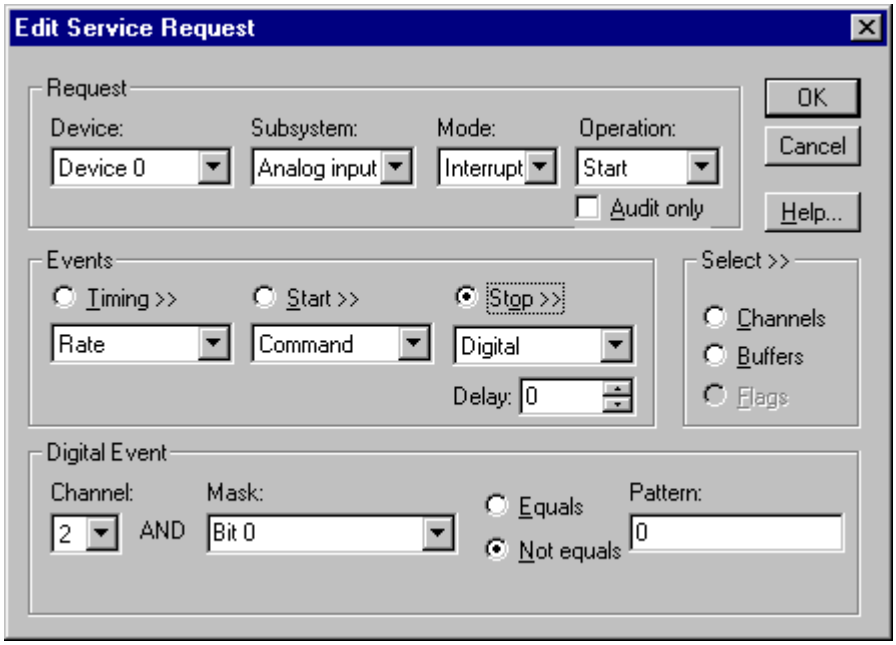
In Stop-on-Command mode, DriverLINX continuously cycles through all the data buffers filling them with analog input data from the data-acquisition hardware.

Terminal Count Stop Event

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

Digital Stop Event (Pre- and About-Triggering)

The DAS-1700 can acquire analog input samples ***until*** the hardware detects a digital trigger condition. Use pre-triggering when you want to synchronize the end of data acquisition with an external signal.



How to set up the DAS-1700 Series for pre-triggered analog input.

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

- Specify the *Channel* as **2**. For hardware-independence, you can specify the hardware external trigger channel by the symbolic constant, *DI_EXTTRG*.
- Users should connect the external trigger signal to the TGIN line.
- Specify the *Mask* property as **1**, or **Bit 0**, to indicate that DriverLINX should only compare a 1-bit digital input value against the *Pattern* property.
- Specify the *Match* property as **Not equals**.
- Specify the *Pattern* property as **0** for a rising, or positive, edge trigger (≠0) or **1** for a falling, or negative, edge trigger (≠1).
- Specify the *Delay* property as an integer from 0 to $2^{16} - 2$. The DAS-1700 continues sampling until it obtains this number of samples after the trigger. The number must be a multiple of the number of channels in the channel list.
- You cannot use a digital stop trigger with a gated clock.
- If both the start trigger and stop trigger are digital events, they must have identical *Pattern* settings.

When the hardware detects the trigger, DriverLINX sends a StopEvent message (or event) that identifies sample following the trigger. For more information, see “Analog Input Messages” on page 63.

Analog Input Channels

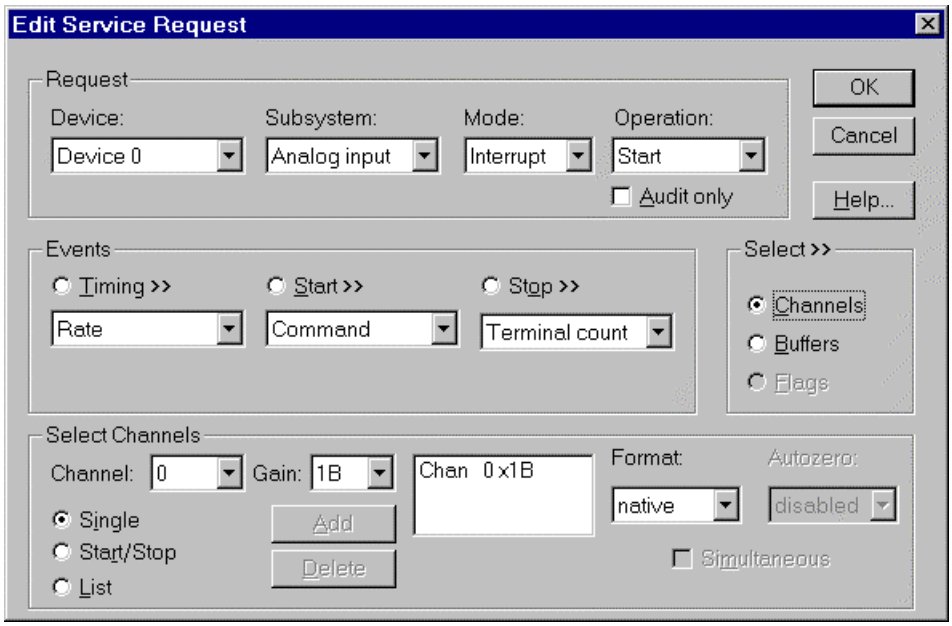
The KPCI-1800 Series models support a variety of channel gains with differential or single-ended connections. The Logical Device configuration sets the default connection type for analog input channels. An application can request a particular connection type for each channel it uses. The channel gains are also application selectable.

The DAS-1700 Series allows applications to specify the analog channels using three techniques:

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

Single Channel Analog Input

In single channel mode, the DAS-1700 Series acquires all data from one channel at the specified gain.



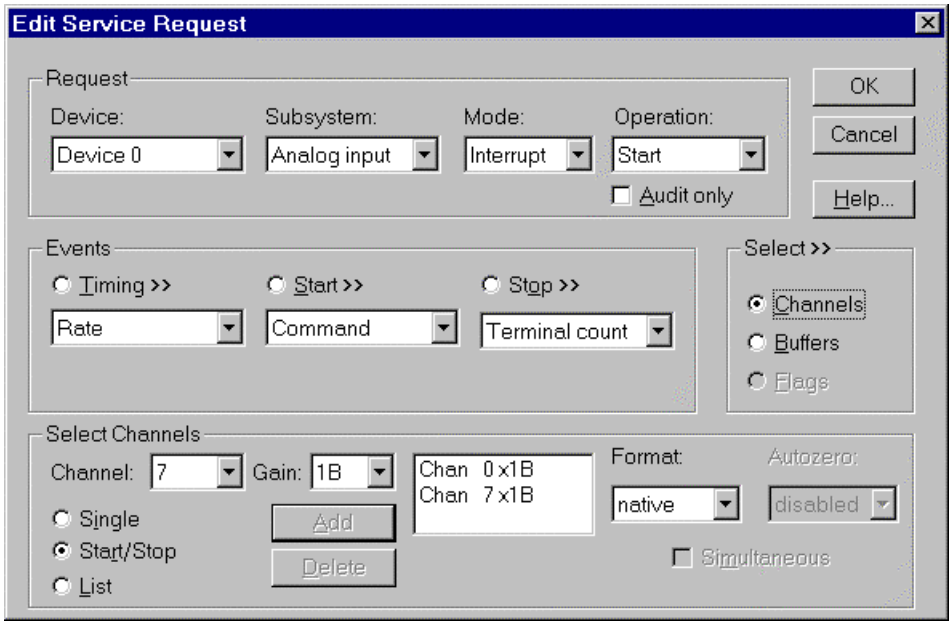
How to set up the DAS-1700 Series for sampling on a single channel.

Multi-channel Analog Input Range

In multi-channel range mode, the DAS-1700 Series acquires data from a consecutive range of analog channels.

- The Start Channel’s gain only applies to the first channel.

- DriverLINX uses the Stop Channel's gain for all the other analog channels in the range.
- The gains may vary but they must all be either unipolar or bipolar.
- If the Start Channel is greater than the Stop Channel, the channel sequence is [Start Channel,… Last Channel, 0,… Stop Channel], where Last Channel is the highest numbered channel for the DAS-1700 model the application is using.
- For simultaneous sampling using an SSH-8 see "Rate Generator: Internal Clocking" on page 41 or "Rate Generator: External Clocking" on page 43.

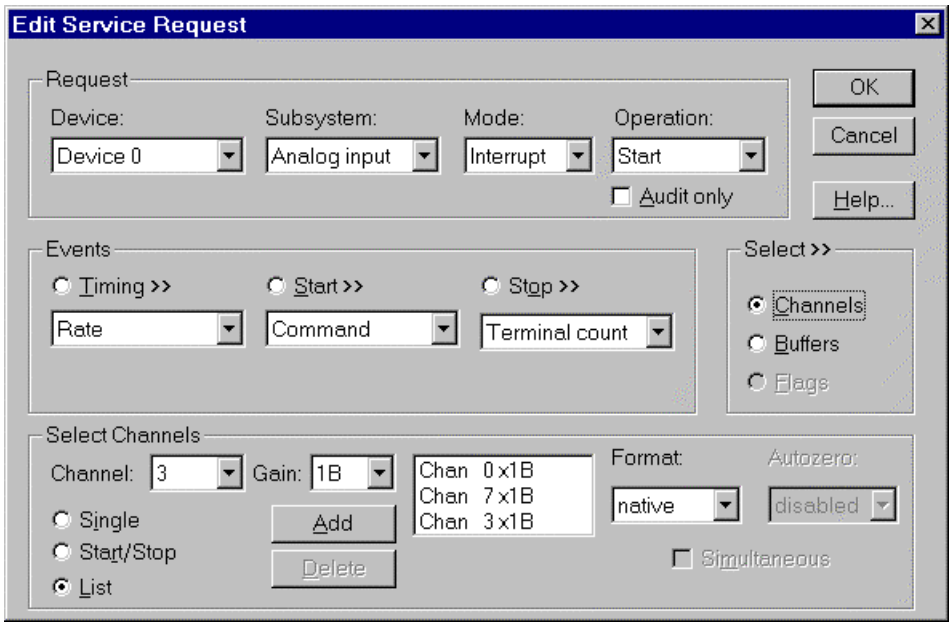


How to set up the DAS-1700 Series for sampling on a consecutive range of channels.

Multi-channel Analog Input List

In multi-channel list mode, the DAS-1700 Series acquires data from an arbitrary list of analog channels.

- The channel-gain list may contain up to 256 channels in any order. The list may repeat the same channel with the same or different gains.
- The gains may vary but they must all be either unipolar or bipolar.
- For simultaneous sampling using an SSH-8 see "Rate Generator: Internal Clocking" on page 41 or "Rate Generator: External Clocking" on page 43.



How to set up the DAS-1700 Series to sample on an arbitrary list of channels.

Analog Input Channel Gains

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

You should be aware that using multiple channels or non-unity gains reduces the maximum sample rate. Consult your hardware manual for details.

DAS-1701 Models

Gain	Range	Gain Code
-1	±5 V	0
-5	±1 V	1
-50	±100 mV	2
-250	±20 mV	3
1	0 … 5 V	4
5	0 … 1 V	5
50	0 … 100 mV	6
250	0 … 20 mV	7

Gains, Ranges, and Hardware Gain Codes for DAS-1701 Models.

DAS-1702 Models

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

Gain Multipliers, Ranges, and Gain Codes for DAS-1702 Models.

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

Expansion Channel Gains

The available gains for an expansion channel are the products of the expansion board's programmable (1 and 50) and the DAS board's programmable gains.

For example, the gains available for a channel on an EXP-1800 attached to a DAS-1702, are: -1, -2, -4, -8, 1, 2, 4, 8, -50, -100, -200, -400, 50, 100, 200, 400.

See "Special…" on page 15 for information on configuring expansion accessories or "Analog Input Expansion Channels" on page 58 for information on selecting expansion channels.

SSH-8 Channel Gains

The available gains for a channel on an SSH-8 unit are the products of the channel's gain on the SSH-8 and the DAS board's programmable gains.

See "Special…" on page 15 for information on configuring an SSH-8 unit.

Analog Input Channel Connection Types

On the DAS-1700 Series, each Analog Input channel can use single-ended or differential connections. When configuring the Analog Input Subsystem, you choose a default configuration for all channels. Applications can use the default configuration or specify the connection type for each channel it uses. This scheme supports applications that use DAS-1700-specific features as well as those that use only generic features.

Each channel in a channel list has a gain code property. To specify a connection type for a channel, an application includes a connection-type flag in its gain code. The following table shows the flag value for each connection type:

Connection Type	Flag Value
Default Configuration	CHAN_SEDIFF_DEFAULT = 0
Single-Ended	CHAN_SEDIFF_SE = 2×2^{13}
Differential	CHAN_SEDIFF_DIFF = 3×2^{13}

Note: The user chooses the default configuration on the Analog Input page of the DriverLINX Configuration Panel. See "Configure DriverLINX Device Dialog" on page 11.

For example, an application that requires or knows a channel's connection type obtains the gain code for a single-ended channel with a bipolar gain of 5, with:

```
Gain2Code (-5) + CHAN_SEDIFF_SE

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

An application that does not require or know a channel's connection type obtains the gain code for a channel with a bipolar gain of 5, with:

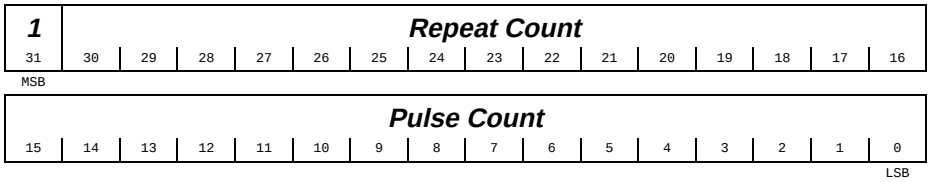
```
Gain2Code (-5)

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

Repeat Mode Sampling on the DAS-1702HR-DA

The DAS-1702HR-DA provides an option to sample the same channel one to 4095 times before advancing to the next channel in the channel/gain queue.

To set up repeat mode sampling, choose a rate or burst timing event and set the *pulses* field as follows:

Pulses Field with a Repeat Count




```
Pulses      = PULSE_REPEAT_COUNT_FLAG
            + RepeatCount × (PULSE_COUNT_MASK + 1)
                  AND PULSE_REPEAT_COUNT_MASK
            + PulseCount AND PULSE_COUNT_MASK
```

Analog Input Expansion Channels

Multiplexers can expand the number of analog input channels from the 16 base channels up to 256 differential analog input channels. The DAS-1700 Series hardware automatically switches the multiplexer channels, allowing you to specify expansion channels numbers along with base channels numbers in a channel list.

To enable DriverLINX to use multiplexers, enable expansion mode in the *Configure DAS-1700 Options* dialog (see Special…). With expansion mode enabled, DriverLINX considers the board to have the original 16 base channels followed by 256 expansion channels.

DriverLINX uses a static numbering scheme for attaching multiplexers. Attaching or removing a mux from a base channel doesn't change the Logical Channel number of any other channel. DriverLINX reserves a fixed number of expansion channels for each potential mux, whether it is attached or not.

To determine the DriverLINX Logical Channel number for a multiplexer channel, use the following formula or refer to the table that follows it. Note that DriverLINX uses 0-based numbering for all channels.

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

Term	Description
<logical chan#>	Logical Channel number to use in channel lists.
<num base chan>	Number of base channels on the DAS-1700 board
<base chan#>	Base channel on the DAS-1700 board where you attached the mux.
<num mux chan>	Number of expansion channels DriverLINX reserves for the mux. (16 for DAS-1700 expansion accessories).
<mux chan#>	Channel on the expansion board where you attached the signal. Mux channels are numbered from 0 to 15.

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

16 + 3 × 16 + 4 = 68.

To specify multiplexer input channels 0, 1, and 2 on an expansion board connected to base channel 0, use 16, 17, and 18 in the channel/gain list.

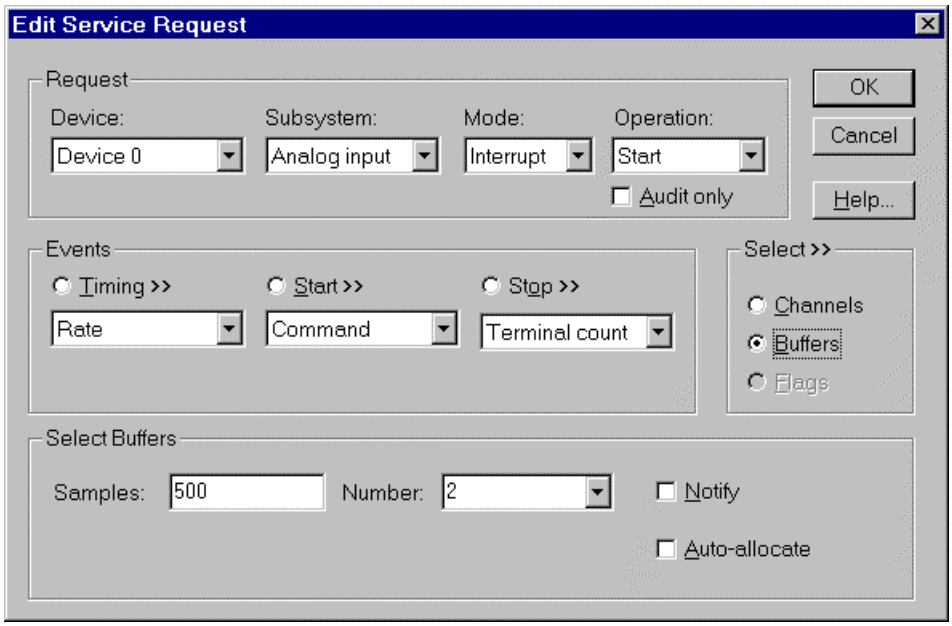
	Base Chan #								
Mux Chan #	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**etc**
0	16	32	48	64	80	96	112	128	
1	17	33	49	65	81	97	113	129	
2	18	34	50	66	82	98	114	130	
3	19	35	51	67	83	99	115	131	
4	20	36	52	68	84	100	116	132	
5	21	37	53	69	85	101	117	133	
6	22	38	54	70	86	102	118	134	
7	23	39	55	71	87	103	119	135	
8	24	40	56	72	88	104	120	136	
9	25	41	57	73	89	105	121	137	
10	26	42	58	74	90	106	122	138	
11	27	43	59	75	91	107	123	139	
12	28	44	60	76	92	108	124	140	
13	29	45	61	77	93	109	125	141	
14	30	46	62	78	94	110	126	142	
15	31	47	63	79	95	111	127	143	

Table of logical channel numbers for DAS-1700 expansion boards.

Analog Input Buffers

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

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



How to set up the DAS-1700 Series to store samples in buffers.

Buffer Size

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

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

Buffer Usage

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

Compatibility with the KPCI-1800 Series

Keithley's KPCI-1800 Series consists of PCI bus-based data-acquisition boards with features similar to the DAS-1800HC models. Developers writing applications they want to be compatible with both series should be aware of the following special requirements for DMA mode buffering on the KPCI-1800 Series.

The KPCI-1800 Series performs direct memory access (DMA) using the PCI bus mastering mode. To maintain the high efficiency that bus mastering provides, the hardware transfers samples in pairs (four bytes at a time) to memory on 4-byte boundaries. This feature has the following consequences:

- *Each buffer must have an even number of samples (a multiple of four bytes). If an application requests a DMA-mode task with an odd number of samples per buffer, DriverLINX returns an "Invalid # of conversions" error.*
- *A sample taken on an odd-numbered clock pulse is not available until the next even-numbered clock pulse. At high sample rates with continuous internal or external clocking, this effect is imperceptible. At very low sample rates, you may observe applications taking twice as long as expected to acquire a sample. With an external clock that produces an odd number of pulses and then stops, the application will not receive the last sample.*

Applications that require an odd number of samples per buffer or use an external clock with an odd, finite number of clock pulses should use interrupt mode instead of DMA mode. For existing applications that use DMA mode but do not meet these requirements, the user can cause DriverLINX to substitute interrupt mode for DMA mode by disabling bus mastering in the KPCI-1800's Configure DriverLINX Device dialog.

Analog Input Data Coding

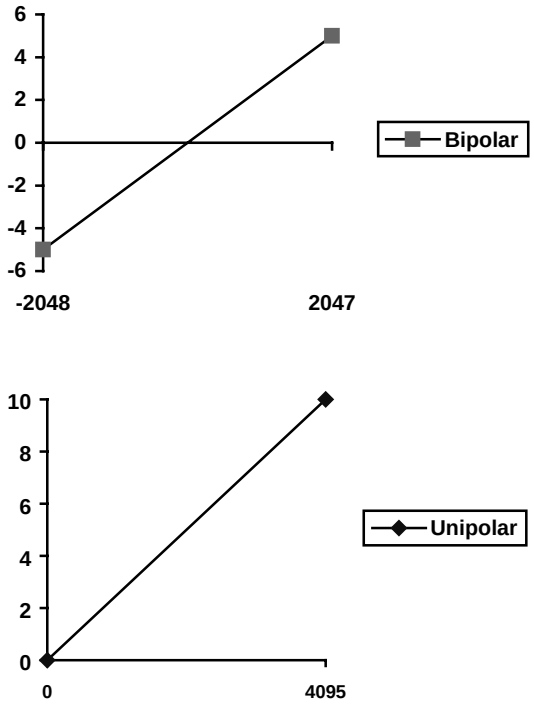
DAS-1700 Series models return Analog Input hardware codes using binary integers for unipolar ranges and offset binary for bipolar ranges. DriverLINX refers to these coding schemes as the "native" format.

For any programmable gain, the DAS-1700 models return hardware codes with the ranges in the following table:

Analog Input Resolution	Polarity	Analog Input Hardware Code
12 bits	Unipolar	0 to 4095
12 bits	Bipolar	-2048 to 2047 (right-justified *without* sign extension)*
16 bits (HR models)	Unipolar	0 to 65535
16 bits (HR models)	Bipolar	-32768 to 32767

Native Analog Input hardware codes for each DAS-1700 Series polarity.

* The KPCI-1800 Series hardware codes are left justified. For compatibility convert to a standard data type with DriverLINX conversion functions.



DAS-1700 Series native Analog Input Codes versus Voltage Range (except HR models).

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

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

Analog Input Messages

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

DriverLINX Message	Explanation
Service Start	DriverLINX has started the acquisition task.
Service Done	DriverLINX has completed the acquisition task.
Buffer Filled	DriverLINX has filled an analog input 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 an analog input data overrun condition.
Critical Error	DriverLINX has encountered an unexpected hardware or software condition.

DriverLINX Event messages for analog input.

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

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

Analog Output Subsystem

The following sections describe how DriverLINX implements Analog Output Subsystem features for the DAS-1700 Series.

Analog Output Modes

The Analog Output Subsystem supports the following modes:

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

Analog Output Operations

The DAS-1700 Series Analog Output Subsystem supports the following DriverLINX operations:

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

Analog Output Pacing, Triggering and Gating Options

The DAS-1700 Series *User's Guides* describe several pacing, triggering and gating options available on DAS-1700 models. Most are available only on the DAS-1700AO models. The following table summarizes these options and identifies which Service Request properties use them. Except as indicated all tasks must use Interrupt or DMA mode.

Parameter	Option	Service Request Properties
Pacing Mode		
	Periodic (paced)	Rate generator timing event
Clock Source		
	Software	Single-value or single-scan (Polled mode)
	Internal D/A clock	Rate timing event with an internal clock source
	Internal A/D clock	Rate timing event with an internal clock source. A period of 0 signifies that the A/D task operates the clock, which controls both tasks
	External +/-	Rate timing event with an external clock source Digital timing event
Trigger		
	Internal (software)	Command start event Command stop event Terminal count stop event
	Digital +/-	Digital start event
	Retrigger	Digital start event (Interrupt mode; task must meet certain conditions)
Gate		
	Level +/-	Rate timing event Digital start event (for use with a digital timing event)

Analog Output Timing Events

Timing Events specify how the hardware paces or clocks the writing of analog output samples. DriverLINX uses the Timing Event to program when the DAS-1700 Series writes the next analog output sample.

The DAS-1700 Series supports the following Timing Events:

- **None**—Task requires no pacing as DriverLINX is writing only a single value or scan.
- **Rate**—The DAS-1700 Series supports fixed rate writing using internal and external clocks. You can also synchronize an Analog Output task with an Analog Input task using a Rate Event. See "Rate Generator: Internal Clocking" on page 67 or "Rate Generator: External Clocking" on page 69.
- **Digital**—DriverLINX uses an external digital output signal to pace the timing of the next sample.

None or Null Timing Event

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

Rate Timing Event

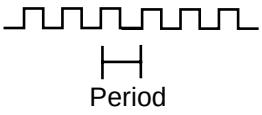
The DAS-1700 Series supports one type of Rate Event for analog output:

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

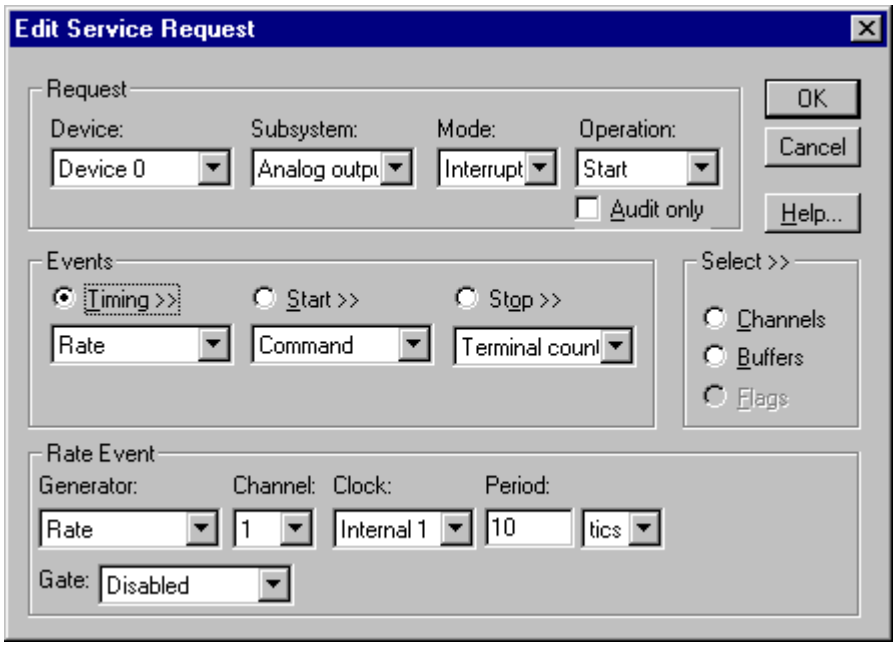
DAS-1700 boards have a 5 MHz master clock frequency, or 0.2 μs tic period. The sample period can range from 20 tics (2μs) to $10\times(2^{16}-1)$ tics (0.131 s). This means the sample rate can range from 7.63 Hz to 250 kHz.

Rate Generator: Internal Clocking

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



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



How to set up the DAS-1700 Series for fixed rate writing using an internal clock.

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

- Specify internal clocking using a **Rate** *Generator* on *Channel* **1** with the **Internal 1** *Clock* source. See "Counter/Timer Subsystem" on page 101 for a description of clock sources.
- The *Period* property specifies the time interval between samples in tics, where an **Internal 1** tic is 0.2 µs, or 5 MHz. The minimum period is 20 tics, or 250 kHz. The maximum period is 655350 tics $(10\times(2^{16}-1))$, or 7.63 Hz.
- The *Gate* property specifies how the TGIN signal affects the operation of the internal clock. Valid settings are Enabled, Disabled, High Level and Low Level. See "Counter/Timer Subsystem" on page 101 for a description of each *Gate* setting.

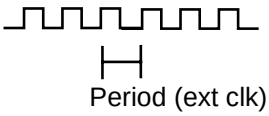
  Note: You cannot use a gated clock with a digital start or stop trigger.

- Check the *Simultaneous* box in the *Channels* section. Multi-channel analog output on the DAS-1700 Series models is always simultaneous.

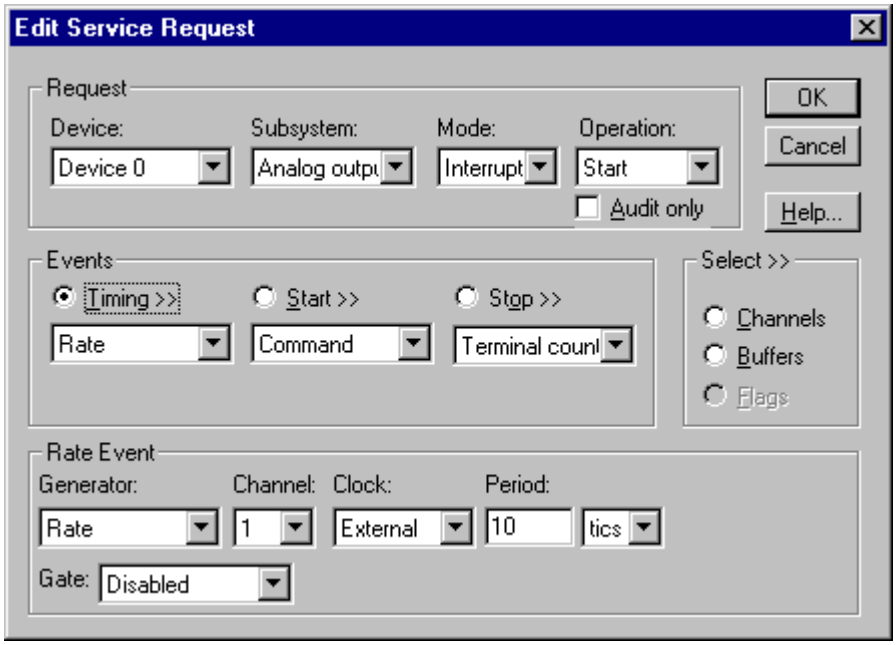
Also see: “Synchronizing an Analog Output Task with an Analog Input Task” on page 72.

Rate Generator: External Clocking

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



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



How to set up the DAS-1700 Series for fixed rate sampling using an external clock.

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

- Specify external clocking using a **Rate** *Generator* on *Channel* **1** with an **External, External+**, or **External-** *Clock* source. See "Counter/Timer Subsystem" on page 101 for a description of clock sources. For hardware-independence, you can specify the hardware external clock channel by the symbolic constant, *DI_EXTCLK*.
- Users should connect the external clock signal to the XPCLK line.
- Specify a *Period* between the minimum and maximum external clocking period. The value doesn't affect the external clock frequency, but DriverLINX requires a valid hardware value in case the application requests a timebase operation and to optimize data transfer between the driver and the application.
- The frequency of the external clock must not exceed 250 kHz.

- The *Gate* property specifies how the TGIN signal affects the operation of the internal clock. Valid settings are Enabled, Disabled, High Level and Low Level. See “Counter/Timer Subsystem” on page 101 for a description of each *Gate* setting.

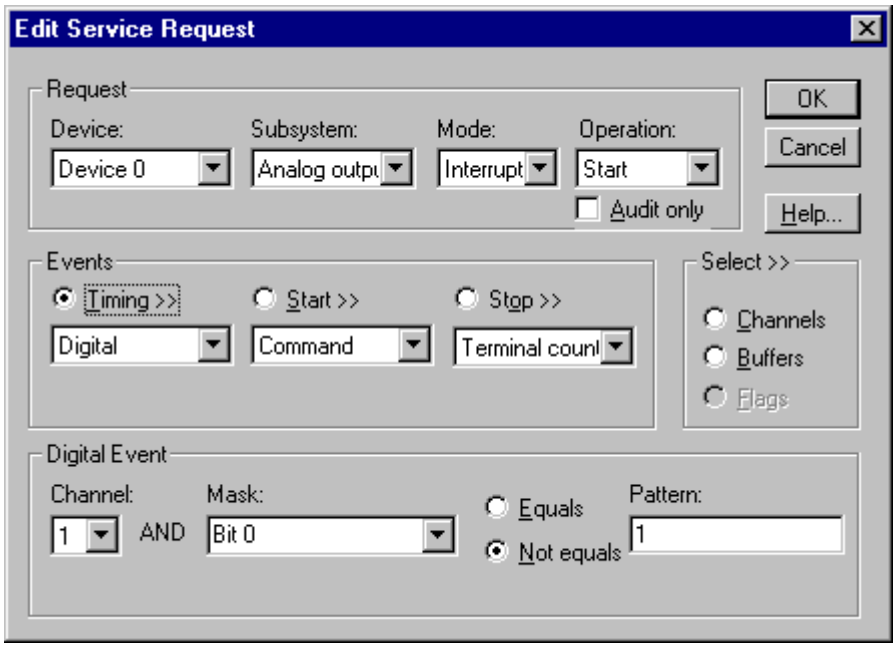
  Note: You cannot use a gated clock with a digital start or stop trigger.

- Check the *Simultaneous* box in the *Channels* section. Multi-channel analog output on the DAS-1700 Series models is always simultaneous.

Also see: “Synchronizing an Analog Output Task with an Analog Input Task” on page 72.

Digital Timing Event

DriverLINX supports Digital Events as aliases for externally clocked Rate Generators. Use this technique for compatibility with data-acquisition products that only support external clock sources, such as some digital I/O products.



How to set up the DAS-1700 Series for external rate sampling using a digital event.

- Specify external clocking using *Channel* **1.** 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 the XPCLK line.
- Specify the *Mask* property as **1,** or **Bit 0**, to indicate that DriverLINX should only compare a 1-bit digital input value against the *Pattern* property.
- Specify the *Match* property as **Not equals**.
- Specify the *Pattern* property as **0** for a rising, or positive, edge clock (≠0) or **1** for a falling, or negative, edge clock (≠1).

Synchronizing an Analog Output Task with an Analog Input Task

On the DAS-1700 Series, DriverLINX can synchronize an analog output task with an analog input task. The analog output task uses the same timing event as the input task.

To synchronize an analog output (AO) task with an analog input (AI) task:

1. Set up the AI service request.
2. Set up the AO service request using an identical timing event as the AI task, except with set the *Period* to zero. Both the AI and AO timing event must use Counter/Timer channel 0.
3. Submit the AO service request. The hardware runs the AO task only while the AI task is running.
4. Submit the AI service request to start both tasks.

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

Analog Output Start Events

Start Events specify when the DAS-1700 hardware starts acquiring analog output data.

The DAS-1700 Series supports the following Start Events:

- **None**—Use this event when the DriverLINX operation does not require a Start Event.
- **Command**—DriverLINX starts the task on software command, i.e., as soon as DriverLINX finishes programming the DAS-1700 hardware for the task.
- **Digital**—The DAS-1700 starts acquiring analog output samples when the hardware detects the digital Logical Channel output satisfies the condition specified in the Start Event.

None or Null Start Event

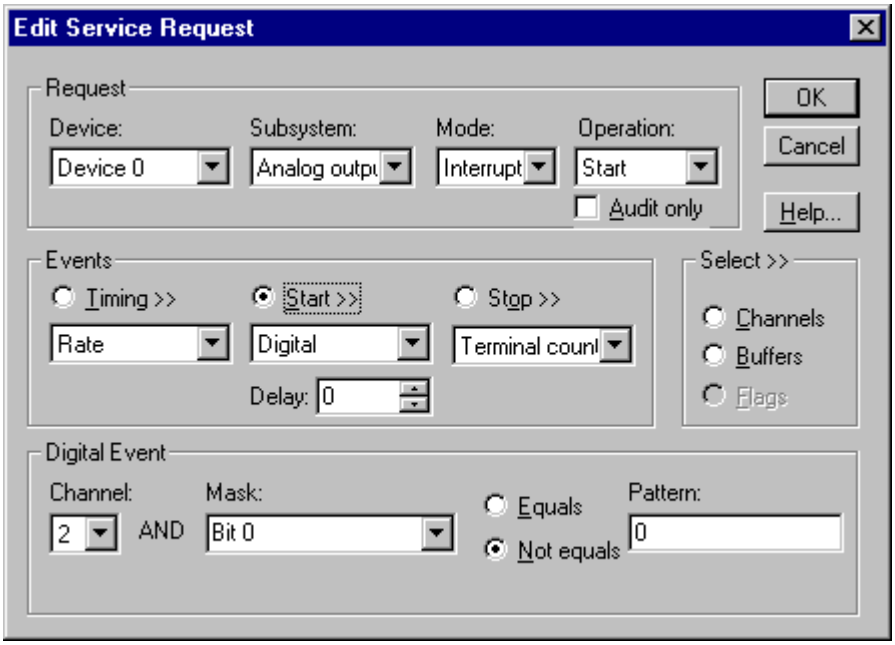
The Null Event specifies that the task does not need a Start Event to begin the task.

Command Start Event

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

Digital Start Event (Post-Triggering)

The DAS-1700 can write analog output samples ***after*** the hardware detects a digital trigger condition. Use post-triggering when you want to synchronize the start of data acquisition with an external signal.



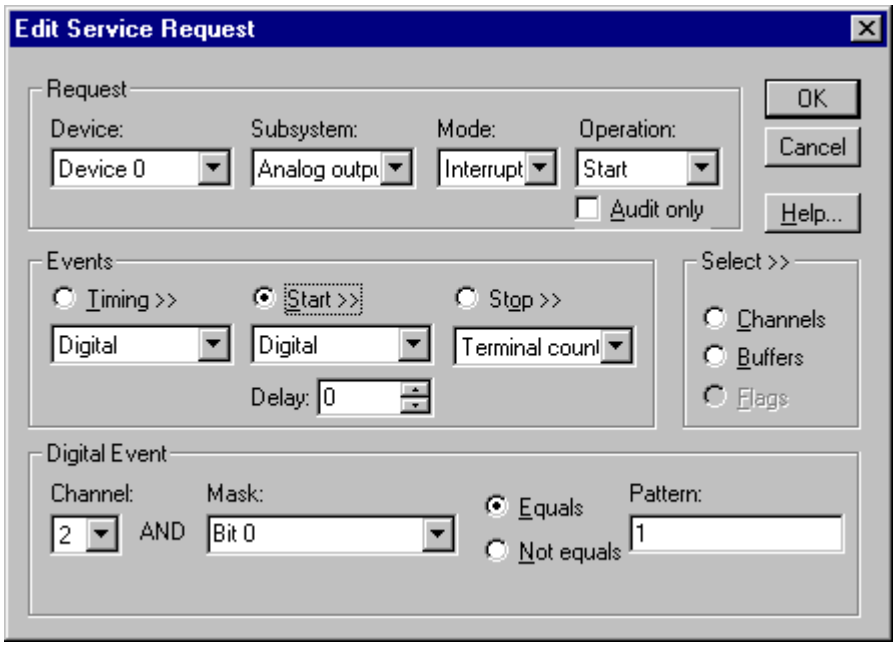
How to set up the DAS-1700 Series for post-triggered analog output.

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

- Specify the *Channel* as **2**. For hardware-independence, you can specify the hardware external trigger channel by the symbolic constant, *DI_EXTTRG*.
- Users should connect the external trigger signal to the TGIN line.
- Specify the *Mask* property as **1**, or **Bit 0**, to indicate that DriverLINX should only compare a 1-bit digital input value against the *Pattern* property.
- Specify the *Match* property as **Not equals**.
- Specify the *Pattern* property as **0** for a rising, or positive, edge trigger (≠0) or **1** for a falling, or negative, edge trigger (≠1).
- Specify the *Delay* property as 0. The DAS-1700 does not support a delay in sampling after the start trigger.
- You cannot use a digital start trigger with a gated clock.

Digital Start Event (Gating)

The DAS-1700 can pause and resume writing analog output samples based on the level of a digital input condition. Use a digital start event to set up gating when using a digital timing event. To set up gating for a rate timing event, see “Rate Timing Event” on page 66.



How to set up the DAS-1700 Series for gated analog output.

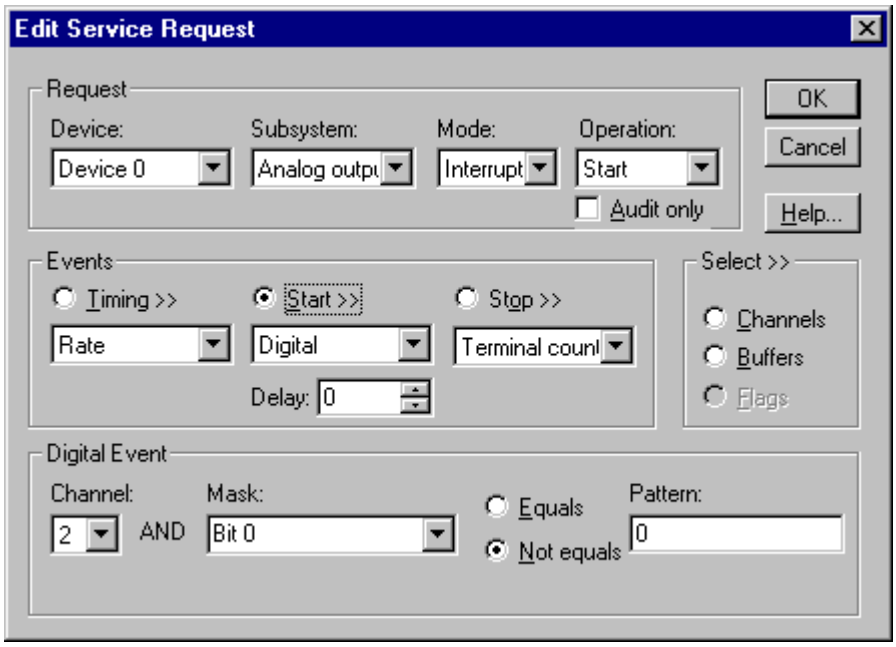
Digital Start Events contain *mask*, *pattern*, and *match* fields. The mask selects which bit in the Logical Channel to use as a gate input. The *pattern* selects whether the gate is active, or enables data-acquisition, when the gate signal is high or low.

- Specify the *Channel* as **2**. For hardware-independence, you can specify the hardware external trigger channel by the symbolic constant, *DI_EXTTRG*.
- Users should connect the external trigger signal to the TGIN line.
- Specify the *Mask* property as **1**, or **Bit 0**, to identify the gate input bit of the Logical Channel.
- Specify the *Match* property as **Equals**.
- Specify the *Pattern* property as **0** for a low-level active gate or **1** for a high-level active gate.
- Specify the *Delay* property as 0. DriverLINX does not use a delay with gating.

Digital Start Event (Retriggering)

In recycle mode, a DAS-1700AO continuously writes analog output samples from its FIFO buffer. A digital trigger can restart the output at the beginning of the FIFO buffer. Use retriggering when you want to repeatedly synchronize the start of a waveform output with an external signal.

To use recycle mode, the device configuration must enable it and the task must meet certain conditions. See “Special…” on page 15 for more information.



How to set up the DAS-1700 Series for post-triggered analog output.

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

- Specify the *Channel* as **2**. For hardware-independence, you can specify the hardware external trigger channel by the symbolic constant, *DI_EXTTRG*.
- Users should connect the external trigger signal to the TGIN line.
- Specify the *Mask* property as **1**, or **Bit 0**, to indicate that DriverLINX should only compare a 1-bit digital input value against the *Pattern* property.
- Specify the *Match* property as **Not equals**.
- Specify the *Pattern* property as **0** for a rising, or positive, edge trigger (≠0) or **1** for a falling, or negative, edge trigger (≠1).
- Specify the *Delay* property as 0. The DAS-1700 does not support a delay in sampling after the start trigger.
- You cannot use a digital start trigger with a gated clock.

Analog Output Stop Events

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

The DAS-1700 Series supports the following Stop Events:

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

None or Null Stop Event

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

Command Stop Event

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

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

Terminal Count Stop Event

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

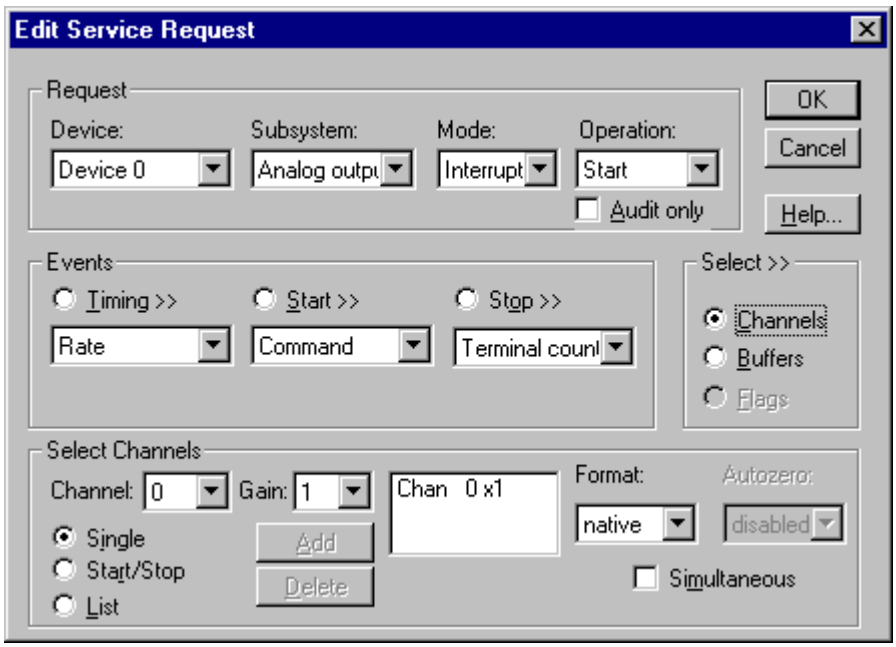
Analog Output Channels

The DAS-1700 Series allows applications to specify the analog channels using three techniques:

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

Single Channel Analog Output

In single channel mode, the DAS-1700 Series writes all data to one channel at the specified gain.

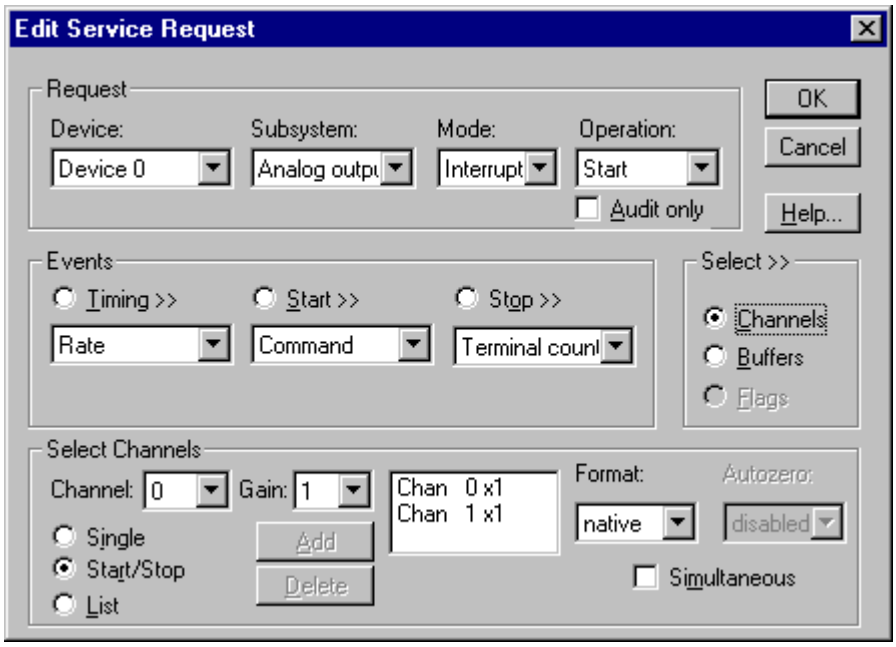


How to set up the DAS-1700 Series for writing to sampling on a single channel.

Multi-channel Analog Output Range

In multi-channel range mode, the DAS-1700 Series writes data to a consecutive range of analog channels.

- The Start Channel's gain only applies to the first channel.
- DriverLINX uses the Stop Channel's gain for all the other analog channels in the range.
- The Stop Channel must be greater than the Start Channel.

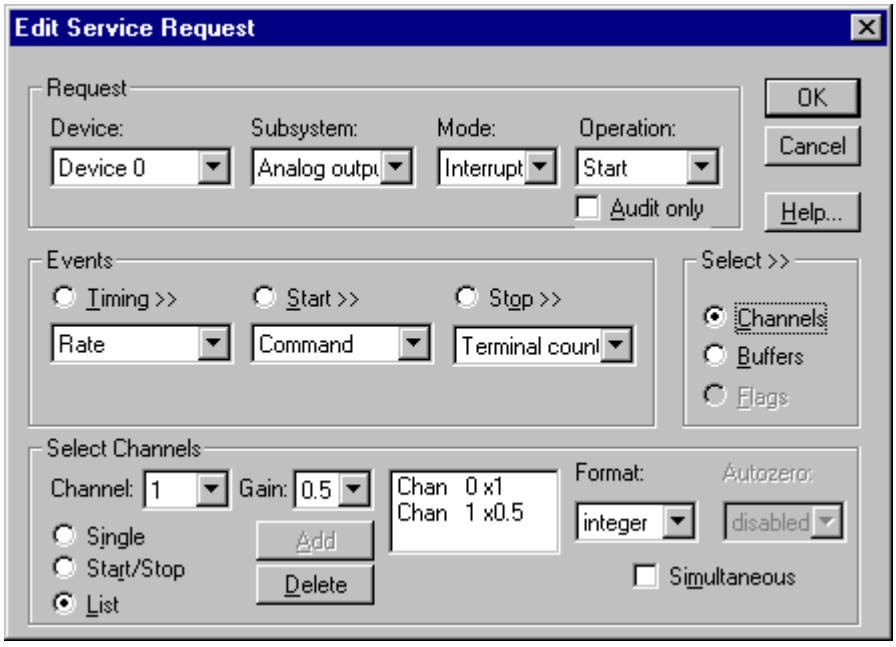


How to set up the DAS-1700 Series for writing to a consecutive range of channels.

Multi-channel Analog Output List

In multi-channel list mode, the DAS-1700 Series writes data to an arbitrary list of analog channels.

- The channel-gain list may contain up to channels in any order and with any allowed gain but may repeat any channel.



How to set up the DAS-1700 Series to write to an arbitrary list of channels.

Analog Output Channel Gains

The DAS-1700 Series models support a variety of channel gains. The following table shows the correspondence between DriverLINX gains, the maximum output signal range, and the gain code for each output range. Note: DriverLINX uses a negative (-) gain value to signify a bipolar (±) range.

AO Models

Gain	Range	Gain Code
-1	±5 V	0
-0.5	±10 V	1

Gains, Ranges, and Gain Codes for AO Models.

DA Models

Gain	Range	Gain Code
-1	±10 V	0

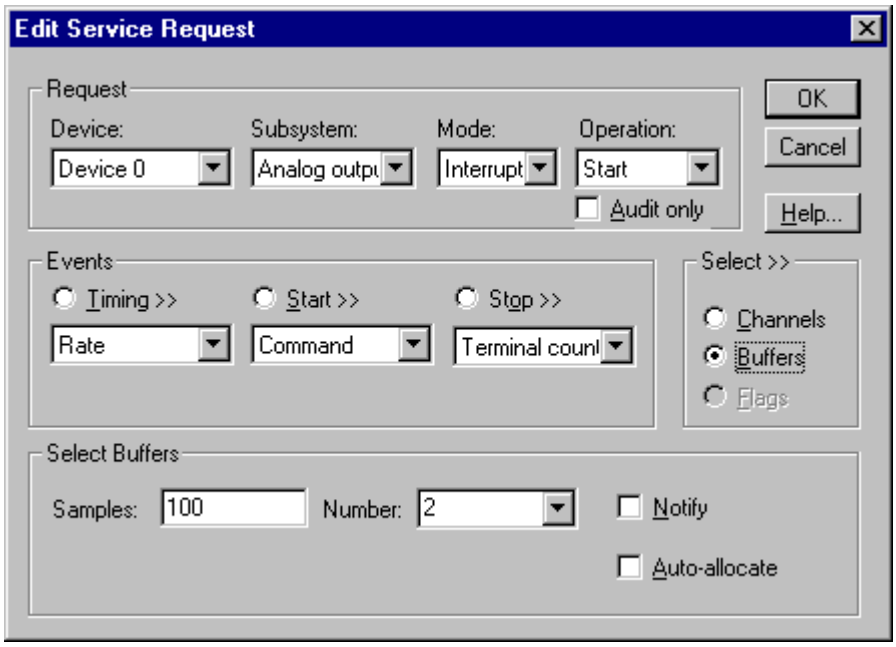
Gains, Ranges, and Gain Codes for DA Models.

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

Analog Output Buffers

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

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



How to set up the DAS-1700 Series to store samples in buffers.

Buffer Size

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

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

Buffer Usage

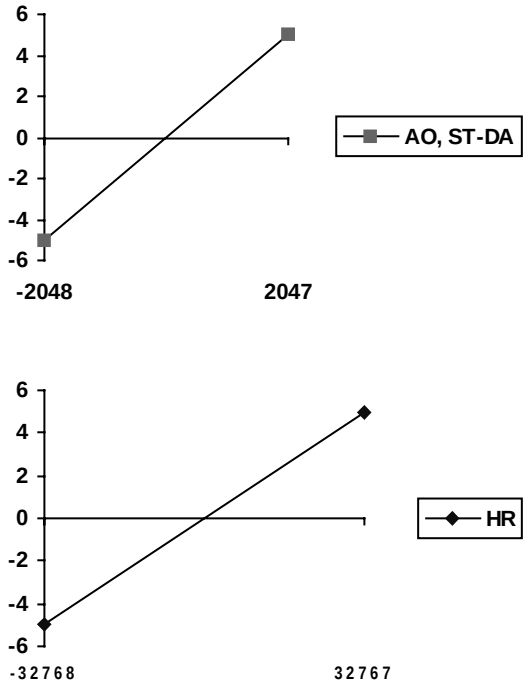
DriverLINX writes from buffers sequentially until the task stops. Except for tasks that stop on terminal count, the last buffer may be only partially used. If the task stops on a trigger, use the StopEvent message (or event) to determine the location of the last sample. For other cases, use a Status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

Analog Output Data Coding

DAS-1700 Series models use a variety of formats to encode analog output data, as shown in the following table. DriverLINX refers to these coding schemes as the "native" format.

Model	Format	Analog Output Resolution	Analog Output Hardware Code
AO, ST-DA	Two's complement, right-shifted	12 bits	-2048 to 2047
HR-DA	Two's complement	16 bits	-32768 to 32767

Native Analog Output hardware codes for each DAS-1700 Series resolution.



DAS-1700 Series native Analog Output Codes versus Voltage Range at unity gain.

DriverLINX refers to the default hardware analog-coding scheme as the "native" format. For computer arithmetic in a higher level language, the 16-bit two's complement integer format is generally easier to use.

DriverLINX automatically converts digital-to-analog codes to integer format, if you specify **integer** for the *Format* property. Or, applications can use DriverLINX's data conversion operations to transform an entire data buffer from many common integer and floating-point formats to native format.

Analog Output Messages

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

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

DriverLINX Event messages for analog output.

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

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

Digital Input Subsystem

The following sections describe how DriverLINX implements Digital Input Subsystem features for the DAS-1700 Series.

Digital Input Modes

The Digital Input Subsystem supports the following modes:

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

Digital Input Operations

The DAS-1700 Series Digital Input Subsystem supports the following DriverLINX operations:

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

Digital Port Configuration

The DAS-1700 Series has separate, dedicated digital input and output ports and doesn't require the application to configure its digital I/O ports.

Digital Input Pacing, Triggering and Gating Options

The DAS-1700 Series *User's Guides* describe several pacing, triggering and gating options available for analog input on DAS-1700 models. As DriverLINX uses the analog input pacer clock for digital input, many of these options also apply to digital input tasks. The following table summarizes these options and identifies which Service Request properties use them. Except as indicated all tasks must use Interrupt or DMA mode.

Parameter	Option	Service Request Properties
Pacing Mode		
	Periodic (paced)	Rate generator timing event
Clock Source		
	Software	Single-value or single-scan (Polled mode)
	Internal	Rate timing event with an internal clock source
	External +/-	Rate timing event with an external clock source Digital timing event
Trigger		
	Internal (software)	Command start event Command stop event Terminal count stop event

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 DAS-1700 Series reads the next digital input sample from the port.

The DAS-1700 Series supports the following Timing Events:

- **None**—Input requires no pacing as DriverLINX is reading only a single value.
- **Rate**—The DAS-1700 Series supports fixed rate writing using internal and external clocks.

None or Null Timing Event

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

Rate Timing Event

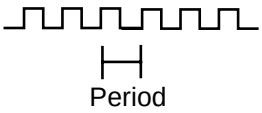
The DAS-1700 Series supports one type of Rate Event for digital output:

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

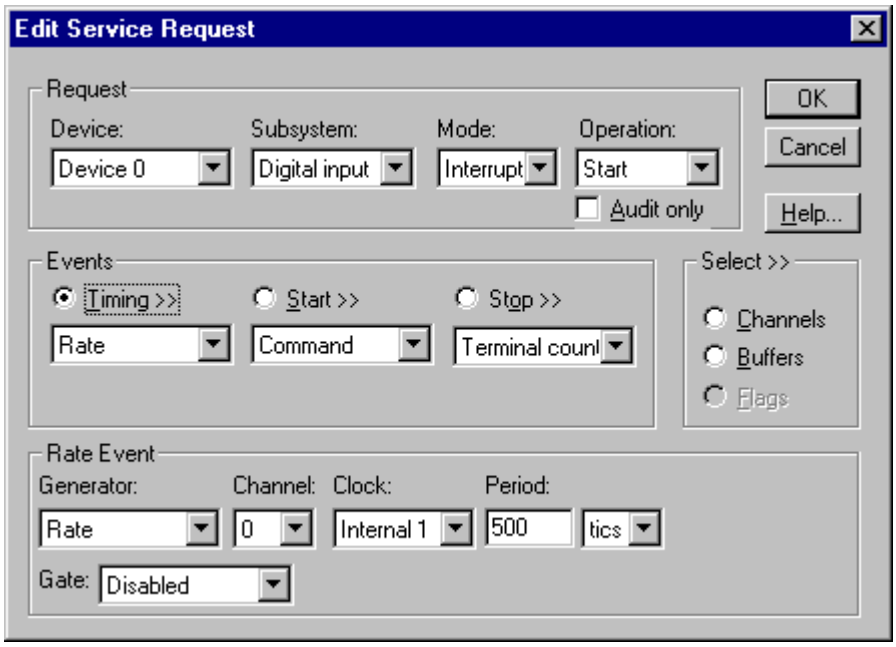
DAS-1700 boards have a 5 MHz master clock frequency, or 0.2 μs tic period. The sample period can range from 500 tics (100 μs) to $2^{32} - 1$ tics (833 s). This means the sample rate can range from 0.00116 Hz to 10kHz.

Rate Generator: Internal Clocking

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



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



How to set up the DAS-1700 Series for fixed rate sampling using an internal clock.

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

- Specify internal clocking using a **Rate** *Generator* on *Channel* **0** with the **Internal 1** *Clock* source. See "Counter/Timer Subsystem" on page 101 for a description of clock sources.
- The *Period* property specifies the time interval between samples in tics, where an **Internal 1** tic is 0.2 µs, or 5 MHz. The minimum period is 500 tics, or 10 kHz. The maximum period is 4294967295 tics ($2^{32}-1$), or 0.00116 Hz.
- Digital input does not support gating. Set the *Gate* property to **Disabled** or **NoConnect**.

Digital Input Start Events

Start Events specify when the DAS-1700 Series hardware starts reading digital input data.

The DAS-1700 Series supports the following Start Events for digital input:

- **None**—Use this event when the DriverLINX operation doesn't require a Start Event.
- **Command**—DriverLINX starts the task on software command, i.e., as soon as DriverLINX finishes programming the DAS-1700 hardware for the task.

None or Null Start Event

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

Command Start Event

The Command Event starts data acquisition as soon as DriverLINX has completed programming the DAS-1700 Series hardware with the task parameters.

Digital Input Stop Events

Stop Events specify when the DAS-1700 Series hardware stops reading digital input data.

The DAS-1700 Series supports the following Stop Events for digital input:

- **None**—Use this event when the DriverLINX operation doesn't require a Stop Event.
- **Command**—DriverLINX stops the task on software command, i.e., when the application issues a Service Request with a *Stop* operation.
- **Terminal count**—DriverLINX stops the task after the DAS-1700 Series hardware has filled all the data buffers once.

None or Null Stop Event

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

Command Stop Event

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

In Stop-on-Command mode, DriverLINX continuously cycles through all the data buffers, reading from the digital port on the DAS-1700 Series.

Terminal Count Stop Event

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

Digital Input Channels

The DAS-1700 Series allows applications to specify the digital channels using three techniques:

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

Digital Input Logical Channels

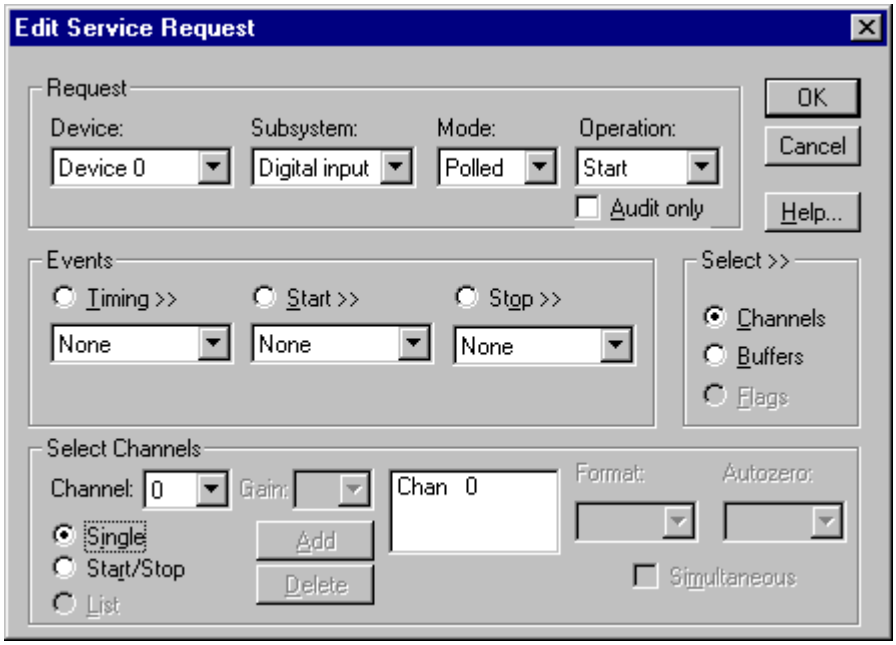
The DAS-1700 Series has a single digital input port that DriverLINX designates as Logical Channel 0. DriverLINX defines two additional Logical Channels for the external clock and trigger signals but applications cannot directly read their values.

DriverLINX defines the following Logical Channels for the DAS-1700 Series digital inputs:

Logical Channel	DriverLINX Function	DAS-1700 Series External Connector
0	Standard Digital Input	DI 0 … DI 3
1	External Clock	XPCLK
2	External Trigger	TGIN

Single Channel Digital Input

In single channel mode, the DAS-1700 Series acquires all data from one channel.



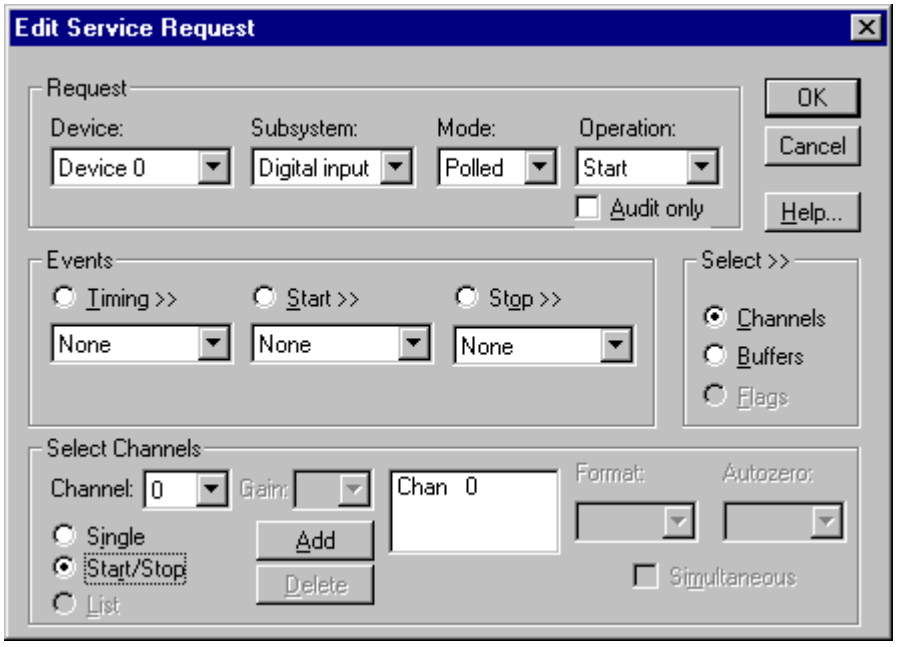
How to set up the DAS-1700 Series to read from a single channel.

Multi-channel Digital Input Range

Even though the DAS-1700 Series has only one digital input channel, DriverLINX supports specifying a channel range for compatibility with applications that use this method.

In multi-channel range mode, the DAS-1700 Series acquires all data from a consecutive range of digital channels.

- Both the Start and Stop Channel must specify channel 0.

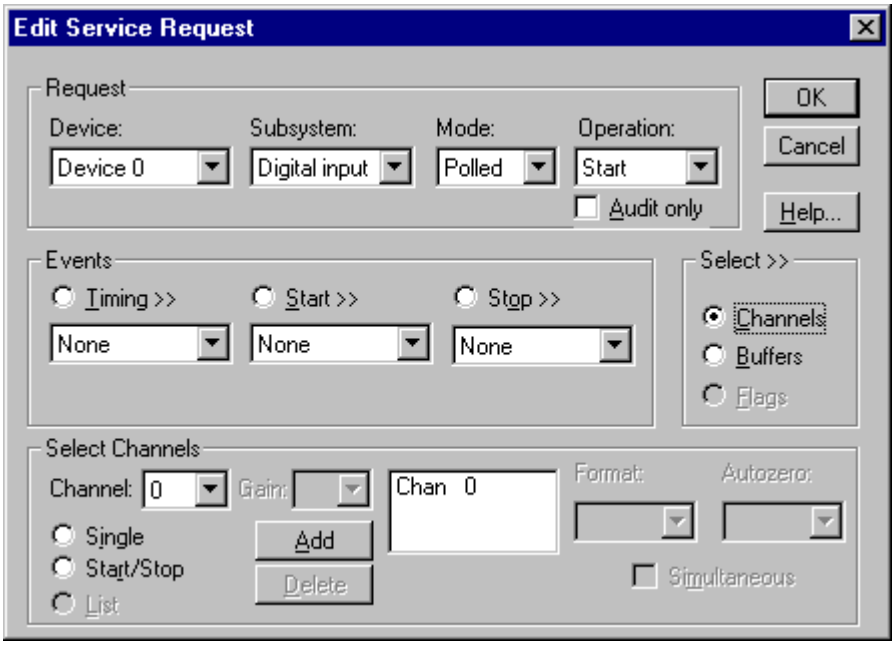


Multi-channel Digital Input List

Even though the DAS-1700 Series has only one digital input channel, DriverLINX supports specifying a channel range for compatibility with applications that use this method

In multi-channel list mode, the DAS-1700 Series acquires all data from a random list of digital channels.

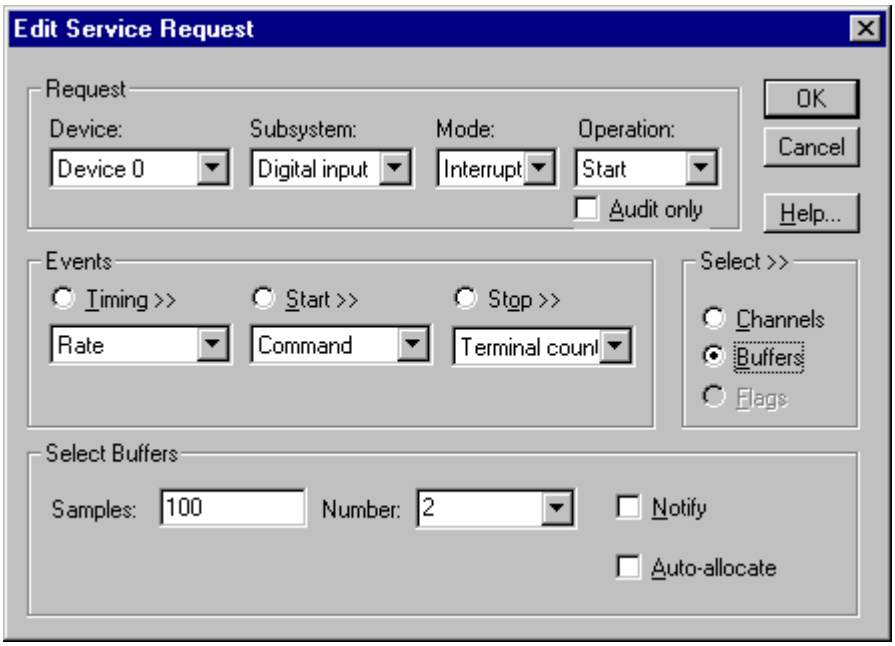
- The channel list may contain only one channel.
- As the DAS-1700 Series only has a single digital input channel available for reading, this technique is equivalent to Single Channel Digital Input.



Digital Input Buffers

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

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



How to set up the DAS-1700 Series to read digital samples using data buffers.

Buffer Usage

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

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 digital input buffer.
Critical Error	DriverLINX has encountered an unexpected hardware or software condition.

DriverLINX Event messages for digital input.

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

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

Digital Output Subsystem

The following sections describe how DriverLINX implements Digital Output Subsystem features for the DAS-1700 Series.

Digital Output Modes

The Digital Output Subsystem supports the following modes:

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

Digital Output Operations

The DAS-1700 Series Digital Output Subsystem supports the following DriverLINX operations:

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

Digital Output Initialization

By default, the Digital Output subsystem writes zero into the digital output port. You can specify a different initial output value using the *Configure DriverLINX Device* dialog. See "Digital Output Subsystem Page" on page 24.

Digital Output Pacing, Triggering and Gating Options

The DAS-1700 Series *User's Guides* describe several pacing, triggering and gating options available for analog input on DAS-1700 models. As DriverLINX uses the analog input pacer clock for digital output, many of these options also apply to digital output tasks. The following table summarizes these options and identifies which Service Request properties use them. Except as indicated all tasks must use Interrupt or DMA mode.

Parameter	Option	Service Request Properties
Pacing Mode		
	Periodic (paced)	Rate generator timing event
Clock Source		
	Software	Single-value or single-scan (Polled mode)
	Internal	Rate timing event with an internal clock source
	External +/-	Rate timing event with an external clock source Digital timing event
Trigger		
	Internal (software)	Command start event Command stop event Terminal count stop event

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 DAS-1700 Series writes the next digital output sample from the port.

The DAS-1700 Series supports the following Timing Events:

- **None**—Output requires no pacing as DriverLINX is writing only a single value.
- **Rate**—The DAS-1700 Series supports fixed rate writing using internal and external clocks.

None or Null Timing Event

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

Rate Timing Event

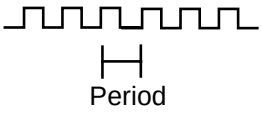
The DAS-1700 Series supports one type of Rate Event for digital output:

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

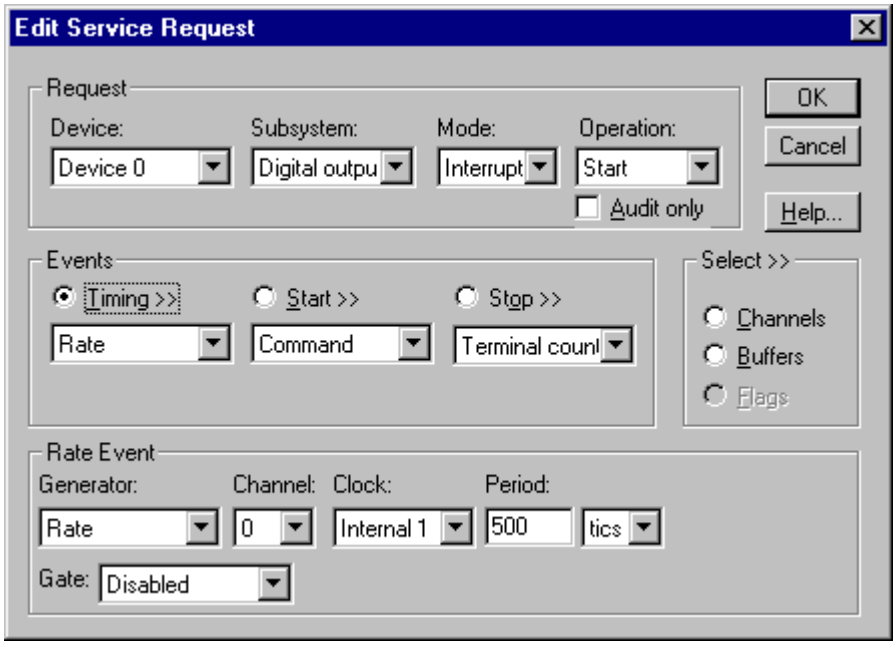
DAS-1700 boards have a 5 MHz master clock frequency, or 0.2 µs tic period. The sample period can range from 500 tics (100 µs) to $2^{32}-1$ tics (833 s). This means the sample rate can range from 0.00116 Hz to 10kHz.

Rate Generator: Internal Clocking

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



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



How to set up the DAS-1700 Series for fixed rate writing using an internal clock.

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

- Specify internal clocking using a **Rate** *Generator* on *Channel* **0** with the **Internal 1** *Clock* source. See "Counter/Timer Subsystem" on page 101 for a description of clock sources.
- The *Period* property specifies the time interval between samples in tics, where an **Internal 1** tic is 0.2 μs, or 5 MHz. The minimum period is 500 tics, or 10 kHz. The maximum period is 4294967295 tics ($2^{32} - 1$), or 0.00116 Hz.
- Digital output does not support gating. Set the *Gate* property to **Disabled** or **NoConnect**.

Digital Output Start Events

Start Events specify when the DAS-1700 Series hardware starts writing digital output data.

The DAS-1700 Series supports the following Start Events for digital output:

- **None**—Use this event when the DriverLINX operation doesn't require a Start Event.
- **Command**—DriverLINX starts the task on software command, i.e., as soon as DriverLINX finishes programming the DAS-1700 hardware for the task.

None or Null Start Event

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

Command Start Event

The Command Event starts data acquisition as soon as DriverLINX has completed programming the DAS-1700 hardware with the task parameters.

Digital Output Stop Events

Stop Events specify when the DAS-1700 Series hardware stops writing digital output data.

The DAS-1700 Series supports the following Stop Events for digital output:

- **None**—Use this event when the DriverLINX operation doesn't require a Stop Event.
- **Terminal count**—DriverLINX stops the task after the DAS-1700 Series hardware has written all the data buffers once.

None or Null Stop Event

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

Terminal Count Stop Event

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

Digital Output Channels

The DAS-1700 Series allows applications to specify the digital channels using three techniques:

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

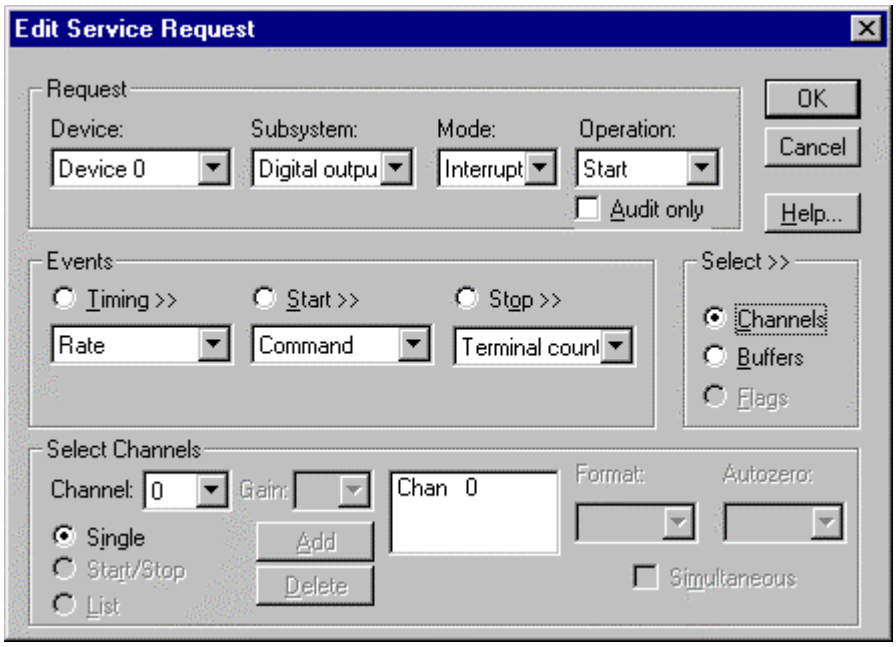
Digital Output Logical Channels

The DAS-1700 Series has one digital output port that DriverLINX designates as channel 0. And, DriverLINX designates the MUX and GEXT signal as another, 5-bit, digital output channel.

Logical Channel	DriverLINX Function	DAS-1700 Series External Connector
0	Standard Digital Output	DO 0 … DO 3
1	Standard Digital Output	MUX4 … MUX7, GEXT

Single Channel Digital Output

In single channel mode, the DAS-1700 Series writes all data from one channel.



How to set up the DAS-1700 Series to write a single digital output channel.

Multi-channel Digital Output Range

In multi-channel range mode, the DAS-1700 Series acquires all data from a consecutive range of digital channels.

- The Start and Stop Channel must specify the same channel as DriverLINX supports only single value output in the digital output subsystem.

Multi-channel Digital Output List

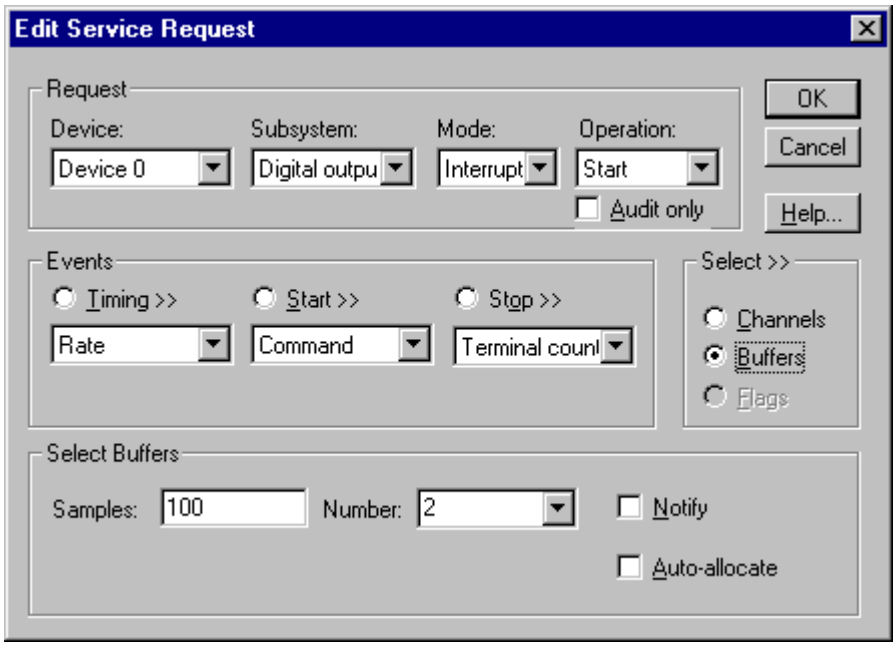
In multi-channel list mode, the DAS-1700 Series acquires all data from a random list of digital channels.

- The channel list may contain only one channel as DriverLINX supports only single value output in the digital output subsystem.

Digital Output Buffers

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

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



How to set up the DAS-1700 Series to store samples in buffers.

Buffer Size

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

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

Buffer Usage

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

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 filled an analog input buffer.
Critical Error	DriverLINX has encountered an unexpected hardware or software condition.

DriverLINX Event messages for digital output.

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

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

Counter/Timer Subsystem

The DAS-1700 Series has counter/timers for analog input/output pacing only. All models use an Intel 8254 Programmable Interval Timer that consists of 3 internal 16-bit counters, Counter 0, Counter 1, and Counter 2.

For analog input pacing, the DAS-1700 boards operate Counters 1 and 2 in a fixed divider, 32-bit configuration. The input of Counter 1 connects to a 5 MHz crystal oscillator and the output of Counter 1 connects to the input of Counter 2.

The DAS-1700 uses Counter 0 for trigger delay and its input, output, and gate control are not available.

DriverLINX implements only one Logical Counter, Logical Channel 0, which corresponds to the internal Analog Input pacing clock (8254 counters 1 and 2). You can use Logical Channel 0 to pace Analog Output or Digital I/O tasks indirectly through hardware interrupts.

The DAS-1700AO has an additional counter/timer for supporting timed Analog Output tasks in hardware. DriverLINX implements two Logical Counters in this case, Logical Counter 0, which corresponds to the internal Analog Input pacing clock, and Logical Counter 1, which corresponds to the internal Analog Output pacing clock.

See “Counter/Timer Subsystem Signals” on page 36 for connection details.

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

Logical Channels	Clocks		Modes	Gates
	Source	Tic Period		
0 AI, DI and DO Pacing	Internal External External+ External-	0.2 µs	Rate Gen Burst Gen	Enabled Disabled No Connect Low Level Low Edge High Level High Edge
1 AO Pacing (AO models only)	Internal External External+ External-	0.2 µs	Rate Gen	Enabled Disabled No Connect Low Level High Level

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

Gate Settings

The *Gate* setting specifies how the TGIN signal affects the operation of the internal or external clock. Valid settings are Enabled, Disabled, No Connect, High Level, Low Level, Low Edge and High Edge.

- **Enabled** selects the gate’s default enabled mode. On the DAS-1700 Series this is High Level Enabled.
- **Disabled** allows the clock to pace samples independently of the gate signal.

- **No Connect** specifies that if the application is running with a board cannot disable its gate, then the user should leave the gate unconnected so that it does not interfere with the data-acquisition task. This is mode is accepted by all DriverLINX drivers.
- **High and Low Level** allow the clock to pace samples only while the gate signal is high or low, respectively.
- **Low and High Edge** are aliases for a digital start trigger. They enable the clock after the first falling or rising edge, respectively.

Counter/Timer Interrupt

DriverLINX supports counter/timer interrupts indirectly. You can set up an INTERRUPT mode analog task and respond to the BufferFilled messages.

Uninstalling DriverLINX

How do I uninstall DriverLINX?

DriverLINX consists of three separate component installations:

- DriverLINX for the Keithley DAS-1700 Series
- DriverLINX Programming Interfaces
- DriverLINX Documentation

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

To uninstall DriverLINX drivers for the Keithley DAS-1700 Series, you must

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

How to Disable a DriverLINX Driver in Windows NT

1. From the Windows Start menu, select "Settings", then "Control Panel". Left click on the DriverLINX Configuration icon in the Control Panel.
2. Select the DAS-1700 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 DAS-1700 card that you are uninstalling.
5. Close the DriverLINX Configuration Panel.
6. When finished, shut down your computer and physically remove any installed DAS-1700 hardware.
7. Reboot Windows.
8. To finish uninstalling, see "How to Remove DriverLINX for the Keithley DAS-1700 Series" on page 104.

How to Disable a DriverLINX Driver in Windows 95/98

1. From the Windows Start menu, select "Settings", then "Control Panel". Left click on the System icon in the Control Panel. Select the "Device Manager" tab in the System Properties dialog.
2. Left click the "+" icon next to "DriverLINX drivers" to display the installed Keithley DAS-1700 devices.

3. Select the DAS-1700 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 DAS-1700 card or driver that you uninstalling.
7. When finished, click "Close", shut down your computer, and physically remove any installed DAS-1700 hardware.
8. Reboot Windows.
9. To finish uninstalling, see "How to Remove DriverLINX for the Keithley DAS-1700 Series" on page 104.

How to Remove DriverLINX for the Keithley DAS-1700 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 DAS-1700" 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 DAS-1700 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 DAS-1700 hardware requires successful completion of four steps.

1. Windows finds free resources for the DAS-1700 board.
2. The DAS-1700 address switches are set to the assigned address resource.
3. You configure the DAS-1700 drivers using the DriverLINX Configuration Panel.
4. Windows loads the DAS-1700 drivers into memory.

If you are having a problem installing or configuring your DAS-1700 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

The DriverLINX installation program runs a wizard that assists you in installing, registering and configuring the DriverLINX driver for your board. If you would like to repeat any steps with the wizard, click here  .

Solving Problems Configuring the Drivers

Windows 95/98 assigns hardware resources for the DAS-1700, but you must still configure the DAS-1700 drivers before using them. The DriverLINX configuration requires that you select the hardware model of your DAS-1700 board.

On Windows NT, you must, also, manually enter the address and interrupt resource assignments. See “Configuring the DAS-1700 Series” on page 11 for more information.

Solving Problems Loading Drivers

Before the DAS-1700 drivers can load, you must

1. Install the DriverLINX software.
2. Install the DAS-1700 hardware into your computer.
3. Configure DriverLINX.
4. Reboot your computer.

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

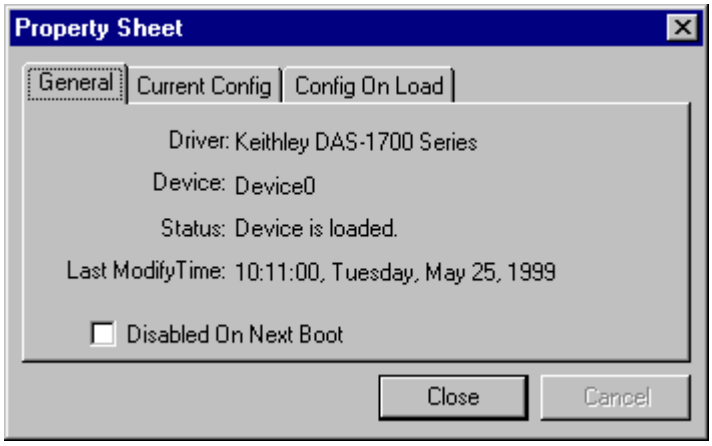
On Windows NT you must determine free hardware resources for the DAS-1700 using Windows NT Diagnostics. On Windows 95/98, the operating system will automatically assign hardware resources to the DAS-1700 cards. Automatic resource assignment can fail sometimes on

- 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 DAS-1700”. 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 “Configuring the DAS-1700 Series” on page 11.
6. Click the “Properties…” button and then select the “General” tab.
7. Do you see “Status: Device is loaded”? If not, did you reboot the computer after configuring? If not, reboot now and repeat the above steps.



8. If you rebooted the computer after configuring and Windows did not load your device, see "Checking for Device Errors" on page 107.

Checking for Device Errors

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

Windows 95/98 maintains additional driver information in the Device Manager. Also see "Getting More Driver Information on Windows 95/98" on page 107.

1. Run "DriverLINX Event Viewer" from the DriverLINX folder.
2. Click on the "+" icon next to "DriverLINX" in the left panel.
3. Select the abbreviation for your driver.
4. Does the first line in the right panel show a current error?
5. Double click on the error line to see more detail and an explanatory message.
6. If you cannot resolve the problem yourself, please provide this error information when contacting technical support.

Getting More Driver Information on Windows 95/98

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

1. Right click on "My Computer" and then select "Properties".
2. Select "Device Manager" and "View devices by type".
3. Does "DriverLINX drivers" appear in the list? If not, see "Solving Problems Installing Drivers" on page 105.
4. Click the "+" next to "DriverLINX drivers".
5. Does your DAS-1700 product appear in the list? If not, see "Solving Problems Installing Drivers" on page 105.
6. Does the icon next to your DAS-1700 product display an exclamation point (!)? If no, Windows has loaded your DAS-1700 driver.
7. Select the line with the "!" and then click "Properties".

8. The General tab will show the reason why the driver did not load.
9. The Resources tab will show if Windows detected an unresolvable hardware conflict.

Getting More Driver Information on Windows NT

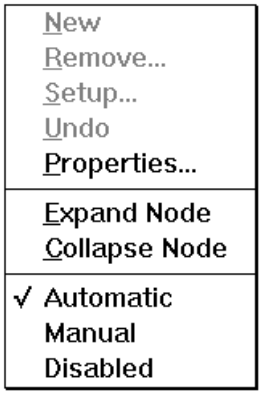
On Windows NT, the only reasons that a driver does not load are

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

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 107 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 DAS-1700”. 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 105, 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.

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.

Clocking

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

D/A

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

DAC

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

DMA

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

Event

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

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

Gating

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

IRQ

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

ISA

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

ISR

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

Logical Device

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

Messages

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

Modes

DriverLINX data-acquisition techniques.

Operations

Allowed DriverLINX data-acquisition commands.

Pacing

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

Process

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

Scan List

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

Service Request

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

Single-scan

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

Subsystem

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

Triggering

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

TTL

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