KPXI High Speed Digitizer Module

User's Manual

KPXI-AI-900-01 Rev. A / January 2007

ECA 42912

WARRANTY

Keithley Instruments, Inc. warrants this product to be free from defects in material and workmanship for a period of 1 year from date of shipment.

Keithley Instruments, Inc. warrants the following items for 90 days from the date of shipment: probes, cables, rechargeable batteries, diskettes, and documentation.

During the warranty period, we will, at our option, either repair or replace any product that proves to be defective.

To exercise this warranty, write or call your local Keithley Instruments representative, or contact Keithley Instruments headquarters in Cleveland, Ohio. You will be given prompt assistance and return instructions. Send the product, transportation prepaid, to the indicated service facility. Repairs will be made and the product returned, transportation prepaid. Repaired or replaced products are warranted for the balance of the original warranty period, or at least 90 days.

LIMITATION OF WARRANTY

This warranty does not apply to defects resulting from product modification without Keithley Instruments' express written consent, or misuse of any product or part. This warranty also does not apply to fuses, software, non-rechargeable batteries, damage from battery leakage, or problems arising from normal wear or failure to follow instructions.

THIS WARRANTY IS IN LIEU OF ALL OTHER WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR USE. THE REMEDIES PROVIDED HEREIN ARE BUYER'S SOLE AND EXCLUSIVE REMEDIES.

NEITHER KEITHLEY INSTRUMENTS, INC. NOR ANY OF ITS EMPLOYEES SHALL BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF ITS INSTRUMENTS AND SOFTWARE EVEN IF KEITHLEY INSTRUMENTS, INC., HAS BEEN ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH DAMAGES. SUCH EXCLUDED DAMAGES SHALL INCLUDE, BUT ARE NOT LIMITED TO: COSTS OF REMOVAL AND INSTALLATION, LOSSES SUSTAINED AS THE RESULT OF INJURY TO ANY PERSON, OR DAMAGE TO PROPERTY.

KEITHLEY

A GREATER MEASURE OF CONFIDENCE

Keithley Instruments, Inc.

Corporate Headquarters • 28775 Aurora Road • Cleveland, Ohio 44139
440-248-0400 • Fax: 440-248-6168 • 1-888-KEITHLEY (534-8453) • www.keithley.com

12/06

KPXI
High Speed Digitizer Module
User's Manual

Cleveland, Ohio, U.S.A.
Document Number: KPXI-AI-900-01 Rev. A / January 2007

Manual Print History

The print history shown below lists the printing dates of all Revisions and Addenda created for this manual. The Revision Level letter increases alphabetically as the manual undergoes subsequent updates. Addenda, which are released between Revisions, contain important change information that the user should incorporate immediately into the manual. Addenda are numbered sequentially. When a new Revision is created, all Addenda associated with the previous Revision of the manual are incorporated into the new Revision of the manual. Each new Revision includes a revised copy of this print history page.

Revision A (Document Number KPXI-AI-900-01) ..... January 2007

KEITHLEY

Safety Precautions

The following safety precautions should be observed before using this product and any associated instrumentation. Although some instruments and accessories would normally be used with non-hazardous voltages, there are situations where hazardous conditions may be present.

This product is intended for use by qualified personnel who recognize shock hazards and are familiar with the safety precautions required to avoid possible injury. Read and follow all installation, operation, and maintenance information carefully before using the product. Refer to the manual for complete product specifications.

If the product is used in a manner not specified, the protection provided by the product may be impaired.

The types of product users are:

Responsible body is the individual or group responsible for the use and maintenance of equipment, for ensuring that the equipment is operated within its specifications and operating limits, and for ensuring that operators are adequately trained.

Operators use the product for its intended function. They must be trained in electrical safety procedures and proper use of the instrument. They must be protected from electric shock and contact with hazardous live circuits.

Maintenance personnel perform routine procedures on the product to keep it operating properly, for example, setting the line voltage or replacing consumable materials. Maintenance procedures are described in the manual. The procedures explicitly state if the operator may perform them. Otherwise, they should be performed only by service personnel.

Service personnel are trained to work on live circuits, and perform safe installations and repairs of products. Only properly trained service personnel may perform installation and service procedures.

Keithley Instruments products are designed for use with electrical signals that are rated Measurement Category I and Measurement Category II, as described in the International Electrotechnical Commission (IEC) Standard IEC 60664. Most measurement, control, and data I/O signals are Measurement Category I and must not be directly connected to mains voltage or to voltage sources with high transient over-voltages. Measurement Category II connections require protection for high transient over-voltages often associated with local AC mains connections. Assume all measurement, control, and data I/O connections are for connection to Category I sources unless otherwise marked or described in the Manual.

Exercise extreme caution when a shock hazard is present. Lethal voltage may be present on cable connector jacks or test fixtures. The American National Standards Institute (ANSI) states that a shock hazard exists when voltage levels greater than 30V RMS, 42.4V peak, or 60VDC are present. A good safety practice is to expect that hazardous voltage is present in any unknown circuit before measuring.

Operators of this product must be protected from electric shock at all times. The responsible body must ensure that operators are prevented access and/or insulated from every connection point. In some cases, connections must be exposed to potential human contact. Product operators in these circumstances must be trained to protect themselves from the risk of electric shock. If the circuit is capable of operating at or above 1000 volts, no conductive part of the circuit may be exposed.

Do not connect switching cards directly to unlimited power circuits. They are intended to be used with impedance limited sources. NEVER connect switching cards directly to AC mains. When connecting sources to switching cards, install protective devices to limit fault current and voltage to the card.

Before operating an instrument, make sure the line cord is connected to a properly grounded power receptacle. Inspect the connecting cables, test leads, and jumpers for possible wear, cracks, or breaks before each use.

When installing equipment where access to the main power cord is restricted, such as rack mounting, a separate main input power disconnect device must be provided, in close proximity to the equipment and within easy reach of the operator.

For maximum safety, do not touch the product, test cables, or any other instruments while power is applied to the circuit under test. ALWAYS remove power from the entire test system and discharge any capacitors before: connecting or disconnecting cables or jumpers, installing or removing switching cards, or making internal changes, such as installing or removing jumpers.

Do not touch any object that could provide a current path to the common side of the circuit under test or power line (earth) ground. Always make measurements with dry hands while standing on a dry, insulated surface capable of withstanding the voltage being measured.

The instrument and accessories must be used in accordance with its specifications and operating instructions or the safety of the equipment may be impaired.

Do not exceed the maximum signal levels of the instruments and accessories, as defined in the specifications and operating information, and as shown on the instrument or test fixture panels, or switching card.

When fuses are used in a product, replace with same type and rating for continued protection against fire hazard.

Chassis connections must only be used as shield connections for measuring circuits, NOT as safety earth ground connections.

If you are using a test fixture, keep the lid closed while power is applied to the device under test. Safe operation requires the use of a lid interlock.

If a  screw is present, connect it to safety earth ground using the wire recommended in the user documentation.

The  symbol on an instrument indicates that the user should refer to the operating instructions located in the manual.

The  symbol on an instrument shows that it can source or measure 1000 volts or more, including the combined effect of normal and common mode voltages. Use standard safety precautions to avoid personal contact with these voltages.

The  symbol on an instrument shows that the surface may be hot. Avoid personal contact to prevent burns.

The  symbol indicates a connection terminal to the equipment frame.

The **WARNING** heading in a manual explains dangers that might result in personal injury or death. Always read the associated information very carefully before performing the indicated procedure.

The **CAUTION** heading in a manual explains hazards that could damage the instrument. Such damage may invalidate the warranty.

Instrumentation and accessories shall not be connected to humans.

Before performing any maintenance, disconnect the line cord and all test cables.

To maintain protection from electric shock and fire, replacement components in mains circuits, including the power transformer, test leads, and input jacks, must be purchased from Keithley Instruments. Standard fuses, with applicable national safety approvals, may be used if the rating and type are the same. Other components that are not safety related may be purchased from other suppliers as long as they are equivalent to the original component. (Note that selected parts should be purchased only through Keithley Instruments to maintain accuracy and functionality of the product.) If you are unsure about the applicability of a replacement component, call a Keithley Instruments office for information.

To clean an instrument, use a damp cloth or mild, water based cleaner. Clean the exterior of the instrument only. Do not apply cleaner directly to the instrument or allow liquids to enter or spill on the instrument. Products that consist of a circuit board with no case or chassis (e.g., data acquisition board for installation into a computer) should never require cleaning if handled according to instructions. If the board becomes contaminated and operation is affected, the board should be returned to the factory for proper cleaning/servicing.

Table of Contents

Appendix	Topic	Page

Appendix	Topic	Page

This page left blank intentionally.

List of Figures

Appendix	Figure	Title	Page

List of Tables

Appendix	Table	Title	Page

Section 1
Introduction

In this section:

Introduction

The Keithley Instruments Model KPXI-AI-2-65M is a 65MS/s, PXI digitizer with 14 bit resolution and a deep onboard memory buffer of 512 MB that features flexible input configurations, including programmable input ranges and user-selectable input impedance. With deep onboard acquisition memory, the Model KPXI-AI-2-65M is not limited by the PCI's 132MB/s bandwidth, and can record waveforms for extended periods of time. The Model KPXI-AI-2-65M is ideal for high-speed waveform capturing such as radar, ultrasound, software radio, and other signal digitizing applications needing large amounts of memory for data storage. Model KPXI-AI-2-65M uses the PXI trigger bus to send and receive trigger and timebase signals to and from other devices. In additional, the Model KPXI-AI-2-65M can phase-lock the reference clock from the PXI backplane or from an external connector for better synchronization capabilities.

This section gives an overview of the Model KPXI-AI-2-65M high speed digitizer.

Safety symbols and terms

The following symbols and terms may be found on the KPXI-Isolated DIO series module or used in this manual.

The  symbol indicates that the user should refer to the operating instructions located in the manual.

The  symbol shows that high voltage may be present on the terminal(s). Use standard safety precautions to avoid personal contact with these voltages.

The  symbol on an instrument shows that the surface may be hot. Avoid personal contact to prevent burns.

The **WARNING** heading used in this manual explains dangers that might result in personal injury or death. Always read the associated information very carefully before performing the indicated procedure.

The **CAUTION** heading used in this manual explains hazards that could damage the unit. Such damage may invalidate the warranty.

Applications

- Software Radio / Wireless Communication
- Radar / Sonar / Lidar
- Ultrasound
- Imaging
- Military
- Semiconductor device high speed signal verification
- Dynamic signal analysis

Features

- PXI specifications Rev. 2.1 compliant
- 3U Eurocard form factor, CompactPCI compliant (PICMG 2.0 R3.0)
- 14-bit A/D resolution
- Up to 60MS/s sampling rate per channel with internal timebase
- Up to 65MS/s sampling rate per channel with external timebase
- Up to 130MS/s sampling rate using alternating mode for single-channel acquisition
- 2-CH simultaneous-sampled single-ended analog inputs

- Programmable input ranges of ±1V and ±5V
- User-selectable (via solder point) input impedance of 50Ω or high input impedance
- >30MHz -3dB bandwidth
- 512MB on-board SODIMM SDRAM
- 2-CH synchronous digital inputs
- Scatter-gather DMA data transfers
- Analog and digital triggering
- Full auto-calibration
- Multiple modules synchronization through PXI trigger bus
- On-board phase-lock loop circuit

Specifications

Refer to the product data sheet for updated KPXI-AI-2-65M high-speed PXI digitizer specifications. Check the Keithley Instruments website at www.keithley.com for the latest updates to the specifications. See below for General Specifications.

General Specifications

Table 1-1
General Specifications

Dimensions	Single 3U PXI module, 100mm by 160mm
Operating temperature	0°C to 50ºC (Operating)
Storage temperature	-20ºC to 80ºC
Humidity	5% to 95% non-condensing
PCI signaling environment	Supports a 32-bit 3.3V or 5V PCI bus
I/O Connector	- BNC x 2 for analog inputs (CH0 & CH1) - SMB x 4 for external timebase (CLK IN), - external digital trigger (TRG IO) - synchronous digital input 0 & 1 (SDI0, SDI1)

Table 1-2
Typical power requirements

Power Rail	Current (mA)
5V	900
12V	305
3.3V	500

Unpacking and inspection

Inspection for damage

CAUTION **Your Model KPXI-AI-2-65M module contains electro-static sensitive components that can be easily be damaged by static electricity.**

Therefore, handle the card on a grounded anti-static mat. The operator should be wearing an anti-static wristband, grounded at the same point as the anti-static mat.

The Model KPXI-AI-2-65M high-speed PXI digitizer module was carefully inspected electrically and mechanically before shipment.

Inspect the module carton for obvious damages. Shipping and handling may damage the module. Make sure there are no shipping and handling damages on the module's carton before continuing.

After opening the module carton, extract the system module and place it only on a grounded anti-static surface with component side up. Save the original packing carton for possible future shipment.

Again, inspect the module for damages. Report any damage to the shipping agent immediately.

Shipment contents

The following items are included with every Model KPXI-AI-2-65M high-speed PXI digitizer module order:

- Model KPXI-AI-2-65M high-speed PXI digitizer module
- CD containing required software and manuals

Instruction manual

A CD-ROM containing this User's Manual and required software is included with each Model KPXI-AI-2-65M high-speed PXI digitizer module order. If a hardcopy of the User's Manual is required, you can order the Manual Package (Keithley Instruments Part Number KPXI-AI-900-01). The Manual Package includes an instruction manual and any pertinent addenda.

Always check the Keithley Instruments' website at www.keithley.com for the latest revision of the manual. The latest manual can be downloaded (in PDF format) from the website.

Repacking for shipment

Should it become necessary to return the Model KPXI General Purpose DIO series module for repair, carefully pack the unit in its original packing carton or the equivalent, and follow these instructions:

- Call Keithley Instruments' repair department at 1-888-KEITHLEY (1-888-534-8453) for a Return Material Authorization (RMA) number.
- Let the repair department know the warranty status of the Model KPXI-AI-2-65M high-speed PXI digitizer module.
- Write ATTENTION REPAIR DEPARTMENT and the RMA number on the shipping label.
- Complete and include the Service Form located at the back of this manual.

CAUTION **The boards must be protected from static discharge and physical shock. Never remove any of the socketed parts except at a static-free workstation. Use the anti-static bag shipped with the product to handle the board. Wear a grounded wrist strap when servicing.**

Supporting software

Keithley Instruments provides versatile software drivers for users' different approaches to building a system.

Programming library

For customers who are writing their own programs, we provide function libraries for many different operating systems, including:

KDIG-DRVR: includes device drivers and DLL's for Windows XP® and Windows 2000®. DLL's are binary compatible across Windows XP/2000. Therefore, all applications developed with KDIG-DRVR are compatible across Windows XP/2000. The developing environment can be VB, VC++, BC5, or any Windows programming language that allows calls to a DLL. Documentation includes a User's Guide (refer to Appendix A: KDIG-DRVR User's Guide), and a Function Reference (refer to Appendix B: KDIG-DRVR Function Reference).

KI-DAQ LabVIEW driver

KI-DAQ contains the VI's, which are used to interface with National Instrument's Lab-VIEW[1] software package. The KI-DAQ driver supports Windows XP/2000. The LabVIEW driver is shipped free with the board. Documentation includes an Interface Guide (refer to Appendix C: KIDAQ®-LabVIEW Compatible Interface Guide), and an interface Function Reference (refer to Appendix D: KIDAQ®-LabVIEW Compatible Function Reference).

The above software drivers are shipped with the board.

1. National Instruments™, NI, and LabVIEW are trademarks of the National Instruments Corporation.

This page left blank intentionally.

Section 2
Installation

In this section:

Introduction

This section contains information about handling and installing Keithley Instruments' KIDAQ® KPXI series cards:

- Handling precautions
- PCI configuration
- Installation

Handling precautions

CAUTION **Use care when handling the KIDAQ® KPXI series cards. KIDAQ® KPXI series cards contain electro-static sensitive components that can be easily damaged by static electricity.**

When handling, make sure to observe the following guidelines:

- Only handle the card on a grounded anti-static mat.
- Wear an anti-static wristband that is grounded at the same point as the anti-static mat.

PCI configuration

Plug-and-play

The Interrupt and I/O port address are the variables associated with automatic configuration; the resource allocation is managed by the system BIOS. Upon system power-on, the internal configuration registers on the board interact with the BIOS. As a plug-and-play component, the board requests an interrupt number via its PCI controller. The system BIOS responds with an interrupt assignment based on the board information and system parameters. These system parameters are determined by the installed drivers and the hardware load recognized by the system. If this is the first time a KIDAQ® KPXI series card will be installed on your Windows® system, a hardware driver needs to be installed. Refer to Installation for detailed information.

Configuration

Configuration is done on a board-by-board basis for all PXI boards on your system. Configuration is controlled by the system and software. There is no jumper setting required (or available) for base address, DMA, and interrupt IRQ.

The configuration is not static, but is subject to change with every boot of the system as new boards are added or removed.

Troubleshooting

If your system doesn't boot or if you experience erratic operation with your PXI board in place, it's likely caused by an interrupt conflict (perhaps the BIOS Setup is incorrectly configured). In general, the solution, is to consult the BIOS documentation that comes with your system.

Installation

Windows will find the new Model KPXI-AI-2-65M module automatically. If this is the first time the Model KPXI-AI-2-65M is running on your Windows system, please refer to the following step-by-step installation procedures.

Step 1. Install driver software

Windows® will find the new module automatically. If this is the first time a KIDAQ® KPXI series digitizer has been installed, a hardware driver needs to be installed. Use the following installation procedure as a guide.

NOTE *Keithley Instruments controllers are pre-loaded with the necessary drivers.*

For Windows XP/2000:

1. Insert the CD shipped with the module. The CD should auto load. From the base menu install the KDIG-DRVR. This is the hardware driver that recognizes the KPXI series digitizer modules. If the CD does not auto load run, then under **x**:\KDIG-DRVR\DISK1\, you will find SETUP.EXE (**x** is the drive letter of your CDROM). This will also run the install.
2. When you complete driver installation, turn off the system.

Step 2. Inspect the module

Keeping the "Handling precautions" information in mind, inspect the module for damage. With the module placed on a firm flat surface, press down on all socketed IC's to make sure that they are properly seated.

If the module does not pass the inspection, do not proceed with the installation.

CAUTION **Do not apply power to the card if it has been damaged.**

Step 3. Check module's analog input impedance setting

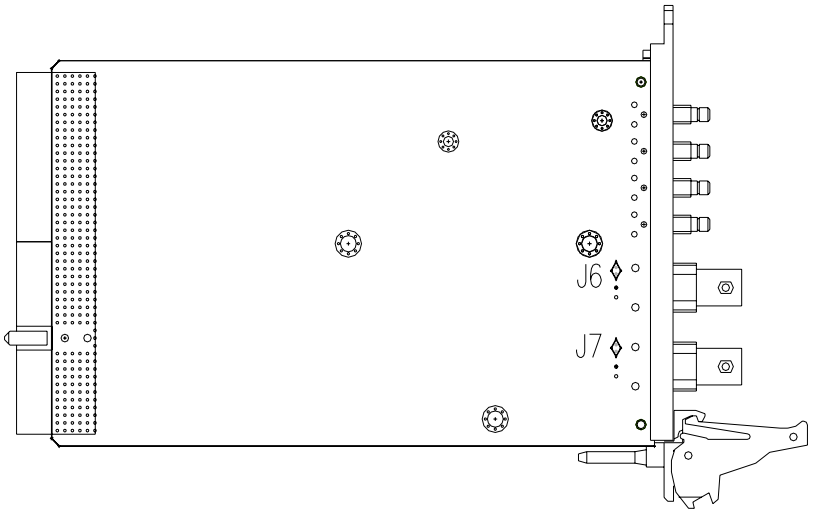
CH0 and CH1 input impedance can be set to 50Ω or 1.5MΩ by soldering gap switches J6 and J7 on the backside of the Model KPXI-AI-2-65M. The location of J6, J7 and the corresponding input impedance settings are shown in Table 2-1 and Figure 2-1. The default setting is 50Ω input impedance.

Table 2-1
Location of solder switches

J6	CH0 Input Impedance
Open	High (1.5MΩ)
Close (Default)	Low (50Ω)

J7	CH1 Input Impedance
Open	High (1.5MΩ)
Close (Default)	Low (50Ω)

Figure 2-1
Location of solder switches



NOTE *If the high input impedance 1.5MΩ is selected, the output impedance of the signal sources should be kept low to avoid the offset voltage caused by the input bias current, which is 2μA min. and 25μA max.*

The card is now ready for installation.

Step 4. Install module

Remove power from the system and install the KPXI card in an available slot.

The PXI connectors are rigid and require careful handling when inserted and removed. Improper handling of modules can easily damage the backplane.

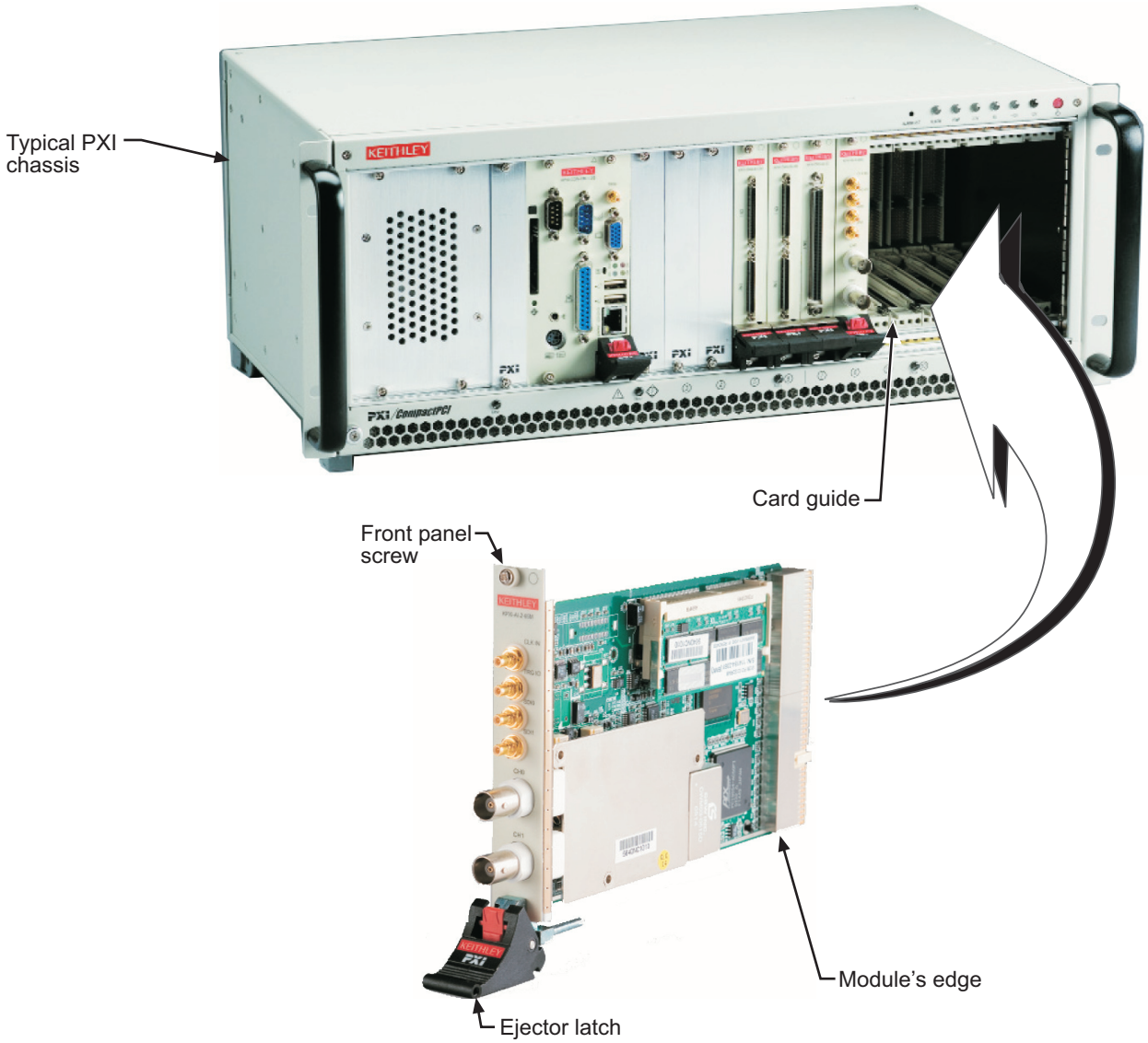
To insert the module into a PXI chassis, use the following procedure as a guide:

1. Turn off the system.
2. Align the module's edge with the card guide in the PXI chassis.
3. Slide the module into the chassis until resistance is felt from the PXI connector.
4. Push the ejector upwards and fully insert the module into the chassis. Once inserted, a "click" can be heard from the ejector latch.
5. Tighten the screw on the front panel.
6. Turn on the system.

To remove a module from a PXI chassis, use the following procedure as a guide:

1. Turn off the system.
2. Loosen the screw on the front panel.
3. Push the ejector downwards and carefully remove the module from the chassis.

Figure 2-2
Typical PXI module installation



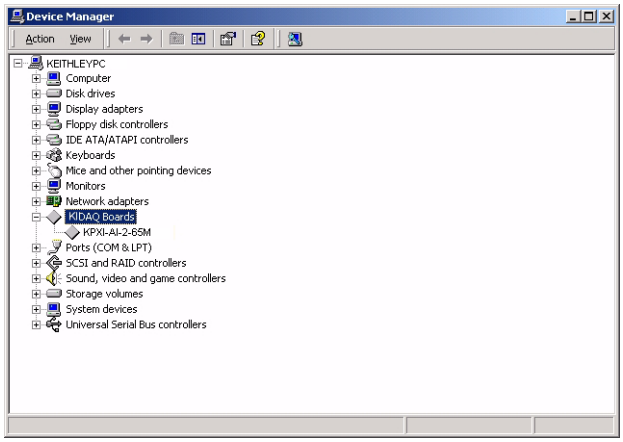
Step 5. Verify installation

When the system is turned on for the first time with a new module present (or a module in a new slot), Windows **Add New Hardware Wizard** attempts to locate the correct driver. If it cannot find the correct driver, even after you have loaded the driver above in Step 1, then force the **Add New Hardware Wizard** to look in Windows system32 directory. The driver files should be in this location. If they are not, shutdown the system, remove the module, and restart the installation process.

When the **Add New Hardware Wizard** finishes, the window will verify whether or not installation was successful. To confirm if the module is installed correctly at a later time, use **Windows Device Manager**. In the **Device Manager** under KIDAQ Boards, look for a device name matching the model number of the newly installed board (see Figure 2-3 for an example). If it is found, installation is complete. If the board appears with a exclamation point or warning in Device Manager, the installation was unsuccessful. If unsuccessful, use **Device Manager** to update the

driver or un-install the module, power down the system, remove the module, and attempt installation again from Step 1.

Figure 2-3
Device manager

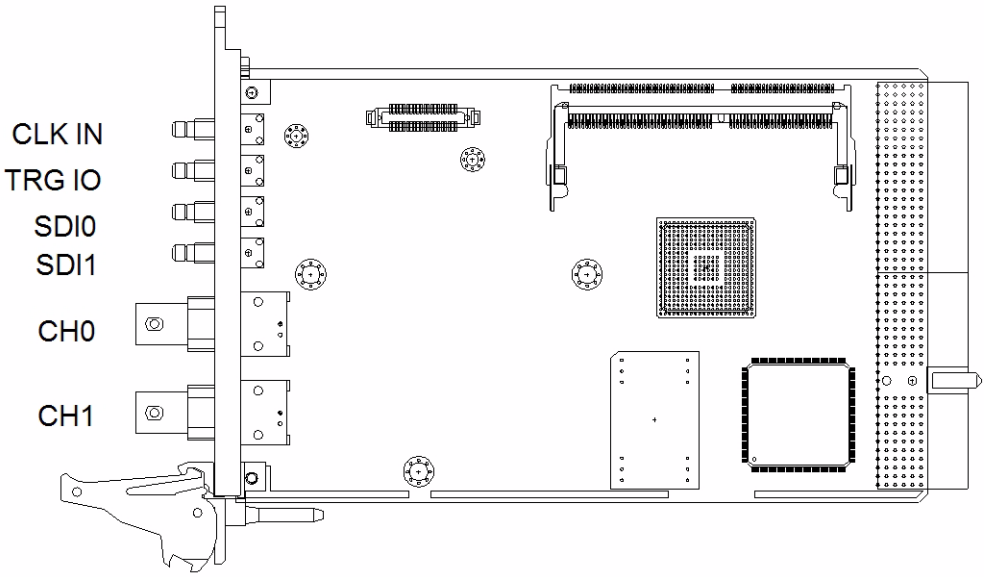


Section 3
Connection and Operation

In this section:

Connection

Figure 3-1
Model KPXI-AI-2-65M connectors



The Keithley Instruments Model KPXI-AI-2-65M module is packaged in a Eurocard form factor compliant with PXI Specifications measuring 160mm in length and 100mm in height (not including connectors). The connector types and functions are described as follows.

CLK IN: The SMB connector is a 50Ω, AC-coupled external timebase input or external 10MHz reference clock input.

TRG IO: The SMB connector is for external digital trigger input or output.

SDI0: The SMB connector is for synchronous digital input channel 0.

SDI1: The SMB connector is for synchronous digital input channel 1.

CH0: The BNC connector is for attaching the analog input signal to measure on channel 0.

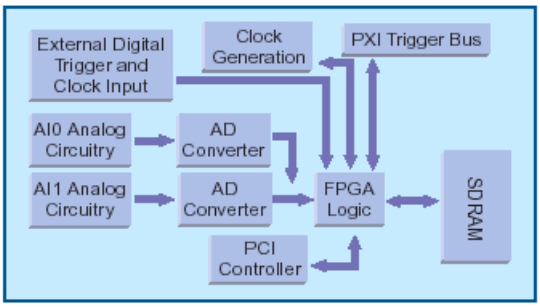
CH1: The BNC connector is for attaching the analog input signal to measure on channel 1.

Overview

Functional block diagram

The following topics overview the Model KPXI-AI-2-65M main features as shown in the functional block diagram below.

Figure 3-2
Model KPXI-AI-2-65M functional block diagram



Analog input

The Model KPXI-AI-2-65M features two analog input channels. Signal bandwidth of each channel exceeds 30MHz, satisfying the Nyquist sampling theory. The input ranges are programmable as either ±5V or ±1V. The 14-bit A/D resolution makes the Model KPXI-AI-2-65M ideal both for time-domain and frequency-domain applications.

Acquisition system

The Model KPXI-AI-2-65M uses a pair of 65MS/s, 14-bit pipeline ADCs to digitize input signals. It provides an internal 60MHz timebase for data acquisition. The maximum real-time sampling rate is 60MS/s with internal timebase and up to 65MS/s with external timebase. By using **Alternating mode** (operating as a single channel), the sampling rate is up to 120MS/s with internal timebase or 130MS/s with external timebase.

Acquisition memory

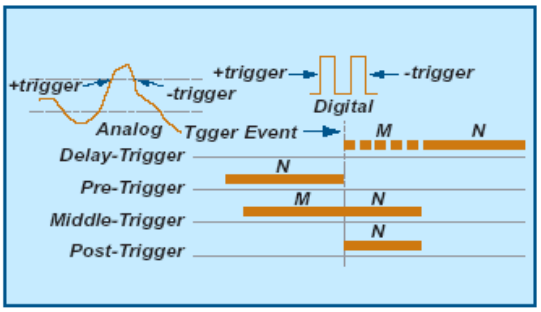
The Model KPXI-AI-2-65M contains 512MB of SODIMM SDRAM. Digitized data is stored in the on-board SDRAM before being transferred to host memory. The Model KPXI-AI-2-65M uses scatter-gather bus mastering DMA to move data to the host memory. If the data throughput from the Model KPXI-AI-2-65M is less than the available PCI bandwidth, the Model KPXI-AI-2-65M will bypass the SDRAM and use the on-board 3k-sample FIFO to achieve real-time transfer directly to the host memory.

Triggering

The Model KPXI-AI-2-65M features flexible triggering functions, such as analog and digital triggering. The analog trigger features programmable trigger thresholds on rising or falling edges of both input channels. The 5V/TTL digital trigger comes from the external SMB connector, PXI trigger bus or PXI_STAR for synchronizing multiple devices.

Post-trigger, pre-trigger, delay-trigger and middle-trigger modes are available to acquire data around the trigger event. The Model KPXI-AI-2-65M also features repeated trigger acquisition to acquire data in multiple segments coming with successive trigger events at extremely short rearming intervals.

Figure 3-3
Model KPXI-AI-2-65M trigger modes

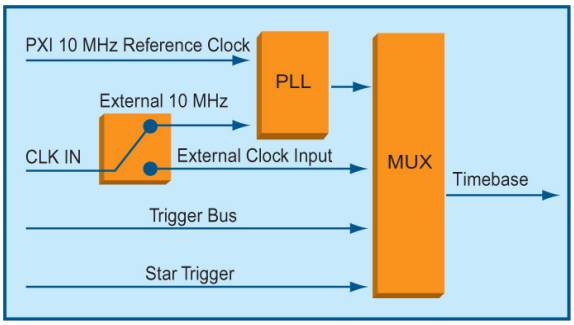


Multiple-instrument synchronization

To achieve multiple-instrument synchronization, Model KPXI-AI-2-65M routes timing and trigger signals between one or more Model KPXI-AI-2-65M devices by the PXI trigger bus. Users can assign timebase or trigger signals to a specific PXI trigger bus[0:7] via software, providing better flexibility for multiple devices synchronization. The Model KPXI-AI-2-65M also adopts PXI_STAR signal for better synchronization performance. Both trigger signals and timebase clock can be transferred to 13 peripheral slots through the PXI_STAR signal from a star trigger controller in slot 2 of the PXI chassis.

The reference clock is used in the Model KPXI-AI-2-65M phase-lock loop (PLL) circuit to synchronize the timebase clock to the reference. The Model KPXI-AI-2-65M can accept a reference clock the CLK IN on the front panel as well as from PXI_CLK10. The reference clock should be a fixed and stable 10MHz clock.

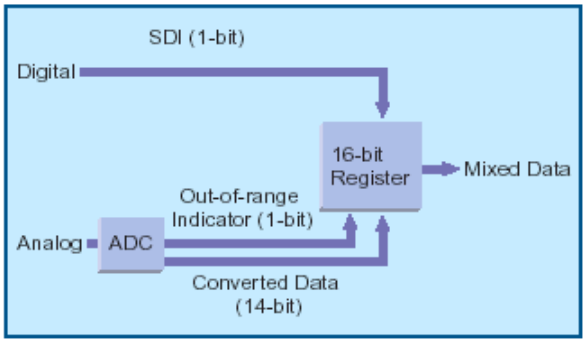
Figure 3-4
Model KPXI-AI-2-65M timebase control



Synchronous digital input

The Model KPXI-AI-2-65M features two high-speed digital input lines. User can clock in digital data synchronous to the analog input timebase clock. Viewing analog and digital correlated operations is effortless.

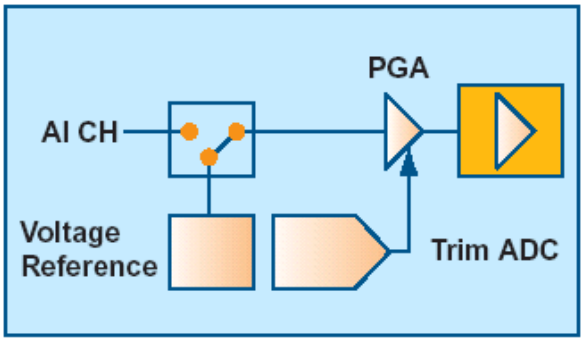
Figure 3-5
Model KPXI-AI-2-65M synchronous digital inputs



Auto calibration

The auto-calibration function of the Model KPXI-AI-2-65M is performed with trim DACs to calibrate offset and gain errors of the analog input channels. Once the calibration process is complete, the calibration constant will be stored in EEPROM. These values are loaded and used as needed by the board. Because all the calibration is conducted automatically by software commands, users do not have to adjust trimpots to calibrate the boards.

Figure 3-6
Model KPXI-AI-2-65M auto calibration

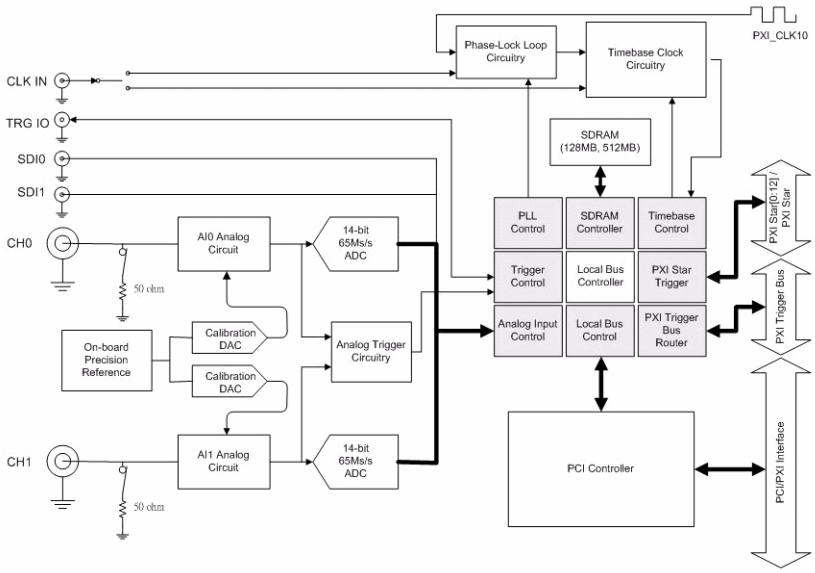


Operation

The operation of the Model KPXI-AI-2-65M is described in this section, including the control and setting of signal sources, timebase sources, trigger sources, trigger modes, data transfers, synchronizing multiple modules, and auto-calibration.

Detailed block diagram

Figure 3-7
Model KPXI-AI-2-65M block diagram



Analog input signal source control

Number of channels

The Model KPXI-AI-2-65M provides two simultaneously sampled analog input channels in SE (single ended) connection. Each channel can be enabled individually.

Signal range and input impedance

The available signal input ranges are ±5V or ±1V, which can be set by software. All signals are DC-coupled. The input impedance for high-speed applications should also be considered. The selectable input impedance values are 50Ω and 1.5MΩ. Please refer to the paragraph on Configuration, specifically for information on setting impedance.

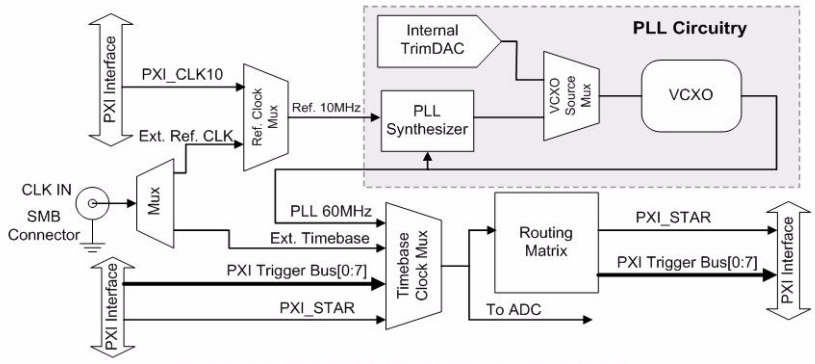
A/D Sampling rate and timebase control

The Model KPXI-AI-2-65M supports six timebase sources for analog input conversion:

- Internal 60MHz (Internal VCXO free run mode)
- Internal 60MHz (timebase clock locked to PXI_CLK10)
- Internal 60MHz (timebase clock locked to external reference 10MHz)
- External sine wave
- PXI Trigger Bus[0:7]
- PXI_STAR

The following diagram shows the timebase clock architecture of the Model KPXI-AI-2-65M.

Figure 3-8
Model KPXI-AI-2-65M timebase clock architecture



Once you have chosen the timebase source, you can set a 24-bit counter to divide the timebase to get the needed sampling rate. The following formula determines the ADC sampling frequency:

Sampling Rate = Timebase Frequency / ADC Clock Divisor

where the ADC Clock Divisor = 1,2,3,4,5… 2^{24}-1(maximum)

Internal timebase clock source

The Model KPXI-AI-2-65M has an onboard Voltage Control Crystal Oscillator (VCXO) running at 60MHz. When selects internal timebase clock, you can have free-run mode (none phase-locked) or PLL mode. In free-run mode, the 60MHz timebase clock is coming from the VCXO which is controlled by an internal trim DAC. In PLL mode, the onboard 60MHz timebase phase locked to the reference 10MHz clock. The reference 10MHz clock comes from PXI backplane (PXI_CLK10) or front panel SMB connector (CLK IN).

External timebase clock source

When users need a specific timebase in some applications that the onboard 60MHz timebase cannot achieve, a timebase clock from an external device can replace the onboard timebase clock. In addition, external timebases also provide a method to synchronize the Model KPXI-AI-2-65M to other measurement modules by distributing/receiving a common clock to/from multiple modules. The Model KPXI-AI-2-65M can receive an external timebase clock from the front panel connector (CLK IN), PXI Star trigger or one of the PXI Trigger bus lines.

Users can supply the timebase from the external SMB connector **CLK IN**, which should be a sine wave signal. This signal is AC coupled with 50Ω input impedance and the valid input level is from 1 to 2 volts peak-to-peak. Note that the external clock must be continuous for correct ADC operation because of the pipeline architecture of the ADC.

Model KPXI-AI-2-65M can receive the timebase clock via one of the trigger bus lines by software selection. The eight PXI trigger bus lines (PXI_TRIG[0:7]) provide intermodule synchronization and communication.

When Model KPXI-AI-2-65M is plugged into a generic peripheral slot, it can receive timebase clock from PXI_STAR. The PXI_STAR signal comes from the star trigger controller and is matched in propagation delay within 1ns and the delay from star trigger slot to a peripheral slot is less than 5ns. According to these hardware features, Model KPXI-AI-2-65M can achieve very good synchronization performance when using PXI_STAR as the timebase clock source.

PLL reference clock source

The Model KPXI-AI-2-65M is equipped with a phase-locked loop circuit to synchronize the timebase clock to reference clock. The Model KPXI-AI-2-65M can receive a reference clock from

the front panel connector (CLK IN) as well as from PXI_CLK10. The reference clock from front panel connector should be a stable 10MHz clock. The input of CLK IN is AC coupled with 50Ω input impedance and the valid input level is from 1 to 2 volts peak-to -peak. Note that the reference clock must be continuous for correct PLL operation. Once the reference clock is disconnected, the timebase clock will still exist but not be phase locked to the reference clock anymore. When using the external timebase clock, it's not valid to lock to the reference clock.

Timebase clock exporting

The Model KPXI-AI-2-65M can export timebase clock to the following two destinations: PXI_STAR or one of PXI Trigger bus lines. When Model KPXI-AI-2-65M plugged into star trigger controller slot of the PXI backplane, it can export timebase clock to the other 13 generic peripheral slots. The Model KPXI-AI-2-65M can also export timebase clock onto one of the eight PXI trigger bus lines. By software programming, user can pick up a trigger line to transmit timebase clock. These two features are very useful when synchronizing multiple measurement modules.

Timebase 50% duty cycle restore function

Because the on-board sample-and-hold circuit is in hold mode when the timebase is high and in track mode when timebase is low, the Model KPXI-AI-2-65M can get the best dynamic response performance for high-frequency analog inputs when the timebase has a 50% duty cycle. However, it is often difficult to maintain a 50% duty cycle; especially when driving the timebase with a single-ended or sine wave input. To ease the constraint of providing an accurate 50% timebase, the Model KPXI-AI-2-65M has an optional duty cycle restore function to restore the timebase duty cycle to 50%, independent of the clock input duty cycle. Low jitter on the rising edge (sampling edge) of the timebase is preserved while the falling edge is interpolated.

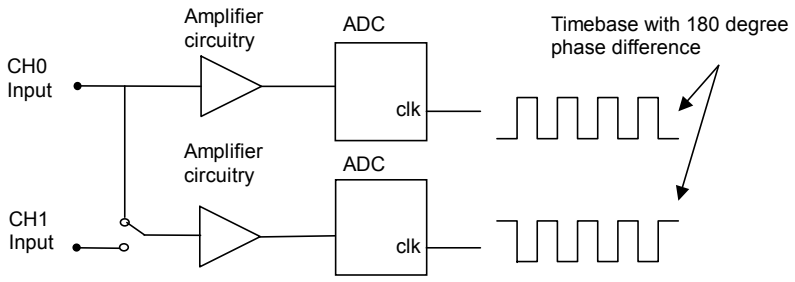
It may be desirable to disable the duty cycle restore function when users use the external timebase input whose frequency is varied. Once the timebase frequency is changed, over 100 clock cycles may be required for the duty cycle restore circuit to settle to the new speed. Duty cycle restore function is software programmable when configuring your Model KPXI-AI-2-65M.

130MS/s Sampling using alternating mode

The Model KPXI-AI-2-65M uses two A/D converters, each running at 60MS/s, to provide a dual-channel simultaneous real-time sampling rate of 60MS/s. (65MS/s with external timebase)

For the single-channel acquisition, the two ADCs can be clocked in a alternating mode to achieve up to 120MS/s sampling (130MS/s with external timebase). Figure 3-9 illustrates the operation. Note that only CH0 can be used in alternating mode. The onboard auto-calibration circuitry allows the two channels to be matched in order to reduce the image signal.

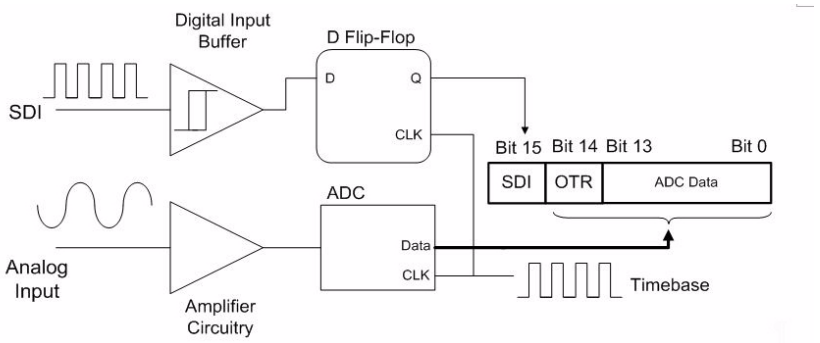
Figure 3-9
Alternating mode operations



Synchronous digital input

The Model KPXI-AI-2-65M contains two synchronous digital input channels, SDI0 and SDI1. These two digital input lines can be sampled synchronously with timebase clock for mixed signal applications. Thus the data transfer can up to 60Mbit/s when using internal 60MHz timebase clock. These two digital input lines are combined with analog input channels, respectively. Please refer to Figure 3-10 for the data format of ADC and SDI. When the analog input channel of Model KPXI-AI-2-65M is configured in alternating mode, only SDI0 is enabled.

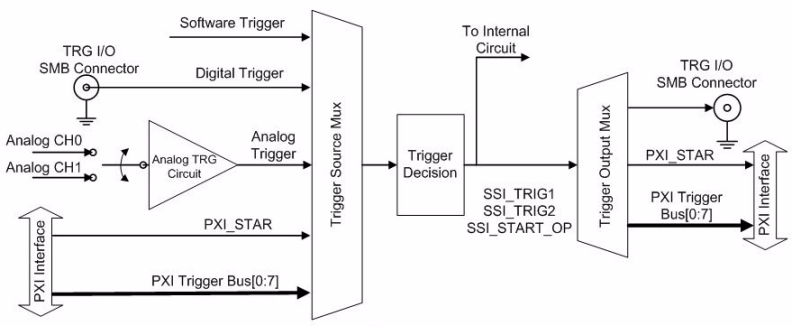
Figure 3-10
Synchronous digital input operations



Trigger sources

In addition to the internal software trigger, the Model KPXI-AI-2-65M also supports external analog triggers, external digital triggers, PXI_STAR triggers and PXI Trigger Bus[0:7]. Users can configure the trigger source by software. Refer to Figure 3-11 for Model KPXI-AI-2-65M trigger architecture.

Figure 3-11
Model KPXI-AI-2-65M trigger architecture



Software trigger

This trigger mode does not need any external trigger source. The trigger asserts right after executing specified function calls to begin the operation.

External analog trigger

Users can choose either CH0 or CH1 as the trigger signal while using external analog trigger source. The trigger level can be set by software with 8-bit resolution. Please refer to Table 3-1 for the ideal transfer characteristic.

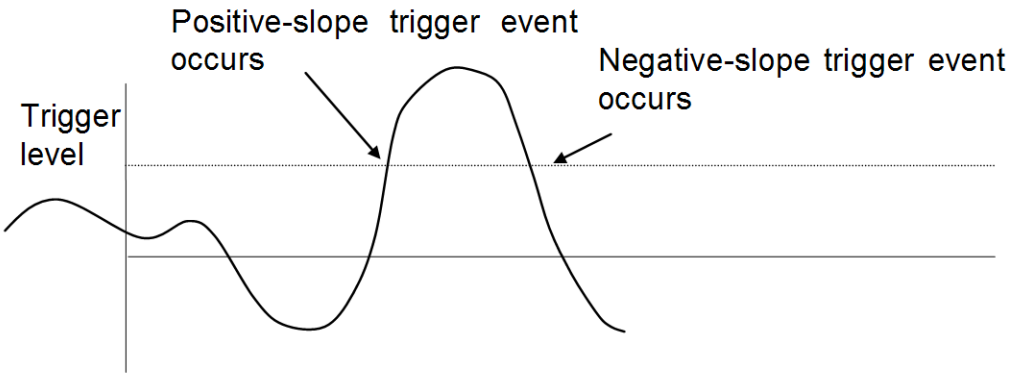
Table 3-1
Analog trigger ideal transfer characteristic

Trigger Level digital setting	Trigger voltage (±5V range)	Trigger voltage (±1V range)
0xFF	4.96V	0.992V
0xFE	4.92V	0.984V
---	---	---
0x81	0.04V	0.008V
0x80	0	0
0x7F	-0.04V	-0.008V
---	---	---
0x01	-4.96V	-0.992V

Table 3-1 contains the trigger conditions for analog triggers, which are illustrated in Figure 3-12 and described as follows:

– Positive-slope trigger - The trigger event occurs when the trigger signal (analog input signal) changes from a voltage that is lower than the specified trigger level to a voltage that is higher than the specified trigger level.
– Negative-slope trigger - The trigger event occurs when the trigger signal (analog input signal) changes from a voltage that is higher then the specified trigger level to a voltage that is lower than the specified trigger level.

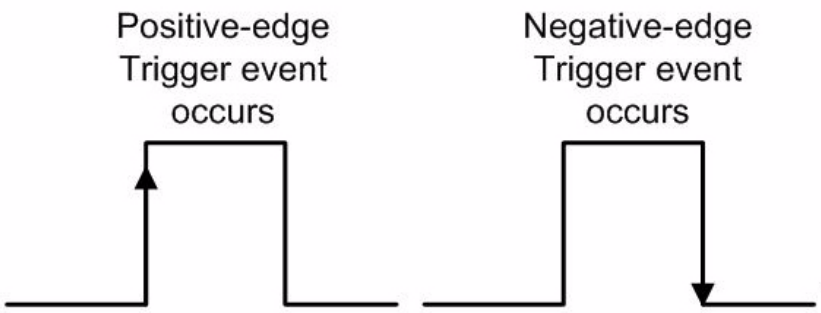
Figure 3-12
Analog trigger conditions



External digital trigger

An external digital trigger occurs when a TTL rising edge or a falling edge is detected at the SMB connector **TRG IO** on the front panel, as illustrated in Figure 3-13. The trigger polarity can be selected by software. Note that the signal level of the external digital trigger signal should be TTL-compatible, and the minimum pulse width is 10ns.

Figure 3-13
External digital trigger input



PXI_STAR Trigger

When users select PXI_STAR as trigger source, the Model KPXI-AI-2-65M can accept a TTL-compatible digital signal as a trigger signal. The trigger occurs when a rising edge or falling edge is detected at PXI_STAR. User can use software to configure the trigger polarity. The minimum pulse width requirement of this digital trigger signal is 10ns.

Triggers from PXI Trigger Bus[0:7]

The Model KPXI-AI-2-65M utilizes PXI Trigger Bus[0:7] as System Synchronization Interface (SSI). Using the interconnected bus provided by PXI Trigger Bus, user can easily synchronize multiple modules. When configured as input, the Model KPXI-AI-2-65M can accept three different SSI signals, SSI_TRIG1, SSI_TRIG2 and SSI_START_OP (for more detail about these signals, please refer to Synchronizing multiple devices). Each signal can be routed from one of the PXI Trigger Bus[0:7] by software programming.

Trigger modes

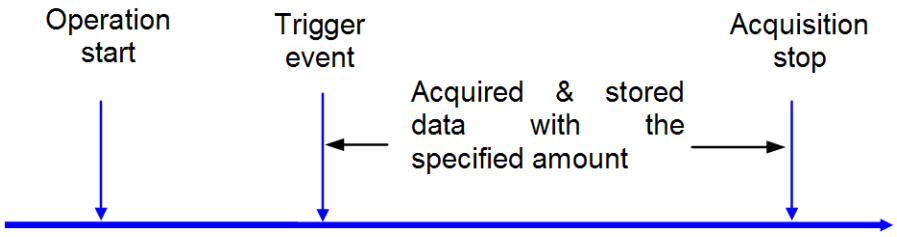
The Model KPXI-AI-2-65M provides 5 trigger sources (internal software trigger, external analog trigger, external digital trigger, PXI_Star trigger, and PXI Trigger Bus[0:7] signals). Users must select one of them as the source of the trigger event. A trigger event occurs when the specified condition is detected on the selected trigger source (For example, a rising edge on the external digital trigger input).

There are 4 trigger modes (pre-trigger, post-trigger, middle-trigger, and delay-trigger) working with the 5 trigger sources to initiate different data acquisition timing when a trigger event occurs. They are described as follows.

Post-trigger acquisition

Use post-trigger acquisition when you want to collect data after the trigger event, as illustrated in Figure 3-14.

Figure 3-14
Post-trigger acquisition

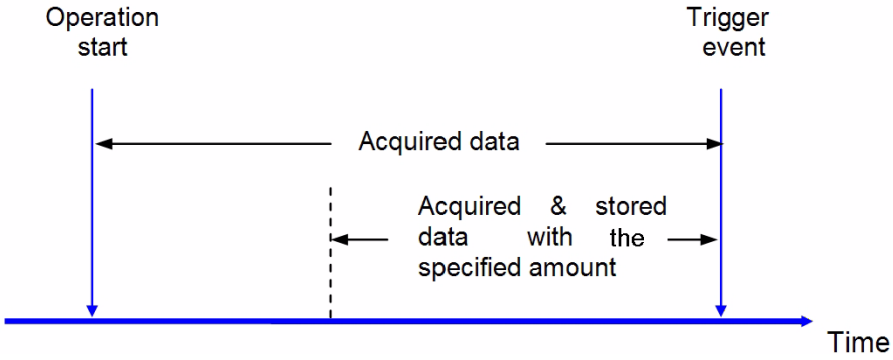


Pre-trigger acquisition

Use pre-trigger acquisition to collect data before the trigger event. The acquisition starts once specified function calls are executed to begin the pre-trigger operation, and it stops when the trigger event occurs.

If the trigger event occurs after the specified amount of data has been acquired, the system only stores the data before the trigger event with the specified amount, as illustrated in Figure 3-15.

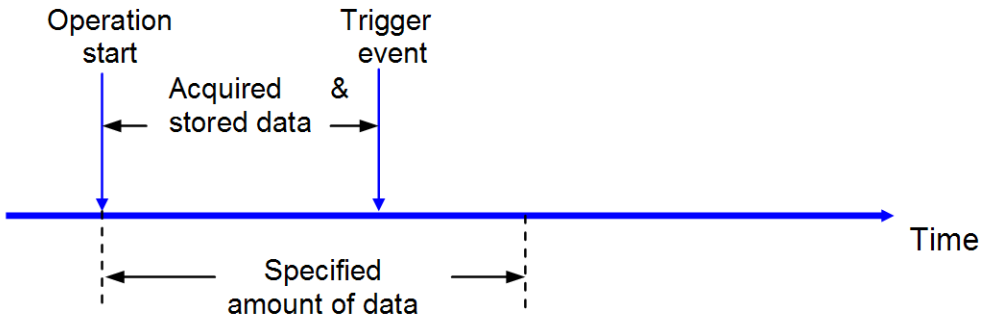
Figure 3-15
Pre-trigger data acquisition



* The trigger event occurs after specified amount of data has been acquired.

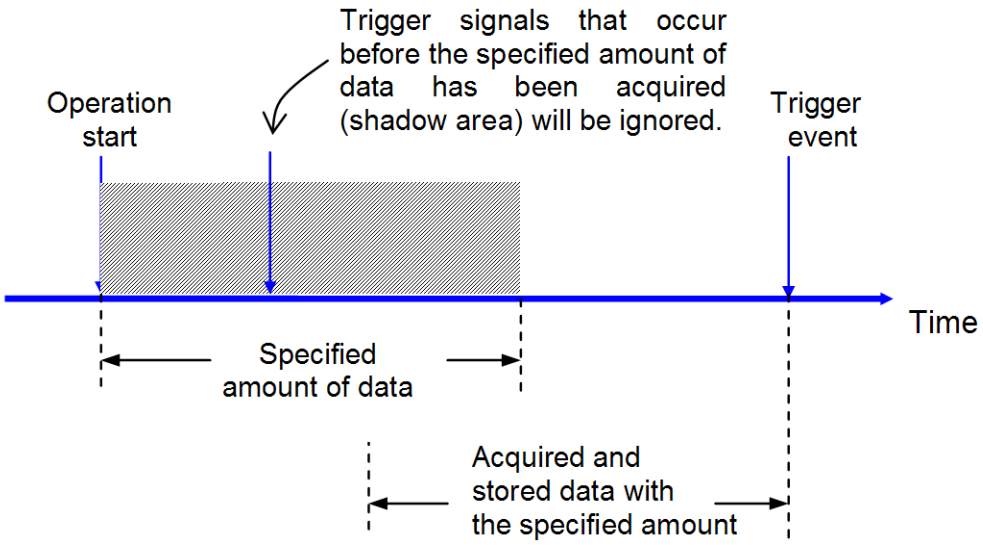
However, if the trigger event occurs before the specified amount of data has been acquired, the system can either stop the acquisition immediately (which implies the stored data will be less then the amount you specified) or ignore the trigger signal until the specified amount of data has been acquired (which assures the user can get the specified amount of data). These can be set by software and are illustrated in Figure 3-16 and Figure 3-17.

Figure 3-16
Pre-trigger with simultaneous data acquisition**



** The trigger signal is accepted anytime after operation starts.

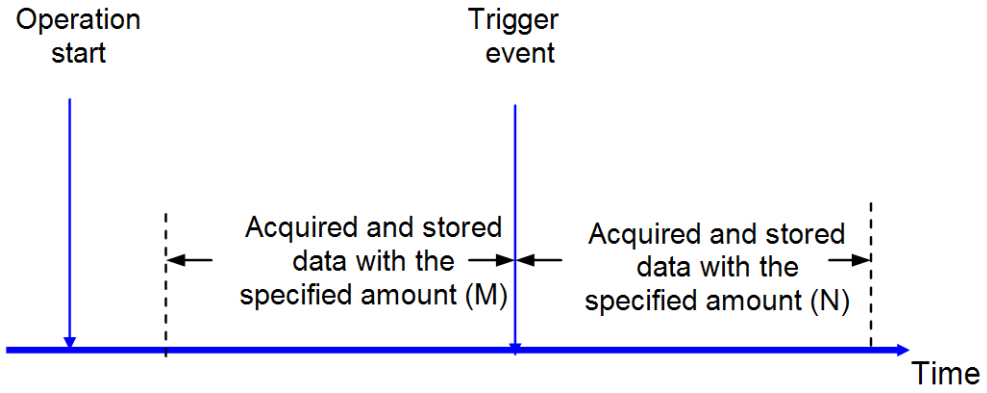
Figure 3-17
Pre-trigger signal ignored until specified data amount acquired



Middle-trigger acquisition

Use middle-trigger acquisition when you want to collect data before and after the trigger event. The amount of stored data before and after the trigger can be set individually (M and N), as illustrated in Figure 3-18.

Figure 3-18
Middle-trigger acquisition

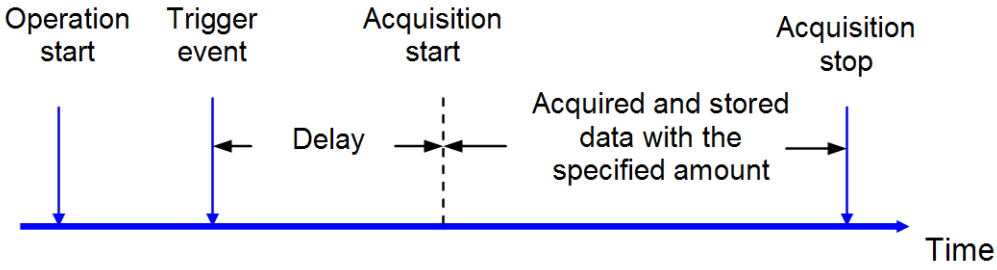


Like pre-trigger mode, the stored data may be less than the amount specified if the trigger event occurs before the specified amount of data (M) has been acquired. Users can also set by program to ignore trigger signals until the specified amount of data (M) has been acquired.

Delay-trigger acquisition

Use delay trigger acquisition to delay the data collection after the trigger event, as illustrated in Figure 3-19. The delay time is specified by a 32-bit counter value so that the maximum delay time is the period of timebase * (2^{32} –1), while the minimum delay time is the period of timebase.

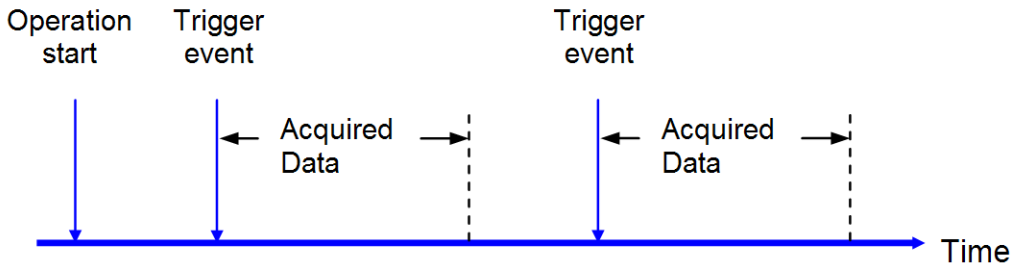
Figure 3-19
Delay-trigger acquisition



Post-trigger or delay-trigger acquisition with re-trigger

Use post-trigger or delay-trigger acquisition with re-trigger function to collect data after several trigger events, as illustrated in Figure 3-20. Users can program the number of triggers then the Model KPXI-AI-2-65M will acquire an additional record each time a trigger is accepted until all the requested records have been stored in memory. After the initial setup, the process does not require software intervention.

Figure 3-20
Post-trigger with re-trigger acquisition

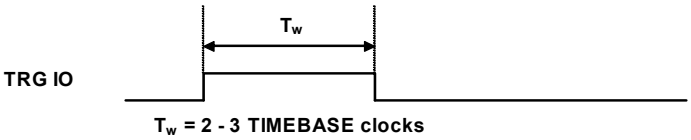


Exporting trigger signals

The Model KPXI-AI-2-65M can export trigger signals to the following connectors: TRG IO, PXI_STAR or PXI Trigger Bus[0:7].

The **TRG IO** on the front panel can also be programmed to output the trigger signal when the trigger source is from software trigger, analog trigger, PXI_STAR trigger, or PXI Trigger Bus[0:7]. The timing characteristic is in Figure 3-21.

Figure 3-21
TRG IO output signal timing



When the Model KPXI-AI-2-65M is plugged into the star trigger controller slot, i.e. slot 2, in a PXI system, it can export trigger signals to other 13 generic peripheral slots (PXI_STAR0 ~ PXI_STAR12) through the PXI star trigger pin. The trigger signal source can be from software trigger, external digital trigger signal from TRG IO, analog trigger and PXI Trigger Bus[0:7]. The output trigger signal timing characteristic is identical to the TRG IO which is shown in Figure 3-21.

The Model KPXI-AI-2-65M utilizes PXI Trigger Bus[0:7] as System Synchronize Interface. When configured as output, the Model KPXI-AI-2-65M can output 3 different trigger signals, SSI_TRIG1, SSI_TRIG2 and SSI_START_OP. User can route these signals to any one of PXI Trigger Bus[0:7] by software programming.

Data transfers

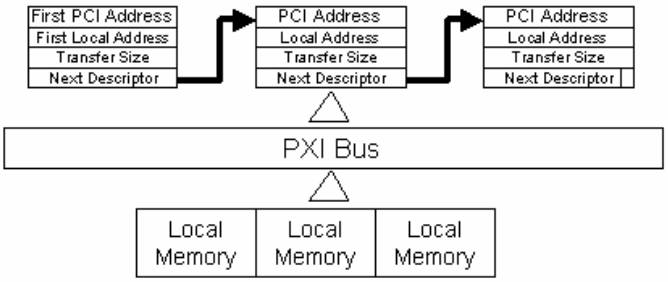
Since the maximum data throughput on the Model KPXI-AI-2-65M (60MS/s * 2 channels * 2 Bytes/channel = 240MB/s) is much higher than the 32bit/33MHz PCI-bus bandwidth, samples are acquired into the onboard SDRAM memory before being transferred to the host computer. Since the number of stored samples per acquisition is limited by the amount of on-board memory, the Model KPXI-AI-2-65M contains 512MB of on-board memory in order to meet application requirements.

Once all the data has been stored in the on-board memory, the data will be transferred to the host computer's memory through bus-mastering DMA.

In a multi-user or multi-tasking OS, like Microsoft Windows®, it is difficult to allocate a large continuous memory block to do the DMA transfer. Therefore, the Model KPXI-AI-2-65M provides the function of scatter /gather DMA to link the non-continuous memory blocks into a linked list so

that users can transfer very large amounts of data without being limited by the fragment of small size memory, as illustrated in Figure 3-22.

Figure 3-22
Scatter/gather DMA for data transfer



If the data throughput from the Model KPXI-AI-2-65M is less than the available PCI bandwidth (For example: 20MS/s * 2 channels * 2 Bytes/channel = 80MB/s), the Model KPXI-AI-2-65M also features on-board 3k-sample FIFO to achieve real-time transfer bypassing the SDRAM, directly to host memory.

AI data format

Table 3-2 illustrates the ideal transfer characteristics of various input ranges of the Model KPXI-AI-2-65M. Bit13-0 is the acquired 14-bit A/D data with binary coding format while bit14 is the out-of-range (OTR) indicator (logic "1" means out-of-range). As described in Synchronous digital input earlier in this section, bit 15 is the synchronous digital input data. SDI0 is within the data format of analog channel 0 while SDI1 is within the data format of analog channel 1.

Table 3-2
Analog input voltage and the output digital code

Description	Analog Input Voltage		Digital code
Full-scale Range	±5V	±1V	
Least significant bit	0.61mV	0.122mV	
> = FSR	>= 5V	>= 1V	7FFF
FSR-1LSB	4.99939V	0.999878V	3FFF
Midscale +1LSB	0.61mV	0.122mV	2001
Midscale	0V	0V	2000
Midscale –1LSB	-0.61mV	-0.122mV	1FFF
-FSR	-5V	-1V	0000
< -FSR	< -5V	< -1V	4000

NOTE *Bit14 is the out-of-range indicator.*

Synchronizing multiple devices

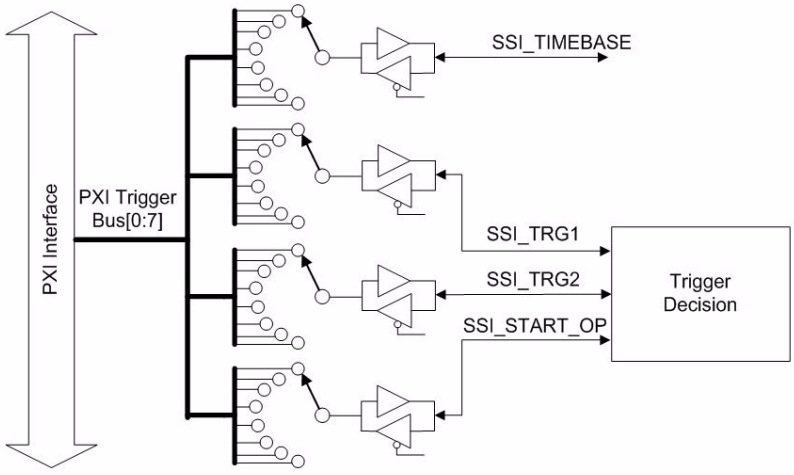
The eight interconnected lines on PXI backplane named as PXI Trigger Bus[0:7] provide a flexible interface for multiple modules synchronization. The Model KPXI-AI-2-65M utilizes the PXI Trigger Bus[0:7] as the System Synchronization Interface (SSI). By providing flexible routing of timebase clock and trigger signals onto PXI Trigger Bus, the Model KPXI-AI-2-65M makes the synchronization between multiple modules easy and simple.

The bi-directional SSI I/Os provide a flexible connection between modules, which allows one SSI master Model KPXI-AI-2-65M to output the SSI signals to other slaves Model KPXI-AI-2-65Ms to receive the signals. Table 3-3 lists SSI timing signals and the functionalities. Figure 3-23 shows the architecture of SSI. Note that it's not allowed to route different signals onto the same trigger bus line.

Table 3-3
Summary of SSI timing signals and the corresponding function

SSI timing signal	Functionality
SSI_TIMEBASE	Input/Output 60MHz timebase signal through SSI
SSI_TRIG1	Input/Output the trigger signal through SSI
SSI_TRIG2	Input/Output the clocked trigger signal through SSI
SSI_START_OP	Input/Output the acquisition start signal in pre-trigger or middle-trigger mode

Figure 3-23
SSI architecture



SSI_TIMEBASE

As an output, the SSI_TIMEBASE signal outputs the onboard 60MHz LVTTL timebase through PXI Trigger Bus.

As an input, the Model KPXI-AI-2-65M accepts the SSI_TIMEBASE signal to be the source of timebase.

SSI_TRIG1

As an output, the SSI_TRIG1 signal reflects the trigger event signal in an acquisition sequence. Users can use the function SSI_SourceConn() to output the SSI_TRIG1 signal.

As an input, the Model KPXI-AI-2-65M accepts the SSI_TRIG1 signal to be the trigger event source. The signal is configured in the rising edge-detection mode. When selecting the trigger sources of the Model KPXI-AI-2-65M, Users can select **TRSRC_SSI_1** to set SSI_TRIG1 as the source of trigger event.

Figure 3-24 and Figure 3-25 show the input and output timing requirements.

Figure 3-24
SSI_TRIG1 output signal timing

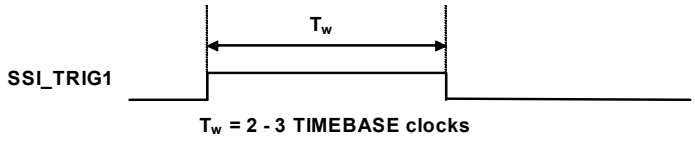
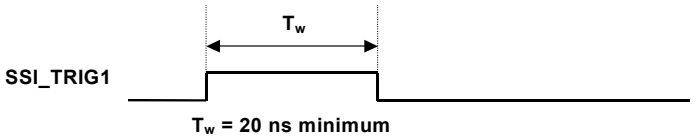


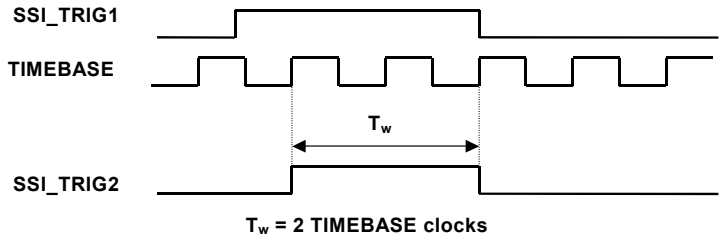
Figure 3-25
SSI_TRIG1 input signal timing



SSI_TRIG2 and SSI_START_OP

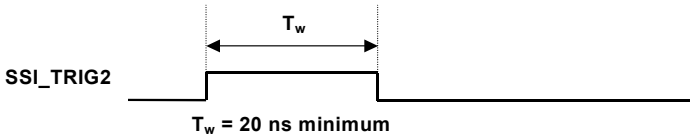
As an output, the SSI_TRIG2 signal is a clocked SSI_TRIG1 signal by TIMEBASE, as illustrated in Figure 3-26.

Figure 3-26
SSI_TRIG2 output signal timing



As an input, the Model KPXI-AI-2-65M accepts the SSI_TRIG2 signal to be the source of a one-clock delayed trigger event. The controller on the Model KPXI-AI-2-65M will then compensate the one-clock delay if using SSI_TRIG2 as the source of trigger event. The signal is configured in the rising edge-detection mode.

Figure 3-27
SSI_TRIG2 input signal timing



As an output, the SSI_START_OP signal reflects the operation start signal in a pre-trigger or middle-trigger acquisition sequence. Please refer to Figure 3-17 through Figure 3-18 for the relationship between the operation start signal and the acquisition sequence.

As an input, the Model KPXI-AI-2-65M accepts the SSI_START_OP signal to be the operation start signal in a pre-trigger or middle-trigger acquisition sequence. The signal is configured in the rising edge-detection mode. Figure 3-28 and Figure 3-29 show the SSI_START_OP signal input and output timing requirements.

For enabling output operations, users can use the function SSI_SourceConn() to output the SSI_TRIG2 and SSI_START_OP signals.

For the input operations, users can select **TRSRC_SSI_2** to set SSI_TRIG2 and SSI_START_OP as the source of the trigger event and operation start signal.

Figure 3-28
SSI_START_OP output signal timing

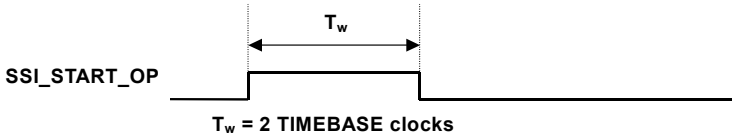
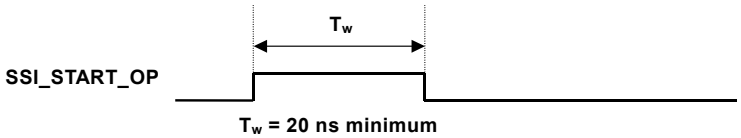


Figure 3-29
SSI_START_OP input signal timing



Comparing the different trigger sources from SSI

When selecting **TRSRC_SSI_1** as the trigger source input, the signal SSI_TRIG1 reflects the trigger event signal in an acquisition sequence. However, when synchronizing multiple Model KPXI-AI-2-65M devices, each Model KPXI-AI-2-65M may recognize the trigger signal with one-clock time difference because the signal is not related to the timebase.

There is another phenomenon if using **TRSRC_SSI_2** in pre-trigger and middle-trigger mode. The operation start signal is generated by a software command so multiple Model KPXI-AI-2-65M devices don't start the data acquisition simultaneously, which may result in the fact that the amount of stored samples are different if the trigger event occurs before the specified amount of data has been acquired.

When selecting **TRSRC_SSI_2** as the trigger source input, SSI_TRIG2 and SSI_START_OP can achieve better synchronization between multiple Model KPXI-AI-2-65M devices. A clocked SSI_TRIG2 can guarantee all Model KPXI-AI-2-65M devices recognize the trigger event at the same clock edge if they use the same timebase. In pre-trigger and middle-trigger mode, SSI_START_OP guarantees all the Model KPXI-AI-2-65M devices start the data acquisition at the same time.

Auto-calibration

NOTE *Before auto-calibration procedure starts, please warm up the module for at least 15 minutes.*

By using the auto-calibration feature of the Model KPXI-AI-2-65M, the calibration software can measure and correct offset and gain errors without any external signal connections, reference voltages, or measurement devices.

After the auto-calibration procedure finishes, the calibration constants can be saved into the EEPROM. In addition to the default bank of factory calibration constants, there are three extra user-modifiable banks in the EERPOM for users to store three sets of calibration constants according to different environments and re-load the calibration constants when necessary.

Because of the fact that errors in measurements will vary with time and temperature, it is recommended that users re-calibrate the Model KPXI-AI-2-65M when the module is installed in a new environment.

Appendix A
KDIG-DRVR User's Guide

In this appendix:

Introduction to KDIG-DRVR

About the KDIG-DRVR software

KDIG-DRVR is a software development kit for Keithley Instruments Model KPXI-AI-2-65M PXI digitizer card. It contains a high performance data acquisition driver for developing custom applications under Windows XP/2000[1] environments.

The memory and data buffer management capabilities free developers from dealing with complex low-level command issues. That is, KDIG-DRVR is constructed to provide a simple programming interface in communication with the Keithley Instruments PXI digitizer. The easy-to-use functions provided by KDIG-DRVR allow a programmer to use the features of the card in a high level way.

Using KDIG-DRVR also allows you to take advantage of the power and features of Microsoft Win32 System for your data acquisition applications, including running multiple applications and using extended memory. Also, using KDIG-DRVR under Visual Basic environment makes it easy to create custom user interfaces and graphics.

KDIG-DRVR hardware support

Although this software is specifically used for the Keithley Instruments Model KPXI-AI-2-65M digitizer card, Keithley will periodically upgrade KDIG-DRVR for other Keithley Instruments digitizer cards. Please refer to the Release Notes for any cards that the current KDIG-DRVR actually supports.

KDIG-DRVR overview

This section describes the classes of functions in KDIG-DRVR and briefly describes each function.

KDIG-DRVR functions are grouped to the following classes:

- General Configuration Function Group
- Analog Input Function Group
  - Analog Input Configuration functions
  - One-Shot Analog Input functions
  - Continuous Analog Input functions
  - Asynchronous Analog Input Monitoring functions
- SSI Function Group
- Calibration Function Group

Fundamentals of building Windows XP/2000 Application

Creating a Windows XP/2000 KDIG-DRVR application using Microsoft Visual C/C++

To create a data acquisition application using KDIG-DRVR and Microsoft Visual C/C++, follow these steps after entering Visual C/C++:

Step 1: Open the project in which you want to use KDIG-DRVR. This can be a new or existing project

Step 2: Include header file **kdigdrvr.h** in the C/C++ source files that call KDIG-DRVR functions. **kdigdrvr.h** contains all the function declarations and constants that you can use to

1. Windows XP, Windows 2000, Microsoft Win32s, Microsoft Visual C/Visual C++, and Microsoft Visual Basic are trademarks of the Microsoft Corporation.

develop your data acquisition application. Incorporate the following statement in your code to include the header file.

```
#include  "kdigdrvr.h"
```

Step 3: Build your application.

Setting the appropriate compile and link options, then build your application by selecting the Build command from Build menu (Visual C/C++ 4.0). Remember to link KDIG-DRVR's import library, KDIG-DRVR.LIB.

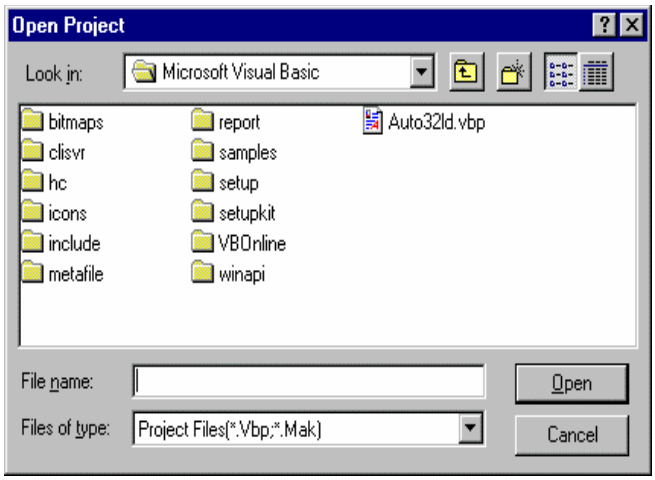
Creating a Windows XP/2000 KDIG-DRVR application using Microsoft Visual Basic

To create a data acquisition application using KDIG-DRVR and Visual Basic, follow these steps after entering Visual Basic:

Step 1: Open the project in which you want to use KDIG-DRVR. This can be a new or existing project.

- Open a new project by selecting the New Project command from the File menu. If it is an existing project, open it by selecting the Open Project command from the File menu. Then the Open Project dialog box appears.

Figure A-1
Open Project dialog box

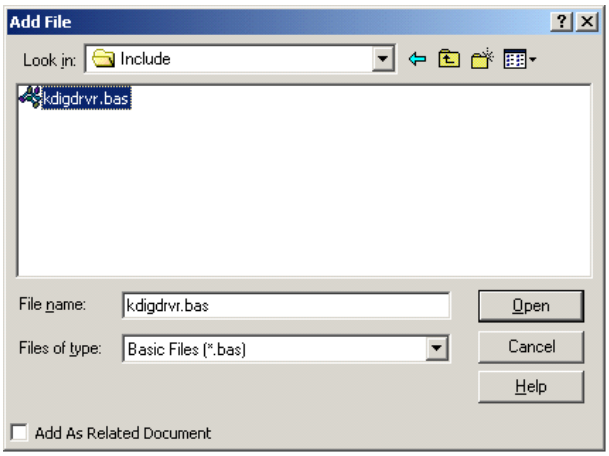


- Change directory to the place the project file is located. Double-click the project file name in the File Name list to load the project.

Step 2: Add file **kdigdrvr.bas** into the project if this file is not already included in the project. This file contains all the procedure declarations and constants that you can use to develop your data acquisition application.

- From the File menu, select the Add File command. The Add File window appears, displaying a list of files in the current directory.
- Select **kdigdrvr.bas** from the Files list by double clicking on it. If you can't find this file in the list, make sure the list is displaying files from the correct directory. By default, **kdigdrvr.bas** is installed in **C:\Keithley\KDIG-DRVR\INCLUDE**.

Figure A-2
Add File dialog box



Step 3: Design the interface for the application. To design the interface, you place the desired elements, such as command button, list box, text box, etc., on the Visual Basic form. These are standard controls from the Visual Basic Toolbox. To place a control on a form, you just move pointer to Toolbox, select the desired control and draw it on the form. Or you can double-click the control icon in the Toolbox to place it on the form.

Step 4: Set properties for the controls. To view the property list, click the desired control and then choose the Properties command from the View menu or press **F4**, or you can also click the Properties button on the toolbar.

Step 5: Write the event code.

The event code defines the action you want to perform when an event occurs. To write the event code, double-click the desired control or form to view the code module and then add code you want. You can call the functions that declared in the file **kdigdrvr.bas** to perform data acquisition operations.

Step 6: Run your application.

- To run the application, choose **Start** from the **Run** menu, or click the Start icon on the toolbar (you can also press **F5**).

Step 7: Distribute your application.

Once you have finished a project, you can save the application as an executable (.EXE) file by using the Make EXE File command on the File menu. And once you have saved your application as an executable file, you've ready to distribute it. When you distribute your application, remember also to include the KDIG-DRVR's DLL and driver files.

Creating a Windows XP/2000 KDIG-DRVR application using Microsoft Visual Basic.NET

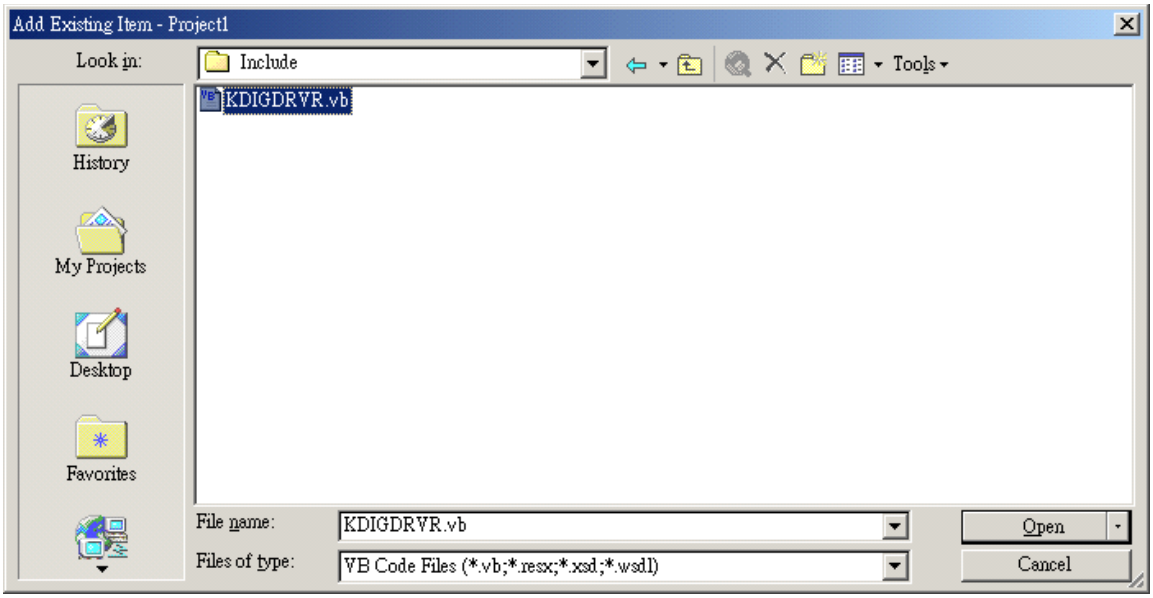
To create a data acquisition application using KDIG-DRVR and Visual Basic.NET, follow these steps after entering Visual Basic.NET:

Step 1: Open the project in which you want to use KDIG-DRVR. This can be a new or existing project.

Step 2: Add file **KDIGDRVR.VB** into the project if this file is not included in the project. This file contains all the procedure declarations and constants that you can use to develop your data acquisition application.

- From the **File** menu, select the **Add File** command. The **Add File window dialog box** appears (Figure A-3), displaying a list of files in the current directory.
- Select **KDIGDRVR.VB** from the Files list by double clicking on it. If you can't find this file in the list, make sure the list is displaying files from the correct directory. By default, **KDIGDRVR.VB** is installed in **C:\Keithley\KDIG-DRVR\INCLUDE**.

Figure A-3
Add Existing Item dialog box



Step 3: Design the interface for the application. To design the interface, you place the desired elements, such as command button, list box, text box, etc., on the Visual Basic form. These are standard controls from the Visual Basic Toolbox. To place a control on a form, you just move pointer to Toolbox, select the desired control and draw it on the form. Or you can double-click the control icon in the Toolbox to place it on the form.

Step 4: Set properties for the controls. To view the property list, click the desired control and then choose the Properties command from the View menu or press **F4**, or you can also click the Properties button on the toolbar.

Step 5: Write the event code.

The event code defines the action you want to perform when an event occurs. To write the event code, double-click the desired control or form to view the code module and then add code you want. You can call the functions that declared in the file **kdigdrvr.bas** to perform data acquisition operations.

Step 6: Run your application.

- To run the application, choose **Start** from the **Run** menu, or click the Start icon on the toolbar (you can also press **F5**).

Step 7: Distribute your application.

Once you have finished a project, you can save the application as an executable (.EXE) file by using the Make EXE File command on the File menu. And once you have saved your application as an executable file, you've ready to distribute it. When you distribute your application, remember also to include the KDIG-DRVR's DLL and driver files.

KDIG-DRVR utilities for Win32

This section introduces the tools that accompany the KDIG-DRVR package.

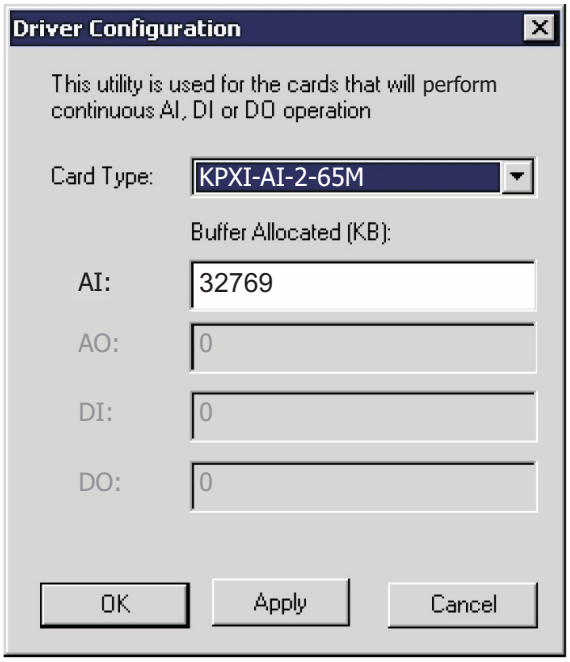
KDIG-DRVR configuration utility (configdrv)

configdrv is used for the users to ***set/modify*** the allocated buffer sizes of AI. The default location of this utility is <InstallDir>\Util directory.

Its unit is page **KB**, i.e. 1024 bytes. Device driver will try to allocate these sizes of memory at system startup time. The size of initially allocated memory is the maximum memory size that DMA or Interrupt transfer can be performed. It will induce an unexpected result in that DMA or Interrupt transfer performed exceeds the initially allocated size.

The "Driver Configuration" window is shown as below.

Figure A-4
Driver configuration window



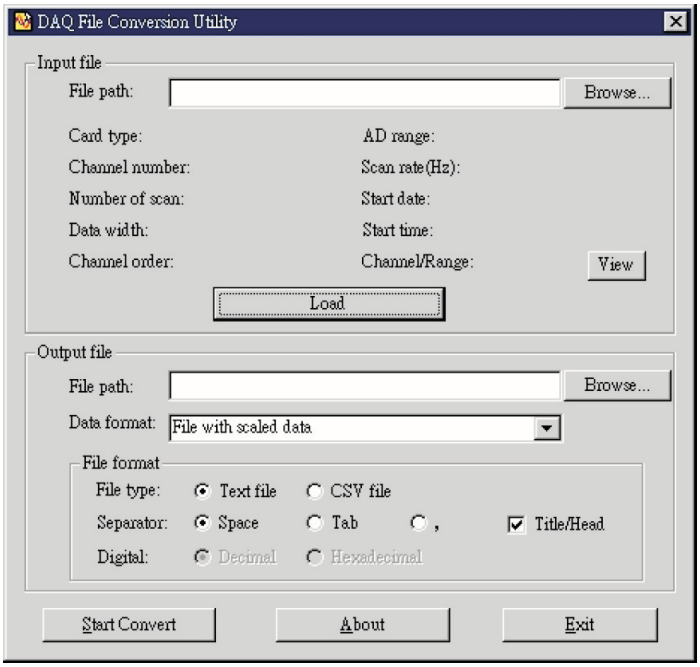
Using **configdrv** to **change the buffer allocated settings** of one of the KDIG-DRVR drivers, select the driver from the **Card Type combo box**.

For the KPXI-AI-2-65M digitizer card only the AI value is settable. The default is 32769 KB.

KDIG-DRVR data file converter utility (KIDAQCvt)

The data files generated by KDIG-DRVR functions performing continuous data acquisition are stored in binary format. Since a binary file can't be read by the normal text editor and can't be used to analyze the accessed data by Microsoft Excel®, KDIG-DRVR provides a convenient tool **KIDAQCvt** to convert the binary file to the file format read easily. The default location of this utility is <InstallDir>\Util directory. The **KIDAQCvt** main window is as the following figure:

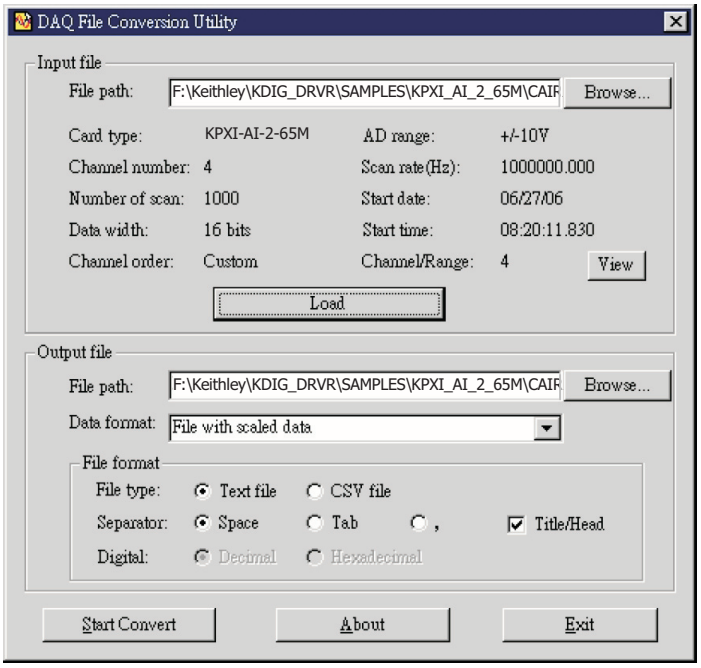
Figure A-5
DAQ File Conversion Utility



The **KIDAQCvt** main window includes two frames. The upper frame, **Input File frame** is used for the source data file and the lower frame is used for the destination file.

To **load the source binary data file**, type the binary data file name in **File Path** field or click **Browser** button to select the source file from **Input File frame**, and then click **Load** button. As the file is loaded, the information related to the data file, e.g. **data type**, **data width**, **AD Range**, etc., are shown in the corresponding fields in **Input File frame**, and the default converted data file path and format are also listed in Figure A-6.

Figure A-6
Loading source binary data file



The default **destination file** with a **.cvt** extension is located in the same directory as the source one. To change the default setting, type the file path you wish or click the **Browser** button from **Output File** frame to select the destination file location.

KIDAQCvt provides three types of data format conversions:

Text file with scaled data:

The data in hexadecimal format is scaled to engineering unit (voltage, amp, etc.) according to the card type, data width and data range and then written to disk in text file format. This type is available for the data accessed from continuous AI operation only.

Binary file with scaled data:

The data in hexadecimal is scaled to engineering unit (voltage, amp, etc.) according to the card type, data width and data range and then written to disk in binary file format. This type is available for the data accessed from continuous AI operation only.

Text file with binary codes:

The data in hexadecimal format or converted to a decimal value is written to disk in text file format. If the original data includes channel information, the raw value will be handled to get the real data value. This type is available for the data accessed form continuous AI and DI operations.

The data separator in converted text file is selectable among **space**, **comma** and **Tab**.

If you want to add title/head which includes the card type information at the beginning of file, check the "Title/Head" box.

After setting the properties (File Path, Format, etc.) related to the converted file, you can push **Start Convert** button from the **Output File** frame to perform the file conversion.

Appendix B

KDIG-DRVR Function Reference

In this appendix:

Function description

This section is provided as a function reference. It contains a detailed description of KDIG-DRVR functions and includes information on KDIG-DRVR Data types as well as a KDIG-DRVR Function reference (functions are arranged alphabetically in the reference). Syntax is provided for Microsoft C/C++ and Borland[1] C++, as well as Visual Basic.

Data types

Table B-1 contains data types defined in **kdigdrvr.h**. These data types are used by the KDIG-DRVR library. It is recommended these data types are used in your application programs. Table B-1 contains data type names, ranges, and the corresponding data types for C/C++,

and Delphi.

NOTE *The data types in Table B-1 are defined in **kdigdrvr.h**, but are not defined in **kdigdrvr.bas** (for .bas definition files, the table is provided only as a reference).*

Table B-1
Suggested data types

Type Name	Description	Range	Type	
			C/C++ (for 32-bit compiler)	Visual Basic
U8	8-bit ASCII character	0 to 255	unsigned char	Byte
I16	16-bit signed integer	-32768 to 32767	short	Integer
U16	16-bit unsigned integer	0 to 65535	unsigned short	Not supported by this type, use the signed integer (I16) instead
I32	32-bit signed integer	-2147483648 to 2147483647	long	Long
U32	32-bit unsigned integer	0 to 4294967295	unsigned long	Not supported by this type, use the signed long integer (I32) instead
F32	32-bit single-precision floating-point	-3.402823E38 to 3.402823E38	float	Single
F64	64-bit double-precision floating-point	-1.797683134862315E308 to 1.797683134862315E309	double	Double

Function reference

KDIG-DRVR is a software driver for Keithley Instruments Digitizer data acquisition cards. It is a high performance data acquisition driver for developing custom applications under Windows environment.

Using KDIG-DRVR also lets you take advantage of the power and features of Microsoft Windows for your data acquisition applications. These include running multiple applications and using extended memory. Also, using KDIG-DRVR under a Windows environment makes it easy to create custom user interfaces and graphics.

1. Borland is a trademark of the Borland Software Corporation.

KDIG_AI_AsyncCheck

Description Check the current status of the asynchronous analog input operation. This function is supported by the following model: KPXI-AI-2-65M

Syntax **Microsoft C/C++ and Borland C++**

```
I16 KDIG_AI_AsyncCheck (U16 CardNumber, BOOLEAN *Stopped,
  U32 *AccessCnt)
```

Visual Basic

```
KDIG_AI_AsyncCheck (ByVal CardNumber As Integer,
  Stopped As Byte, AccessCnt As Long) As Integer
```

Parameters **CardNumber**: The card id of the card that performs the asynchronous operation.

Stopped: Whether the asynchronous analog input operation has completed. If *Stopped* = TRUE, the analog input operation has stopped. Either the number of A/D conversions indicated in the call that initiated the asynchronous analog input operation has completed or an error has occurred. If *Stopped* = FALSE, the operation is not yet complete. (constants TRUE and FALSE are defined in kdigdrvr.h)

AccessCnt: In the condition that the trigger acquisition mode is not used, *AccessCnt* returns the number of A/D data that has been transferred at the time calling **`KDIG_AI_AsyncCheck().`**

If any trigger mode is enabled and double-buffered mode is enabled, *AccessCnt* returns the next position after the position the last A/D data is stored in the circular buffer at the time calling **`KDIG_AI_AsyncCheck().`**

Return Code NoError, ErrorInvalidCardNumber, ErrorCardNotRegistered, ErrorFuncNotSupport

KDIG_AI_AsyncClear

Description Stop the asynchronous analog input operation. This function is supported by the following model: KPXI-AI-2-65M

Syntax **Microsoft C/C++ and Borland C++**

Description I16 KDIG_AI_AsyncClear (U16 CardNumber, U32 *StartPos, U32 *AccessCnt)

Visual Basic

```
KDIG_AI_AsyncClear (ByVal CardNumber As Integer,
  StartPos As Long, AccessCnt As Long) As Integer
```

Parameters **CardNumber**:The card id of the card that performs the asynchronous operation.

StartPos: In the condition that the trigger acquisition mode is not used, *StartPos* is zero. If "*Pre-trigger or middle trigger*" mode of AI is used, *StartPos* returns the position of the first AD data in the data buffer at the time calling **`KDIG_AI_AsyncClear().`**

AccessCnt: In the condition that the pre-/middle trigger acquisition mode is not used, *AccessCnt* returns the number of A/D data that has been transferred at the time calling **`KDIG_AI_AsyncClear().`**

If double-buffered mode is enabled, *AccessCnt* returns the next position after the position the last A/D data is stored in the circular buffer.

Return Code NoError, ErrorInvalidCardNumber, ErrorCardNotRegistered, ErrorFuncNotSupport

KDIG_AI_AsyncDblBufferHalfReady

Description Checks whether the next half buffer of data in circular buffer is ready for transfer during an asynchronous double-buffered analog input operation. This function is supported by the following model: KPXI-AI-2-65M

Syntax **Microsoft C/C++ and Borland C++**

```
I16 KDIG_AI_AsyncDblBufferHalfReady (U16 CardNumber,
  BOOLEAN *HalfReady, BOOLEAN *StopFlag)
```

Visual Basic

```
KDIG_AI_AsyncDblBufferHalfReady(ByVal CardNumber As Integer,
  HalfReady As Byte, StopFlag As Byte) As Integer
```

Parameters **CardNumber:** The card id of the card that performs the asynchronous double-buffered operation.

HalfReady: Whether the next half buffer of data is available. (constants TRUE and FALSE are defined in kdigdrvr.h)

StopFlag: Whether the asynchronous analog input operation has completed. If *StopFlag* = TRUE, the analog input operation has stopped. If *StopFlag* = FALSE, the operation is not yet complete. (constants TRUE and FALSE are defined in kdigdrvr.h)

Return Code NoError, ErrorInvalidCardNumber, ErrorCardNotRegistered, ErrorFuncNotSupport

KDIG_AI_AsyncDblBufferMode

Description Enables or disables double-buffered data acquisition mode. **The double buffer mode of continuous A/D conversions is only supported for Post-trigger and Delay trigger mode operation.** This function is supported by the following model: KPXI-AI-2-65M

Syntax **Microsoft C/C++ and Borland C++**

```
I16 KDIG_AI_AsyncDblBufferMode (U16 CardNumber,
  BOOLEAN Enable)
```

Visual Basic

```
KDIG_AI_AsyncDblBufferMode (ByVal CardNumber As Integer,
  ByVal Enable As Byte) As Integer
```

Parameters **CardNumber:** The card id of the card that double-buffered mode to be set.

Enable: Whether the double-buffered mode is enabled or not.

TRUE: double-buffered mode is enabled.

FALSE: double-buffered mode is disabled.

(constants TRUE and FALSE are defined in kdigdrvr.h)

Return Code NoError, ErrorInvalidCardNumber, ErrorCardNotRegistered, ErrorFuncNotSupport

KDIG_AI_AsyncDblBufferToFile

Description If the continuous AI function is *KDIG_AI_ContReadChannelToFil*e, *KDIG_AI_ContReadMultiChannelsToFile, and KDIG_AI_ContScanChannelsToFil*e, calling this function to log the data of the circular buffer into a disk file. This function is supported by the following model: KPXI-AI-2-65M

Syntax **Microsoft C/C++ and Borland C++**

```
I16 KDIG_AI_AsyncDblBufferToFile (U16 CardNumber)
```

Visual Basic

```
KDIG_AI_AsyncDblBufferToFile (ByVal CardNumber As Integer)
  As Integer
```

Parameters **CardNumber:** The card id of the card that double-buffered mode to be set.

Return Code NoError, ErrorInvalidCardNumber, ErrorCardNotRegistered, ErrorFuncNotSupport

KDIG_AI_AsyncReStartNextReady

Description Checks the status of re-start mode of continuous AI. This function is only valid for post trigger and delay trigger mode of re-started continuous AI operation. This function is supported by the following model: KPXI-AI-2-65M

Syntax **Microsoft C/C++ and Borland C++**

```
I16 KDIG_AI_AsyncReStartNextReady (U16 wCardNumber,
  BOOLEAN *DaqReady, BOOLEAN *StopFlag, U16 *RdyDaqCnt)
```

Visual Basic

```
KDIG_AI_AsyncReStartNextReady (ByVal CardNumber As Integer,
  DaqReady As Byte, StopFlag As Byte, RdyDaqCnt As Byte)
  As Integer
```

Parameters **CardNumber:** The card id of the card that performs the asynchronous double-buffered operation.

DaqReady: Whether next data conversion is completed. (constants TRUE and FALSE are defined in kdigdrvr.h)

StopFlag: Whether the definite count of re-start mode of data conversion is completed. If *StopFlag* = TRUE, the analog input operation has stopped. If *StopFlag* = FALSE, the operation is not yet complete. (constants TRUE and FALSE are defined in kdigdrvr.h)

RdyDaqCnt: If re-start count is definite, this argument returns the count of finished data acquisition. If the re-start count is infinite, this argument returns the index of the buffer which stores the most recent data

Return Code NoError, ErrorInvalidCardNumber, ErrorCardNotRegistered, ErrorFuncNotSupport

KDIG_AD_Auto_Calibration_ALL

Description Use this function to calibrate your KDIG-DRVR device. When the function is called, the device goes into a self-calibration cycle. The function does not return until the self-calibration is completed. This function is supported by the following model: KPXI-AI-2-65M

Syntax **Microsoft C/C++ and Borland C++**

```
I16 KDIG_AD_Auto_Calibration_ALL(U16 CardNumber)
```

Visual Basic

```
KDIG_AD_Auto_Calibration_ALL (ByVal wCardNumber As Integer)
  As Integer
```

Parameters **CardNumber:** The card id of the card that want to perform this operation.

Return Code NoError, ErrorInvalidCardNumber, ErrorCardNotRegistered, ErrorFuncNotSupport, ErrorInvalidIoChannel

KDIG_AI_CH_Config

Description Informs KDIG-DRVR library of the AI range selected for the specified channel of the card with card ID *CardNumber*. After the function "KDIG_Register_Card" is called, all of the analog input channels the configured as configured as AD_B_5_V by default. If you wish to perform the device with the default settings, it is not necessary to call this function to configure the channel(s) again. Otherwise, this function has to be called to program the device for the settings you want before calling function to perform analog input operation. This function is supported by the following model: KPXI-AI-2-65M

Syntax **Microsoft C/C++ and Borland C++**

```
I16 KDIG_AI_CH_Config (U16 wCardNumber, U16 wChannel,
  U16 wAdRange)
```

Visual Basic

```
KDIG_AI_CH_Config (ByVal CardNumber As Integer,
  ByVal Channel As Integer, ByVal AdRange As Integer)
  As Integer
```

Parameters **CardNumber:** The card id of the card that want to perform this operation.

Channel: The A/D channel wished to do the channel setting.

Valid values:

KPXI-AI-2-65M: 0 through 1 or All_Channels (-1)

AdRange: The settings for analog input channel. This argument is an integer expression formed from one or more of the manifest constants defined in kdigdrvr.h.

We define some constants to represent various A/D input ranges in kdigdrvr.h. Please refer to AI range codes for the valid range values. The default setting is AD_B_5_V.

Return Code NoError, ErrorInvalidCardNumber, ErrorCardNotRegistered, ErrorFuncNotSupport

KDIG_AI_Config

Description Informs KDIG-DRVR library of the source of Timebase, conversion source and sampling mode for the KDIG-DRVR device with card ID *CardNumber*. After the function "KDIG_Register_Card" is called, the device is configured as the following by default:

Time Base: KDIG_IntTimeBase

A/D conversion source: KDIG_AI_ADCONVSRC_TimePacer

A/D sampling mode: Double-edge-sampling disabled

adDutyRestore: AD duty cycle restore function enabled

Auto reset buffer: Enabled (AutoResetBuf: TRUE)

If you wish to perform the device with the default settings, it is not necessary to call this function to make the configuration again. Otherwise, this function has to be called before calling function to perform continuous analog input operation. This function is supported by the following model: KPXI-AI-2-65M

Syntax

Microsoft C/C++ and Borland C++

```
I16 KDIG_AI_Config (U16 wCardNumber, U16 TimeBase,
  BOOLEAN adDutyRestore, U16 ConvSrc, BOOLEAN doubleEdged,
  BOOLEAN AutoResetBuf)
```

Visual Basic

```
KDIG_AI_Config (ByVal CardNumber As Integer,
  ByVal TimeBase As Integer, ByVal adDutyRestore As Byte,
  ByVal ConvSrc As Integer, ByVal doubleEdged As Byte,
  ByVal AutoResetBuf As Byte) As Integer
```

Parameters **CardNumber:** The card id of the card that want to perform this operation.

TimerBase: The Time Base the device selected. The valid values are:

KDIG_IntTimeBase: Internal timer as the time base

KDIG_ExtTimeBase: External timer as the time base

KDIG_StartTimeBase: The clock source is from PXI_START line

KDIG_SSITimeBase: The timer based on the SSI source

KDIG_PLL_REF_PXICLK10: phase-locked loop (PLL) reference clock source with an PXI_CLK10 to be the source for the phase-locked loop

KDIG_PLL_REF_EXT10: phase-locked loop (PLL) reference clock source with an external reference clock to be the source for the phase-locked loop

adDutyRestore:

TRUE: Activate AD duty cycle restore function (default)
FALSE: Deactivate AD duty cycle restore function

ConvSrc: A/D Conversion Source Selection. The valid values are:
KDIG_AI_ADCONVSRC_TimePacer: timer pacer (default)

doubleEdged:

FALSE: Disable AD alternating mode (default)
TRUE: Enable AD alternating mode

AutoResetBuf:

FALSE: The AI buffers set by function "KDIG_AI_ContBufferSetup" are retained and must call function "KDIG_AI_ContBufferReset" to reset the buffer

TRUE: The AI buffers set by function "KDIG_AI_ContBufferSetup" are reset automatically by driver while the AI operation is finished

Return Code NoError, ErrorInvalidCardNumber, ErrorCardNotRegistered, ErrorFuncNotSupport

KDIG_AI_ContBufferReset

Description This function reset all the buffers set by function *KDIG_AI_ContBufferSetup* for continuous analog input. The function has to be called if the data buffers won't be used. This function is supported by the following model: KPXI-AI-2-65M

Syntax **Microsoft C/C++ and Borland C++**

```
I16 KDIG_AI_ContBufferReset (U16 wCardNumber)
```

Visual Basic

```
KDIG_AI_ContBufferReset (ByVal CardNumber As Integer)
  As Integer
```

Parameters **CardNumber:** The card id of the card that want to perform this operation.

Return Code NoError, ErrorInvalidCardNumber, ErrorCardNotRegistered, ErrorFuncNotSupport, ErrorTransferCountTooLarge , ErrorContIoNotAllowed

KDIG_AI_ContBufferSetup

Description This function set up the buffer for continuous analog input. The function has to be called repeatedly to setup all of the data buffers (at most 2 buffers). This function is supported by the following model: KPXI-AI-2-65M

Syntax **Microsoft C/C++ and Borland C++**

```
I16 KDIG_AI_ContBufferSetup (U16 wCardNumber,
  void *pwBuffer, U32 dwReadCount, U16 *BufferId)
```

Visual Basic

```
KDIG_AI_ContBufferSetup (ByVal CardNumber As Integer,
  Buffer As Any, ByVal ReadCount As Long,
  BufferId As Integer) As Integer
```

Parameters **CardNumber:** The card id of the card that want to perform this operation.

Buffer: The starting address of the memory to contain the input data.

ReadCount: The size (in samples) of the buffer and its value must be even.

BufferId: Returns the index of the buffer currently set up.

Return Code NoError, ErrorInvalidCardNumber, ErrorCardNotRegistered, ErrorFuncNotSupport, ErrorTransferCountTooLarge , ErrorContIoNotAllowed

KDIG_AI_ContReadChannel

Description This function performs continuous A/D conversions on the specified analog input channel at a rate as close to the rate you specified. **The double buffer mode of continuous A/D conversions is only supported for Post-trigger and Delay trigger mode operation.** This function is supported by the following model: KPXI-AI-2-65M

Syntax

Microsoft C/C++ and Borland C++

```
I16 KDIG_AI_ContReadChannel (U16 CardNumber, U16 Channel,
  U16 BufId, U32 ReadScans, U32 ScanIntrv, U32 SampIntrv,
  U16 SyncMode)
```

Visual Basic

```
KDIG_AI_ContReadChannel (ByVal CardNumber As Integer,
  ByVal Channel As Integer, ByVal BufId As Integer,
  ByVal ReadScans As Long, ByVal ScanIntrv As Long,
  ByVal SampIntrv As Long, ByVal SyncMode As Integer)
  As Integer
```

Parameters **CardNumber:** The card id of the card that want to perform this operation.

Channel: Analog input channel number

Range: 0 through 1 for KPXI-AI-2-65M

BufId: The buffer ID (returned from function *KDIG_AI_ContBufferSetup*) of the buffer containing the acquired data. If not performing two steps of data acquisition (DAQSTEPPED), the size of the buffer with buffer id of *BufId* must have a length equal to the value of parameter *ReadScans*. If double-buffered mode is enabled, The starting buffer id should be 0. You can ignore this argument. Please refer to AI data format for the data format in the buffer with *BufId*.

ReadScans: If double-buffered mode is disabled, *ReadScans* is the total number of scans to be performed. For pre-trigger and middle-trigger mode operation, the ReadScans must be a multiple of 32. The total data bytes can't exceed the size of onboard sdram. For double-buffered acquisition, *ReadScans* is the size (in samples) allocated for each channel in the circular buffer. This value must be a multiple of 2.

NOTE *For pre-trigger and middle trigger mode of AI, the **ReadScans** has to be equal to **postTrigScans+preTrigScans**.*

ScanIntrv: The length of the scan interval (that is, the counter value between the initiation of each scan sequence). The scan rate will be *TimeBase/ScanIntrv*. The value of *TimeBase* depends on the card type.

The valid range of the value is as follows:

KPXI-AI-2-65M: 1 through 16777215

SampIntrv: The length of the sample interval (that is, the counter value between each A/D conversion within a scan sequence). The A/D conversion rate will be *TimeBase/SampIntrv*. The value of *TimeBase* depends on the card type.

The valid range of the value is as follows:

KPXI-AI-2-65M: 1 through 16777215

SyncMode: Whether this operation is performed synchronously or asynchronously. If pre-/middle trigger mode is enabled by calling **`KDIG_AI_Trig_Config()`**, this operation should be performed *asynchronously*.

Valid values:

SYNCH_OP:
Synchronous A/D conversion, that is, the function does not return until the A/D operation complete.

ASYNCH_OP:
Asynchronous A/D conversion.

Return Code NoError, ErrorInvalidCardNumber, ErrorCardNotRegistered, ErrorFuncNotSupport, ErrorInvalidIoChannel, ErrorInvalidAdRange, ErrorTransferCountTooLarge, ErrorContIoNotAllowed, ErrorInvalidSampleRate

KDIG_AI_ContReadChannelToFile

Description This function performs continuous A/D conversions on the specified analog input channel at a rate as close to the rate you specified and saves the acquired data in a disk file. The data is written to disk in binary format, with the lower byte first (little endian). Please refer to AI data format for the data file structure. **The double buffer mode of continuous A/D conversions is only supported for Post-trigger and Delay trigger mode operation. This** function is supported by the following model: KPXI-AI-2-65M

Syntax

Microsoft C/C++ and Borland C++

```
I16 KDIG_AI_ContReadChannelToFile (U16 CardNumber,
  U16 Channel, U16 BufId, U8 *FileName, U32 ReadScans,
  U32 ScanIntrv, U32 SampIntrv, U16 SyncMode)
```

Visual Basic

```
KDIG_AI_ContReadChannelToFile (ByVal CardNumber As Integer,
  ByVal Channel As Integer, ByVal BufId As Integer,
  ByVal FileName As String, ByVal ReadScans As Long,
  ByVal ScanIntrv As Long, ByVal SampIntrv As Long,
  ByVal SyncMode As Integer) As Integer
```

Parameters **CardNumber:** The card id of the card that want to perform this operation.

Channel: Analog input channel number

Range: 0 through 1 for KPXI-AI-2-65M

BufId: The buffer ID (returned from function`KDIG_AI_ContBufferSetup`) of the buffer containing the acquired data. If not performing two steps of data acquisition (DAQSTEPPED), the size of the buffer with buffer id of *BufId* must

have a length equal to the value of parameter *ReadScans*. If double-buffered mode is enabled, The starting buffer id should be 0. You can ignore this argument. Please refer to AI data format for the data format in the buffer with *BufId*.

FileName: Name of data file which stores the acquired data

ReadScans: If double-buffered mode is disabled, the total number of scans to be performed. The total data bytes can't exceed the size of onboard sdram. For double-buffered acquisition, *ReadScans* is the size (in samples) allocated for each channel in the circular buffer. This value must be a multiple of 2. For pre-trigger and middle-trigger mode operation, the ReadScans must be a multiple of 32.

NOTE *For pre-trigger and middle trigger mode of AI, the* ***ReadScans*** *has to be equal to* ***postTrigScans+preTrigScans****.*

ScanIntrv: The length of the scan interval (that is, the counter value between the initiation of each scan sequence). The scan rate will be *TimeBase/ScanIntrv*. The value of *TimeBase* depends on the card type.

The valid range of the value is as follows:

KPXI-AI-2-65M: 1 through 16777215

SampIntrv: The length of the sample interval (that is, the counter value between each A/D conversion within a scan sequence). The A/D conversion rate will be *TimeBase/SampIntrv*. The value of *TimeBase* depends on the card type.

The valid range of the value is as follows:

KPXI-AI-2-65M: 1 through 16777215

SyncMode: Whether this operation is performed synchronously or asynchronously. If pre-/middle trigger mode is enabled by calling **`KDIG_AI_Trig_Config()`**, this operation should be performed *asynchronously*.

Valid values:

SYNCH_OP:
Synchronous A/D conversion, that is, the function does not return until the A/D operation complete.

ASYNCH_OP:
Asynchronous A/D conversion

Return Code NoError, ErrorInvalidCardNumber, ErrorCardNotRegistered, ErrorFuncNotSupport, ErrorInvalidIoChannel, ErrorInvalidAdRange, ErrorTransferCountTooLarge, ErrorContIoNotAllowed, ErrorInvalidSampleRate, ErrorOpenFile

KDIG_AI_ContReadMultiChannels

Description This function performs continuous A/D conversions on the specified analog input channels at a rate as close to the rate you specified. **The double buffer mode of continuous A/D conversions is only supported for Post-trigger and Delay**

trigger mode operation. This function is supported by the following model: KPXI-AI-2-65M

Syntax

Microsoft C/C++ and Borland C++

```
I16 KDIG_AI_ContReadMultiChannels (U16 CardNumber,
  U16 NumChans, U16 *Chans, U16 BufId, U32 ReadScans,
  U32 ScanIntrv, U32 SampIntrv, U16 SyncMode)
```

Visual Basic

```
KDIG_AI_ContReadMultiChannels (ByVal CardNumber As Integer,
  ByVal NumChans As Integer, chans As Integer,
  ByVal BufId As Integer, ByVal ReadScans As Long,
  ByVal ScanIntrv As Long, ByVal SampIntrv As Long,
  ByVal SyncMode As Integer) As Integer
```

Parameters

CardNumber: The card ID of the card that want to perform this operation.

numChans: The number of analog input channels in the array *Chans*. The valid value:

KPXI-AI-2-65M: 1 through 2

Chans: Array of analog input channel numbers.

KPXI-AI-2-65M: numbers in *Chans* must be within 0 and 1.

BufId: The buffer ID (returned from function *KDIG_AI_ContBufferSetup*) of the buffer containing the acquired data. If not performing two steps of data acquisition (DAQSTEPPED), the size of the buffer with buffer id of *BufId* must have a length equal to the value of parameter *ReadScans*. If double-buffered mode is enabled, The starting buffer id should be 0. You can ignore this argument. Please refer to AI data format for the data format in the buffer with *BufId*.

ReadScans: If double-buffered mode is disabled, the total number of scans to be performed. The total data bytes can't exceed the size of onboard sdram. For double-buffered acquisition, *ReadScans* is the size (in samples) allocated for each channel in the circular buffer. This value must be a multiple of 2. For pre-trigger and middle-trigger mode operation, the ReadScans must be a multiple of 32.

NOTE *For pre-trigger and middle trigger mode of AI, the* ***ReadScans*** *has to be equal to* ***postTrigScans+preTrigScans****.*

ScanIntrv: The length of the scan interval (that is, the counter value between the initiation of each scan sequence). The scan rate will be *TimeBase/ScanIntrv*. The value of *TimeBase* depends on the card type.

The valid range of the value is as follows:

KPXI-AI-2-65M: 1 through 16777215

SampIntrv: The length of the sample interval (that is, the counter value between each A/D conversion within a scan sequence). The A/D conversion rate will be *TimeBase/SampIntrv*. The value of *TimeBase* depends on the card type.

The valid range of the value is as follows:

KPXI-AI-2-65M: 1 through 16777215

SyncMode: Whether this operation is performed synchronously or asynchronously. If pre-/middle trigger mode is enabled by calling **`KDIG_AI_Trig_Config()`**, this operation should be performed **asynchronously**.

Valid values:

SYNCH_OP:
Synchronous A/D conversion, that is, the function does not return until the A/D operation complete.

ASYNCH_OP:
Asynchronous A/D conversion

Return Code NoError, ErrorInvalidCardNumber, ErrorCardNotRegistered, ErrorFuncNotSupport, ErrorInvalidIoChannel, ErrorInvalidSampleRate, ErrorInvalidAdRange, ErrorTransferCountTooLarge, ErrorContIoNotAllowed

KDIG_AI_ContReadMultiChannelsToFile

Description This function performs continuous A/D conversions on the specified analog input channels at a rate as close to the rate you specified and saves the acquired data in a disk file. The data is written to disk in binary format, with the lower byte first (little endian). Please refer to DATA file format for the data file structure and AI data format for the format of the data in the data file. **The double buffer mode of continuous A/D conversions is only supported for Post-trigger and Delay trigger mode operation.** This function is supported by the following model: KPXI-AI-2-65M

Syntax

Microsoft C/C++ and Borland C++

```
I16 KDIG_AI_ContReadMultiChannelsToFile (U16 CardNumber,
  U16 NumChans, U16 *Chans, U16 BufId, U8 *FileName,
  U32 ReadScans, U32 ScanIntrv, U32 SampIntrv, U16 SyncMode)
```

Visual Basic

```
KDIG_AI_ContScanChannelsToFile (ByVal wCardNumber As
Integer, ByVal wChannel As Integer, ByVal BufId As Integer,
ByVal FileName As String, ByVal ReadScans As Long, ByVal
ScanIntrv As Long, ByVal SampIntrv As Long, ByVal SyncMode
As Integer) As Integer
```

Parameters **CardNumber:** The card ID of the card that want to perform this operation.

numChans: The number of analog input channels in the array *Chans*. The valid value:

KPXI-AI-2-65M: 1 through 2

Chans: Array of analog input channel numbers.

KPXI-AI-2-65M: numbers in *Chans* must be within 0 and 1.

BufId: The buffer ID (returned from function *KDIG_AI_ContBufferSetup*) of the buffer containing the acquired data. If not performing two steps of data acquisition (DAQSTEPPED), the size of the buffer with buffer id of *BufId* must have a length equal to the value of parameter *ReadScans*. If double-buffered mode is enabled, The starting buffer id should be 0. You can ignore this argument. Please refer to AI data format for the data format in the buffer with *BufId*.

FileName: Name of data file which stores the acquired data

ReadScans: If double-buffered mode is disabled, the total number of scans to be performed. The total data bytes can't exceed the size of onboard sdram. For double-buffered acquisition, *ReadScans* is the size (in samples) allocated for each channel in the circular buffer. This value must be a multiple of 2. For pre-trigger and middle-trigger mode operation, the ReadScans must be a multiple of 32.

NOTE *For pre-trigger and middle trigger mode of AI, the* ***ReadScans*** *has to be equal to* ***postTrigScans+preTrigScans****.*

ScanIntrv: The length of the scan interval (that is, the counter value between the initiation of each scan sequence). The scan rate will be *TimeBase/ScanIntrv*. The value of *TimeBase* depends on the card type.

The valid range of the value is as follows:

KPXI-AI-2-65M: 1 through 16777215

SampIntrv: The length of the sample interval (that is, the counter value between each A/D conversion within a scan sequence). The A/D conversion rate will be *TimeBase/SampIntrv*. The value of *TimeBase* depends on the card type.

The valid range of the value is as follows:

KPXI-AI-2-65M: 1 through 16777215

SyncMode: Whether this operation is performed synchronously or asynchronously. If pre-/middle trigger mode is enabled by calling **`KDIG_AI_Trig_Config()`**, this operation should be performed **asynchronously**.

Valid values:

SYNCH_OP:
Synchronous A/D conversion, that is, the function does not return until the A/D operation complete.

ASYNCH_OP:
Asynchronous A/D conversion

Return Code NoError, ErrorInvalidCardNumber, ErrorCardNotRegistered, ErrorFuncNotSupport, ErrorInvalidIoChannel, ErrorInvalidSampleRate, ErrorInvalidAdRange, ErrorTransferCountTooLarge, ErrorContIoNotAllowed, ErrorOpenFile

KDIG_AI_ContScanChannels

Description This function performs continuous A/D conversions on the specified continuous analog input channels at a rate as close to the rate you specified. This function takes advantage of the hardware simultaneous or auto-scan functionality to perform multi-channel analog input. **The double buffer mode of continuous A/D conversions is only supported for Post-trigger and Delay trigger mode operation.** This function is supported by the following model: KPXI-AI-2-65M

Syntax

Microsoft C/C++ and Borland C++

```
I16 KDIG_AI_ContScanChannels (U16 CardNumber, U16 Channel,
  U16 BufId, U32 ReadScans, U32 ScanIntrv, U32 SampIntrv,
  U16 SyncMode)
```

Visual Basic

```
KDIG_AI_ContScanChannels (ByVal wCardNumber As Integer,
  ByVal wChannel As Integer, ByVal BufId As Integer,
  ByVal ReadScans As Long, ByVal ScanIntrv As Long,
  ByVal SampIntrv As Long, ByVal SyncMode As Integer)
  As Integer
```

Parameters **CardNumber:** The card ID of the card that want to perform this operation.

Channel: The largest channel number of specified continuous analog input channel. The channel order for acquiring data is as follows:

KPXI-AI-2-65M: number of *Channel* must be within 0 and 1. The continuous scan sequence is ascending and the first one must be zero. For example, 0,

BufId: The buffer ID (returned from function *KDIG_AI_ContBufferSetup*) of the buffer containing the acquired data. If not performing two steps of data acquisition (DAQSTEPPED), the size of the buffer with buffer id of *BufId* must have a length equal to the value of parameter *ReadScans*. If double-buffered mode is enabled, The starting buffer id should be 0. You can ignore this argument. Please refer to AI data format for the data format in the buffer with *BufId*.

ReadScans: If double-buffered mode is disabled, the total number of scans to be performed. The total data bytes can't exceed the size of onboard sdram. For double-buffered acquisition, *ReadScans* is the size (in samples) allocated for each channel in the circular buffer. This value must be a multiple of 2. For pre-trigger and middle-trigger mode operation, the ReadScans must be a multiple of 32.

NOTE *For pre-trigger and middle trigger mode of AI, the **ReadScans** has to be equal to **postTrigScans+preTrigScans**.*

ScanIntrv: The length of the scan interval (that is, the counter value between the initiation of each scan sequence). The scan rate will be *TimeBase/ScanIntrv*. The value of *TimeBase* depends on the card type.

The valid range of the value is as follows:

KPXI-AI-2-65M: 1 through 16777215

SampIntrv: The length of the sample interval (that is, the counter value between each A/D conversion within a scan sequence). The A/D conversion rate will be *TimeBase/SampIntrv*. The value of *TimeBase* depends on the card type.

The valid range of the value is as follows:

KPXI-AI-2-65M: 1 through 16777215

SyncMode:

Whether this operation is performed synchronously or asynchronously. If pre-/middle trigger mode is enabled by calling **`KDIG_AI_Trig_Config()`**, this operation should be performed **asynchronously**.

Valid values:

SYNCH_OP:
Synchronous A/D conversion, that is, the function does not return until the A/D operation complete.

ASYNCH_OP:
Asynchronous A/D conversion

Return Code NoError, ErrorInvalidCardNumber, ErrorCardNotRegistered, ErrorFuncNotSupport, ErrorInvalidIoChannel, ErrorInvalidSampleRate, ErrorInvalidAdRange, ErrorTransferCountTooLarge, ErrorContIoNotAllowed, ErrorLastChannelNotZero, ErrorDiffRangeNotSupport, ErrorChannelNotDescending, ErrorChannelNotAscending

KDIG_AI_ContScanChannelsToFile

Description This function performs continuous A/D conversions on the specified continuous analog input channels at a rate as close to the rate you specified and saves the acquired data in a disk file. The data is written to disk in binary format, with the lower byte first (little endian). Please refer to DATA file format for the data file structure and Appendix C, *AI Data Format* for the format of the data in the data file. This function takes advantage of the hardware simultaneous or auto-scan functionality to perform multi-channel analog input. **The double buffer mode of continuous A/D conversions is only supported for Post-trigger and Delay trigger mode operation.** This function is supported by the following model: KPXI-AI-2-65M

Syntax

Microsoft C/C++ and Borland C++

```
I16 KDIG_AI_ContScanChannelsToFile (U16 CardNumber,
  U16 Channel, U16 BufId, U8 *FileName, U32 ReadScans,
  U32 ScanIntrv, U32 SampIntrv, U16 SyncMode);
```

Visual Basic

```
KDIG_AI_ContScanChannelsToFile (
  ByVal wCardNumber As Integer, ByVal wChannel As Integer,
  ByVal BufId As Integer, ByVal FileName As String,
  ByVal ReadScans As Long, ByVal ScanIntrv As Long,
  ByVal SampIntrv As Long, ByVal SyncMode As Integer)
  As Integer
```

Parameters **CardNumber:** The card ID of the card that want to perform this operation.

Channel: The largest channel number of specified continuous analog input channel. The channel order for acquiring data is as follows:

KPXI-AI-2-65M: number of *Channel* must be within 0 and 1. The continuous scan sequence is ascending and the first one must be zero. For example, 0, 1.

BufId: The buffer ID (returned from function `KDIG_AI_ContBufferSetup`) of the buffer containing the acquired data. If not performing two steps of data acquisition (DAQSTEPPED), the size of the buffer with buffer id of *BufId* must have a length equal to the value of parameter *ReadScans*. If double-buffered mode is enabled, The starting buffer id should be 0. You can ignore this argument. Please refer to AI data format for the data format in the buffer with *BufId*.

FileName: Name of data file which stores the acquired data

ReadScans: If double-buffered mode is disabled, the total number of scans to be performed. The total data bytes can't exceed the size of onboard sdram. For double-buffered acquisition, *ReadScans* is the size (in samples) allocated for each channel in the circular buffer. This value must be a multiple of 2. For pre-trigger and middle-trigger mode operation, the ReadScans must be a multiple of 32.

NOTE *For pre-trigger and middle trigger mode of AI, the* ***ReadScans*** *has to be equal to* ***postTrigScans+preTrigScans****.*

ScanIntrv: The length of the scan interval (that is, the counter value between the initiation of each scan sequence). The scan rate will be *TimeBase/ScanIntrv*. The value of *TimeBase* depends on the card type.

The valid range of the value is as follows:

KPXI-AI-2-65M: 1 through 16777215

SampIntrv: The length of the sample interval (that is, the counter value between each A/D conversion within a scan sequence). The A/D conversion rate will be *TimeBase/SampIntrv*. The value of *TimeBase* depends on the card type.

The valid range of the value is as follows:

KPXI-AI-2-65M: 1 through 16777215

SyncMode: Whether this operation is performed synchronously or asynchronously. If pre-/middle trigger mode is enabled by calling **`KDIG_AI_Trig_Config()`**, this operation should be performed **asynchronously**.

Valid values:

SYNCH_OP:
Synchronous A/D conversion, that is, the function does not return until the A/D operation complete.

ASYNCH_OP:
Asynchronous A/D conversion

Return Code NoError, ErrorInvalidCardNumber, ErrorCardNotRegistered, ErrorFuncNotSupport, ErrorInvalidIoChannel, ErrorInvalidSampleRate, ErrorInvalidAdRange, ErrorTransferCountTooLarge, ErrorContIoNotAllowed, ErrorLastChannelNotZero, ErrorDiffRangeNotSupport, ErrorChannelNotDescending, ErrorChannelNotAscending

KDIG_AI_ContStatus

Description While performing continuous A/D conversions, this function is called to get the A/D status. Please refer to the manual for your device for the AI status the device might meet. This function is supported by the following model: KPXI-AI-2-65M

Syntax **Microsoft C/C++ and Borland C++**

```
I16 KDIG_AI_ContStatus (U16 CardNumber, U32 *Status)
```

Visual Basic

```
KDIG_AI_ContStatus (ByVal CardNumber As Integer,
  Status As Long) As Integer
```

Parameters **CardNumber:** The card id of the card that want to perform this operation.

Status: The continuous AI status returned. The description of the parameter *Status* for various card types is the following:

KPXI-AI-2-65M:

bit 0 ~ 1: not used
bit 2: '1' indicates trigger signal is ever happened
bit 3: not used
bit 4: '1' indicates M_counter counts to zero status
bit 5: '1' indicates AD duty restore function is enabled
bit 6 ~13: not used
bit 14: '1' indicates SC counter is enabled
bit 15 ~ 17: not used
bit 18: '1' indicates A/D DMA FIFO is empty
bit 19: '1' indicates A/D Pre-FIFO is empty
bit 20: '1' indicates A/D Post-FIFO is empty
bit 21 ~ 31: not used

Return Code NoError, ErrorInvalidCardNumber, ErrorCardNotRegistered

KDIG_AI_ContVScale

Description This function converts the values of an array of acquired binary data from an continuous A/D conversion call to the actual input voltages. The acquired binary data in the reading array might include the channel information (please refer to continuous functions, KDIG_AI_ContReadChannel or KDIG_AI_ContScanChannels, for the detailed data format); however, The calculated voltage values in the voltage array returned will not include the channel message. This function is supported by the following model: KPXI-AI-2-65M

Syntax **Microsoft C/C++ and Borland C++**

```
I16 KDIG_AI_ContVScale (U16 wCardNumber, U16 adRange,
  void *readingArray, F64 *voltageArray, I32 count)
```

Visual Basic

```
KDIG_AI_ContVScale  (ByVal CardNumber As Integer,
  ByVal AdRange As Integer, readingArray As Integer,
  voltageArray As Double, ByVal count As Long) As Integer
```

Parameters **CardNumber:** The card id of the card that want to perform this operation.

AdRange: The analog input range the continuous specified channel is setting. Please refer to AI range codes for the valid range values.

readingArray: Acquired continuous analog input data array

voltageArray: computed voltages array returned

Return Code NoError, ErrorInvalidCardNumber, ErrorCardNotRegistered, ErrorFuncNotSupport, ErrorInvalidAdRange

KDIG_AI_ConvertCheck

Description Check whether the data conversion is completed. This function is supported by the following model: KPXI-AI-2-65M

Syntax **Microsoft C/C++ and Borland C++**

```
I16 KDIG_AI_ConvertCheck (U16 CardNumber, BOOLEAN *Stopped)
```

Visual Basic

```
KDIG_AI_ ConvertCheck (ByVal CardNumber As Integer,
  Stopped As Byte) As Integer
```

Parameters **CardNumber**: The card id of the card that performs the asynchronous operation.

Stopped: Whether the data conversion is completed. If *Stopped* = TRUE, the data conversion operation has stopped. If *Stopped* = FALSE, the AD conversion is not yet complete. (constants TRUE and FALSE are defined in kdigdrvr.h)

Return Code NoError, ErrorInvalidCardNumber, ErrorCardNotRegistered, ErrorFuncNotSupport

KDIG_AI_DMA_Transfer

Description Starts transferring acquired data from on-board SDRAM to the specified buffer. This function is only valid for two steps of data acquisition. If the buffer has a length smaller than the total data count, KDIG-DRVR calls *KDIG_AI_DMA_TransferBySize* internally to fetch the data stored in on-board SDRAM. Please refer to the description of the function *KDIG_AI_DMA_TransferBySize* for the details. This function is supported by the following model: KPXI-AI-2-65M

Syntax **Microsoft C/C++ and Borland C++**

```
Microsoft C/C++ and Borland C++
I16 KDIG_AI_DMA_Transfer (U16 wCardNumber, U16 BufId)
```

Visual Basic 5

```
KDIG_AI_DMA_Transfer (ByVal CardNumber As Integer,
  ByVal BufId As Integer) As Integer
```

Parameters **CardNumber:** The card id of the card that want to be performed this operation.

BufId: The buffer ID (returned from function *KDIG_AI_ContBufferSetup*) of the buffer which will store the data from SDRAM. If the buffer with buffer id of *BufId* has a length smaller than the total data count, you have to call this function repeatedly to fetch the data stored in on-board SDRAM. Please refer to AI data format for the data format in the buffer with *BufId*.

Return Code NoError, ErrorInvalidCardNumber, ErrorCardNotRegistered, ErrorFuncNotSupport

KDIG_AI_DMA_TransferBySize

Description Starts transferring acquired data from on-board SDRAM to the specified buffer. This function is only valid for two steps of data acquisition.

Since the amount of memory can be locked depends on the system, the driver may not lock memory as much as you wish. KDIG-DRVR allows the applications to break the large sized data transfer into several times of smaller data retrievals. This function is supported by the following model: KPXI-AI-2-65M

The code segment for several smaller size of DMA transfer is as follows:

```
[Code Segment]
•••
KDIG_AI_Set_Mode (card, DAQSTEPPED, 1); //two steps data acquisition mode enabled
KDIG_AI_ContBufferSetup (card, ai_buf, locksize, &Id); //a small size of buffer locked
KDIG_AI_ContReadChannel (card, 0, Id, readcnt, 1, 1, ASYNCH_OP);
do {
         KDIG_AI_ConvertCheck(card, &fStop); // check if DAQ conversion is completed
    } while (!fStop);

//DAQ conversion stopped
do {
        //smaller size of data transfer
        KDIG_AI_DMA_TransferBySize (card, 10, Id, locksize, &numRead, &remains, &complete);
    } while(!complete); //check if all the data are read out

KDIG_AI_AsyncClear(card, &startPos, &count);
•••
```

Syntax

Microsoft C/C++ and Borland C++

```
I16 KDIG_AI_DMA_TransferBySize (U16 wCardNumber,
  F32 timeLimit, U16 BufId, U32 ReadCount, U32 *numRead,
  U32 *dataNotTransferred, U8 *complete);
```

Visual Basic

```
KDIG_AI_DMA_TransferBySize (ByVal CardNumber As Integer,
  ByVal timeLimit As Single, ByVal BufId As Integer,
  ByVal ReadCount As Long, numRead As Long,
  dataNotTransferred As Long, complete As Byte) As Integer
```

Parameters **CardNumber:** The card id of the card that want to be performed this operation.

timeLimit: time limit (the unit is in second) for the dma transfer operation. The default value is 1 sec.

BufId: The buffer ID (returned from function *KDIG_AI_ContBufferSetup*) of the buffer which will store the data from SDRAM. If the buffer with buffer id of *BufId* has a length smaller than the total data count, you have to call this function repeatedly to fetch the data stored in on-board SDRAM. Please refer to AI data format for the data format in the buffer with *BufId*.

ReadCount: The ***count of data*** to transfer.

numRead: return the actual ***count of data*** transferred.

dataNotTransferred:
return the ***count of data*** still remain in on-board SDRAM (not be read out).

complete:
1: no data remains (all of the data in on-board SDRAM have been read out)
0: There are data remaining on SDRAM.

Return Code NoError, ErrorInvalidCardNumber, ErrorCardNotRegistered, ErrorFuncNotSupport

KDIG_AI_EventCallBack

Description Controls and notifies the user's application when a specified DAQ event occurs. The notification is performed through a user-specified callback function.

For Windows version, the event message will be removed automatically after calling *KDIG_AI_Async_Clear*. The event message can also be manually removed by set the parameter "*mode*" to be 0. This function is supported by the following model: KPXI-AI-2-65M

Syntax

Microsoft C/C++ and Borland C++

```
I16 KDIG_AI_EventCallBack (U16 CardNumber, I16 mode,
  I16 EventType, U32 callbackAddr)
```

Visual Basic

```
KDIG_AI_EventCallBack (ByVal CardNumber As Integer,
  ByVal mode As Integer, ByVal EventType As Integer,
  ByVal callbackAddr As Long) As Integer
```

Parameters **CardNumber**: The card id of the card that want to be performed this operation.

mode: add or remove the event message.

The valid values:

0: remove
1: add

EventType: event criteria. The valid values are:

DAQEnd: Notification for the completeness of asynchronous analog input operation

DBEvent: Notification for the next half buffer of data in circular buffer is ready for transfer

callbackAddr: the address of the user callback function. KDIG-DRVR calls this function when the specified event occurs. If you wish to remove the event message, set *callbackAddr* to 0.

Return Code NoError, ErrorInvalidCardNumber, ErrorCardNotRegistered, ErrorFuncNotSupport

KDIG_AI_InitialMemoryAllocated

Description This function returns the available memory size for analog input in the device driver in argument *MemSize*. The continuous analog input transfer size can not exceed this size. This function is supported by the following models: KPXI-AI-2-65M

Syntax **Microsoft C/C++ and Borland C++**

```
I16 KDIG_AI_InitialMemoryAllocated (U16 CardNumber,
  U32 *MemSize)
```

Visual Basic

```
KDIG_AI_InitialMemoryAllocated (ByVal CardNumber As Integer,
  MemSize As Long) As Integer
```

Parameters **CardNumber:** The card id of the card that want to perform this operation.

MemSize: The available memory size for continuous AI in device driver of this card. The unit is KB (1024 bytes).

Return Code NoError, ErrorInvalidCardNumber, ErrorCardNotRegistered

KDIG_AI_Set_Mode

Description Configures advanced mode of continuous AI operation. The advanced modes include the following:

-Two steps of data acquisition (The 1st step is data acquired and then stored in SDRAM and the 2nd step is data are transferred from SDRAM to system memory). After the data are acquired, using ***KDIG_AI_DMA_Transfer/ KDIG_AI_DMA_TransferBySize*** to fetch the data stored in on-board SDRAM.

-Re-start continuous AI in kernel with single buffer or dual buffers definitely or indefinitely.

This function is supported by the following model: KPXI-AI-2-65M

Syntax **Microsoft C/C++ and Borland C++**

```
I16 KDIG_AI_Set_Mode (U16 wCardNumber, U16 modeCtrl,
  U16 Iter)
```

Visual Basic

```
KDIG_AI_Set_Mode (ByVal CardNumber As Integer,
  ByVal modeCtrl As Integer, ByVal Iter As Integer)
  As Integer
```

Parameters **CardNumber:** The card id of the card that want to perform this operation.

modeCtrl: The advanced A/D mode.

Valid values:

DAQSTEPPED: two steps of data acquisition (The 1st step is data acquired and then stored in SDRAM and the 2nd step is data are transferred from SDRAM to system memory)

RestartEn: re-start continuous AI in kernel is enabled

DualBufEn: dual buffers mode is enabled

When two constants are used to form the modeCtrl argument, the constants are combined with the bitwise-OR operator(|).

Iter: the number of times to start data acquisition. If the value is zero, the data acquisition will be re-started infinitely

Return Code NoError, ErrorInvalidCardNumber, ErrorCardNotRegistered, ErrorFuncNotSupport

KDIG_AI_Trig_Config

Description Informs KDIG-DRVR library of the trigger source, trigger mode and trigger properties for the KDIG-DRVR device with card ID *CardNumber*.

After the function "KDIG_Register_Card" is called, the device is configured as the following by default:

A/D trigger mode: KDIG_AI_TRGMOD_POST

A/D trigger source: KDIG_AI_TRGSRC_SOFT

A/D trigger polarity: KDIG_AI_TrgNegative

If you wish to perform the device with the default settings, it is not necessary to call this function to make the configuration again. Otherwise, this function has to be called before calling function to perform continuous analog input operation. This function is supported by the following model: KPXI-AI-2-65M

Syntax **Microsoft C/C++ and Borland C++**

```
I16 KDIG_AI_Trig_Config (U16 wCardNumber, U16 trigMode,
  U16 trigSrc, U16 trigPol, U16 anaTrigchan,
  F64 anaTriglevel, U32 postTrigScans, U32 preTrigScans,
  U32 trigDelayTicks, U32 reTrgCnt)
```

Visual Basic

```
KDIG_AI_Trig_Config (ByVal CardNumber As Integer,
  ByVal trigMode As Integer, ByVal TrigSrc As Integer,
  ByVal trigPol As Integer, ByVal anaTrigChan As Integer,
  ByVal anaTriglevel As Double, ByVal postTrigScans As Long,
  ByVal preTrigScans As Long, ByVal trigDelayTicks As Long,
  ByVal reTrgCnt As Long) As Integer
```

Parameters **CardNumber:** The card id of the card that want to perform this operation.

TrigMode: The setting for A/D Trigger control.

KDIG_AI_TRGMOD_POST: Post Trigger Mode (default)
KDIG_AI_TRGMOD_DELAY: Delay Trigger Mode
KDIG_AI_TRGMOD_PRE: Pre-Trigger Mode
KDIG_AI_TRGMOD_MIDL: Middle-Trigger Mode

TrigSrc: The setting for A/D Trigger source.

KDIG_AI_TRGSRC_SOFT: software (default)

KDIG_AI_TRGSRC_ANA: from analog trigger pin

KDIG_AI_TRGSRC_ExtD: from external digital trigger pin

KDIG_AI_TRSRC_SSI_1: trigger event signal is from SSI source

KDIG_AI_TRSRC_SSI_2: both acquisition start and clocked trigger event signal are from SSI

KDIG_AI_TRSRC_PXIStart: trigger event signal are from PXI_START pin (only valid for PXI modules not plugged in slot 2)

TrigPol: The setting for A/D Trigger signal polarity.

KDIG_AI_TrgPositive: Trigger positive edge active (default)

KDIG_AI_TrgNegative: Trigger negative edge active

AnaTrigchan: The setting for Analog Trigger channel.

CH0ATRIG: AI channel 0

CH1ATRIG: AI channel 1

AnaTrigLevel: The setting of Trigger level. The valid range depends on the AI range:

AD_B_5_V: -5V ~ +5V

AD_B_1_V: -1V ~ +1V

postTrigScans: This argument is only valid for Middle trigger.

For ***Middle trigger***, *postTrigScans* indicates the number of data will be accessed after a specific trigger event.

preTrigScans: This argument is only valid for ***Middle trigger*** and ***Pre-Trigger***. The number of scans of data KDIG-DRVR collects before looking for the very first trigger. Setting **preTrigScans** to 0 causes KDIG-DRVR to look for the first trigger as soon as the DAQ process begins.

trigDelayTicks: The number of timer ticks will be ignored after a specific trigger event.

ReTrgCnt: The accepted trigger times in an acquisition. This argument is only valid for Delay trigger and Post trigger mode. The valid range of the value is as follows:

KPXI-AI-2-65M: 1 through 16777215

NOTE *For pre-trigger and middle trigger mode of AI, the ReadScans parameter of continuous AI functions KDIG_AI_ContXXXX has to be equal to postTrigScans+preTrigScans.*

Return Code NoError, ErrorInvalidCardNumber, ErrorCardNotRegistered, ErrorFuncNotSupport

KDIG_AI_VoltScale

This function converts the AD raw data to the actual input voltage. This function is supported by the following model: KPXI-AI-2-65M

Syntax **Microsoft C/C++ and Borland C++**

```
I16 KDIG_AI_VoltScale (U16 CardNumber, U16 AdRange,
  I16 reading, F64 *voltage)
```

Visual Basic

```
KDIG_AI_VoltScale (ByVal CardNumber As Integer,
  ByVal AdRange As Integer, ByVal reading As Integer,
  Voltage As Double) As Integer
```

Parameters **CardNumber:** The card id of the card that want to perform this operation.

AdRange: The analog input range the specified channel is setting. Please refer to AI range codes for the valid range values.

reading: The raw data of the AD Conversion.

voltage: Computed voltage value.

Return Code NoError, ErrorInvalidCardNumber, ErrorCardNotRegistered, ErrorFuncNotSupport, ErrorInvalidAdRange

KDIG_EEPROM_CAL_Constant_Update

Description Save new calibration constants to the specified *bank* of EEPROM. This function is supported by the following model: KPXI-AI-2-65M

Syntax **Microsoft C/C++ and Borland C++**

```
I16 KDIG_EEPROM_CAL_Constant_Update(U16 wCardNumber,
  U16 bank)
```

Visual Basic

```
KDIG_EEPROM_CAL_Constant_Update (
  ByVal wCardNumber As Integer, ByVal bank As Integer)
  As Integer
```

Parameters **CardNumber:** The card id of the card that want to perform this operation.

bank: The storage location on EEPROM. The valid range of the value of bank is 0 through 3.

Return Code NoError, ErrorInvalidCardNumber, ErrorCardNotRegistered, ErrorFuncNotSupport, ErrorInvalidIoChannel

KDIG_Get_SDRAMSize

Description Get the size of onboard SDRAM in unit of MBytes. This function is supported by the following model: KPXI-AI-2-65M

Syntax **Microsoft C/C++ and Borland C++**

```
I16 KDIG_Get_SDRAMSize  (U16 CardNumber, U32 *sdramsize)
```

Visual Basic

```
KDIG_Get_SDRAMSize (ByVal CardNumber As Integer, sdramsize
As Long) As Integer
```

Parameters **CardNumber:** The card id of the card that want to perform this operation.

sdramsize: The size of onboard sdram in unit of MBytes

Return Code NoError, ErrorInvalidCardNumber, ErrorCardNotRegistered, ErrorFuncNotSupport, ErrorInvalidIoChannel

KDIG_GetPXIGeographAddr

Description Get the physical slot number of the slot where the specified PXI module installed in. This function is supported by the following model: KPXI-AI-2-65M

Syntax **Microsoft C/C++ and Borland C++**

```
I16 KDIG_GetPXIGeographAddr (U16 wCardNumber, U8* geo_addr)
```

Visual Basic

```
KDIG_GetPXIGeographAddr (ByVal CardNumber As Integer,
geo_addr As Byte) As Integer
```

Parameters **CardNumber:** The card id of the card that want to perform this operation.

geo_addr: The physical slot number of the slot the module plugged in.

Return Code NoError, ErrorInvalidCardNumber, ErrorCardNotRegistered, ErrorFuncNotSupport, ErrorInvalidIoChannel

KDIG_Load_CAL_Data

Description Load calibration constants from the specified *bank* of EEPROM. This function is supported by the following model: KPXI-AI-2-65M

Syntax **Microsoft C/C++ and Borland C++**

```
I16 KDIG_Load_CAL_Data (U16 CardNumber, U16 bank)
```

Visual Basic

```
KDIG_Load_CAL_Data (ByVal CardNumber As Integer,
  ByVal bank As Integer) As Integer
```

Parameters **CardNumber**: The card id of the card that want to perform this operation.

bank: The storage bank on EEPROM. The valid range of the value of bank is 0 through 3.

Return Code NoError, ErrorInvalidCardNumber, ErrorCardNotRegistered, ErrorFuncNotSupport, ErrorInvalidIoChannel

KDIG_Register_Card

Description Initializes the hardware and software states of a KDIG-DRVR data acquisition card, and then returns a numeric card ID that corresponds to the card initialized. KDIG_Register_Card must be called before any other KDIG-DRVR library functions can be called for that card. The function initializes the card and variables internal to KDIG-DRVR library. Because KDIG-DRVR devices meet the plug-and-play design, the base address (pass-through address) and IRQ level are assigned

by system BIOS directly. This function is supported by the following model: KPXI-AI-2-65M

Syntax **Microsoft C/C++ and Borland C++**

```
I16 KDIG_Register_Card (U16 CardType, U16 card_num)
```

Visual Basic

```
KDIG_Register_Card (ByVal CardType As Integer,
  ByVal card_num As Integer) As Integer
```

Parameters **CardType**: The type of card to be initialized. Keithley will periodically upgrade KDIG-DRVR to add support for new Digitizer data acquisition cards. Please refer to the *Release Notes* for the card types that the current release of KDIG-DRVR actually supports. Following are the constants defined in kdigdrvr.h that represent the Digitizer devices that KDIG-DRVR supports: KPXI_AI_2_65M

card_num: The sequence number of the card with ***the same card type*** (as defined in argument *CardType*) plugged in the PCI/PXI slot. The card sequence number setting is according to the PCI slot sequence in the mainboard. The first card (in the most prior slot) is with card_num=0. For example, if there are two KPXI-AI-2-65M cards plugged on your PC, the KPXI-AI-2-65M card in the prior slot should be registered with card_num=0, and the other one with card_num=1.

Return Code This function returns a numeric card id for the card initialized. The range of card id is between 0 and 31. If there is any error occurs, it will return negative error code, the possible error codes are listed below:

ErrorTooManyCardRegistered, ErrorUnknownCardType, ErrorOpenDriverFailed, ErrorOpenEventFailed

KDIG_Release_Card

Description There are at most 32 cards that can be registered simultaneously. This function is used to tell KDIG-DRVR library that this registered card is not used currently and can be released. This would make room for new card to register. Also by the end of a program, you need to use this function to release all cards that were registered. This function is supported by the following model: KPXI-AI-2-65M

Syntax **Microsoft C/C++ and Borland C++**

```
I16 KDIG_Release_Card (U16 CardNumber)
```

Visual Basic

```
KDIG_Release_Card (ByVal CardNumber As Integer) As Integer
```

Parameters **CardNumber:** The card id of the card that want to be released.

Return Code NoError

KDIG_Route_Signal

Description This function is used to route an internal signal to the SSI line, PXI_START line or the PXI trigger bus line, or to enable clock sharing through the PXI trigger bus line or the PXI_START line. This function is supported by the following model: KPXI-AI-2-65M

Syntax **Microsoft C/C++ and Borland C++**

```
I16 KDIG_Route_Signal (U16 wCardNumber, U16 signal,
  U16 Line, U16 dir)
```

Visual Basic

```
KDIG_Route_Signal (ByVal CardNumber As Integer,
  ByVal signal As Integer, ByVal Line As Integer,
  ByVal dir As Integer) As Integer
```

Parameters **CardNumber**: The card id of the card that want to perform this operation.

signal: The signal is routed to the specified line.

The valid signal codes are as follows:

SSI_TIME: Clock signal

SSI_TRIG_SRC1: trigger event signal

SSI_TRIG_SRC2_S: acquisition start trigger signal output

SSI_TRIG_SRC2_T: clocked trigger signal output

Line: The trigger line to drive.

The valid lines are as follows:

PXI_TRIG_n: PXI trigger bus lines (n is line number and the value is 0 through 7)

PXI_START_TRIG: PXI_START line

TRG_IO: TRG_IO pin

dir: The direction of the connection.

0: receive signal from the connection line.

1: transmit signal to the connection line

Return Code NoError, ErrorInvalidCardNumber, ErrorCardNotRegistered, ErrorFuncNotSupport, ErrorNotStartTriggerModule, ErrorInvalidRouteLine, ErrorInvalidSignalCode, ErrorInvalidSignalDirection

KDIG_Signal_DisConn

Description Disconnects a device signal from the SSI line, PXI_START line or the PXI trigger bus line. This function is supported by the following model: KPXI-AI-2-65M

Syntax **Microsoft C/C++ and Borland C++**

```
I16 KDIG_Signal_DisConn (U16 wCardNumber, U16 signal,
  U16 Line)
```

Visual Basic

```
KDIG_Signal_DisConn (ByVal CardNumber As Integer,
  ByVal signal As Integer, ByVal Line As Integer) As Integer
```

Parameters **CardNumber**: The card id of the card that want to perform this operation.

signal: The specified signal is disconnected from the specified line.

The valid signal codes are as follows:

SSI_TIME: Clock signal

SSI_TRIG_SRC1: trigger event signal

SSI_TRIG_SRC2_S: acquisition start trigger signal output

SSI_TRIG_SRC2_T: clocked trigger signal output

Line: specified the line that is to be disconnected from the signal.

The valid lines are as follows:

PXI_TRIG_n: PXI trigger bus lines (n is line number and the value is 0 through 7)

PXI_START_TRIG: PXI_START line

TRG_IO: TRG_IO pin

Return Code NoError, ErrorInvalidCardNumber, ErrorCardNotRegistered, ErrorFuncNotSupport, ErrorInvalidRouteLine, ErrorInvalidSignalCode, ErrorInvalidSignalDirection

KDIG_SoftTriggerGen

Description Generates a software trigger signal. This function is supported by the following model: KPXI-AI-2-65M

Syntax **Microsoft C/C++ and Borland C++**

```
I16 KDIG_SoftTriggerGen(U16 wCardNumber, U8 op)
```

Visual Basic

```
KDIG_SoftTriggerGen (ByVal CardNumber As Integer,
  ByVal op As Byte) As Integer
```

Parameters **CardNumber**: The card id of the card that want to perform this operation.

op: The operation which waiting for the software trigger signal.

The valid op codes are as follows:

SOFTTRIG_AI: Start or Stop trigger for analog input operation

Return Code NoError, ErrorInvalidCardNumber, ErrorCardNotRegistered, ErrorFuncNotSupport, InvalidCounter

KDIG_SSI_SourceClear

Description Disconnects all of the device signals from the SSI bus trigger, PXI_START line or the PXI trigger bus lines. This function is supported by the following model: KPXI-AI-2-65M

Syntax

Microsoft C/C++ and Borland C++

```
I16 KDIG_SSI_SourceClear (U16 wCardNumber)
```

Visual Basic

```
KDIG_SSI_SourceClear (ByVal CardNumber As Integer)
  As Integer
```

Parameters **CardNumber**: The card id of the card that want to perform this operation.

Return Code NoError, ErrorInvalidCardNumber, ErrorCardNotRegistered, ErrorFuncNotSupport

Status Codes

This section lists status codes returned by KDIG-DRVR (including the name and description).

Each KDIG-DRVR function returns a status code that indicates whether the function was performed successfully. When a KDIG-DRVR function returns a negative number, it means that an error occurred while executing the function.

Table B-2
Status codes returned by KDIG-DRVR

Code	Status Name	Description
0	NoError	No error occurred
-1	ErrorUnknownCardType	The C*ardType* argument is not valid.
-2	ErrorInvalidCardNumber	The CardNumber argument is out of range (larger than 31).
-3	ErrorTooManyCardRegistered	There have been 32 cards that were registered.
-4	ErrorCardNotRegistered	No card registered as id *CardNumber.*
-5	ErrorFuncNotSupport	The function called is not supported by this type of card.
-6	ErrorInvalidIoChannel	The specified *Channel* or *Port* argument is out of range.
-7	ErrorInvalidAdRange	The specified analog input range is invalid.
-8	ErrorContIoNotAllowed	The specified continuous IO operation is not supported by this type of card.
-9	ErrorDiffRangeNotSupport	All the analog input ranges must be the same for multi-channel analog input.
-10	ErrorLastChannelNotZero	The channels for multi-channel analog input must be ended with or started from zero.
-11	ErrorChannelNotDescending	The channels for multi-channel analog input must be contiguous and in descending order.
-12	ErrorChannelNotAscending	The channels for multi-channel analog input must be contiguous and in ascending order.
-13	ErrorOpenDriverFailed	Failed to open the device driver.
-14	ErrorOpenEventFailed	Open event failed in device driver.
-15	ErrorTransferCountTooLarge	The size of transfer is larger than the size of Initially allocated memory in driver.
-16	ErrorNotDoubleBufferMode	Double buffer mode is disabled.
-17	ErrorInvalidSampleRate	The specified sampling rate is out of range.
-18	ErrorInvalidCounterMode	The value of the *Mode* argument is invalid.
-19	ErrorInvalidCounter	The value of the *Ctr* argument is out of range.
-20	ErrorInvalidCounterState	The value of the *State* argument is out of range.
-21	ErrorInvalidBinBcdParam	The value of the *BinBcd* argument is invalid.
-22	ErrorBadCardType	The value of Card Type argument is invalid
-23	ErrorInvalidDaRefVoltage	The value of DA reference voltage argument is invalid
-24	ErrorAdTimeOut	Time out for AD operation
-25	ErrorNoAsyncAI	Continuous Analog Input is not set as Asynchronous mode
-26	ErrorNoAsyncAO	Continuous Analog Output is not set as Asynchronous mode
-27	ErrorNoAsyncDI	Continuous Digital Input is not set as Asynchronous mode
-28	ErrorNoAsyncDO	Continuous Digital Output is not set as Asynchronous mode
-29	ErrorNotInputPort	The value of AI/DI port argument is invalid
-30	ErrorNotOutputPort	The value of AO/DO argument is invalid
-31	ErrorInvalidDioPort	The value of DI/O port argument is invalid
-32	ErrorInvalidDioLine	The value of DI/O line argument is invalid
-33	ErrorContIoActive	Continuous IO operation is not active
-34	ErrorDblBufModeNotAllowed	Double Buffer mode is not allowed
-35	ErrorConfigFailed	The specified function configuration failed
-36	ErrorInvalidPortDirection	The value of DIO port direction argument is invalid
-37	ErrorBeginThreadError	Failed to create thread
-38	ErrorInvalidPortWidth	The port width setting is not allowed
-39	ErrorInvalidCtrSource	The clock source setting is invalid
-40	ErrorOpenFile	Failed to Open file

Table B-2 (continued)
Status codes returned by KDIG-DRVR

Code	Status Name	Description
-41	ErrorAllocateMemory	The memory allocation failed
-42	ErrorDaVoltageOutOfRange	The value of DA voltage argument is out of range
-43	ErrorInvalidSyncMode	The sync. mode of operation is invalid
-44	ErrorInvalidBufferID	The buffer id selected is invalid
-45	ErrorInvalidCNTInterval	The counter value is invalid
-46	ErrorReTrigModeNotAllowed	The Re-Trigger mode of operation is invalid
-47	ErrorResetBufferNotAllowed	The buffer is not allowed to be reset
-48	ErrorAnaTriggerLevel	The value of analog trigger level is invalid
-49	ErrorDAQEvent	The DAQEvent is invalid
-50	ErrorInvalidDataSize	The data size is invalid. For example, the data size of pre-trigger and middle trigger must be a multiple of 32.
-51	ErrorOffsetCalibration	The AD offset calibration failed
-52	ErrorGainCalibration	The AD gain calibration failed
-53	ErrorCountOutofSDRAMSize	The Data count is out of the size of the onboard SDRAM
-54	ErrorNotStartTriggerModule	The module is not installed in a start trigger slot
-55	ErrorInvalidRouteLine	The value of routing line is invalid
-56	ErrorInvalidSignalCode	The value of signal code is invalid
-57	ErrorInvalidSignalDirection	The signal connection direction is invalid
-58	ErrorTRGOSCalibration	The analog trigger level offset calibration failed
-201	ErrorConfigIoctl	The configuration API failed
-202	ErrorAsyncSetIoctl	The async. mode API failed
-203	ErrorDBSetIoctl	The double-buffer setting API failed
-204	ErrorDBHalfReadyIoctl	The half-ready API failed
-205	ErrorContOPIoctl	The continuous data acquisition API failed
-206	ErrorContStatusIoctl	The continuous data acquisition status API setting failed
-207	ErrorPIOIoctl	The polling data API failed
-208	ErrorDIntSetIoctl	The dual interrupt setting API failed
-209	ErrorWaitEvtIoctl	The wait event API failed
-210	ErrorOpenEvtIoctl	The open event API failed
-211	ErrorCOSIntSetIoctl	The COS interrupt setting API failed
-212	ErrorMemMapIoctl	The memory mapping API failed
-213	ErrorMemUMapSetIoctl	The memory Un-mapping API failed
-214	ErrorCTRIoctl	The counter API failed
-215	ErrorGetResIoctl	The resource getting API failed

AI range codes

The analog input range for the digitizers is contained in Table B-3.

Table B-3
Analog input range of digitizers

Input	Range
AD_B_5_V	Bipolar -5V to +5V
AD_B_1_V	Bipolar -1V to +1V

AI data format

This section lists the AI data format for the cards performing analog input operation, as well as the calculation methods to retrieve the A/D converted data and the channel where the data was read (see Table B-4).

Table B-4
AI data format

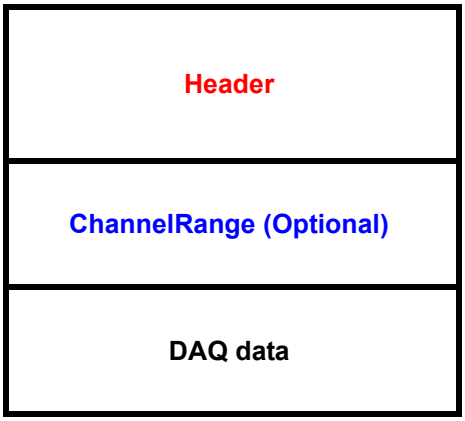
Card Type	Data Format	Value calculation * channel no. (CH#) * A/D converted data (ND) * Value returned from AI function (OD)
KPXI-AI-2-65M	Every 16-bit unsigned integer data: b15 b14 D13 D12 D11 ..... D1D0 where D13,D11, ... , D0 : A/D converted data b14 : Over-voltage indicator	ND = OD & 0x3fff

DATA file format

This paragraph describes the file format of the data files generated by the functions performing continuous data acquisition followed by storing the data to disk.

The data file includes three parts: Header, ChannelRange (optional) and Data block. The file structure is as is Figure B-1.

Figure B-1
DATA file format



Header

The *header* part records the information related to the stored data and its total length is 60 bytes. The data structure of the file header is contained in Table B-5.

Table B-5
Data file header

Header			*Total Length: 60 bytes*
Elements	**Type**	**Size (bytes)**	**Comments**
ID	char	10	file ID *ex. KeithleyD3*
card_type	short	2	card Type *ex. KPXI_AI_2_65M*
num_of_channel	short	2	number of scanned channels *ex. 1, 2*
Channel_no	unsigned char	1	channel number where the data read from (only available as the num_of_channel is 1) *ex. 0, 1*
num_of_scan	long	4	the number of scan for each channel (total count / num_of_channel)
data_width	short	2	the data width 0: 8 bits, 1: 16 bits, 2: 32 bits
channel_order	short	2	the channel scanned sequence 0: normal (ex. 0-1-2-3) 1: reverse (ex. 3-2-1-0) 2: custom* (ex. 0, 1, 3)
ad_range	short	2	the AI range code Please refer to DATA file format *ex. 0 (AD_B_5V)*
scan_rate	double	8	The scanning rate of each channel (total sampling rate / num_of_channel)
num_of_channel_range	short	2	The number of ChannelRange* structure
start_date	char	8	The starting date of data acquisition *ex. 11/11/06*
start_time	char	8	The starting time of data acquisition *ex. 18:30:25*
start_millisec	char	3	The starting millisecond of data acquisition *ex. 360*
reserved	char	6	not used

* If the num_of_channel_range is 0, the ChannelRange block won't be included in the data file.
* The channel_order is set to "custom" only when the card supports variant channel scanning order.

ChannelRange

The *ChannelRange* part records the channel number and data range information related to the stored data. This part consists of several channel and range units. The length of each unit is 2 bytes. The total length depends on the value of *num_of_channel_range* (one element of the file header) and is calculated as the following formula:

Total Length = 2 * num_of_channel_range bytes

The data structure of each ChannelRange unit is contained in Table B-6:

Table B-6
Data structure of ChannelRange unit

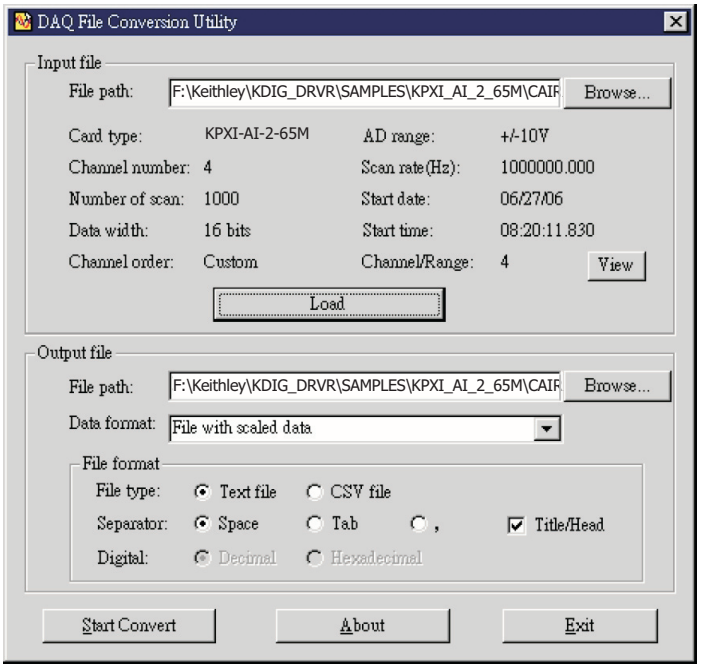
ChannelRange Unit *Length: 2 bytes*			
Elements	**Type**	**Size (bytes)**	**Comments**
channel	char	1	scanned channel number *ex. 0, 1*
range	char	1	the AI range code of *channel* Please refer to DATA file format. *ex. 0 (AD_B_10V)*

Data Block

The last part is the data block. The data is written to file in 16-bit binary format, with the lower byte first (little endian). For example, the value 0x1234 is written to disk with 34 first followed by 12. The total length of the data block depends on the data width and the total data count.

The file is written in Binary format and can't be read in normal text editor. You can use any binary file editor to view it or the functions used for reading files, e.g. fread, to get the file information and data value. KDIG-DRVR provides a useful utility KIDAQCvt for you to convert the binary file. The **KIDAQCvt** main window is shown in Figure B-2.

Figure B-2
DAQ File Conversion Utility



KIDAQCvt first translates the information stored in the header part and the ChannelRange part and then displays the corresponding information in the "Input File" frame of KIDAQCvt main window. After setting the properties (File Path, Format, …etc.) of the converted file and push "*Start Convert*" button in the "Output File" frame, KIDAQCvt gets rid of header and ChannelRange parts and converts the data in data block according to the card type and the data width. Finally, KIDAQCvt writes the converted data to disk. You thus can use any text editor or Excel to view or analyze the accessed data.

Appendix C

KIDAQ®-LabVIEW Compatible Interface Guide

In this appendix:

Introduction to KIDAQ®-LabVIEW

This introduction describes how to program your application in LabVIEW[1] using the Keithley KIDAQ driver.

Overview

Install the KDAQ-DRVR, KDIO-DRVR, or KDIG-DRVR device driver that works with your module before installing the KIDAQ LabVIEW driver. Refer to driver installation information elsewhere in the product manual for the correct driver installation procedure for your module.

KIDAQ LabVIEW VIs (Virtual Instrumentation files) were designed for LabVIEW 6.0 or later. All VIs are stored in 6.0 format. The KIDAQ driver provides a set of VIs that control the KPXI modules from within LabVIEW for fast and simple programming.

To not conflict with the naming of the functions already present in LabVIEW, all KIDAQ LabVIEW VIs have a "**`KI`**" prefix. For example, the Analog Input Read VI is called "**`KI AI Read`**".

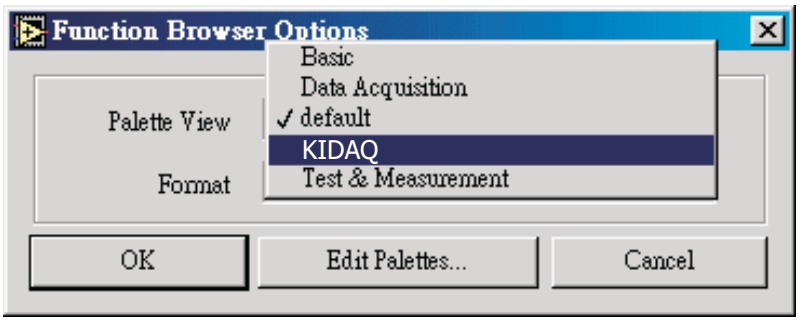
Using KIDAQ LabVIEW VIs in LabVIEW

To use KIDAQ LabVIEW VIs, refer to the following procedure as a guideline (using LabVIEW versions 6.0 through 7.2):

NOTE *LabVIEW 8 (and later versions) uses a new interface. In LabVIEW 8, the KIDAQ VI set will appear at the bottom of the LabVIEW function palette. To personalize your function palette, click the* ***Tools menu*** *item, select* ***Advanced*** *and* ***edit palette set...*** *from the menus.*

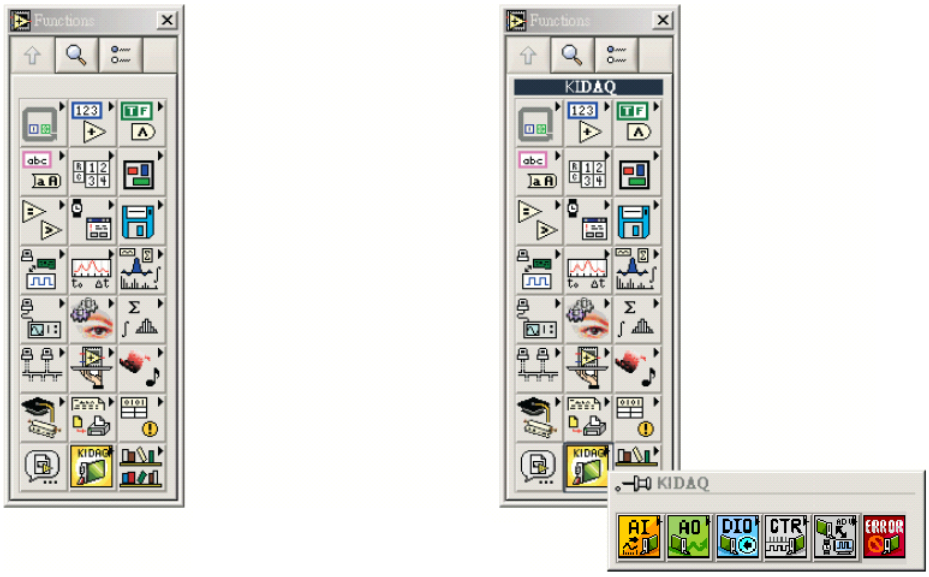
1. Click the **Options** button in the **Controls** or **Functions** palette toolbar to display the **Function Browser Options** dialog box.
2. Select **KIDAQ** view from the **Palette Set** pull-down menu (Figure C-1).
3. Click **OK**. The **Functions** palettes change to the ***KIDAQ*** view.
4. Then, find KIDAQ LabVIEW VIs in **KIDAQ** icon on the **Functions** palette (Figure C-2).

Figure C-1
Function Browser Options



1. LabVIEW™ is a trademark of the National Instruments Corporation. All other trademarks are the property of their respective owners.

Figure C-2
Functions palette



The **KIDAQ** palette contains four sub-palettes that contain the different classes of data acquisition VIs. The VIs are classified as follows:

- Analog Input VIs
- Analog Output VIs
- Digital I/O VIs
- Timer/Counter VIs
- Calibration and Configuration VIs
- Error Handler VI

Most of the VI sub-palettes arrange the VIs in different levels, Easy, Intermediate, or Advanced, according to their functionality.

KIDAQ LabVIEW Programming

The KIDAQ LabVIEW VIs Overview briefly describes each VI in KIDAQ LabVIEW. All applications developed with KIDAQ LabVIEW are compatible across Windows XP and 2000. For detailed function information, refer to Appendix D, the KIDAQ®-LabVIEW Compatible Function Reference.

You can find the detailed description of each VI using any of the following ways:

- Select the **Show Help** command in the **Help** menu in LabVIEW. Then, when you put the mouse cursor on KIDAQ LabVIEW VI, LabVIEW will show the description of the VI.
- Refer to Appendix D of this User's Manual.
- Contact Keithley Instruments via phone, email, or on the web at www.keithley.com for further information.

Device Driver Handling

Device Driver Handling describes how to configure the KIDAQ PXI cards Windows® XP/2000 device driver.

Windows XP/2000 Device Driver

Once Windows XP/2000® has started, the Plug and Play function of Windows XP/2000® operating system will find the new Keithley PXI cards. If this is the first time to install Keithley PXI cards in your Windows XP/2000® system, you will be informed to install the device driver. Refer to driver installation information elsewhere in the product manual for the correct driver installation procedure for your module.

Driver Utility

NOTE *The KDAQ-DRVR, KDIO-DRVR, or KDIG-DRVR device driver should be installed before the KDAQ LabVIEW driver. Refer to driver installation information elsewhere in the product manual for the correct driver installation procedure for your module.*

KIDAQ LabVIEW provides a PXI Configuration Utility (*configdrv.exe*). These utilities are used to ***set/change*** the allocated buffer sizes of AI, AO, DI and DO (Analog Input, Analog Output, Digital Input, Digital Output). The allocated buffer sizes of AI, AO, DI, DO represent the sizes of contiguous Initially Allocated memory for continuous analog input, analog output, digital input, digital output respectively. Its unit is page *KB*, i.e. 1024 bytes. The device driver will try to allocate these sizes of memory at system startup time. If this size of memory is not available, the driver will allocate as much memory as system can provide. The size of initially allocated memory is the maximum memory size that DMA or Interrupt transfer can be performed. It will induce an unexpected result in that DMA or Interrupt transfer performed exceeds the initially allocated size.

KIDAQ Utilities

This section, KIDAQ Utilities, describes all utilities included in the KIDAQ software.

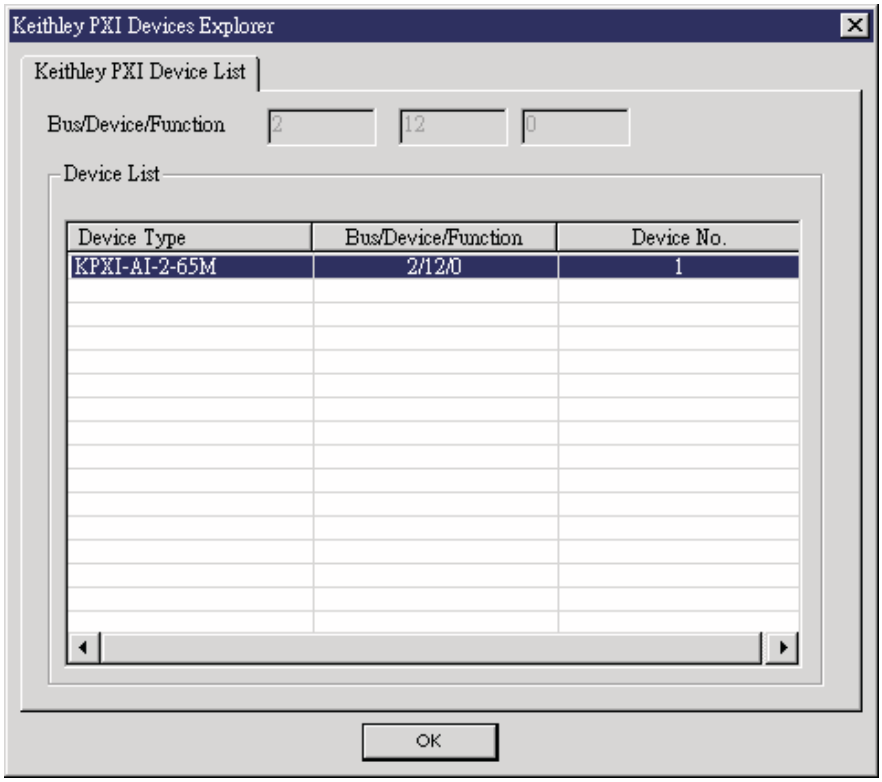
KIDAQ Registry/Configuration utility

configdrv is used to modify the allocated buffer sizes of AI, AO, DI and DO (Windows® XP/2000). The default installation directory for this utility is **C:\Keithley\KIDAQ\Util**. It can also be found in the start menu under **Programs** -> **Keithley** -> **KIDAQ LabVIEW Driver** -> **Configuration Utility**. For detailed information on this utility, refer to device driver guide for you module.

KIDAQ Devices Explorer

Device Browser (KPXIConf.exe) displays the currently installed and detected KIDAQ hardware. The default installation directory for this utility is **C:\Keithley\KIDAQ\Util**. It can also be found in the start menu under **Programs** -> **Keithley** -> **KIDAQ LabVIEW Driver** -> **Device Explorer**. The *Keithley PXI Devices Explorer* main window is shown in Figure C-3:

Figure C-3
Keithley PXI Devices Explorer



The *Device Browser* main window contains three columns, *Device Type*, *Location (Bus/Device/Function)* and *Device Number*.

Device Type: Type of KIDAQ board installed

Location (Bus/Device/Function): The location the device is plugged into

Device Number: Number of device at PXI bus (Starts from 1)

Using this utility, user can view all of the KIDAQ devices connected to your system and get the device number corresponding to the device plugged on a specified PXI slot.

KIDAQ LabVIEW VIs Overview

This section briefly describes each VI in the KIDAQ LabVIEW driver. The setup program detects the system (Windows® XP/2000), and installs the correct platform drivers to the system. All applications developed with KIDAQ LabVIEW are compatible across Windows® XP/2000.

You can find the detailed description of each VI using any of the following ways:

- Select the **Show Help** command in the **Help** menu in LabVIEW. Then, when you put the mouse cursor on KIDAQ LabVIEW VI, LabVIEW will show the description of the VI.
- Refer to Appendix D of this document
- Contact Keithley Instruments via phone, email, or on the web at www.keithley.com for further information

KIDAQ LabVIEW VIs are grouped into the following LabVIEW palettes:

- **Analog Input VIs**
- **Analog Output VIs**
  - Advanced Analog Output VIs

- **Digital I/O VIs**
  - Advanced Digital I/O
- **Timer/Counter VIs**
  - Intermediate Timer/Counter VIs
  - Advanced Timer/Counter VIs
- **Calibration and Configuration VIs**
- **Error Handler VI**

Analog Input VIs

KI AI Acquire Waveform: Acquires a specified number of samples at a specified sample rate from a single input channel and returns the acquired data.

KI AI Acquire Waveforms: Acquires data from the specified channels and samples the channels at the specified scan rate.

KI AI Sample Channel: Measures the signal attached to the specified channel and returns the measured data.

KI AI Sample Channels: Performs a single reading from each of the specified channels.

KI AI Clear: The KI AI Clear VI stops an acquisition associated with task ID in.

KI AI Config: Configures an analog input operation for a specified set of channels.

KI AI Read: Reads data from a buffered data acquisition.

KI AI Single Scan: Returns one scan of data directly from the board analog input channels for a non-buffered acquisition.

KI AI Start: Starts a buffered analog input operation.

Analog Output VIs

KI AO Generate Waveform: Generates a timed and buffered waveform for the given output channel at the specified update rate.

KI AO Generate Waveforms: Generates timed and buffered waveforms for the given output channels at the specified update rate.

KI AO Update Channel: Writes a specified value to an analog output channel.

KI AO Update Channels: Writes values to each of the specified analog output channels.

KI AO Clear: The KI AO Clear VI stops an analog output generation associated with task ID.

KI AO Config: Configures a buffered analog output operation.

KI AO Start: Starts a buffered analog output operation.

KI AO Wait: waits until the waveform generation of the task completes before returning.

KI AO Write: writes data into the buffer for a buffered analog output operation.

Advanced Analog Output VIs

KI AO Trigger and Gate Config: Configures the trigger conditions for analog output operations.

Digital I/O VIs

KI Read from Digital Line: Reads the logical state of a digital line on a digital channel that you configure.

KI Read from Digital Port: Reads a digital channel that you configure.

KI Write to Digital Line: Sets the output logic state of a digital line to high or low on a digital channel that you specify.

KI Write to Digital Port: Outputs a decimal pattern to a digital channel that you specify.

KI DIO Clear: Stops an acquisition associated with task ID.

KI DIO Config: Creates the taskID, establishes the handshake parameters, and allocates a buffer to hold the scans.

KI DIO Read: Calls the VI to read data from the internal transfer buffer and returns the data read in pattern.

KI DIO Start: Starts a buffered digital I/O operation.

KI DIO Write: Writes digital output data to the internal transfer buffer.

Advanced Digital I/O VIs

KI DIO Port Config: Configures a digital Channel and returns a taskID to be used with Port VIs.

Timer/Counter VIs

KI Count Events or Time: Configures one or two counters to count external events.

KI Generate Delayed Pulse: Configures and starts a counter to generate a single pulse with the specified delay and pulse-width.

KI Generate Pulse Train: Configures the specified counter to generate a continuous pulse-train.

KI Measure Pulse Width or Period: Measures the pulse-width (length of time a signal is high or low) or period (length of time between adjacent rising or falling edges) of a TTL signal.

Intermediate Timer/Counter VIs

KI Continuous Pulse Generator Config: Configures a counter to generate a continuous TTL pulse-train.

KI Counter Divider Config: Configures the specified counter to divide a signal.

KI Counter Read: Reads the counter or counters identified by task ID.

KI Counter Start: Starts the counters identified by task ID.

KI Counter Stop: Stops a count operation immediately or conditionally on an input error.

KI Delayed Pulse Generator Config: Configures a counter to generate a single pulse with the specified delay and pulse-width.

KI Down Counter or Divider Config: Configures the specified counter to count down or divide a signal.

KI Event or Time Counter Config: Configures one or two counters to count external events.

KI UpDown Counter Config: Configures one counter to count edges in the signal on the specified counter's SOURCE pin or the number of cycles of a specified internal timebase signal.

Advanced Timer/Counter VIs

KI ICTR Control: This VI control counters on the KIDAQ devices that use 82C54 chip.

Calibration and Configuration VIs

KI KPXI-DAQ Series Calibrate and Digitizer Series calibrate: calibrates Keithley PXI DAQ device.

KI Route Signal: routes an internal signal to the specified I/O connector or SSI bus line, or to enable clock sharing through the SSI bus clock line.

KI SSI Control: Connects or disconnects trigger and timing signals between DAQ devices along the Real-Time System Integration (SSI) bus.

Error Handler VI

KI Error Handler: explains non-zero error codes and shows dialog box with information about error.

Distribution of Applications

To install an application using KIDAQ LabVIEW on another computer, you also must install the necessary driver files and supporting libraries on the target machine. You can create an automatic installer to install your program and all of the files needed to run that program or you can manually install the program and program files. Whichever installation method you choose, you must install the following files:

NOTE *Do not replace any files on the target computer if the file on the target computer has a newer version than the file you are installing.*

Windows XP/2000

LLB files

kidaq_pci.llb in **C:\Keithley\KI-DAQ\LLB**

Required support DLLs

Pci-lv.dll in **C:\Windows\system32**. This file should be copied to the same system32 directory on the target machine. On Windows 2000 the Windows directory is named winnt instead of Windows.

Driver files

The corresponding driver files in **C:\Windows\system32\drivers**, e.g. **ksdaq4M2.sys** for **KPXI-SDAQ-4-2M**. These files should be copied to:

- **Windows\system32\drivers** directory (for Windows XP).
- **Winnt\system32\drivers** directory (for Windows 2000).

The corresponding INF file in **\Windows\inf**, e.g. **ksdaq4M2.inf** for **KPXI-SDAQ-4-2M**. These files should be copied to:

- **Windows\inf** directory (for Windows XP).
- **Winnt\inf** directory (for Windows 2000).

The location of the device configuration utility is: **C:\Keithley\KI-DAQ\Util\configdrv**

Appendix D

KIDAQ®-LabVIEW Compatible Function Reference

In this appendix:

Introduction

This function reference provides a detailed description of LabVIEW[1] compatible interfaces for Keithley Instruments PXI DAQ modules.

Hardware support

Keithley Instruments will periodically upgrade KIDAQ LabVIEW to add support for new Keithley Instruments PXI data acquisition modules. This release of KIDAQ LabVIEW supports the following hardware:

KPXI-DIO series:

- **KPXI-DIO-16-16**: 32 channels isolated Digital I/O card
- **KPXI-DIO-48**: 48-bit digital I/O card
- **KPXI-RDI-8-16**: 8 relay output and 16 isolated input card
- **KPXI-DIO-32-80M**: 40 Mbytes/sec Ultra-high speed 32 channels digital I/O module with bus mastering DMA transfer supporting scatter gather technology
- **KPXI-DIO-32-32**: 32 isolated channels DI & 32 isolated channels DO card
- **KPXI-DIO-64-0**: 64 isolated channels DI card
- **KPXI-DIO-0-64**: 64 isolated channels DO card

KPXI-DAQ series:

- **KPXI-SDAQ-4-2M**: 2MHz 4 channels simultaneous A/D and 2 channels D/A output device with bus mastering DMA transfer capability
- **KPXI-SDAQ-4-500K**: 500kHz 4 channels simultaneous A/D and 2 channels D/A output device with bus mastering DMA transfer capability
- **KPXI-DAQ-64-3M**: 3MHz 64 channels multiplexed A/D and 2 channels D/A output device with bus mastering DMA transfer capability
- **KPXI-DAQ-64-500K**: 500kHz 64 channels multiplexed A/D and 2 channels D/A output device with bus mastering DMA transfer capability
- **KPXI-DAQ-64-250K**: 250kHz 64 channels multiplexed A/D and 2 channels D/A output device with bus mastering DMA transfer capability
- **KPXI-DAQ-96-3M**: 3MHz 96 channels multiplexed A/D device with bus mastering DMA transfer capability
- **KPXI-AO-4-1M**: High Performance 4 channels analog output Multi-function device with bus mastering DMA transfer capability
- **KPXI-AO-8-1M**: High Performance 8 channels analog output Multi-function device with bus mastering DMA transfer capability

Digitizer series:

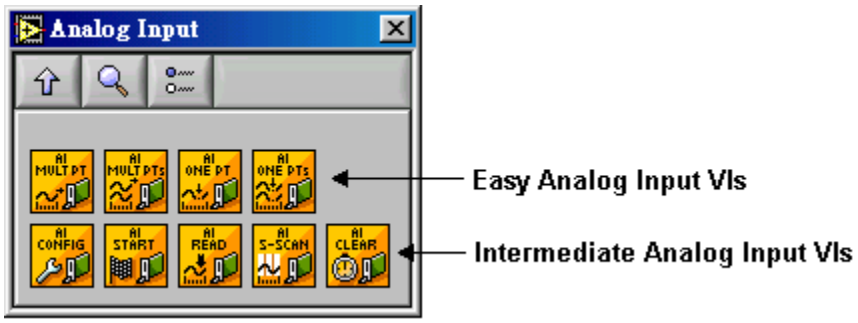
- **KPXI-AI-2-65M**: 130MHz or 2 channels simultaneous A/D digitizer with bus mastering DMA transfer capability

1. LabVIEW™ is a trademark of the National Instruments Corporation. All other trademarks are the property of their respective owners.

Analog input VIs

Analog Input VIs (virtual instruments) are available in the Analog Input palette (Figure D-1).

Figure D-1
Analog input palette



Easy analog input VIs

KI AI acquire waveform

This VI acquires a specified number of samples at a specified sample rate from a single input channel and returns the acquired data. This VI performs a timed measurement of a waveform on a single analog input channel. If an error occurs, a dialog box appears providing error information.

Table D-1
KI AI acquire waveform

I16	**device:** Number of the device (beginning from 1). The utility ***Device Browser*** can be used to get the information of current device configuration.
U16	**sub type:** is the sub-type of the device you assigned to the Keithley Instruments PXI device during configuration.
abc	**channel:** identifies the analog input channel you want to measure. The default input is channel 0. The valid channel for each Keithley Instruments PXI device is as follows: • **KPXI-SDAQ-4-2M**: 0 through 3 • **KPXI-SDAQ-4-500K**: 0 through 3 • **KPXI-DAQ-64-3M**: 0 through 63 • **KPXI-DAQ-64-500K**: 0 through 63 • **KPXI-DAQ-64-250K**: 0 through 63 • **KPXI-DAQ-96-3M**: 0 through 95 • **KPXI-AO-4-1M**: 0 through 7 • **KPXI-AO-8-1M**: 0 through 3 • **KPXI-AI-2-65M**: 0 through 1
I32	**number of samples:** is the number of samples the VI acquires. The default for this parameter is 1000 samples, except KPXI-AI-2-65M. For KPXI-AI-2-65M, the default value is 1024.
SGL	**sample rate:** is the requested number of samples per second for the analog input. The default for this parameter is a rate of 1000 samples/second.

Table D-1 (continued)
KI AI acquire waveform

SGL	**high limit:** is the maximum scaled data in Volts. The default input is 0. If both high limit and low limit are 0, high limit keeps the default settings.
SGL	**low limit:** is the minimum scaled data in Volts. The default input is 0. If both high limit and low limit are 0, low limit keeps the default settings.
U16	**input config:** defines the mode that the channel should be scanned. 0: No change (default input) 1: Differential (default setting) 2: Referenced single-ended 3: Nonreferenced single-ended
[SGL]	**waveform:** contains scaled analog input data.
SGL	**actual sample period:** is the actual interval between samples, which is the inverse of the actual sample rate. The actual sample period can differ from the requested sample rate, depending on the capabilities of the hardware.

KI AI acquire waveforms

Acquires data from the specified channels at the specified scan rate. This VI performs a timed measurement of multiple waveforms on the specified analog input channels. If an error occurs, a dialog box appears, giving you the error information.

Table D-2
KI AI acquire waveforms

I16	**device:** Number of the device (beginning from 1). The utility **Device Browser** can be used to get the information of current device configuration.
U16	**sub type:** is the sub-type of the device you assigned to the Keithley Instruments PXI device during configuration. Only the following series of devices need to specify the sub type.

Table D-2 (continued)
KI AI acquire waveforms

[abc]	**channels:** specifies the set of analog input channels you want to measure. The order of the channels in the scan list defines the order in which the channels are scanned. If x, y, and z refer to channels, you can specify a list of channels in a single element by "x,y,z". If x refers to the first channel in a consecutive channel range and y refers to the last channel, you can specify the range by "x:y". The default input is channel 0. The valid channel order for acquiring data is as follows: **KPXI-SDAQ-4-2M**: numbers in channels must be within 0 and 3 and the continuous scan sequence is ascending with consecutive channels. **KPXI-SDAQ-4-500K**: numbers in channels must be within 0 and 3 and the continuous scan sequence is ascending with consecutive channels. **KPXI-DAQ-64-3M**: numbers in channels must be within 0 and 63 and there is no restriction of channel order setting; therefore you can set the channel order as you wish. **KPXI-DAQ-64-500K**: numbers in channels must be within 0 and 63 and there is no restriction of channel order setting; therefore you can set the channel order as you wish. **KPXI-DAQ-64-250K**: numbers in channels must be within 0 and 63 and there is no restriction of channel order setting; therefore you can set the channel order as you wish. **KPXI-DAQ-96-3M**: numbers in channels must be within 0 and 95 and there is no restriction of channel order setting; therefore you can set the channel order as you wish. **KPXI-AO-4-1M**: numbers in channels must be within 0 and 3 and the continuous scan sequence is ascending with consecutive channels. **KPXI-AO-8-1M**: numbers in channels must be within 0 and 7 and the continuous scan sequence is ascending with consecutive channels. **KPXI-AI-2-65M**: numbers in channels must be within 0 and 1 and the continuous scan sequence is ascending with consecutive channels.
I32	**number of samples/ch:** is the number of samples per channel. The default is 1000 samples/ch, except KPXI-AI-2-65M. For KPXI-AI-2-65M, the default value is 1024 samples/ch.
SGL	**scan rate:** is the requested number of scans per second. The default is 1000 scans/s. A scan is one sample/channel.
SGL	**high limit:** is the maximum scaled data in Volts. The default input is 0. If both high limit and low limit are 0, high limit keeps the default settings.
SGL	**low limit:** is the minimum scaled data in Volts. The default input is 0. If both high limit and low limit are 0, low limit keeps the default settings.
U16	**input config:** defines the mode that the channel should be scanned. 0: No change (default input) 1: Differential (default setting) 2: Referenced single-ended 3: Nonreferenced single-ended
[SGL]	**waveforms:** is a 2D array that contains analog input data in Volts.
SGL	**actual scan period:** is the actual interval between scans, which is the inverse of the actual scan rate. The actual scan period can differ from the requested scan rate, depending on the capabilities of the hardware.

KI AI sample channel

This VI performs a single, un-timed measurement of a channel. It measures the signal attached to the specified channel and returns the measured data (in Volts). If an error occurs, a dialog box appears, giving you the error information.

Table D-3
KI AI sample channel

I16	**device:** Number of the device (beginning from 1). The utility **Device Browser** can be used to get the information of current device configuration.
U16	**sub type:** is the sub-type of the device you assigned to the Keithley Instruments PXI device during configuration.
abc	**channel:** identifies the analog input channel you want to measure. The default input is channel 0. The valid channel for each Keithley Instruments PXI device is as follows: **KPXI-SDAQ-4-2M**: 0 through 3 **KPXI-SDAQ-4-500K**: 0 through 3 **KPXI-DAQ-64-3M**: 0 through 63 **KPXI-DAQ-64-500K**: 0 through 63 **KPXI-DAQ-64-250K**: 0 through 63 **KPXI-DAQ-96-3M**: 0 through 95 **KPXI-AO-4-1M**: 0 through 3 **KPXI-AO-8-1M**: 0 through 7
SGL	**high limit:** is the maximum scaled data in Volts. The default input is 0. If both high limit and low limit are 0, high limit keeps the default setting.
SGL	**low limit:** is the minimum scaled data in Volts. The default input is 0. If both high limit and low limit are 0, low limit keeps the default setting.
U16	**input config:** defines the mode that the channel should be scanned. 0: No change (default input) 1: Differential (default setting) 2: Referenced single-ended 3: Nonreferenced single-ended
SGL	**sample:** contains the scaled analog input data for the specified channel.

KI AI sample channels

This VI measures a single value from each of the specified analog input channels. If an error occurs, a dialog box appears, giving you the error information.

Table D-4
KI AI sample channels

I16	**device:** Number of the device (beginning from 1). The utility **Device Browser** can be used to get the information of current device configuration.
U16	**sub type:** is the sub-type of the device you assigned to the Keithley Instruments PXI device during configuration.

Table D-4 (continued)
KI AI sample channels

abc	**channels:** specifies the set of analog input channels you want to measure. The order of the channels in the scan list defines the order in which the channels are scanned. If x, y, and z refer to channels, you can specify a list of channels in a single element by "x,y,z". If x refers to the first channel in a consecutive channel range and y refers to the last channel, you can specify the range by "x:y". The default input is channel 0. The valid channel order for acquiring data is as follows: **KPXI-SDAQ-4-2M**: numbers in channels must be within 0 and 3 and the continuous scan sequence is ascending with consecutive channels. **KPXI-SDAQ-4-500K**: numbers in channels must be within 0 and 3 and the continuous scan sequence is ascending with consecutive channels. **KPXI-DAQ-64-3M**: numbers in channels must be within 0 and 63 and there is no restriction of channel order setting; therefore you can set the channel order as you wish. **KPXI-DAQ-64-500K**: numbers in channels must be within 0 and 63 and there is no restriction of channel order setting; therefore you can set the channel order as you wish. **KPXI-DAQ-64-250K**: numbers in channels must be within 0 and 63 and there is no restriction of channel order setting; therefore you can set the channel order as you wish. **KPXI-DAQ-96-3M**: numbers in channels must be within 0 and 95 and there is no restriction of channel order setting; therefore you can set the channel order as you wish. **KPXI-AO-4-1M**: numbers in channels must be within 0 and 3 and the continuous scan sequence is ascending with consecutive channels. **KPXI-AO-8-1M**: numbers in channels must be within 0 and 7 and the continuous scan sequence is ascending with consecutive channels.
SGL	**high limit:** is the maximum scaled data in Volts. The default input is 0. If both high limit and low limit are 0, high limit keeps the default setting.
SGL	**low limit:** is the minimum scaled data in Volts. The default input is 0. If both high limit and low limit are 0, low limit keeps the default setting.
U16	**input config:** defines the mode that the channel should be scanned. 0: No change (default input) 1: Differential (default setting) 2: Referenced single-ended 3. Nonreferenced single-ended
[SGL]	**sample:** is a 1D array that contains scaled analog input data.

Intermediate analog input VIs

KI AI clear

This VI stops an acquisition operation. Before beginning a new acquisition, you must call the KI AI Config VI.

Table D-5
KI AI clear

U32	**taskID in:** identifies the group and the I/O operation.
	error in (no error): describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the error in cluster in error out.

Table D-5 (continued)
KI AI clear

		**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
		**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
		**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.
	**taskID out:** has the same value as taskID in.	
	**error out:** contains error information. If the error in cluster indicated an error, the error out cluster contains the same information. Otherwise, error out describes the error status of this VI.	
		**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
		**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
		**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.

KI AI config

Configures a buffered analog input operation, including configuring the hardware and allocating a buffer.

Table D-6
KI AI config

SGL	**interchannel delay:** For devices with both scan and channel clocks (KPXI-DAQ series devices), you can use interchannel delay to specify the waiting time between sampling channels within a scan. Select a default interchannel delay automatically, giving the hardware time to settle between channels. The default value for interchannel delay is -1.0, which tells the AI Config VI to use the channel clock rate LabVIEW selects.	
U16	**sub type:** is the sub-type of the device you assigned to the Keithley Instruments PXI device during configuration.	
[ ]	**measurement mode structure:** This input is not used by Keithley Instruments PXI devices and is ignored.	
	U16	**measurement mode:** is not used
	U32	**reserved:** is not used
[ ]	**coupling & input config:** is an array of clusters. Each array element contains the configuration for the channel or channels specified by the corresponding element of the channels array. KIDAQ LabVIEW uses only input config. The default for the coupling & input config array is an empty array, which means the parameters keep their default settings.	
	U16	**coupling:** This input is not used by Keithley Instruments PXI devices and is ignored.
	U16	**input config:** defines the mode that the channel should be scanned. 0: No change (default input) 1: Differential (default setting) 2: Referenced single-ended 3: Nonreferenced single-ended
[ ]	**input limits:** is an array of clusters. Each array element contains the expected signal limits for the channels specified by the corresponding element of channels. If there are fewer elements in this array than in channels, the VI uses the last array element for the rest of the channels. The default for the input limits array is an empty array, which means the input limits keep their default settings.	
	SGL	**high limit:** is the maximum scaled data in Volts. The default input is 0.
	SGL	**low limit:** is the minimum scaled data in Volts. The default input is 0.
I16	**device:** Number of the device at PXI-bus (beginning from 1). The utility **Device Browser** can be used to get the information of current device configuration.	

Table D-6 (continued)
KI AI config

[abc]	**channels:** specifies the set of analog input channels. The order of the channels in the scan list defines the order in which the channels are scanned. channels is an array of strings. You can use one channel entry per element or specify the entire scan list in a single element, or use any combination of these two methods. If x, y, and z refer to channels, you can specify a list of channels in a single element by "x,y,z". If x refers to the first channel in a consecutive channel range and y refers to the last channel, you can specify the range by "x:y". The valid channel order for acquiring data is as follows: **KPXI-SDAQ-4-2M**: numbers in channels must be within 0 and 3 and the continuous scan sequence is ascending with consecutive channels. **KPXI-SDAQ-4-500K**: numbers in channels must be within 0 and 3 and the continuous scan sequence is ascending with consecutive channels. **KPXI-DAQ-64-3M**: numbers in channels must be within 0 and 63 and there is no restriction of channel order setting; therefore you can set the channel order as you wish. **KPXI-DAQ-64-500K**: numbers in channels must be within 0 and 63 and there is no restriction of channel order setting; therefore you can set the channel order as you wish. **KPXI-DAQ-64-250K**: numbers in channels must be within 0 and 63 and there is no restriction of channel order setting; therefore you can set the channel order as you wish. **KPXI-DAQ-96-3M**: numbers in channels must be within 0 and 95 and there is no restriction of channel order setting; therefore you can set the channel order as you wish. **KPXI-AO-4-1M**: numbers in channels must be within 0 and 7 and the continuous scan sequence is ascending with consecutive channels. **KPXI-AO-8-1M**: numbers in channels must be within 0 and 3 and the continuous scan sequence is ascending with consecutive channels. **KPXI-AI-2-65M**: numbers in channels must be within 0 and 1 and the continuous scan sequence is ascending with consecutive channels.	
I32	**buffer size:** is the number of scans you want the buffer to hold. The default for this parameter is 1000 scans, except KPXI-AI-2-65M. For KPXI-AI-2-65M, the default value is 1024 scans.	
I16	**group:** is the number, from 0 to 15, that you assign to the specified set of channels. The default input and setting for group is 0. If you only have one acquisition for this device, leave this input unwired and use group 0.	
	error in (no error): describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the error in cluster in error out.	
	TF	**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
	I32	**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.

Table D-6 (continued)
KI AI config

		**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.
	**number of buffers:** This input is not used by Keithley Instruments PXI devices and is ignored.	
	**allocation mode:** This input is not used by Keithley Instruments PXI devices and is ignored.	
	**number of AMUX boards:** This input is not used by Keithley Instruments PXI devices and is ignored.	
	**taskID:** identifies the group and the I/O operation.	
	**number of channels:** is the total number of channels in the group.	
	**error out:** contains error information. If the error in cluster indicated an error, the error out cluster contains the same information. Otherwise, error out describes the error status of this VI.	
		**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
		**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
		**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.

KI AI Read

This VI reads specified number of scans of data from a buffered analog input acquisition.

KI AI Read is a polymorphic VI that you can configure to output the following kinds of data:

- 2-byte Binary Array (KPXI-AI-2-65M and KDAQ-DRVR series devices)(Table D-7)
- Scaled and 2-byte Binary Arrays (KPXI-AI-2-65M and KDAQ-DRVR series devices)(Table D-8)
- Scaled Array (Table D-9)

Table D-7
2-byte binary array

	conditional retrieval specification: This input is not used by Keithley Instruments PXI devices and is ignored.	
	U16	**mode:** This input is not used by Keithley Instruments PXI devices and is ignored.
	I32	**channel index:** This input is not used by Keithley Instruments PXI devices and is ignored.
	U16	**slope:** This input is not used by Keithley Instruments PXI devices and is ignored.
	SGL	**level:** This input is not used by Keithley Instruments PXI devices and is ignored.
	SGL	**hysteresis:** This input is not used by Keithley Instruments PXI devices and is ignored.
	I32	**skip count:** This input is not used by Keithley Instruments PXI devices and is ignored.
	I32	**offset:** This input is not used by Keithley Instruments PXI devices and is ignored.
U32	**taskID in:** identifies the group and the I/O operation.	
I32	**number of scans to read:** is the number of scans the VI is to retrieve from the acquisition buffer. The default input is -1, which set number of scans to read equal to the value of the number of scans to acquire parameter when the KI AI Start was called. If number of scans to read is -1 and number of scans to acquire is 0, KIDAQ LabVIEW sets number of scans to read to be the half of the buffer size.	
SGL	**time limit in sec:** is the time limit for the read operation. The default input is -1.0, which means KIDAQ LabVIEW calculates a time limit based on the value of number of scans to read and the scan rate. If the scan rate is unknown, the VI uses 1 second as the time limit. The resolution of the timeout clock is about 55 ms.	
	error in (no error): describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the error in cluster in error out.	
	TF	**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.

Table D-7 (continued)
2-byte binary array

	I32	**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
	abc	**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.
		read/search position: This input is not used by Keithley Instruments PXI devices and is ignored. The starting point for the read is the position where the read mark points to. Initially, the read mark points to the beginning of the acquisition buffer. As you retrieve data from the buffer using this VI, KIDAQ LabVIEW increments the read mark to point to the next block of data to be read.
	U16	**position:** This input is not used by Keithley Instruments PXI devices and is ignored.
	I32	**read offset:** This input is not used by Keithley Instruments PXI devices and is ignored.
U32		**scan backlog:** is the amount of data remaining in the buffer after this VI completes.
I32		**number read:** is the number of scans returned. This number is identical to number of scans to read unless an error or timeout appears or the VI reaches the end of the data.
U32		**taskID out:** has the same value as ***taskID in***.
[I16] or I32		**binary data:** is a 2D array that contains unscaled analog input data.
TF		**retrieval complete:** is TRUE when the acquisition finishes and no backlog data remains.
		error out: contains error information. If the error in cluster indicated an error, the error out cluster contains the same information. Otherwise, error out describes the error status of this VI.
	TF	**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
	I32	**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
	abc	**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.

Table D-8
Scaled and Binary Arrays

	**conditional retrieval specification:** This input is not used by Keithley Instruments PXI devices and is ignored.	
		**mode:** This input is not used by Keithley Instruments PXI devices and is ignored.
		**channel index:** This input is not used by Keithley Instruments PXI devices and is ignored.
		**slope:** This input is not used by Keithley Instruments PXI devices and is ignored.
		**level:** This input is not used by Keithley Instruments PXI devices and is ignored.
		**hysteresis:** This input is not used by Keithley Instruments PXI devices and is ignored.
		**skip count:** This input is not used by Keithley Instruments PXI devices and is ignored.
		**offset:** This input is not used by Keithley Instruments PXI devices and is ignored.
	**taskID in:** Identifies the group and the I/O operation.	
	**number of scans to read:** is the number of scans the VI is to retrieve from the acquisition buffer. The default input is -1, which set number of scans to read equal to the value of the number of scans to acquire parameter when the KI AI Start was called. If number of scans to read is -1 and number of scans to acquire is 0, KIDAQ LabVIEW sets number of scans to read to be the half of the buffer size.	
	**time limit in sec:** is the time limit for the read operation. The default input is -1.0, which means KIDAQ LabVIEW calculates a time limit based on the value of number of scans to read and the scan rate. If the scan rate is unknown, the VI uses 1 second as the time limit. The resolution of the timeout clock is about 55 ms.	
	**error in (no error):** describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the error in cluster in error out.	
		**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
		**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
		**source:** Identifies where an error occurred. The source string is usually the name of the VI that produced the error.

Table D-8 (continued)
Scaled and Binary Arrays

	**read/search position:** This input is not used by Keithley Instruments PXI devices and is ignored. The starting point for the read is the position where the read mark point to. Initially, the read mark points to the beginning of the acquisition buffer. As you retrieve data from the buffer using this VI, KIDAQ LabVIEW increments the read mark to point to the next block of data to be read.	
		**position:** This input is not used by Keithley Instruments PXI devices and is ignored.
		**read offset:** This input is not used by Keithley Instruments PXI devices and is ignored.
	**scan backlog:** is the amount of data remaining in the buffer after this VI completes.	
	**number read:** is the number of scans returned. This number is identical to number of scans to read unless an error or timeout appears or the VI reaches the end of the data.	
	**taskID out:** has the same value as ***taskID in***.	
 or 	**binary data:** is a 2D array that contains unscaled analog input data.	
	**scaled data:** is a 2D array that contains analog input data in Volts.	
	**retrieval complete:** is TRUE when the acquisition finishes and no backlog data remains.	
	**error out:** contains error information. If the error in cluster indicated an error, the error out cluster contains the same information. Otherwise, error out describes the error status of this VI.	
		**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
		**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
		**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.

Table D-9
Scaled Array

	**conditional retrieval specification:** This input is not used by Keithley Instruments PXI devices and is ignored.	
		**mode:** This input is not used by Keithley Instruments PXI devices and is ignored.
		**channel index:** This input is not used by Keithley Instruments PXI devices and is ignored.
		**slope:** This input is not used by Keithley Instruments PXI devices and is ignored.
		**level:** This input is not used by Keithley Instruments PXI devices and is ignored.
		**hysteresis:** This input is not used by Keithley Instruments PXI devices and is ignored.
		**skip count:** This input is not used by Keithley Instruments PXI devices and is ignored.
		**offset:** This input is not used by Keithley Instruments PXI devices and is ignored.
	**taskID in:** Identifies the group and the I/O operation.	
	**number of scans to read:** is the number of scans the VI is to retrieve from the acquisition buffer. The default input is -1, which set number of scans to read equal to the value of the number of scans to acquire parameter when the KI AI Start was called. If number of scans to read is -1 and number of scans to acquire is 0, KIDAQ LabVIEW sets number of scans to read to be the half of the buffer size.	
	**time limit in sec:** is the time limit for the read operation. The default input is -1.0, which means KIDAQ LabVIEW calculates a time limit based on the value of number of scans to read and the scan rate. If the scan rate is unknown, the VI uses 1 second as the time limit. The resolution of the timeout clock is about 55 ms.	
	**error in (no error):** describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the error in cluster in error out.	
		**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
		**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
		**source:** Identifies where an error occurred. The source string is usually the name of the VI that produced the error.

Table D-9 (continued)
Scaled Array

	**read/search position:** This input is not used by Keithley Instruments PXI devices and is ignored. The starting point for the read is the position where the read mark point to. Initially, the read mark points to the beginning of the acquisition buffer. As you retrieve data from the buffer using this VI, KIDAQ LabVIEW increments the read mark to point to the next block of data to be read.	
		**position:** This input is not used by Keithley Instruments PXI devices and is ignored.
		**read offset:** This input is not used by Keithley Instruments PXI devices and is ignored.
	**scan backlog:** is the amount of data remaining in the buffer after this VI completes.	
	**number read:** is the number of scans returned. This number is identical to number of scans to read unless an error or timeout appears or the VI reaches the end of the data.	
	**taskID out:** has the same value as taskID in.	
	**scaled data:** is a 2D array that contains analog input data in Volts.	
	**retrieval complete:** is TRUE when the acquisition finishes and no backlog data remains.	
	**error out:** contains error information. If the error in cluster indicated an error, the error out cluster contains the same information. Otherwise, error out describes the error status of this VI.	
		**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
		**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
		**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.

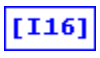
KI AI Single Scan

This VI returns one scan of data from the analog input channels for a non-buffered acquisition.

Table D-10
KI AI single scan

Binary Array	
	**taskID in:** identifies the group and the I/O operation.

Table D-10 (continued)
KI AI single scan

	**opcode:** This input is not used by Keithley Instruments PXI devices and is ignored.	
	**time limit in sec:** This input is not used by Keithley Instruments PXI devices and is ignored.	
	**error in (no error):** describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the error in cluster in error out.	
		**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
		**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
		**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.
	**data remaining:** This input is not used by Keithley Instruments PXI devices and is ignored.	
	**taskID out:** has the same value as taskID in.	
 or 	**binary data:** contains the unscaled binary data in Volts. The array index represents the channel.	
	**acquisition state:** This input is not used by Keithley Instruments PXI devices and is ignored	
	**error out:** contains error information. If the error in cluster indicated an error, the error out cluster contains the same information. Otherwise, error out describes the error status of this VI.	
		**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
		**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
		**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.

KI AI start

Starts a buffered analog input operation. This VI sets the scan rate, the number of scans to acquire, the conversion clock source, and the trigger conditions. The VI then starts an acquisition.

Table D-11
KI AI start

U32	**taskID in:** identifies the group and the I/O operation.	
I32	**number of scans to acquire:** is the total number of scans to acquire. A scan is one point per channel. With the default input -1, the device acquires exactly one buffer of data. The buffer size input to the KI AI Config VI determines the size of the buffer. The number of total scans includes any pretrigger scans requested. If you set number of scans to acquire to 0, the device acquires data indefinitely into the buffer until you stop the acquisition with the KI AI Clear VI. In this case, the VI ignores the pretrigger scans input. For KPXI-AI-2-65M, the number of scans to acquire has to be equal to the buffer size input to the KI AI Config VI.	
SGL	**scan rate:** is the number of scans/s to acquire. This is equivalent to the sampling rate per channel. The default for this parameter is 1000 scans/s. If you enter 0, the on-board internal clock is disabled and the external clock is used.	
	error in (no error): describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the error in cluster in error out.	
	TF	**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
	I32	**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
	abc	**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.
U16	**edge or slope:** 0: Do not change the default setting (default input). 1: Leading edge for digital trigger; positive slope for analog trigger. 2: Trailing edge for digital trigger; negative slope for analog trigger.	
I32	**pretrigger scans:** is the number of scans you want to save in the buffer before the trigger. The default for this parameter is 0, which means no data before the trigger is saved.	
U16	**trigger type:** specifies the type of trigger to start or stop the acquisition. 0: No trigger (default input). 1: Analog trigger (default setting). 2: Digital trigger. 3: SSI digital start trigger (for KPXI-AI-2-65M, the signal is through PXI trigger bus 3) and analog trigger (for the applications need both start and stop triggers, e.g. middle trigger or pre-trigger mode of operation). 4: SSI digital start trigger (for KPXI-AI-2-65M, the signal is through PXI trigger bus 3) and digital trigger (for the applications need both start and stop triggers, e.g. middle trigger or pre-trigger mode of operation).	
I32	**number of buffers to acquire:** This input is not used by Keithley Instruments PXI devices and is ignored. There is always only one buffer.	

Table D-11 (continued)
KI AI start

U16	**scan clock source:** identifies the A/D clock source. 0: Do not change the clock source setting (default input). 1: An internal timebase is used (default setting). 2: You supply a signal through the I/O connector 1 (for KPXI-DAQ series devices, the signal is from AFI0/AFI1 and for KPXI-AI-2-65M the signal is from CLK IN connector). 3: You supply a signal through the I/O connector 2 (for KPXI-DAQ series devices and KPXI-AI-2-65M, the signal is from SSI cable). 4: An internal timebase with double edged enabled (only available for KPXI-AI-2-65M). 5: You supply a signal through the I/O connector 1 with double edged enabled (only available for KPXI-AI-2-65M). 6: a signal from SSI cable with double edged enabled (only available for KPXI-AI-2-65M). 7: external timebase from SSI cable (for KPXI-DAQ series devices, the timebase is 40MHz and for KPXI-AI-2-65M, the time base is 60 MHz). 8: both conversion signal and external timebase from SSI cable (only available for KPXI-DAQ series devices. The timebase is 40MHz).	
	analog chan and level: contains the following parameter.	
	abc	**trigger channel:** specifies where the trigger comes from. When trigger type is 1 (analog trigger), the default for trigger channel is 0, i.e. analog input channel 0. When trigger type is 2 (digital trigger): 0: external digital pin (default). 1 : the signal from SSI cable. 2: both start and stop trigger signals are from SSI cable (available for KPXI-DAQ series devices or KPXI-AI-2-65M). 3~10: the signal is from PXI trigger bus 0 to 7. (only available for KPXI-AI-2-65M) 11: the signal is PXI_START signal. (only available for KPXI-AI-2-65M)
	SGL	**level:** level (measured in Volts) which analog source must cross for a trigger to occur. The default input for level is 0.0.
	additional trig params: cluster contains the following parameters:	
	SGL	**hysteresis:** This input is not used by Keithley Instruments PXI devices and is ignored.
	U16	**coupling:** This input is not used by Keithley Instruments PXI devices and is ignored.
	SGL	**delay:** specifies how long the device waits after a trigger occurs before sampling data. You express delay in seconds. The default input and setting are 0.0s (no delay).
	I32	**skip count:** This input is not used by Keithley Instruments PXI devices and is ignored.
	SGL	**time limit:** This input is not used by Keithley Instruments PXI devices and is ignored.
U32	**taskID out:** has the same value as ***taskID in***.	

Table D-11 (continued)
KI AI start

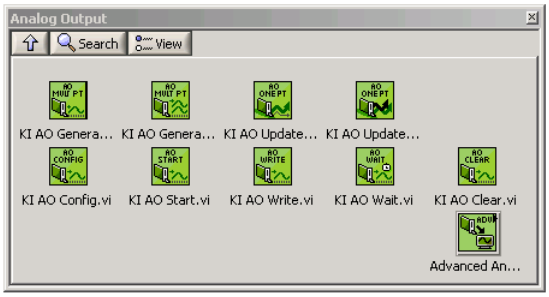
	error out: contains error information. If the error in cluster indicated an error, the error out cluster contains the same information. Otherwise, error out describes the error status of this VI.	
	TF	**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
	I32	**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
	abc	**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.
SGL	**actual scan rate:** may differ slightly from the requested scan rate, depending on the hardware capabilities.	
[ ]	**actual trigger params:** cluster may differ slightly from the requested trigger inputs, depending on the hardware capabilities. It contains the following parameters.	
	SGL	**actual level:** is the analog trigger level the VI used.
	SGL	**actual hysteresis:** This input is not used by Keithley Instruments PXI devices and is ignored.
	SGL	**actual delay:** is the delay the VI used.

Analog output VIs

Easy analog output VIs

Analog Output VIs (virtual instruments) are available in the Analog Output palette (Figure D-2).

Figure D-2
Analog output palette



KI AO generate waveform

Generates a timed and buffered waveform for the given output channel at the specified update rate. The KI AO Generate Waveform VI generates a waveform on a specified analog output channel. It does not return until the generation is complete.

NOTE *This VI is not supported for Keithley KDIO Series devices.*

Table D-12
KI AO generate waveform

	**device:** Number of the device (beginning from 1). The utility **Device Browser** can be used to get the information of current device configuration.
	**sub type:** is the sub-type of the device you assigned to the Keithley Instruments PXI device during configuration.
	**channel:** identifies the analog output channel. The default input is channel 0. The valid channel for each Keithley Instruments PXI series device is as follows: **KPXI-SDAQ-4-500K/KPXI-SDAQ-4-2M/KPXI-DAQ-64-3M/KPXI-DAQ-64-500K/ KPXI-DAQ-64-250K/KPXI-DAQ-96-3M**: 0 or 1 **KPXI-AO-4-1M**: 0 through 3 **KPXI-AO-8-1M**: 0 through 7
	**high limit:** is the highest expected level of the signal in Volts you want to generate.
	**low limit:** is the lowest expected level of the signal in Volts you want to generate.
	**reference source:** is the internal/external setting of the reference voltage for this channel. 0: Do not change the reference source setting (default input). 1: Internal (default setting). 2: External.
	**update rate:** is the number of updates to generate per second. The default rate is 1000 update/s.
	**waveform:** is a 1D array that contains analog output data to be written the specified channel expressed in Volts. The data must be supplied.

KI AO generate waveforms

Generates timed and buffered waveforms for the given output channels at the specified update rate. The KI AO Generate Waveforms VI generates waveforms on specified analog output channels. It does not return until the generation is complete.

NOTE *This VI is not supported for Keithley KDIO Series devices.*

Table D-13
KI AO generate waveforms

	**device:** Number of the device (beginning from 1). The utility **Device Browser** can be used to get the information of current device configuration.
	**sub type:** is the sub-type of the device you assigned to the Keithley Instruments PXI device during configuration.

Table D-13 (continued)
KI AO generate waveforms

abc	**channels:** Specifies the set of analog output channels you want to use. If x, y, and z refer to channels, you can specify a list of channels by separating the individual channels with commas (for example, x,y,z). If x refers to the first channel in a consecutive channel range and y refers to the last channel, you can specify the range by separating the first and last channels by a colon (for example, x:y). See KI AO generate waveform above for available channels on each module.
SGL	**high limit:** is the highest expected level of the signal in Volts you want to generate.
SGL	**low limit:** is the lowest expected level of the signal in Volts you want to generate.
U16	**reference source:** is the internal/external setting of the reference voltage for this channel. 0: Do not change the reference source setting (default input). 1: Internal (default setting). 2: External.
SGL	**update rate:** is the number of updates to generate per second. The default rate is 1000 update/s.
[SGL]	**waveforms:** is a 2D array that contains analog output data expressed in volts. You must supply this data. The channel order of the data must be the same channel order specified in channels. You must specify waveforms, where the first (top) dimension is the update number and the second (bottom) dimension is the channel number.

KI AO update channel

Writes a single value to the specified analog output channel. If an error occurs, a dialog box appears, giving you the error information.

Table D-14
KI AO update channel

I16	**device:** Number of the device (beginning from 1). The utility **Device Browser** can be used to get the information of current device configuration.
U16	**sub type:** is the sub-type of the device you assigned to the Keithley Instruments PXI device during configuration.
abc	**channel:** identifies the analog output channel. The default input is channel 0. The valid channel for each Keithley Instruments PXI device is as follows: **KPXI-SDAQ-4-500K/KPXI-SDAQ-4-2M/KPXI-DAQ-64-3M/KPXI-DAQ-64-500K/ KPXI-DAQ-64-250K/KPXI-DAQ-96-3MKI**: 0 or 1 **KPXI-AO-4-1M**: 0 through 3 **KPXI-AO-8-1M**: 0 through 7
SGL	**high limit:** is the highest expected level of the signal in Volts you want to generate.
SGL	**low limit:** is the lowest expected level of the signal in Volts you want to generate.

Table D-14 (continued)
KI AO update channel

U16	**reference source:** is the internal/external setting of the reference voltage for this channel. 0: Do not change the reference source setting (default input). 1: Internal (default setting). 2: External.
SGL	**value:** contains the value to be written to the specified analog output channel expressed in the physical units of your signal. You must supply this data. All boards require Voltage for the physical unit.

KI AO update channels

Writes values to each of the specified analog output channels. If an error occurs, a dialog box appears, giving you the error information.

Table D-15
KI AO update channels

I16	**device:** Number of the device (beginning from 1). The utility **Device Browser** can be used to get the information of current device configuration.
U16	**sub type:** is the sub-type of the device you assigned to the Keithley Instruments PXI device during configuration.
abc	**channels:** Specifies the set of analog output channels you want to use. If x, y, and z refer to channels, you can specify a list of channels by "x,y,z". If x refers to the first channel in a consecutive channel range and y refers to the last channel, you can specify the range by "x:y". See KI AO update channel above for available channels on each module.
SGL	**high limit:** is the highest expected level of the signal in Volts you want to generate.
SGL	**low limit:** is the lowest expected level of the signal in Volts you want to generate.
U16	**reference source:** is the internal/external setting of the reference voltage for this channel. 0: Do not change the reference source setting (default input). 1: Internal (default setting). 2: External.
[SGL]	**value:** is a 1D array that contains the analog output data expressed in the physical units of your signal. You must supply this data. All boards require Voltage for the physical unit.

Intermediate analog output VIs

KI AO clear

This VI stops an analog output generation associated with taskID in and releases associated internal resources, including buffers. Before beginning a new signal generation, you must call the KI AO Config VI.

NOTE *This VI is not supported for Keithley KDIO Series devices.*

Table D-16
KI AO clear

U32	**taskID in:** identifies the group and the I/O operation.	
	error in (no error): describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the error in cluster in error out.	
	TF	**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
	I32	**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
	abc	**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.
U32	**taskID out:** has the same value as taskID in.	
	error out: contains error information. If the error in cluster indicated an error, the error out cluster contains the same information. Otherwise, error out describes the error status of this VI.	
	TF	**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
	I32	**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
	abc	**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.

KI AO Config

Configures a buffered analog output operation, including configuring the hardware and allocating a buffer.

NOTE *This VI is not supported for Keithley KDIO Series devices*.

Table D-17
KI AO Config

SGL	**interchannel delay:** This input is not used by Keithley Instruments PXI devices and is ignored.
U16	**sub type:** is the sub-type of the device you assigned to the Keithley Instruments PXI device during configuration.
[ ]	**limit settings:** is an array of clusters. Each array element contains the expected signal limits for the channels specified by the corresponding element of channels. If there are fewer elements in this array than in channels, the VI uses the last array element for the rest of the channels. The default for the limit settings array is an empty array, which means the limit settings keep their default settings.

Table D-17 (continued)
KI AO Config

	SGL	**high limit:** is the highest scaled data in Volts.
	SGL	**low limit:** is the lowest scaled data in Volts.
	SGL	**reference source:** is the internal/external setting of the reference voltage for this channel. 0: Do not change the reference source setting (default input). 1: Internal (default setting). 2: External.
I16	**device:** Number of the device (beginning from 1). The utility **Device Browser** can be used to get the information of current device configuration.	
[abc]	**channels:** specifies the set of analog output channels. channels is an array of strings. If x, y, and z refer to channels, you can specify a list of channels in a single element by separating the individual channels by commas. For example, "x,y,z". If x refers to the first channel in a consecutive channel range and y refers to the last channel, you can specify the range by separating the first and last channels by a colon. For example, "x:y". The valid channel order for data is as follows: **KPXI-SDAQ-4-500K/KPXI-SDAQ-4-2M/KPXI-DAQ-64-3M/KPXI-DAQ-64-500K/ KPXI-DAQ-64-250K/KPXI-DAQ-96-3M**: numbers in **channels** must be within 0 and 1 **KPXI-AO-4-1M**: numbers in **channels** must be within 0 and 3 **KPXI-AO-8-1M**: numbers in **channels** must be within 0 and 7	
I32	**buffer size:** is the number of updates you want the buffer to hold. The default for this parameter is 1000 scans.	
I16	**group:** is the number, from 0 to 15, that you assign to the specified set of channels. The default input and setting for group is 0. If you only have one update operation for this device, leave this input unwired and use group 0.	
	error in (no error): describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the error in cluster in error out.	
	TF	**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
	I32	**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
	abc	**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.
U16	**allocation mode:** This input is not used by Keithley Instruments PXI devices and is ignored.	
U32	**taskID:** identifies the group and the I/O operation.	
I32	**number of channels:** is the total number of channels in the group.	

Table D-17 (continued)
KI AO Config

	**error out:** contains error information. If the error in cluster indicated an error, the error out cluster contains the same information. Otherwise, error out describes the error status of this VI.	
		**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
		**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
		**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.

KI AO start

Starts a buffered analog output operation. This VI sets the update rate, and then starts the generation.

NOTE *This VI is not supported for Keithley KDIO Series devices.*

Table D-18
KI AO start

	**taskID in:** identifies the group and the I/O operation.	
	**update rate:** is the number of updates/s to generate. This is equivalent to the update rate per channel. The default for this parameter is 1000 updates/s. If you enter 0, the on-board internal clock is disabled and the external clock is used.	
	**error in (no error):** describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the error in cluster in error out.	
		**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
		**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
		**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.
	**number of buffer iterations:** is the number of times KIDAQ LabVIEW has to generate the waveform from the output buffer. After generating the buffer the specified number of times, the generation stops. The default value is 1, which means KIDAQ LabVIEW generates the buffer only once. If you use a value of 0, KIDAQ LabVIEW generates the buffer continuously, until you stop the operation with the KI AO Clear VI.	
	**clock:** 0: Do not change the default setting (default input). 1: Update clock 1 (default setting).	

Table D-18 (continued)
KI AO start

	**clock source:** specifies the source of the clock. 0: Do not change the clock source setting (default input). 1: Internal (default setting). 6: I/O connector. 7: SSI (RTSI) Connection.	
	**taskID out:** has the same value as taskID in.	
	**error out:** contains error information. If the error in cluster indicated an error, the error out cluster contains the same information. Otherwise, error out describes the error status of this VI.	
		**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
		**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
		**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.
	**actual update rate:** may differ slightly from the requested update rate, depending on the hardware capabilities.	

KI AO wait

This VI waits until the waveform generation of the task completes before returning.

NOTE *This VI is not supported for Keithley KDIO Series devices*.

Table D-19
KI AO wait

	**taskID in:** identifies the group and the I/O operation.	
	**update rate:** is the number of updates/s to generate. This is equivalent to the update rate per channel. The default for this parameter is 1000 updates/s.	
	**error in (no error):** describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the error in cluster in error out.	
		**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
		**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
		**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.
	**check every N updates:** informs the VI how often to check the status of the task to see if generation completes. This parameter default is to check every 5 updates.	

Table D-19 (continued)
KI AO wait

	**taskID out:** has the same value as taskID in.	
	**error out:** contains error information. If the error in cluster indicated an error, the error out cluster contains the same information. Otherwise, error out describes the error status of this VI.	
		**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
		**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
		**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.

KI AO write

This VI writes data into the buffer for a buffered analog output operation. The data written into the buffer will then be generated (transferred from the buffer to the DAC) at the update rate specified in KI AO Start.

KI AO Write is a polymorphic VI that you can configure to output the following kinds of data:

- Binary Array
- Scaled Array

NOTE *This VI is not supported for Keithley KDIO Series devices.*

Table D-20
KI AO write binary array

	**taskID in:** identifies the group and the I/O operation.	
	**binary data:** is a 2D array that contains unscaled analog output data. The channel order of the data must be the same as the channel order you specify in channels. You must specify waveforms, where the first (top) dimension is the update number and the second (bottom) dimension is the channel number. The length of the data array determines the number of updates the VI writes. When no data is wired, this VI is still useful for reporting update progress information.	
	**time limit in sec:** is the time limit for the output operation. The default input is -1.0, which means KIDAQ LabVIEW calculates a time limit based on the value of number of updates to generate and the update rate. If the update rate is unknown, the VI uses 1 second as the time limit. The resolution of the timeout clock is about 55 ms.	
	**allow regeneration:** is not used by Keithley Instruments PXI devices and is ignored.	
	**error in (no error):** describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the error in cluster in error out.	
		**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.

Table D-20 (continued)
KI AO write binary array

	I32	**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
	abc	**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.
U32	**taskID out:** has the same value as ***taskID in***.	
I32	**number of updates done:** is the number of updates the VI has generated; that is, the number of updates the VI has actually transferred from the buffer to the onboard FIFO.	
I32	**number of buffers done:** is the number of times the VI has generated an entire buffer; that is, the number of times the VI has actually transferred all the data in the buffer to the onboard FIFO.	
TF	**generation complete:** is TRUE when the generation finishes.	
	error out: contains error information. If the error in cluster indicated an error, the error out cluster contains the same information. Otherwise, error out describes the error status of this VI.	
	TF	**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
	I32	**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
	abc	**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.

Table D-21
KI AO write binary array scaled array

U32	**taskID in:** Identifies the group and the I/O operation.
[SGL]	**scaled data:** is a 2D array that contains analog output data expressed in volts. The channel order of the data must be the same the channel order you specify in channels. You must specify waveforms, where the first (top) dimension is the update number and the second (bottom) dimension is the channel number. The length of the data array determines the number of updates the VI writes. When no data is wired, this VI is still useful for reporting update progress information.
SGL	**time limit in sec:** is the time limit for the output operation. The default input is -1.0, which means KIDAQ LabVIEW calculates a time limit based on the value of number of updates to generate and the update rate. If the update rate is unknown, the VI uses 1 second as the time limit. The resolution of the timeout clock is about 55 ms.
TF	**allow regeneration:** is not used by Keithley Instruments PXI devices and is ignored.

Table D-21 (continued)
KI AO write binary array scaled array

	**error in (no error):** describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the error in cluster in error out.	
		**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
		**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
		**source:** Identifies where an error occurred. The source string is usually the name of the VI that produced the error.
	**taskID out:** has the same value as ***taskID in***.	
	**number of updates done:** is the number of updates the VI has generated; that is, the number of updates the VI has actually transferred from the buffer to the onboard FIFO.	
	**number of buffers done:** is the number of times the VI has generated an entire buffer; that is, the number of times the VI has actually transferred all the data in the buffer to the onboard FIFO.	
	**generation complete:** is TRUE when the generation finishes.	
	**error out:** contains error information. If the error in cluster indicated an error, the error out cluster contains the same information. Otherwise, error out describes the error status of this VI.	
		**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
		**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
		**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.

Advanced analog output VIs

KI AO Trigger and Gate Config

Configures the trigger conditions for analog output operations.

NOTE *This VI is not supported for Keithley KDIO Series devices.*

Table D-22
KI AO Trigger and Gate Config

U32	**taskID in:** identifies the group and the I/O operation.	
	error in (no error): describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the error in cluster in error out.	
	TF	**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
	I32	**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
	abc	**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.
U16	**trigger or gate source:** specifies the source of trigger signal. 0: No change (default input). 1: None (default setting). 2: External WFDTRIG pin . 3: SSI (RTSI) pin . 5: ATCOUT (the output of the analog trigger circuitry).	
U16	**trigger or gate condition:** selects a rising or falling edge trigger. 0: No change (default input). 1: None (default setting). 2: Trigger on rising edge. 3: Trigger on falling edge.	
[abc]	**trigger or gate source specification:** is not used by Keithley Instruments PXI devices and is ignored.	
	additional trig params: cluster contains the following parameters:	
	SGL	**delay:** specifies how long the device waits after a trigger occurs before waveform generates. You express delay in seconds. The default input and setting are 0.0s (no delay).
U32	**taskID out:** has the same value as ***taskID in***.	
	error out: contains error information. If the error in cluster indicated an error, the error out cluster contains the same information. Otherwise, error out describes the error status of this VI.	

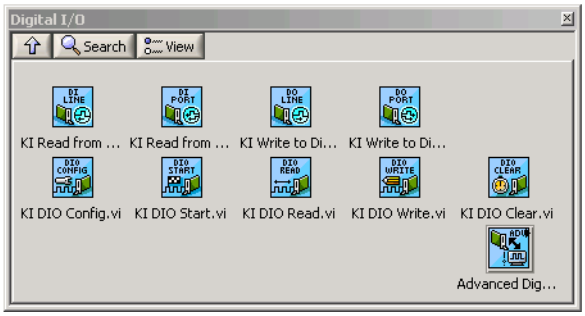
Table D-22 (continued)
KI AO Trigger and Gate Config

	TF	**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
	I32	**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
	abc	**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.
[⋯]	**actual trigger params:** cluster may differ slightly from the requested trigger inputs, depending on the hardware capabilities. It contains the following parameters.	
	SGL	**actual delay:** is the delay the VI used.

Digital I/O VIs

Two classes of Digital I/O VIs are available in the Digital I/O palette: the Easy Digital I/O VIs, Intermediate Digital I/O VIs and Advanced Digital I/O VIs (Figure D-3).

Figure D-3
Digital I/O palette



Easy Digital I/O VIs

KI Read from Digital Line

Reads the logical state of a digital line on a digital port. If an error occurs, a dialog box appears, giving you the error information.

NOTE *When you call this VI on a digital I/O port that is part of an 8255 PPI and your iteration terminal is left at 0, the 8255 PPI goes through a configuration phase, where all the ports within the same PPI chip get reset to logic low, regardless of the data direction. The data*

direction on other ports, however, is maintained. To avoid this effect, connect a value other than 0 to the iteration terminal once you have configured the desired ports.

Table D-23
KI Read from Digital Line

I16	**device:** Number of the device (beginning from 1). The utility **Device Browser** can be used to get the information of current device configuration.
U16	**sub type:** is the sub-type of the device you assigned to the Keithley Instruments PXI device during configuration.
abc	**digital channel:** is the port number to read. **KPXI-DIO-16-16**: 1 **KPXI-DIO-48**: 0: P1A, 1: P1B, 2: P1C Lower, 3: P1C Upper 4: P2A, 5: P2B, 6: P2C Lower, 7: P2C Upper **KPXI-RDI-8-16**: 1 **KPXI-DIO-32-80M**: 3 (aux. input port) **KPXI-DIO-32-32**: 2 or 3 **KPXI-DIO-64-0**: 1 or 2 **KPXI-DIO-0-64**: 3 **KPXI-DAQ series devices**: 0: P1A, 1: P1B, 2: P1C Lower, 3: P1C Upper
I16	**Line:** is the individual port bit or line to be used for I/O.
TF	**Line state:** is TRUE for high logic, and FALSE for low logic.
I16	**Port width:** is the total width or the number of lines of the port in bits. For example, you can combine two 4-bit ports into an 8-bit port on a KPXI-DIO-48 device by setting port width to 8.
I32	**iteration:** When iteration is 0 (default), KIDAQ LabVIEW re-configures the port. If iteration is greater than zero, KIDAQ LabVIEW uses the existing configuration, which improves performance. It can be used to optimize operation when you execute this VI in a loop.

KI Read from Digital Port

Reads a digital channel that you configure. If an error occurs, a dialog box appears, giving you the error information.

NOTE *When you call this VI on a digital I/O port that is part of an 8255 PPI and your iteration terminal is left at 0, the 8255 PPI goes through a configuration phase, where all the ports within the same PPI chip get reset to logic low, regardless of the data direction. The data direction on other ports, however, is maintained. To avoid this effect, connect a value other than 0 to the iteration terminal once you have configured the desired ports.*

Table D-24
KI Read from Digital Port

I16	**device:** Number of the device (beginning from 1). The utility **Device Browser** can be used to get the information of current device configuration.
U16	**sub type:** is the sub-type of the device you assigned to the Keithley Instruments PXI device during configuration.

Table D-24 (continued)
KI Read from Digital Port

	**digital channel:** is the port number to read. **KPXI-DIO-16-16**: 1 **KPXI-DIO-48**: 0: P1A, 1: P1B, 2: P1C Lower, 3: P1C Upper 4: P2A, 5: P2B, 6: P2C Lower, 7: P2C Upper **KPXI-RDI-8-16**: 1 **KPXI-DIO-32-80M**: 3 (aux. input port) **KPXI-DIO-32-32**: 2 or 3 **KPXI-DIO-64-0**: 1 or 2 **KPXI-DIO-0-64**: 3 **KPXI-DAQ series devices**: 0: P1A, 1: P1B, 2: P1C Lower, 3: P1C Upper
	**pattern:** is the data the VI reads from the digital port.
	**port width:** is the total width or the number of lines of the port in bits. For example, you can combine two 4-bit ports into an 8-bit port on a KPXI-DIO-48 device by setting port width to 8.
	**iteration:** When iteration is 0 (default), KIDAQ LabVIEW re-configures the port. If iteration is greater than zero, KIDAQ LabVIEW uses the existing configuration, which improves performance. It can be used to optimize operation when you execute this VI in a loop.

KI Write to Digital Line

Sets the logic state of a digital line on a specified digital port. If an error occurs, a dialog box appears, giving you the error information.

NOTE *When you call this VI on a digital I/O port that is part of an 8255 PPI and your iteration terminal is left at 0, the 8255 PPI goes through a configuration phase, where all the ports within the same PPI chip get reset to logic low, regardless of the data direction. The data direction on other ports, however, is maintained. To avoid this effect, connect a value other than 0 to the iteration terminal once you have configured the desired ports*.

Table D-25
KI Write to Digital Line

	**device:** Number of the device (beginning from 1). The utility **Device Browser** can be used to get the information of current device configuration.
	**sub type:** is the sub-type of the device you assigned to the Keithley Instruments PXI device during configuration.

Table D-25 (continued)
KI Write to Digital Line

	**digital channel:** is the port number to write. **KPXI-DIO-16-16**: 0 **KPXI-DIO-48**: 0: P1A, 1: P1B, 2: P1C Lower, 3: P1C Upper 4: P2A, 5: P2B, 6: P2C Lower, 7: P2C Upper **KPXI-RDI-8-16**: 0 **KPXI-DIO-32-80M**: 1 (aux. output port) **KPXI-DIO-32-32**: 0 (DO) or 1 (LED) **KPXI-DIO-64-0**: 0 (LED) **KPXI-DIO-0-64**: 0 (DO Low), 1 (DO High) **KPXI-DIO-0-64**: 0 (DO Low), 1 (DO High), 2 (LED) **KPXI-DAQ series devices**: 0: P1A, 1: P1B, 2: P1C Lower, 3: P1C Upper
	**line:** is the individual port bit or line to be used for I/O.
	**line state:** is TRUE for high logic, and FALSE for low logic.
	**port width:** is the total width or the number of lines of the port in bits. For example, you can combine two 4-bit ports into an 8-bit port on a KPXI-DIO-48 device by setting port width to 8.
	**iteration:** When iteration is 0 (default), KIDAQ LabVIEW re-configures the port. If iteration is greater than zero, KIDAQ LabVIEW uses the existing configuration, which improves performance. It can be used to optimize operation when you execute this VI in a loop.

KI Write to Digital Port

Writes a digital pattern to a digital port. If an error occurs, a dialog box appears, giving you the error information.

NOTE *When you call this VI on a digital I/O port that is part of an 8255 PPI when your iteration terminal is left at 0, the 8255 PPI goes through a configuration phase, where all the ports within the same PPI chip get reset to logic low, regardless of the data direction. The data direction on other ports, however, is maintained. To avoid this effect, connect a value other than 0 to the iteration terminal once you have configured the desired ports.*

Table D-26
KI Write to Digital Port

	**device:** Number of the device (beginning from 1). The utility **Device Browser** can be used to get the information of current device configuration.
	**sub type:** is the sub-type of the device you assigned to the Keithley Instruments PXI device during configuration.

Table D-26 (continued)
KI Write to Digital Port

abc	**digital channel:** is the port number to write. **KPXI-DIO-16-16**: 0 **KPXI-DIO-48**: 0: P1A, 1: P1B, 2: P1C Lower, 3: P1C Upper 4: P2A, 5: P2B, 6: P2C Lower, 7: P2C Upper **KPXI-RDI-8-16**: 0 **KPXI-DIO-32-80M**: 1 (aux. output port) **KPXI-DIO-32-32**: 0 (DO) or 1 (LED) **KPXI-DIO-64-0**: 0 (LED) **KPXI-DIO-0-64**: 0 (DO Low), 1 (DO High) **KPXI-DIO-0-64**: 0 (DO Low), 1 (DO High), 2 (LED) **KPXI-DAQ series devices series**: 0: P1A, 1: P1B, 2: P1C Lower, 3: P1C Upper
U32	**pattern:** is the bit pattern writes to the digital port.
I16	**port width:** is the total width or the number of lines of the port in bits. For example, you can combine two 4-bit ports into an 8-bit port on a KPXI-DIO-48 device by setting port width to 8. If you are using channel names, port width is not needed and is ignored.
I32	**iteration:** When iteration is 0 (default), KIDAQ LabVIEW re-configures the port. If iteration is greater than zero, KIDAQ LabVIEW uses the existing configuration, which improves performance. It can be used to optimize operation when you execute this VI in a loop.

Intermediate Digital I/O VIs

KI DIO Clear

This VI stops a digital input or output acquisition. Before beginning a new acquisition, you must call the KI DIO Config VI.

NOTE *This VI is not supported for KPXI-DAQ series devices.*

Table D-27
KI DIO Clear

U32		**taskID in:** identifies the group and the I/O operation.
		error in (no error): describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the error in cluster in error out.
	TF	**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
	I32	**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.

Table D-27 (continued)
KI DIO Clear

		**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.
	**taskID out:** has the same value as ***taskID in***.	
	**error out:** contains error information. If the error in cluster indicated an error, the error out cluster contains the same information. Otherwise, error out describes the error status of this VI.	
		**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
		**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
		**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.

KI DIO Config

Configures a buffered digital I/O operation, including configuring the hardware and allocating a buffer. The VI only applies to KPXI-DIO-32-80M devices.

NOTE *This VI is not supported for KPXI-DAQ series devices.*

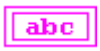
Table D-28
KI DIO Config

	**device:** Number of the device (beginning from 1). The utility **Device Browser** can be used to get the information of current device configuration.
	**sub type:** is the sub-type of the device you assigned to the Keithley Instruments PXI device during configuration.
	**port list:** Specifies the set of digital ports, each of which is 8, 16 or 32 lines wide. The valid ports are as follows: KPXI-DIO-32-80M 0 (digital output port), 2 (digital input port)
	**port width:** is the total width or the number of lines of the port in bits. port width is only valid for KPXI-DIO-32-80M which supports 8-bit, 16-bit and 32-bit of data acquisition
	**group direction:** sets the direction for the group. 0: Do not change the group direction setting (default input). 1: Input (default setting). 2: Output.
	**error in (no error):** describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the error in cluster in error out.
	 **status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.

Table D-28 (continued)
KI DIO Config

		**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
		**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.
	**number of scans/ updates:** specifies how much memory to allocate for the buffer. The default input for number of scans/updates is -1, which means KIDAQ LabVIEW leaves the current setting for number of scans/updates unchanged. The default setting for number of scans/updates is 1000.	
	**group:** is the number the VI assigns to the set of ports, ranging from 0 to 15. The default input and setting for group is 0.	
	**handshaking mode parameters:** affects the handshaking operation of KPXI-DIO-32-80M devices.	
		**signal mode:** This input is not used by Keithley Instruments PXI devices and is ignored.
		**edge mode:** This input is not used by Keithley Instruments PXI devices and is ignored.
		**request polarity:** specifies active high or low handshaking request signals. 0: Do not change the request polarity setting (default input). 1: Active low requests (default setting). 2: Active high requests.
		**acknowledge polarity:** specifies active high or low handshaking acknowledge signals. 0: Do not change the acknowledge polarity setting (default input). 1: Active low acknowledges (default setting). 2: Active high acknowledges.
		**acknowledge modify mode:** This input is not used by Keithley Instruments PXI devices and is ignored.
		**acknowledge modify amount:** This input is not used by Keithley Instruments PXI devices and is ignored
		**hardware double-buffer mode:** This input is not used by Keithley Instruments PXI devices and is ignored.
	**terminator:** is TRUE if output port terminator is on and is FALSE if output port terminator is off. Terminator affects only KPXI-DIO-32-80M.	
	**burst handshaking enable:** is TRUE if burst handshaking mode is enabled and is FALSE if burst handshaking mode is disabled. burst handshaking enable affects only KPXI-DIO-32-80M.	
	**fifo control:** controls the DO FIFO. This parameter is only valid for KPXI-DIO-32-80M.	

Table D-28 (continued)
KI DIO Config

		**fifo wait enable**: TRUE: delay output data until FIFO is not almost empty FALSE: digital output does not wait for FIFO is not almost empty.
		**threshold:** Is the programmable almost empty threshold of both PORTB FIFO and PORTA FIFO (if output port width is 32).
	**taskID:** identifies the group and the I/O operation.	
	**error out:** contains error information. If the error in cluster indicated an error, the error out cluster contains the same information. Otherwise, error out describes the error status of this VI.	
		**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
		**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
		**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.

KI DIO Read

Reads data from the internal buffer and returns the data read in pattern.

KI DIO Read is a polymorphic VI that you can configure to output the following kinds of data:

- U8 Array (with port width 8)
- U16 Array (with port width 16)
- U32 Array (with port width 32)

NOTE *This VI is not supported for KPXI-DAQ series devices.*

Table D-29
KI DIO Read

	**taskID in:** identifies the group and the I/O operation.	
	**number of scans to read:** is the number of scans to retrieve from buffer. This parameter defaults to -1, which means leaving the number of scans to read setting unchanged. The default setting is equal to the size of the buffer, which you set by KI DIO Config VI. If number of scans to read is 0, you can check the scan backlog to determine how many scans have accumulated. The VI waits until the data is available or the time limit expires.	
	**read location:** This input is not used by Keithley Instruments PXI devices and is ignored. The starting point for the read is the position where the read mark point to. Initially, the read mark points to the beginning of the acquisition buffer. As you retrieve data from the buffer using this VI, the read mark is incremented to point to the next block of data to be read.	
		**read offset:** This input is not used by Keithley Instruments PXI devices and is ignored.

Table D-29 (continued)
KI DIO Read

	U16	**read mode:** This input is not used by Keithley Instruments PXI devices and is ignored.
	**error in (no error):** describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the error in cluster in error out.	
	TF	**Status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
	I32	**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
	abc	**Source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.
	**time limit in sec:** timeout for data read. The default input is -1.0, which means KIDAQ LabVIEW calculates a time limit based on the value of number of scans to read and the scan rate. If the scan rate is unknown, the VI uses 1 second as the time limit. The resolution of the timeout clock is about 55 ms.	
U32	**taskID out:** has the same value as ***taskID in***.	
[U8] or [U16] or [U32]	**port data:** is a 1D array containing the digital data that the VI obtained from the internal buffer. Each element in this array is an 8-bit, 16-bit or 32-bit unsigned integer that represents a single port data.	
	**error out:** contains error information. If the error in cluster indicated an error, the error out cluster contains the same information. Otherwise, error out describes the error status of this VI.	
	TF	**Status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
	I32	**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
	abc	**Source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.
I32	**scan backlog:** is the amount of data in the buffer that remains unread after this VI completes.	
I32	**number read:** is the number of scans returned.	
TF	**retrieval complete:** is TRUE when the total number of the scans you specified in the KI DIO Start VI has been read.	

KI DIO Start

Starts a buffered digital I/O operation.

NOTE *This VI is not supported for KPXI-DAQ series devices.*

Table D-30
KI DIO Start

		**taskID in:** identifies the group and the I/O operation.
		**number of scans /updates to acquire or generate:** is the total number of scans to acquire or generate. With the default input -1, the device acquires or generates exactly one buffer of data. The buffer size input to the KI DIO Config VI determines the size of the buffer. If number of scans/updates to acquire or generate is 0, the device acquires or generates data continuously until you stop the operation.
		**trigger type:** specifies the type of trigger. 0:Do not change (default input). 1:Start trigger. KIDAQ LabVIEW waits trigger signal to start DIO operation.
		**trigger mode:** sets the trigger on or off. 0: Do not change (default input). 1: Off (default setting). 2: On.
		**trigger condition:** specifies when the digital operation triggers. 0: Do not change (default input). 1: Trigger on rising edge (default setting). 2: Trigger on falling edge.
		**error in (no error):** describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the error in cluster in error out.
		**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
		**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
		**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.
		**handshake source:** determines the source of the conditions that perform a data transfer. 0: Do not change the handshake source setting (default input). 1: Internal clock. 2: I/O connector (default setting). When handshake source is 1, the clock frequency control determines the clock rate. When handshake source is 2, you must connect the handshake signal to the proper line on the I/O connector.
		**clock frequency:** is the rate to which you want to handshake the data. This parameter is expressed in scans/s or updates/s. This parameter defaults to -1.0. The default setting is undefined.
		**taskID out:** has the same value as ***taskID in***.

Table D-30 (continued)
KI DIO Start

	**error out:** contains error information. If the error in cluster indicated an error, the error out cluster contains the same information. Otherwise, error out describes the error status of this VI.	
		**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
		**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
		**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.

KI DIO Write

Writes digital output data to the internal transfer buffer. You can call the KI DIO Write VI after the transfer begins to retrieve the output status information.

NOTE *This VI is not supported for KPXI-DAQ series devices.*

Table D-31
KI DIO Write

	**taskID in:** identifies the group and the I/O operation.	
 or  or 	**digital data:** is a 1D array containing digital output data. Each element in this array is an 8-bit, 16-bit and 32-bit unsigned integer that represents a single port data. If you call this VI with an empty array, you can examine buffer iterations and generation complete to retrieve the output progressing information.	
	**error in (no error):** describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the error in cluster in error out.	
		**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
		**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
		**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.
	**time limit in sec:** timeout for data write. The default input is -1.0, which means KIDAQ LabVIEW calculates a time limit based on the value of number of updates and the scan rate. If the scan rate is unknown, the VI uses 1 second as the time limit. The resolution of the timeout clock is about 55 ms.	
	**write location:** Determines where the write begins. Contains the following parameters.	

Table D-31 (continued)
KI DIO Write

	I32	**write offset:** The VI adds the value of write offset to the write mark to determine where the write begins. The default input is -1, which means leaving the write offset setting unchanged. This parameter defaults to a setting of 0.
	U16	**write mode:** Setting write mode to 2 moves the write mark to the beginning of the buffer before the VI adds write offset to the write mark. 0: Do not change the write mode setting (default input). 1: Write at the write mark plus the write offset (default setting). 2: Write at the beginning of the buffer plus the write offset.
U32	**taskID out:** has the same value as ***taskID in***.	
	error out: contains error information. If the error in cluster indicated an error, the error out cluster contains the same information. Otherwise, error out describes the error status of this VI.	
	TF	**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
	I32	**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
	abc	**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.
I32	**buffer iterations:** indicates the current number of complete iterations of the buffer.	
TF	**generation complete:** is TRUE when the number of updates to generate has finished.	

Advanced Digital I/O VIs

KI DIO Port Config

Configures a digital channel. You can use the task ID that this VI returns only in digital port VIs.

Table D-32
KI DIO Port Config

	**device:** Number of the device (beginning from 1). The utility **Device Browser** can be used to get the information of current device configuration.	
	**sub type:** is the sub-type of the device you assigned to the Keithley Instruments PXI device during configuration.	
	**digital channel:** is the port number to write. **KPXI-DIO-16-16**: 0 **KPXI-DIO-48**: 0: P1A, 1: P1B, 2: P1C Lower, 3: P1C Upper 4: P2A, 5: P2B, 6: P2C Lower, 7: P2C Upper **KPXI-RDI-8-16**: 0 **KPXI-DIO-32-80M**: 1 (auxiliary output port) **KPXI-DIO-32-32**: 0 (DO) or 1 (LED) **KPXI-DIO-64-0**: 0 (LED) **KPXI-DIO-0-64**: 0 (DO Low), 1 (DO High) **KPXI-DIO-0-64**: 0 (DO Low), 1 (DO High), 2 (LED) **KPXI-DAQ series devices**: 0: P1A, 1: P1B, 2: P1C Lower, 3: P1C Upper	
	**port width:** is the total width or the number of lines of the port in bits. port width is only valid for KPXI-DIO-32-80M which supports 8-bit, 16-bit and 32-bit of data acquisition	
	**error in (no error):** describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the error in cluster in error out.	
		**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
		**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
		**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.
	**line direction map:** specifies the direction of each line in the port. If a bit is 0 in the line map, the line is an input line. If a bit is 1, the line is an output line. Set line direction map to -1 to make all the lines in a port output lines. Set line direction map to 0 to make all the lines in a port input lines. Port C (e.g. P1C, P2C, etc.) are the only ports on which you can configure lines for different directions. The least significant bit in the line map corresponds to line 0 in the port. The line direction map parameter defaults to 0.	
	**wired OR map:** is not used and ignored.	

Table D-32 (continued)
KI DIO Port Config

U32	**taskID out:** uniquely identifies the digital group. Use this value as the task ID to refer to this group in subsequent digital port VIs.	
	error out: contains error information. If the error in cluster indicated an error, the error out cluster contains the same information. Otherwise, error out describes the error status of this VI.	
	TF	**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
	I32	**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
	abc	**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.

Counter VIs

Six Counter VIs are contained in the Counter palette.

Easy Counter VIs

KI Count Events or Time

Configures one or two counters to count external events. An external event is a high or low signal transition on the specified ***GPTCn_SRC*** pin of the counter.

NOTE *This VI is not supported for Keithley KDIO series devices.*

Table D-33
KI Count Events or Time

I16	**device:** Number of the device (beginning from 1). The utility **Device Browser** can be used to get the information of current device configuration.
U16	**sub type:** is the sub-type of the device you assigned to the Keithley Instruments PXI device during configuration.
[abc]	**counter:** is an array of strings that specifies the counter(s) the VI controls.
I16	**counter size:** This input is not used by Keithley Instruments PXI devices and is ignored.
TF	**start/restart:** is TRUE to configure and start the counter(s).
TF	**stop:** is TRUE to stop the counter(s).
U16	**source edge:** is the edge of the counter clock signal. 0: Count on low to high transition. 1: Count on high to low transition.

Table D-33 (continued)
KI Count Events or Time

U32	**count:** is the value of the counter at the time it is read. If there are two counters assigned to the task ID, the value of the higher order counter is multiplied by 10000 hex, shifting it to the left 16 bits. The higher order counter is then added to the value of the lower counter.
DBL	**seconds till overflow:** This input is not used by Keithley Instruments PXI devices and is ignored.
DBL	**seconds since last call:** This input is not used by Keithley Instruments PXI devices and is ignored.
DBL	**seconds since start:** This input is not used by Keithley Instruments PXI devices and is ignored.

KI Generate Delayed Pulse

Configures and starts a counter to generate a single pulse with the specified delay and pulse-width on the counter ***GPTCn_OUT*** pin. A single pulse consists of a delay phase (phase 1), followed by a pulse phase (phase 2), and then returns to the phase 1 level.

NOTE *This VI is not supported for Keithley KDIO series devices.*

Table D-34
KI Generate Delayed Pulse

I16	**device:** Number of the device (beginning from 1). The utility **Device Browser** can be used to get the information of current device configuration.
U16	**sub type:** is the sub-type of the device you assigned to the Keithley Instruments PXI device during configuration.
[abc]	**counter:** is an array of strings that specifies the counter(s) the VI controls.
DBL	**pulse delays (s or cycles):** is the desired duration of the first phase of the pulse, phase 1. The unit is seconds if timebase source is 0 (internal) and cycles if timebase source is 1 (external). If pulse delay is 0.0 and timebase source is 0, the VI selects a minimum delay of three cycles of the timebase used.
DBL	**pulse-width (s or cycles):** is the desired duration of the second phase of the pulse, phase 2. The unit is seconds if timebase source is 0 (internal) and cycles if timebase source is 1 (external). If pulse-width is 0.0 and timebase source is 0, the VI selects a minimum width of three cycles of the timebase used.
U16	**timebase source:** is the clock source. Timbebase source is 0 to use an internal signal and 1 to use an external signal for the timebase.

Table D-34 (continued)
KI Generate Delayed Pulse

U16	**gate mode:** specifies how the counter ***GPTCn_GATE*** signal is used. 0: Ungated/software start: ignore the gate source and start when the VI is called (default). 1: Count while the gate signal is TTL high. 2: Count while the gate signal is TTL low. 3: Start counting on the rising edge of the TTL gate signal. 4: Start counting on the falling edge of the TTL gate signal. 5: Restart counting on each rising edge of the TTL gate signal. 6: Restart counting on each falling edge of the TTL gate signal. Use gate mode 3 or 4 to generate one delayed pulse on the first gate edge after starting. Use gate mode 5 or 6 to generate a delayed pulse for each gate edge (i.e., retriggerable one-shot behavior).
U16	**pulse polarity:** is the polarity of the second phase (phase 2) of the pulse. 0: High pulse: phase 1 (the delay) is a low TTL level and phase 2 is a high level (default). 1: Low pulse: phase 1 is a high TTL level and phase 2 is a low level.
U32	**taskID of counter out:** is the task ID of the specified counter, which generates the delayed pulse.
DBL	**actual delay (s or cycles):** is the achieved delay. It may differ from the desired delay because the hardware has limited resolution and range.
	actual width (s or cycles): is the achieved pulse-width. It may differ from the desired width because the hardware has limited resolution and range.

KI Generate Pulse-Train

Configures the specified counter to generate a continuous pulse-train on the GPTCn_OUT pin. The signal has the prescribed frequency, duty cycle, and polarity. Each cycle of the pulse-train consist of a delay phase (phase 1) followed by a pulse phase (phase 2).

NOTE *This VI is not supported for Keithley DIO series devices.*

Table D-35
KI Generate Pulse-Train

I16	**device:** Number of the device (beginning from 1). The utility **Device Browser** can be used to get the information of current device configuration.
U16	**sub type:** is the sub-type of the device you assigned to the Keithley Instruments PXI device during configuration.
[abc]	**counter:** is an array of strings that specifies the counter(s) the VI controls.
I32	**number of pulses:** is the number of pulses you want in the pulse-train. If the value is 0 (default), the VI generates a continuous pulse-train.
DBL	**frequency:** (Hz) is the desired repetition rate of the pulse-train.
	duty cycle: is the desired ratio of the durations of phase 2 (phase two) of the pulse to the period of one cycle (1/frequency); default is 0.5. If duty cycle is 0.0 or 1.0, the VI computes the closest achievable duty cycle using a minimum period of three timebase cycles. A duty cycle very close to 0.0 or 1.0 may not be possible.

Table D-35 (continued)
KI Generate Pulse-Train

DBL	**timebase:** is the frequency of the clock. If the value of timebase is 0 or 10000000, internal signal is used; otherwise, an external signal is used.	
U16	**gate mode:** specifies how the counter ***GPTCn_GATE*** signal is used. 0: ungated/software start: ignore the gate source and start when the VI is called. (default). 1: Count while the gate signal is TTL high. 2: Count while the gate signal is TTL low. 3: start (continuous) pulse-train on the rising edge of the TTL gate signal. 4: start (continuous) pulse-train on the falling edge of the TTL gate signal. If number of pulses is 0 (continuous pulse-train), gate mode 3 or 4 generates one pulse per gate edge, which is the behavior of a retriggerable one shot. If number of pulses –1, gate mode 3 or 4 generates a continuous pulse-train.	
U16	**pulse polarity:** is the polarity of the second phase (phase 2) of the pulse. 0: High pulse: phase 1 (the delay) is a low TTL level and phase 2 is a high level (default). 1: Low pulse: phase 1 is a high TTL level and phase 2 is a low level.	
U32	**taskID of counter out:** is the task ID of the specified counter, which generates the pulse train.	
U32	**taskID of counter -1 out:** this output is not used by Keithley Instruments PXI devices and is ignored.	
	actual parameters out: is a cluster of lesser parameters. These parameters may differ from the desired parameters because the hardware has limited resolution and range.	
	DBL	**frequency:** (Hz) is the achieved frequency.
	DBL	**duty cycle:** is the achieved duty cycled.
	DBL	**pulse delay:** is the achieved minimum delay to the gating pulse.
	DBL	**pulse-width:** is the achieved width of the gating pulse.

KI Measure Pulse-Width or Period

Measures the pulse-width (length of time a signal is high or low) or period (length of time between adjacent rising or falling edges) of a TTL signal connected to the counter ***GPTCn_GATE*** pin. The method used gates an internal timebase clock with the signal being measured. This VI is useful in measuring the period or frequency (1/period) of relatively low frequency signals, when many timebase cycles occur during the gate.

NOTE *This VI is not supported for Keithley KDIO Series devices.*

Table D-36
KI Measure Pulse-Width or Period

I16	**device:** Number of the device (beginning from 1). The utility **Device Browser** can be used to get the information of current device configuration.

Table D-36 (continued)
KI Measure Pulse-Width or Period

U16	**sub type:** is the sub-type of the device you assigned to the Keithley Instruments PXI device during configuration.
[abc]	**counter:** is an array of strings that specifies the counter(s) the VI controls.
U16	**type of measurement:** identifies the type of pulse-width or period measurement to make. The following illustration demonstrates the various values for type of measurement. 0: Measure high pulse-width from rising to falling edge. 1: Measure low pulse-width from falling to rising edge. 2: Measure period between adjacent rising edges. (default) 3: Measure period between adjacent falling edges
DBL	**timebase:** is the frequency of the clock. If the value of timebase is 0 or 10000000, internal signal is used; otherwise, an external signal is used.
DBL	**pulse-width/period (s) out:** is the measured pulse-width or period; it equals count/timebase and may be valid or invalid.
SGL	**time limit in sec:** is the period to wait for a valid measurement. If time limit is -1.0 (default), the time limit is set to five seconds or four times the range of the counter at the selected timebase (4*65,536/timebase) in seconds.
TF	**valid?:** is TRUE if counter has not underflowed (if **count ?4**) or overflowed.
I32	**count:** is the value of the counter at the time it is read. For best accuracy, choose a timebase frequency that maximizes the count without overflowing it. If there are two counters assigned to the task ID, the value of the higher order counter is multiplied by 10000 hex, shifting it to the left 16 bits. The higher order counter is then added to the value of the lower counter.
TF	**counter overflow?:** is TRUE if counter reaches TC. Overflow does not produce an error.
TF	**timeout:** is TRUE if a valid reading is not within the prescribed or computed time limit. The timeout parameter does not produce an error.

Intermediate Counter VIs

KI Continuous Pulse Generator Config

Configures a counter to generate a continuous TTL pulse-train on its GPTCn_OUT pin.

NOTE *This VI is not supported for Keithley KDIO series devices.*

Table D-37
KI Continuous Pulse Generator Config

I16	**device:** Number of the device (beginning from 1). The utility **Device Browser** can be used to get the information of current device configuration.

Table D-37 (continued)
KI Continuous Pulse Generator Config

U16		**sub type:** is the sub-type of the device you assigned to the Keithley Instruments PXI device during configuration.
[abc]		**counter:** is the counter this VI controls.
		error in (no error): describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the error in cluster in error out.
	TF	**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
	I32	**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
	abc	**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.
DBL		**frequency:** (Hz) is the desired repetition rate of the pulse-train.
DBL		**duty cycle:** is the desired ratio of the durations of phase 2 (phase two) of the pulse to the period of one cycle (1/frequency); default is 0.5. If duty cycle is 0.0 or 1.0, the VI computes the closest achievable duty cycle using a minimum period of three timebase cycles. A duty cycle very close to 0.0 or 1.0 may not be possible.
DBL		**timebase:** is the frequency of the clock. If the value of timebase is 0 or 10000000 (10MHz), internal signal is used; otherwise, an external signal is used.
U16		**gate mode:** specifies how the counter ***GPTCn_GATE*** signal is used. 0: Ungated/software start: ignore the gate source and start when Counter Start VI is called (default). 1: Count while the gate signal is TTL high after the Counter Start VI is called. 2: Count while the gate signal is TTL low after the Counter Start VI is called. 3: Start counting on the rising edge of the TTL gate signal after the Counter Start VI is called. 4: Start counting on the falling edge of the TTL gate signal after the Counter Start VI is called. If gate mode is 3 or 4, the counter generates a single pulse on each edge.
U16		**pulse polarity:** is the polarity of the second phase (phase 2) of the pulse. 0: High pulse: phase 1 (the delay) is a low TTL level and phase 2 is a high level (default). 1: Low pulse: phase 1 is a high TTL level and phase 2 is a low level.
U32		**taskID:** is the task ID of the specified counter, which generates the pulse train.
		error out: contains error information. If the error in cluster indicated an error, the error out cluster contains the same information. Otherwise, error out describes the error status of this VI.
	TF	**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
	I32	**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.

Table D-37 (continued)
KI Continuous Pulse Generator Config

	abc	**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.
DBL		**actual frequency:** (Hz) is the achieved frequency. It may differ from the desired frequency because the hardware has limited resolution and range.
DBL		**actual duty cycle:** is the achieved duty cycled. It may differ from the desired duty cycle because the hardware has limited resolution and range.

KI Counter Divider Config

Configures the specified counter to divide a signal on the counter GP_TC_CLK pin or on an internal timebase signal using a count value called the timebase divisor. The result is that the signal on the counter GP_TC_OUT pin is equal to the frequency of the input signal/timebase divisor.

This VI is not supported for Keithley KDAQ series devices.

Table D-38
KI Counter Divider Config

I16	**device:** Number of the device (beginning from 1). The utility **Device Browser** can be used to get the information of current device configuration.
U16	**sub type:** is the sub-type of the device you assigned to the Keithley Instruments PXI device during configuration.
[abc]	**counter:** is the counter this VI controls.
U16	**gate mode:** specifies how the signal on the counter's GATE pin is used. 0: Ungated/software start: ignore the gate source and start when KI Counter Start VI is called (default). 1: Count while the gate signal is TTL high after the KI Counter Start VI is called.
U16	**source edge:** This input is not used by Keithley Instruments PXI devices and is ignored.
U16	**output:** This input is not used by Keithley Instruments PXI devices and is ignored.
U16	**updown source:** specifies how the signal on the counter's UPDN pin is used. 0: software control: ignore the UPDN source and control by **updownctrl** (default). 1: hardware control.
U16	**updown control:** specifies the specified counter to count down or count up if ***updown*** source is configured to be software control. 0: count down (default). 1: count up.

Table D-38 (continued)
KI Counter Divider Config

	**timebase:** (Hz) is set to the frequency of the internal signal whose cycles are counted, or is set to <=0.0 (default) to count the rising edges of the signal on the counter GP_TC_CLK pin.	
	**timebase divisor:** is the count down or divide value. For example, if the input frequency is 24000000 Hz, timebase divisor is 240000, and the output is pulsed, the frequency of the counter's GP_TC_OUT signal is 100 Hz.	
	**error in (no error):** describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the error in cluster in error out.	
		**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
		**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
		**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.
	**taskID:** identifies the group and the I/O operation.	
	**error out:** contains error information. If the error in cluster indicated an error, the error out cluster contains the same information. Otherwise, error out describes the error status of this VI.	
		**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
		**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
		**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.

KI Counter Read

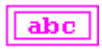
Reads the counter or counters identified by task ID.

NOTE *This VI is designed to read general purpose counter of Keithley KDAQ series devices.*

Table D-39
KI Counter Read

	**taskID in:** identifies the group and the I/O operation.	
	**error in (no error):** describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the error in cluster in error out.	
		**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.

Table D-39 (continued)
KI Counter Read

		**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
		**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.
	**counter list:** is the set of counters to read. Use this array only to read a subset of counters identified by task ID; otherwise, leave it empty. This input is only valid for KPXI-DAQ series devices.	
	**taskID out:** has the same value as ***taskID*** in.	
	**count:** is the value of the counter at the time it is read. If there are two counters assigned to the task ID, the value of the higher order counter is shifted to 16 bits to scale it, and then it is added to the value of the lower counter.	
	**error out:** contains error information. If the error in cluster indicated an error, the error out cluster contains the same information. Otherwise, error out describes the error status of this VI.	
		**Status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
		**Code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
		**Source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.
	**overflow:** This input is not used by Keithley Instruments PXI devices and is ignored.	

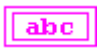
KI Counter Start

Starts the counters identified by task ID. This applies only to Keithley KDAQ series devices.

Table D-40
KI Counter Start

	**taskID in:** identifies the group and the I/O operation.	
	**error in (no error):** describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the error in cluster in error out.	
		**Status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
		**Code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.

Table D-40 (continued)
KI Counter Start

		**Source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.
	**counter list:** is the set of counters to read. Use this array only to read a subset of counters identified by task ID; otherwise, leave it empty. This input is only valid for Keithley KDAQ series devices.	
	**taskID out:** has the same value as ***taskID*** in.	
	**error out:** contains error information. If the error in cluster indicated an error, the error out cluster contains the same information. Otherwise, error out describes the error status of this VI.	
		**Status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
		**Code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
		**Source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.

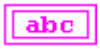
KI Counter Stop

Stops a count operation immediately or conditionally on an input error. This applies only to Keithley KDAQ series devices.

Table D-41
KI Counter Stop

	**taskID in:** identifies the group and the I/O operation.	
	**error in (no error):** describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the error in cluster in error out.	
		**Status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
		**Code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
		**Source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.
	**stop when:** This input is not used by Keithley Instruments PXI devices and is ignored.	

Table D-41 (continued)
KI Counter Stop

	**taskID out:** has the same value as ***taskID in***.	
	**error out:** contains error information. If the error in cluster indicated an error, the error out cluster contains the same information. Otherwise, error out describes the error status of this VI.	
		**Status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
		**Code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
		**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.

KI Delayed Pulse Generator Config

Configures a counter to generate a single pulse with the specified delay and pulse-width on the counter GPTCn_OUT pin.

NOTE *This VI is not supported for Keithley KDIO Series devices.*

Table D-42
KI Delayed Pulse Generator Config

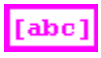
	**device:** Number of the device (beginning from 1). The utility **Device Browser** can be used to get the information of current device configuration.
	**sub type:** is the sub-type of the device you assigned to the Keithley Instruments PXI device during configuration.
	**counter:** is an array of strings that specifies the counter(s) the VI controls.
	**pulse delays (s or cycles):** is the desired duration of the first phase of the pulse, phase 1. The unit is seconds if timebase source is 0 (internal) and cycles if timebase source is 1 (external). If pulse delay is 0.0 and timebase source is 0, the VI selects a minimum delay of three cycles of the timebase used.
	**pulse-width (s or cycles):** is the desired duration of the second phase of the pulse, phase 2. The unit is seconds if timebase source is 0 (internal) and cycles if timebase source is 1 (external). If pulse-width is 0.0 and timebase source is 0, the VI selects a minimum width of three cycles of the timebase used.
	**timebase source:** is the clock source. Timbebase source is 0 to use an internal signal and 1 to use an external signal (from ***GPTCn_SRC*** pin) for the timebase.

Table D-42 (continued)
KI Delayed Pulse Generator Config

	**gate mode:** specifies how the counter ***GPTCn_GATE*** signal is used. 0: Ungated/software start: ignore the gate source and start when Counter Start VI is called (default).. 1: Count while the gate signal is TTL high after the Counter Start VI is called. 2: Count while the gate signal is TTL low after the Counter Start VI is called. 3: Start counting on the rising edge of the TTL gate signal after the Counter Start VI is called. 4: Start counting on the falling edge of the TTL gate signal after the Counter Start VI is called. Use gate mode 3 or 4 to generate one delayed pulse on the first gate edge after starting.	
	**pulse polarity:** is the polarity of the second phase (phase 2) of the pulse. 0: High pulse: phase 1 (the delay) is a low TTL level and phase 2 is a high level (default). 1: Low pulse: phase 1 is a high TTL level and phase 2 is a low level.	
	**error in (no error):** describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the error in cluster in error out.	
		**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
		**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
		**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.
	**taskID:** identifies the group and the I/O operation.	
	**error out:** contains error information. If the error in cluster indicated an error, the error out cluster contains the same information. Otherwise, error out describes the error status of this VI.	
		**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
		**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
		**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.
	**actual delay (s or cycles):** is the achieved delay. It may differ from the desired delay because the hardware has limited resolution and range.	
	**actual width (s or cycles):** is the achieved pulse-width. It may differ from the desired width because the hardware has limited resolution and range.	

KI Down Counter or Divider Config

Configures the specified counter to count down or divide a signal on the counter ***GPTCn_SRC*** pin or on an internal timebase signal using a count value called the timebase divisor. The result is that the signal on the counter GPTCn_OUT pin is equal to the frequency of the input signal/timebase divisor.

NOTE *This VI is not supported for Keithley KDIO Series devices.*

Table D-43
KI Down Counter or Divider Config

I16		**device:** Number of the device (beginning from 1). The utility **Device Browser** can be used to get the information of current device configuration.
U16		**sub type:** is the sub-type of the device you assigned to the Keithley Instruments PXI device during configuration.
[abc]		**counter:** is an array of strings that specifies the counter(s) the VI controls.
DBL		**timebase:** (Hz) is set to the frequency of the internal signal whose cycles are counted, or is set to <=0.0 (default) to count the rising edges of the signal on the counter ***GPTCn_SRC*** pin.
I32		**timebase divisor:** is the count down or divide value. For example, if the input frequency is 10000000 Hz, timebase divisor is 100000, and the output is pulsed, the frequency of the counter's GPTCn_OUT signal is 100 Hz.
U16		**gate mode:** specifies how the counter ***GPTCn_GATE*** signal is used. 0: Ungated/software start: ignore the gate source and start when Counter Start VI is called (default).. 1: Count while the gate signal is TTL high after the Counter Start VI is called. 2: Count while the gate signal is TTL low after the Counter Start VI is called. 3: Start counting on the rising edge of the TTL gate signal after the Counter Start VI is called. 4: Start counting on the falling edge of the TTL gate signal after the Counter Start VI is called.
U16		**source edge:** is the edge of the counter clock signal. 0: Count on low to high transition. 1: Count on high to low transition.
U16		**output:** is the behavior of the output signal when counter reaches TC. 0: High pulse lasting one cycle of the source or timebase signal (default). 1: Low pulse lasting one cycle of the source or timebase signal. 2: High toggle lasting until the next TC. 3: Low toggle lasting until the next TC. The effect of output modes 0 and 1 is to divide-down the source of timebase frequency by the timebase divisor. The effect of output modes 2 and 3 is to divide the frequency by twice the timebase divisor.
		error in (no error): describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the error in cluster in error out.
	TF	**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
	I32	**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.

Table D-43 (continued)
KI Down Counter or Divider Config

		**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.
	**taskID:** identifies the group and the I/O operation.	
	**error out:** contains error information. If the error in cluster indicated an error, the error out cluster contains the same information. Otherwise, error out describes the error status of this VI.	
		**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
		**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
		**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.

KI Event or Time Counter Config

Configures one or two counters to count external events. An external event is a high or low signal transition on the specified ***GPTCn_SRC*** pin of the counter.

NOTE *This VI is not supported for Keithley KDIO Series devices.*

Table D-44
KI Event or Time Counter Config

	**device:** Number of the device (beginning from 1). The utility **Device Browser** can be used to get the information of current device configuration.
	**sub type:** is the sub-type of the device you assigned to the Keithley Instruments PXI device during configuration.
	**counter:** is an array of strings that specifies the counter(s) the VI controls.
	**count limit:** this input is not used by Keithley Instruments PXI devices and is ignored.
	**gate mode:** specifies how the counter ***GPTCn_GATE*** signal is used. 0: Ungated/software start: ignore the gate source and start when Counter Start VI is called (default).. 1: Count while the gate signal is TTL high after the Counter Start VI is called. 2: Count while the gate signal is TTL low after the Counter Start VI is called. 3: Start counting on the rising edge of the TTL gate signal after the Counter Start VI is called. 4: Start counting on the falling edge of the TTL gate signal after the Counter Start VI is called.
	**counter size:** is not used by Keithley Instruments PXI devices and is ignored.

Table D-44 (continued)
KI Event or Time Counter Config

	**error in (no error):** describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the error in cluster in error out.	
		**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
		**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
		**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.
	**source edge:** is the edge of the counter clock signal. 0: Count on low to high transition. 1: Count on high to low transition.	
	**event source/timebase:** (Hz) is set to the frequency of the internal signal whose cycles are counted, or is set to <=0.0 (default) to count the rising edges of the signal on the counter ***GPTCn_SRC*** pin.	
	**taskID:** identifies the group and the I/O operation.	
	**error out:** contains error information. If the error in cluster indicated an error, the error out cluster contains the same information. Otherwise, error out describes the error status of this VI.	
		**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
		**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
		**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.

KI Pulse-Width or Period Measurement Config

Configures the specified counter to measure the pulse-width or period of a TTL signal connected to its ***GPTCn_GATE*** pin.

NOTE *This VI is not supported for Keithley KDIO Series devices.*

Table D-45
KI Pulse-Width or Period Measurement Config

	**device:** Number of the device (beginning from 1). The utility **Device Browser** can be used to get the information of current device configuration.	
	**sub type:** is the sub-type of the device you assigned to the Keithley Instruments PXI device during configuration.	
	**counter:** is an array of strings that specifies the counter(s) the VI controls.	
	**timebase:** (Hz) is set to the frequency of the internal signal whose cycles are counted, or is set to <=0.0 (default) to count the rising edges of the signal on the counter ***GPTCn_SRC*** pin.	
	**type of measurement:** identifies the type of pulse-width or period measurement to make. The following illustration demonstrates the various values for type of measurement. 0: Measure high pulse-width from rising to falling edge. 1: Measure low pulse-width from falling to rising edge. 2: Measure period between adjacent rising edges. (default) 3: Measure period between adjacent falling edges	
	**error in (no error):** describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the error in cluster in error out.	
		**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
		**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
		**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.
	**taskID:** identifies the group and the I/O operation.	
	**error out:** contains error information. If the error in cluster indicated an error, the error out cluster contains the same information. Otherwise, error out describes the error status of this VI.	
		**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
		**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
		**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.

KI UpDown Counter Config

Configures one counter to count edges in the signal on the specified counter's SOURCE pin or the number of cycles of a specified internal timebase signal.

Table D-46
KI UpDown Counter Config

I16	**device:** Number of the device (beginning from 1). The utility **Device Browser** can be used to get the information of current device configuration.
U16	**sub type:** is the sub-type of the device you assigned to the Keithley Instruments PXI device during configuration.
[abc]	**counter:** is the counter this VI controls.
U16	**gate mode:** specifies how the signal on the counter's GATE pin is used. 0: Ungated/software start: ignore the gate source and start when Counter Start VI is called (default). 1: Count while the gate signal is TTL high after the KI Counter Start VI is called.
U16	**source edge:** This input is not used by Keithley Instruments PXI devices and is ignored.
U16	**output:** This input is not used by Keithley Instruments PXI devices and is ignored.
U16	**updown source:** specifies how the signal on the counter's UPDN pin is used. 0: software control: ignore the UPDN source and control by updown control (default). 1: hardware control.
U16	**updown control:** specifies the specified counter to count down or count up if updown source is configures to be software control. 0: count down (default). 1: count up.
DBL	**timebase:** (Hz) is set to the frequency of the internal signal whose cycles are counted, or is set to <=0.0 (default) to count the rising edges of the signal on the counter SOURCE pin.
	error in (no error): describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the error in cluster in error out.
	TF **status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
	I32 **code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
	abc **source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.
I32	**count:** is the count value.
U32	**taskID:** identifies the group and the I/O operation.

Table D-46 (continued)
KI UpDown Counter Config

	**error out:** contains error information. If the error in cluster indicated an error, the error out cluster contains the same information. Otherwise, error out describes the error status of this VI.	
		**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
		**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
		**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.

Advanced Counter VIs

KI ICTR Control

This VI control counters on the Keithley Instruments PXI devices that use 82C54 chip. Control operations include starting, stopping, and setting the state of active acquisitions.

NOTE *This VI is not supported for Keithley KDAQ series devices.*

Table D-47
KI ICTR Control

	**device:** Number of the device (beginning from 1). The utility **Device Browser** can be used to get the information of current device configuration.
	**sub type:** is the sub-type of the device you assigned to the Keithley Instruments PXI device during configuration.
	**counter:** is the counter this VI controls. KPXI-DIO-48: 0, 1 or 2
	control code: 0: Setup mode 0 – Toggle output from low to high on TC (default). 1: Setup mode 1 – Programmable one-shot. 2: Setup mode 2 – Rate generator. 3: Setup mode 3 – Square wave rate demerara. 4: Setup mode 4 – Software-triggered strobe. 5: Setup mode 5 – Hardware-triggered strobe. 6: Read. 7: Reset.

Table D-47 (continued)
KI ICTR Control

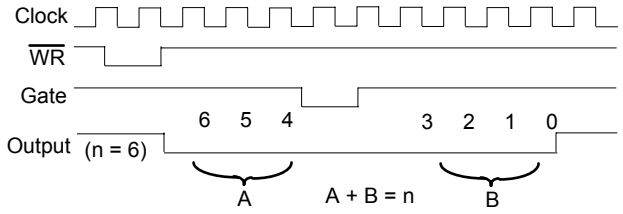
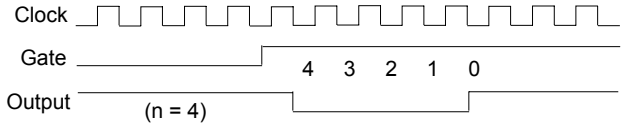
U16	**count:** is the period between output pulses. If control code is 0, 1, 4, or 5, count can be 0 through 65,535 in binary counter operation and 0 through 9,999 in binary-coded decimal (BCD) counter operation. If control code is 2 or 3, count can be 2 through 65,535 and 0 in binary counter operation and 2 through 9,999 and 0 in BCD counter operation. **Setup mode 0: Toggle output from low to high on terminal count.** In this mode, as shown in the figure below, the output goes low after the mode set operation, and the counter begins to count down while the gate input is high. When terminal count is reached, the output goes high and remains high until the selected counter is set to a different mode. 

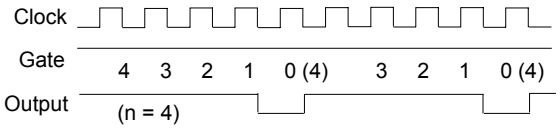
Table D-47 (continued)
KI ICTR Control

count: (continued)

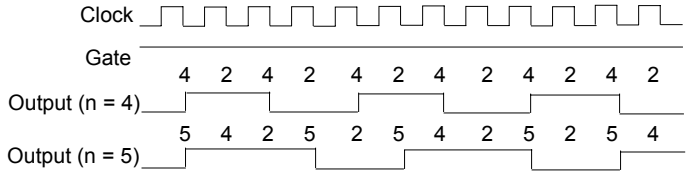
Setup mode 1: Programmable one-shot
In this mode, as shown in the figure below, the **Output** goes low following the rising edge of **Gate** input ***and*** the falling edge of the clock. The **Output** and goes high on terminal count.



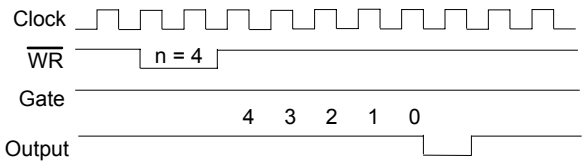
Setup mode 2: Rate generator
In this mode, the output goes low for one period of the clock input. **count** indicates the period from one output pulse to the next.



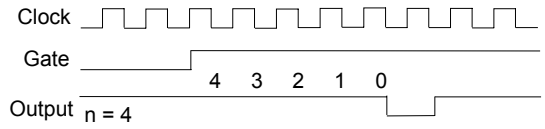
Setup mode 3: Square wave rate generator
In this mode, the output stays high for one half of the **count** clock pulses and stays low for the other half.



Setup mode 4: Software-triggered strobe
In this mode, the output is initially high, and the counter begins to count down while the gate input is high. On terminal count, the output goes low for one clock pulse, then goes high again. The following diagram shows the SOFT_TRIG mode timing diagram.



Setup mode 5: Hardware-triggered strobe
This mode is similar to Setup mode 4 except that the gate input is used as a trigger to start counting. The following diagram shows the HARD_TRIG mode timing diagram.



U16

output state: is only valid when control code is 7 (reset).
0: Low (default input).
1: High.

Table D-47 (continued)
KI ICTR Control

U16	**binary or bcd:** controls whether the counter operates as a 16-bit binary counter or as a 4-decade BCD counter. 0: ***BinBcd.*** 1: 16-bit binary counter (default input)	
	Keithley Instruments PXI extensions: additional features for Keithley Instruments PXI devices.	
	U16	**clock source:** defines the clock source for the timer/counter. 0: ECK1 (default input) 1: COUT n-1 2: CK1 3: COUT 10
	error in (no error): describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the error in cluster in error out.	
	TF	**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
	I32	**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
	abc	**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.
U16	**read value:** When you set control code to 6 (read), read value returns the value the VI read from the counter.	
U32	**taskID:** has the same value as ***taskID in***.	
	error out: contains error information. If the error in cluster indicated an error, the error out cluster contains the same information. Otherwise, error out describes the error status of this VI.	
	TF	**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
	I32	**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
	abc	**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.

Calibration and Configuration VIs

Calibration VIs

KI KPXI-DAQ series devices and Digitizer Series Calibrate

Use this VI to calibrate KPXI-DAQ series devices and Digitizer Series device and to select a set of calibration constants to be used by KIDAQ LabVIEW.

NOTE *This VI is not supported for Keithley KDIO devices.*

Table D-48
KI KPXI-DAQ series devices and Digitizer Series Calibrate

U32		**taskID in:** identifies the group and the I/O operation.
error		**error in (no error):** describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the error in cluster in error out.
	TF	**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
	I32	**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
	abc	**Source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.
U16		**operation:** determines the operation the VI performs. 0: No change (default input). 1: Set default load area (default setting). 2: Self-calibrate. Setting the default load area, or value 1, does not perform a calibration; it sets the default load area to the area specified by calibration constants. Self-calibrate, or value 2, performs a calibration using the internal voltage reference.
U16		**calibration constants:** specifies which set of calibration constants KIDAQ LabVIEW uses. 0: No change (default input). 1: Factory EEPROM area, i.e. Bank 0 (default setting). 2: EEPROM Bank 0 area. 3: EEPROM Bank 1 area. 4: EEPROM Bank 2 area. 5: EEPROM Bank 3 area.
DBL		**reference voltage:** this input is not used by Keithley Instruments PXI devices and is ignored.
U32		**taskID out:** has the same value as ***taskID in***.
error		**error out:** contains error information. If the error in cluster indicated an error, the error out cluster contains the same information. Otherwise, error out describes the error status of this VI.
	TF	**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.

Table D-48 (continued)
KI KPXI-DAQ series devices and Digitizer Series Calibrate

		**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
		**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.

Other Calibration and Configuration VIs

KI Route Signal

Use this VI to route an internal signal to the specified I/O connector or SSI bus line, or to enable clock sharing through the SSI bus clock line.

NOTE *This VI is not supported for Keithley KDIO devices.*

Table D-49
KI Route Signal

	**taskID in:** identifies the group and the I/O operation.	
	**error in (no error):** describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the error in cluster in error out.	
		**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
		**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
		**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.
	**signal name:** allows you to select the SSI line. The valid signal name for KPXI-DAQ series devices are as follows: 0 : Do not change signal name (default input). 1 : AI conversion. 2 : AO update. 3 : AI trigger. 4 : AO trigger. 19 : SSI Clock. The valid signal names for KPXI-AI-2-65M are the following: 0 : Do not change signal name (default input). 3 : AI trigger. 19 : SSI Clock.	
	**signal name line number:** this input is not used by Keithley Instruments PXI devices and is ignored.	

Table D-49 (continued)
KI Route Signal

U16	**signal source:** is the signal that KIDAQ LabVIEW routes to the location designated in signal name. There is only one valid signal source for most signal names. The valid signal source for KPXI-DAQ series devices are the following: 0 : Do not change signal source (default input). 1 : None (default setting). 2 : AI Start Trigger. 3 : AI Stop Trigger. 4 : AI Convert. 7 : AO Update. 8 : AO Start Trigger. 21 : Board Clock The valid signal sources for KPXI-AI-2-65M devices are the following: 0 : Do not change signal source (default input). 1 : None (default setting). 2 : AI Start Trigger. 3 : AI Stop Trigger. 21 : Board Clock	
U32	**signal source line number:** this input is not used by Keithley Instruments PXI devices and is ignored.	
U32	**taskID out:** has the same value as ***taskID in***.	
	error out: contains error information. If the error in cluster indicated an error, the error out cluster contains the same information. Otherwise, error out describes the error status of this VI.	
	TF	**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
	I32	**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
	abc	**Source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.

KI SSI Control

Connects or disconnects trigger and timing signals between DAQ devices along the Real-Time System Integration (SSI) bus.

NOTE *This VI is not supported for Keithley KDIO Series devices.*

Table D-50
KI SSI Control

U32	**taskID in:** identifies the group and the I/O operation.

Table D-50 (continued)
KI SSI Control

U16	**board signal:** allows you to select the SSI line. The valid signal name for KPXI-DAQ series devices are as follows: 0 : AI conversion. 1 : AO update. 2 : AI trigger. 3 : AO trigger. 4 : Board Clock. 5 : AI Start The valid signal name for KPXI-AI-2-65M are the following: 2 : AI trigger. 4 : Board Clock.
U16	**trigger line:** this input is not used by Keithley Instruments PXI devices and is ignored.
U16	***direction***: 1 : The board transmits the signal to the bus.
U16	***control code:*** 0 : Do not change the control code setting (default input). 1 : Clear. 2 : Connect (default input). 3 : Disconnect. 4 : Construct the trigger line ***usemap*** only.
U16	**device out:** has the same value as device.
I32	**status:** This input is not used by KPXI-DAQ series devices and is ignored.
[I16]	**trigger line usemap:** provides a list of free and busy SSI trigger lines. If trigger line i is not busy, ***trigger line usemap[i]*** shows a value of 0. If ***trigger line i*** is busy, the VI sets ***trigger line usemap[i]*** to the device number of the device driving the line. Making only a receive connection to trigger line i does not set the ***[i]th*** element of trigger line ***usemap***.

Service VIs

KI Error Handler

The KI Error Handler VI explains a non-zero error codes and shows dialog box with information about error. An error code equaling 0 (zero) means no error occurred.

Table D-51
KI Error Handler

		error in (no error): describes error conditions occurring before the VI executes. If an error has already occurred, the VI returns the value of the error in cluster in error out.
	TF	**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.

Table D-51 (continued)
KI Error Handler

		**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
		**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.
	**error out:** contains error information. If the error in cluster indicated an error, the error out cluster contains the same information. Otherwise, error out describes the error status of this VI.	
		**status:** is TRUE if an error occurred. If status is TRUE, the VI does not perform any operations.
		**code:** is the error code number identifying an error. A value of 0 means no error, a negative value means a fatal error, and a positive value is a warning. Refer to Table D-52 for a code description.
		**source:** identifies where an error occurred. The source string is usually the name of the VI that produced the error.

Error Codes

The Error Codes for KIDAQ LabVIEW VIs are contained in Table D-52.

Table D-52
Error Codes: KIDAQ LabVIEW VIs

Code	Name	Description
0	NoError	No error occurred
-10006	badLineError	The line is invalid
-10007	badChanError	The value of AI/AO channel or DI/O port is invalid.
-10008	badGroupError	The group is invalid.
-10009	badCounterError	The value of input terminal **Ctr** is out of range.
-10010	badCountError	The value of input terminal **State** is out of range.
-10012	badRangeError	The specified A/D or D/A voltage value is out of range.
-10019	badClkFrequencyError	The frequency is invalid.
-10025	limitsOutOfRangeError	The value of AI range is invalid.
-10026	badBufferSpecificationError	The requested number of buffers or the buffer size is not allowed.
-10027	badDAQEventError	The DAQ events could not be opened.
-10041	badTaskIDError	The specified task ID is invalid.

Table D-52 (continued)
Error Codes: KIDAQ LabVIEW VIs

Code	Name	Description
-10081	badPretrigCountError	The pretrigger sample count is invalid.
-10083	badTrigModeError	The trigger mode is invalid.
-10084	badTrigCountError	The trigger count is invalid.
-10086	badExtRefError	The external reference is invalid.
-10087	badTrigTypeError	The trigger type is invalid.
-10088	badTrigLevelError	The trigger level is invalid.
-10089	badTotalCountError	The DMA or interrupt transfer size is larger than the memory allocated in driver.
-10100	badPortWidthError	The requested digital port width is not a multiple of the hardware port width or is not attainable by the DAQ hardware.
-10121	gpctrBadCtrNumberError	Invalid ***counterNumber*** used.
-10122	gpctrBadParamValueError	Invalid ***paramValue*** used.
-10240	noDriverError	Open device driver failed.
-10242	functionNotFoundError	The function is not supported by this type of card.
-10341	badConnectError	The SSI signal/line cannot be connected as specified.
-10370	badScanListError	The scan list is invalid.
-10401	unknownDeviceError	The specified device is not a Keithley Instruments PXI device, the driver does not support the device.
-10402	deviceNotFoundError	No device is located in the specified slot or the device number is invalid.
-10409	groupBusyError	The specified group is in use.
-10411	counterBusyError	The specified counter is in use.
-10444	memFullError	Fail to allocate a driver internal use memory.
-10604	activeWriteError	Once data generation has started, only the transfer buffers originally written to may be updated.
-10608	noTransferInProgError	No transfer is in progress for the specified resource.
-10609	transferInProgError	A transfer is already in progress for the specified resource, or the operation is not allowed because the device is in the process of performing transfers, possibly with different resources.

Table D-52 (continued)
Error Codes: KIDAQ LabVIEW VIs

Code	Name	Description
-10612	badLineDirError	The specified line does not support the specified transfer direction.
-10613	badChanDirError	The specified channel does not support the specified transfer direction, or you have performed an operation on a digital port or line configured for the opposite direction.
-10618	badClkSrcError	The specified source signal cannot be assigned to the clock resource.
-10621	badTrigError	The specified trigger signal cannot be assigned to the trigger resource.
-10629	invalidOpModeError	The specified operating mode is invalid, or the resources have not been configured for the specified operating mode.
-10631	noInfiniteModeError	Continuous input or output transfers are not allowed in the current operating mode, or continuous operation is not allowed for this type of device.
-10634	noContTransferInProgressError	No continuous (double buffered) transfer is in progress.
-10636	noContWithSynchError	You cannot start a continuous (double-buffered) operation with a synchronous function call.
-10681	badChanRangeError	All channels of this board must have the same range.
-10697	rateNotSupportedError	The value of input terminal **SampleRate** is invalid.
-10800	timeOutError	The operation could not complete within the time limit.
-10801	calibrationError	An error occurred during the calibration process.
-10810	internalDriverError	An unexpected error occurred inside the driver when performing this given operation.
-10849	Unable to open a file	Fail to open a data file for storing input data.
-10856	osError	An unexpected error occurred from the operating system while performing the given operation.

AI Range Codes

The Analog Input Range for Keithley Instruments PXI devices are contained in Table D-53 and Table D-54:

Table D-53
Analog Input Range

Item	Range
1	Bipolar -10V to +10V

Table D-53 (continued)
Analog Input Range

Item	Range
2	Bipolar -5V to +5V
3	Bipolar -2.5V to +2.5V
4	Bipolar -1.25V to +1.25V
5	Bipolar -0.625V to +0.625V
6	Bipolar -0.3125V to +0.3125V
7	Bipolar -0.5V to +0.5V
8	Bipolar -0.05V to +0.05V
9	Bipolar -0.005V to +0.005V
10	Bipolar -1V to +1V
11	Bipolar -0.1V to +0.1V
12	Bipolar -0.01V to +0.01V
13	Bipolar -0.001V to +0.001V
14	Unipolar 0 to +20V
15	Unipolar 0 to +10V
16	Unipolar 0 to +5V
17	Unipolar 0 to +2.5V
18	Unipolar 0 to +1.25V
19	Unipolar 0 to +1V
20	Unipolar 0 to +0.1V
21	Unipolar 0 to +0.01V
22	Unipolar 0 to +0.001V
23	Bipolar -2V to +2V
24	Bipolar –0.25V to +0.25V
25	Bipolar –0.2V to +0.2V
26	Unipolar 0 to +4V
27	Unipolar 0 to +2V

Table D-53 (continued)
Analog Input Range

Item	Range
28	Unipolar 0 to +0.5V
29	Unipolar 0 to +0.4V

Table D-54
Valid analog input ranges (specified by module)

Model	Range
KPXI-SDAQ-4-500K KPXI-SDAQ-4-2M KPXI-DAQ-64-500K KPXI-DAQ-64-250K	1, 2, 3, 4, 15, 16, 17, 18
KPXI-DAQ-64-3M KPXI-DAQ-96-3M	1, 2, 3, 4, 7, 8, 10, 15, 16, 17, 19, 20, 23, 24, 25, 26, 27, 28, 29
KPXI-AO-4-1M KPXI-AO-8-1M	1, 15
KPXI-AI-2-65M	2, 10

AI Data Format

Table D-55
Analog Input data format (by Model)

Model	AI Data Format
KPXI-SDAQ-4-2M	16-bit signed integer data: D13 D12 D11 ..... D1 D0 b1 b0 Where: D13, D12, ... , D0 : A/D converted data b1, b0 : Simultaneous Digital Input data.
KPXI-SDAQ-4-500K	16-bit unsigned integer data: D15 D14 D13 ..... D1 D0 Where: D15, D14, ... , D0 : A/D converted data
KPXI-DAQ-64-3M	16-bit signed integer data: D11 D10 D9 ..... D1 D0 b3 b2 b1 b0 Where: D11, D10, ... , D0 : A/D converted data b3, b2, b1, b0 : Simultaneous Digital Input data.
KPXI-DAQ-96-3M	16-bit signed integer data: D11 D10 D9 ..... D1 D0 b3 b2 b1 b0 Where: D11, D10, ... , D0 : A/D converted data b3, b2, b1, b0 : not used.
KPXI-DAQ-64-500K KPXI-DAQ-64-250K	16-bit signed integer data: D15 D14 D13 ..... D1 D0 Where: D15, D14, ... , D0 : A/D converted data
KPXI-AO-4-1M KPXI-AO-8-1M	16-bit signed integer data: D13 D12 D11 ..... D1 D0 b1 b0 Where D13, D12, ... , D0 : A/D converted data b1, b0 : AI Auto-scan Channel.
KPXI-AI-2-65M	16-bit unsigned integer data: b15 b14 D13 D12 D11 .....….. D1D0 Where: D13, D12, ... , D0 : A/D converted data b14 : Over-voltage indicator

KEITHLEY

Service Form

Model No. _____ **Serial No.** _____ **Date** _____

Name and Telephone No. _____

Company _____

List all control settings, describe problem and check boxes that apply to problem. _____

_____

_____

❒ Intermittent ❒ Analog output follows display ❒ Particular range or function bad; specify

_____

❒ IEEE failure ❒ Obvious problem on power-up ❒ Batteries and fuses are OK

❒ Front panel operational ❒ All ranges or functions are bad ❒ Checked all cables

Display or output (check one)

❒ Drifts ❒ Unable to zero ❒ Unstable

❒ Overload ❒ Will not read applied input

❒ Calibration only ❒ Certificate of calibration required ❒ Data required

(attach any additional sheets as necessary)

Show a block diagram of your measurement including all instruments connected (whether power is turned on or not). Also, describe signal source.

Where is the measurement being performed? (factory, controlled laboratory, out-of-doors, etc.) _____

_____

What power line voltage is used? _____ Ambient temperature? _____ °F

Relative humidity? _____ Other? _____

Any additional information. (If special modifications have been made by the user, please describe.)

_____

_____

Be sure to include your name and telephone number on this service form.

KEITHLEY

A GREATER MEASURE OF CONFIDENCE

Keithley Instruments, Inc.

Corporate Headquarters • 28775 Aurora Road • Cleveland, Ohio 44139 • 440-248-0400 • Fax: 440-248-6168 • 1-888-KEITHLEY • www.keithley.com

12/06