DIO-64 User Manual

Intelligent Digital I/O System

Version 1.04

VIEWPOINT
SYSTEMS

Viewpoint Systems, Inc. does not warrant that the Program will meet Customer's requirements or will operate in the combinations which may be selected by the Customer or that the operation of the Program will be uninterrupted or error free or that all Program defects will be corrected.

VIEWPOINT SYSTEMS, INC. DOES NOT AND CANNOT WARRANT THE PERFORMANCE OR RESULTS THAT MAY BE OBTAINED BY USING THIS SOFTWARE. ACCORDINGLY, THE SOFTWARE AND ITS DOCUMENTATION ARE SOLD "AS IS" WITHOUT WARRANTY AS TO THEIR PERFORMANCE, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE RESULTS AND PERFORMANCE OF THE PROGRAM IS ASSUMED BY YOU.

NEITHER VIEWPOINT SYSTEMS, INC. NOR ANYONE ELSE WHO HAS BEEN INVOLVED IN THE CREATION, PRODUCTION, OR DELIVERY OF THIS SOFTWARE SHALL BE LIABLE FOR ANY DIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, SUCH AS, BUT NOT LIMITED TO, LOSS OF ANTICIPATED PROFITS OR BENEFITS, RESULTING FROM THE USE OF THE PROGRAM OR ARISING OUT OF ANY BREACH OF ANY WARRANTY. SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF DIRECT INCIDENTAL OR CONSEQUENTIAL DAMAGES, FOR THE ABOVE MAY NOT APPLY TO YOU.

Viewpoint Systems Inc.
800 West Metro Parkway
Rochester, NY 14623
June 2003
VS-DIO64-01032

Table of Contents

Overview

The DIO-64 provides for 64 bits of TTL compatible digital I/O for a wide variety of digital applications. The PCI-DIO-64 is available PCI compatible systems and the PXI-DIO-64 is available for CompactPCI/PXI form factors. The strength of the DIO-64 comes from the ability to deal with digital data acquisition on the basis of signal changes. Sampling digital inputs at high rates can generate a great deal of data, most of which is redundant. The DIO-64 only records digital input scans when a change has been detected on certain bits. Likewise, the DIO-64 allows you to specify digital output waveforms as concisely as possible.

DIO-64 features include:

- High speed — the DIO-64 is designed to measure time intervals as short as 50 nanoseconds.
- Intelligence — The heart of the DIO-64 is an Altera Apex20k FPGA with a custom program.
- RTSI/PXI — allows for multi-board triggers and clocks
- Compatibility — the DIO-64 is compatible with the DIO-128
- Driver software — drivers and LabVIEW VIs provided
- Examples — LabVIEW examples show how to make the most of the DIO-64

System Requirements

IBM compatible PC with available PCI slots

OR

PXI or CompactPCI chassis with IBM compatible controller and available slots

Microsoft Windows 2000 or XP operating system

LabVIEW 6i (6.0) or higher

Getting Started

Installing the Software

NOTE: **Please install the software before installing the DIO-64 board in your computer. If the DIO-64 driver software is installed first, the Operating System (OS) will correctly recognize the DIO-64 board and link it to the driver.**

You must have the sufficient access rights to the computer in order to correctly install the DIO-64 driver software. This typically means that you have *administrator* rights or the equivalent.

The DIO-64 software is installed through an installation program found on the accompanying CD-ROM. The installation program will run automatically when the CD-ROM is inserted into the CD-ROM drive. If *AutoPlay* has been disabled on your computer, you can manually start the installation program by running the setup.exe program found in the CD-ROM's root directory.

Software Updates

Software updates will be made available through the Viewpoint Systems web site. Check the DIO-64 web page at http://www.ViewpointUSA.com/dio64 periodically for the latest driver software and new examples.

Installing the Hardware

It is important to follow appropriate electrostatic precautions when handling the DIO-64 board. The board is shipped in an electrostatic bag. Be sure that you have removed any electrostatic hazard by attaching a grounding wrist strap or touching the computer chassis before removing the DIO-64 from its bag. Save the bag for future use when the DIO-64 is removed from the computer system.

There are some pullup / pulldown resistor packs that effect the operation of the DIO-64 digital I/O ports. Please read the *Digital Input/Outputs Section (page16)* if your application needs these resistors in a particular configuration. Make any changes necessary before installing the board. This can be done at any time, but the board will need to be removed from the PC in order to change the resistor pack configuration.

PCI

1. remove the cover from the PC
2. select an open PCI slot
3. install the PCI-DIO-64 board
4. screw the DIO-64 in place
5. replace the cover

CompactPCI/PXI

1. select the CompactPCI/PXI slot
2. remove the cover plate, if any
3. push the CompactPCI/PXI ejector handle down (in the insert position)
4. carefully slide the PXI-DIO-64 board into the slot
5. make sure that the safety screw in the top corner is not binding up on the chassis
6. raise the ejector handle, allowing it to pull the board the rest of the way into the slot.
7. tighten the safety screw

Restart the system

The system will discover your newly installed DIO-64 card the next time it is started. The Plug and Play Manager will associate the DIO-64 card with its driver. During this process you may see the following dialog box:

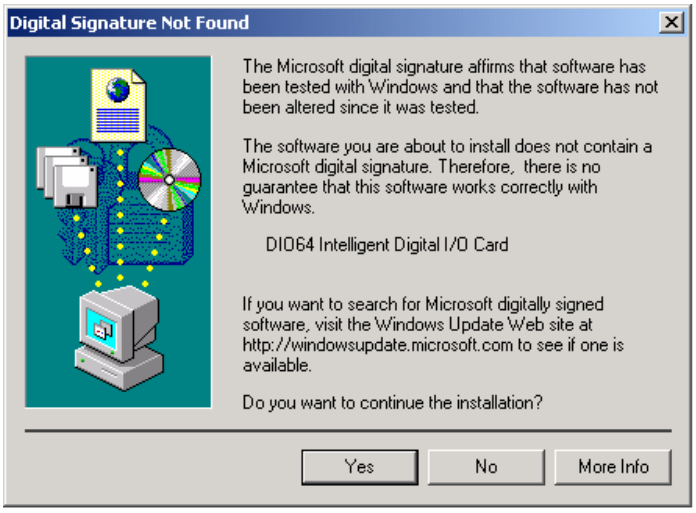


Figure 1 - Digital Signature Dialog

Please indicate that “yes”, you do want to continue the installation.

System Setup Verification

The DIO-64 has a green LED to show that power has been applied. Once the PC is powered on, this LED should illuminate showing that the DIO-64 has power.

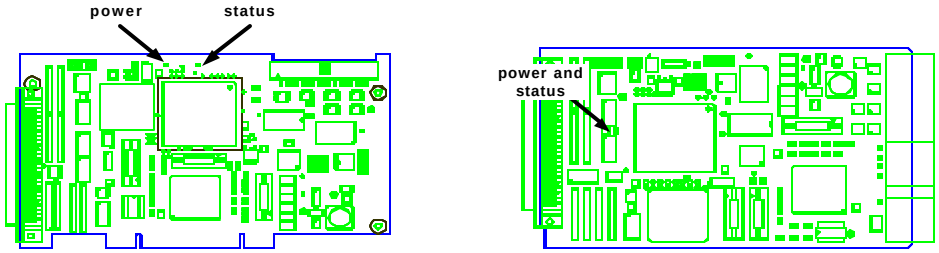


Figure 2 - PCI and PXI LED locations

When installing multiple instances of the same PCI card, it is not always clear which board is seen by the system as board #1, #2, etc. The order is typically determined by the order in which the OS discovers the board's presence. The order will typically be the same on a particular computer, but may vary between different brands or models of computers.

The DIO-64 makes this identification easier by blinking a yellow LED on the board to help identify how the system has assigned board #s. The yellow LED will blink the appropriate number of times for the board # the OS has assigned. For example board #2 will blink twice, pause, and then blink twice again, over and over. This blinking begins at the point the OS has started the DIO-64 device driver during the boot process. The blinking will continue until an application loads the DIO-64 FPGA control program, after which the LED will blink steadily.

When the installation is completed, you can test your setup. Perform the steps below to verify that your system is properly installed.

1. Start LabVIEW
2. Open and run the file *DIO64 Simple One Board Input.vi* that is located in the *dio64 examples.llb*.
3. Toggle the digital input lines either by using a function generator or by temporarily grounding one of the digital I/O lines by connecting it to pin 17. If the graph is set to show the bit being toggled, the graph should display the state change and old state. For instance, if a bit is triggered from high to low, the right side of the graph will show a state change from high to low. Previous to this state change, a high signal will be displayed.

Removing the hardware

If you want to remove the hardware from the system, use the system's *Hardware Wizard*. The DIO-64 will show up in the list of devices if you click on the "Show hidden devices" check box.

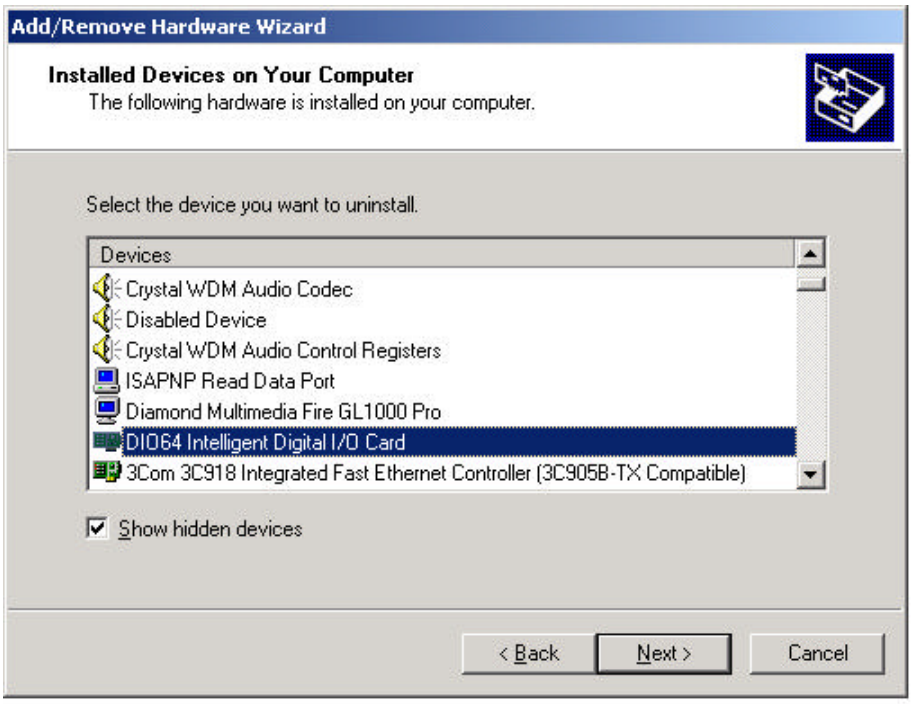


Figure 3 - Hardware Wizard Dialog

Removing the software

Please use the *Add/remove Software* applet found in the control panel to remove the DIO-64 software. This ensures the complete and proper removal of the software from your system.

Theory of Operation

Overview

The board used by the DIO-64 Intelligent Digital I/O System is a high channel count, intelligent board capable of solving many types of digital acquisition tasks. The onboard logic allows the board to handle tasks that would be difficult or impossible to accomplish by relying only on the main PC CPU. The DIO-64 hardware is designed to work in conjunction with other data acquisition hardware with extensive clocking/triggering options via RTSI/PXI triggers and clocks. The DIO-64 complements its state change monitoring functionality with output capabilities that make it well suited for stimulus-response style acquisition and control applications.

Concepts

Signals

- Data — The DIO-64 has 64 TTL compatible digital I/O lines organized as 4 16-bits ports. These ports can be specified as inputs or outputs according to the following table:

A (0-15)	B (16-31)	C (32-47)	D (48-63)
In 0	In 1	In 2	In 3
Out 0	In 0	In 1	In 2
Out 0	Out 1	In 0	In 1
Out 0	Out 1	Out 2	In 0
Out 0	Out 1	Out 2	Out 3

Figure 4 - Port direction possibilities

- Scan clock — The Scan clock determines the rate at which the data lines are sampled.
- Start trigger — The Start trigger determines when the digital I/O operation begins. The start trigger signal can be generated in a number of ways.
- Stop trigger — The Stop trigger stops a digital I/O operation that is in progress. The stop trigger signal can be generated in a number of ways

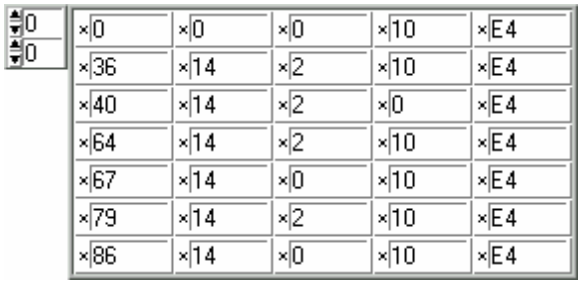
DIO-64 Data Format

The DIO-64 data is formatted as an array of scans of U16 integer words. The first two words of the array are the timestamp for that scan, least significant word first. The remaining words are the data associated with that scan.

Note that timestamps are the number of digital scan clock ticks that have occurred from the arming of the system. These counts can be converted to time (in seconds) by dividing by the *Scan Rate*. The DIO-64 will force a scan at timestamp wrap, which is at 0x7FFFFFFF. This guarantees that the host acquisition application will see at least one scan in every timestamp period.

For instance, the following data are from a 3-port DIO-64 state change acquisition run, with a scan rate of 200 kHz:

Here are 6 scans with 5 words of information. The first two columns in the array are the timestamps. The remaining columns correspond to the data on the ports used for this acquisition run.



0					
0					
	×0	×0	×0	×10	×E4
	×36	×14	×2	×10	×E4
	×40	×14	×2	×0	×E4
	×64	×14	×2	×10	×E4
	×67	×14	×0	×10	×E4
	×79	×14	×2	×10	×E4
	×86	×14	×0	×10	×E4

Figure 5 - example data

Deciphering this data would give the following results:

At timestamp	this happened	at time (secs)
0x00140036		0
0x00140036	bit 2 (A2) on	6.553870
0x00140040	bit 20 (B4) off	6.553920
0x00140064	bit 20 (B4) on	6.554100
0x00140067	bit 2 (A2) off	6.554115
0x00140079	bit 2 (A2) on	6.554205
0x00140086	bit 2 (A2) off	6.554270

Figure 6 – example data deciphered

NOTE: (timestamp) / (scan rate in Hz) = time in (secs)

For a VI that converts data from the DIO-64 to timestamps and boolean or word data, refer to the *Separate Data.vi* utility in the *dio64 tools.llb* file.

Sequence of Events

All DIO-64 operations go through the same sequence of events.

- The board is opened
- The FPGA program is loaded
- The board is **configured**
- The operation is **armed**
- The boards waits for a start trigger
- The operation has **started**
- The data acquisition operation runs
- The board waits for a stop trigger
- The operation has **stopped**

At the time the operation is **configured**, the board has been basically configured for the operation. I/O direction, clocking and triggering options have been established. Ports designated as outputs will begin driving. See *Digital Output Initial State Section (page 17)* for more details on establishing initial output conditions

At the time the operation is **armed**, the board is completely initialized for the operation and is basically waiting for a start trigger. It is important to realize that for an operation that has no start trigger defined (NONE), the driver will **arm** and **start** the operation simultaneously.

A stop trigger or software stop operation terminates the data acquisition in progress. The operation is **stopped**. No data will be collected. The application can go through the sequence again for the next operation.

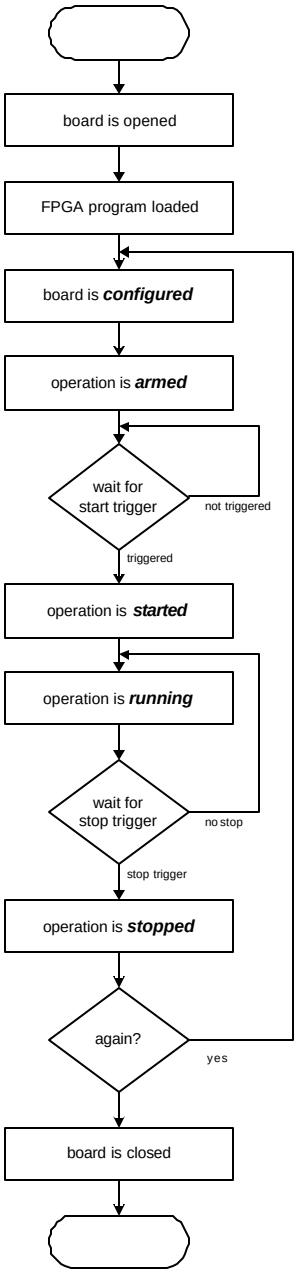


Figure 7 - DIO-64 flow

Modes of operation

Input Mode

The input state machine monitors 1-4 ports (16-64 bits) at a rate specified by the current *scan clock*. The *scan clock* can be derived from a variety of sources (see below). The digital input ports are sampled on each rising edge of the scan clock. If the state has changed on any of the indicated bits, the DIO-64 records the timestamp and current state of all inputs. This *scan* is placed in a FIFO for transfer to the PC.

Basic Operation

The DIO-64 board can be used as an input device. It can perform high-speed digital sampling and logging of data only when certain channels (per the channel mask) have changed. Whenever one or more digital channels have changed, data from all channels and a timestamp are stored in a FIFO buffer. The host can monitor the status of the acquisition and pull data out of the buffer without disturbing the digital acquisition.

Programming Steps

Programming applications for input-only operation involves running VIs in the following order:

1.	DIO64 Open.vi	Initialize the board.
2.	DIO64 Load.vi	Download the FPGA program on the board.
3.	DIO64 Set Attribute.vi	Optional – Not Required. Set any attributes needed for application here.
4.	DIO64 In Start.vi	The input operation is configured and armed. Once the start trigger is observed the board will start acquiring data into the buffer and the operation is started.
5.	DIO64 In Status.vi	Find out how much data is waiting to be read.
6.	DIO64 In Read.vi	Read in the data from the board's buffer.
7.	DIO64 In Stop.vi	Stop acquiring data. This VI forces a stop trigger, if an external stop trigger has not already stopped the operation. The input operation is stopped.

8. DIO64 Close.vi Close the selected board.

Steps 1, 2, 3, 6, and 7 are usually only run once per board in a program. Usually a program will loop over steps 4 and 5 repeatedly until all the desired data has been acquired.

Input Data Format

The basic DIO-64 data format was described in a previous section *(page7).*

In order to ensure that the initial state of all digital channels is known, the FPGA will always acquire a scan of the digital input channels when the start trigger is received, internal or external.

Output Mode

The output state machine counts falling edges on the *scan clock* and drives the output data when the scan clock count matches the timestamp on the next scan in the output FIFO. The DIO-64 can drive outputs on 1-4 ports (16-64 bits). Complex digital waveforms can be generated by combining scans and writing the timestamp and data into the DIO-64's output FIFO.

The DIO-64 can output digital data in 2 modes: streaming and looping.

The DIO-64 streaming mode is when the application continuously feeds new data to the output FIFO. This data can come from a previously recorded set of data using the DIO-64's input capabilities **OR** the data can be produced programmatically.

The DIO-64 looping mode allows the DIO-64 to repetitively generate an output waveform from a block of scans that fits within the FPGA's output FIFO. The DIO-64 will loop back to the beginning of the waveform after it has completed one pass through the FIFO data. This can be repeated a fixed number of times or continuously. This can also be done in a retriggerable fashion, where a start trigger initiates each waveform.

Cyclic Output

Basic Operation

The DIO-64 can generate a digital output waveform by issuing new scans at specified times. Each digital output pattern is given a timestamp, and a group of these patterns can be set to output for a finite or infinite number of repetitions.

Programming Steps

Programming applications for output operations involves running VIs in the following order:

1.	DIO64 Open.vi	Initialize the board.
2.	DIO64 Load.vi	Download the FPGA program on the board.
3.	DIO64 Set Attribute.vi	Optional – Not Required. Only set attributes for Input Buffer size or Output Buffer size here, other attributes are set below.
4.	DIO64 Out Config.vi	Configure the output parameters for the board.
5.	DIO64 Set Attribute.vi	Optional – Not Required. Only set attributes for any non Buffer size attributes here.
6.	DIO64 Out Status.vi	Get the current status of the board.
7.	DIO64 Out Write.vi	Set the outputs to an initial value, or load in the data to be used for cyclic output.
8.	DIO64 Out Start.vi	Start the output operation
9.	DIO64 Out Stop.vi	Stop outputting data.
10.	DIO64 Close.vi	Close the selected board.

It is important to know that the DIO-64 automatically sets the clock to the timestamp found in the first scan in the FIFO on a start trigger AND when the FIFO wraps back to the beginning. This example and most applications typically use time 0 for the timestamp in the first scan.

The cyclic operation can be used to generate a *retriggerable cyclic output* mode. Specify the start trigger source. This signal will initiate an output sequence. Specify the stop trigger as *Output FIFO.* This causes the DIO-64 to stop the output sequence when the last scan in the FIFO has been issued. It then re-arms itself and waits for the next start trigger.

Streaming Output

Basic Operation

The DIO-64 can generate a digital output waveform by issuing new scans at specified times. An application can generate a continuous stream of digital patterns by keeping the DIO-64 FIFO full with new scans.

Programming Steps

Programming applications for output-only operation involves running VIs in the following order:

1.	DIO64 Open.vi	Initialize the board.
2.	DIO64 Load.vi	Download the FPGA program on the board.
3.	DIO64 Set Attribute.vi	Optional – Not Required. Only set attributes for Input Buffer size or Output Buffer size here, other attributes are set below.
4.	DIO64 Out Config.vi	Configure the output parameters for the board.
5.	DIO64 Set Attribute.vi	Optional – Not Required. Only set attributes for any non Buffer size attributes here.
6.	DIO64 Out Status.vi	Get the current status of the board.
7.	DIO64 Out Write.vi	Set the outputs to an initial value, or load in the data to be used for cyclic output.
8.	DIO64 Out Start.vi	Start the output operation
9.	DIO64 Out Status.vi	Get the current status of the board.
10.	DIO64 Out Write.vi	Set the outputs to an initial value, or load in the data to be used for cyclic output.
11.	DIO64 Out Stop.vi	Stop outputting data.
12.	DIO64 Close.vi	Close the selected board.

All steps, except steps 9 and 10, are usually run only once per board in a program. When operating in cyclic mode, steps 9 and 10 are frequently omitted.
Otherwise, a program will usually loop over steps 9 and 10 repeatedly until all the desired data has been output.

Output Data Format

The basic DIO-64 data format was described in a previous section *(page7).*

It is important that the data scans that the application is providing to the DIO-64 have timestamps that will occur in the future. If the application puts data in the FIFO that happens in the "past", the DIO-64 will wait for the next scan clock timer wrap to issue that scan. This could take quite awhile for lower clock rates and the DIO-64 will appear to have hung.

Simultaneous Input and Output

Basic Operation

The DIO-64 can perform input and output operations simultaneously. An application can generate a continuous stream of digital patterns and sample the digital inputs at the same time.

The port designations are remapped when using combined input and output operation. The first output port is port 1 and maps to physical port A, the second is port 2 and maps onto physical port B, etc. The first input port is also named port 1 and is mapped to the first non-output port; the second is port 2 and is mapped to the second non-output port, etc.

Programming Steps

Programming applications for combined input and output operation involves running VIs in the following order:

1.	DIO64 Open.vi	Initialize the board.
2.	DIO64 Load.vi	Download the FPGA program on the board.
3.	DIO64 Set Attribute.vi	Optional – Not Required. Only set attributes for Input Buffer size or Output Buffer size here, other attributes are set below.
4.	DIO64 Out Config.vi	Configure the output parameters for the board.
5.	DIO64 Set Attribute.vi	Optional – Not Required. Only set attributes for any non Buffer size attributes here.
6.	DIO64 Out Status.vi	Get the current status of the board.

7.	DIO64 Out Write.vi	Set the outputs to an initial value, or load in the data to be used for cyclic output.
8.	DIO64 In Start.vi	Start the I/O operation
9.	DIO64 Out Status.vi	Get the current status of the board.
10.	DIO64 Out Write.vi	Set the outputs to an initial value, or load in the data to be used for cyclic output.
11.	DIO64 In Status.vi	Find out how much data is waiting to be read.
12.	DIO64 In Read.vi	Read in the data from the board's buffer.
13.	DIO64 Out Stop.vi	Stop outputting data.
14.	DIO64 Close.vi	Close the selected board.

Steps 1-8 and 13-14 should be only run once in a program. When outputting in cyclic mode, steps 9 and 10 are frequently omitted. Steps 9-10 and/or 11-12 may be repeated as many times as necessary to read and write the desired data. The order of steps 9-12 is not important, except that the appropriate status must be performed before a read or write operation.

Signal details

Digital Input/Outputs

The digital channels are organized into four 16-channel ports (A through D). Combined input and output operation does not allocate ports in the same way as input-only or output-only operation. Refer to the section on combined input and output operation for more information.

The DIO-64's 64 digital inputs are interfaced through transceivers into the FPGA which latches the current state of all inputs with the digital scan clock, ensuring that a scan will minimal timing skew. These transceivers expect TTL level inputs.

The DIO-64's digital outputs are also TTL compatible and are latched on write operations.

The DIO-64 digital I/O lines can be jumpered to pull-up or pull-down a digital I/O line through a resistor. Whether a line is pulled up or down is determined by resistor packs on the DIO-64 board. There are eight resistor packs, 2 for each of the four data ports A,B,C,D, which can affect 8 digital lines. If the pack is aligned toward the top of the socket the data bits are pulled high. Be sure that the small triangle on the resistor pack is at the top of the socket. If the pack is aligned toward the bottom of the socket the data bits are pulled low. Be sure that the small triangle on the resistor pack is at the bottom of the socket. If the pack is removed the data bits are not pulled high or low. See the diagrams to the right for an illustration of each resistor pack and socket.

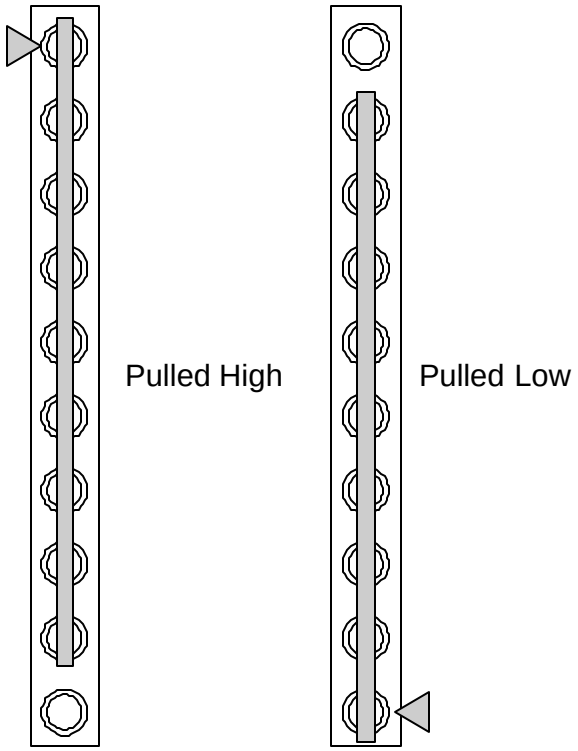


Figure 8 - Resistor pack settings

See the table below for the mapping between resistor packs to data bits.

	High	Low
A	XU6	XU8
B	XU2	XU4
C	XU7	XU9
D	XU3	XU5

Figure 9 - Resistor pack designations

Figure 10 - PCI PXI resistor pack locations

Digital I/O Port Direction

One of the most important parameters of a DIO-64 operation is the specification of which ports are inputs and which ports are outputs. The DIO-64 allows each port (a group of 16 digital bits or lines) to be an input or an output.

A (0-15)	B (16-31)	C (32-47)	D (48-63)
In 0	In 1	In 2	In 3
Out 0	In 0	In 1	In 2
Out 0	Out 1	In 0	In 1
Out 0	Out 1	Out 2	In 0
Out 0	Out 1	Out 2	Out 3

Figure 11 - possible port I/O designations

The method of specifying the port direction configuration is to explicitly specify the number of inputs and outputs with the *DIO64 Load* VI.

NOTE: This is a change from the way that the DIO-128 system determined the port configuration. The DIO-128 established the configuration through a combination of which DSP control program was loaded and through the number of ports specified in the *DIO128 Out Config* and *DIO128 In Start* VIs.

Digital Output Initial State

Applications that use digital outputs often have very specific requirements for the initial state of the digital lines. A digital line that controls a furnace needs to be

setup such that the furnace does not turn on as soon as the PC is powered up. The pullup/pulldown resistors, mentioned earlier, will establish the power up state for a digital line.

When a port is designated as an output port, the board will initially drive what it thinks the current value for that port should be. An application can force a value that will be used as the initial output value by using the *DIO64 Output Force Output* VI before the *DIO64 OutConfig* VI. The FPGA will set all the digital lines as inputs until the *DIO64 OutConfig* VI executes. If the application has not established the initial output value, the driver will default to 0.

NOTE: The application must fully specify the number output ports when calling *DIO64 Load* in order for the *DIO64 Output Force Output* VI to correctly operate before calling the *DIO64 Output Config* VI.

An application can use the *DIO64 Out Get Input* VI before the *DIO64 OutConfig* has been executed in order to determine the initial state of the digital ports that will soon become outputs.

Digital Output Final State

By default the DIO-64 digital output ports revert back to inputs when the DIO-64 Close VI is executed, or the application terminates unexpectedly. This is to allow the pullup/pulldown resistors to return the outputs to the intended “inactive” state.

NOTE: This is a change from the way that the DIO-128 digital outputs behaved. See the Differences between DIO-64 and DIO-128 section on page 48 for information on how to return to the DIO-128 behavior.

Clocks

Scan Clock

The pacing of DIO-64 operations is controlled by the system's *scan clock*. The DIO-64 latches inputs or drives outputs on the rising edge of the scan clock. The DIO-64 scan clock can be derived from one of four sources: a division of the major clock source, an external clock, RTSI/PXI Trigger 0, 10 MHz OCXO. The major clock sources available are described below.

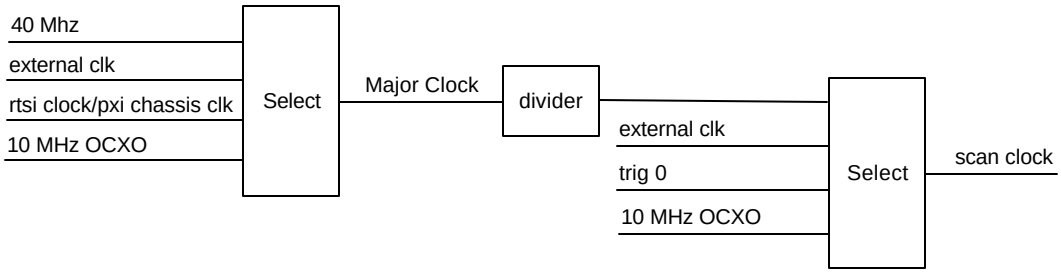


Figure 12 - Scan clock sources

Major Clock

The major clock signal on the DIO-64 is available to use as the system's scan clock. The DIO-64 major scan clock can be derived from one of four sources: an onboard 40 MHz clock, an external clock, the RTSI clock (PCI-DIO-64) or PXI Chassis Clock 10MHz (PXI-DIO-64), or optionally, the 10 MHz Oven-controlled oscillator. It is important to realize that the minimum divisor available to use against the major clock source is 2.

40 MHz clock

The primary clock available on the DIO-64 is a 40 MHz clock. This clock can be used as a major clock and can be divided and used as the scan clock.

External clock

An external clock source can be brought in through the edge connector on pin 20. It can be used directly as the scan clock, or used as the major clock and divided before being used as the scan clock.

Trigger 0

A clock present on the RTSI/PXI Trigger 0 line can be used directly as the scan clock.

RTSI clock

The RTSI clock allows a common time base to be shared by all boards on a RTSI chassis. It can be used as the major clock and divided before being used as the scan clock. This is available on the PCI-DIO-64.

PXI chassis clock

The PXI chassis clock allows a common time base to be shared by all boards in a PXI chassis. It can be used as the major clock and divided before being used as the scan clock. This is available on the PXI-DIO-64 when used in a PXI chassis. This is typically a 10 MHz clock.

10 MHz OCXO (Oven controlled oscillator)

The 10 MHz OCXO option provides the DIO-64 with a high precision clock source. This clock can be used directly as a scan clock or divided as a major clock. When the OCXO option is installed, the DIO-64 can distribute this clock to other boards in the system by generating the RTSI clock (PCI-DIO-64) or PXI chassis clock (PXI-DIO-64).

Triggers

DIO-64 triggering options allow for the precise control over when a digital data acquisition operation starts and stops.

Start trigger

The Start trigger initiates the data acquisition operation. The Start trigger can be selected from one of the following signals:

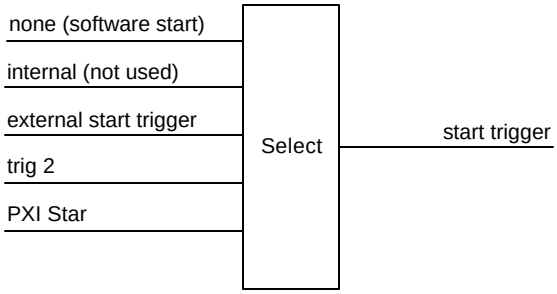


Figure 13 - Start trigger sources

Source	Description
None	Triggered immediately (software trigger)
Internal	Future expansion: do not use this option
External Start trigger	Triggered with signal applied to pin # 24 on external connector
Trig 2	Triggered via the RTSI/PXI Trigger 2 line
PXI Star	Triggered via the PXI Star Trigger line (PXI-DIO-64 only)

Figure 14 - Start trigger descriptions

The DIO-64 supports 3 types of start triggers:

Type	Description
level	Triggers if signal is asserted (positive or negative)
edge	Waits for signal to be de-asserted, and then triggers when the signal transitions to asserted (rising or falling)
edge to edge	Same as edge, but the start trigger signal is used for both start and stop triggers

Figure 15 - Start trigger types

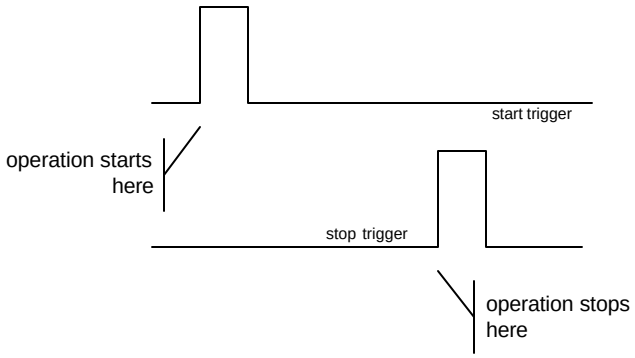


Figure 16 - Edge triggered timing

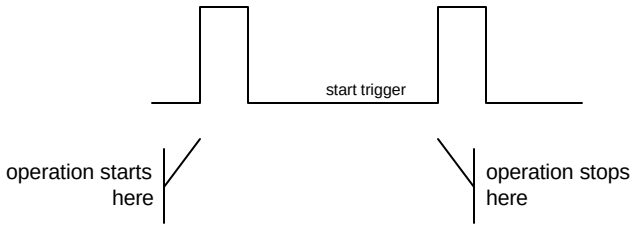


Figure 17 – Edge to Edge trigger timing

The DIO-64 supports high and low level type triggers and rising and falling edge type triggers.

Stop trigger

The Stop trigger terminates the data acquisition operation. The Stop trigger can be selected from one of the following signals:

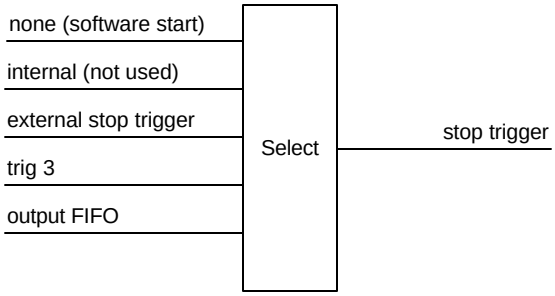


Figure 18 - Stop trigger sources

Source	Description
None	Triggered immediately (software trigger)
Internal	Future expansion: do not use this option
External Stop trigger	Triggered with signal applied to pin # 25 on external connector
Trig 3	Triggered via the RTSI/PXI Trigger 3 line
Output FIFO	Triggered when the Output FIFO transfers the last scan in a retriggerable cyclic output transfer. A new Start trigger will cause the entire waveform to be generated again.

Figure 19 - Stop trigger description

The DIO-64 only supports edge stop triggers.

Special PXI features

The PXI-DIO-64 can allow your application to take advantage of many specialized PXI features. These features are only available when the PXI-DIO-64 is used in a PXI compatible chassis. These features are not available when the PXI-DIO-64 is used in a CompactPCI chassis.

A PXI chassis contains three types of slots. Slot 1 is the controller slot for the chassis. Slot 2 is designated at the Star Trigger Controller slot. A PXI peripheral with Star Trigger Controller capabilities can be used in this slot. The remaining slots are for PXI peripheral cards. The PXI-DIO-64 can be used in the Star Trigger Controller and in any of the peripheral slots.

Star Trigger

The PXI Star Trigger line is a special trigger that is specifically designed to minimize the propagation delay from a PXI Star Trigger Controller in Slot 2 and PXI peripherals in the remaining PXI slots. Each peripheral receives a signal that has been compensated by the chassis in order to minimize the skew seen between peripherals in the chassis.

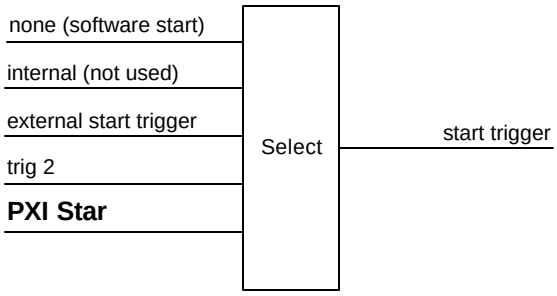


Figure 20 - PXI Star as a start trigger

The PXI-DIO-64 can use the PXI Star Trigger as a start trigger input for a DIO-64 operation when placed in a peripheral slot (not Slot 2).

If the PXI-DIO-64 is placed in the Star Trigger Slot, it cannot, currently, generate a signal on the Star Trigger line for use by other peripherals.

PXI chassis Clock

The PXI chassis provides a PXI Chassis clock that can be used by PXI peripherals to synchronize operations to a single clock. The PXI specification specifies that the chassis will provide a 10 MHz clock. A Star Trigger Controller (in Slot 2) can override this clock and provide its own 10 MHz clock for use throughout the chassis.

The PXI-DIO-64, when equipped with the optional 10 MHz OCXO, can supply this precision clock to the entire chassis. Enabling the PXI chassis clock attribute with the *DIO-64 Set Attribute* VI enables this.

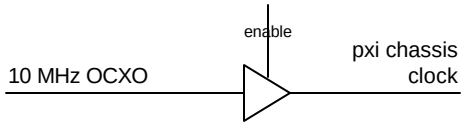


Figure 21 - PXI chassis clock enable

PXI Trigger lines

The PXI chassis provides 8 general-purpose lines that are routed to all peripherals in the chassis. These lines (PXI Trigger 0-7) can be used to share triggers and clocks between peripherals in the chassis.

The PXI-DIO-64 can make use of each of the PXI Trigger lines. Each line can be used for a particular purpose. Please refer to the section that pertains to the signal listed for more details.

PXI Trigger	Description
0	Scan clock
1	(not used)
2	Start Trigger
3	Stop Trigger
4	DAQ Modulo
5	(not used)
6	(not used)
7	Scan clock

Figure 22 - PXI Trigger line assignments

The DIO-64 will automatically mirror the scan clock, start and stop trigger signals on the designated PXI triggers by default, if the particular signal has not been configured to use a PXI trigger as its source. For instance, if an operation has been configured to use an internally generated clock, the driver will automatically drive that clock signal out PXI Trigger 0. There is an attribute that will disable this functionality, disconnecting the DIO-64 from driving the trigger lines. Use the *DIO64 Set Attribute* VI to enable or disable this feature.

PXI Trigger 4 can optionally drive a divided scan clock to other boards in the system. This function is controlled by the DAQ Clock Modulo field in the Start Control found in the *DIO64 In Start* and *DIO64 OutConfig* VIs.

PXI Trigger 7 can also optionally drive one of three of the DIO-64 internal clocks. The choices are:

- A 20 MHz clock (based on the 40 MHz board clock)
- A 10 MHz clock (based on the 40 MHz board clock)
- Optionally, the 10 MHz OCXO clock (if installed)

There is an attribute that will control this functionality. Use the *DIO64 Set Attribute* VI to enable and choose the source or disable this feature.

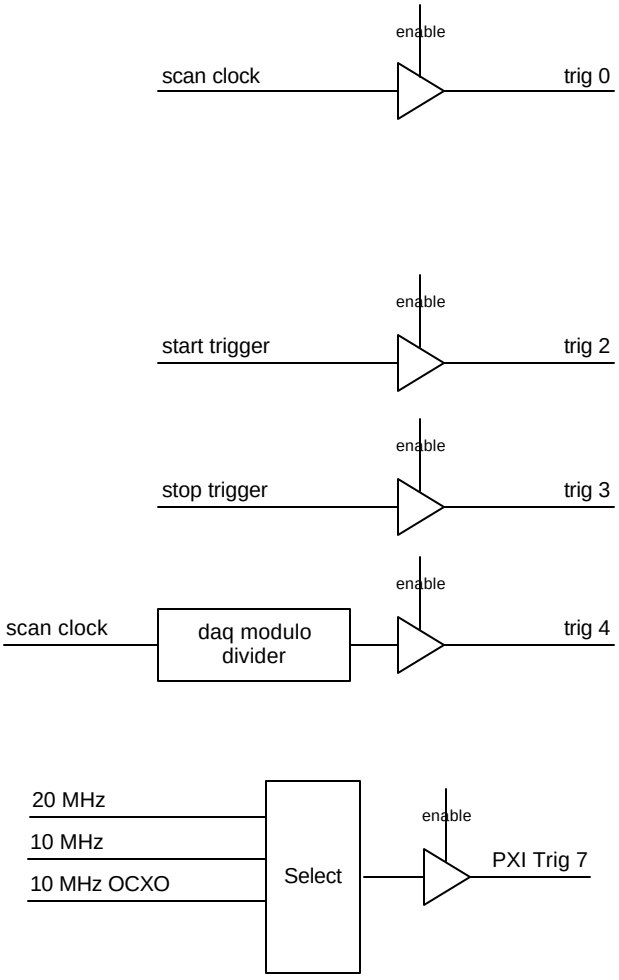


Figure 23 - PXI trigger enables

Special RTSI features

The PCI-DIO-64 with its integrated RTSI support can allow your application to take advantage of the multi-board synchronization features of data acquisition boards that support the RTSI bus.

RTSI Clock

The RTSI bus provides a RTSI clock that can be used by RTSI peripherals to synchronize operations to a single clock. The DIO-64 can use the RTSI clock as a source for its Major Clock source and through the divider as a scan clock. The DIO-64 can also optionally drive one of three of the DIO-64 internal clocks out as the RTSI clock. The choices are:

- A 20 MHz clock (based on the 40 MHz board clock)
- A 10 MHz clock (based on the 40 MHz board clock)
- Optionally, the 10 MHz OCXO clock (if installed)

There is an attribute that will control this functionality. Use the *DIO64 Set Attribute* VI to enable and choose the source or disable this feature.

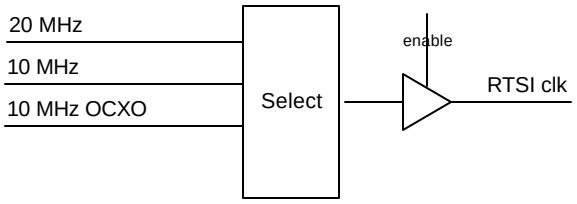


Figure 24 - RTSI clock sources

RTSI Trigger lines

The RTSI bus provides 7 general-purpose lines and one clock signal that are routed to all connected boards. These lines (RTSI Trigger 0-6 and RTSI clock) can be used to share triggers and clocks between peripherals in the chassis.

The PCI-DIO-64 can make use of each of the RTSI Trigger lines. Each line can be used for a particular purpose. Please refer to the section that pertains to the signal listed for more details.

RTSI Trigger	Description
0	Scan clock
1	(not used)
2	Start Trigger
3	Stop Trigger
4	DAQ Modulo
5	(not used)
6	(not used)

Figure 25 - RTSI trigger assignments

The DIO-64 will automatically mirror the scan clock, start and stop trigger signals on the designated RTSI triggers by default, if the particular signal has not been configured to use a RSTI trigger as its source. For instance, if an operation has been configured to use an internally generated clock, the driver will automatically drive that clock signal out RTSI Trigger 0. There is an attribute that will disable this functionality, disconnecting the DIO-64 from driving the trigger lines. Use the *DIO64 Set Attribute* VI to enable or disable this feature.

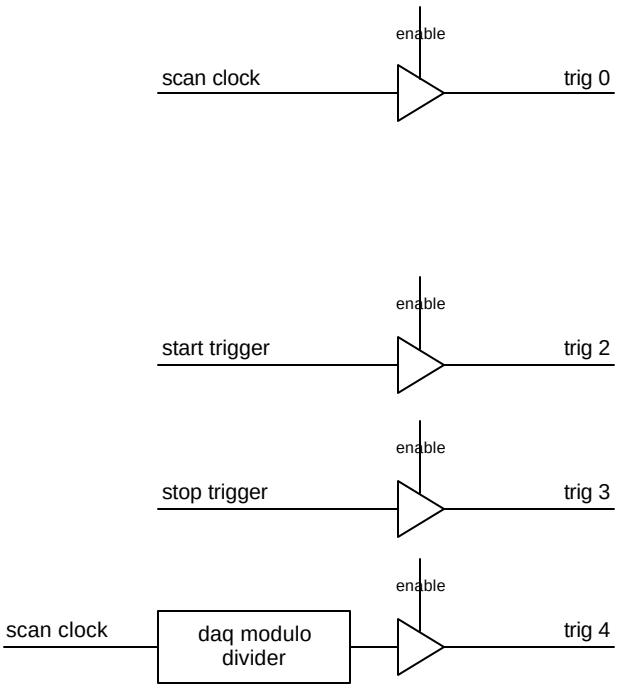


Figure 26 - RTSI trigger enables

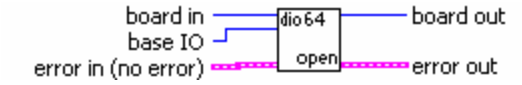
VI Reference

All of the DIO-64 VIs share common parameters. These parameters are defined once below, and are omitted from the individual VI descriptions.

	error in (no error) error in is a cluster that describes the error status before this VI executes. If error in indicates that an error occurred before this VI was called, this VI may choose not to execute its function, but just pass the error through to its error out cluster. If no error has occurred, then this VI executes normally and sets its own error status in error out. Use the error handler VIs to look up the error code and to display the corresponding error message. Using error in and error out clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.
U16	**board in** board in is a user defined value between 0 and 7 that is used as a handle to the particular digital board. The board number is assigned by the Windows Device Manager, and is roughly determined by the order in which the boards are discovered by the OS. All subsequent VIs reference the physical board via this board number.
	error out error out is a cluster that describes the error status after this VI executes. If an error occurred before this VI was called, error out is the same as error in. Otherwise, error out shows the error, if any, that occurred in this VI. Use the error handler VIs to look up the error code and to display the corresponding error message. Using error in and error out clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.
U16	**board out** board out is a flow-through parameter which is the same as board in.

DIO64 Open.vi

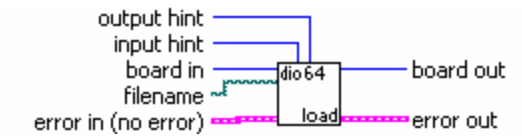
DIO64 Open initializes the selected DIO-64 board.



U16	**base IO** base IO is no longer used with the DIO-64. It is left for backward compatibility with DIO-128 applications.

DIO64 Load.vi

DIO64 Load loads the designated FPGA code onto the DIO-64 board.



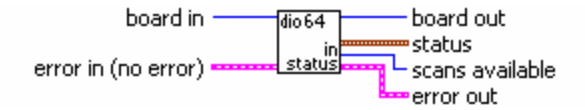
	filename filename is the filename of the FPGA code image. In order to accomodate upgraded DIO-128 applications, if the filename specified end with .bnm the VI will automatically substitute the default DIO-64 code image filename. If the file is not found where specified in this control, the system searches for this file in the \WINDOWS and \WINDOWS\SYSTEM directories.
I32	**input hint** (Required) The input hint allows the user specify, at load time, the number of input ports the application will be using. The default (-1) defers that decision until the InputStart.
I32	**output hint** (Required) The output hint allows the user specify, at load time, the number of output ports the application will be using. The default (-1) defers that decision until the OutputCOnfig.

DIO64 Close.vi

This VI closes the connection to the selected digital I/O board.

DIO64 In Status.vi

DIO64 In Status checks the status of the selected digital I/O board. This operation is not allowed before DIO64 Load has been executed.

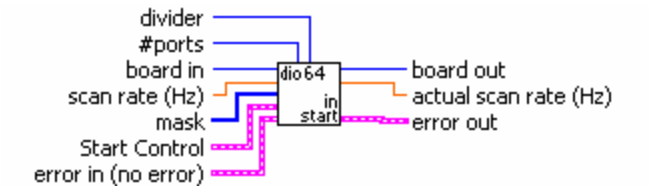


	status status is a cluster containing information about the current status of the digital card. This cluster needs to be passed to the DIO128 In Read.vi. The parameters of interest are as follows: Time0 is the low byte of the time elapsed. This is the fifth element of the cluster. Time1 is the high byte of the time elapsed. This is the sixth element of the cluster. portCount is the number of ports currently being scanned. This is the second element of the cluster. Note: When running in simple mode and portCount is 1, Time0 and Time1 are not updated to increase system performance. All other variables in this cluster are for internal use only.
U32	**scans available** Scans available is the number of scans stored in the buffer since the last read. A scan occurs whenever a monitored bit transitions from low-to-high or high-to-low.

DIO64 In Start.vi

DIO64 In Start starts the acquisition on the digital I/O board. This operation is not allowed before DIO64 Load has been executed or while an operation is in progress.

Note: This VI is only used for input-only and combined input and output board modes. This VI will generate an error in the output-only board mode.



DBL	**scan rate (Hz)** Scan rate (Hz) is the scan rate of the digital board in Hz. This input can only be used when the Major Clock source's rate is known. If the Major Clock source is set to "external" or "RTSI clock/PXI chassis clock" you should instead specify the divider to be used.
U16	**mask** Mask is an array of U16 numbers representing the mask of bits to be sampled. A '1' in a bit position indicates that the bit should be sampled. Each element of the array represents a 16 bit port on the digital card. Element 0 corresponds to port A, element 1 corresponds to port B, etc.
	Start Control In Start Control is a cluster specifying advanced clocking and triggering options.

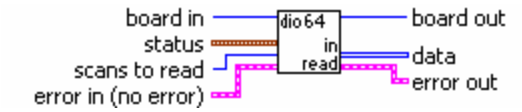
	flags unused. The flags parameter is unused for the DIO-64. It is left for backward compatibility with DIO-128 applications.
	clock control internal (0) - internal clock base external (1) - clocked via pin 20 of P2 (rear) connector RTSI 0 (2) - clocked via RTSI/PXI Trigger 0 line RTSI Clk PXI Trig 7 (3) - clocked via either the RTSI Clk or PXI Trig 7 line, depending on PCI or PXI DIO-64 board
U16	**DAQ clock modulo** DAQ clock modulo is the divisor of the DIO-128 scan clock applied to RTSI Trigger4. A modulo of 0 indicates that no clock is applied to the RTSI Trigger4 connector.
	start trigger

	source none (0) - starts immediately internal (1) - Future expansion: do not use this option external (2) - triggered with signal applied to pin 24 on external connector trig 2 (3) - triggered via RTSI/PXI Trigger 2 PXI Star (4) - triggered via the PXI Star Trigger - PXI-DIO-64 only
	type level sensitive (0) - triggering responds to a high or low level (selectable) edge sensitive (1) - triggering responds to a transition: high to low or low to high (selectable) edge to edge sensitive (2) - triggering responds to a transition: high to low or low to high (selectable) with both the start and stop trigger being driven by the "start" trigger source
	sense rising edge (0) - triggering responds to a low to high transition in edge sensitive mode, or a high in level sensitive mode

		falling edge (1) - triggering responds to a high to low transition in edge sensitive mode, or a low in level sensitive mode
	stop trigger	
		source none (0) - starts immediately internal (1) - Future expansion: do not use this option external (2) - triggered with signal applied to pin 25 on external connector trig 3 (3) - triggered via RTSI/PXI Trigger 3 output FIFO (4) - stop trigger caused by driving the last scan in the output FIFO
		type type is not used on the DIO-64. The stop trigger is alway an edge trigger. This is left for DIO-128 compatibility
		sense falling edge (0) - triggering responds to a high to low transition in edge sensitive mode, or a low in level sensitive mode rising edge (1) - triggering responds to a low to high transition in edge sensitive mode, or a high in level sensitive mode
U8	**#ports** #ports is the number of 16-bit ports that will be processed during the monitoring of the scans. A value of 1 indicates port A is processed, a value of 2 indicates ports A and B are processed, etc. Note that performance is increased when fewer ports are used. Valid values are between 1 and 8.	
U32	**divider** The divider is used to directly specify the divider to be used against the Major Clock source before it is used as the scan clock.	
DBL	**actual scan rate (Hz)** actual scan rate (Hz) is the scan rate that is actually used for the input operation. This value is the adjustment made to the input scan rate to make it an integer divisor of the Major Clock source.	

DIO64 In Read.vi

DIO64 In Read reads transition data from the selected digital I/O board as 16-bit words. This operation is only allowed during an input or input/output operation.

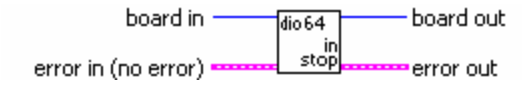


U32	**scans to read** scans to read is the number of scans to read from the digital I/O board. This field should be passed in from a previous DIO64 Status.vi.
	status status is a cluster containing information about the current status of the digital

	card. This cluster needs to be passed to the DIO128 In Read.vi. The parameters of interest are as follows: Time0 is the low byte of the time elapsed. This is the fifth element of the cluster. Time1 is the high byte of the time elapsed. This is the sixth element of the cluster. portCount is the number of ports currently being scanned. This is the second element of the cluster. Note: When running in simple mode and portCount is 1, Time0 and Time1 are not updated to increase system performance. All other variables in this cluster are for internal use only.
[U16]	**data** data is the actual data read from the digital I/O board. The data is passed back as a two-dimensional array of U16 numbers. The first column is the low order word of the 32bit timestamp. The second column is the high order word.

DIO64 In Stop.vi

DIO64 In Stop stops acquisition on the selected digital I/O board. When this VI is executed, all RTSI/PXI lines in the DIO-64 System are reset to an input state. This operation is only allowed during an input or input/output operation.

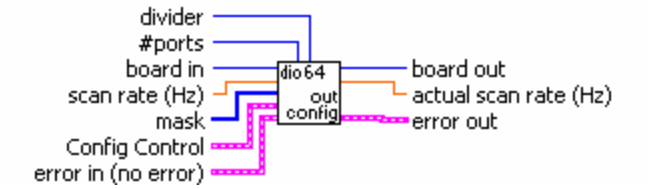


DIO64 Out Config.vi

DIO64 Out Config establishes the parameters for an output operation on the digital I/O board. This must be performed before data is written or the output operation is started. This operation is only allowed before DIO64 Load and not while an operation is in progress.

Note: That a cyclic operation must involve at least 3 transitions.

Note: This VI is only used for output-only and combined input and output board modes. This VI will generate an error in input-only board mode.



DBL	**scan rate (Hz)** Scan rate (Hz) is the scan rate of the digital board in Hz. This input can only be used when the Major Clock source's rate is known. If the Major Clock source is set to "external" or "RTSI clock/PXI chassis clock" you should instead specify the divider to be used.
[U16]	**mask** mask is an array of U16 numbers representing the mask of bits to be output. A '1' in a bit position indicates that the bit should be output. Each element of the array represents a 16 bit port on the digital card. Element 0 corresponds to port A, element 1 corresponds to port B, etc.
	Config Control Start Control is a cluster specifying advanced clocking and triggering options.

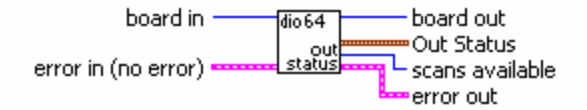
	flags unused. The flags parameter is unused for the DIO-64. It is left for backward compatibility with DIO-128 applications.
	clock control internal (0) - internal clock base external (1) - clocked via pin 20 of P2 (rear) connector RTSI 0 (2) - clocked via RTSI/PXI Trigger 0 line RTSI Clk PXI Trig 7 (3) - clocked via either the RTSI Clk or PXI Trig 7 line, depending on PCI or PXI DIO-64 board
	start trigger

	source none (0) - starts immediately internal (1) - Future expansion: do not use this option external (2) - triggered with signal applied to pin 24 on external connector trig 2 (3) - triggered via RTSI/PXI Trigger 2 PXI Star (4) - triggered via the PXI Star Trigger - PXI-DIO-64 only
	type level sensitive (0) - triggering responds to a high or low level (selectable) edge sensitive (1) - triggering responds to a transition: high to low or low to high (selectable) edge to edge sensitive (2) - triggering responds to a transition: high to

			low or low to high (selectable) with both the start and stop trigger being driven by the "start" trigger source.
			sense rising edge (0) - triggering responds to a low to high transition in edge sensitive mode, or a high in level sensitive mode falling edge (1) - triggering responds to a high to low transition in edge sensitive mode, or a low in level sensitive mode
		stop trigger Stop Trigger is only supported under combined input and output board mode. Stop triggering will generate and error in the output-only board mode.	
			source none (0) - starts immediately internal (1) - Future expansion: do not use this option external (2) - triggered with signal applied to pin 25 on external connector trig 3 (3) - triggered via RTSI/PXI Trigger 3 output FIFO (4) - stop trigger caused by driving the last scan in the output FIFO
			type type is not used on the DIO-64. The stop trigger is always an edge trigger. This is left for DIO-128 compatibility
			sense falling edge (0) - triggering responds to a high to low transition in edge sensitive mode, or a low in level sensitive mode rising edge (1) - triggering responds to a low to high transition in edge sensitive mode, or a high in level sensitive mode
		DAQ clock modulo DAQ clock modulo is the divisor of the DIO-64 scan clock applied to RTSI/PXI Trigger4. A modulo of 0 indicates that no clock is applied to the RTSI Trigger4 connector.	
		repetitions repetitions specifies the number of times a cycle should repeat. If repetitions is set to 0, the cycles will repeat until explicitly stopped. The #transitions field must be set in order for this value to have any effect.	
		#transitions #transitions specifies the number of transitions in the FIFO that constitute a cycle. If this value is 0, the output operation will only track data as it is fed to the FIFO.	
	#ports #ports is the number of 16-bit ports that will be processed during the output operation. A value of 1 indicates port A is processed, a value of 2 indicates ports A and B are processed, etc. Note that performance is increased when fewer ports are used. Valid values are between 1 and 4.		
	divider The divider is used to directly specify the divider to be used against the Major Clock source before it is used as the scan clock.		
	actual scan rate (Hz) actual scan rate (Hz) is the scan rate that is actually used for the output operation. This value is the adjustment made to the output scan rate to make it an integer divisor of the 16 MHz internal clock.		

DIO64 Out Status.vi

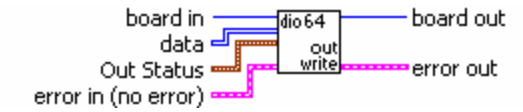
DIO64 Out Status checks the output status of the selected digital I/O board. This operation is not allowed before DIO Load has been executed.



	Out Status status is a cluster containing information about the current status of the digital card. This cluster needs to be passed to the DIO64 Out Write.vi. The parameters of interest are as follows: Time0 is the low byte of the time elapsed. This is the fifth element of the cluster. Time1 is the high byte of the time elapsed. This is the sixth element of the cluster. portCount is the number of ports currently being output to. This is the second element of the cluster. CurReps increments every time a repetitive cyclic operation repeats. CurTrans increments every time a scan is output during a repetitive cyclic operation All other variables in this cluster are for internal use only.
U32	**scans available** scans available is the number of scans that can be written. It is based on the amount of space available in the output buffer.

DIO64 Out Write.vi

DIO64 Out Write fills the output buffer with the data specified. This operation is only allowed after a DIO64 Out Config or during an output operation.



[U16]	**data** data is the data to be written to the digital I/O board. The data must be passed as a two-dimensional array of U16 numbers. The first column is the low order word of the 32-bit timestamp. The second column is the high order word.
906	**Out Status** status is a cluster containing information about the current status of the digital card. This cluster needs to be passed to the DIO64 Out Write.vi. The parameters of interest are as follows: Time0 is the low byte of the time elapsed. This is the fifth element of the cluster. Time1 is the high byte of the time elapsed. This is the sixth element of the cluster. portCount is the number of ports currently being output to. This is the second element of the cluster. All other variables in this cluster are for internal use only.

DIO64 Out Start.vi

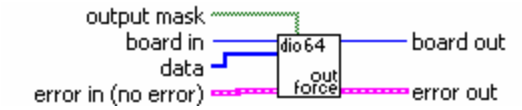
DIO64 Out Start writes the data in the output buffer to the DIO-64 ports. This VI should only be used for output-only operation. For combined input and output operation, DIO64 In Start.vi will start both input and output. This operation is allowed only after a DIO64 Out Config and during an output or input/output operation.

DIO64 Out Stop.vi

DIO64 Out Stop stops an output operation. This operation is allowed only during an output or input/output operation.

DIO64 Out Force Output.vi

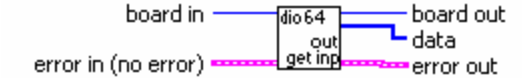
DIO64 Out Force Output immediately writes the specified data to the output ports. This operation is not allowed before DIO Load has been executed.



[U16]	**data** data is a one-dimensional array of length n. Valid values of n are 1 through 4. The data contains no timestamps - it is raw data that will be output to ports A, B, C, and D. Array element one will output to port A, element two to port B, etc. n may be larger than the number of output ports specified in DIO64 Out Config.vi for the output-only operation.
[TF]	**output mask** The output mask array, allows the application to specify which ports to update. If an element of the output mask array is true, the corresponding data element will be applied to the output port.

DIO64 Out Get Input.vi

DIO64 Out Get Input immediately returns the value on all the input ports that were valid during the last scan clock. This operation is not allowed before DIO Load has been executed.



[U16]	**data** data is a one-dimensional array with eight elements. Element one corresponds to port A, element two to port B, etc. Data returned for ports A, B, C, and D will be invalid for all ports configured to be outputs. Ports that are neither inputs nor outputs will be read as inputs.

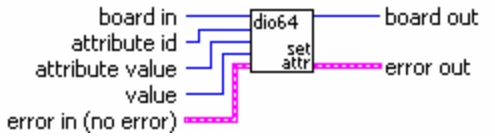
DIO64 Get Attribute.vi

This VI allows the application to retrieve one of many miscellaneous DIO-64 parameters. This operation is not allowed before DIO Load has been executed.

	attribute id attribute id: The particular DIO-64 attribute to retrieve. See the manual for details on each attribute
U32	**value** value: The current value of the attribute chosen.

DIO64 Set Attribute.vi

This VI allows the application to set one of many miscellaneous DIO-64 parameters. This operation is not allowed before DIO Load has been executed or during an operation. When OutConfig.VI is called in a program, care must be used when calling Set Attribute. The Attributes, Input Buffer Size and Output Buffer Size may only be set with Set Attribute.VI before a call to OutConfig,VI. Likewise when OutConfig.Vi is called and any Attribute not mentioned above is to be set the Call to Set Attribute.VI must be made after the Outconfig.VI call and before the OutStart/InStart.VI calls.



	attribute value attribute value: Specifies the value of the attribute to be set. If set to "long value" then the attribute will be set to the contents of the VALUE control. If set to default, the attribute will be reset to the driver's default value for that particular attribute. Other attribute values can be used for certain attributes.
U32	**value**: The new value of the attribute chosen, if the attribute value control is set to "long value".

DIO64 attributes

<table>
<tr><th>DIO64 attribute</th><th>Description</th></tr>
<tr><td>Input mode</td><td>One of four settings that defines how the driver accomplishes the input operation.
<table>
<tr><td>Polled</td><td>Simplest and slowest, the driver polls the DIO-64 board in order to obtain current results.</td></tr>
<tr><td>Interrupt</td><td>The DIO-64 generates an interrupt when the number of scans in the onboard input FIFO exceeds the Input threshold percent. The driver then directly reads the results. If no scans are detected within the Input DMA timeout period, an interrupt is automatically generated.</td></tr>
<tr><td>Packet</td><td>The DIO-64 generates an interrupt when the number of scans in the onboard FIFO exceeds the Input threshold percent. The driver initiates a DMA operation to read the results. If no scans are detected within the Input DMA timeout period, an interrupt is automatically generated.</td></tr>
<tr><td>Demand</td><td>The driver establishes a circular DMA operation in order to allow the board to transfer data from its FIFO to PC memory as needed. If no scans are detected within the Input DMA timeout period, an interrupt is automatically generated. This offers the highest performance.</td></tr>
</table>
The default is demand.</td></tr>
</table>

<table>
<tr><td>Output mode</td><td>One of four settings that defines how the driver accomplishes the output operation.<table><tr><td>Polled</td><td>Simplest and slowest, the driver polls the DIO-64 board in order to obtain current results.</td></tr><tr><td>Interrupt</td><td>The DIO-64 generates an interrupt when the number of scans in the onboard output FIFO falls below the Output threshold percent. The driver then directly sends any new scans</td></tr><tr><td>Packet</td><td>The DIO-64 generates an interrupt when the number of scans in the onboard output FIFO falls below the Output threshold percent. The driver initiates a DMA operation to send new scans.</td></tr><tr><td>Demand</td><td>The driver establishes a circular DMA operation in order to allow the board to transfer data from the PC memory to its FIFO as needed. This offers the highest performance. When a non-cyclic output demand operation is started the application must have previously filled the application buffer with at least enough scans to fill the FPGA FIFO.</td></tr></table>Note that cyclic operations (operations that repeat transitions held within the FPGA FIFO) do not require any data movement after the initial FIFO load. The output mode for this type of operation is irrelevant.
The default is packet.</td></tr>
<tr><td>Input buffer size</td><td>This attribute controls the size of the driver’s input FIFO in bytes. The default is 2 Mbytes (2097152). This default can be overridden by the DefaultInputBufferSize registry entry.</td></tr>
<tr><td>Output buffer size</td><td>This attribute controls the size of the driver output FIFO in bytes. The default is 512 kbytes (524288). This default can be overridden by the DefaultOutputBufferSize registry entry.</td></tr>
</table>

<table>
<tr><td>Major clock source</td><td>Specifies one of four potential sources for the Major clock. This signal will be divided before being used as a scan clock.<table><tr><td>Local Clock (40 MHz)</td></tr><tr><td>External clock</td></tr><tr><td>RTSI clock/PXI chassis clock</td></tr><tr><td>10 MHz clock</td></tr></table>See discussion on page 19.
The default is 0 (Local Clock (40 MHz)).</td></tr>
<tr><td>Input threshold percent</td><td>This attribute controls when the DIO-64 board initiates an interrupt or DMA transfer during an input operation.
The default is 50%.</td></tr>
<tr><td>Output threshold percent</td><td>This attribute controls when the DIO-64 board initiates an interrupt or DMA transfer during an output operation.
The default is 50%.</td></tr>
<tr><td>Input DMA timeout</td><td>This attribute controls when the DIO-64 forces any data that may be held in the onboard FIFO if no other conditions are met. The timeout is specified in milliseconds.
The default is 100 mSecs.</td></tr>
<tr><td>RTSI/PXI global enable</td><td>This attribute controls whether the DIO-64 is able to drive signals out the RTSI/PXI trigger lines. If enabled the board may drive these trigger lines. If disabled the board will not be able to drive these lines.
NOTE: this is equivalent to using the individual trigger enables below. If one of the triggers enables listed below is enabled, it will allow the board to drive that particular trigger line.
The default is enabled.</td></tr>
<tr><td>RTSI/PXI clock source</td><td>Specifies one of three possible source to be used as the RTSI clock/PXI Trigger 7.<table><tr><td>20 MHz</td></tr><tr><td>10 MHz</td></tr><tr><td>10 MHz OCXO</td></tr></table>The default is 0 (20 MHz).</td></tr>
<tr><td>RTSI clock/PXI Trig 7 enable</td><td>This attribute controls whether the DIO-64 will drive the clock specified above out the RTSI clock or PXI Trigger 7 lines. Valid settings are enabled or disabled.
The default is disabled.</td></tr>
</table>

<table>
<tr><td>External clock enable</td><td>This attribute controls whether the DIO-64 will drive the scan clock out the External Clock pin # 20. Valid settings are enabled or disabled.
The default is disabled.</td></tr>
<tr><td>PXI clock enable</td><td>This attribute controls whether the DIO-64 will drive the 10 MHz OCXO clock out to be used as the PXI chassis clock. This is only valid is the OCXO is installed and if the DIO-64 is inserted in the Star Trigger Slot, Slot 2. Valid settings are enabled or disabled.
The default is disabled.</td></tr>
<tr><td>Scan clock Trig 0 enable</td><td>This attribute controls whether the DIO-64 will drive the scan clock out RTSI/PXI Trigger 0. Valid settings are enabled or disabled.
The default is disabled.</td></tr>
<tr><td>Start trigger Trig 2 enable</td><td>This attribute controls whether the DIO-64 will drive the Start trigger out RTSI/PXI Trigger 2. Valid settings are enabled or disabled.
The default is disabled.</td></tr>
<tr><td>Stop trigger Trig 3 enable</td><td>This attribute controls whether the DIO-64 will drive the Stop trigger out RTSI/PXI Trigger 3. Valid settings are enabled or disabled.
The default is disabled.</td></tr>
<tr><td>Daq modulo Trig 4 enable</td><td>This attribute controls whether the DIO-64 will drive the DAQ modulo clock out RTSI/PXI Trigger 4. Valid settings are enabled or disabled.
The default is disabled.</td></tr>
<tr><td>PXI Star enable</td><td>Not currently available</td></tr>
<tr><td>Port routing</td><td>This attribute allows the DIO-64 to redirect RTSI trigger lines onto PORT A bits.
<table>
<tr><td>Bit 0</td><td>TRIG 1 to port A 13 input</td></tr>
<tr><td>Bit 1</td><td>TRIG 5 to port A 14 input</td></tr>
<tr><td>Bit 2</td><td>TRIG 6 to port A 15 input</td></tr>
<tr><td>Bit 3</td><td>Port A 13 output to TRIG 1</td></tr>
<tr><td>Bit 4</td><td>Port A 14 output to TRIG 5</td></tr>
<tr><td>Bit 5</td><td>Port A 15 output to TRIG 6</td></tr>
</table>
The default is 0.</td></tr>
</table>

<table>
<tr><td>Static output ports</td><td>This attribute is a bit mask that specifies whether an output port is updated from the FIFO or from an Output Force operation.<table><tr><td>Bit 0</td><td>Port A</td></tr><tr><td>Bit 1</td><td>Port B</td></tr><tr><td>Bit 2</td><td>Port C</td></tr><tr><td>Bit 3</td><td>Port D</td></tr></table>This can be used to designate an output port as one that is used only by the output force operations.
The default is 0.</td></tr>
<tr><td>Serial number</td><td>This is a read only attribute that returns the DIO-64 serial number.</td></tr>
<tr><td>Rearm enable</td><td>When disabled a stop trigger will stop and disarm the operation in progress. When enabled, a stop trigger will stop the operation in progress, but leave it armed. This allows the next start trigger to restart the operation.
The default is disabled.</td></tr>
<tr><td>SCLK enable</td><td>This attribute controls whether the scan clock is driven out the external connector pin 19. Disabling this SCLK output can limit the noise generated by high frequency signals through the DIO-64 cable.
The default is disabled. The default can be overridden by the DriveExternalSclk registry entry.</td></tr>
<tr><td>FPGA info</td><td>This is a read only attribute that returns the FPGA info. This information is only for use by Viewpoint product support personnel.</td></tr>
</table>

Performance Hints

General Considerations

There are two main parameters that define the DIO-64 performance: the scan clock rate and the # of scans per second that can be transferred to the host PC. The combination of these two parameters and the application program generating or digesting the resultant data, define the raw data handling capability of the entire system.

The scan clock rate is limited by the ability of the FPGA to execute each step of the operation that it is asked to perform. For input operations the FPGA must sample the inputs, compare with the last value, push the data onto the FIFO if it has changed. For the internally generated onboard clock the top scan clock rate is 20 MHz. For other clocks (external, via RTSI/PXI, OCXO) the frequency limit is 16 MHz.

The PCI bus and the ability of the host PC to interact with the DIO-64 mainly limit the # of scans per second that your application will be able to acquire or generate. The PCI bus is theoretically capable of transferring 132 Mbytes per second. The DIO-64 hardware is capable of saturating the PCI bus in the computer. The actually rate depends on the other devices in your system and how much data they need to transfer across the bus. There also is a certain amount of overhead in the PCI protocol and consumed by the DIO-64 driver in interacting with the DIO-64.

Do not underestimate the horsepower required to process the DIO-64 data streams when an application pushes the scans per second performance of the system. Assume that an application sampling a 4 port data stream that generates scans at ~10 Mscans/second. This translates to ~120 Mbytes per second, which would in itself, stress the PCI bus. If an application expected to pick up data from the input FIFO every 10 milliseconds (100 Hz), it would have to process 100,000 scans in the next 10 milliseconds in order to be ready for the next round. Managing a constant high scan/second data stream requires careful programming.

A data stream with sporadic scans will not stress the system nearly as much a one with a constant scans. Encoders and clocks generate a relatively constant stream of data.

Cable issues

The current cable also places a limitation on the practical upper limit of the scan clock. The current cable has a maximum bandwidth of ~10 MHz. It is not advised to push higher frequency signals through this cable. If your system needs higher bandwidth, call Viewpoint and ask about a custom cable.

Priority between Input and Output operations

In order to maximize the performance of the DIO-64 when performing input and output simultaneously, the application needs to inform the DIO-64 driver which aspect is more important. By default, the driver assumes that the input acquisition has higher priority over the output side. This can be overridden through a DIO-64 attribute. Use the *DIO64 Set Attribute* VI to set the desired priority. The input and output mode attributes control which side of the acquisition receives the highest priority. Only one of these attributes can be set to *demand*. The driver will force the other attribute to *packet* mode if there is a conflict. There is only one demand DMA resource available to the DIO-64 hardware.

FPGA Control program

Loading the appropriate custom FPGA program through the *DIO64 Load* VI programs the FPGA on the DIO-64 board. The *dio64.cat* FPGA program catalog file actually contains 5 different versions of the FPGA program. Each version works with a different combination of input and output port configurations.

It is important to realize the side effects that occur when the DIO-64's FPGA is reprogrammed:

- The FPGA is reset, which has the effect of changing all of the DIO64 digital I/O and RTSI/PXI lines as inputs.
- It takes ~1-2 seconds to reprogram the chip.
- All I/O lines are left as inputs.

The input and output hint parameters on the *DIO64 Load* VI allow the driver to determine which FPGA program is loaded. Hints are required; the following permutations are allowable (Input/Output hints): 4/0; 3/1; 2/2; 1/3; 0/4. Later, the *DIO64 Out Config* and *DIO64 In Start* VIs check that a correct FPGA program was loaded so that the application can continue.

NOTE: The *DIO64 Load* VI ignores any attempt to load files with the extension .bnm (the extension used by DIO-128 DSP control programs). If an application attempts to load a .bnm file, the *DIO64 Load* VI will load from the default DIO-64 FPGA program catalog. This was done in order to accommodate DIO-128 programs that have been upgraded to use the DIO-64.

Driver Buffer Allocation

The default I/O buffers allocated by the driver at driver start-up are contiguous in memory. This allows the DIO-64 to perform DMA operations in a very efficient manner.

The buffers used by the driver can be resized through a DIO-64 attribute. When the buffers are resized, the driver must request a new memory block from the OS. The OS will attempt to satisfy the request with memory that may not be contiguous in memory. This has a slight performance penalty, because during a DMA I/O operation the DIO-64 must constantly figure out where in memory the data belongs.

The default input and output buffer allocation sizes can be overridden through a setting in the registry. The following DWORD registry values set the input or output buffer sizes in bytes.

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Dio64\Parameters\DefaultInputBufferSize
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Dio64\Parameters\DefaultOutputBufferSize

The hard coded default input buffer size is 2097152 bytes.

The hard coded default output buffer size is 524288 bytes.

NOTE: These settings apply to all DIO-64s in the PC. If there are multiple DIO-64s and a large buffer size is requested, the OS may not be able to satisfy all the requests. The hard coded default buffer size is then used instead.

Differences between DIO-64 and DIO-128

Signal Differences

The external start and stop triggers (pin #s 24 and 25) on the DIO-64 are not pulled up on Rev A PCI DIO-64 card (part # 170000-0001 Rev A).

The PXI version has 10k pull-up resistors on the external start and stop triggers.

The DIO-64 resets all ports to inputs upon *DIO-64 Close* and/or application exit. This allows the pull-up/pull-down resistors to return the signals to the desired "inactive" state. The DIO-128 used to keep driving whatever output state the digital outputs were at when the application closed. This DIO-128 output behavior can be restored by using the following registry entry:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Dio64\Parameters\ DriveOutputsOnExit

A DWORD value of 1 will enable the old, DIO-128 behavior.

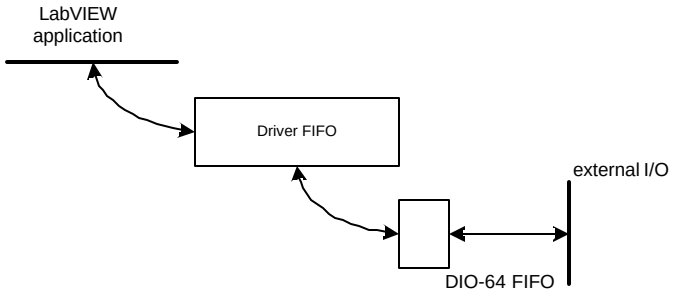
Mode differences

No *Simple vs. Master* Modes

The DIO-64 does not have the concept of simple vs. master/slave modes. The FPGA lets the DIO-64 operate at full speed no matter what combination of features are in use.

FIFO size

The DIO-64 FPGA and driver uses a different configuration FIFO buffers than was found on the DIO-128. The FPGA has a smaller FIFO at its disposal, and the driver adds a new driver FIFO to the picture. A DIO-128 system read and wrote data directly to a medium sized FIFO located on the DSP.The DIO-64 driver allocates a larger FIFO from the PC memory and uses a variety of methods to move data to and from the DIO-64.



VI differences

Function Invalid at this time

The DIO64 library has definite rules as to when each function can be called. See the Mode descriptions (page 8) and the individual VI documentation for details. The DIO128 was more forgiving of applications that executed functions in an order other than what is described in the manual.

DIO64 Open.vi

The *DIO64 Open* VI no longer uses the base I/O parameter.

DIO64 Load.vi

The parameters of the *DIO64 Load* VI are different from what was used with the DIO128, but the basic behavior is the same. The *DIO64 Load* VI no longer uses the filename input, the driver expects to use the *DIO64.CAT* file found in the SYSTEM32 subdirectory.

The *DIO64 Load* VI also has 2 additional required input terminals. The *input hint* and *output hint* parameters indicate to the driver the FPGA Input/output mix to use on the DIO-64 board. The following permutations are allowable (Input/Output hints): 4/0; 3/1; 2/2; 1/3; 0/4.

DIO64 In Start and DIO64 OutConfig VIs

These VIs now have both the scan rate and divider inputs. The scan rate input can be used when the clock rate of the Major Clock Source is known. The driver will calculate the divider to be used during the I/O operation. If the application specifies the divider directly, this is what will be used during the I/O operation.

DIO64 Out Force Output.vi

This VI now allows you to specify ports that you want affected. The data element will be driven out the output port if the corresponding element in the output mask is true.

U16 to U32

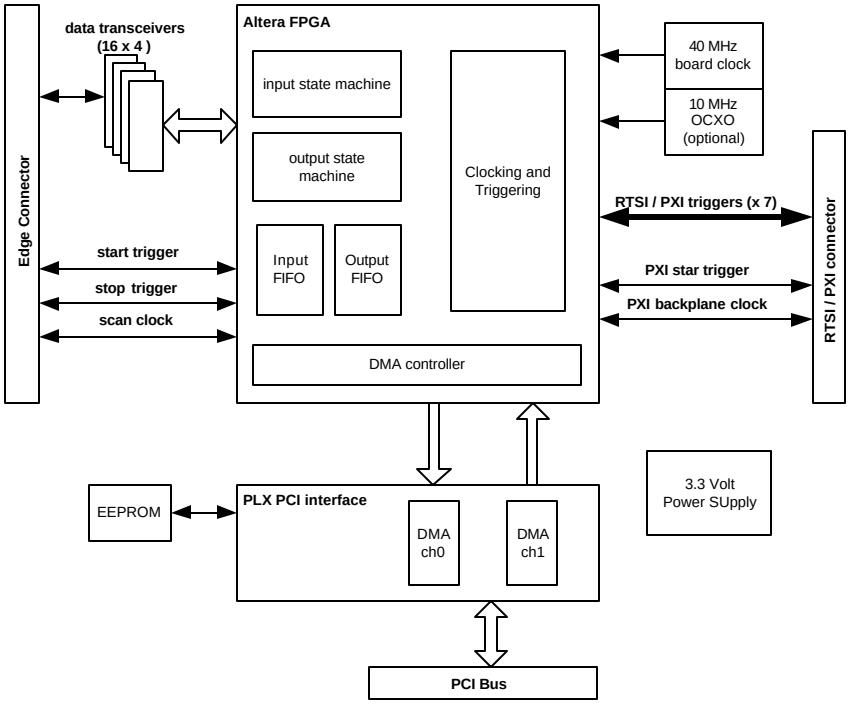
In order to accommodate the larger FIFOs found in the DIO-64 driver, certain VI controls and indicators have been converted from U16 to U32. Most times this change will not affect the algorithm being used, but it is a good idea to double check the code that may use these values. The signals that have been modified are: s*cans available* found in the *DIO64 In Status* and *DIO64 Out Status* VIs and *scans to read* in the *DIO64 In Read* VI.

The clusters found in the DIO64 In Status and DIO64 Out Status VIs also have elements that have changed from U16 to U32.

New Errors generated

The DIO-64 versions of *DIO64 In Status* and *DIO64 Out Status* will generate buffer overflow (on inputs) or buffer underflow (on output) errors. The DIO-128 VIs did not generate these errors.

DIO-64 Block diagram



DIO-64 external connector assignments

A0	1	51	C0
A1	2	52	C1
A2	3	53	C2
A3	4	54	C3
A4	5	55	C4
A5	6	56	C5
A6	7	57	C6
A7	8	58	C7
A8	9	59	C8
A9	10	60	C9
A10	11	61	C10
A11	12	62	C11
A12	13	63	C12
A13	14	64	C13
A14	15	65	C14
A15	16	66	C15
GND	17	67	GND
GND	18	68	GND
SCLK	19	69	GND
ExtClk	20	70	GND
DIO Enabled	21	71	GND
Data Changed	22	72	GND
GND	23	73	GND
Start Trig	24	74	GND
Stop Trig	25	75	GND
GND	26	76	GND
GND	27	77	GND
GND	28	78	GND
GND	29	79	GND
GND	30	80	GND
GND	31	81	GND
GND	32	82	GND
GND	33	83	GND
GND	34	84	GND
B0	35	85	D0
B1	36	86	D1
B2	37	87	D2
B3	38	88	D3
B4	39	89	D4
B5	40	90	D5
B6	41	91	D6
B7	42	92	D7
B8	43	93	D8
B9	44	94	D9
B10	45	95	D10
B11	46	96	D11
B12	47	97	D12
B13	48	98	D13
B14	49	99	D14
B15	50	100	D15

Hardware Details

FPGA

The heart of the DIO-64 is an Altera **F**ield **P**rogrammable **G**ate **A**rray (FPGA). The FPGA is a sophisticated logic device that is capable of being "programmed" in the field (after manufacture). These devices are capable of operating at much higher speeds than programs running on DSPs or microprocessors.

The DIO-64's FPGA is loaded with the appropriate program for the combination of input and output ports being used for any given application. It is important to realize that until the application has indicated the I/O configuration that will be used, all I/O pins are designated as inputs. This has implications that must be taken into account when using the DIO-64 for output applications.

PCI bus

The PCI-DIO-64 is a PCI 2.2 compliant, 32 bit, 33 MHz, 5 Volt data acquisition card. It uses a PLX 9054 PCI interface chip that provides 2 DMA channels capable of transferring data at full PCI data rates. The PCI-DIO-64 supports National Instrument's RTSI inter-board timing and trigger bus. RTSI enables sharing of clocks and triggers between multiple RTSI compatible boards in a system.

PXI bus

The PXI-DIO-64 is a CompactPCI/PXI, 3U, 32 bit, 33 MHz, 5 Volt data acquisition card. It uses a PLX 9054 PCI interface chip that provides 2 DMA channels capable of transferring data at full PCI data rates. When used in a CompactPCI system the PXI specific features (trigger lines and clock) are not available. When used in a PXI compatible system, the PXI-DIO-64 can be used as a star-trigger controller, a chassis-timing source (with optional OCXO), or in any PXI peripheral slot. More information on PXI can be found at http://www.pxisa.org

Additional LabVIEW VIs

Example VIs

These example VIs are found in *dio64 examples.llb*.

DIO64 Simple One Board Input.vi

This VI demonstrates a common method to configure and read from the DIO-64. All monitored ports are collected and a single channel is displayed. Parameters in the section above the graph must be set before the VI is started for their values to take effect.

DIO64 Cont Input to Disk.vi

This VI demonstrates how to continuously monitor and write data to a binary file. A bit mask is used and the number of state changes is monitored. All parameters, except Output Data File, must be set before running the VI for their values to take effect.

DIO64 Read Data from Disk.vi

This VI is a reader for data collected with Cont DIO64 Input to Disk.vi. The file data is read and displayed in a bit pattern and a single channel is displayed graphically. The speed of playback is adjustable and the scan number and corresponding time are shown. The port and bit to view on the graph must be set before running the VI.

DIO64 with Analog RTSI.vi

This VI demonstrates simultaneous digital and analog input acquisition with synchronization through the RTSI bus. The analog acquisition is clocked at a modulo of the digital acquisition. All parameters in the sections above the graph must be set before the VI is started for their values to take effect.

DIO64 Output Data.vi

This VI illustrates the DIO-64 output capabilities for "short" duration data. A data set can be created and then output once, continuously, or for a specific number of repetitions.

DIO64 Output Data from Disk.vi

This VI is an example of streaming "long" duration data using the DIO-64 output capabilities. A user-specified data file captured using the Cont DIO64 Input to Disk.vi is replayed.

DIO64 Simple Output/Input Stimulus Test.vi

This VI is an example of performing both input and output on the same DIO-64 board. It allows for the creation of the stimulus data set and for graphing the response.

Utility VIs

The following VIs are located in the file *dio64 tools.llb*.

Bit Mask to Word Mask.vi

This VI converts a bit mask pattern to a word mask pattern. It is useful to select the bits to sample on a bit-by-bit basis yet the DIO-64 is configured with 16-bit words. This VI provides the necessary conversion.

DIO Bit to SCXI Mod,Chan.vi

The arrangement of bits on the DIO-64 is different than that of National Instruments AT-DIO-32F. This is not significant unless you are using a channel labeled device such as the SCXI-1162/1326. In this case, you can use this VI along with the SCXI Mod,Chan to DIO Bit.vi to map the DIO-64 to the AT-DIO-32F.

DIO Index to Port,Bit.vi

This VI allows you to determine the port number and bit offset within that port given a channel number of any bit on the board. The type of port, input or output, has no effect on the results.

Extract Bit.vi

Given a two-dimensional array of digital data as well as a desired bit, this VI will return a one-dimensional array of data collected on that bit.

SCXI Mod,Chan to DIO Bit.vi

This VI will map an SCXI channel to a DIO-64 channel. See also DIO Bit to SCXI Mod,Chan.vi.

Separate Data.vi

This VI separates the data returned from a DIO-64 read into an array of timestamps and a two-dimensional array of either 16-bit words or boolean bit values.

Word, Mask to Bit Values.vi

The purpose of this VI is to format the data collected with a DIO-64 to an easily readable format. The data collected will be ANDed with the selected channel mask and returned as a boolean array.

DIO64 Error Codes

Error #	Description
-8	Illegal board number - the board number must be between 0 and 7.
-9	The requested board number has not been opened.
-10	The buffers have over or under run.
-12	Invalid parameter.
-13	No driver interface.
-14	Board does not have the OCXO option installed.
-15	Only available on PXI.
-16	Stop trigger source is invalid.
-17	Port number conflicts. Check the hints used in DIO64 Load.vi.
-18	Missing DIO64.cat file.
-19	Not enough system resources available.
-20	Invalid DIO64.cat file.
-21	Required image not found.
-22	Error programming the FPGA.
-23	File not found.
-24	Board error.
-25	Function call invalid at this time.
-26	Not enough transitions specified for operation.

Diagnostics

Problem: When running one of the example VIs that graphically displays data, I set the digital Port to View and Bit to View but I still see no change of state on the graph.

Solution: The bit mask that appears above the graph must have this bit active (red) to see state changes in the graph. This bit mask must be set before running the VI. Also, Ports to Analyze must be set high enough to include the Port to View.

Problem: I get an error that I am trying to load a VI that was created with a later version of LabVIEW.

Solution: The DIO-64 LabVIEW VIs are written for use by LabVIEW versions 6.0 and later.

Problem: When running the *DIO64 with Analog RTSI* VI, nothing happens when signals are applied to the analog and digital inputs and the time scale does not change.

Solution: Verify that the RTSI connector is attached to the DIO-64 and the analog input board being used. Ensure that the connector is plugged in all the way.

Problem: When running the *DIO64 In Start* VI or the *DIO64 Out Config* VI, I get error -12, invalid parameter. What caused this error?

Solution: Most likely, a parameter in the Start Control or Config Control is not supported. This is often because you are choosing an option that is not valid for the DIO-64 board you have. PXI specific feature are not support by the PCI-DIO-64. Any settings that use the OXCO are only valid if your DIO-64 has the OCXO installed.

Specifications

These specifications are typical for 25 ºC unless otherwise noted.

Digital I/O

Number of channels ..... 64 input/output (4 banks of 16 bits)
4 control pins
Compatibility ..... 5 V TTL / CMOS

Digital logic levels

Level	Minimum	Maximum
Input low voltage	-0.5 V	0.8 V
Input high voltage	2 V	5.5 V
Input leakage current	± 01.uA	±5uA
Sink Current		24 mA
Setup time to scan clock	5 ns	
Output low voltage (I_{out}=24mA)	-	0.55 V
Output high voltage* (I_{out}=24mA)	4.2 V	-
Source Current		24 mA
Propagation delay from scan clock		25 ns

Power-on state for outputs ..... High-impedance,
pulled up or down,
10 kOhm (selectable, every 8 bits)

PCI, PXI..... DMA, interrupts, programmed I/O

Operation

The DIO64 input mode monitors a specified number of digital input bits at a particular scan rate. This scan rate determines the minimum pulse duration that can be sampled. If any digital input changes from one scan to the next, the DIO64 saves the current time and state of all the digital inputs in a FIFO. This FIFO is read by the application, and the sampled data can be processed.

The DIO64 output mode allows an application to describe a digital waveform that will be driven out the digital output bits. The application describes this waveform as a series of scans. Each scan has a timestamp and the digital data. The DIO64 will drive the next scan out when its timer has reached the time specified by the scan's timestamp. The DIO64 also allows the waveform to repeat for a fixed or continuous number of times.

The DIO64 can perform both input and/or output modes without sacrificing performance.

Direction ..... Input or output in the following permutations:

Bank			
A	**B**	**C**	**D**
In	In	In	In
Out	In	In	In
Out	Out	In	In
Out	Out	Out	In
Out	Out	Out	Out

Performance Benchmarks

		Input/Output	
		16 or 32 bit	**48 or 64 bits**
Max Scan Rate	Internal Clock	20 MHz	
	External Clock	16 MHz	
Max Burst Transfer Rate (filling internal FIFO, < 512 scans, millions scans/second)		20.0	
Max Sustained Transfer Rate (filling driver's FIFO, millions scans/second)		12.5	8.0
Max Continuous Transfer Rate (filling application buffers, millions scans/second, benchmarked on Dell Dimension GX 240, 1.8 GHz PC)		1.6	1.0

Memory

Input ..... 512 scan onboard FIFO
Output ..... 512 scan onboard FIFO

Clock Sources

Major clock sources ..... 40 MHz crystal (±100 ppm)
External clock
RTSI Clk (PCI only)
PXI Clock (PXI Only)
Optional 10 MHz oven controlled crystal (±100 ppb)

Bus Interfaces

PCI, PXI ..... Master, slave

RTSI Features (PCI only)

Trigger lines ..... Trigger 0-6 as I/O via internal Register
RTSI Clock ..... generate and use

PXI Features (PXI only)

Trigger lines 7
Star Trigger ..... controller and peripheral
PXI 10 Clock ..... generate and use

Power Requirements

	+5 VDC (+/-10%)*	
Device	**Typical**	**Maximum**
PCI/DIO-64	1 amp	4.5 amp
PXI/DIO-64	1 amp	4.5 amp

Physical

Dimensions.....**PCI** 6.9 by 4.2 in. (17.5 by 10.7 cm)
PXI STD 3U

Connectors

External.....100 pin 0.05” center female D subminiature
RTSI.....34 pin 0.1” center male ribbon (PCI only)

Environment

Operating Temperature .....0-70 ºC

Oven-Controlled Crystal Oscillator (OCXO) (optional)

Frequency.....10 MHz
Stability .....100 ppb
Duty Cycle .....40% to 60%
Warm-up time (to within 0.1 ppm of
operating frequency).....3 minutes @ 25 ºC
Frequency stability versus supply
Aging.....±4ppm in 10 years max, ±1ppm/year
Power Supply effect .....±0.06ppm max
Optional frequency adjustment .....±15 ppm range

Note: You can use the OCXO to replace the PXI 10 MHz backplane clock when the DIO-64 is installed in the PXI star trigger slot. (PXI only)

Contacting Us

Viewpoint Systems, Inc.

800 West Metro Parkway

Rochester, NY 14623

voice: 585-475-9555

fax: 585-475-9645

e-mail: support@ViewpointUSA.com

Technical support is available any business day from 9:00 AM to 5:00 PM Eastern time. Of course, you may fax or e-mail questions at any time.


VIEWPOINT
SYSTEMS