DIO-128 User Manual

Dynamic Digital I/O System

Viewpoint Software Solutions, 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 SOFTWARE SOLUTIONS, 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 SOFTWARE SOLUTIONS, 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 Software Solutions, Inc.
2320 Brighton Townline Road
Rochester, NY 14623

Printed in the U.S.A.
June 1998
Part Number: VSS-128002

Table of Contents

1. Introduction

The DIO-128 Dynamic Digital I/O System combines an intelligent plug-in data acquisition (DAQ) card with customized driver and application software. The DIO-128 card provides 128 TTL-compatible digital lines for a wide variety of high speed and high channel count digital applications. The supplied software includes a LabVIEW® user interface and digital signal processing (DSP) code specialized for time-interval measurements, state change monitoring, time-interval based output, and signal generation.

DIO-128 features include:

- High speed: The DIO-128 is designed to measure time intervals as short as 500 nanoseconds.
- High channel count: The DIO-128 gives you the ability to group boards together for large system timing analysis problems. Up to 128 input or 64 output channels are available per board. The high channel density per slot allows for easy PC system configuration. The maximum number of boards in the system is only restricted by the number of free slots available in the PC.
- Intelligence: The onboard Analog Devices 2181 16 MHz DSP with 16 kword data and 16 kword program memory is capable of 32 MIPS. The large data memory space allows for a buffer of approximately 16,000 words, alleviating many Windows™ programming problems.
- RTSI® interface: This hardware bus allows for accurate triggering and timing between combinations of DIO-128s and other RTSI compatible data acquisition cards.
- Compatibility: The DIO-128 can be used with all signal conditioning options available for National Instruments' AT-DIO-32F™ as well as Viewpoint Software's OPTO32-128 digital isolation module.

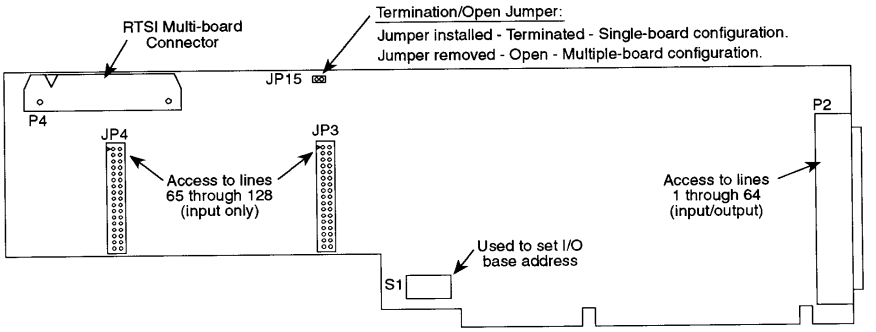
- Driver software included: All the software you need to use the board with LabVIEW to create high-level applications is included (DSP control programs, Windows interface DLLs, LabVIEW VIs). The driver software supports Windows 95 and Windows NT 4.0.
- Examples: A large list of LabVIEW example VIs provide a great starting place for a variety of applications.

2. Getting Started

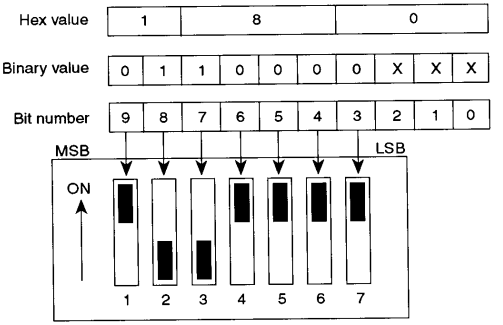
The following sections describe the steps necessary to install and configure your DIO-128 system.

2.1 Hardware Configuration

The DIO-128 is based on Dataq Instrument's DI-660 digital I/O board. The DI-660 is an I/O mapped device requiring eight bytes of I/O address space in the range from 100 (hex) to 3F0 (hex). A 100-pin connector on the back of the board and two 34-pin connectors (JP3 and JP4) on the board provide the signal interface.



The I/O base address for the DI-660 is set with the switches of the S1 switch block. These switches represent bits 3 through 9 of the base address (bits 0 through 2 do not matter) and are used to turn an address line either on or off. Valid I/O base addresses are from 100 (hex) in increments of 8. A switch in the ON position selects a 0 for the bit it represents. A switch in the OFF position selects a 1 for the bit it represents. The value of an address line exists only when the switch is OFF. The following example illustrates a base address setting of 180 (hex).



Refer to the I/O Connector Pinouts section on page 44 for pin descriptions.

The board setting of 180 (hex) is the default board address. Before installing the board in your computer, make sure that the selected I/O space is free. If the space is not free, error -7 will occur when you run the DIO128 Load.vi. Use the following procedure to find out if an I/O space is free in Windows 95: Click the right mouse button on My Computer and select properties. Select the Device Manager tab then click on properties (Computer should be selected). Select Input/output (I/O) and scroll down to see if a desired I/O space is used.

Recommended cabling for this board includes a National Instruments NB5 cable to interface from the P2 connector on the back of the DIO-128 board to a pair of CB-50s. Connect a 34-pin straight-through cable to JP3 and JP4 to interface the high channels of the DIO-128 to 34-pin termination blocks. All of these cables and termination blocks are available through Viewpoint Software.

2.2 Installing the Software

The DIO-128 software requires 3.2 megabytes of free hard disk space. The software is installed with a Windows setup program as follows:

1. Place the DIO-128 diskette in a 3.5" floppy drive.
2. Select the Start button and choose Run.
3. Type A:\SETUP.EXE and press ENTER to install the software.

You will be prompted as to whether you are running Windows 95 or Windows NT. Select the proper operating system to ensure proper software installation.

A destination path for the files will be requested. The default path is C:\DIO128.

Note: Windows NT requires you to be logged in with administrator rights to successfully install the DIO-128 software.

Installed Files

The setup program copies the following files to your system.

DIO128_32.DLL

This DLL is copied to the System directory under the current version of Windows. This DLL is needed for the software to communicate with the DIO-128 board.

DIO128_95.DLL, DIO128_NT.DLL

These files work in conjunction with DIO128_32.DLL and are optimized for Windows 95 and Windows NT. Only one of these files is installed, based on which operating system is being used.

NEXTIN.BNM, NEXTOUT.BNM, NEXTIO.BNM

These files contain DSP code that is downloaded to the DIO-128 with the DIO128 Load.vi. These files are also copied to the System directory under the current version of Windows. NEXTIN.BNM is used for input-only, NEXTOUT.BNM is used for output-only, and NEXTIO.BNM is used for simultaneous input and output.

DIO128_32.LLB

This file is a LabVIEW VI library that contains the driver VIs described in the Driver VIs section on page 27.

DIO128.SYS

This is a system file used by the DIO128 software.

DIO128CONFIG.EXE

This file is for use configuring the DIO-128 board under Windows NT. This file is not installed on systems running Windows 95. See The DIO-128 Configuration Progam on page 7 for more information on using this program.

README.TXT

Please review this file for last minute additions and changes.

VER1.X\DIO12832.LLB
This file contains an LLB that contains VIs that were included with previous input-only versions of the software. The VIs do not reflect the new VI names introduced in version 2.0. This LLB is intended to be used by those who have existing applications written previous to version 2.0 to upgrade their applications. These VIs can be loaded without breaking any wires on the diagram. The user can then perform a replace operation on the old VIs with the new 2.0 VIs.

EXAMPLES\DIOEXMPL.LLB
This file is a LabVIEW library that contains the example VIs described in the Examples section on page 40.

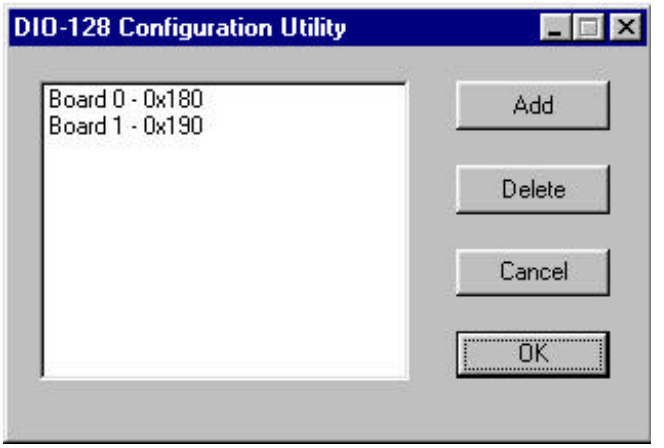
EXAMPLES\DIOTOOLS.LLB
This file is a LabVIEW library that contains tools for use with the DIO-128 board. The VIs in the library are described in the Utility VIs section on page 41.

2.3 The DIO-128 Configuration Program (Windows NT only)

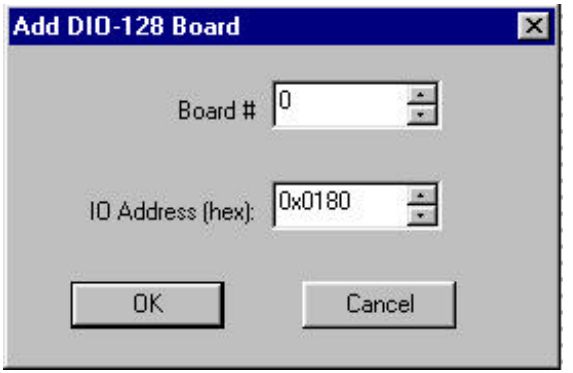
Note: The DIO-128 configuration program is for use with Windows NT only. You must login to Windows NT as administrator prior to running this program. This program should NOT be used in Windows 95. The configuration process is performed by the DIO128 Open.vi for Windows 95 users.

This program allows the user to configure the software for the DIO-128 hardware settings (i.e., I/O address). This step needs to be done only once or whenever you need to change your hardware setup.

The interface for the DIO-128 Configuration Utility is shown below.



No boards will be listed if you are running the configuration program for the first time. Select Add to add a new board to the board list. When you select Add, the add dialog box opens to add a new DIO-128 board.



In this dialog, you must assign a unique number to the DIO-128 board and specify its I/O address (see the Hardware Configuration section on page 3 for more information on I/O addresses). The board number will be used in any LabVIEW programs to identify the board you want to access. Valid board number values are 0 through 7. The I/O address must be the same as the I/O address set on the board. Valid I/O address values are 0x100 through 0x3F0. The default settings of board number 0 and I/O address 0x180 should be correct for most single board installations. If multiple boards are used, you must add each board separately. You may arbitrarily assign unique board numbers, but it is customary to assign the first board to be board number 0, the second board number 1, etc. No two boards can have the same I/O address or board number.

After you have added all your boards, verify that the information is correct, and then select OK to save the settings to the registry. If some of the information is incorrect, select the board for which the information is incorrect and select Delete. You can then select Add to add the board again and specify the correct information.

Whenever you make a change to the configuration, you must reboot your computer for the changes to take effect.

2.4 System Setup Verification

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

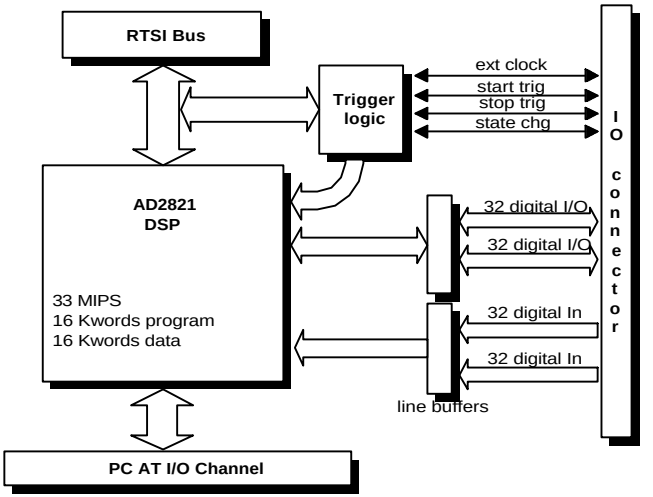
1. Start LabVIEW
2. Open and run the file Simple One Channel.vi which is located in \DIO128\EXAMPLES\DIOEXMPL.LLB.
3. Toggle the digital input lines either by using a function generator or by temporarily grounding one of the digital I/O lines by connecting it to pin 17. If the graph is set to show the 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.

If you experience any problems, refer to the Troubleshooting section starting on page 52.

3. Theory of Operation

3.1 Overview

The board used by the DIO-128 Dynamic Digital I/O System is a high channel count, intelligent board capable of solving many types of digital acquisition tasks. The onboard DSP allows the board to handle tasks that would be difficult or impossible to accomplish by relying only on the main PC CPU. The DIO-128 hardware was designed to be used in conjunction with other data acquisition hardware with extensive clocking/triggering options via RTSI and external connections. In order to acquire state change data from more than 128 channels of digital data, the DIO-128 has been designed to allow multiple boards to work together to handle larger state change problems. The DIO-128 complements its state change monitoring functionality with output capabilities that make it well suited for stimulus-response style acquisition and control applications.



DIO-128 Block Diagram

3.2 Board I/O

Digital Channels

The digital channels are organized into eight 16 channel ports (A through H). All eight ports are useable as inputs, but only ports A, B, C, and D can be used as outputs. 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.

Port	A	B	C	D	E	F	G	H
Connector	P2	P2	P2	P2	JP4	JP4	JP3	JP3

Connector P2 is the 100-pin rear connector, connector JP4 is the 34-pin connector under the RTSI connector, and connector JP3 is the 34-pin connector closest to the rear connector (P2).

The DIO-128's 128 digital inputs are interfaced through transceivers that latch the current state of all inputs with the digital scan clock, ensuring that a scan will not show any timing skew. These transceivers expect TTL level inputs and are pulled high through 4.7 kΩ resistors.

The DIO-128's digital outputs are also TTL compatible and are latched on write operations. The outputs use the same pull-up resistors (4.7 kΩ) that the inputs use.

External Clocking/Triggering

The DIO-128 provides for external connections to the digital scan clock and the start and stop triggers. These connections are pulled high through 4.7 kΩ resistors.

RTSI

The DIO-128 includes a RTSI connector to provide multiple board synchronization and triggering. The DIO-128 can source or sink its digital scan clock and start and stop triggers over the RTSI connection. RTSI trigger 1 is a special open collector connection for use with multiple DIO-128 boards. This line is pulled high through a pull-up resistor on the master DIO-128. Any board that wishes can pull the line low and all boards will see the change.

RTSI Trigger #	Use	Edge Sense
0	digital scan clock	Falling
1	DIO-128 voting line, open collector	N/A
2	start trigger	Selectable
3	stop trigger	Selectable
4	general purpose	N/A

The RTSI connector mirrors any activity on the clocking/triggering connections of the DIO-128. For instance, if the board has been configured for internal clocking, the digital scan clock is presented on RTSI trigger 0. If the board has been configured for an external start trigger, that trigger is presented on RTSI trigger 2. The Edge/Level sense for start and stop triggers are respectively set in DIO128 In Start.vi and DIO128 Out Config.vi. See the Driver VIs section starting on page 27 for more information on selecting the edge or level these triggers sense.

Please note that the DIO-128 cannot disconnect fully from the RTSI connections. The system should not be configured so that multiple boards are attempting to drive the same RTSI trigger line. When the DIO-128 is stopped, all RTSI connections are forced to be inputs.

The RTSI interface opens up opportunities for mixing DIO-128 boards with other acquisition hardware. There are a large number of acquisition boards available with RTSI connectors. It is possible to combine digital and analog acquisition to provide a hybrid data monitoring system. In these scenarios, the DIO-128 might provide the analog scan clock via RTSI trigger 4. The state change control

program can clock RTSI trigger 4 as a divided down digital scan clock. Another possibility would be to have the RTSI capable analog acquisition card provide the digital scan clock and/or start and stop triggers.

Refer to the DIO128 with Analog RTSI.vi example to see a sample of combining the DIO-128 with another data acquisition board.

4. DSP Control Programs and Board Modes

The Analog Devices ADSP-2181 DSP is the heart of the DIO-128 system. It controls every aspect of the digital acquisition process. The DSP internally doubles its 16 MHz clock to provide 32 MIPS sustained performance and it allows for single cycle interrupt latency. The ADSP-2181 has 16k words of on-chip instruction RAM and 16k words of on-chip data RAM. The DIO-128 communicates between the ADSP-2181 and the host PC through the DSP's DMA port and the PC's I/O bus.

The DIO-128 can be configured for different personalities or board modes by the selection of the DSP control program. The DSP control program is not hard coded into the system. It is loaded through a simple loader before starting any application. After loading, the control program is started and a verification step is performed to ensure that the host and DIO-128 are communicating properly.

Three control programs are provided which are used to optimally perform input-only, output-only, and combined input and output operations. There are a large number of other digital acquisition problems that could be solved with the DIO-128 and the correct control program. Viewpoint Software can provide other custom solutions. Please feel free to contact us with your unique digital acquisition problems.

4.1 Input-Only Operation

Basic Operation

The input-only (state change detection) DSP control program (NEXTIN.BNM) performs high speed digital sampling and logs 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.

With the input-only control program provided, the DSP on the DIO-128 handles two concurrent tasks. The highest priority task is the digital acquisition task which monitors the digital channels and trigger lines. This task is driven by an onboard timer or externally through the external clock or RTSI trigger 0. The other task handles communication with the host PC application.

Programming Steps

Programming applications for input-only operation involves running VIs in the following order for *each* DIO-128 board:

1. DIO128 Open.vi - Initialize the board.
2. DIO128 Load.vi - Download the DSP software to the board.
3. DIO128 In Start.vi - Start acquiring data to the board's buffer.
4. DIO128 In Status.vi - Find out how much data is waiting to be read.
5. DIO128 In Read.vi - Read in the data from the board's buffer.
6. DIO128 In Stop.vi - Stop acquiring data.
7. DIO128 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.

Modes of Operation

An acquisition run can consist of one or more of the following options, depending on the mode of operation:

	Mode	
Options	Simple	Master/Slave
Basic	X	X
DAQ Clock Modulo	X	X
Start Trigger	X	X
Stop Trigger		X

From the table above, choose one of the modes listed below:

- Simple: The simple mode allows for single DIO-128 board operation only. This mode does not monitor for timestamp overflow. One channel simple mode operation offers the highest sampling rates and can handle the highest number of transitions per second. To increase acquisition performance, one channel Basic and DAQ Clock Modulo operations do not return time values during status operations (i.e. the Time 0 and Time 1 fields of the status block are not updated). All other acquisition modes will update the time values during status operations. See the Driver VIs section starting on page 27 for more information.
- Master/Slave: The master/slave mode allows for single or multiple DIO-128 board synchronized operation. This mode offers more functionality at a slight performance cost. This mode always checks RTSI trigger 1 for state changes. This mode monitors for timestamp overflow and introduces extra scans if necessary. The master DIO-128 board is determined by convention to be the DIO-128 board that is providing the digital scan clock. The slave DIO-128 boards typically use RTSI trigger 0 for the scan clock.

From the table above, choose one or more of the options listed:

- Basic: Basic state change monitoring only
- DAQ Clock Modulo: Drives the RTSI Trigger 4 line with a modulo of the DIO-128 scan clock
- Start Trigger: Waits for a start trigger before starting
- Stop Trigger: Acquires data until a stop trigger is received

The simple acquisition mode removes any overhead involved with monitoring the DIO-128 multi-board state changed RTSI line (RTSI trigger 1) that exists in the master/slave mode, allowing for increased sampling rates.

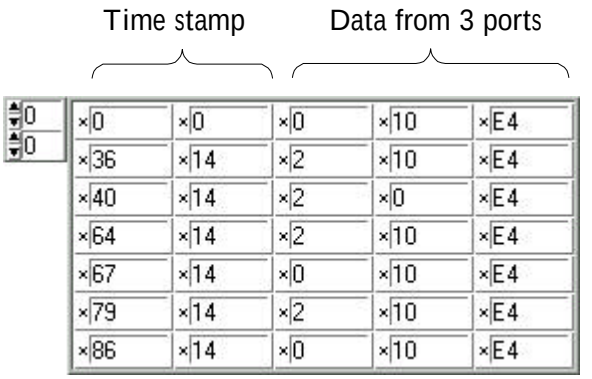
Input Data Format

The data is read back from the DIO-128 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 actual data on the digital inputs at the time the scan was acquired.

In order to ensure that the initial state of all digital channels is known, the state change DSP control program will always acquire a scan of the digital input channels when acquisition begins. If start trigger operation is used, this initial scan will be time stamped with the time the start trigger occurred. If start trigger operation is not used, then the initial timestamp will be zero.

Note that timestamps are the number of digital scan clock ticks that have occurred since DIO128 In Start.vi has been called. For example, if the scan rate is set to 1 kHz, a timestamp value of 1000 will indicate one second, and 7400 will indicate 7.4 seconds. When in the master/slave mode, the control program also 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 is from a 3-port DIO-128 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. You can see that:

At timestamp	with this data	this happened	at time (secs)
0x00000000	0x00E400100000		0
0x00140036	0x00E400100002	bit 2 (A2) on	6.553870
0x00140040	0x00E400000002	bit 20 (B8) off	6.553920
0x00140064	0x00E400100002	bit 20 (B8) on	6.554100
0x00140067	0x00E400100000	bit 2 (A2) off	6.554115
0x00140079	0x00E400100002	bit 2 (A2) on	6.554205
0x00140086	0x00E400100000	bit 2 (A2) off	6.554270

Note: (decimal_timestamp) / (scan rate in Hz) = time in seconds

For a VI that converts data from the DIO-128 to timestamps and boolean or word data, refer to the Separate Data.vi utility in the EXAMPLES\DIOTOOLS.LLB file.

Multiple DIO-128 Boards

In order to acquire state change data from more than 128 channels of digital data the DIO-128 has been designed to allow multiple boards to work together to handle larger state change problems. When using multiple DIO-128 boards, you must select the master/slave mode. The master DIO-128 is designated as the board which supplies the timing signals to the other boards (slaves). RTSI trigger 1 serves as a "voting" line which each DIO-128 asserts if it has detected a state change. At the end of the sampling period, all boards check this line and save the current sample if any board has asserted the RTSI trigger 1 line.

For proper operation, the host PC application must use some care with the sequence in which it programs the various boards in the system. The typical scenario is as follows:

1. Make sure all boards are stopped.
2. Program the start options on any slaves.
3. Program the start options on the master DIO-128.
4. Start data collection.
5. Check the status on the master; see if the buffer contains any data.
6. Read the buffer on the master.
7. For each slave, check the status and use the status and the master's number of scans for the read.
8. Repeat steps 5 through 7 until finished collecting state changes.
9. Stop the master.
10. Stop all the slaves.

Refer to the Simple Two Board.vi for an example of this operation.

4.2 Output-Only Operation

Basic Operation

The output-only DSP control program (NEXTOUT.BNM) performs digital output at specified time intervals. 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.

As with input-only operation, the DSP performs two concurrent tasks for output. The first lower priority task is responsible for communication with the PC, and the second higher priority task controls the digital outputs. Communication with the PC will not delay outputs. A maximum of four output ports can be configured. The number of output ports is determined by DIO128 Out Config.vi.

Programming Steps

Programming applications for output-only operation involves running VIs in the following order for *each* DIO-128 board:

1. DIO128 Open.vi - Initialize the board.
2. DIO128 Load.vi - Download the DSP software to the board.
3. DIO128 Out Config.vi - Configure the output parameters for the board.
4. DIO128 Out Status.vi - Get the current status of the board.
5. DIO128 Out Write.vi - Set the outputs to an initial value, or load in the data to be used for cyclic output (see the Output Repetition section on page 22).
6. DIO128 Out Start.vi - Start outputting data.
7. DIO128 Out Status.vi - Get the current status of the board and find out how much space is left in the board's output buffer.
8. DIO128 Out Write.vi - Send more data to output.
9. DIO128 Out Stop.vi - Stop outputting data.
10. DIO128 Close.vi - Close the selected board.

All steps, except steps 7 and 8, are usually run only once per board in a program. When operating in cyclic mode, steps 7 and 8 are frequently omitted. Otherwise, a program will usually loop over steps 7 and 8 repeatedly until all the desired data has been output.

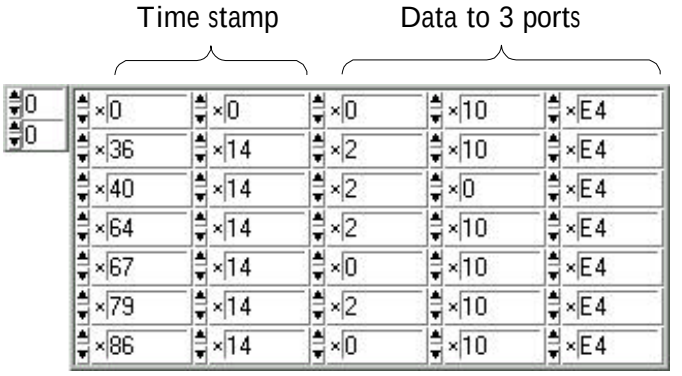
Modes of Operation

The output-only operation only supports the simple operating mode but does not support Master/Slave mode. Start triggering is supported, but stop triggering and DAQ clock modulo are not supported. Multiple boards may be used, but they will operate independently. Please refer to Modes of Operation in the Input-Only section on page 14 for descriptions of the different modes.

	Mode	
Options	Simple	Master/Slave
Basic	X	
DAQ Clock Modulo		
Start Trigger	X	
Stop Trigger		

Output Data Format

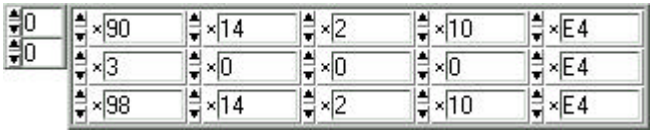
The array of timestamps and data values are the same for output and input. The data values will be output when the counts on the internal clock equal the timestamp. For example, the following array is sent to a DIO-128 running in 3-port continuous mode output with a scan rate of 200 kHz:



The DIO-128 will produce the following output sequence:

At timestamp	this data is output	at time (secs)
0x00000000	0x00E400100000	0
0x00140036	0x00E400100002	6.553870
0x00140040	0x00E400000002	6.553920
0x00140064	0x00E400100002	6.554100
0x00140067	0x00E400100000	6.554115
0x00140079	0x00E400100002	6.554205
0x00140086	0x00E400100000	6.554270

The following array is then sent to the DIO-128:



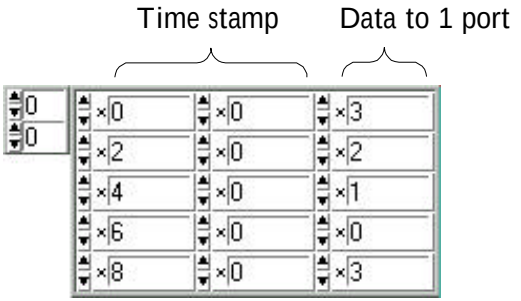
The DIO-128 will produce the following output sequence:

At timestamp	this data is output	at time (secs)
0x00140090	0x00E400100002	6.554320
0x00000003	0x00E400000000	10737.418250
0x00140098	0x00E400100002	10743.972595

Notice that since the second row of the array has a timestamp less than the first one, it will not output until the internal timer wraps after 0x7FFFFFFF ticks (time for 0x7FFFFFFF ticks = 2,147,483,647 ticks / 200,000 ticks per second = 10737.418235 seconds).

Output Repetition

Various modes of repetition can be performed on the output arrays. Output without repetition as illustrated above is continuous mode output. The array may be cyclically output for a finite or infinite number of times. Also, any number of transitions can be output repetitively any number of times. Consider the following U16 array:



The first two columns are the timestamp (the least significant word is in the first column), and the third column is the data to be output. If this array is sent to a DIO-128 in continuous cyclic mode and the scan rate is 1 kHz, then the following output will be observed:

At timestamp	this data is output	at time (secs)
0x00000000	0x0003	0
0x00000002	0x0002	0.002
0x00000004	0x0001	0.004
0x00000006	0x0000	0.006
0x00000008	0x0003	0.008
0x00000000	0x0003	0.008
0x00000002	0x0002	0.010
0x00000004	0x0001	0.012
0x00000006	0x0000	0.014
0x00000008	0x0003	0.016
0x00000000	0x0003	0.016
0x00000002	0x0002	0.018
...	...	...

The output would be a 250 Hz square wave on bit 0, and a 125 Hz square wave on bit 1 of the first output port. Note that when the end of the array of output data is reached, output is immediately repeated. When timestamp 8 is reached, it is output and then the clock is immediately reset to zero and timestamp 0 is output. Because the last output data element is immediately followed by the first, they should have the same output value (in this example they both output 0x0003).

If the same array given above is sent to a DIO-128 operating in limited cyclic mode at 1 kHz sampling rate and 3 transitions are specified for 3 repetitions, then the following would be observed on the output:

At timestamp	this data is output	at time (secs)
0x00000000	0x0003	0
0x00000002	0x0002	0.002
0x00000004	0x0001	0.004
0x00000000	0x0003	0.004
0x00000002	0x0002	0.006
0x00000004	0x0001	0.008
0x00000000	0x0003	0.008
0x00000002	0x0002	0.010
0x00000004	0x0001	0.012

4.3 Combined Input and Output Operation

Basic Operation

The combined input and output DSP control program (NEXTIO.BNM) combines the functionality of the input-only and the output-only operations. Performance is reduced by the added complexity of this control program. The DSP must run three concurrent tasks: input, output, and communication with the PC. The input and output tasks run at equal priority, with input being performed before output. If a signal is routed from an output to an input, the inputs will acknowledge an output change on the subsequent scan.

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 to 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. A maximum of four output ports can be configured. The number of output ports is determined by the DIO128 Out Config.vi. As an example, if there are three output ports and two input ports, then ports A, B, and C will be outputs, and ports D and E will be inputs.

Note: Adding output capabilities to an application originally designed for input-only operation will require rewiring the connections to the DIO-128 since the input ports will be remapped.

Programming Steps

Programming applications for combined input and output operation involves running VIs in the following order for *each* DIO-128 board:

1. DIO128 Open.vi - Initialize the board.
2. DIO128 Load.vi - Download the DSP software to the board.
3. DIO128 Out Config.vi - Configure the output parameters for the board.
4. DIO128 Out Status.vi - Get the current status of the board.
5. DIO128 Out Write.vi - Set the outputs to an initial value, or load in the data to be used for cyclic output (see the Output Repetition section on page 22).
6. DIO128 In Start.vi - Start data input *and* output.
7. DIO128 Out Status.vi - Get the current status of the board and find out how much space is left in the board's output buffer.
8. DIO128 Out Write.vi - Send more data to the output buffer.
9. DIO128 In Status.vi - Find out how much data is waiting to be read.
10. DIO128 In Read.vi - Read in the data from the board's buffer.
11. DIO128 Out Stop.vi - Stop outputting data.
12. DIO128 In Stop.vi - Stop acquiring data.
13. DIO128 Close.vi - Close the selected board.

Steps 1-6 and 11-13 should be only run once per board in a program. When outputting in cyclic mode, steps 7 and 8 are frequently omitted. Steps 7-8 and/or 9-10 may be repeated as many times as necessary to read and write the desired data. The order of steps 7-10 is not important, except that the appropriate status must be performed before a read or write operation.

Modes of Operation

Combined input and output operation only supports the simple operating mode as described in the input-only operation section. Start triggering is supported but stop triggering and DAQ clock modulo are not supported. Specify the mode, clocking options, and triggering options in the DIO128 In Start VI. Only the output repetitions and the number of transitions parameters should be specified in the DIO128 Out Config VI. Multiple boards may be used, but they will operate independently. If multiple boards are going to be used, it is recommended that each board be used in either input-only or output-only operation.

	Mode	
Options	Simple	Master/Slave
Basic	X	
DAQ Clock Modulo		
Start Trigger	X	
Stop Trigger		

Data Formats

The data formats are identical to those of the input-only and output-only operations.

5. Software Reference

5.1 Driver VIs

These VIs are arranged in the order in which you will typically execute them, first for input and then for output. For examples on how to use these VIs, open any of the examples provided with this package.

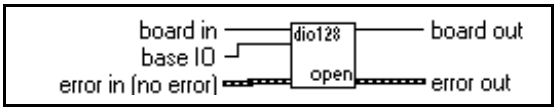
The following four parameters are found in each VI that is described in this section. There are two board number parameters and two error parameters associated with each VI.

- *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 in the DIO128 Open.vi for Windows 95 and is set using the DIO-128 configuration program under Windows NT. All subsequent VIs reference the physical board via this board number.
- *board out* is a flow-through parameter which is the same as board in.
- *error in (no error)* is a standard error cluster input. This cluster includes a status boolean that indicates whether or not an error has occurred previous to this VI, a 32-bit error code that lists the error number that occurred before this VI was called, and a source string which shows which previous VI caused the error.
- *error out* is a standard error cluster output. This cluster includes a status boolean that indicates whether or not an error occurred, a 32-bit error code that lists the error number that occurred, and a source string which shows the VI that caused the error.

DIO128 Open.vi

This VI initializes the selected DIO-128 board.

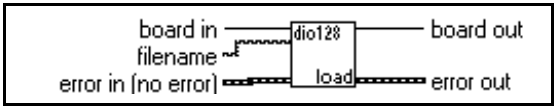
Note: This VI operates differently under Windows NT than it does under Windows 95 (see below).



U16 *base IO* is the base I/O address of the DIO-128 board in hexadecimal format. The default base I/O address is 180 hex. base IO is only used in Windows 95. Windows NT ignores this value and uses the settings set by the DIO-128 configuration program. See the DIO-128 Configuration Program section on page 7 for more information on the Windows NT configuration program.

DIO128 Load.vi

This VI loads the desired DSP code onto the DIO-128 board.

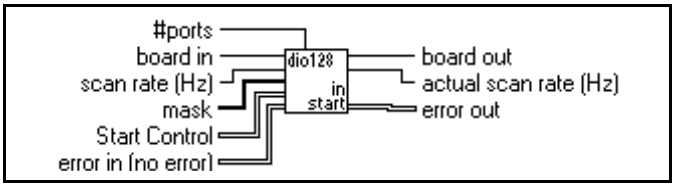


filename is the filename of the DSP code image. The default filename is NEXTIN.BNM. If the file is not found where specified in this control, the system searches for this file in the \WINDOWS and \WINDOWS\SYSTEM directories.

DIO128 In Start.vi

This VI starts the acquisition on the digital I/O board. If you have multiple DIO-128 boards connected via the RTSI connector, start the master board last. See the Modes of Operation section on page 15 for a definition of master board.

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.



U8 *#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. See the Data Transfer Rates section on page 58 for more information. Valid values are between 1 and 8.

DBL *scan rate (Hz)* is the scan rate of the digital board in Hz. The minimum scan rate possible is 244 Hz. The scan rate is converted to an integer divisor of the 16 MHz internal clock.

U16 *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 is a cluster specifying advanced clocking and triggering options. See the Advanced Control section on page 49 for information about these advanced features.

flags

Enum Values:

- simple (0) - single board (no RTSI Trigger 1 connections)
- master/slave (1) - single board or multiple boards (RTSI Trigger 1 connections)

clock control

Enum Values:

- internal (0) - internal clock base
- external (1) - clocked via pin 20 of P2 (rear) connector
- RTSI 0 (2) - clocked via RTSI Trigger 0 line

start trigger

source

Enum Values:

- none (0) - starts immediately
- internal (1) - Future expansion: do not use this option
- external (2) - triggered with signal applied to pin 24 of P2 (rear) connector
- RTSI 2 in (3) - triggered via RTSI Trigger 2

type

Enum Values:

- 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)

sense

Enum Values:

- 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

Note: Stop Trigger is only supported under input-only board mode configured as master-slave.

source

Enum Values:

- none (0) - stops immediately upon DIO128 In Stop execution
- internal (1) - Future expansion: do not use this option
- external (2) - triggered with signal applied to pin 25 of P2 (rear) connector
- RTSI 3 in (3) - triggered via RTSI Trigger 3

type

Enum Values:

- level sensitive (0) - triggering responds to a high or low level (selectable)
- edge sensitive (1) - not currently supported

sense

Enum Values:

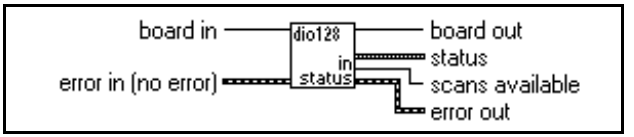
- falling edge (0) - triggering responds to a low in level sensitive mode
- rising edge (1) - triggering responds to a high in level sensitive mode

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.

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 16 MHz internal clock.

DIO128 In Status.vi

This VI checks the input status of the selected digital I/O board.



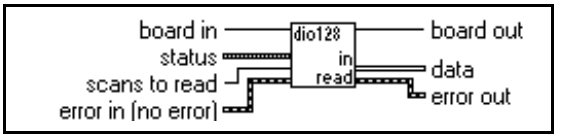
- *status* is a cluster containing information about the current status of the digital card. This cluster needs to be passed to the DIO128 In Read.vi. The parameters of interest are as follows:
  - *Time0* is the low byte of the time elapsed. This is the fifth element of the cluster.
  - *Time1* is the 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.
- *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.

DIO128 In Read.vi

This VI reads transition data from the selected digital I/O board as 16-bit words.



- *status* is a cluster of status information. This data should be input from a previous DIO128 In Status.vi. See DIO128 In Status.vi on the previous page for a description of the status parameters.
- *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 DIO128 In Status.vi.
- *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. See the Input Data Format section on page 17 for a description of the structure of the data.

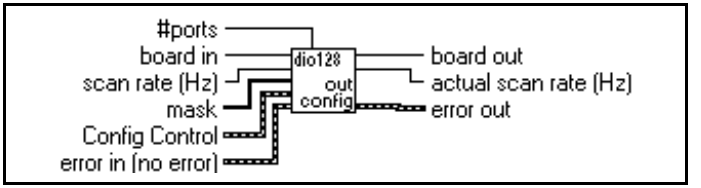
DIO128 In Stop.vi

This VI stops acquisition on the selected digital I/O board. When this VI is executed, all RTSI lines in the DIO-128 System are reset to an input state, clearing all devices from driving any of the RTSI lines.

DIO128 Out Config.vi

DIO128 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.

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.



U8 *#ports* is the number of 16-bit ports that will be processed during the outputting 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. See the Data Transfer Rates section on page 58 for more information. Valid values are between 1 and 4.

DBL *scan rate (Hz)* is the scan rate of the digital board in Hz. The minimum scan rate possible is 244 Hz. The scan rate is converted to an integer divisor of the 16 MHz internal clock.

U16 *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. Below is a two port example — Port A is masked for bits 0-15, Port B for bits 0-14:

Config Control is a cluster specifying advanced clocking and triggering options. See the Advanced Control section on page 49 for information about these advanced features.

Note: In combined Input and Output mode, specify the mode, clocking options, and triggering options in the DIO128 In Start VI. Only the output repetitions and the number of transitions parameters should be specified in this VI.

flags

Enum Values:

- simple (0) - single board (no RTSI Trigger 1 connections)

clock control

Enum Values:

- internal (0) - internal clock base
- external (1) - clocked via pin 20 of P2 (rear) connector
- RTSI 0 (2) - clocked via RTSI Trigger 0 line

start trigger

source

Enum Values:

- none (0) - starts immediately
- internal (1) - Future expansion: do not use this option
- external (2) - triggered with signal applied to pin 24 of P2 (rear) connector
- RTSI 2 in (3) - triggered via RTSI Trigger 2

type

Enum Values:

- 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)

sense

Enum Values:

- 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 is not currently supported for output-only or combined input and output mode.

DAQ clock modulo is not currently supported for output-only or combined input and output mode.

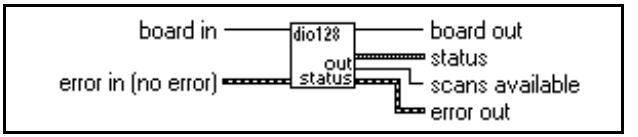
repetitions specifies the number of times a cycle should repeat (see #transitions). If repetitions is set to 0, the cycles will repeat until explicitly stopped. The #transitions field must be set in order for this value to have any effect.

#transitions specifies the number of transitions in the output buffer that constitute a cycle. If this value is 0, the output operation will only track data as it is fed to the buffer.

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.

DIO128 Out Status.vi

This VI checks the output status of the selected digital I/O board.



status is a cluster containing information about the current status of the digital card. This cluster needs to be passed to the DIO128 Out Write.vi. The parameters of interest are as follows:

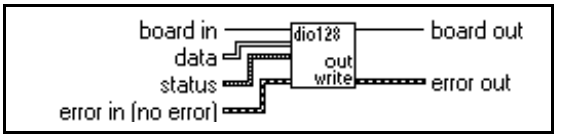
- *Time0* is the low byte of the time elapsed. This is the fifth element of the cluster.
- *Time1* is the high byte of the time elapsed. This is the sixth element of the cluster.
- *portCount* is the number of ports currently being output to. This is the second element of the cluster.

All other variables in this cluster are for internal use only.

scans available is the number of scans that can be written. It is based on the amount of space available in the output buffer.

DIO128 Out Write.vi

DIO128 Out Write fills the output buffer with the data specified.

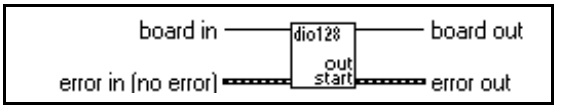


data is the data to be written to the digital I/O board. The data must be passed as a two-dimensional array of U16 numbers. The first column is the low order word of the 32-bit timestamp. The second column is the high order word. See the Output Data Format section on page 21 for a description of the structure of the data.

status is a cluster of status information. This data should be input from a previous DIO128 Out Status vi. See DIO128 Out Status.vi on the previous page for a description of the status parameters.

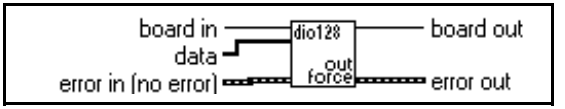
DIO128 Out Start.vi

DIO128 Out Start writes the data in the output buffer to the DIO-128 ports. This VI should only be used for output-only operation. For combined input and output operation, DIO128 In Start.vi will start both input and output.



DIO128 Out Force Output.vi

DIO128 Out Force Output immediately writes the specified data to the output ports. This VI is intended to be used to set initial output conditions.

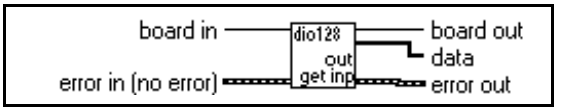


data is a one dimensional array of length *n*. Valid values of *n* are 1 through 4. The data contains no timestamps — it is raw data that will be output to ports A, B, C, and D. Array element one will output to port A, element two to port B, etc. *n* may be larger than the number of output ports specified in DIO128 Out Config.vi for the output-only operation.

DIO128 Out Get Input.vi

DIO128 Out Get Input immediately returns the value on all ports that were valid during the last scan clock.

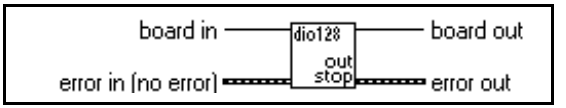
Note: This VI is only used for output-only and combined input and output operation. This VI will generate an error in input-only board mode.



[U16] *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.

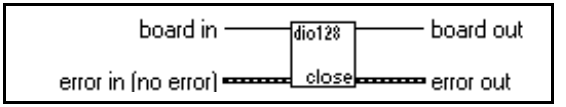
DIO128 Out Stop.vi

DIO128 Out Stop stops an output operation.



DIO128 Close.vi

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



5.2 Examples

These example VIs are found in \DIO128\EXAMPLES\DIOEXMPL.LLB.

Simple One Board Input.vi

This VI demonstrates a common method to configure and read from the DIO-128. 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.

Cont DIO128 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 are monitored. All parameters, except Output Data File, must be set before running the VI for their values to take effect.

Read DIO128 Data from Disk.vi

This VI is a reader for data collected with Cont DIO128 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.

DIO128 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.

Simple Two Board Input.vi

This VI is an example of using multiple DIO-128 boards for monitoring digital signals. Timing is synchronized through the RTSI

bus. Parameters in the sections above the graph must be set before the VI is started for their values to take effect.

Output DIO128 Data.vi

This VI illustrates the DIO-128 output capabilities for "short" duration data. A data set can be created and then output once, continuously, or for a specific number of repetitions.

Output DIO128 Data from Disk.vi

This VI is an example of streaming "long" duration data using the DIO-128 output capabilities. A user-specified data file captured using the Cont DIO128 Input to Disk.vi is replayed.

Simple Output/Input DIO128 Stimulus Test.vi

This VI is an example of performing both input and output on the same DIO-128 board. It allows for the creation of the stimulus data set and for graphing the response.

5.3 Utility VIs

The following VIs are located in the file \DIO128\EXAMPLES\DIOTOOLS.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-128 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-128 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-128 to the AT-DIO-32F.

DIO128 Calculate Clock Divider.vi

If you are clocking a National Instruments DAQ card through RTSI Trigger4, you can use this VI to calculate the clock divider needed. This VI will also report the actual clock frequency used for the DAQ card. See the Multiple Boards section on page 51 for a description of the clock divider.

DIO128 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-128 channel. See also DIO Bit to SCXI Mod,Chan.vi.

Separate Data.vi

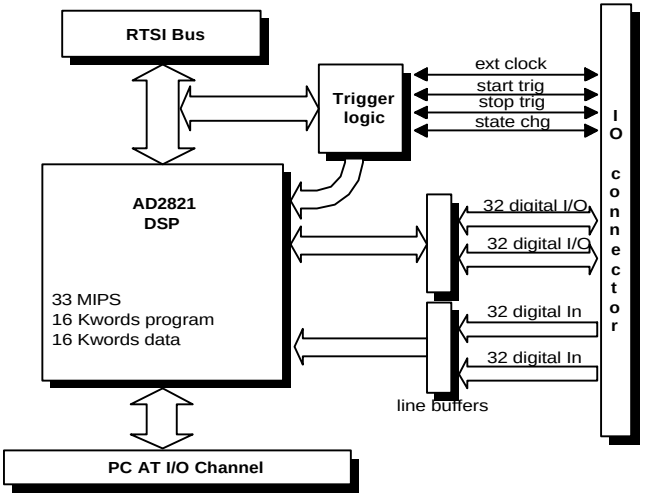
This VI separates the data returned from a DIO-128 read into an array of timestamps and a two-dimensional array of either 16-bit words or boolean bit values.

Word, Mask to Bit Values.vi

The purpose of this VI is to format the data collected with a DIO-128 to an easily readable format. The data collected will be ANDed with the selected channel mask and returned as a boolean array.

6. Hardware Reference

6.1 DIO-128 Block Diagram



6.2 JP15 Multiboard Jumper

The JP15 jumper needs to be installed on exactly one DIO-128 board. If you are using multiple boards, connect this jumper on only one board, preferably the master board (refer to the Multiple DIO-128 Boards section on page 18 to determine which board is the master). This jumper connects a pull-up resistor to the sample negotiation pin (RTSI Trigger1). If you have more than one resistor connected you could damage your DIO-128 boards.

6.3 I/O Connector Pinouts

Below are the pin connectors for the DIO-128 board. The bits are grouped by ports. For example, pin A0 is the connection for port A, bit 0. Refer to page 3 for a diagram of the pin connector locations.

P2

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

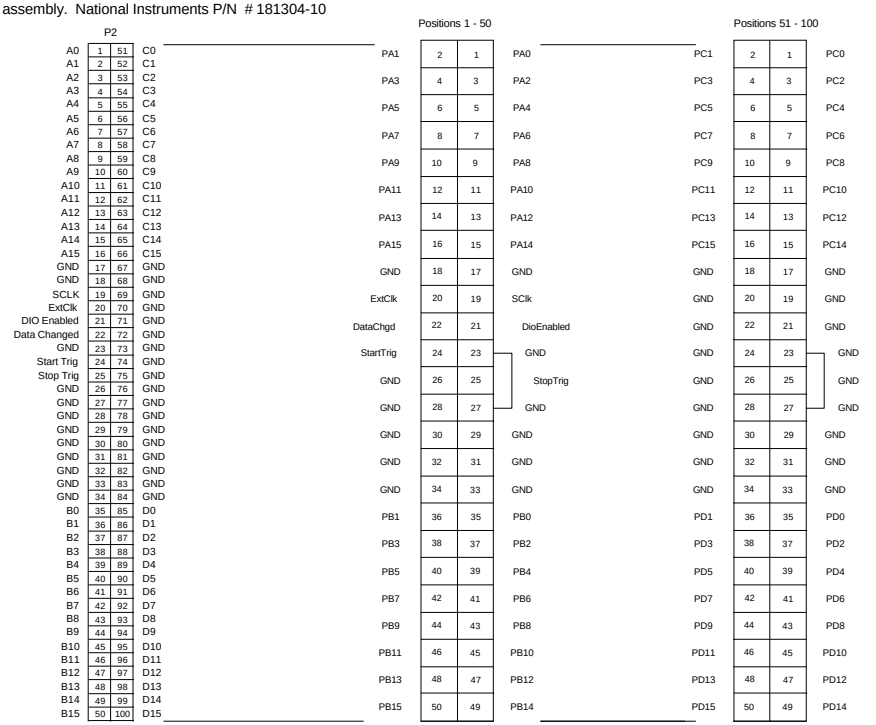
JP4

E0	1	2	E1
E2	3	4	E3
E4	5	6	E5
E6	7	8	E7
E8	9	10	E9
E10	11	12	E11
E12	13	14	E13
E14	15	16	E15
GND	17	18	GND
F0	19	20	F1
F2	21	22	F3
F4	23	24	F5
F6	25	26	F7
F8	27	28	F9
F10	29	30	F11
F12	31	32	F13
F14	33	34	F15

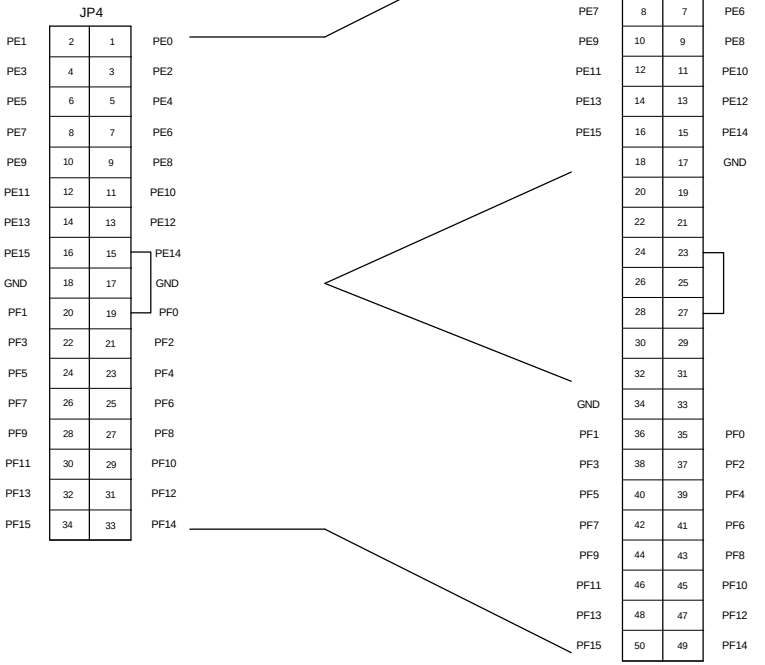
JP3

G0	1	2	G1
G2	3	4	G3
G4	5	6	G5
G6	7	8	G7
G8	9	10	G9
G10	11	12	G11
G12	13	14	G13
G14	15	16	G15
GND	17	18	GND
H0	19	20	H1
H2	21	22	H3
H4	23	24	H5
H6	25	26	H7
H8	27	28	H9
H10	29	30	H11
H12	31	32	H13
H14	33	34	H15

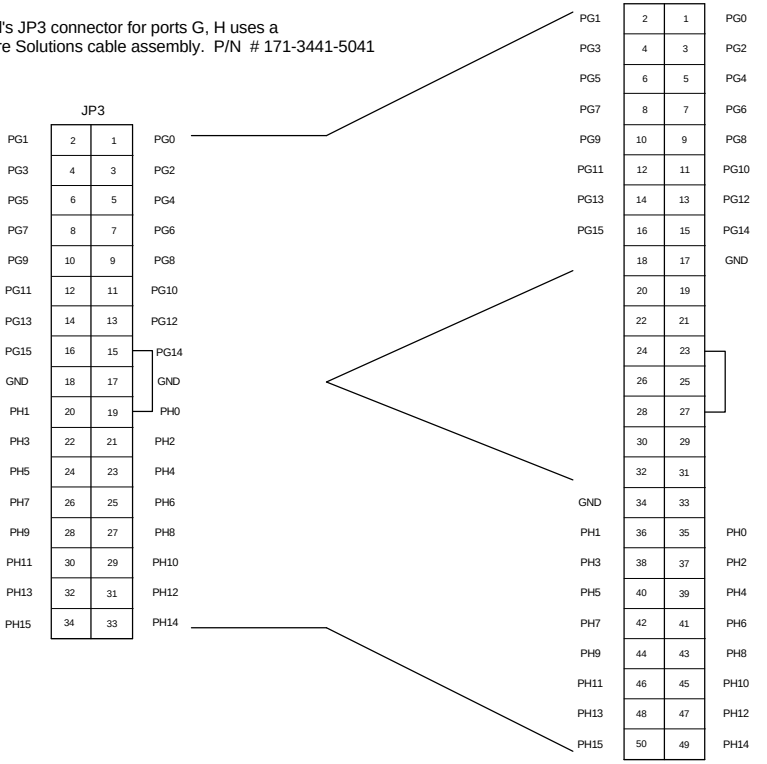
The DIO-128 card's P2 connector for ports A,B,C,D uses a National Instruments NB5 cable assembly. National Instruments P/N # 181304-10



The DIO-128 card's JP4 connector for ports E, F uses a
Viewpoint Software Solutions cable assembly. P/N # 171-3441-5041



The DIO-128 card's JP3 connector for ports G, H uses a Viewpoint Software Solutions cable assembly. P/N # 171-3441-5041



7. Advanced Control: External Clocking/Triggering

The DIO-128 has the ability to synchronize external signals and other data acquisition cards. Physically, these connections are made through the P2 (rear) connector and the RTSI connector. The software is configured via the Start Control cluster in the DIO128 In Start.vi.

Note: If the P2 connector on the DIO-128 board has pins 1-50 connected to an SCXI module, external clocking and triggering will not be available.

7.1 Clocking

Clocking refers to sampling of the digital input lines which can be accomplished internally, via an external signal, or via the RTSI bus.

For the simplest acquisition, set the clock control to internal (0). In this mode, the acquisition starts when the DIO128 In Start.vi is executed and stops when the DIO128 In Stop.vi is executed and samples at the clock rate specified with the scan rate (Hz) control.

Clocking via an External Signal

With external clocking, you can control the sampling of the digital input lines with an external clock source.

To externally clock the DIO-128:

1. Connect the external clock to pin 20 of P2 (rear connector)
2. Set clock control in the Start Control cluster to external (1)
3. Set flags in the Start Control cluster to master/slave (1)

The DIO-128 input lines are sampled whenever a high-to-low transition occurs on the external clock pin.

Clocking via the RTSI Bus

Alternately, you can clock the DIO-128 from the RTSI bus which is useful if you have multiple DIO-128 cards or if you want to drive the DIO-128 from a National Instruments DAQ card.

To clock via the RTSI bus:

1. Set clock control in the in the Start Control cluster to RTSI 0 (2)
2. Set flags in the Start Control cluster to master/slave (1)

Note: The timestamp returned with acquired data is in terms of clock tics. To convert these tics to seconds, you must know the scan rate.

7.2 Triggering

Triggering consists of starting and stopping the acquisition, which can be accomplished automatically, via an external signal, or through the RTSI bus.

For the simplest acquisition, set the start and stop trigger to none (0). In this mode, the acquisition starts when the DIO128 In Start.vi is executed and stops when the DIO128 In Stop.vi is executed.

Triggering via an External Signal

External triggering is accomplished by connecting the trigger source to pin 24 (start) or pin 25 (stop) of the P2 (rear) connector. The acquisition will start/stop when a low level signal is applied to the appropriate pin. As with the other I/O pins, the trigger lines are pulled high with a 4.7KΩ resistor.

To start the acquisition with an external trigger:

1. Connect the start trigger signal to the Start Trig pin 24 of P2
2. Set start trigger in the Start Control cluster to external (2)
3. Set flags in the Start Control cluster to master/slave (1)

To stop acquisition with an external trigger:

1. Connect the stop trigger signal to the Stop Trig pin 25 of P2

2. Set stop trigger in the Start Control cluster to external (2)
3. Set flags in the Start Control cluster to master/slave (1)

Triggering via the RTSI Bus

To start the acquisition via the RTSI bus:

1. Set start trigger in the Start Control cluster to RTSI 2 in (3)
2. Set flags in the Start Control cluster to master/slave (1)

To stop the acquisition via the RTSI bus:

1. Set stop trigger in the Start Control cluster to RTSI 3 in (3)
2. Set flags in the Start Control cluster to master/slave (1)

7.3 Multiple Boards

One of the key features of the DIO-128 is the ability to synchronize multiple DIO-128 boards and/or to synchronize with National Instruments DAQ cards. This synchronization is accomplished through the RTSI connector. It is important when connecting multiple boards to make sure that there is only one board driving the clock or trigger. All other boards should have the clock or trigger flags set to the RTSI connector. Also, the flags control for all boards should be set to master/slave.

If you are driving a National Instruments DAQ card, you can clock that card at a different rate than the DIO-128 by setting the DAQ clock modulo. The modulo is the divisor of the DIO-128 scan clock. For example, if you have the DIO-128 internally clocked at 100 kHz and want to acquire analog data at 1 kHz, set the DAQ clock modulo to 100. When connected in this fashion, the divided clock signal is presented on RTSI Trigger4.

Note: The DIO-128 takes control of the Trigger0 and Trigger1 lines. Do not use these lines for any other functions. Also, if you are using multiple DIO-128 boards, make sure that the JP15 jumper is installed on only one of the boards. If your setup includes a National Instruments board connected through RTSI, you still need to have one DIO-128 with a jumper installed on JP-15.

8. Troubleshooting

8.1 Error Codes

Error #	Location	Explanation
-13	DIO128 Open.vi under Windows NT	Requested Board # has not been configured in the DIO-128 configuration program, or computer was not rebooted after changing settings in the DIO-128 configuration program.
-12	All VIs	Invalid parameter
-11	All VIs	Bad handshake between DIO-128 and host computer.
-10	DIO In Status, DIO Out Status	Bad status block, DSP code corrupted, running too fast for the current mode
-9	All VIs except DIO128 Open.vi	Requested Board # has not been opened
-8	All VIs	Illegal Board # – Board # must be between 0 and 7
-7	DIO Load	DSP code doesn't handshake, error during load, bad .BNM file, board not at I/O location
-6	DIO Load	.BNM incorrectly formed
-5	DIO Load	.BNM incorrectly formed
-4	DIO Load	.BNM incorrectly formed
-3	DIO Load	.BNM incorrectly formed
-2	DIO Load	.BNM incorrectly formed
-1	DIO Load	.BNM file not found

8.2 Diagnostics

Problem: Whenever I open a DIO-128 VI, I am prompted to find the file DIO128.DLL.

Solution: The DLL was not installed in your WINDOWS\SYSTEM directory. There is a copy of the DLL in the \DIO128 directory. Copy this file to your WINDOWS\SYSTEM directory.

Problem: I get one of the errors listed that reports that the .BNM is incorrectly formed (errors -2 to -6).

Solution: The BNM file has become corrupt. Please reinstall the software from floppy. If there is still a problem, please contact Viewpoint Software (See Contacting Us on page 60).

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: You are using either the 16-bit version of LabVIEW 4.0 or LabVIEW 3.1.1 or earlier and are trying to load a 32-bit version of a DIO-128 VI. Either upgrade your software to LabVIEW 4.0 and ensure that the 32-bit LabVIEW is loaded or contact Viewpoint Software to obtain the 16-bit version of the DIO-128 software. Note that the 16-bit version of the software does not support output functionality.

Problem: Upon loading a VI with DIO-128 sub VIs, I get prompted to select a library but it looks like the system is searching for DIO128.DLL. If I point the system to DIO128.DLL, it just prompts me again. What is wrong?

Solution: You are using the 32-bit version of LabVIEW 4.0 and are trying to load a 16-bit version of a DIO-128 VI. Delete the LLB that contains that VI and reinstall the software. Use the LLB that is in the directory that you specified when installing the DIO-128 files.

Problem: When running the DIO128 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-128 and the analog input board being used. Ensure that the connector is plugged in all the way.

Problem: When running the Simple Two Board Input.vi, nothing happens when signals are applied to the digital inputs and the time scale does not change.

Solution: Verify that the RTSI connector is attached to the DIO-128 boards being used. Ensure that the connector is plugged in all the way.

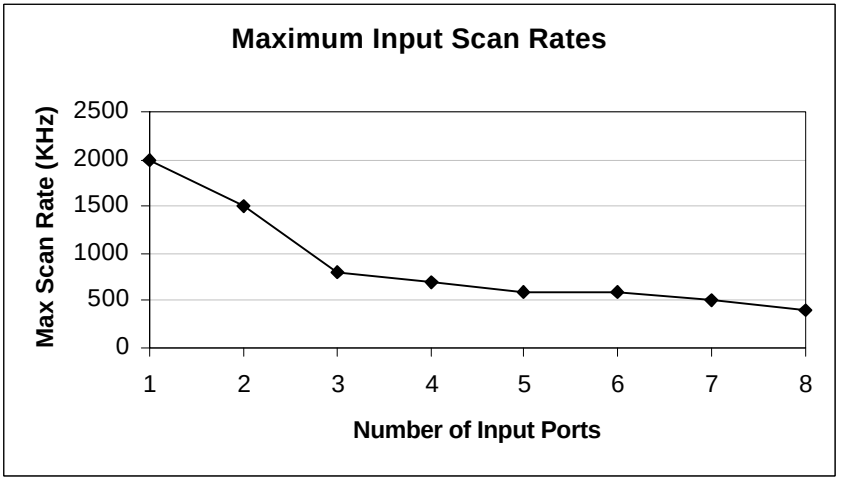
Problem: When running the DIO128 In Start VI or the DIO128 Out Config VI, I get error -12, invalid parameter. What caused this error?

Solution: Most likely, a parameter in the Start Control or Config Control is not supported. Refer to the Driver VIs section beginning on page 27 to verify the options used are supported by the type of operation being performed.

9. Technical Specifications

9.1 Board Sampling Rates

Input-Only Operation

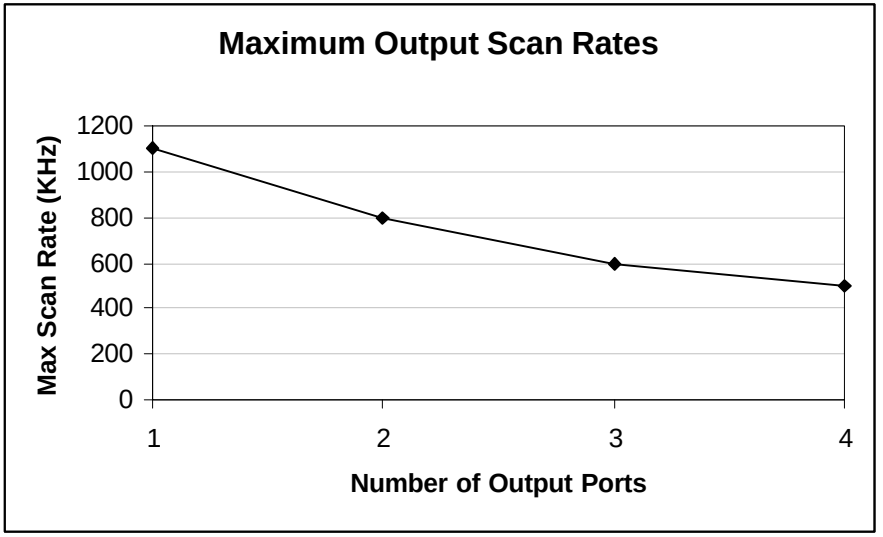


The graph above shows the sampling rates possible in input-only operation for different hardware setups using one DIO-128 board. If the DIO-128 is run at rates higher than these, either an error (usually -11) will result, or the internal clock will create erroneous timestamps (the error can be as much as one clock tick per transition, with the error summing over all transitions).

The time to transfer the scans to the PC should also be taken into account when selecting a scan rate. See the graphs shown in the Data Transfer Rates section on page 57.

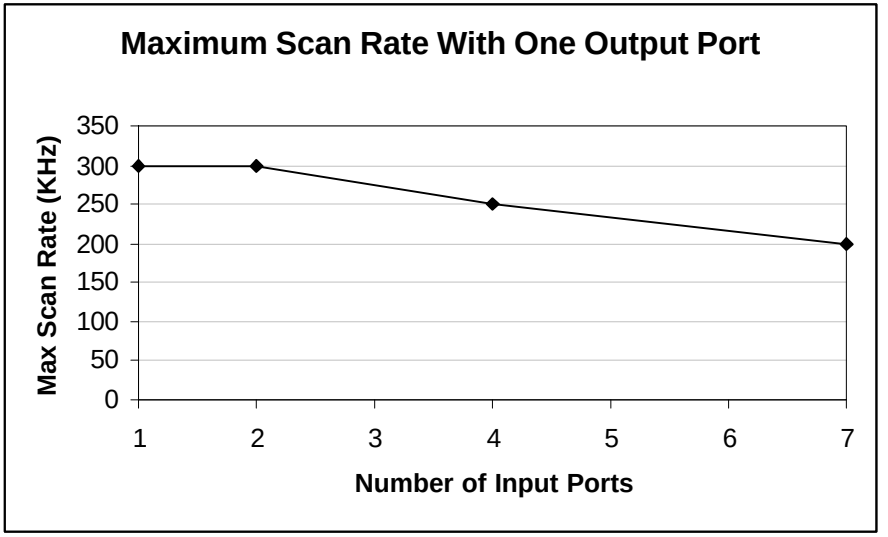
In the above graph, a single board is running in simple mode with a continuous 10 kHz input signal (20,000 transitions per second). Running the board in Master/Slave mode, with start or stop trigger, or DAQ clock modulo will decrease the maximum scan rate by as much as 50%.

Output-Only Operation



The above graph was obtained while running the DIO-128 in cyclic mode output-only operation. The maximum output frequency at 50% duty cycle is half the maximum scan rate since two transitions are needed to produce one square wave pulse. Running at faster rates is possible, but errors in transition timing will result. If the board is run at a faster rate, a transition may be off by up to one timer tick, but these timing errors are summed across all transitions. So, as an example, if a four port output is set to produce 100,000 transitions and is sampling at 1 MHz, then the last transition may be off by as much as 100,000 ticks (0.1 second).

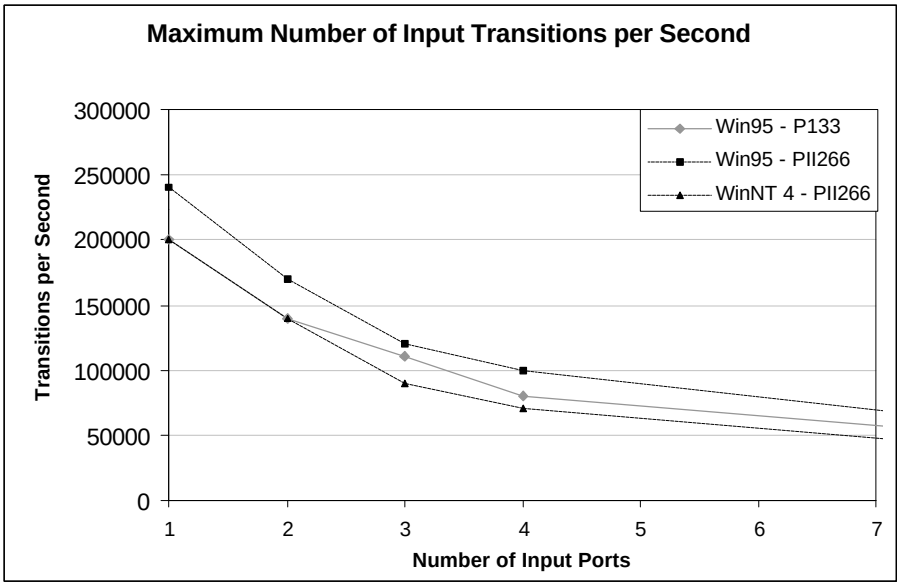
Combined Input and Output Operation



Data for the above graph was obtained by running the DIO-128 in combined input and output operation. A single output port running in cyclic mode was used. Increasing the number of output ports (maximum of 4) will only slightly decrease the maximum scan rates. See the sections on input-only and output-only operation for a description of the symptoms typically observed when the DIO-128 is run at too high of a scan rate.

9.2 Data Transfer Rates

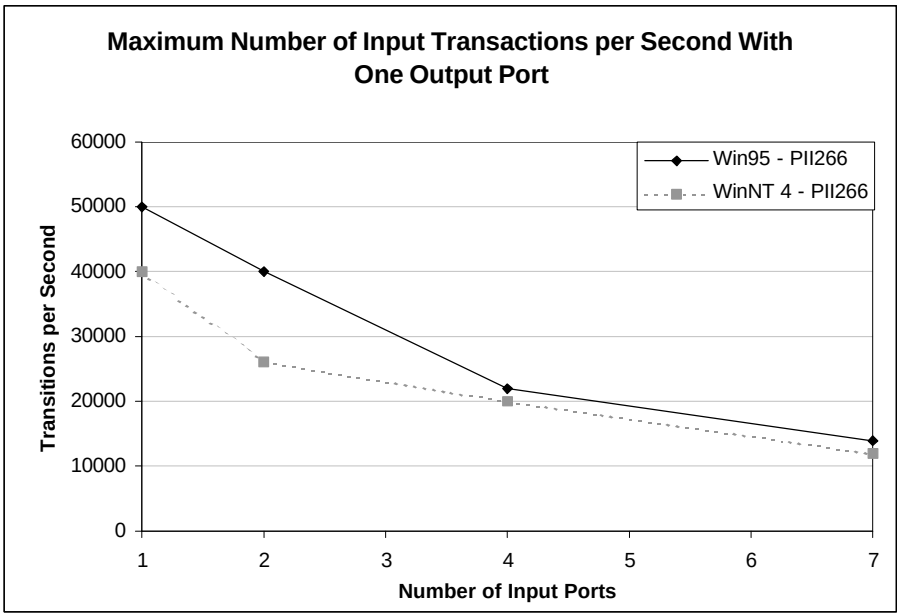
Input-Only Operation



The above graph shows the number of scans per second that were transferred from the buffer on the DIO-128 board to LabVIEW. These transfer rates were observed on a 133 MHz Pentium machine with 32 MB of RAM under Windows 95, and a 266 MHz Pentium II machine with 64 MB of RAM under Windows 95 and Windows NT 4.0.

If the DIO-128 is run at very high scan rates, then it will spend a high percentage of its time sampling the inputs and will not be able to transfer data to the PC at as high a rate. When the DIO-128 is running at its maximum recommended scan rate, then the maximum number of transitions per second will typically decrease by 10-30%.

Combined Input and Output Operation



The above graph shows the maximum number of transitions per second that can be input while simultaneously performing cyclic output on one channel. These transfer rates were observed on a 266 MHz Pentium II machine with 64 MB of RAM under Windows 95 and Windows NT 4.0.

The rates observed are not possible if running near the maximum scan rate. Also note that only a single output port was used to acquire data for the above graph. Increasing the number of output ports used (maximum of 4) will not significantly change the transfer rates graphed above.

9.3 Board Specifications

Power Requirements: +5V

Operating Temperature Range: 0 to 60 degrees C

Storage Temperature Range: -40 to 100 degrees C

Humidity: 0 to 90%, non-condensing

Power Consumption: 2A @ +5V

Max. I/O Line Current: Source: -15mA
Sink: 24mA

10. Contacting Us

Viewpoint Software Solutions, Inc.
2320 Brighton Townline Road
Rochester, NY 14623

voice: 716-475-9555
fax: 716-475-9645
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.