

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Visit our website - *Click HERE*

ComputerBoards, Inc.

LabVIEW Extension to Universal *Library*™ Programmers Manual

LabVIEW Extensions Rev 3.3 March, 1996

for use with

Universal Library
Revision 3.3 and above

&

LabVIEW Rev 3.0 and above

ComputerBoards, Inc.
125 High Street, #6, Mansfield, MA 02048
(508) 261-1123 FAX 261-1094

(C) Copyright 1996 ComputerBoards, Inc. NO PART OF THIS MANUAL MAY BE REPRODUCED IN ANY FORM WITHOUT WRITTEN PERMISSION FROM COMPUTERBOARDS, INC.

NOTICE OF COPYRIGHT AND TRADEMARKS 1
Introduction 2
Installation and Configuration 3
Installation 3
CB.CFG File 3
Overview of the Universal Library LabVIEW Extensions 4
Analog I/O VI's: 4
Digital I/O VI's: 5
Thermocouple Input VI's: 5
Counter VI's: 6
Streamer File VI's: 6
Misc. VI's: 7
Memory Board VI's: 7
Revision Control VI's 8
How to Use the LabVIEW Extensions (VI's) 9
LabVIEW Extension Program Development: 9
Contexts: 9
Example VI's: 10
Universal Library Virtual Instruments (VI's) 11
DeclRev.VI New R3.3 ID 11
GetRev.VI New R3.3 ID 12
ACvtData.VI Changed R3.3 MOD 13
ACnvPrDt.VI Changed R3.3 MOD 14
ACalData.VI New R3.3 15
AIn.VI 16
AInScBg.VI Changed R3.3 ID 17
AInScFg.VI Changed R3.3 ID 19
ALoadQue.VI 21
AOut.VI 22
AOutScFg.VI 23
AOutScBg.VI 24
APretrBg.VI 26
APretrFg.VI 28
ATrig.VI 30

COUNTERS - AN INTRODUCTION 32
C8254Cfg.VI 33
C8536Cfg.VI 34
C8536Ini.VI 35
C9513Cfg.VI 36
C9513Ini.VI 38
CFreqIn.VI 40
CIn.VI 41
CLoad.VI 42
CStore.VI 43
DBitIn.VI 44
DBitOut.VI 45
InByte.VI/ InWord.VI 46
OutByte.VI / OutWord.VI 47
DCfgPrt.VI 48
DIn.VI 49
DInScBg.VI 50
DInScFg.VI 52
DOut.VI 53
DOutScBg.VI 54
DOutScFg.VI 55
ErrHdlng.VI 56
ErrMsg.VI 57
FileAInS.VI 58
FileInfo.VI 60
FilePret.VI 61
FromEng.VI 63
ToEng.VI 63
GetBoard.VI 64
GetCfg.VI 65
GetStat.VI 69
MemRdPrt.VI 70
MemRead.VI 71
MemReset.VI 72

MemSetDT.VI 73
MemWrite.VI 74
OptAIn.VI Changed R3.3 ID 75
SelChan.VI 77
ScaleArr.VI 78
ScalePnt.VI 79
SetCfg.VI 80
StopBg.VI 82
TIn.VI Changed R3.3 ID 83
TInScan.VI Changed R3.3 ID 84
Using the File VI's 86
Overview 86
Hard Disk vs Ram Disk Files 86
Maximum Sampling Speed 86
How To Determine Maximum Sampling Speed 87
Speeding Up Disk Files (De-Fragmenting): 87
What is a Ram Disk? 87
Installing a RAM Disk: 87
Using The Ram Disk 88
Board Specific Information 89
CIO-, PC104- and PCM- CTR Series Boards Limitations 89
CIO-DAC Series Boards Limitations 89
CIO- and PC104- DAS08 Series Boards 89
Analog Input VI's: 89
Maximum Sampling Rate : 89
Options: 89
Wiring Requirements : 90
Analog Output VI's: 90
Digital I/O VI's: 90
Counter I/O VI's: 90
CIO-DAS08/Jr Board 90
CIO-DAS800 Series Boards 90
CIO- and PC104- DAS16 Series Boards 91
Analog Input VI's: 91

A/D Range: 91
Maximum Sampling Rate: 92
Analog Output VI's: 92
Digital I/O VI's: 92
Counter I/O VI's: 92
CIO-DAS48/PGA Board 92
Analog Input VI's: 92
A/D Ranges: 92
Analog Output VI's: 92
Digital I/O VI's: 93
Counter I/O VI's: 93
CIO-DAS1400 & CIO-DAS1600 Series Boards 93
Analog Input Vi's: 93
CIO-DAS1402, CIO-DAS1602, CIO-DAS1602/16 Ranges 93
Maximum Sampling Rate: 93
Options 93
Analog Output: 93
Digital I/O VI's: 93
Counter I/O VI's: 94
CIO- and PC104- DIxx Series Boards 94
CIO- and PC104- DIO Series Boards 94
CIO-, PC104- and PCM- DIO24/CTR3 94
CIO- and PC104- DOxx and CIO-DOxxDD Series Boards 94
CIO-DDA06 95
CIO-DISO48 95
CIO-EXP Series Boards 95
CIO-INT32 95
CIO- ans PC104- PDISO8 and CIO-PDISO16 95
CIO-PDMA16 95
CIO-RELAY16 95
PPIO-DIO24H 95
PPIO-AI8 95
PPIO-CTR06 Boards 96
cSBX-DDA04 Board 96

PCM-DAC02 Board 96
PCM-DAS08 Board 96
PCM-DAS16x/xx Series Boards 96
Appendix A. Error Codes 98

NOTICE OF COPYRIGHT AND TRADEMARKS

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form by any means, electronic, mechanical, by photocopying, recording or otherwise without the prior written permission of ComputerBoards, Inc.

MEGA-FIFO, Universal*Library*, *Insta*Cal, *Harsh Environment Warranty* the CIO-, PCM-, and PPIO- part number prefix and **Computer**Boards are registered trademarks of ComputerBoards, Inc.

DT-Connect is a trademark of Data Translation, Inc.

LabVIEW is trademark of National Instruments.

IBM, PC and PC/AT are trademarks of International Business Machines Corp.

Information furnished by ComputerBoards, Inc. is believed to be accurate and reliable; however, no responsibility is assumed by ComputerBoards, Inc. for its use; nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or copyrights of ComputerBoards, Inc.

Each original copy of Universal Library LabVIEW Extension is licensed for use on one CPU at a time. It is theft to make copies of this program for simultaneous use. There is nothing but your conscience to protect ComputerBoards' intellectual property in this product once it is in your possession.

IMPORTANT REVISION NOTES

If any Universal Library functions have changed due to a revision of the software, the heading for that VI will have the note Changed Rx.x Code. The word 'Changed' indicates just that. The Rx.x is the revision number when the change was made and the code will be MOD, meaning the VI in your program must be modified to include that option before you can use that VI with that or later revisions of the library, or, the code will be ID, meaning you will only have to modify your code If Desired to use that VI with the additions and changes.

To simplify management of new features and to prevent the forced rewrite of older code, there is a new VI that declares the revision number your program was written for. The revision declaration will allow old code to run with new revisions even if substantial changes have been made to VI's. Please read more about this under DeclRev.VI.

Introduction

The Computer Boards Universal Library LabVIEW Extensions are the labview virtual instruments that you need to construct your own programs in LabVIEW that use any of the Computer Boards data acquisition and control boards. This is the manual for the LabVIEW extensions to the Universal Library. Although the LabVIEW extensions follow very closely the syntax of the UniversalLibrary, there are some small differences. For LabVIEW syntax, please use this manual. If you decide to use the Universal Library with another language, please use the Universal Library programming manual, not this one.

The ComputerBoards products supported by Revision 3.3 of the Universal Library LabVIEW Extensions are:

CIO-CTR05 & 10	CIO-DAS16/Jr.	CIO-DO24DD, 48DD	PC104-DAS16Jr/12&16
CIO-CTR10HD & 20HD	CIO-DAS16/M1	CIO-DIO48H	PC104-DAS08/12
CIO-DAC02	CIO-DAS1601/12	CIO-DUAL-AC5	PC104-DIO48
CIO-DAC08 & 8i	CIO-DAS1602/12	CIO-DISO48	PC104-DI48,DO48
CIO-DAC16 & 16i	CIO-DAS1600/16	CIO-EXP16, EXP32	PC104-PDISO8
CIO-DAS08	CIO-DAS16Jr/CTR	CIO-EXP-GP	MEGA-FIFO
CIO-DAS08/Jr & /Jr-AO	CIO-DAS48-PGA	CIO-EXP-RTD/16	PCM-COM485
CIO-DAS08-AOH, L, M	CIO-DDA06/12 & 16	CIO-EXP-BRIDGE/16	PCM-D24/C3
CIO-DAS08-PGH, L, M	CIO-DDA06/Jr	CIO-INT32	PCM-DAC02
CIO-DAS800, 801, 802	CIO-DI48, 96, 192	CIO-PDISO8, PDISO16	PCM-DAS08
CIO-DAS1401/12	CIO-DIO24	CIO-PDMA16	PCM-DAS16x/12, 16
CIO-DAS1402/12 & 16	CIO-DIO24/CTR3	CIO-RELAY08, 16	PCM-DAS16/330
CIO-DAS16 & 16F	CIO-DIO24H	CIO-SSH16	PPIO-AI08
CIO-DAS16/330	CIO-DIO48, 96, 192	PC104-CTR10HD	PPIO-CTR06
CIO-DAS16/330i	CIO-DO48H,96H, 192H	PC104-DAC06/12	PPIO-DIO24H

As new boards are designed, they are added to the Universal Library. Please call if you do not see your board listed here.

Installation and Configuration

This chapter describes how to install the software on your computer and how to configure the software for the boards that you will be using with it.

Installation

Please refer to the Installation and Configuration section of the Universal Library Programmers Manual for detailed installation instructions.

CB.CFG File

The LabVIEW VI's are dependent on the CB.CFG file in the same way as the Universal Library functions are. Please refer to the Installation and Configuration section of the Universal Library Programmers Manual for an explanation of the CFG file and its interaction with the Universal Library(LabVIEW Extensions).

Overview of the Universal Library LabVIEW Extensions

The Universal Library LabVIEW Extensions consist of a set of low level VI's that you may wire together to form your application. These VI's are grouped according to their purpose. All of the groups except for Misc. are based on which type of devices they are used with.

Analog I/O VI's:

The Analog VI names all begin with "A". These VI's perform analog input and output and convert analog data.

AIn.VI - Single Analog input
Takes a single reading from an analog input channel (A/D).

AInScBg.VI- Background Analog Input Scan
Repeatedly scans a range of analog input (A/D) channels in the background. The channel range, the number of iterations, the sampling rate, and the A/D range can all be specified. The data that is collected is stored in an array.

AInScFg.VI- Foreground Analog Input Scan
Repeatedly scans a range of analog input (A/D) channels in the foreground. The channel range, the number of iterations, the sampling rate, and the A/D range can all be specified. The data that is collected is stored in an array.

ALoadQue.VI - Load Chan/Gain queue
Loads a series of chan/gain pairs into A/D board's queue. These chan/gains will be used with all subsequent analog input VI's.

AOut.VI - Single analog output
Outputs a single value to an analog output (D/A).

AOutScBg.VI - Background Analog output scan
Repeatedly scans a range of analog output (D/A) channels in the background. The channel range, the number of iterations, and the rate can all be specified. The data values from consecutive elements of an array are sent to each D/A channel in the scan.

AOutScFg.VI - Foreground Analog output scan
Repeatedly scans a range of analog output (D/A) channels in the foreground. The channel range, the number of iterations, and the rate can all be specified. The data values from consecutive elements of an array are sent to each D/A channel in the scan.

APretrBg.VI- Analog pre-triggered input in the Background
Repeatedly scans a range of analog input (A/D) channels in the background while waiting for a trigger signal. When a trigger occurs it returns the specified number of samples and points before the trigger occurred. The channel range, the sampling rate, and the A/D range can all be specified. All of the data that is collected is stored in an array.

APretrFg.VI- Analog pre-triggered input in the foreground
Repeatedly scans a range of analog input (A/D) channels in the foreground while waiting for a trigger signal. When a trigger occurs it returns the specified number of samples and points before the trigger occurred. The channel range, the sampling rate, and the A/D range can all be specified. All of the data that is collected is stored in an array.

ATrig.VI - Analog trigger
Reads the analog input and waits until it goes above or below a specified threshold. When the trigger condition is met the current sample is returned.

ACvtData.VI - Converts analog data
Each raw sample from analog input is a 16 bit value. On 12 bit A/D boards it consists of a 12 bit A/D value along with a four bit channel number. On 16 bit A/D boards it contains the 16 bit A/D value.

This conversion is done automatically by the AIn VI. It can also be done automatically by the AInScxx VI with the CONVERTDATA option. In some cases though, it may be useful or necessary to collect the data and then do the conversion sometime later. The ACvtData.VI takes a buffer full of unconverted data and converts it.

ACalData.VI - Calibrates analog data
Each raw sample from a board with software calibration factors which must be applied to the sample may be acquired and calibrated then passed to an array, or acquired then passed to the array without calibration. This VI applies the calibration factors to an array of data after the acquistion is complete. The only case where you would withold calibration until after the acquistion run was complete is on slower CPUs or when processing time is at a premium. Applying calibration factors in real time on a per sample basis does eat up machine cycles.

ACnvPrDt.VI - Convert pre-trigger data
When data is collected with the APretrxx.VI, the same conversion needs to be done as described above for ACvtData. There is a further complication though.

APretrxx.VI collects analog data into an array. It treats the array like a circular buffer. While it is waiting for the trigger to occur it fills the array. When it gets to the end it resets to the start and begins again. When the trigger signal occurs it continues collecting data into the circular buffer until the requested number of samples have been collected.

When the data acquisition is complete all of the data is in the array but it is in the wrong order. The first element of the array does not contain the first data point. The data has to be rotated in the correct order.

This conversion can be done automatically by the APretrxx VI with the CONVERTDATA option. In some cases though, it may be useful or necessary to collect the data and then do the conversion sometime later. The ACnvPrDt.VI takes a buffer full of unconverted data and converts it.

Digital I/O VI's:

The digital VI names all begin with "D". These VI's perform digital input and output. They operate on all types of digital I/O ports.

DBitIn.VI - Digital bit input
Reads a single bit from a digital input port.

DBitOut.VI - Digital bit output
Sets a single bit on a digital output port.

DCfgPort.VI - Configures digital outputs
Selects whether a digital port is an input or an output.

DIn.VI - Digital byte input
Reads a specified digital input port.

DInScBg - Digital multiple byte input in the background.
Reads a specified number of bytes from a digital input port at a specified rate.

DInScFg - Digital multiple byte input in the foreground.
Reads a specified number of bytes from a digital input port at a specified rate.

DOut.VI - Digital byte output
Writes a byte to a digital output port.

DOutScBg.VI - Digital multiple byte output in the background.
Writes a series of bytes to a digital output port at a specified rate.

DOutScFg.VI - Digital multiple byte output in the foreground.
Writes a series of bytes to a digital output port at a specified rate.

Thermocouple Input VI's:

The thermocouple VI names begin with "T". These VI's convert a raw analog input from an EXP board to temperature.

TIn.VI - Single thermocouple input
Reads an EXP channel, filters it, does the cold junction compensation, linearization and converts it to temperature.

TInScan.VI - Scan a range of thermocouple inputs
Reads the temperature from a range of EXP channels as described above. Returns the temperature values to an array.

Counter VI's:

The counter VI names begin with "C". These VI's load, read and configure counters. There are three types of counter chips used in Computer Board products - 8254s, 8536s and 9513's. Some of the counter commands only apply to one type of counter.

C8254Cfg.VI - Configures 8254 counter
Selects the basic operating mode of an 8254 counter.

C8536Cfg.VI - Sets operating mode of 8536 counter.
This VI sets all of the programmable options that are associated with an 8536 counter chip.

C8536Ini.VI - Initializes 8536 counter
Initializes and selects all of the chip level features for a 8536 counter board. The options that are set by this command are associated with each counter chip, not the individual counters within it.

C9513Cfg.VI - Sets operating mode of 9513 counter.
This VI sets all of the programmable options that are associated with a 9513 counter chip. It is similar in purpose to C8254Cfg.VI except that it is used with a 9513 counter.

C9513Ini.VI - Initializes 9513 counter
Initializes and selects all of the chip level features for a 9513 counter board. The options that are set by this command are associated with the each counter chip, not the individual counters within it.

CFreqIn.VI - Measures frequency of a signal
This VI measures the frequency of a signal by counting it for a specified period of time (GatingInterval) and then converting the count to count/sec (Hz). It only works with 9513 counters.

CIn.VI - Reads a counter.
Reads a counters current values.

CLoad.VI - Load a counter.
Loads a counter with an initial count value.

CStore.VI - Store counter value when interrupt occurs.
Installs an interrupt handler that will store the current count whenever an interrupt occurs. This VI only works with 9513 counters.

Streamer File VI's:

The file VI names begin with "F". These VI's create, fill, and read "streamer" files. These VI's also let you collect huge amounts of analog input data. The amount of data is limited only by your available disk space.

FileAInS.VI - Transfer analog input data directly to file.
Very similar to AInScxx.VI except that the data is stored in a file instead of an array.

FilePret.VI - Pre-triggered analog input to a file.
Very similar to APretrxx.VI except that the data is stored in a file instead of an array.

FileInfo.VI - Reads "streamer" file information.
Each streamer file contains information about how much data is in the file and the conditions under which it was collected (sampling rate, channels, etc.). This VI reads that information.

FileRead.VI - Reads data from "streamer" file.
Reads a selected number of data points from a streamer file into an array.

Misc. VI's:

These VI's are the odds and ends. They have to do with error handling and managing background operations.

ErrHdlng.VI - Selects type of error handling.
The universal library has a number of different methods of handling errors. This VI selects which of these methods will be used with all subsequent library calls. The options include stopping the program when an error occurs and printing error messages.

ErrMsg.VI - Returns an error message for a given error.
All library VI's return error codes. This VI converts an error code to an error message.

GetStat.VI - Returns status of background operation.
Once a background operation is started your program will need to periodically check on its progress. This VI returns the current status of the process.

StopBg.VI - Stop a background process.
It is sometimes necessary to stop a background process in the case of an error or if the process has been set up to run continuously forever. This VI will stop a background process that is running.

OptAIn.VI - Analog In Option Generator
The AInScxx.VI's have an input called options which should be wired to this VI's output. It generates a value based on the ANDED values of its inputs.

ScaleArr.VI - Converts raw data in an array to engineering data in an array.
ScalePnt.VI - Converts a raw data point to engineering units.

Memory Board VI's:

The memory board VI's all begin with "M". These VI's read/write and control memory boards (MEGA-FIFO).

The most common use for the memory boards is to store large amounts of data from an A/D board via a DT-Connect cable between the two boards. To do this, you should use the EXTMEMORY option with AInScxx.VI or APretrxx.VI.

Once the data has been transferred to the memory board you can use the memory VI's to retrieve the data.

MemSetDT.VI - Set DT-Connect Mode on Memory Board
The memory boards have a DT-Connect interface which can be used to transfer data through a cable between two boards rather than through the PC's system memory. The DT-Connect port on the memory board can be configured as either an input (from an A/D) or as an output (to a D/A). This VI configures the port.

MemReset.VI - Resets the Memory Board Address
The memory board is organized as a sequential device. When data is transferred to the memory board it is automatically put in the next address location. This VI resets the current address to the location 0.

MemRead.VI - Read Data From Memory Board
Reads a specified number of points from a memory board starting at a specified address.

MemWrite.VI - Writes Data To The Memory Board
Writes a specified number of points to a memory board starting at a specified address.

MemRdPrt.VI - Reads Data Collected With APretrxx.VI
The APretrxx.VI writes the pre-triggered data to the memory board in a scrambled order. This VI unscrambles the data and returns it in the correct order.

Revision Control VI's

As new revisions of the library are released, bugs from previous revisions are fixed and occasionally new functions are added. It is ComputerBoards goal to preserve existing programs you have written and therefore to never change the order or number of arguments in a function. Sometimes it is not possible to achieve this goal as evidenced in the changes from revision 3.2 to 3.3. In revision 3.3 we added support for multiple background tasks, a feature customers had requested for some time. Allowing multiple background tasks meant adding the argument BoardNum to several functions. Doing so would have meant that programs written for version 3.2 would not run with 3.3 if they called those functions. If not for the new DeclRev VI, the programs would have to be rewritten in each instance where the effected VI's are used.

The revision control VI initializes the DLL so that the functions are interpreted according to the format of the revision you wrote your program in. This VI is new in revision 3.3 so to take advantage of it, the VI must be added to your program. Please see the full explanation of the VI.

DeclRev.VI - Declares the revison # of the Universal Library that your program VI's are designed in.

GetRev.VI - Polls the installed Universal Library version number and returns it.

The Universal Library LabVIEW Extensions are a complete set of low level VI's that will allow you to construct your own programs that use any of the ComputerBoards data acquisition and control boards listed in the table on page two of this manual. This manual assumes that you have a working knowledge of LabVIEW and is not intended to instruct the user on how to develop LabVIEW programs. Only very limited support is available on programming in this environment. For detailed information on how to create, edit, and execute virtual instruments, refer to the *LabVIEW for Windows User Manual*.

NOTE:
If you havent read the section regarding REVISION CONTROL VI's on the previous page, please do so now.

LabVIEW Extension Program Development:

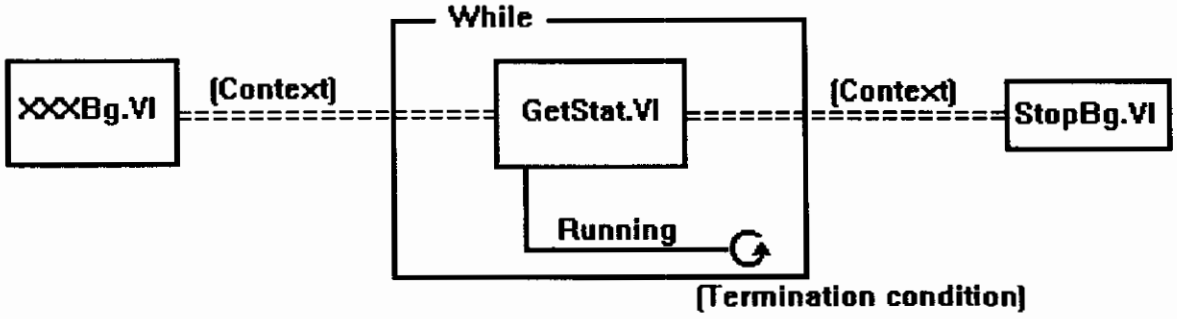
Many of the low level VI's take a choice of several values as an input (analog voltage range, digital port specifier, etc.). String tables are provided rather than numeric values for these inputs. To make use of these string menus, the following procedure is recommended.

1) Open a new VI.
2) Include the low level ComputerBoards VI by choosing DAS16.LLB and selecting an appropriate VI for your application. Its front panel will appear.
3) Select the desired string menu control.
4) CTRL-Drag it to the new VI's front panel or copy(^C) and paste(^V) it.
5) Change to the new VI's diagram panel and wire the string menu control to the corresponding VI's input connector.

NOTE:
Menu inputs are mapped to values used by the Universal Library. Therefore, you should never pass a known numeric value to the VI directly or change the order or contents of the menus.

Contexts:

There are two distinct VI's for every UL function featuring background operation. One is for foreground operation only and the other is for background operation only. The last two letters of their names are "Fg" and "Bg" for foreground and background respectively. Their parameter lists differ in that background VI's have a Context output that must be wired to subsequent VI's(GetStat.Vi and StopBg.VI). Context is an output data structure that contains information such as the board number, the data array, the size of the data array, the initial status of the operation, and the error code. Connecting a Probe to the Context wire will display the elements in the data structure and allow you to check their intermediate values if desired. In general, the background VI's should conform to the wiring pattern shown below. There are several example programs that effectively demonstrate the correct wiring and use of Contexts. Please refer to those VI's for more details.



Example VI's:

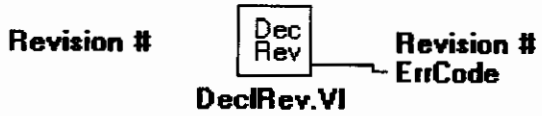
There are many example programs included with this package that demonstrate how to use the low level VI's. It is strongly suggested that you take a look at these examples before you begin in order to help you understand how to integrate the extensions into your program. The following is a list of the example programs and a brief description of what they do:

Example VI	Description
XAIN	Single analog input in a while loop with a metered display
XAINSCBG	Analog input scan in the background. Uses GetStat, StopBg and OptAin VI's. Display's data on a graph.
XAINSCFG	Analog input scan in the foreground. Uses SelChan and OptAIn VI's. Display's data on a graph.
XAOUT	Single analog output. Demonstrates sequences, case statements, for loops and while loops.
XAOUTSCB	Analog output scan in the background. Uses GetStat and StopBg VI's.
XAOUTSCF	Analog output scan in the foreground. Generates sinusoidal data.
XAPRETRB	Analog pre-trigger in the background. Uses GetStat, StopBg and ACnvPrDt VI's. Displays a graph.
XAPRETRF	Analog pre-trigger in the foreground. Uses SelChan and ACnvPrDt VI's. Displays a graph.
XASCFILE	Analog input to a file. Uses FileAInS and FileRead. Display's data on a graph.
XASCMEM	Analog input to memory board. Uses MemReset and MemRead. Display's data on a graph.
XCFREQ	Displays frequency of signal at counter input. Uses C9513Ini and CFreqIn.
XCSTORE	Stores counter values when interupts occur and diplays them. Uses CStore, GatStat and StopBg.
XCTR8254	Configures, loads and reads the counter. Displays the count. Uses C8254Cfg, CLoad and CIn.
XCTR8536	Inits, configs, loads and reads the counter. Displays the count. Uses C8536Cfg, C8536Cfg, CLoad and CIn.
XCTR9513	Inits, configs, loads and reads the counter. Displays the count. Uses C9513Cfg, C9513Cfg, CLoad and CIn.
XDBITIN	Configures and reads a digital bit. Toggles an LED accordingly. Uses DCfgPort and DBitIn.
XDBITOUT	Configures and writes a digital bit. Uses DCfgPort and DBitOut.
XDIN	Configures and reads a digital port. Toggles 8 LED's accordingly. Uses DCfgPort and DIn.
XDINSCBG	Reads multiple bytes in the background. Uses DCfgPort, DInScBg, GetStat and StopBg.
XDINSCFG	Reads multiple bytes in the foreground. Uses DCfgPort, DInScFg.
XDOUT	Configures and writes a digital port. Uses DCfgPort and DOut.
XDOUTSBG	Writes multiple bytes in the background. Uses DCfgPort, DOutScBg, GetStat and StopBg.
XDOUTSFG	Writes multiple bytes in the foreground. Uses DCfgPort, DOutScFg

DeclRev.VI New R3.3 ID

Description: Initializes Universal Library with the revision number of the library used to write your program. Must be the first Universal Library VI to be called by your program.

Summary:



Inputs: RevNum - Revision number of Library to interpret VI arguments

Outputs: RevNum - Revision number of Library to interpret VI arguments
ErrCode - 0 - If revision less than 3.2 is declared then revison 3.2 is assumed.

Default: Any program not containing this line of code will be defaulted to revision 3.2 argument assignments.

As new revisions of the library are released bugs from previous revisions are fixed and occasionally new functions are added. It is ComputerBoards goal to preserve existing programs you have written and therefore to never change the order or number of arguments in a function. Sometimes it is not possible to achieve this goal as evidenced in the changes from revision 3.2 to 3.3. In revision 3.3 we added support for multiple background tasks, a feature customers had requested for some time. Allowing multiple background tasks meant adding the argument BoardNum to several functions. Doing so would have meant that programs written for version 3.2 would not run with 3.3 if they called those functions. If not for the new DeclRev VI, the programs would have to be rewritten in each instance where the affected VI's are used.

The revision control VI initializes the DLL so that the VI's are interpreted according to the format of the revision you wrote your program in. This VI is new in revision 3.3 so to take advantage of it, the VI must be added to your program.

The VI works by interpreting the UL function call from your program and filling in any arguments needed to run with the new revision. For example, the VI ACvtData had the argument BoardNum added in revision 3.3.

If your program has declared you are running code written for revision 3.2 and you call this VI, the argument BoardNum is ignored. If you want the benefits afforded by the added argument BoardNum you must rewrite your program with the new VI and declare you are at revision 3.3 (or higher) in DeclRev.

GetRev.VI **New R3.3 ID**

Description: Gets the revision level of the Universal Library DLL and VXD.

Summary:



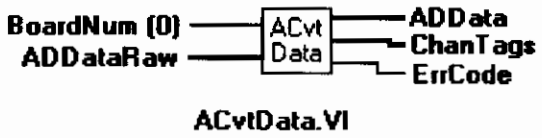
Inputs: None

Outputs: DLLRevNum - Place holder for the revision number of Library DLL
VXDRevNum - Place holder for the revision number of Library VXD
ErrCode - Error code if revision levels of VXD and DLL are incompatible.

ACvtData.VI Changed R3.3 MOD

Description: Converts the raw data collected by AInScFg.VI or AInScBg.VI into 12 bit A/D values. The AInScxx VI's can return either raw A/D data or converted data depending on whether or not the CONVERTDATA option was set. The raw data is a 16 bit value that contains a 12 bit A/D value and a 4 bit channel tag. The converted data consists of just the 12 bit A/D value.

Summary:



ACvtData.VI

Inputs: BoardNum - The board number used to collect the data.
ADDataRaw - Data and channel tags from AInScxx.VI.

Outputs: ADData - converted data
ChanTags - channel tags if available
ErrCode - Error code

When you collect data with AInScxx and you don't use the CONVERTDATA option then you must use this VI to convert the data after it is collected. There are cases where the CONVERTDATA option is not allowed. For example - if you are using the DMAIO option with AInScBg. In this case this VI should be used to convert the data after the data collection is complete.

Each raw data point consists of a 12 bit A/D value with a 4 bit channel number. This VI pulls each data point apart and puts the A/D value into the ADData array and the channel number into the ChanTags array.

Note - 12 Bit A/D Boards
1) Upon returning from ACvtData , ADData array contains only 12 bit A/D data.

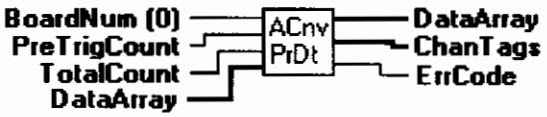
Note - 16 Bit A/D Boards
This VI is not for use with 16 bit A/D boards.

ACnvPrDt.VI Changed R3.3 MOD

Description: Converts the raw data collected by APretrFg.VI or APretrBg.VI. The APretrxx VI can return either raw A/D data or converted data depending on whether or not the CONVERTDATA option was used. The raw data as it is collected is not in the correct order. After the data collection is completed it must be rearranged into the correct order. This VI correctly orders the data, starting with the first pretrigger data point and ending with the last.

Change at revision 3.3 is to support multiple background tasks. It is now possible to run 2 boards with DMA or REP-INSW background convert-and-transfer features active, therefore, the convert vi must know which board the data came from.

Summary:



ACnvPrDt.VI

Inputs: PreTrigCount - Number of pre-trigger samples
Total Count - Total number of samples that were collected
DataArray - Data and channel tags

Outputs: DataArray - Converted data
ChanTags - channel tags if available
ErrCode - Error code

When you collect data with APretrFg VI or APretrBg VI and you don't use the CONVERTDATA option then you must use this VI to convert the data after it is collected. There are cases where the CONVERTDATA option is not allowed. For example - if you use the BACKGROUND option with APretrxx.VI. In those cases this VI should be used to convert the data after the data collection is complete.

Each raw data point consists of a 12 bit A/D value with a 4 bit channel number. This VI pulls each data point apart and puts the A/D value into the DataArray and the channel number into the ChanTags array.

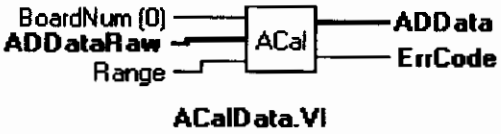
Note - 12 Bit A/D Boards
1) Upon returning from APretrxx.VI, DataArray contains only 12 bit A/D data.

Note - 16 Bit A/D Boards
This VI is for use with 16 bit A/D boards only insofar as ordering the data.

ACalData.VI New R3.3

Description: Calibrates the raw data collected by AInScxx from boards with real time software calibration when the real time calibration has been turned off. The AInScxx VI can return either raw A/D data or calibrated data depending on whether or not the NOCALIBRATEDATA option was used.

Summary:



ACalData.VI

Inputs: BoardNum - Board Number
NumPoints - Number of samples to convert
Gaon - The programmable gain/range used when the data was collected
ADData - Pointer to data array

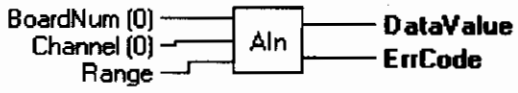
Outputs: ErrCode - Error Code

When you collect data with AInScxx and you use the NOCALIBRATEDATA option then you must use this VI to calibrate the data after it is collected.

AIn.VI

Description: Reads an A/D input channel. This VI reads the specified A/D channel from the specified board. If the specified A/D board has programmable gain then it sets the gain to the specified range. The raw A/D value is converted to an A/D value and returned to DataValue.

Summary:



AIn.VI

Inputs: BoardNum - board number may be 0 to 9.
Channel - A/D channel number
Range - A/D Range

Outputs: DataValue - Value of A/D sample
ErrCode - Error code

BoardNum - refers to the board number associated with the board when it was installed with the InstaCal™ configuration program. The specified board must have an A/D.

Channel - The maximum allowable channel depends on which type of A/D board is being used. For boards that have both single ended and differential inputs the maximum allowable channel number also depends on how the board is configured. For example, a CIO-DAS1600 has 8 chans for differential, 16 for single ended.

Range - If the selected A/D board does not have a programmable gain feature then this argument should always be set to NOT PROGRAMMABLE. If the A/D board does have programmable gain then set the Range argument to the desired A/D range. Not all A/D boards support the same A/D ranges. Refer to the A/D board manual for a list of supported A/D Ranges.

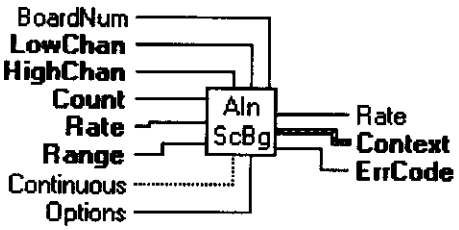
BIP10VOLTS	+/- 10 volts	UNI10VOLTS	0 - 10 volts
BIP5VOLTS	+/- 5 volts	UNI5VOLTS	0 - 5 volts
BIP2PT5VOLTS	+/- 2.5 volts	UNI2PT5VOLTS	0 - 2.5 volts
BIP1PT67VOLTS	+/- 1.67 volts	UNI2VOLTS	0 - 2 volts
BIP1PT25VOLTS	+/- 1.25 volts	UNI1PT67VOLTS	0 - 1.67 volts
BIP1VOLTS	+/- 1 volts	UNI1PT25VOLTS	0 - 1.25 volts
BIPPT625VOLTS	+/- 0.625 volts	UNI1VOLTS	0 - 1 volts
BIPPT5VOLTS	+/- 0.5 volts	UNIPT1VOLTS	0 - 0.1 volts
BIPPT1VOLTS	+/- 0.1 volts	UNIPT01VOLTS	0 - 0.01 volts
BIPPT05VOLTS	+/-0.05 volts	MA4TO20	4 - 20 mA
BIPPT01VOLTS	+/- 001 volts	MA2TO10	2 - 10 mA
BIPPT005VOLTS	+/- 0.005 volts	MA1TO5	1 - 5 mA
		MAPT5TO2PT5	0.5 - 2.5 mA

AInScBg.VI Changed R3.3 ID

Description: Scans a range of A/D channels in the background and stores the samples in an array. This VI reads the specified number of A/D samples at the specified sampling rate from the specified range of A/D channels from the specified board. If the A/D board has programmable gain then it sets the gain to the specified range. The collected data is returned to the data array. This VI immediatly returns control to your program and the data collection from the A/D into ADData will continue in the background. Use the GetStat VI to check on the status of the background operation. Use the StopBg VI to terminate the background process before or after it has completed.

Changes: Revision 3.3 added no real time calibration option. See OptAIn.VI for details.

Summary:



AInScBg.VI

Inputs: BoardNum - board number may be 0 to 9.
LowChan - First A/D channel of scan
HighChan - Last A/D channel of scan
Count - Number of A/D samples to collect
Rate - Sample rate in scans per second
Range - A/D range code
Continuous - Run the VI in an endless loop
Options - Bit fields that control various options ***NOTE1**

Outputs: Rate - Actual rate the board sampled
Context - Output data structure ***NOTE2**
ErrCode - Error Code

EXPLANATION OF THE ARGUMENTS:

BoardNum - refers to the board number associated with the board when it was installed with the InstaCal configuration program. The specified board must have an A/D.

Low / High Channel # - The maximum allowable channel depends on which type of A/D board is being used. For boards that have both single ended and differential inputs the maximum allowable channel number also depends on how the board is configured. For example, a CIO-DAS1600 has 8 chans for differential, 16 for single ended.

Count - Specifies the total number of A/D samples that will be collected. If more than one channel is being sampled then the number of samples collected per channel is equal to Count / (HighChan- LowChan+1).

Rate - This is the rate at which scans are triggered. If you are sampling 4 channels, 0-3, then specifying a rate of 10,000 scans per second (10KHz) will result in the A/D converter rate of 40KHz: 4 channels at 10,000 samples per channel per second. This is different from some software where you specify the total A/D chip rate. In those systems, the per channel rate is equal to the A/D rate divided by the number of channels in a scan.

Caution! You will generate an error if you specify a total A/D rate beyond the capability of the board. For example; if you specify rate LowChan = 0, HighChan = 7 (8 channels total) and Rate = 20,000 and you are using a CIO-DAS16/Jr, you will get an error. You have specified a total rate of 8*20,000 = 160,000. The CIO-DAS16/Jr is capable of converting 120,000 samples per second. The maximum sampling rate depends on the A/D board that is being used. It is also dependent on the sampling mode options.

Range - If the selected A/D board does not have a programmable range feature, then this argument should be set to NOT PROGRAMMABLE. Otherwise the gain can be set to any of the following ranges that are supported by the selected A/D board. Refer to board specific information for the list of ranges supported by each board.

BIP10VOLTS	+/- 10 volts	UNI10VOLTS	0 - 10 volts
BIP5VOLTS	+/- 5 volts	UNI5VOLTS	0 - 5 volts
BIP2PT5VOLTS	+/- 2.5 volts	UNI2PT5VOLTS	0 - 2.5 volts
BIP1PT67VOLTS	+/- 1.67 volts	UNI2VOLTS	0 - 2 volts
BIP1PT25VOLTS	+/- 1.25 volts	UNI1PT67VOLTS	0 - 1.67 volts
BIP1VOLTS	+/- 1 volts	UNI1PT25VOLTS	0 - 1.25 volts
BIPPT625VOLTS	+/- 0.625 volts	UNI1VOLTS	0 - 1 volts
BIPPT5VOLTS	+/- 0.5 volts	UNIPT1VOLTS	0 - 0.1 volts
BIPPT1VOLTS	+/- 0.1 volts	UNIPT01VOLTS	0 - 0.01 volts
BIPPT05VOLTS	+/-0.05 volts	MA4TO20	4 - 20 mA
BIPPT01VOLTS	+/- 001 volts	MA2TO10	2 - 10 mA
BIPPT005VOLTS	+/- 0.005 volts	MA1TO5	1 - 5 mA
		MAPT5TO2PT5	0.5 - 2.5 mA

CONTINUOUS - This option puts the VI in an endless loop. Once it collects the required number of samples, it resets to the start of ADData and begins again. The only way to stop this operation is with StopBg VI.

Options - For a detailed explanation of the Options field, please refer to the OPTAIN.VI section of this manual.

***NOTE1:** The OptAIn.VI must be wired to this input.

Context - Data structure containing information from a background operation. Some of the information included is the board number, the data array, the array size, and the initial status of the background operation.

***NOTE2:**

Wiring of this VI should conform to the following pattern:

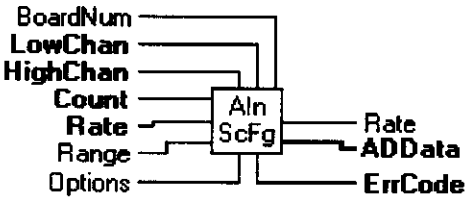
AInScBg.VI starts a background operation. GetStat.VI checks for completion(boolean output called "Running"). StopBg.VI terminates the operation, if not already done, and frees memory aliases. Data output from the background operation is passed to GetStat.VI and StopBg.VI via "Context", and can be wired from one or both of them for intermediate or final actions, respectively. The demo VI's illustrate this process effectively.

AInScFg.VI Changed R3.3 ID

Description: Scans a range of A/D channels in the foreground and stores the samples in an array. This VI reads the specified number of A/D samples at the specified sampling rate from the specified range of A/D channels from the specified board. If the A/D board has programmable gain then it sets the gain to the specified range. The collected data is returned to the data array. This VI will not return control to your program until all requested data has been collected and returned to ADData.

Changes: Revision 3.3 added no real time calibration option. See OptAIn.VI for details.

Summary:



AInScFg.VI

Inputs: BoardNum - board number may be 0 to 9.
LowChan - First A/D channel of scan
HighChan - Last A/D channel of scan
Count - Number of A/D samples to collect
Rate - Sample rate in scans per second
Range - A/D range code
Options - Bit fields that control various options ***NOTE1**

Outputs: Rate - Actual rate the board sampled
ADData - Data array to store A/D values in
ErrCode - Error Code

EXPLANATION OF THE ARGUMENTS:

BoardNum - refers to the board number associated with the board when it was installed with the InstaCal configuration program. The specified board must have an A/D.

Low / High Channel # - The maximum allowable channel depends on which type of A/D board is being used. For boards that have both single ended and differential inputs the maximum allowable channel number also depends on how the board is configured. For example, a CIO-DAS1600 has 8 chans for differential, 16 for single ended.

Count - Specifies the total number of A/D samples that will be collected. If more than one channel is being sampled then the number of samples collected per channel is equal to Count / (HighChan- LowChan+1).

Rate - This is the rate at which scans are triggered. If you are sampling 4 channels, 0-3, then specifying a rate of 10,000 scans per second (10KHz) will result in the A/D converter rate of 40KHz: 4 channels at 10,000 samples per channel per second. This is different from some software where you specify the total A/D chip rate. In those systems, the per channel rate is equal to the A/D rate divided by the number of channels in a scan.

Caution! You will generate an error if you specify a total A/D rate beyond the capability of the board. For example; if you specify rate LowChan = 0, HighChan = 7 (8 channels total) and Rate = 20,000 and you are using a CIO-DAS16/Jr, you will get an error. You have specified a total rate of 8*20,000 = 160,000. The CIO-DAS16/Jr is capable of converting 120,000 samples per second. The maximum sampling rate depends on the A/D board that is being used. It is also dependent on the sampling mode options.

Range - If the selected A/D board does not have a programmable range feature, then this argument should be set to NOT PROGRAMMABLE. Otherwise the gain can be set to any of the following ranges that are supported by the selected A/D board. Refer to board specific information for the list of ranges supported by each board.

BIP10VOLTS	+/- 10 volts	UNI10VOLTS	0 - 10 volts
BIP5VOLTS	+/- 5 volts	UNI5VOLTS	0 - 5 volts
BIP2PT5VOLTS	+/- 2.5 volts	UNI2PT5VOLTS	0 - 2.5 volts
BIP1PT67VOLTS	+/- 1.67 volts	UNI2VOLTS	0 - 2 volts
BIP1PT25VOLTS	+/- 1.25 volts	UNI1PT67VOLTS	0 - 1.67 volts
BIP1VOLTS	+/- 1 volts	UNI1PT25VOLTS	0 - 1.25 volts
BIPPT625VOLTS	+/- 0.625 volts	UNI1VOLTS	0 - 1 volts
BIPPT5VOLTS	+/- 0.5 volts	UNIPT1VOLTS	0 - 0.1 volts
BIPPT1VOLTS	+/- 0.1 volts	UNIPT01VOLTS	0 - 0.01 volts
BIPPT05VOLTS	+/-0.05 volts	MA4TO20	4 - 20 mA
BIPPT01VOLTS	+/- 001 volts	MA2TO10	2 - 10 mA
BIPPT005VOLTS	+/- 0.005 volts	MA1TO5	1 - 5 mA
		MAPT5TO2PT5	0.5 - 2.5 mA

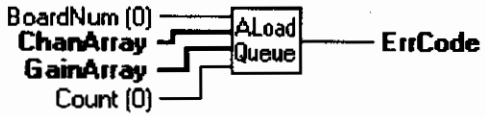
Options - For a detailed explanation of the Options field, please refer to the OPTAIN.VI section of this manual.

***NOTE1**: The OptAIn.VI must be wired to this input.

ALoadQue.VI

Description: Loads A/D board's channel/gain queue. This VI only works with A/D boards that have channel/gain queue hardware.

Summary:



ALoadQue.VI

Inputs: BoardNum - board number may be 0 to 9.
ChanArray - Array containing channel values
GainArray - Array containing A/D range values
Count - Number of elements in ChanArray and GainArray or 0 to disable the board's chan/gain queue.

Outputs: ErrCode - Error code

EXPLANATION OF THE ARGUMENTS:

BoardNum - refers to the board number associated with the board when it was installed with the configuration program. The specified board must have an A/D and a channel/gain queue.

ChanArray - This array should contain all of the channels that will be loaded into the channel gain queue.

GainArray - This array should contain each of the A/D ranges that will be loaded into the channel gain queue.

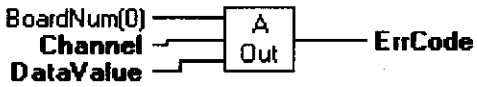
Count - Specifies the total number of chan/gain pairs that will be loaded into the queue. ChanArray and GainArray should contain at least Count elements. Set Count=0 to disable the board's chan/gain queue.

Normally the AInScxx VI scans a fixed range of channels (from LowChan to HighChan) at a fixed A/D range. If you load the channel gain queue with this VI, then all subsequent calls toAInScxx will cycle through the chan/range pairs that you have loaded into the queue.

AOut.VI

Description: Sets the value of a D/A output.

Summary:



AOut.VI

Inputs: BoardNum - board number may be 0 to 9.
Channel - A/D channel number
DataValue - Value to set D/A to

Outputs: ErrCode - Error code

EXPLANATION OF THE ARGUMENTS:

BoardNum - refers to the board number associated with the board when it was installed with the configuration program. The specified board must have a D/A.

Channel - The maximum allowable channel depends on which type of D/A board is being used.

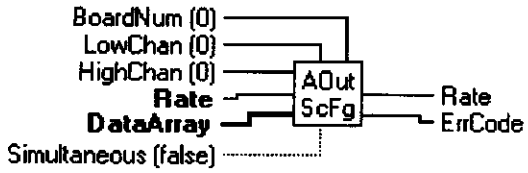
DataValue - Must be in the range 0 - N where N is the value 2 ^ Resolution - 1 of the converter.

Simultaneous Update Boards: If you have set the simultaneous update jumper for simultaneous operation, you should use AOutScan.VI for simultaneous update of multiple channels. AOut.VI always writes the D/A data then reads the D/A, which causes the D/A output to be updated.

AOutScFg.VI

Description: Outputs values to a range of D/A channels in the foreground.

Summary:



AOutScFg.VI

Inputs: BoardNum - board number may be 0 to 9.
LowChan - First A/D channel of scan
HighChan - Last A/D channel of scan
Rate - Sample rate in scans per second
Data Array - Data array to store A/D values in.
Simultaneous - Simultaneous update

Outputs: ErrCode - Error code.

EXPLANATION OF THE ARGUMENTS:

BoardNum - refers to the board number associated with the board when it was installed with the configuration program. The specified board must have a D/A.

LowChan/HighChan - The maximum allowable channel depends on which type of D/A board is being used. When using the SBX-DDA04 board you may specify the channel range with LowChan/HighChan OR you can set LowChan and HighChan = NOTUSED and specify the channel number in the 4 MSB's of the data values in DataArray. The D/A channels in the range specified by LowChan- HighChan are set from consecutive elements in DataArray.

Rate - For most D/A boards the Rate is ignored and can be set to NOTUSED. For D/A boards with trigger and transfer methods which allow fast output rates, such as the cSBX-DDA04 , Rate should be set to the D/A output rate (in scans/sec)

If supported, this is the rate at which scans are triggered. If you are updating 4 channels, 0-3, then specifying a rate of 10,000 scans per second (10KHz) will result in the D/A converter rates of 10KHz: (one D/A per channel). The data transfer rate will be 40,000 words per second; 4 channels * 10,000 updates per scan.

Caution! You will generate an error if you specify a total D/A rate beyond the capability of the board. For example; if you specify rate LowChan = 0, HighChan = 3 (4 channels total) and Rate = 100,000 and you are using a cSBX-DDA04, you will get an error. You have specified a total rate of 4*100,000 = 400,000. The cSBX-DDA04 is rated to 330,000 updates per second.

The maximum update rate depends on the D/A board that is being used. It is also dependent on the sampling mode options.

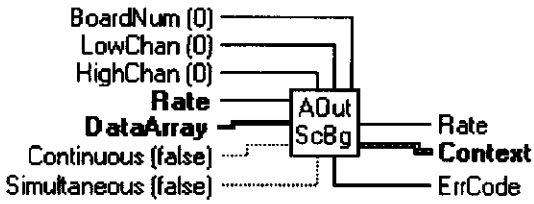
DataArray - The data array should be filled with D/A values in the range 0 - N where N is the value 2 ^ Resolution-1 of the converter. There should be at least HighChan-LowChan+1 elements in the array.

Simultaneous - When this options is set (and the appropriate switches are set on the board) all of the D/A voltages will be updated simultaneously when the last D/A in the scan is updated. This generally means that all the D/A values will be written to the board, then a read of a D/A address causes all D/As to be updated with new values simultaneously.

AOutScBg.VI

Description: Outputs values to a range of D/A channels in the background. This VI may only be used with boards that support interrupt, DMA or REP-INSW transfer methods. When this option is used the D/A operations will begin running in the background and control will immediately return to the VI. Use GetStat.VI to check the status of background operation. Use StopBg.VI to terminate background operations before or after they are completed.

Summary:



AOutScBg.VI

Inputs: BoardNum - board number may be 0 to 9.
LowChan - First A/D channel of scan
HighChan - Last A/D channel of scan
Rate - Sample rate in scans per second
DataArray - Data array to store A/D values in.
Continuous - Run the VI in an endless loop
Simultaneous - Simultaneous update

Outputs: Error code.

EXPLANATION OF THE ARGUMENTS:

BoardNum - refers to the board number associated with the board when it was installed with the configuration program. The specified board must have a D/A.

LowChan/HighChan - The maximum allowable channel depends on which type of D/A board is being used. When using the SBX-DDA04 board you may specify the channel range with LowChan/HighChan OR you can set LowChan and HighChan = NOTUSED and specify the channel number in the 4 MSB's of the data values in DataArray. The D/A channels in the range specified by LowChan- HighChan are set from consecutive elements in DataArray.

Rate - For most D/A boards the Rate is ignored and can be set to NOTUSED. For D/A boards with trigger and transfer methods which allow fast output rates, such as the cSBX-DDA04 , Rate should be set to the D/A output rate (in scans/sec)

If supported, this is the rate at which scans are triggered. If you are updating 4 channels, 0-3, then specifying a rate of 10,000 scans per second (10KHz) will result in the D/A converter rates of 10KHz: (one D/A per channel). The data transfer rate will be 40,000 words per second; 4 channels * 10,000 updates per scan.

Caution! You will generate an error if you specify a total D/A rate beyond the capability of the board. For example; if you specify rate LowChan = 0, HighChan = 3 (4 channels total) and Rate = 100,000 and you are using a cSBX-DDA04, you will get an error. You have specified a total rate of 4*100,000 = 400,000. The cSBX-DDA04 is rated to 330,000 updates per second.

The maximum update rate depends on the D/A board that is being used. It is also dependent on the sampling mode options.

DataArray - The data array should be filled with D/A values in the range 0 - N where N is the value 2 ^ Resolution-1 of the converter. There should be at least HighChan-LowChan+1 elements in the array.

Continuous - This option may only be used with boards which support interrupt, DMA or REP-INSW transfer methods. This option puts the VI in an endless loop. Once it outputs the specified (by Count) number of D/A values, it resets to the start of DataArray and begins again. The only way to stop this operation is with StopBg.VI.

Simultaneous- When this options is set (and the appropriate switches are set on the board) all of the D/A voltages will be updated simultaneously when the last D/A in the scan is updated. This generally means that all the D/A values will be written to the board, then a read of a D/A address causes all D/As to be updated with new values simultaneously.

Context - Data structure containing information from a background operation. Some of the information included is the board number, the data array, the array size, and the initial status of the background operation.

***NOTE2:**

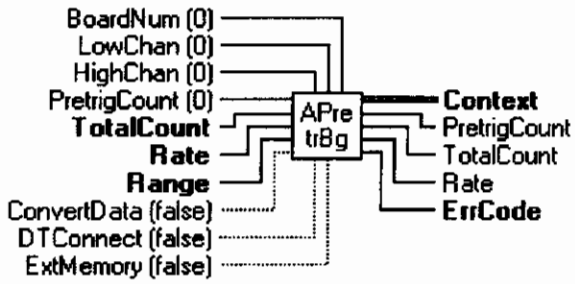
Wiring of this VI should conform to the following pattern:

AOutScBg.VI starts a background operation. GetStat.VI checks for completion(boolean output called "Running"). StopBg.VI terminates the operation, if not already done, and frees memory aliases. Data output from the background operation is passed to GetStat.VI and StopBg.VI via "Context", and can be wired from one or both of them for intermediate or final actions, respectively. The demo VI's illustrate this process effectively.

APretrBg.VI

Description: Waits for a trigger to occur and then returns a specified number of analog samples before and after the trigger occurred. The trigger input line (See board users manual but generally Digital In 0) must be at TTL low before this VI is called or a TRIGSTATE error will occur. The trigger occurs on the rising edge of the trigger input signal (transition to TTL high).After this VI is called, execution will return immediately to the next point in your program and the data collection from the A/D into DataArray will continue in the background. Use GetStat.VI to check on the status of the background operation. Use StopBg.VI to terminate the background process before or after it has completed.

Summary:



APretrBg.VI

Inputs:
BoardNum - board number may be 0 to 9.
LowChan - First A/D channel of scan
HighChan - Last A/D channel of scan
PretrigCount - Number of pre-trigger A/D samples to collect
TotalCount - Total number of A/D samples to collect
Rate - Sample rate in scans per second
Range - A/D Range code or 0
ConvertData - Convert data option (Boolean)
DTConnect - DT connect option(Boolean)
ExtMemory - External memory option(Boolean)

Outputs:
Context - Output data structure ***NOTE**
PretrigCount - Number of pre-trigger A/D samples collected
TotalCount - Total number of A/D samples collected
Rate - Actual sample rate in scans per second
ErrCode - Error code

EXPLANATION OF THE ARGUMENTS:

BoardNum - refers to the board number associated with the board when it was installed with the configuration program. The specified board must have an A/D.

LowChan/HighChan - The maximum allowable channel depends on which type of A/D board is being used. For boards that have both single ended and differential inputs the maximum allowable channel number also depends on how the board is configured (8 chans for differential, 16 for single ended).

PretrigCount - Specifies the number of samples before the trigger that will be returned. PretrigCount must be less than 32000 and PretrigCount must also be less than TotalCount-512. If the trigger occurs too early, then fewer than the requested number of pre-trigger samples will be collected. In that case a TOOFEW error will occur. The PretrigCount will be set to indicate how many samples were collected and the post trigger samples will still be collected.

TotalCount - Specifies the total number of samples that will be collected and stored in DataArray. TotalCount must be greater than or equal to PretrigCount+512. If the trigger occurs too early then fewer than the requested number of samples will be collected. In that case a TOOFEW error will occur. The TotalCount will be set to indicate how many samples were actually collected.

Range - If the selected A/D board does not have a programmable gain feature then this argument should be set to NOT PROGRAMMABLE. Otherwise the Range can be set to any of the following ranges that are supported by the selected A/D board. Refer to board specific information for a list of the A/D ranges supported by each board.

BIP10VOLTS	+/- 10 volts	UNI10VOLTS	0 - 10 volts
BIP5VOLTS	+/- 5 volts	UNI5VOLTS	0 - 5 volts
BIP2PT5VOLTS	+/- 2.5 volts	UNI2PT5VOLTS	0 - 2.5 volts
BIP1PT67VOLTS	+/- 1.67 volts	UNI2VOLTS	0 - 2 volts
BIP1PT25VOLTS	+/- 1.25 volts	UNI1PT67VOLTS	0 - 1.67 volts
BIP1VOLTS	+/- 1 volts	UNI1PT25VOLTS	0 - 1.25 volts
BIPPT625VOLTS	+/- 0.625 volts	UNI1VOLTS	0 - 1 volts
BIPPT5VOLTS	+/- 0.5 volts	UNIPT1VOLTS	0 - 0.1 volts
BIPPT1VOLTS	+/- 0.1 volts	UNIPT01VOLTS	0 - 0.01 volts
BIPPT05VOLTS	+/-0.05 volts	MA4TO20	4 - 20 mA
BIPPT01VOLTS	+/- 001 volts	MA2TO10	2 - 10 mA
BIPPT005VOLTS	+/- 0.005 volts	MA1TO5	1 - 5 mA
		MAPT5TO2PT5	0.5 - 2.5 mA

DataArray - The data array for the pretrigger data.

DTConnect - When DTCONNECT option is used with this VI, the data from ALL A/D conversions is sent out the DT-CONNECT interface. While this VI is waiting for a trigger to occur, it will send data out the DT-CONNECT interface continuously. If you have a Computer Boards memory board plugged into the DT-CONNECT interface then you should use EXTMEMORY option rather than this option.

ExtMemory - If you use this option to send the data to a connected memory board then you must use MemRdPrt.VI to later read the pre-trigger data from the memory board. If you use MemRead.VI, the data will NOT be in the correct order. Everytime this option is used it will overwrite any data that is already stored in the memory board. All data should be read from the board (with MemRdPrt.VI) before collecting any new data.

ConvertData - This option should always be set to FALSE(default) for background operations.

Context - This is a data structure containing output information including the board number, the contents of DataArray, the size of DataArray, and the initial status of the background operation. This CONTEXT must be wired to subsequent VI's in order to process this VI correctly.

***NOTE:**

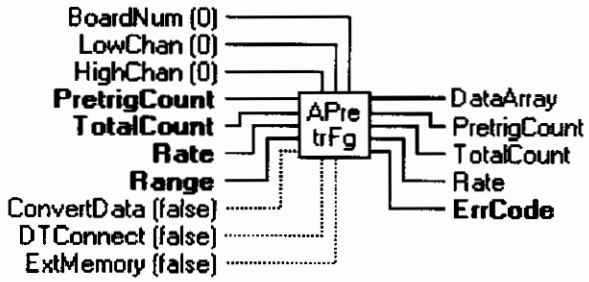
Wiring of this VI should conform to the following pattern:

APretrBg.VI starts a background operation. GetStat.VI checks for completion(boolean output called "Running"). StopBg.VI terminates the operation, if not already done, and frees memory aliases. Data output from the background operation is passed to GetStat.VI and StopBg.VI via "Context", and can be wired from one or both of them for intermediate or final actions, respectively. The demo VI's illustrate this process effectively.

APretrFg.VI

Description: Waits for a trigger to occur and then returns a specified number of analog samples before and after the trigger occurred. The trigger input line (See board users manual but generally Digital In 0) must be at TTL low before this VI is called or a TRIGSTATE error will occur. The trigger occurs on the rising edge of the trigger input signal (transition to TTL high).This VI will not return to your program until all of the requested data has been collected and returned to DataArray.

Summary:



APretrFg.VI

Inputs:
BoardNum - board number may be 0 to 9.
LowChan - First A/D channel of scan
HighChan - Last A/D channel of scan
PretrigCount - Number of pre-trigger A/D samples to collect
TotalCount - Total number of A/D samples to collect
Rate - Sample rate in scans per second
Range - A/D Range code or 0
ConvertData - Convert data option (Boolean)
DTConnect - DT connect option(Boolean)
ExtMemory - External memory option(Boolean)

Outputs:
DataArray - Data array to store A/D values in.
PretrigCount - Number of pre-trigger A/D samples collected
TotalCount - Total number of A/D samples collected
Rate - Actual sample rate in scans per second

EXPLANATION OF THE ARGUMENTS:

BoardNum - refers to the board number associated with the board when it was installed with the configuration program. The specified board must have an A/D.

LowChan/HighChan - The maximum allowable channel depends on which type of A/D board is being used. For boards that have both single ended and differential inputs the maximum allowable channel number also depends on how the board is configured (8 chans for differential, 16 for single ended).

PretrigCount - Specifies the number of samples before the trigger that will be returned. PretrigCount must be less than 32000 and PretrigCount must also be less than TotalCount-512. If the trigger occurs too early, then fewer than the requested number of pre-trigger samples will be collected. In that case a TOOFEW error will occur. The PretrigCount will be set to indicate how many samples were collected and the post trigger samples will still be collected.

TotalCount - Specifies the total number of samples that will be collected and stored in ADData. TotalCount must be greater than or equal to PretrigCount+512. If the trigger occurs too early then fewer than the requested number of samples will be collected. In that case a TOOFEW error will occur. The TotalCount will be set to indicate how many samples were actually collected.

Range - If the selected A/D board does not have a programmable gain feature then this argument should be set to NOT PROGRAMMABLE. Otherwise the Range can be set to any of the following ranges that are supported by the selected A/D board. Refer to board specific information for a list of the A/D ranges supported by each board.

BIP10VOLTS	+/- 10 volts	UNI10VOLTS	0 - 10 volts
BIP5VOLTS	+/- 5 volts	UNI5VOLTS	0 - 5 volts
BIP2PT5VOLTS	+/- 2.5 volts	UNI2PT5VOLTS	0 - 2.5 volts
BIP1PT67VOLTS	+/- 1.67 volts	UNI2VOLTS	0 - 2 volts
BIP1PT25VOLTS	+/- 1.25 volts	UNI1PT67VOLTS	0 - 1.67 volts
BIP1VOLTS	+/- 1 volts	UNI1PT25VOLTS	0 - 1.25 volts
BIPPT625VOLTS	+/- 0.625 volts	UNI1VOLTS	0 - 1 volts
BIPPT5VOLTS	+/- 0.5 volts	UNIPT1VOLTS	0 - 0.1 volts
BIPPT1VOLTS	+/- 0.1 volts	UNIPT01VOLTS	0 - 0.01 volts
BIPPT05VOLTS	+/-0.05 volts	MA4TO20	4 - 20 mA
BIPPT01VOLTS	+/- 001 volts	MA2TO10	2 - 10 mA
BIPPT005VOLTS	+/- 0.005 volts	MA1TO5	1 - 5 mA
		MAPT5TO2PT5	0.5 - 2.5 mA

DataArray - The data array for the pretrigger data.

ConvertData - The data is collected into a "circular" buffer. When the data collection is complete the data is in the wrong order. If you use the CONVERTDATA option then the data will be automatically rotated into the correct order and converted to 12 bit values when the data acquisition is complete. Otherwise you must call ACnvPrDt.VI to rotate the data.

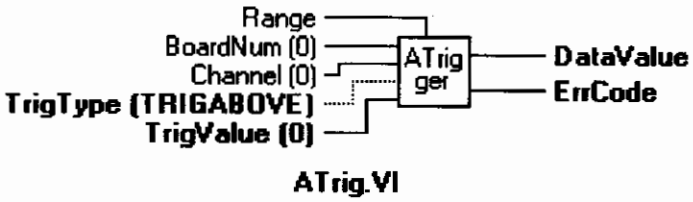
DTConnect - When DTCONNECT option is used with this VI the data from ALL A/D conversions is sent out the DT-CONNECT interface. While this VI is waiting for a trigger to occur, it will send data out the DT-CONNECT interface continuously. If you have a Computer Boards memory board plugged into the DT-CONNECT interface then you should use EXTMEMORY option rather than this option.

ExtMemory - If you use this option to send the data to a connected memory board then you must use MemRdPrt.VI to later read the pre-trigger data from the memory board. If you use MemRead.VI the data will NOT be in the correct order. Everytime this option is used it will overwrite any data that is already stored in the memory board. All data should be read from the board (with MemRdPrt.VI before collecting any new data.

ATrig.VI

Description: Waits for a specified analog input channel to go above or below a specified value. This VI continuously reads the specified channel and compares its value to TrigValue. Depending on whether TrigType is ABOVE or BELOW it waits for the first A/D sample that is above or below TrigValue. It returns the first sample that meets the trigger criteria to DataValue.

Summary:



Inputs: Range - Input range code
BoardNum - board number may be 0 to 9.
Channel - A/D channel number (0-7 for differential input, 0-15 for single ended)
TrigType - TRIGABOVE or TRIGBELOW - Specifies whether waiting for the analog input to be ABOVE or BELOW the specified trigger value.
TrigValue - The threshold value that all A/D values are compared to

Outputs: DataValue - The value of the first A/D sample that met the trigger criteria is returned here.
ErrCode - Error code

EXPLANATION OF THE ARGUMENTS:

BoardNum - refers to the board number associated with the board when it was installed with the configuration program. The specified board must have an A/D.

Chan - The maximum allowable channel depends on which type of A/D board is being used. For boards that have both single ended and differential inputs the maximum allowable channel number also depends on how the board is configured. For example a CIO-DAS1600 has 8 chans for differential, 16 for single ended.

TrigValue - Must be in the range 0 - 4095 for 12 bit A/D boards, or 0-65,535 for 16 bit A/D boards.

Range - If the selected A/D board does not have a programmable gain feature then this argument should be set to NOT PROGRAMMABLE. Otherwise the Range can be set to any of the following ranges that are supported by the selected A/D board. Refer to board specific information for a list of the A/D ranges supported by each board.

BIP10VOLTS	+/- 10 volts	UNI10VOLTS	0 - 10 volts
BIP5VOLTS	+/- 5 volts	UNI5VOLTS	0 - 5 volts
BIP2PT5VOLTS	+/- 2.5 volts	UNI2PT5VOLTS	0 - 2.5 volts
BIP1PT67VOLTS	+/- 1.67 volts	UNI2VOLTS	0 - 2 volts
BIP1PT25VOLTS	+/- 1.25 volts	UNI1PT67VOLTS	0 - 1.67 volts
BIP1VOLTS	+/- 1 volts	UNI1PT25VOLTS	0 - 1.25 volts
BIPPT625VOLTS	+/- 0.625 volts	UNI1VOLTS	0 - 1 volts
BIPPT5VOLTS	+/- 0.5 volts	UNIPT1VOLTS	0 - 0.1 volts
BIPPT1VOLTS	+/- 0.1 volts	UNIPT01VOLTS	0 - 0.01 volts
BIPPT05VOLTS	+/-0.05 volts	MA4TO20	4 - 20 mA
BIPPT01VOLTS	+/- 001 volts	MA2TO10	2 - 10 mA
BIPPT005VOLTS	+/- 0.005 volts	MA1TO5	1 - 5 mA
		MAPT5TO2PT5	0.5 - 2.5 mA

Windows Caution - This VI should be used with caution in Windows programs. All active windows will be locked on the screen until the trigger condition is satisfied. All keyboard and mouse actions will also be locked until the trigger condition is satisfied.

COUNTERS - AN INTRODUCTION

Universal Library LabVIEW Extensions provide VI's for initialization and configuration of counter chips. It is important to note what this means:

1) VI's can configure a counter for any of the counters operations.

2) Counter configuration does not include USE of counters such as event counting and pulse width. Counter use is accomplished by programs which use the counter VI's.

3) Some counter USE VI's may be available. A VI CFreqIn is provided. Others may be added to later revisions.

For you to use a counter for any but the simplest counting VI, you must have, read, understand and employ the information contained in the chip manufacturer's data sheet. Technical support of the VI's does not include providing, interpreting or explaining the counter chip data sheet.

Counter chip data sheets are available from:

82C54

Intel Corporation
Literature Department
(800) 548-4725
Request part No. 230843, Micro and Peripherals Handbook, a 2 volume set.
As of this writing there is no charge for this set.

AM9513

Advanced Micro Devices, Inc.
901 Thompson Place
P.O. Box 3453
Sunnyvale, CA 94088
(408) 732-2400

As of this writing the data book, the Am9513 Technical Manual is available free of charge.

Z8536

As of this writing, the only ComputerBoards part that employs the Z8536 is the CIO-INT32. The data book for the chip is included with the CIO-INT32.

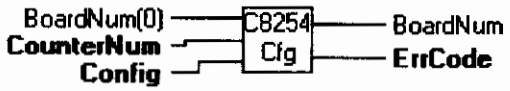
Counter Chip Variables

Universal Library counter initialization and configuration VI's include names for bit patterns, such as ALEGATE, which stands for Active Low Enabled Gate N. In any case where Universal Library has a name for a bit pattern, it is allowed to substitute the bit pattern as a numeric. This will work, but your programs will be harder to read and debug.

C8254Cfg.VI

Description: Configures 8254 counter for desired operation. There are three types of counter chips used by Computer Boards - 8254, 8536 and 9513 chips. This VI can only be used with 8254 counters.

Summary:



C8254Cfg.VI

Inputs: BoardNum - board number may be 0 to 9.
CounterNum - counter number to configure
Config - the action to take on terminal count and the waveform if any.

Outputs: BoardNum - Board number
ErrCode - Error code

EXPLANATION OF THE ARGUMENTS:

CounterNum - Selects one of the counter channels. An 8254 has 3 counters. The value may be 1, 2 or 3.

Config - Refer to the 8254 data sheet for a detailed description of each of the configurations. May be set to one of the following constants:

HIGHONLASTCOUNT: Output of counter (OUT N) transitions from low to high on terminal count and remains high until reset. See Mode0 on 8254 data sheet.

ONESHOT: Output of counter (OUT N) transitions from high to low on rising edge of GATE N, then back to high on terminal count. See mode 1 on 8254 data sheet

RATEGENERATOR: Output of counter (OUT N) pulses low for one clock cycle on terminal count and reloads the counter and re-cycles. See mode 2 on 8254 data sheet.

SQUAREWAVE : Output of counter (OUT N) is high for count < 1/2 terminal count then low until terminal count, whereupon it recycles. This mode generates a square wave. See mode 3 on 8254 data sheet.

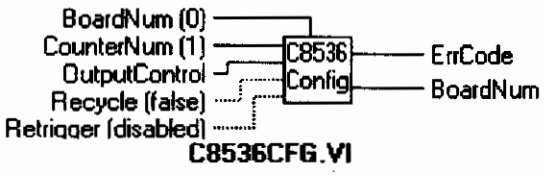
SOFTWARESTROBE : Output of counter (OUT N) pulses low for one clock cycle on terminal count. Count starts after counter is loaded. See mode 4 on 8254 data sheet.

HARDWARESTROBE : Output of counter (OUT N) pulses low for one clock cycle on terminal count. Count starts on rising edge at GATE N input. See mode 5 on 8254 data sheet.

C8536Cfg.VI

Description: Configures 8536 counter for desired operation. There are three types of counter chips used by Computer Boards - 8254, 8536 and 9513 chips. This VI can only be used with 8536 counters.

Summary:



Inputs: BoardNum - board number may be 0 to 9.
CounterNum - counter number to configure
OutputControl - Specifies counter output signal used.
Recycle - Execute once or reload and rexecute until stopped.
Retrigger - Enable or disable retriggering.

Outputs: ErrCode - Error code
BoardNum - Board number

EXPLANATION OF THE ARGUMENTS:

CounterNum - Selects one of the counter channels. An 8536 has 3 counters. The value may be 1, 2 or 3.

OutputControl - Specifies the action of the output signal. The options for this argument are:
HIGHPULSEONTC - Output will transition from low to high for one clock pulse on terminal count.
TOGGLEONTC - Output will change state on terminal count.
HIGHUNTILTC - Output will transition to high at the start of counting then go low on terminal count.

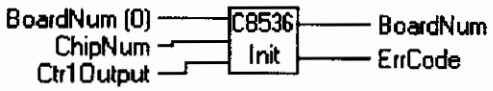
RecycleMode - If set to RECYCLE (as opposed to ONETIME) then the counter will automatically reload to the starting count every time it reaches 0, then counting will continue.

Retrigger - If set to ENABLED then every trigger on the counter's trigger input will initiate loading of the initial count and counting will proceed from initial count.

C8536Ini.VI

Description: Initializes the counter linking features of an 8536 counter chip. See the 8536 data sheet Counter/Timer Link Controls section for a complete description of the hardware affected by this mode. The linking of counters 1 & 2 must be accomplished prior to enabling the counters.

Summary:



C8536INI.VI

Inputs: BoardNum - board number may be 0 to 9.
ChipNum - chip number to configure
CtrlOutput - Specifies counter output signal used.

Outputs: ErrCode - Error code
BoardNum - Board number

EXPLANATION OF THE ARGUMENTS:

BoardNum - Board number of board with 8536 counter installed.

ChipNum - Selects one of the 8536 chips on the board, 1 to n.

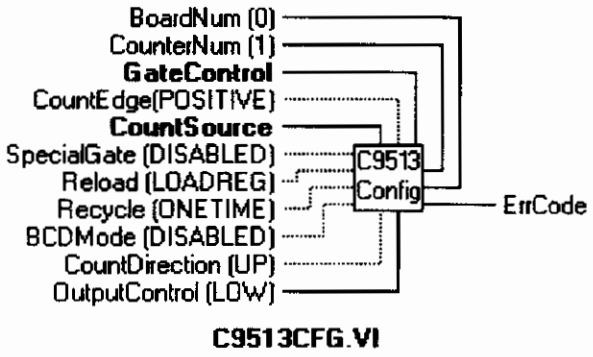
CtrlOutput - Specifies how the counter 1 is to be linked to counter 2, if at all. The options for this argument are:

NOTLINKED -Counter 1 is not connected to any other counters inputs.
GATECTR2 - Output of counter 1 is connected to the GATE of counter #2.
TRIGCTR2 - Output of counter 1 is connected to the trigger of counter #2.
INCTR2 - Output of counter 1 is connected to counter #2 clock input.

C9513Cfg.VI

Description: Sets all of the configurable options of a 9513 counter.

Summary:



Inputs: BoardNum - board number may be 0 to 9.
CounterNum - counter number (1 - n)
GateControl - gate control
CountEdge - which edge to count
CountSource - which of the available count sources to use.
SpecialGate - special gate may be enabled or disabled.
Reload - Load or load and hold.
Recycle - Execute once or reload and recycle.
BCDMode - Counter may operate in Binary Coded Decimal if desired.
CountDirection - AM9513 may count up or down.
OutputControl - The type of output desired.

Output: ErrCode - Error Code

EXPLANATION OF THE ARGUMENTS:

ADVICE: *The information provided here and in C9513Ini will only help you understand how Universal Library syntax corresponds to the 9513 data sheet. It is not a substitute for the data sheet. You cannot program and use a 9513 without the manufacturers' data book.*

BoardNum - refers to the board number associated with the board when it was installed with the configuration program. The specified board must have a 9513 counter.

CounterNum - counter number (1 - n) where n is the number of counters on the board. A CIO-CTR5 has 5, a CIO-CTR10 has 10.

GateControl - gate control variables are:

VI Syntax	Corresponds to 9513 description in Counter Mode Register Description
NOGATE	No gating
AHLTCPREVCTR	Active high TCN-1
AHLNEXTGATE	Active High Level GATE N + 1
AHLPREVGATE	Active High Level GATE N - 1
AHLGATE	Active High Level GATE N
ALLGATE	Active Low Level GATE N
AHEGATE	Active High Edge GATE N
ALEGATE	Active Low Edge GATE N

CountEdge - which edge to count. Refered to Source Edge in 9513 data book.

VI Syntax	Corresponds to 9513 description in Counter Mode Register Description
POSITIVEEDGE	Count on Rising Edge
NEGATIVEEDGE	Count on Falling Edge

CountSource -

VI Syntax	Corresponds to 9513 description in Counter Mode Register Description	
TCPREVCTR	TCN - 1	(Terminal count of previous counter)
CTRINPUT1	SRC 1	(Counter Input 1)
CTRINPUT2	SRC 2	(Counter Input 2)
CTRINPUT3	SRC 3	(Counter Input 3)
CTRINPUT4	SRC 4	(Counter Input 4)
CTRINPUT5	SRC 5	(Counter Input 5)
GATE1	GATE 1	
GATE2	GATE 2	
GATE3	GATE 3	
GATE4	GATE 4	
GATE5	GATE 5	
FREQ1	F1	
FREQ2	F2	
FREQ3	F3	
FREQ4	F4	
FREQ5	F5	

SpecialGate -

VI Syntax	Corresponds to 9513 description in Counter Mode Register Description
ENABLED	Enable Special Gate
DISABLED	Disable Special Gate

Reload -

VI Syntax	Corresponds to 9513 description in Counter Mode Register Description
LOADREG	Reload from Load
LOADANDHOLDREG	Reload from Load or Hold except in Mode X which reloads only from Load

RecycleMode -

VI Syntax	Corresponds to 9513 description in Counter Mode Register Description
ONETIME	Count Once
RECYCLE	Count Repetitively

BCDMode -

VI Syntax	Corresponds to 9513 description in Counter Mode Register Description
DISABLED	Binary Count
ENABLED	BCD Count

CountDirection -

VI Syntax	Corresponds to 9513 description in Counter Mode Register Description
COUNTDOWN	Count Down
COUNTUP	Count Up

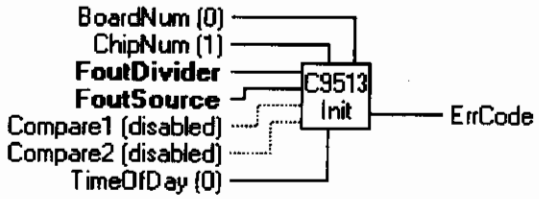
OutputControl -

VI Syntax	Corresponds to 9513 description in Counter Mode Register Description
ALWAYSLOW	Inactive, Output Low
HIGHPULSEONTC	Active High Terminal Count Pulse
TOGGLEONTC	TC Toggled
DISCONNECTED	Inactive, Output High Impedence
LOWPULSEONTC	Active Low Terminal Count Pulse
3, 6, 7 (numeric values)	Illegal

C9513Ini.VI

Description: Initializes all of the chip level features of a 9513 counter chip. This VI can only be used with 9513 counters.

Summary:



C9513INI.VI

Inputs: BoardNum - board number may be 0 to 9.
ChipNum - Specifies which 9513 chip is to be initialized.
FoutDivider - F-Out divider (0-15)
FoutSource - Specifies source of the signal for F-Out signal.
Compare1 - ENABLED or DISABLED
Compare2 - ENABLED or DISABLED
TimeOfDay - DISABLED or 1-3

Output: ErrCode - Error code

EXPLANATION OF THE ARGUMENTS:

BoardNum - refers to the board number associated with the board when it was installed with the configuration program. The specified board must have a 9513 counter.

ChipNum - Specifies which 9513 chip is to be initialized. For a CTR05 board this should be set to 1. For a CTR10 board it should be either 1 or 2.

FoutDivider -

VI Syntax	Corresponds to 9513 description in Counter Mode Register Description
0	Divide by 16
1	Divide by 1
2 ... 15	Divide by the number 2 ... 15

FoutSource -

VI Syntax	Corresponds to 9513 description in Counter Mode Register Description	
TCPREVCTR	TCN - 1	(Terminal count of previous counter)
CTRINPUT1	SRC 1	(Counter Input 1)
CTRINPUT2	SRC 2	(Counter Input 2)
CTRINPUT3	SRC 3	(Counter Input 3)
CTRINPUT4	SRC 4	(Counter Input 4)
CTRINPUT5	SRC 5	(Counter Input 5)
GATE1	GATE 1	
GATE2	GATE 2	
GATE3	GATE 3	
GATE4	GATE 4	
GATE5	GATE 5	
FREQ1	F1	
FREQ2	F2	
FREQ3	F3	
FREQ4	F4	
FREQ5	F5	

Compare1 -

VI Syntax	Corresponds to 9513 description in Counter Mode Register Description
DISABLED	Disabled
ENABLED	Enabled

Compare2 -

VI Syntax	Corresponds to 9513 description in Counter Mode Register Description
DISABLED	Disabled
ENABLED	Enabled

TimeOf Day -

VI Syntax	Corresponds to 9513 description in Counter Mode Register Description
0	TOD Disabled
1	TOD Enabled / 5 Input
2	TOD Enabled / 6 Input
3	TOD Enabled / 10 Input

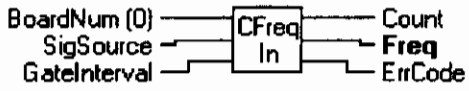
No Arguments - For:

VI Set To	Corresponds to 9513 description in Counter Mode Register Description
0 (FOUT on)	FOUT Gate
0 (Data bus matches board)	Data Bus Width
1 (Disable Increment)	Data Pointer Control
1 (BCD Scaling)	Scalar Control

CFreqIn.VI

Description: Measures the frequency of a signal. This VI can only be used with 9513 counters. This VI uses internal counters #5 and #4.

Summary:



CFreqIn.VI

Inputs: BoardNum - board number may be 0 to 9.
SigSource - specifies which signal will be measured
GateInterval - gating interval in milliseconds

Outputs: Count - The raw count is returned here
Freq - the measured frequency in Hz returned here.
ErrCode - Erroe code

EXPLANATION OF THE ARGUMENTS:

BoardNum - refers to the board number associated with the board when it was installed with the configuration program. The specified board must have a 9513 counter.

SigSource - specifies which signal will be measured

CTRINPUT1	Counter input to counter 1. CTR1
CTRINPUT2 .. 10	Counter inputs 2 through 10
GATE1 .. 10	Gate 1 through 10
FREQ1 .. 10	Frequency source 1 through 10

The counters in a single chip are numbered CTR1 through CTR5. On a board with 2 9513 chips, the second 9513 chip holds counters 6-10.

GateInterval - specifies the time (in milliseconds) that the counter will be counting. The optimum GateInterval depends on the frequency of the measured signal. The counter can count up to 65535.

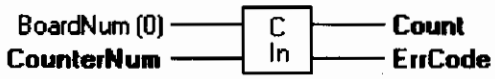
If the gating interval is too low, then the count will be too low and the resolution of the frequency measurement will be poor. For example, if the count changes from 1 to 2 the measured frequency doubles.
If the gating interval in too long then the counter will overflow and a FREQOVERRUN error will occur.

SigSource - CTRINPUTx refers to the external counter inputs on the board connector. All of the other choices refer to internal signals within the 9513 counter chip. Refer to the 9513 data sheet for details on these signal sources.

CIn.VI

Description: Reads the current count from a counter

Summary:



CIN.VI

Inputs: BoardNum - board number may be 0 to 9.
CounterNum - counter number to read

Outputs: Count - Count returned here
ErrCode - Error code

EXPLANATION OF THE ARGUMENTS:

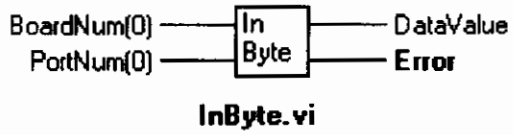
BoardNum - refers to the board number associated with the board when it was installed with the configuration program. The specified board must have a counter.

CounterNum - The counter to read current count from. Valid values are 1 to 10, up to the number of counters on the board.

1	Counter input to counter 1. CTR1
2 .. 10	Counter inputs 2 through 10

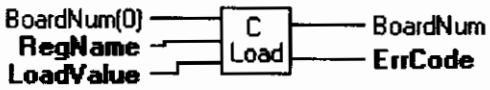
CLoad.VI

Description: Loads the specified counter's LOAD, HOLD or ALARM register with a count. When you want to load a counter with a value to count from, it is never loaded directly into the counter's count register. It is loaded into the load register. From there, the counter, once enabled, loads the count from the hold register, generally on the first valid pulse.



InByte.vi

Summary:



CLOAD.VI

Inputs: BoardNum - board number may be 0 to 9.
RegName - Register to load LoadValue into.
LoadValue - Value to be loaded into RegName

Outputs: BoardNum - Board number
ErrCode - Error code

EXPLANATION OF THE ARGUMENTS:

BoardNum - refers to the board number associated with the board when it was installed with the configuration program. The specified board must have a counter.

RegName - The counter to read current count from. Valid values are:

LOADREG1 .. 10	Load registers 1 through 10. This may span several chips.
HOLDREG1 .. 10	Hold registers 1 through 10. This may span several chips. (9513 only)
ALARMREG1CHIP1	Alarm register 1 of the first counter chip. (9513 only)
ALARMREG2CHIP1	Alarm register 2 of the first counter chip. (9513 only)
ALARMREG1CHIP2	Alarm register 1 of the second counter chip. (9513 only)
ALARMREG2CHIP2	Alarm register 2 of the second counter chip. (9513 only)

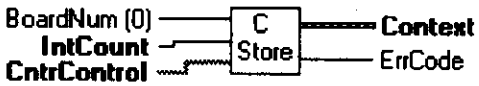
LoadValue - The value to be loaded. Must be between 0 and 2 ^ resolution - 1 of the counter. For example, a 16 bit counter is 2 ^ 16 - 1, or 65,535.

Counter Types: There are several counter types supported. Please refer to the data sheet for the registers available for a counter type.

CStore.VI

Description: Installs an interrupt handler that will store the current count whenever an interrupt occurs. This VI can only be used with 9513 counters. This VI will continue to operate in the background until either IntCount has been satisfied or StopBg.VI is called.

Summary:



CStore.VI

Inputs: BoardNum - board number may be 0 to 9.
IntCount - Number of interrupts
CntrControl - Array with each element set to either ENABLED or DISABLED

Outputs: Context - Output data structure
ErrCode -Error code

EXPLANATION OF THE ARGUMENTS:

BoardNum - refers to the board number associated with the board when it was installed with the configuration program. The specified board must have a 9513 counter.

IntCount - The counters will be read every time an interrupt occurs until IntCount interrupts have occurred. If IntCount is = 0 then the VI will run until StopBg.VI is called.

CntrControl - The array should have an element for each counter on the board. (5 elements for CTR-05 board, 10 elements for a CTR-10). Each element corresponds to a possible counter channel. Each element should be set to either DISABLED or ENABLED. All channels that are set to ENABLED will be read when an interrupt occurs.

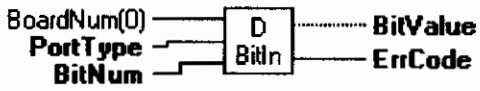
Context - Data structure containing information from a background operation. Some of the information included is the board number, the data array, the array size, and the initial status of the background operation.

NOTE:
Wiring of this VI should conform to the following pattern:
Start CStore.VI. GetStat.VI checks for completion(boolean output called "Running"). StopBg.VI terminates the operation, if not already done, and frees memory aliases. Data output from the background operation is passed to GetStat.VI and StopBg.VI via "Context", and can be wired from one or both of them for intermediate or final actions, respectively. The demo VI's illustrate this process effectively.

DBitIn.VI

Description: Reads the state of a single digital input bit. This VI treats all of the DIO ports on a boards as a single very large port. It lets you read the state of any individual bit within this large port. You must use DCfgPrt.VI to configure the port for input first.

Summary:



DBitIn.VI

Inputs: BoardNum - Board number may be 0 to 9.
PortType - Specifies which type of digital port to read
BitNum - Specifies which bit to read BitValue

Outputs: BitValue - Place holder for return value of bit - the bit's value (0 or 1) is returned here
ErrCode - Error code

EXPLANATION OF THE ARGUMENTS:

PortType- There are two general types of digital I/O - 8255 and other. Some boards (DIO Series) use an 8255 for digital I/O. For these boards PortType should be set to FIRSTPORTA. Other boards don't use 8255. For these boards PortType should be set to AUXPORT. Some boards have both types of digital I/O (DAS1600). Set PortNum to either FIRSTPORTA or AUXPORT depending on which digital inputs you wish to read.

BitNum - This specifies the bit number within the single large port. The specified bit must be in a port that is currently configured as an input.

The tables below show which bit numbers are in which 82C55 and 8536 digital chips. The most 82C55 chips on a single board is eight (8), on the CIO-DIO196. The most (2) 8536 chips occur on the CIO-INT32.

82C55 Bit#	Chip #	Address
0 - 23	1	Base + 0
24 - 47	2	Base + 4
48 - 71	3	Base + 8
72 - 96	4	Base + 12
96 - 119	5	Base + 16
120 - 143	6	Base + 20
144 - 167	7	Base + 24
168 - 191	8	Base + 28

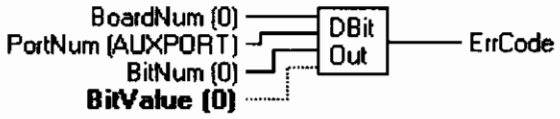
8536 Bit#	Chip #	Address
0 - 19	1	Base + 0
20 - 39	2	Base + 4

BitValue - Place holder for return value of bit. Value will be 0 or 1. A 0 indicates a low reading, a 1 indicates a logic high reading. Logic high does not necessarily mean 5V. See the board manual for chip input specifications.

DBitOut.VI

Description: Sets the state of a single digital output bit. This VI treats all of the DIO chips on a board as a single very large port. It lets you set the state of any individual bit within this large port. You must use DCfgprt.VI to configure the port for output first.

Summary:



DBitOut.VI

Inputs: BoardNum - board number may be 0 to 9.
PortNum - Specifies which digital port (AUXPORT, FIRSTPORTA).
BitNum - specifies which bit to write
BitValue - the bit's value (0 or 1)

Output: ErrCode - Error code

EXPLANATION OF THE ARGUMENTS:

PortType - There are two general types of digital I/O - 8255 and other. Some boards (DIO Series) use an 8255 for digital I/O. For these boards PortType should be set to FIRSTPORTA. Other boards don't use 8255. For these boards PortType should be set to AUXPORT. Some boards have both types of digital I/O (CIO-DAS1600). Set PortNum to either FIRSTPORTA or AUXPORT depending on which digital inputs you wish to write.

BitNum - This specifies the bit number within the single large port. The specified bit must be in a port that is currently configured as an output.

The tables below show which bit numbers are in which 82C55 and 8536 digital chips. The most 82C55 chips on a single board is eight (8), on the CIO-DIO196. The most (2) 8536 chips occur on the CIO-INT32.

82C55 Bit#	Chip #	Address
0 - 23	1	Base + 0
24 - 47	2	Base + 4
48 - 71	3	Base + 8
72 - 96	4	Base + 12
96 - 119	5	Base + 16
120 - 143	6	Base + 20
144 - 167	7	Base + 24
168 - 191	8	Base + 28

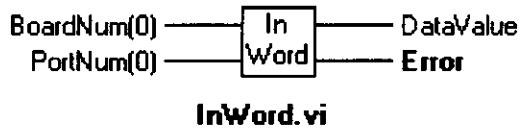
8536 Bit#	Chip #	Address
0 - 19	1	Base + 0
20 - 39	2	Base + 4

BitValue - The value to set the bit to. Value will be 0 or 1. A 0 indicates a logic low output, a 1 indicates a logic high output. Logic high does not necessarily mean 5V. See the board manual for chip specifications.

InByte.VI/ InWord.VI

Description: Reads a byte or word from a hardware register on a board.

Summary:



Inputs: BoardNum - board number may be 0 -9
PortNum - register on the board to read

Outputs: DataValue - Value read from port
Error - Error code

EXPLANATION OF THE ARGUMENTS:

BoardNum - refers to the board number associated with the board when it was installed with the configuration program.

PortNum - register within the board. Boards are set to a particular base address. The registers on the boards are at addresses that are offsets from the base address of the board (BaseAdr+0, BaseAdr+2, etc). This argument should be set to the offset for the desired register. This function takes care of adding the base address to the offset, so that the board's address can be changed without changing the code.

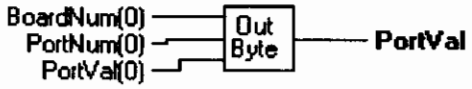
NOTES:
InByte.VI is used to read 8 bit ports. InWordVI is used to read 16 bit ports.

Returns: The current value of the specified register

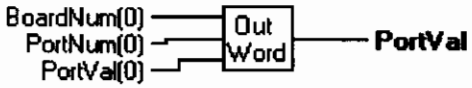
OutByte.VI / OutWord.VI

Description: Writes a byte or word to a hardware register on a board.

Summary:



OutByte.vi



OutWord.vi

Arguments: BoardNum - board number may be 0 -9
PortNum - register on the board to write to
PortVal - value to write to register

EXPLANATION OF THE ARGUMENTS:

BoardNum - refers to the board number associated with the board when it was installed with the configuration program.

PortNum - register within the board. Boards are set to a particular base address. The registers on the boards are at addresses that are offsets from the base address of the board (BaseAdr+0, BaseAdr+2, etc). This argument should be set to the offset for the desired register. This function takes care of adding the base address to the offset, so that the board's address can be changed without changing the code.

PortVal - Value that will be written to the register

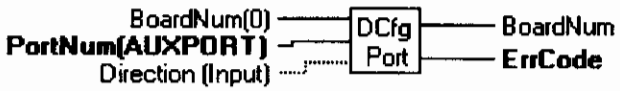
NOTES:
OutByte.VI is used to write to 8 bit ports. OutWord.VI is used to write to 16 bit ports.

Returns: error code or 0 if no errors

DCfgPrt.VI

Description: Configures a digital port as Input or Output. This mode is for use with 82C55 chips and 8536 chips. See the board user's manual for details of chip operation.

Summary:



DCfgPort.VI

Inputs: BoardNum - board number may be 0 to 9.
PortNum - Specifies which digital I/O port to configure.
Direction - DIGITALOUT or DIGITALIN

Outputs: BoardNum - Board number
ErrCode - Error code

EXPLANATION OF THE ARGUMENTS:

PortNum - The specified port must be configurable. The AUXPORT is not configurable. It is always configured for inputs and outputs.

The tables below show which ports and bit numbers are in which 82C55 and 8536 digital chips. The most 82C55 chips on a single board is eight (8), on the CIO-DIO196. The most (2) 8536 chips occur on the CIO-INT32.

Mnemonic	Bit#	8255 Chip #	Chip Address	8536 Chip #	Chip Address
FIRSTPORTA	0 - 7	1A	Base + 0	1A	Base + 0
FIRSTPORTB	8 - 15	1B		1B	
FIRSTPORTCL	16 - 19	1CL		1C	
FIRSTPORTCH	20 - 23	1CH		Not present	
SECONDPORTA	24 - 31	2A	Base + 4	2A	Base + 4
SECONDPORTB	32 - 39	2B		2B	
SECONDPORTCL	40 - 43	2CL		2C	
SECONDPORTCH	44 - 47	2CH		No port C High in 8536 chips	

and so on to the last chip on the board as: THIRDPORT@, FOURTHPORT@, FIFTHPORT@, SIXTHPORT@, SEVENTHPORT@ and

EIGHTHPORTA	168 - 175	8A	Base + 28
EIGHTHPORTB	176 - 183	8B	
EIGHTHPORTCL	184 - 187	8CL	
EIGHTHPORTCH	188 - 191	8CH	

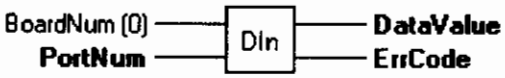
Direction - DIGITALOUT or DIGITALIN configures entire eight or four bit port for output or input.

Returns: Error code or 0 if no errors

DIn.VI

Description: Reads a digital input port

Summary:



DIn.VI

Inputs: BoardNum - board number may be 0 to 9.
PortNum - Specifies which digital I/O port to read.

Outputs: DataValue - Digital input value.
ErrCode - Error code

EXPLANATION OF THE ARGUMENTS:

PortNum - The specified port must be configured for input. The AUXPORT is not configurable.

The tables below show which ports are in which 82C55 and 8536 digital chips. The most 82C55 chips on a single board is eight (8), on the CIO-DIO196. The most (2) 8536 chips occur on the CIO-INT32.

Mnemonic	8255 Chip #	Chip Address	8536 Chip #	Chip Address
FIRSTPORTA	1A	Base + 0	1A	Base + 0
FIRSTPORTB	1B		1B	
FIRSTPORTCL	1CL		1C	
FIRSTPORTCH	1CH		Not present	
SECONDPORTA	2A	Base + 4	2A	Base + 4
SECONDPORTB	2B		2B	
SECONDPORTCL	2CL		2C	
SECONDPORTCH	2CH		No port C High in 8536 chips	

and so on to the last chip on the board as: THIRDPORT@, FOURTHPORT@, FIFTHPORT@, SIXTHPORT@, SEVENTHPORT@ and

EIGHTHPORTA	8A	Base + 28
EIGHTHPORTB	8B	
EIGHTHPORTCL	8CL	
EIGHTHPORTCH	8CH	

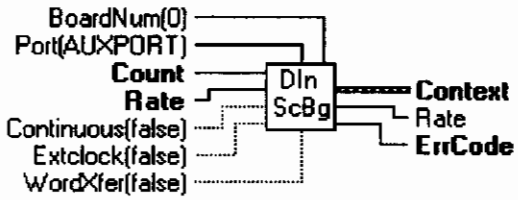
The size of the ports vary. If it is an eight bit port then the returned value will be in the range 0- 255. If it is a four bit port the value will be in the range 0 -15.

DInScBg.VI

Description: Multiple reads of digital input port of a high speed digital port on a board with a pacer clock. As of this revision of the manual and software, that is the CIO-PDMA16 only.

When this VI is used, control will return immediately to the next point in your program and the transfer from the digital input port to the array in the Context will continue in the background. Use GetStat.VI to check on the status of the background operation. Use StopBg.VI to terminate the background process before it has completed.

Summary:



DINSCBG.VI

Inputs: BoardNum - board number may be 0 to 9.
Port - Specifies which digital I/O port to read
Count - number of times to read digital input
Rate - Number of times per second (Hz) to read
Continuous - Continuous or single
Extclock - External or internal clock
WordXfer - Word or Byte transfer

Outputs: Context - Output data structure ***NOTE**
Rate - Actual rate returned here
ErrCode - Error code

EXPLANATION OF THE ARGUMENTS:

PortNum - Specifies which digital I/O port to read. The two choices are FIRSTPORTA or FIRSTPORTB. The specified port must be configured as an input.

Count - The number of times to read digital input

Rate - Number of times per second (Hz) to read the port. The actual sampling rate in some cases will vary a small amount from the requested rate. The actual rate will be returned to the Rate argument.

Continuous - This option puts the VI in an endless loop. Once it transfers the required number of bytes it resets to the start of the input array and begins again. The only way to stop this operation is with StopBg.VI.

ExtClock - If this option is used then transfers will be controlled by the signal on the trigger input line rather then by the internal pacer clock. Each transfer will be triggered on the rising edge of the trigger input signal. When this option is used the Rate argument is ignored. The transfer rate is dependent on the trigger signal.

WordXfer - Normally this VI reads a single (byte) port. If WORDXFER is specified then it will read two adjacent ports on each read and store the value of both ports together as the low and high byte of a single array element in the input array.

Transfer Method - May not be specified. DMA is used.

Context - Data structure containing information from a background operation. Some of the information included is the board number, the data array, the array size, and the initial status of the background operation.

NOTE:

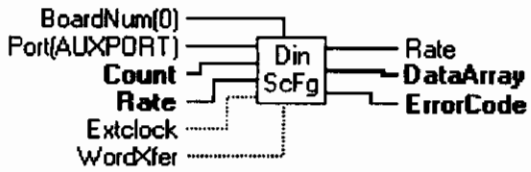
Wiring of this VI should conform to the following pattern:

DInScBg.VI starts a background operation. GetStat.VI checks for completion(boolean output called "Running"). StopBg.VI terminates the operation, if not already done, and frees memory aliases. Data output from the background operation is passed to GetStat.VI and StopBg.VI via "Context", and can be wired from one or both of them for intermediate or final actions, respectively. The demo VI's illustrate this process effectively.

DInScFg.VI

Description: Multiple reads of digital input port of a high speed digital port on a board with a pacer clock. As of this revision of the manual and software, that is the CIO-PDMA16 only. The cDInScFg VI will not return to your program until all of the requested data has been collected and returned to input array.

Summary:



DInScFg.VI

Inputs: BoardNum - board number may be 0 to 9.
Port - Specifies which digital I/O port to read
Count - number of times to read digital input
Rate - Number of times per second (Hz) to read
Extclock - External or internal clock
WordXfer - Word or Byte transfer

Outputs: Rate - Actual rate returned here
DataArray - Data from scan is returned here
ErrCode - Error code

EXPLANATION OF THE ARGUMENTS:

PortNum - Specifies which digital I/O port to read. The two choices are FIRSTPORTA or FIRSTPORTB. The specified port must be configured as an input.

Count - The number of times to read digital input

Rate - Number of times per second (Hz) to read the port. The actual sampling rate in some cases will vary a small amount from the requested rate. The actual rate will be returned to the Rate argument.

DataArray - Data from the scan is returned here.

EXTCLOCK - If this option is used then transfers will be controlled by the signal on the trigger input line rather then by the internal pacer clock. Each transfer will be triggered on the rising edge of the trigger input signal. When this option is used the Rate argument is ignored. The transfer rate is dependent on the trigger signal.

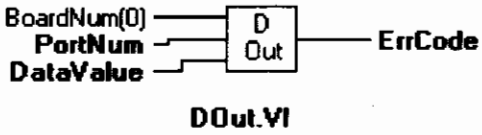
WORDXFER - Normally this VI reads a single (byte) port. If WORDXFER is specified then it will read two adjacent ports on each read and store the value of both ports together as the low and high byte of a single array element in DataArray[].

Transfer Method - May not be specified. DMA is used.

DOut.VI

Description: Writes a byte to a digital output port .

Summary:



DOut.VI

Inputs: BoardNum - board number may be 0 to 9.
PortNum - Specifies which digital I/O port to read.
DataValue - Digital output value input here.

Output: ErrCode - Error code

EXPLANATION OF THE ARGUMENTS:

PortNum - The specified port must be configured for output. The AUXPORT is not configurable.

The tables below show which ports are in which 82C55 and 8536 digital chips. The most 82C55 chips on a single board is eight (8), on the CIO-DIO196. The most (2) 8536 chips occur on the CIO-INT32.

Mnemonic	8255 Chip #	Chip Address	8536 Chip #	Chip Address
FIRSTPORTA	1A	Base + 0	1A	Base + 0
FIRSTPORTB	1B		1B	
FIRSTPORTCL	1CL		1C	
FIRSTPORTCH	1CH		Not present	
SECONDPORTA	2A	Base + 4	2A	Base + 4
SECONDPORTB	2B		2B	
SECONDPORTCL	2CL		2C	
SECONDPORTCH	2CH		No port C High in 8536 chips	

and so on to the last chip on the board as: THIRDPORT@, FOURTHPORT@, FIFTHPORT@, SIXTHPORT@, SEVENTHPORT@ and

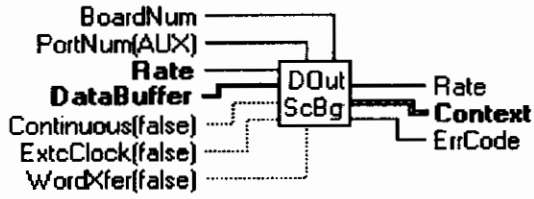
EIGHTHPORTA	8A	Base + 28
EIGHTHPORTB	8B	
EIGHTHPORTCL	8CL	
EIGHTHPORTCH	8CH	

The size of the ports varies. If it is an eight bit port then the output value should be in the range 0- 255. If it is a four bit port the value should be in the range 0 - 15.

DOutScBg.VI

Description: Multiple writes to digital output port of a high speed digital port on a board with a pacer clock. As of this revision of the manual and software, that is the CIO-PDMA16 only. When this VI used, control will return immediately to the next point in your program and the transfer to the digital output port from DataBuffer will continue in the background. Use GetStat.VI to check on the status of the background operation. Use StopBg.VI to terminate the background process before it has completed.

Summary:



DOutScBg.VI

Inputs: BoardNum - board number may be 0 to 9.
PortNum - Specifies which digital I/O port to write
Rate - Number of times per second (Hz) to write
DataBuffer - Digital output values.
Extclock - Extternal or internal clock
WordXfer - Word or byte transfer

Outputs: Rate - Actual scan rate
ErrCode - Error code

EXPLANATION OF THE ARGUMENTS:

PortNum - Specifies which digital I/O port to read. The two choices are FIRSTPORTA or FIRSTPORTB. The specified port must be configured as an output.

Rate - Number of times per second (Hz) to write to the port. The actual update rate in some cases will vary a small amount from the requested rate. The actual rate will be returned to the Rate argument.

DataBuffer - Data to the scan is input here

Continuous - This option puts the VI in an endless loop. Once it transfers the required number of bytes it resets to the start of DataBuffer and begins again. The only way to stop this operation is with StopBg.VI.

ExtClock - If this option is used then transfers will be controlled by the signal on the trigger input line rather than by the internal pacer clock. Each transfer will be triggered on the rising edge of the trigger input signal. When this option is used the Rate argument is ignored. The transfer rate is dependent on the trigger signal.

WordXfer - Normally this VI reads a single (byte) port. If WORDXFER is specified then it will write two adjacent ports as the low and high byte of a single array element in DataBuffer[].

Transfer Method - May not be specified. DMA is used.

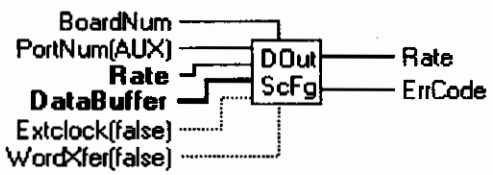
Context - Data structure containing information from a background operation. Some of the information included is the board number, the data array, the array size, and the initial status of the background operation.

NOTE:
Wiring of this VI should conform to the following pattern:
DOutScBg.VI starts a background operation. GetStat.VI checks for completion(boolean output called "Running"). StopBg.VI terminates the operation, if not already done, and frees memory aliases. Data output from the background operation is passed to GetStat.VI and StopBg.VI via "Context", and can be wired from one or both of them for intermediate or final actions, respectively. The demo VI's illustrate this process effectively.

DOutScFg.VI

Description: Multiple writes to digital output port of a high speed digital port on a board with a pacer clock. As of this revision of the manual and software, that is the CIO-PDMA16 only. The DOutScFg.VI will not return to your program until all of the requested data has been output.

Summary:



DOutScFg.VI

Inputs: BoardNum - board number may be 0 to 9.
PortNum - Specifies which digital I/O port to write
Rate - Number of times per second (Hz) to write
DataBuffer - Digital output values
Extclock - External or internal clock
WordXfer - Word or byte transfer

Outputs: Rate - Actual rate is returned here
ErrCode - Error code

EXPLANATION OF THE ARGUMENTS:

PortNum - Specifies which digital I/O port to read. The two choices are FIRSTPORTA or FIRSTPORTB. The specified port must be configured as an output.

Rate - Number of times per second (Hz) to write to the port. The actual update rate in some cases will vary a small amount from the requested rate. The actual rate will be returned to the Rate argument.

DataBuffer - Data to the scan is input here

ExtClock - If this option is used then transfers will be controlled by the signal on the trigger input line rather than by the internal pacer clock. Each transfer will be triggered on the rising edge of the trigger input signal. When this option is used the Rate argument is ignored. The transfer rate is dependent on the trigger signal.

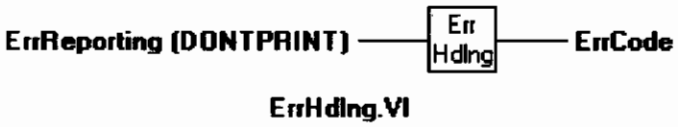
WordXfer - Normally this VI reads a single (byte) port. If WORDXFER is specified then it will write two adjacent ports as the low and high byte of a single array element in DataBuffer[].

Transfer Method - May not be specified. DMA is used.

ErrHdlng.VI

Description: Sets the error handling for all subsequent VI calls. All VI's return error codes after each call. In addition other error handling features have been built into the library. This VI controls those features. If the LabVIEW Extension cannot find the configuration file CB.CFG, it always terminates the program regardless of the ErrHdlng setting.

Summary:



Input: ErrReporting - type of error reporting.

Output: ErrCode - Error code

EXPLANATION OF THE ARGUMENTS:

Warnings vs Fatal Errors - All errors that can occur are classified as either "warnings" or "fatal". Errors that can occur in normal operation in a bug free program (disk is full, too few samples before trigger occurred) are classified as "warnings". All other errors indicate a more serious problem and are classified as "fatal".

ErrReporting - This argument controls when the library will print error messages on the screen. The default is DONTPRINT If it is set to:

DONTPRINT - Errors will not generate a message to the screen. In that case your program must always check the re turned error code after each library call to determine if an error occurred.

PRINTWARNINGS - Only warning errors will generate a message to the screen. Your program will have to check for fatal errors.

PRINTFATAL - Only fatal errors will generate a message to the screen. Your program must check for warning errors.

PRINTALL - All errors will generate a message to the screen.

ErrHandling - This argument specifies what class of error will cause the program to halt. The default is DONTSTOP. If ErrReporting is set to:

DONTSTOP - The program will always continue executing when an error occurs.

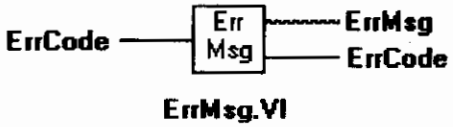
STOPFATAL - The program will halt if a "fatal" error occurs.

STOPALL - Will stop whenever any error occurs.

ErrMsg.VI

Description: Returns the error message associated with an error code. Each VI returns an error code. If the error code is not equal to 0 it indicates that an error occurred. Call this VI to convert the returned error code to a descriptive error message.

Summary:



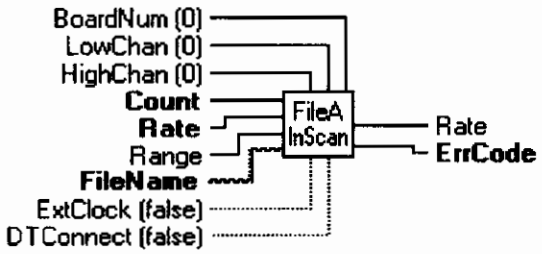
Input: ErrCode - error code that was returned by any VI.

Outputs: ErrMsg - error message returned here
ErrCode - error code that was returned by any VI.

FileAInS.VI

Description: Scan a range of A/D channels and store the samples in a disk file. This VI reads the specified number of A/D samples at the specified sampling rate from the specified range of A/D channels from the specified board. If the A/D board has programmable gain then it sets the gain to the specified range. The collected data is returned to a file in binary format. Use FileRead.VI to load data from that file into an array. Filexxx.VI's are only for boards with REP-INSW transfer capability.

Summary:



FileAInS.VI

Inputs: BoardNum - board number may be 0 to 9.
LowChan - First A/D channel of scan
HighChan - Last A/D channel of scan
Count - Number of samples to collect
Rate - Sample rate in samples per second (Hz)
Range - Range code
FileName - Name of disk file
Extclock - External or internal clock
DTConnect - DT connect option

Outputs: Rate - Actual sampling rate
ErrCode - Error code

EXPLANATION OF THE ARGUMENTS:

BoardNum - refers to the board number associated with the board when it was installed with the configuration program. The specified board must have an A/D.

Channel - The maximum allowable channel depends on which type of A/D board is being used. For boards that have both single ended and differential inputs the maximum allowable channel number also depends on how the board is configured (8 chans for differential, 16 for single ended).

Count - Specifies the total number of A/D samples that will be collected. If more than one channel is being sampled then the number of samples collected per channel is equal to Count / (HighChanLowChan+1).

Rate - The maximum sampling rate depends on the A/D board that is being used.

Range - If the selected A/D board does not have a programmable range feature, then this argument should be set to NOT PROGRAMMABLE. Otherwise the gain can be set to any of the following ranges that are supported by the selected A/D board. Refer to board specific information for the list of ranges supported by each board.

BIP10VOLTS	+/- 10 volts	UNI10VOLTS	0 - 10 volts
BIP5VOLTS	+/- 5 volts	UNI5VOLTS	0 - 5 volts
BIP2PT5VOLTS	+/- 2.5 volts	UNI2PT5VOLTS	0 - 2.5 volts
BIP1PT67VOLTS	+/- 1.67 volts	UNI2VOLTS	0 - 2 volts
BIP1PT25VOLTS	+/- 1.25 volts	UNI1PT67VOLTS	0 - 1.67 volts
BIP1VOLTS	+/- 1 volts	UNI1PT25VOLTS	0 - 1.25 volts
BIPPT625VOLTS	+/- 0.625 volts	UNI1VOLTS	0 - 1 volts
BIPPT5VOLTS	+/- 0.5 volts	UNIPT1VOLTS	0 - 0.1 volts

BIPPT1VOLTS	+/- 0.1 volts	UNIPT01VOLTS	0 - 0.01 volts
BIPPT05VOLTS	+/-0.05 volts	MA4TO20	4 - 20 mA
BIPPT01VOLTS	+/- 001 volts	MA2TO10	2 - 10 mA
BIPPT005VOLTS	+/- 0.005 volts	MA1TO5	1 - 5 mA
		MAPT5TO2PT5	0.5 - 2.5 mA

FileName - The name file must already exist. It should have been previously created with the MAKESTRM.EXE program.

ExtClock - If this option is used then conversions will be controlled by the signal on the trigger input line rather than by the internal pacer clock. Each conversion will be triggered on the rising edge of the trigger input signal. When this option is used the Rate argument is ignored. The sampling rate is dependent on the trigger signal.

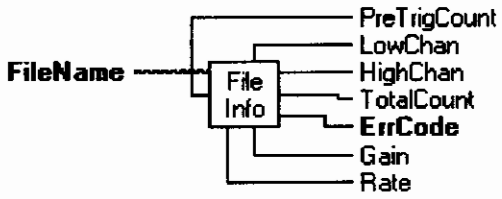
DTConnect - If TRUE, samples are sent to the DT-Connect port if the board is equipped with one. If FALSE samples are not output to the DT-Connect port. This is the default.

OVERRUN Error - This error indicates that the data was not written to the file as fast as the data was sampled. Consequently some data was lost.

FileInfo.VI

Description: Returns information about a streamer file. When FileAInS.VI or FilePret.VI fill the streamer file, information is stored about how the data was collected (sample rate, channels sampled etc.). This VI returns that information. Filexxx VI's are only for boards with REP-INSW transfer capability.

Summary:



FileInfo.VI

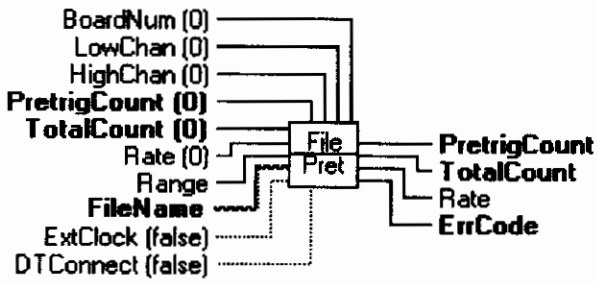
Inputs: FileName - Name of streamer file

Outputs: PreTrigCount - Number of pre-trigger points collected
LowChan - Low A/D channel of scan
HighChan - High A/D channel of scan
TotalCount - Total number of points collected
ErrCode - Error code
Gain - Range of A/D when data was collected
Rate - Sampling rate when data was collected

FilePret.VI

Description: Scan a range of channels continuously while waiting for a trigger. Once the trigger occurs, return the specified number of samples including the specified number of pre-trigger samples to a disk file. This VI waits for a trigger signal to occur on the Trigger Input. Once the trigger occurs, it returns the specified number (TotalCount) of A/D samples including the specified number of pre trigger points. It collects the data at the specified sampling rate (Rate) from the specified range (LowChan-HighChan) of A/D channels from the specified board. If the A/D board has programmable gain then it sets the gain to the specified range. The collected data is returned to a file. Filexxx.VI's are only for boards with REP-INSW transfer capability.

Summary:



FilePret.VI

Inputs:
BoardNum - board number may be 0 to 9.
LowChan - First A/D channel of scan
HighChan - Last A/D channel of scan
PretrigCount - Number of pre-trigger samples to collect.
TotalCount - Total number of samples to collect.
Rate - Sample rate in samples per second (Hz)
Range - A/D Range or 0
FileName - Name of disk file
ExtClock - External or internal clock
DTConnect - DT connect option

Outputs:
PretrigCount - Number of pre-trigger sample collected.
TotalCount - Total number of samples collected.
Rate - Actual sampling rate
ErrCode - error code

EXPLANATION OF THE ARGUMENTS:

BoardNum - refers to the board number associated with the board when it was installed with the configuration program. The specified board must have an A/D and pretrigger capability.

Channel - The maximum allowable channel depends on which type of A/D board is being used. For boards that have both single ended and differential inputs the maximum allowable channel number also depends on how the board is configured (8 chans for differential, 16 for single ended).

PretrigCount - Specifies the number of samples before the trigger that will be returned. PretrigCount must be less than 16000 and PreTrigCount must also be less than TotalCount-512.

If the trigger occurs too early, then fewer than the requested number of pre-trigger samples will be collected. In that case a TOOFEW error will occur. The PretrigCount will be set to indicate how many samples were collected and the post trigger samples will still be collected.

TotalCount - Specifies the total number of samples that will be collected and stored in the file. TotalCount must be greater then or equal to PretrigCount+512. If the trigger occurs too early then fewer than the requested number of samples will be collected. In that case a TOOFEW error will occur. The TotalCount will be set to indicate how many samples were actually collected.

Rate - The maximum sampling rate depends on the A/D board that is being used.

Range - If the selected A/D board does not have a programmable range feature, then this argument should be set to 0. Otherwise the gain can be set to any of the following ranges that are supported by the selected A/D board. Refer to board specific information for the list of ranges supported by each board.

BIP10VOLTS	+/- 10 volts	UNI10VOLTS	0 - 10 volts
BIP5VOLTS	+/- 5 volts	UNI5VOLTS	0 - 5 volts
BIP2PT5VOLTS	+/- 2.5 volts	UNI2PT5VOLTS	0 - 2.5 volts
BIP1PT67VOLTS	+/- 1.67 volts	UNI2VOLTS	0 - 2 volts
BIP1PT25VOLTS	+/- 1.25 volts	UNI1PT67VOLTS	0 - 1.67 volts
BIP1VOLTS	+/- 1 volts	UNI1PT25VOLTS	0 - 1.25 volts
BIPPT625VOLTS	+/- 0.625 volts	UNI1VOLTS	0 - 1 volts
BIPPT5VOLTS	+/- 0.5 volts	UNIPT1VOLTS	0 - 0.1 volts
BIPPT1VOLTS	+/- 0.1 volts	UNIPT01VOLTS	0 - 0.01 volts
BIPPT05VOLTS	+/-0.05 volts	MA4TO20	4 - 20 mA
BIPPT01VOLTS	+/- 001 volts	MA2TO10	2 - 10 mA
BIPPT005VOLTS	+/- 0.005 volts	MA1TO5	1 - 5 mA
		MAPT5TO2PT5	0.5 - 2.5 mA

FileName - The named file must already exist. It should have been previously created with the MAKESTRM.EXE program.

EXTCLOCK - If this option is used then conversions will be controlled by the signal on the trigger input line rather than by the internal pacer clock. Each conversion will be triggered on the rising edge of the trigger input signal. When this option is used the Rate argument is ignored. The sampling rate is dependent on the trigger signal.

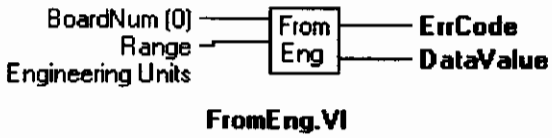
DTCONNECT - If TRUE, samples are sent to the DT-Connect port if the board is equipped with one. If FALSE, samples are not output to the DT-Connect port. This is the default.

OVERRUN Error - This error indicates that the data was not written to the file as fast as the data was sampled. Consequently some data was lost. The value in *Count will be the number of points that were successfully collected.

FromEng.VI

Description: Converts a voltage (or current) in engineering units to a D/A count value for uotput to a D/A.

Summary:



Arguments: BoardNum - Board number, may be 0-9
Range - D/A range to use in conversion
EngUnits - Voltage (or current) value to convert
DataVal - D/A count equivalent to voltage returned here
ErrCode - error code

EXPLANATION OF THE ARGUMENTS:

BoardNum - refers to the board number associated with the D/A board when it was installed. This function uses the board number to determine whether to do a 12 bit or 16 bit conversion.

Range - D/A voltage (or current) range. Some D/A boards have programmable voltage ranges, others set the voltage range via switches on the board. In either case the selected range must be passed to this function. Each D/A board supports different voltage and/or current ranges. Refer to the board's hardware manual for a list of allowed ranges used by the board.

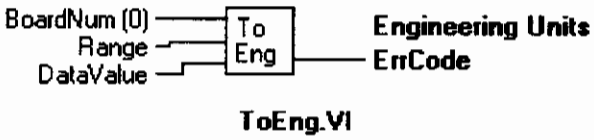
EngUnits - The voltage (or current) value that you wish to set the D/A to. This value should be within the range specified by the Range argument.

DataVal - The function returns a D/A count to this variable that is equivalent to the EngUnits argument.

ToEng.VI

Description: Converts an A/D count value to an equivalent voltage value.

Summary:



Arguments: BoardNum - Board number, may be 0-9
Range - A/D range to use in conversion
DataVal - A/D count value returned from an A/D board
EngUnits - Equivalent voltage (or current) value returned to this variable
ErrCode - error code

EXPLANATION OF THE ARGUMENTS:

BoardNum - refers to the board number associated with the A/D board when it was installed. This function uses the board number to determine whether to do a 12 bit or 16 bit conversion.

Range - A/D voltage (or current) range. Some A/D boards have programmable voltage ranges, others set the voltage range via switches on the board. In either case the selected range must be passed to this function. Each A/D board supports different voltage and/or current ranges. Refer to the board's hardware manual for a list of allowed ranges used by the board.

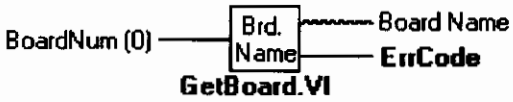
DataVal - A/D count returned from an A/D board.

EngUnits - The voltage (or current) value that is quivalent to DataVal is returned to this variable. The value will be within the range specified by the Range argument.

GetBoard.VI

Description: Returns the boardname of a specified board.

Summary:



Arguments: BoardNum - Board number may be 0 to 9 or -2 for GETFIRST or -3 for GETNEXT
BoardName - Board name string returned to this variable
ErrCode - error code

EXPLANATION OF THE ARGUMENTS

Board - refers to the board number associated with a board when it was installed.

BoardName - A string variable that the board name will be returned to. This string variable must be pre-allocated to be at least as large as BOARDNAMELEN. This size is guaranteed to be large enough to hold the longest board name string.

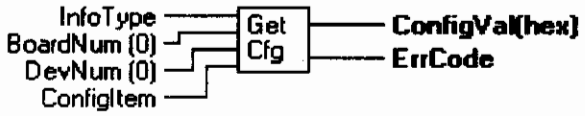
NOTES:
There are two distinct ways of using this function. The first is to pass a board number as the Board argument. In that case the string that is returned will describe the board type of the installed board.

The other way to use the function is to set Board to GETFIRST(-2) or GETNEXT(-3) to get a list of all board types that are supported by the library. If Board is set to -2 it will return the first board type in the list of supported boards. Subsequent calls with Board=-3 will return each of the other board types supported by the library. When you reach the end of the list BoardName will be set to an empty string. Refer to the ulgt04 example programs for more details.

GetCfg.VI

Description: Returns a configuration option for a board. The configuration information for all boards is stored in the CB.CFG file. This information is loaded from CB.CFG by all programs that use the library. The current configuration can be changed within a running program with the cbSetConfig() function. This function returns the current configuration information.

Summary



GetCfg.VI

Arguments: InfoType - Which class of configuration information you want to retreive
BoardNum - Board name (0-9)
DevNum - Specifies which device within board
ConfigItem - Sepcifies which configuration item
ConfigVal - Current configuration value returned here
ErrCode - error code

EXPLANATION OF THE ARGUMENTS

InfoType - The configuration information for each board is grouped into different categories. This argument specifies which category you want. It should be set to one of the following constants:

GLOBALINFO - Information about the configuration file
BOARDINFO - general information about a board
DIGITALINFO - information about a digital device
COUNTERINFO - information about a counter device
EXPANSIONINFO - information about an expansion device
MISCINFO - One of the miscellaneous options for the board

BoardNum - refers to the board number associated with a board when it was installed.

DevNum - Selects a partilcuar device. If InfoType=DIGITALINFO then DevNum specifies which of the board's digital devices you want information on. If InfoType=COUNTERINFO then DevNum specifies which of the board's counter device.

ConfigItem - Specifies which configuration item you wish to retreive. Refer to the table below for a lsit of all of the possible values for ConfigItem.

ConfigVal - The specified configuration item is returned to this variable.

NOTES:
The list of ConfigItem values for each category of configuration information is:

InfoType = GLOBALINFO

GIVERSION - CB.CFG file format. This information is used by the library to determine compatibility.
GINUMBOARDS - Maximum number of installable boards
GINUMEXPBOARDS - Maximim number of expansions boards allowed to be installed.

InfoType = BOARDINFO

BIBASEADR - Base address of board
BIBOARDTYPE - Returns a number in the range of 100 to 8000 Hex. Determines board series.

Series	No.	Series	No.	Series	No.	Series	No.
DAS16	100	SSH	200	EXP	300	DIO	400
PDMA	500	DAC	600	DDA	700	CTR	800
PDI	900	SBX	A00	PPIO	B00	DAS08	C00
DAS1600	E00	MEM	F00	RELAY	1000	INT	2000
PCM	4000	COM	5000	DAS800	6000	DMM	7000

BIINTLEVEL - Interrupt level. 0 for none or 1 - 15
BIDMACHAN - DMA channel. 0, 1 or 3
BIINITIALIZED - TRUE (-1) or FALSE (0)
BICLOCK - Clock freq in MHz (1 or 10) or 0 for not supported.
BIRANGE - Selected volatge range. For switch selectable gains only.

BIRANGE: If the selected A/D board does not have a programmable gain feature then this argument returns the range as defined by the installed InstaCal settings, which, if InstaCal and the board were installed correctly, corresponds to the input range as set via the switches on the board. Refer to board specific information for a list of the A/D ranges supported by each board.

Library Name	Range	BIRANGE #	Library Name	Range	BIRANGE #
BIP10VOLTS	+/- 10 volts	1	UNI10VOLTS	0 - 10 volts	100
BIP5VOLTS	+/- 5 volts	0	UNI5VOLTS	0 - 5 volts	101
BIP2PT5VOLTS	+/- 2.5 volts	2	UNI2PT5VOLTS	0 - 2.5 volts	102
BIP1PT25VOLTS	+/- 1.25 volts	3	UNI2VOLTS	0 - 2 volts	103
BIP1VOLTS	+/- 1 volts	4	UNI1PT25VOLTS	0 - 1.25 volts	104
BIPPT625VOLTS	+/- 0.625 volts	5	UNI1VOLTS	0 - 1 volts	105
BIPPT5VOLTS	+/- 0.5 volts	6	UNIPT1VOLTS	0 - 0.1 volts	106
BIPPT1VOLTS	+/- 0.1 volts	7	UNIPT01VOLTS	0 - 0.01 volts	107
BIPPT05VOLTS	+/-0.05 volts	8	UNI1PT67VOLTS	0 - 1.67 volts	108
BIPPT01VOLTS	+/- 001 volts	9	MA4TO20	4 - 20 mA	200
BIPPT005VOLTS	+/- 0.005 volts	10	MA2TO10	2 - 10 mA	201
BIP1PT67VOLTS	+/-1.67 volts	11	MA1TO5	1 - 5 mA	202
			MAPT5TO2PT5	0.5 to 2.5mA	203

BINUMADCHANS - Number of A/D channels
BIUSESEXPS - Supports expansion boards TRUE/FALSE
BIDINUMDEVS - Number of digital devices
BIDIDEVNUM - Index into digital information for first device
BICINUMDEVS - Number of counter devices
BICIDEVNUM - Index into counter information for first device
BINUMDACHANS - Number of D/A channels
BIWAITSTATE - Setting of Wait State jumper. 1 = enabled, 0 = disabled
BINUMIOPORTS - Number of IOPorts used by board
BIPARENTBOARD - Board number of parent board
BIDTBOARD - Board number of connected DT board

InfoType = DIGITALINFO
DIBASEADR - Base address
DIINITIALIZED - TRUE (-1) or FALSE (0)
DIDEVTYPE - Device Type - AUXPORT, FIRSTPORTA etc
DIMASK - Bit mask for this port
DIREADWRITE - Read required before write TRUE/FALSE
DICONFIG - Current configuration INPUT or OUTPUT
DINUMBITS - Number of bits in port
DICURVAL - Current value of outputs

InfoType = COUNTERINFO
CIBASEADR - Base address
CIINITIALIZED - TRUE (-1) or FALSE (0)
CICTRTYPE - 1 = 8254, 2 = 9513 , 3 = 8536 type counter chip.
CICTRNUM - Which counter on chip
CICONFIGBYTE - Configuration byte

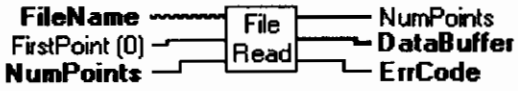
InfoType = EXPANSIONINFO
XIBOARDTYPE - Board type
XIMUXADCHAN1 - A/D channel board is connect to
XIMUXADCHAN2 - 2nd A/D channel board is connected to

XIRANGE1 - Range (gain) of low 16 chans
XIRANGE2 - Range (gain) of high 16 chans
XICJCCHAN - A/D channel that CJC is connected to
XITHERMTYPE - Thermocouple type
XINUMEXPCHANS - Number of expansion channels on board
XIPARENTBOARD - Board number of parent A/D board

FileRead.VI

Description: Reads data from a streamer file. Filexxx.VI's are only for boards with REP-INSW transfer capability.

Summary:



FileRead.VI

Inputs: FileName - Name of streamer file
FirstPoint - Index of first point to read
NumPoints - Number of points to read

Outputs: NumPoints - Number of points read
DataBuffer - Data buffer that data was read into.
ErrCode - error code

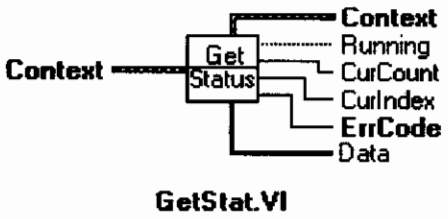
Data Format - The data is returned as 16 bits. The 16 bits may represent 12 bits of analog plus 4 bits of channel, or 16 bits of analog. Use ACvtData.VI to correctly load the data into an array.

Loading Portions of Files - The file may contain much more data than can fit in DataBuffer. In those cases use TotalCount and FirstPoint to read a selected piece of the file into DataBuffer. Call FileInfo.VI first to find out how many points are in the file.

GetStat.VI

Description: Returns status about background operation currently running

Summary:



GetStat.VI

Inputs: Context - Input data structure from a background operation. ***NOTE**

Outputs: Context - Output data structure
Running - Status of background operation.
CurCount - current count returned to this variable.
CurIndex - current index returned to this variable.
ErrCode -Error code
Data - Data array from context

EXPLANATION OF THE ARGUMENTS:

Context - Data structure containing information from a background operation. Some of the information included is the board number, the data array, the array size, and the initial status of the background operation.

NOTE:
Wiring of this VI should conform to the following pattern:
Start a background operation. GetStat.VI checks for completion(boolean output called "Running"). StopBg.VI terminates the operation, if not already done, and frees memory aliases. Data output from the background operation is passed to GetStat.VI and StopBg.VI via "Context", and can be wired from one or both of them for intermediate or final actions, respectively. The demo VI's illustrate this process effectively.

Running - Indicates whether or not a background process is currently executing.

CurCount - Specifies how many points have been input or output. It can be used to gauge how far along the operation is towards completion.

CurIndex - This is an index into the data buffer that points at the start of the last completed channel scan. This can be used to provide a real- time display for a background operation. DataBuffer[CurIndex] points to the start of the last complete channel scan that was put in or taken out of the buffer. If no points in the buffer have been accessed yet then CurIndex will equal -1.

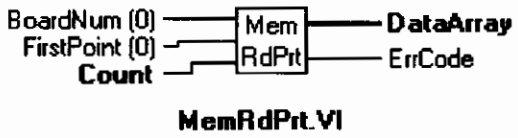
If you use the CONVERTDATA option with either the CONTINUOUS option or with pre-triggering VI's then CBIndex will return the index of the last A/D sample, rather than the start of the last completed channel scan.

For many background operations CurCount=CurIndex. For Pre-Trigger inputs though they are different. CurCount indicates how many points of the TotalCount have been collected. CurCount will rise to PreTrigCount and then stop until the trigger occurs. CurIndex though will constantly increase and reset as it goes around and around the circular buffer while waiting for the trigger to occur.

MemRdPrt.VI

Description: Reads pre-trigger data from a memory board that has been collected with the APretrxx.VI. This VI can only be used to retrieve data that has been collected with the APretrxx.VI. After each APretrxx call, all data must be unloaded from the memory board with this VI. If any more data is sent to the memory board then the pre-trigger data will be lost.

Summary:



MemRdPrt.VI

Inputs: BoardNum - board number may be 0 to 9.
FirstPoint - Index of first point to read or FROMHERE.
Count - Number of points (words) to read

Outputs: DataArray - Output data array
ErrCode - Error code

EXPLANATION OF THE ARGUMENTS:
FirstPoint - Use the FirstPoint argument to specify the first point to be read. For example, to read points #200 - #250, set FirstPoint=200 and Count=50.

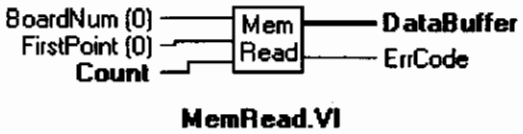
If you are going to read a large amount of data from the board in small chunks then set FirstPoint to FROMHERE to read each successive chunk. Using FROMHERE speeds up the operation of MemRead.VI when working with large amounts of data.

DT Connect Conflicts - The MemRead.VI can not be called while a DT Connect transfer is in progress. For example, if you start collecting A/D data to the memory board in the background (by calling AInScxx with the DTCONNECT + BACK-GROUND options). You can not call MemRead.VI until the AInScxx has completed. If you do you will get a DTACTIVE error.

MemRead.VI

Description: Reads data from a memory board into an array.

Summary:



MemRead.VI

Inputs: BoardNum - board number may be 0 to 9.
FirstPoint - Index of first point to read or FROMHERE.
Count - Number of points (words) to read

Outputs: DataBuffer - Output data array

EXPLANATION OF THE ARGUMENTS:

FirstPoint - Use the FirstPoint argument to specify the first point to be read. For example, to read points #200 - #250, set FirstPoint=200 and Count=50.

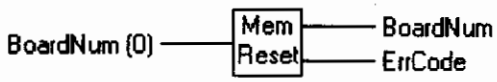
If you are going to read a large amount of data from the board in small chunks then set FirstPoint to FROMHERE to read each successive chunk. Using FROMHERE speeds up the operation of MemRead.VI when working with large amounts of data.

DT-CONNECT Conflicts - The MemRead.VI can not be called while a DT-CONNECT transfer is in progress. For example, if you start collecting A/D data to the memory board in the background (by calling AInScxx.VI with the DTCONNECT + BACKGROUND options). You can not call MemRead.VI until the AInScxx.VI has completed. If you do you will get a DTACTIVE error.

MemReset.VI

Description: Resets the memory board pointer to the start of the data. The memory boards are sequential devices. They contain a counter which points to the 'current' word in memory. Every time a word is read or written this counter increments to the next word.

Summary:



MemReset.VI

Input: BoardNum - board number may be 0 to 9.

Outputs: BoardNum - board number may be 0 to 9.
ErrCode - Error code

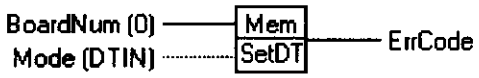
This VI is used to reset the counter back to the start of the memory. Between successive calls to AInScxx you would call this VI so that the second AInScxx overwrites the data from the first call. Otherwise the data from the first AInScxx will be followed by the data from the second AInScxx in the memory on the card.

Likewise, anytime you call MemRead or MemWrite it will leave the counter pointing to the next memory location after the data that you read or wrote. Call MemReset to reset back to the start of the memory buffer before the next call to AInScxx.

MemSetDT.VI

Description: Sets the DT Connect Mode of a Memory Board

Summary:



MemSetDT.VI

Inputs: BoardNum - board number may be 0 to 9.
Mode - Direction of memory board DT Transfer

Output: ErrCode -Error code

EXPLANATION OF THE ARGUMENTS:

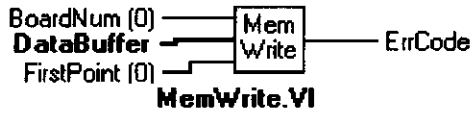
Mode - Must be set to either DTIN or DTOUT. Set the Mode on the memory board to DTIN if you wish to transfer data from an A/D board to the memory board. Set Mode=DTOUT if you wish to transfer data from a memory board to a D/A board.

This command only controls the direction of data transfer between the memory board and another board that is connected to it via a DT Connect cable.

MemWrite.VI

Description: Writes data from an array to the memory card

Summary:



Inputs: BoardNum - board number may be 0 to 9.
DataBuffer - Pointer to the data array
FirstPoint - Index of first point to write or FROMHERE.

Output: ErrCode - Error code

EXPLANATION OF THE ARGUMENTS:

FirstPoint - Use the FirstPoint argument to specify where in the board's memory to write the first point. For example, to write to locations #200 - #250, set FirstPoint=200 and Count=50.

If you are going to write a large amount of data to the board in small chunks then set FirstPoint to FROMHERE to write each successive chunk. Using FROMHERE speeds up the operation of MemWrite when working with large amounts of data.

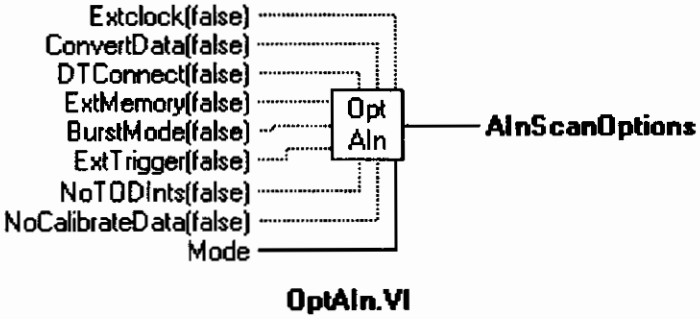
DT Connect Conflicts - The MemWrite VI can not be called while a DT Connect transfer is in progress. For example, if you start collecting A/D data to the memory board in the background (by calling AInScxx with the DTCONNECT + BACK-GROUND options). You can not call MemWrite until the AInScxx has completed. If you do you will get a DTACTIVE error.

OptAIn.VI Changed R3.3 ID

Description: Generates option input for AInScxx VI's

Change at Rev3.3 added NoCalibrateData option.

Summary:



Inputs: Extclock - External or internal clock
ConvertData - Seperate data and channel tags
DTConnect - DT connect option
ExtMemory - External memory option(Mega Fifo board)
BurstMode - Burst mode option (board specific)
ExtTrigger - External trigger option
NoTODInts - Option to disable time of day interupts
NoCalibrateData -
Mode - Sampling mode used

Output: AInScanOptions - Anded value of input options.

EXPLANATION OF THE ARGUMENTS:

The output of this VI must be wired to the options input of the AInScFg VI or AInScBg VI.

ExtClock - If this option is used then conversions will be controlled by the signal on the trigger input line rather than by the internal pacer clock. Each conversion will be triggered on the rising edge of the trigger input signal. When this option is used the Rate argument is ignored. The sampling rate is dependent on the trigger signal.

ConvertData - If the CONVERTDATA option is used then the data that is returned to data buffer (array) will automatically be converted to 12 bit A/D values. If NOCONVERTDATA is used then the data from 12 bit A/D boartds will be returned as 16 bit values that will contain both a 12 bit A/D value and a 4 bit channel number. After the data collection is complete you can call ACvtData to convert the data after the fact. CONVERTDATA may not be specified if you are using a VI and DMA transfers, or a 16 bit A/D board.

DTConnect - All A/D values will be sent to the A/D board's DT CONNECT port.

ExtMemory -Data is returned to a data buffer (array). EXTMEMORY causes the command to send the data to a connected memory board via the DT-Connect interface rather than returning the data to data buffer (array). Everytime this option is used it overwrites any data already stored in the memory board. The data should be unloaded with the MemRead VI before collecting new data.

BurstMode - Enables burst mode sampling. The A/D is clocked at the maximum A/D rate between samples in order to minimize channel to channel skew. Scans are initiated at the rate specified by rate.

ExtTrigger - If this option is specified the sampling will not begin until the trigger input line is asserted (See hardware manual for details but this is generally a TTL High).

NoTODInts - If this option is specified then the system's time-of-day interrupts are disabled for the duration of the scan. These interrupts are used to update the systems real time clock and are also used by various other programs. These interrupts can limit the maximum sampling speed of some boards - particularly the PCM-DAS08. If the interrupts are turned off using this option then the realtime clock will fall behind by the length of time that the scan takes.

NoCalibrateData - Turns off real time software calibration for boards which are software calibrated by applying calibration factors to the data on a sample by sample basis as it is acquired. Examples are the PCM-DAS16/330 and PCM-DAS16x/12. Turning off software calibration saves CPU time during a high speed acquisition run. This may be required if your processor is less than a 150MHz Pentium and you desire an acquisition speed in excess of 200KHz. These numbers may not apply to your system. Only trial will tell for sure. DO NOT use this option if you do not have to. If this option is used the data must be calibrated after the acquisition run with the ACalData VI.

Mode:

SINGLEIO - A/D conversions and transfers to memory are initiated by an interrupt. One interrupt per conversion.

DMAIO - A/D conversions are initiated by a trigger. Transfers are initiated by a DMA request.

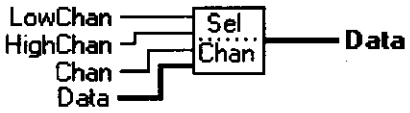
BLOCKIO - A/D conversions are initiated by a trigger. Transfers are handled by REP-INSW.

AInScanOptions - All the input are ANDED together and the result is passed to this parameter for input to AInScxx.VI's.

SelChan.VI

Description: Selects data for one channel from array with interleaved data for several channels.

Summary:



SelChan.VI

Inputs: LowChan - Low channel
HighChan - High channel
Chan - Channel to view
Data - Input array

Output: Data - Output array

EXPLANATION OF THE ARGUMENTS:

LowChan - Low channel of the scan specified in one of the scanning VI's.

HighChan - High channel of the scan specified in one of the scanning VI's.

Chan - The channel who's data you wish to view.

Data - Array holding data for all channel

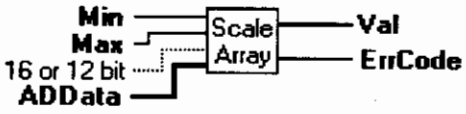
Data(output) - Array holding data for one channel.

NOTE:
See AInScFg.VI for an example of the proper use of this VI.

ScaleArr.VI

Description: Converts raw data from an entire array to engineering data.

Summary:



ScaleArr.VI

Inputs: Min - Lower limit of range
Max - Upper limit of range
16 or 12 bit - Length of raw data
ADData - Unconverted data array

Output: Val - Converted data array
ErrCode - Error code

EXPLANATION OF THE ARGUMENTS:

Min - Lower limit of selected range. For example, if the selected range is BIP5VOLTS, then this value will be -5.

Max - Upper limit of selected range. For example, if the selected range is BIP5VOLTS, then this value will be +5.

16 or 12 bit - Length of data to be converted. Depends on the type of card being used.

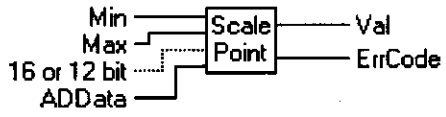
ADData- Array with unconverted raw data.

Val - Array with converted engineering data.

ScalePnt.VI

Description: Converts raw data point to engineering data.

Summary:



ScalePnt.VI

Inputs: Min - Lower limit of range
Max - Upper limit of range
16 or 12 bit - Length of raw data
ADData - Unconverted data array single element.

Output: Val - Converted data array element
ErrCode - Error code

EXPLANATION OF THE ARGUMENTS:

Min - Lower limit of selected range. For example, if the selected range is BIP5VOLTS, thenthis value will be -5.

Max - Upper limit of selected range. For example, if the selected range is BIP5VOLTS, thenthis value will be +5.

16 or 12 bit - Length of data to be converted. Depends on the type of card being used.

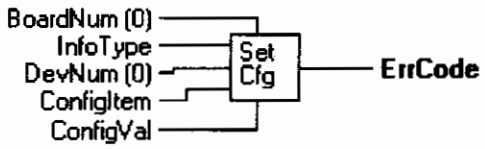
ADData- Array element with unconverted raw data.

Val - Converted engineering data.

SetCfg.VI

Description: Sets a configuration option for a board. The configuration information for all boards is stored in the CB.CFG file. All programs that use the library read this file. This function can be used to override the configuration information stored in the CB.CFG file

Summary:



SetCfg.VI

Arguments: InfoType - Which class of configuration information you want to retreive
BoardNum - Board name (0-9)
DevNum - Specifies which device within board
ConfigItem - Specifies which configuration item
ConfigVal - New value to set option to

EXPLANATION OF THE ARGUMENTS

InfoType - The configuration information for each board is grouped into different categories. This argument specifies which category you want. It should be set to one of the following constants:

GLOBALINFO - Information about the configuration file
BOARDINFO - general information about a board
DIGITALINFO - information about a digital device
COUNTERINFO - information about a counter device
EXPANSIONINFO - information about an expansion device
MISCINFO - One of the miscellaneous options for the board

BoardNum - refers to the board number associated with a board when it was installed.

DevNum - Selects a partilcuar device. If InfoType=DIGITALINFO then DevNum specifies which of the board's digital devices you want information on. If InfoType=COUNTERINFO then DevNum specifies which of the board's counter device.

ConfigItem - Specifies which configuration item you wish to retreive. Refer to the table below for a lsit of all of the possible values for ConfigItem.

ConfigVal - The value to set the sepcified configuration item to.

NOTES:
The list of ConfigItem values for each category of configuration information is:

InfoType = GLOBALINFO
GIVERSION - CB.CFG file format
GINUMBOARDS - Number of configured boards
GINUMEXPBOARDS - Number of expansions configured boards

InfoType = BOARDINFO
BIBASEADR - Base address of board
BIBOARDTYPE - Board Type
BIINTLEVEL - Interrupt level
BIDMACHAN - DMA channel
BIINITIALIZED - TRUE (-1) or FALSE (0)
BICLOCK - Clock freq in MHz (1 or 10)
BIRANGE - Selected volatge range

BINUMADCHANS - Number of A/D channels
BIUSESEXPS - Supports expansion boards TRUE/FALSE
BIDINUMDEVS - Number of digital devices
BIDIDEVNUM - Index into digital information for first device
BICINUMDEVS - Number of counter devices
BICIDEVNUM - Index into counter information for first device
BINUMDACHANS - Number of D/A channels
BIWAITSTATE - Setting of Wait State jumper
BINUMIOPORTS - Number of IOPorts used by baord
BIPARENTBOARD - Board number of parent board
BIDTBOARD - Board number of connected DT board

InfoType = DIGITALINFO
DIBASEADR - Base address
DIINITIALIZED - TRUE (-1) or FALSE (0)
DIDEVTYPE - Device Type - AUXPORT, FIRSTPORTA etc
DIMASK - Bit mask for this port
DIREADWRITE - Read require before write TRUE/FALSE
DICONFIG - Current configuration INPUT or OUTPUT
DINUMBITS - Number of bits in port
DICURVAL - Current value of outputs

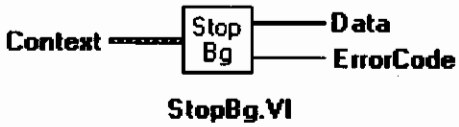
InfoType = COUNTERINFO
CIBASEADR - Base address
CIINITIALIZED - TRUE (-1) or FALSE (0)
CICTRTYPE - 8254 or 9513 counter
CICTRNUM - Which counter on chip
CICONFIGBYTE - Configuration byte

InfoType = EXPANSIONINFO
XIBOARDTYPE - Board type
XIMUXADCHAN1 - A/D channel board is connect to
XIMUXADCHAN2 - 2nd A/D channel board is connected to
XIRANGE1 - Range (gain) of low 16 chans
XIRANGE2 - Range (gain) of high 16 chans
XICJCCHAN - A/D channel that CJC is connected to
XITHERMTYPE - Thermocouple type
XINUMEXPCHANS - Number of expansion channels on board
XIPARENTBOARD - Board number of parent A/D board

StopBg.VI

Description: Stops any background operation that is in progress for the specified board. This VI can be used to stop any VI that is running in the background.

Summary:



Input: Context - Input data structure from a background operation.

Output: Data - Output array extracted from Context
ErrorCode - Error code

EXPLANATION OF THE ARGUMENTS:

Context - Data structure containing information from a background operation. Some of the information included is the board number, the data array, the array size, and the initial status of the background operation.

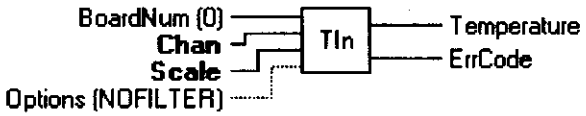
NOTE:
Wiring of this VI should conform to the following pattern:

Start a background operation. GetStat.VI checks for completion(boolean output called "Running"). StopBg.VI terminates the operation, if not already done, and frees memory aliases. Data output from the background operation is passed to GetStat.VI and StopBg.VI via "Context", and can be wired from one or both of them for intermediate or final actions, respectively. The demo VI's illustrate this process effectively.

TIn.VI Changed R3.3 ID

Description: Reads a multiplexer input channel, linearizes it according to selected thermocouple type, and returns the temperature in degrees. The CJC channel, the gain, and thermocouple type, are read from the configuration file. They should be set by running the InstaCal™ configuration program.

Summary:



TIn.VI

Inputs: BoardNum - board number may be 0 to 9.
Chan - Channel to read
Scale - The temperature scale for which to calulate the temperature in degrees.
Options - Bit fields that control various options

Outputs: Temperature - Temperature returned here
ErrCode - Error code

EXPLANATION OF THE ARGUMENTS:

Chan - Input channel to read. The channel number is calculated using the following formula:

A/DChan = A/D channel that mux is connected to
MuxChan = Mux board input channel number
Chan = (ADChan+1) * 16 + MuxChan

For example, if you had an EXP16 connected to a CIO-DAS08 via the CIO-DAS08 channel 0 (remember, DAS08 channels are numbered 0,1,2,3,4,5,6 & 7), AND if you had a thermocouple connected to channel 5 of the EXP16, the value for Chan would be (0+1)*16 + 5 = 21.

Scale - Specifies the temperature scale that the input will be converted to. Choices are CELSIUS, FAHRENHEIT and KELVIN.

Temperature - The temperature is returned here. Temperature is reported in degrees.

A/D Range - IMPORTANT - If the EXP board is connected to an A/D that does not have programmable gain (DAS08, DAS16, DAS16F) then the A/D board (switches) must be set for +- 5 volt range.

Options:

FILTER - The TIn applies a smoothing function to thermocouple readings very much like the electrical smoothing inherent in all themocouple instruments. This is the default.

FILTEROFF - If you use the NOFILTER option then the thermocouple will readings will not be smoothed and you will see a scattering of readings around an mean.

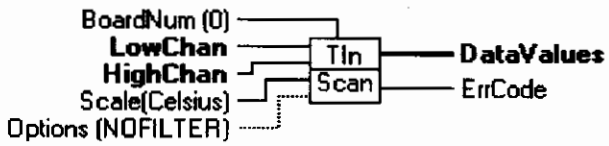
Note on CJC Channel: The CJC channel is set in the InstaCal install program. If you have multiple EXP boards, the LabVIEW VI will apply the CJC reading to the linearization formula in this manner. First, if you have chosen a CJC channel for the EXP board that the channel you are reading is on, it will use the CJC temp reading from that channel. Second, if you have left the CJC channel for the EXP board that the channel you are reading is on to NOT SET, the VI will use the CJC reading from the next lower EXP board with a CJC channel selected.

For example: If you have 4 CIO-EXP16 boards connected to a CIO-DAS08 on channel 0,1,2 and 3, and you have chosen CIO-EXP16 #1 (connected to CIO-DAS08 channel 0) to have its CJC read on CIO-DAS08 channel 7 , AND, you have left CIO-EXP16s 2, 3 and 4 CJC channels to NOT SET, then those CIO-EXP boards will all use the CJC reading from CIO-EXP16 #1, connected to channel 7 for linearization. As you can see, it is important to keep the CIO-EXP boards in the same case and out of any breezes to ensure a clean CJC reading.

TInScan.VI Changed R3.3 ID

Description: Reads a range of channels from a multiplexer board, linearizes them according to thermocouple type, and returns the temperatures to an array in degrees. The CJC channel, the gain, and thermocouple type are read from the configuration file. Use the InstaCal™ configuration program to change any of these options.

Summary:



TInScan.VI

Inputs: BoardNum - board number may be 0 to 9.
LowChan - Low mux channel of scan
HighChan - High mux channel of scan
Scale - CELSIUS, FAHRENHEIT or KELVIN
Options - Bit fields that control various options

Outputs: DataValues - Temperaturse returned here
ErrCode - Error code

EXPLANATION OF THE ARGUMENTS:

LowChan/HighChan - Specify the range of multiplexer channels that will be scanned. These channel numbers are calculated using the following formula:

A/DChan = A/D channel that mux is connected to
MuxChan = Mux board input channel number
Chan = (ADChan+1) * 16 + MuxChan

For example, if you had an EXP16 connected to a CIO-DAS08 via the CIO-DAS08 channel 0 (remember, DAS08 channels are numbered 0,1,2,3,4,5,6 & 7), AND if you had a thermocouple connected to channel 5 of the EXP16, the value for Chan would be (0+1)*16 + 5 = 21.

Scale - Specifies the temperature scale that the input will be converted to. Choices are CELSIUS, FAHRENHEIT and KELVIN.

DataValues[] - The temperature is returned in degrees. Each element in the array corresponds to a channel in the scan. DataBuffer must be at least large enough to hold HighChan - LowChan + 1 temperature values.

Chan - Input channel to read. The channel number is calculated using the following formula:

A/DChan = A/D channel that mux is connected to
MuxChan = Mux board input channel number
Chan = (ADChan+1) * 16 + MuxChan

Options:

FILTER - The TIn applies a smoothing function to thermocouple readings very much like the electrical smoothing inherent in all themocouple instruments. This is the default.

FILTEROFF - If you use the NOFILTER option then the thermocouple will readings will not be smoothed and you will see a scattering of readings around an mean.

Range - IMPORTANT - If the EXP board is connected to an A/D that does not have programmable gain (DAS08, DAS16, DAS16F) then the A/D board (switches) must be set for +- 5 volt range.

Note on CJC Channel: The CJC channel is set in the InstaCal install program. If you have multiple EXP boards, Universal Library will apply the CJC reading to the linearization formula in this manner. First, if you have chosen a CJC channel for the EXP board that the channel you are reading is on, it will use the CJC temp reading from that channel. Second, if you have left the CJC channel for the EXP board that the channel you are reading is on to NOT SET, the library will use the CJC reading from the next lower EXP board with a CJC channel selected.

For example: If you have 4 CIO-EXP16 boards connected to a CIO-DAS08 on channel 0,1,2 and 3, and you have chosen CIO-EXP16 #1 (connected to CIO-DAS08 channel 0) to have its CJC read on CIO-DAS08 channel 7 , AND, you have left CIO-EXP16s 2, 3 and 4 CJC channels to NOT SET, then those CIO-EXP boards will all use the CJC reading from CIO-EXP16 #1, connected to channel 7 for linearization. As you can see, it is important to keep the CIO-EXP boards in the same case and out of any breezes to ensure a clean CJC reading.

Using the File VI's

One of the most powerful features of the ComputerBoards Universal Library LabVIEW Extensions is the ability to collect very large amounts of data to a "streamer" file. The amount of data that can be collected is limited only by the size of your hard disk.

Once all of the data has been streamed to a file your program can read it back into arrays and process it in chunks.

.

Overview

The library contains four VI's that are used with "streamer" files. FileAInS.VI and FilePret.VI read the A/D and store the data in a "streamer" file. FileInfo.VI returns information about the streamer file. FileRead.VI reads data from a "streamer" file to an array.

In addition to these VI's, the extensions comes with three utility programs - MAKESTRM.EXE, FRAGTEST.EXE and RDSTREAM.EXE

MAKESTRM creates a "streamer" file. This program should be run to allocate a file large enough to hold all of the data that will be later collected with FileAInS.VI or FilePret.VI.

The syntaxis C:\MAKESTRM filename # <enter>

FRAGTEST checks an existing disk file to see if it is fragmented. In order to run at the faster sampling rates the "streamer" file must not be fragmented. Refer to "Speeding up Disk Files" below for more information.

The syntaxis C:\FRAGTEST filename <enter>

RDSTREAM reads a "streamer" file and prints its contents on the screen.

The syntaxis C:\RDSTREAM filename <enter>

Hard Disk vs Ram Disk Files

The simplest type of files to use is a standard DOS file on a hard disk. The advantage of hard disk files is that they can be very large. The file size is only limited by the amount of free space on the disk. Hard disk files have the disadvantage of being slower than RAM disks. RAM disk (or virtual disk) files are faster but they are limited in size by the amount of available memory in your computer.

Maximum Sampling Speed

The maximum sustainable sampling rate that can be specified with the File VI's is very hard to predict. It depends on the speed of the CPU and the speed of the disk.

In addition to the variation in sampling speed from machine to machine there can also be variations on the same machine between consecutive operations of the same program. When reading an A/D to memory (non-streaming modes) there is a hard and fast maximum sampling speed that can not be exceeded. When using the streaming modes the maximum rate is much fuzzier and must be arrived at by trial and error.

A rough guideline of attainable speeds are that on a 33 MHz 80386 machine with a fast hard disk it should be possible to collect a megabyte of data at 200khz sampling rate to a disk file. It should also be possible to collect a megabyte of data to a RAM disk at 330 kHz.

In general the maximum sustainable speed for FilePret.VI will be somewhat less than for FileAInS.VI.

Another characteristic of these "streaming" modes is that the more data you collect the lower the maximum speed will be. On any machine with any speed disk you can collect 32000 samples to a disk file at the maximum A/D speed of 330 kHz. If you are pushing the upper limits of speed you will find that you can collect 100k samples at a faster rate than you can collect 500k samples, etc.

How To Determine Maximum Sampling Speed

The only way to determine the maximum safe speed is to try it repeatedly. Remember if it works the first time it will not necessarily work the next time. Therefore the only way to be sure that you can reliably run at a particular speed is to try it numerous times or else increase the speed to the point where it begins to fail everytime so that you get some sense of whether or not you are pushing the speed limit on your computer.

To test it, write a program that calls FileAInS.VI or FilePret.VI (depending on whether you need pre-trigger data). Check the error code that is returned. If you get an OVERRUN error (error code of 29) it means that the sampling rate is too high. Whenever you get OVERRUN error, some data was collected but not all of it. It is often useful to check how much data was collected to find out whether it was almost fast enough or not even close.

Speeding Up Disk Files (De-Fragmenting):

Because of the way that disks work, the time that it takes to write to them can vary tremendously. A large disk file is made up of many small pieces that are written individually to the disk. If the file is contiguous (each piece is side by side) the speed is very fast. If the file is fragmented (pieces are in different places on the disk) the speed is much slower. If you create a large disk file the odds are overwhelming that it will be fragmented to some degree and the maximum sampling speed will be much lower than it would be for an unfragmented file.

To get around this problem you should use a Disk Optimizer or De-Fragmenter program immediately before creating the streamer file with MAKESTRM. Once you create the streamer file it will remain unfragmented so long as you don't erase it and recreate it.

Probably the most widely known Disk Optimizer program comes as part of the Norton Utilities, it is called Speed Disk or SD. To run it type: SD /Q

This will execute the "Quick" optimize, which for these purposes works as well as the Full Optimization.

After de-fragmenting the disk create a streamer file that is large enough to hold as much data as you plan to collect with FileAInS.VI or FilePret.VI. To create the disk file run the standalone MAKESTRM.EXE program. This will create a streamer file of the required size.

Once the file is created run FRAGTEST.EXE to see whether or not the file is fragmented. It is possible that the file may be fragmented even though you just de-fragmented the disk. The reason for this is that the disk may contain some bad sectors which could not be moved when the disk was optimized. When you create the new file if it hits one of these bad sectors it has to skip over it - hence fragmention.

If FRAGTEST reports that the file is fragmented create a second file and test that with FRAGTEST. Repeat this until FRAGTEST reports that the file is OK. Once you have an unfragmented disk file you can try using it with FileAInS.VI or FilePret.VI to collect data. If the maximum sampling speed is still too slow you should probably switch to a RAM disk.

What is a Ram Disk?

A RAM disk is not really a disk. It is a device driver that sets aside some of the computer's memory and makes it appear to DOS as a disk drive. When you install a RAM disk on your computer it appears exactly as if you have another VERY fast hard disk drive. For example if you have one hard disk - drive C: then when you install the RAM disk it will appear as if you have another hard disk - drive D:

Once the RAM disk is installed all DOS commands work exactly the same on the RAM disk as on the hard disk. For example you can COPY, DEL, MKDIR, CD just as you would on a hard disk.

Installing a RAM Disk:

The RAM disk driver comes with DOS. Refer to your DOS manual for more information. In older versions of DOS it is called either RAMDRIVE.SYS or VDISK.SYS. To install it you must add one line to your \CONFIG.SYS file. Find which directory the DOS files are installed in on your machine. CD to that directory and look for a file called RAMDRIVE.SYS or VDISK.SYS. If it is not there look at the other .SYS files in the directory and refer to your DOS manual to find out if any of them are a RAM Disk driver. Once you have located the file add an entry to the \CONFIG.SYS file.

If the RAMDRIVE.SYS file was in a directory called DOS then you would add the following line to the \CONFIG.SYS file.

```
device=c:\dos\ramdrive.sys
```

The default size for the RAM disk is usually 64K. You will almost certainly want to make it larger than that. The larger you make it the more data you can collect but the less memory will be available for other programs.

To set up a 4 megabyte RAM disk you would add the following line to your CONFIG.SYS file:

```
device=c:\dos\ramdrive.sys 4000
```

If your computer is an 80286 or an 80386 then you should install the RAM disk in extended memory (above 1M) by specifying the /e option: device=c:\dos\ramdrive.sys 4000 /e

Once you add the new line to the \CONFIG.SYS file, reboot the computer (Press CTRL-ALT-DEL) to install the RAM disk. When the machine reboots it should print a message on the screen describing the RAM disk.

Using The Ram Disk

To use the RAM disk you just specify the drive letter in the FileName argument of FileAInS.VI or FilePret.VI. For example if the RAM disk is drive D: on you system then you could set the name of the "streamer" file in your program to "d:TEST.DAT"

This file can be created with the MAKESTRM.EXE program supplied with the UniversalLibrary.

To set up a file larg enough to hold 1 million samples, include this line in your AUTOEXEC.BAT file:

```
C:\CB\MAKESTRM D:\TEST.DAT 1000000
```

The name TEST.DAT is an example. Use the name of your preference.

When you execute FileAInS.VI or FilePret.VI it will fill up the file on your RAM drive. This file will be lost as soon as the power is switched off so if you wish to keep the data you must copy it to the hard disk before turning the computer off.

Board Specific Information

This chapter describes the uses of the Universal Library LabVIEW Extensions with each of the data acquisition boards in Computer Boards product line.

CIO-, PC104- and PCM- CTR Series Boards Limitations

Counter Vi's: These boards use a 9513 counter chip so they don't support 8254 vi's (C8254Cfg.VI).

Digital I/O Vi's These boards support all of the digital I/O vi's except DInScxx.VI and DOutScxx.VI. The PortNum is AUXPORT and it reads 8 and writes 8 digital bits for CTR05 and 16 bits for CTR10.

The Auxilliary digital outputs on the CIO-CTR10 are set up as one 16 bit port. When using DOut, the digital values are written as a single 16 bit port.

CTR05 supports the counters numbered 1-5

CTR10 and CTR10HD support the counters numbered 1-10

CTR20HD supports the counters numbered 1-20

CIO-DAC Series Boards Limitations

These boards only support the analog output vi's. When used with AOutScFg.VI, the Rate is ignored, Count should be set to HighChan - LowChan + 1.

CIO- and PC104- DAS08 Series Boards

Analog Input VI's:

All analog input vi's except for APretrxx.VI any Filexx.VI are available.
A/D Ranges DAS08 - The DAS08 does not have programmable gain so the Range argument to analog input vi's should be set to 0.

CIO-DAS08-PGH and CIO-DAS08-AOH supports the following A/D ranges:

BIP10VOLTS = +- 10 volts	BIP5VOLTS = +- 5 volts
BIP1VOLTS = +- 1 volts	BIPPT5VOLTS = +- .5 volts
BIPPT1VOLTS = +- .1 volts	BIPPT05VOLTS = +- .05 volts
BIPPT01VOLTS = +- .01 volts	BIPPT005VOLTS = +- .005 volts
UNI10VOLTS = 0 - 10 volts	UNI1VOLTS = 0 - 1 volts
UNIPT1VOLTS = 0 - .1 volts	UNIPT01VOLTS = 0 - .01 volts

CIO-DAS08-PGL and CIO-DAS08-AOL supports the following A/D ranges:

BIP10VOLTS = +- 10 volts	BIP5VOLTS = +- 5 volts
BIP2PT5VOLTS = +- 2.5 volts	BIP1PT25VOLTS = +- 1.25 volts
BIPPT625VOLTS = + 0.625 volts	
UNI10VOLTS = 0 - 10 volts	UNI5VOLTS = 0 - 5 volts
UNI2PT5VOLTS = 0 - 2.5 volts	UNI1PT25VOLTS = 0 - 1.25 volts

Maximum Sampling Rate :

The maximum analog sampling rate is dependent on the speed of the computer that the board is installed in. In general it is somewhere in the range of 5-50 kHz. If the speed you request cannot be sustained, an overrun error will occur.

Options:

The DTCONNECT, EXTMEMORY, BLOCKIO and DMAIO options are not available with this board. BURSTMODE generates a warning message.

Wiring Requirements :

Before using the timed analog input vi's (AInScxx.VI, FileAInS.VI). The output of counter 1 (pin #6) must be wired to the Interrupt input (pin #24). An interrupt level must have been selected in InstaCal and the CB.CFG file saved.

Analog Output VI's:

AOutScFg.VI ignores the Rate argument and Count must be less than HighChan - LowChan + 1. The CIO-DAS08, CIO-DAS08-PGL and CIO-DAS08-PGH boards do not support the analog output vi's.

Digital I/O VI's:

The CIO-DAS08 and CIO-DAS08/AO support the following digital I/O ports AUXPORT, FIRSTPORTA, FIRSTPORTB, FIRSTPORTCL, andFIRSTPORTCH. These boards to do not support the DInScanxx.VI's or DOutScxx.VI's.

Counter I/O VI's:

The CIO-DAS08 and CIO-DAS08AO have an 8254 counter. Only those vi's that apply to the 8254 can be used.

CIO-DAS08/Jr Board

Analog Input Vi's
All analog input vi's except APreTrig.vi and Filexx.vi are avaliable.

A/D Ranges
This board does not have programable A/D ranges so the Range argument to the analog input vi's are ignored.

Sampling Speed
This board does not have a timer so the Rate argument to the analog scanning vi's are ignored. The scan will execute at the maximum speed possible. This speed will vary with CPU speed.

Options:
The DTCONNECT, EXTMEMORY, BLOCKIO, DMAIO, and SINGLEIO are not avaliable with this board.

Analog Output Vi's:
The analog output vi's only work if the optional D/A converters are installed on the board. In that case, the AOutScxx vi's ignore the Rate argument and the Count argument may not be greater then 2. The CONTINUOUS and BACKGROUND options are not allowed.

Digital I/O Vi's:
The digital I/O lines are adresses as AUXPORT. The maximum bit number for both input and output is 7.

Counter I/O Vi's:
This board does not have any counters and does not support any of the counter I/O vi's

CIO-DAS800 Series Boards

Analog Input Vi's
All analog input vi's except APreTrig.vi are available.

A/D Ranges -

DAS-800 the Range argument is ignored.

DAS-801 supports the following A/D ranges:
BIP10VOLTS = +/- 10 volts
BIP5VOLTS = +/- 5 volts

BIP1VOLTS = +/- 1 volts
BIPPT5VOLTS = +/- .5 volts
BIPPT05VOLTS = +/- .05 volts
BIPPT01VOLTS = +/- .01 volts

UNI10VOLTS = 0 - 10 volts
UNI1VOLTS = 0 - 1 volts
UNIPT1VOLTS = 0 - .1 volts
UNIPT01VOLTS = 0 - .01 volts

DAS-802 supports the following A/D ranges:

BIP10VOLTS = +/- 10 volts
BIP5VOLTS = +/- 5 volts
BIP2PT5VOLTS = +/- 2.5 volts
BIP1PT25VOLTS = +/- 1.25 volts
BIPPT625VOLTS = +/- .625 volts

UNI10VOLTS = 0 - 10 volts
UNI5VOLTS = 0 - 5 volts
UNI2PT5VOLTS = 0 - 2.5 volts
UNI1PT25VOLTS = 0 - 1.25 volts

Maximum Sampling Speed
The maximum sampling speed for these boards is 50 kHz.

Options:
The DTCONNECT, EXTMEMORY, and DMAIO options are not avaliable with this board.

Analog Output Vi's:
These boards do not have D/A converters and do not support analog output vi's.

Digital I/O Vi's:
The digital I/O lines are adresses as AUXPORT. The maximum bit number for input is 2 and for output is 3.

Counter I/O Vi's:
These board have 8254 counters. Only vi's that apply to the 8254 counter can be used.

CIO- and PC104- DAS16 Series Boards

Analog Input VI's:

Pre-trigger vi's - APretrxx.VI, Filexx.VI are available only with the DAS16/330, DAS16/330i, DAS16/M1; boards which support the REPINSW command.

A/D Range:

CIO-DAS16 and CIO-DAS16F do not have programmable gain so the Range argument to analog input vi's should be set to 0.

CIO-DAS16/Jr, CIO-DAS16/330, CIO-DAS16/330i, CIO-DAS16/M1, PC104-DAS16/Jr support the following programmable A/D ranges:

BIP10VOLTS = +- 10 volts
BIP5VOLTS = +- 5 volts
BIP2PT5VOLTS = +- 2.5 volts
BIP1PT25VOLTS = +- 1.25 volts
BIPPT625VOLTS = += .625 volts

UNI10VOLTS = 0 - 10 volts
UNI5VOLTS = 0 - 5 volts
UNI2PT5VOLTS = 0 - 2.5 volts
UNI1PT25VOLTS = 0 - 1.25 volts

Maximum Sampling Rate:

The maximum analog sampling rate varies per board according to the following table.:

CIO-DAS16	50 kHz	CIO-DAS16F	100 kHz	CIO-DAS16Jr	120 kHz
CIO-DAS16/330	330 kHz	CIO-DAS16/330i	330 kHz		

CIO-DAS16/M1: When using EXTMEMORY option, or if count <= 512, 1MHz, otherwise about 500KHz to PC memory.
@SubSubHead@Options:
The DTCONNECT & EXTMEMORY options are available with the CIO-DAS1600, CIO-DAS16/330, CIO-DAS16/330i and the CIO-DAS16/M1 board.

The BLOCKIO option is available with the CIO-DAS16/330, CIO-DAS16/330i and the CIO-DAS16/M1 board.

The DMAIO option can not be used while using the chan/gain queue on the DAS-330i board.

Single I/O and DMA I/O are not available on the CIO-DAS16/M1

Analog Output VI's:

AOutScFg.VI ignores the Rate argument and Count must be less than HighChan - LowChan + 1. AOutScBg.VI is not allowed.

Digital I/O VI's:

All of the boards support the AUXPORT. The CIO-DAS1600, CIO-DAS16 and CIO-DAS16F also support FIRSTPORTA, FIRSTPORTB, FIRSTPORTCL, and FIRSTPORTCH. These boards do not support the DInScxx.VI's or DOutScxx.VI's.

Counter I/O VI's:

The CIO- and PC104- DAS16 boards have an 8254 counter. Only those vi's that apply to the 8254 can be used. AOutScBg.VI is not allowed.

CIO-DAS48/PGA Board

Analog Input VI's:

This board does not support any of the pre-triggering vi's.
This board does not have an A/D pacer clock so the Rate argument to AInScxx.VI is ignored and the Count argument must never be greater than the number of channels in the scan. i.e. AinScxx.VI can only be used to take a single A/D sample from a range of channels.

A/D Ranges:

The board may be configured with a jumper for either voltage or current input.
The voltage input ranges are:

BIP10VOLTS = +/- 10 volts	BIP5VOLTS = +/- 5 volts
BIP2PT5VOLTS = +/- 2.5 volts	BIP1PT25VOLTS = +/- 1.25 volts
BIPPT625VOLTS = +/- 1.625 volts	
UNI10VOLTS= 0 - 10 volts	UNI5VOLTs = 0 -5 volts
UNI2PT5VOLTS= 0 - 2.5 volts	UNI1PT25VOLTS = 0 - 1.25 volts

The current input ranges are:

MA4TO20 = 4 - 20 ma	MA2TO10 = 2 - 10 ma
MA1TO5 = 1 - 5 ma	MAPT5TO2PT5 = .5 - 2.5 ma

@SubSubHead@Options
No options are allowed with the AinScxx.VI.

Analog Output VI's:

The CIO-DAS49/PGA board does not support the analog output vi's.

Digital I/O VI's:

The CIO-DAS48/PGA does not support any of the digital I/O vi's.

Counter I/O VI's:

The CIO-DAS48/PGA does not support any of the counter I/O vi's.

CIO-DAS1400 & CIO-DAS1600 Series Boards

Analog Input Vi's:

All analog input VI's except for APreTrxx.VI and FilePret.VI can be used with these boards.
A/D Ranges.

IMPORTANT: The CIO-DAS1400 and CIO-DAS1600 A/D ranges are configured with a combination of a switch (Unipoalr / Bipolar) and a programmable gain code. The state of this switch is set in the configuration file using InstaCal™. Once the UNI/BIP switch setting is selected, only matching ranges may be used in Universal Library programs.

The allowed A/D voltage ranges are:

CIO-DAS1402, CIO-DAS1602, CIO-DAS1602/16 Ranges

BIP10VOLTS= +/- 10 volts	BIP5VOLTS= +/- 5 volts
BIP2PT5VOLTS= +/- 2.5 volts	BIP1PT25VOLTS= +- 1.25 volts
UNI10VOLTS= 0 - 10 volts	UNI5VOLTS= 0 - 5 volts
UNI2PT5VOLTS= 0 - 2.5 volts	UNI1PT25VOLTS = 0 - 1.25 volts

@SubSubHead@CIO-DAS1401, CIO-DAS1601/12 Ranges

BIP10VOLTS = +/- 10 volts	BIP1VOLTS = +/- 1 volts
BIPPT1VOLTS = +/- .1 volts	BIPPT01VOLTS = +/- .01 volts
UNI10VOLTS = 0 - 10 volts	UNI1VOLTS = 0 - 1 volt
UNIPT1VOLTS = 0 - .1 volt	UNIPT01VOLTS = 0 - .01 volt

Maximum Sampling Rate:

The maximum analog (A/D) sampling rate for CIO-DAS1400 & CIO-DAS1600is: 330 kHZ to DT-Connect, 160 kHZ to memory CIO-DAS1600/16 - 100 kHz

Options

The BLOCKIO option can not be used with these boards.
The DTCONNECT and EXTMEMORY options cannot be used with the DAS1400 board.
The CONVERTDATA option can not be used with the DAS1600/16 board. This board does not return channel tags so there is no need for this option.
When EXTMEMORY is used with the CIO-DAS1600 the GetStat.VI will not return the current count and current index. This is a limitation imposed by maintaining identical registers to the KM-DAS1600.

Analog Output:

The CIO-DAS1400 board does not support the analog output vi's.
The CIO-DAS1600 series boards do. AOutScxx.VI ignores the Rate argument and Count must be no more then HighChanLowChan+1, The CONTINUOUS and BACKGROUND operations are not allowed.

Digital I/O VI's:

All of the boards have an AUXPORT, FIRSTPORTA, FIRSTPORTB, FIRSTPORTCL, and FIRSTPORTCH.
The AUXPORT has 4 digital inputs and 4 digital outputs. These boards support all of the digital I/O vi's except DInScxx.VI and DOutScxx.VI.

Counter I/O VI's:

These boards all have an 8254 counter. They do not support any of the 9513 counter vi's - C9513Ini.VI, C9513Cfg.VI, CStore.VI, CFreq.VI.

CIO- and PC104- DIxx Series Boards

These boards support the commands DBitIn.vi and DIn.vi.

CIO-DI48 supports FIRSTPORTA, FIRSTPORTB, FIRSTPORTCL, and FIRSTPORTCH for bits 0 to 23 and SECONDPORTA to SECONDPORTCH. Maximum bit is 47.

CIO-DI96 supports FIRSTPORTA, FIRSTPORTB, FIRSTPORTCL, and FIRSTPORTCH for bits 0 to 23 and FOURTHPORTA to FOURTHPORTCH. Maximum bit is 95

CIO-DI192 supports FIRSTPORTA, FIRSTPORTB, FIRSTPORTCL, and FIRSTPORTCH for bits 0 to 23 up through EIGHTHPORTA to EIGHTHPORTCH. Maximum bit is 191.

CIO- and PC104- DIO Series Boards

These boards support the digital I/O commands except for DInScxx.VI and DOutScxx.VI.
All of these boards support the following digital I/O ports - FIRSTPORTA, FIRSTPORTB, FIRSTPORTCL, and FIRSTPORTCH.

The DIO48 board also supports: SECONDPORTA, SECONDPORTB, SECONDPORTCL, SECONDPORTCH
The DIO-96 board also supports: SECONDPORTA, SECONDPORTB, SECONDPORTCL, SECONDPORTCH, THIRDPORTA, THIRDPORTB, THIRDPORTCL, THIRDPORTCH, FOURTHPORTA, FOURTHPORTB, FOURTHPORTCL, FOURTHPORTCH

The DIO-192 board also supports: FIFTHPORTA ..... EIGHTHPORTCH
The maximum bit number for DBitIn.VI and DBitOut.VI is 23 for the DIO-24, 47 for the DIO-48, 95 for the DIO-96, and 191 for the DIO-192.

CIO-, PC104- and PCM- DIO24/CTR3

Digital Vi's: This board supports all the digital vi's except DInScxx.VI and DOutScxx.VI. Ports supported are FIRSTPORTA, FIRSTPORTB, FIRSTPORTCL and FIRSTPORTCH.

Counter Vi's: This board has an 82C54 counter chip and is supported by the config and init vi's for the 82C54.

CIO- and PC104- DOxx and CIO-DOxxDD Series Boards

These boards support the commands DBitOut.vi and DOut.vi.

CIO-DO48 supports FIRSTPORTA, FIRSTPORTB, FIRSTPORTCL, and FIRSTPORTCH for bits 0 to 23 and SECONDPORTA to SECONDPORTCH. Maximum bit is 47.

CIO-DO96 supports FIRSTPORTA, FIRSTPORTB, FIRSTPORTCL, and FIRSTPORTCH for bits 0 to 23 and FOURTHPORTA to FOURTHPORTCH. Maximum bit is 95

CIO-DO192 supports FIRSTPORTA, FIRSTPORTB, FIRSTPORTCL, and FIRSTPORTCH for bits 0 to 23 up through EIGHTHPORTA to EIGHTHPORTCH. Maximum bit is 191

CIO-DO24DD supports FIRSTPORTA, FIRSTPORTB, FIRSTPORTCL, and FIRSTPORTCH for bits 0 to 23. Maximum bit is 23.

CIO-DO48DD supports FIRSTPORTA, FIRSTPORTB, FIRSTPORTCL, and FIRSTPORTCH for bits 0 to 23 and SECONDPORTA to SECONDPORTCH. Maximum bit is 47.

CIO-DDA06

AOutScFg.VI ignores the Rate argument and Count must be less than HighChan - LowChan + 1. AOutScBg is not allowed.

Digital I/O Vi's: This board supports all the digital vi's except DInScxx.VI and DOutScxx.VI. The board supports the following digital ports: FIRSTPORTA, FIRSTPORTB, FIRSTPORTCL, FIRSTPORTCH.

CIO-DISO48

Digital Vi's: This board supports all the digital vi's except DInScxx.VI and DOutScxx.VI. Ports supported are FIRSTPORTA, SECONDPORTA, THIRDPORTA, FOURTHPORTA, FIFTHPORTA & SIXTHPORTA.

CIO-EXP Series Boards

EXP boards are only used in combination with an A/D board.

Channel numbers for accessing the EXPansion boards begin at 16. To calculate the channel number for access to EXP channels, use the following formula:

Chan = (ADChan + 1) * 16 + EXPChan

If all A/D channels are not used for EXP output, direct input to the A/D board is still available at these channels (using channel numbers below 16).

CIO-INT32

Counter Vi's: This board has two 8536 chips, which have both counter and digital I/O and interrupt vectoring capabilities. Use the counter vi's for configuration and initialization that specify the 8536.

Digital I/O Vi's: This board supports all the digital vi's exceptDInScxx.VI and DOutScxx.VI. The board supports the following digital ports: FIRSTPORTA, FIRSTPORTB, FIRSTPORTCL, SECONDPORTA, SECONDPORTB and SECONDPORTCL.

CIO- ans PC104- PDISO8 and CIO-PDISO16

This boards support all of the digital I/O commands except for DInScxx.VI and DOutScxx.VI. The digital I/O port is addressed as AUXPORT.

CIO-PDMA16

This boards support all of the digital I/O commands except for DBitIn.VI and DBitOut.VI.
This board supports the AUXPORT, FIRSTPORTA, and FIRSTPORTB. The DInScxx.VI and DOutScxx.VI only work with FIRSTPORTA and FIRSTPORTB ports.

CIO-RELAY16

Digital Vi's: This board supports all digital output vi's except DOutScxx.VI. Ports are addressed as FIRSTPORTA and FIRSTPORTB

PPIO-DIO24H

Digital Vi's: This board supports all digital vi's except DInScxx.VI and DOutScxx.VI. Ports are addressed as FIRSTPORTA, FIRSTPORTB, FIRSTPORTCL and FIRSTPORTCH.

PPIO-AI8

Analog Input: This board does not support timed A/D input. AInScFg.VI Rate is ignored. AInScBg.VI is not allowed.

Digital I/O - AUXPORT is used for access to the digital I/O bits.

The PPIO-AI8 does not have programmable gain so the Range argument to analog input vi's should be set to 0.

PPIO-CTR06 Boards

Counter Vi's: This board uses an 82C54 so it supports the 8254 vi's - C8254Cfg.VI.
Digital I/O Vi's This board has a single digital port - AUXPORT. Depending on a jumper setting on the board it either has 8 digital inputs and 4 digital outputs OR 4 digital inputs and 8 digital outputs. This board does not support the DInScxx.VI or DOutScxx.VI.

cSBX-DDA04 Board

Digital I/O - AUXPORT is 8 bits of input, 14 bits of output.
Analog Output AOutScxx.VI maximum allowable Rate is 300,000.
Digital Output AOutScxx.VI maximum allowable rate is 500,000 - READ NOTE

Note: The cSBX-DDA4 boards allows interleaving of analog and digital output data. In order to support interleaving, a control bit, the MSb of each 16 bit word of analog or digital data indicates the data type. MSb = 0 for analog data and MSb = 1 for digital data. The data is passed to the board and there directed to the correct output type by hardware on the board which detects and acts on the MSb control bit.

PCM-DAC02 Board

Analog Output Vi's: Support AOut.VI for 2 channels - 0,1. When used with AOutScFg.VI Rate is ignored, Count should be set to HighChan - LowChan + 1. AOutScBg is not supported.

Digital I/O Vi's: Supports 2 configurable 4 bit ports - FIRSTPORTA and FIRSTPORTB. Each may be independently configured as either Inputs or Outputs via DCfgPort.VI).

PCM-DAS08 Board

Analog Input Vi's: Supports all analog input except APretxx.VI and FilePret.VI
A/D Range argument is ignored. Input range is fixed in hardware. Default is +/-5V but custom ranges may be ordered.
Sample Rate. See hardware manual for restrictions.
Options for AInScxx: Does not support DMAIO, BLOCKIO, DTCONNECT, EXTMEMORY.
Analog Output Vi's: None on this board.
Digital I/O Vi's. Supports digital I/O as AUXPORT. Max DBit is 3.

PCM-DAS16x/xx Series Boards

Analog Input Vi's - All analog input VI's except cbAPreT.VI are available.
These DAS16x/12 boards support the following A/D ranges:

BIP10VOLTS = +/- 10 volts	UNI10VOLTS = 0 - 10 volts
BIP5VOLTS = +/- 5 volts	UNI5VOLTS = 0 - 5 volts
BIP2PT5VOLTS = +/- 2.5 volts	UNI2PT5VOLTS = 0 - 2.5 volts
BIP1PT25VOLTS = +/- 1.25 volts	UNI1PT25VOLTS = 0 - 1.25 volts

These DAS16x/16 boards support the following A/D ranges:

BIP10VOLTS = +/- 10 volts	BIP5VOLTS = +/- 5 volts
BIP2PT5VOLTS = +/- 2.5 volts	BIP1PT25VOLTS = +/- 1.25 volts

These DAS16x/330 boards support the following A/D ranges:

BIP10VOLTS = +/- 10 volts	BIP5VOLTS = +/- 5 volts

Maximum Sampling Speed : The maximum sampling speed for these boards is 100 kHz.

Options: The DTCONNECT, EXTMEMORY, and DMAIO options are not avaalable with this board.
Analog Output Vi's: These boards do not have D/A converters and do not support analog output vi's.
Digital I/O Vi's: The digital I/O lines are adressed as two 4 bit ports - FIRSTPORTA and FIRSTPORTB. Each of these ports is individually configurable as input or output via cbDConfigPort. The maximum bit number that can be used with DbitIn.VI or DBitOut.VI is 7.
Counter I/O VI's: These board have 8254 counters. Only VI's that apply to the 8254 counter can be used.

Appendix A. Error Codes

Each entry in the list has four parts. The error code number, it's symbolic name, it's error message, and a explanation.

 # SYMBOLIC_NAME *Error message*
Description

0 NOERRORS *No error has occurred.*
The vi executed successfully

1 BADBOARD *Invalid board number.*
The board number that was specified does not match any of the boards that are listed in the configuration file. Run the configuration program to check which board numbers are configured.

2 DEADDIGITALDEV *Digital device is not responding - is base address correct?*
The digital device on the specified board is not responding. Either the board was installed incorrectly or the board is defective. Run the configuration program and make sure that the correct boards was installed.

3 DEADCOUNTERDEV *Counter device is not responding - is base address correct?*
The counter device on the specified board is not responding. Either the board was installed incorrectly or the board is defective. Run the configuration program and make sure that the correct board was installed.

4 DEADDADEV *D/A is not responding - is base address correct?*
The D/A device on the specified board is not responding. Either the board was installed incorrectly or the board is defective. Run the configuration program and make sure that the correct board was installed.

5 DEADADDEV *A/D is not responding - is base address correct?*
The A/D device on the specified board is not responding. Either the board was installed incorrectly or the board is defective. Run the configuration program and make sure that the correct boards was installed.

6 NOTDIGITALCONF *Selected board does not have digital I/O.*
A digital I/O vi was called with a board number that referred to a board that does not support digital I/O. Run the configuration program to see which type of board that board number refers to.

7 NOTCOUNTERCONF *Selected board does not have a counter.*
A counter vi was called with a board number that referred to a board that does not have a counter. Run the configuration program to see which type of board that board number refers to.

8 NOTDACONF *Selected board does not have a D/A.*
An analog output vi was called with a board number that referred to a board that does not have an analog output (D/A). Run the configuration program to see which type of board that board number refers to.

9 NOTADCONF *Selected board does not have an A/D.*
An analog input vi was called with a board number that referred to a board that does not have an analog input (A/D). Run the configuration program to see which type of board that board number refers to.

10 NOTMUXCONF *Selected board does not have thermocouple input.*
A thermocouple input vi was called with a board number that is not connected to an EXP board. Run the configuration program to view/change the board configuration.

11 BADPORTNUM *Invalid digital port number.*
The Port number that was specified for a digital I/O vi does not exist on the board that was specified.

12 BADCOUNTERDEVNUM *Invalid counter device.*
The Counter Number that was specified for a counter vi does not exist on the board that was specified.

13 BADDADEVNUM *Invalid D/A device.*
The D/A channel that was specified for an analog output vi does not exist on the board that was specified.

14 BADSAMPLEMODE *Invalid sample mode.*

A sample mode (INTIO, DMAIO or BLOCKIO) was specified in the Options argument that is not supported on this board. Try running the vi without setting any of the Sample Mode options.

15 BADINT *Board configured for invalid interrupt level.*

No interrupt was selected in InstaCal and one is required , or, the interrupt level for this board was set in the configuration file to either 10 or 11. The switch on the board is set for "compatible mode". These interrupts are not valid in compatible mode. Either change the switch setting on the board, or change the interrupt level with the configuration program to something other than 10 or 11.

16 BADADCHAN *Invalid A/D channel number.*

An invalid channel argument was passed to an analog input vi. The range of valid channel numbers depends on which A/D board you are using - refer to the board manual. For some boards it also depends on how the board is configured (with a switch). For those boards run the configuration program and check how many channels the board is configured for.

17 BADCOUNT *Invalid count.*

An invalid Count argument was specified to a vi.

18 BADCNTRCONFIG *Invalid counter configuration specified.*

An invalid Config argument was passed to C8254Cfg.VI. The only legal values are HIGHONLASTCOUNT, ONESHOT, RATEGENERATOR, SQUAREWAVE, SOFTWARESTROBE and HARDWARESTROBE.

19 BADDAVAL *Invalid D/A value.*

An invalid D/A value was passed as an argument to an analog output vi. The only legal values are 0 - 4095 for 12 bit boards or 0-65,535 for 16 bit boards.

20 BADDACHAN *Invalid D/A channel number.*

An inavlid D/A channel was passed as an argument to an analog output vi. The legal range of values depends on which D/A board you are using. Refer to the board manual to find how many D/A channels it has.

22 ALREADYACTIVE *Background operation already in progress.*

An attempt was made to start a second background process before the first one had completed. Background processes are started whenever the BACKGROUND option is used or by the CStore VI. To stop a background operation call StopBg.VI. To wait for a background process to complete call GetStat.VI and wait for status=IDLE.

24 BADRATE *Invalid sampling rate.*

Invalid sampling rate argument was specified. The rate was either a negative number or it was higher than the selected board supports. Refer to board specific information for maximum rates for each board.

25 COMPATMODE *Board switches set for Compatible mode.*

An operation was attempted that is not possible when the board's switch is set for 'compatible' operation. Most likely causes are using the BLOCKIO option or the pre-triggering vi's. Either turn off the 'compatible' mode switch on the board or don't use the BLOCKIO option or the pre-triggering vi's.

26 TRIGSTATE *Incorrect initial trigger state - D0 must start at TTL low.*

The pre-triggering command requires that the trigger be "off" when the vi is first called. It then waits for the trigger signal. Make sure that the Trigger Input line (D0) is held at TTL low before calling the pre-trigger vi.

27 ADSTATUSHUNG *A/D is not responding.*

The A/D board is not responding as it should. Usually indicates some kind of hardware problem - either defective hardware or more than one board at the same base address.

28 TOOFEW *Trigger occurred before requested number of samples were collected.*

A pre-trigger vi was called and the trigger signal occurred before the requested number of samples could be collected. This is only a warning message. The vi continued anyway. The data that was returned to the array will contain fewer than the expected number of points. The vi will return the actual number of pre-trigger points and the total number of points. You can use these two values to find your way around the data in the array.

29 OVERRUN *Data overrun - data was lost.*

Data was lost during an analog input because the computer could not keep up with the A/D sampling rate. This typically can only happen with the file input vi's. Possible solutions include lowering the sampling rate, defragmenting the "streamer" file, switching to a RAM disk, or lowering the count.

30 BADRANGE *Invalid voltage or counter range..*

Invalid Range argument was specified to an analog input vi. There are two possible cases. Either the board does not support software programmable gains, in which case the gain arguments should be set to 0, or else it does support programmable gain but not the one you specified. Refer to board specific information for a list of allowable A/D ranges for each board.

31 NOPROGGAIN *This A/D board does not have programmable gain.*

Invalid Range argument was passed to an analog input vi. The selected board does not support programmable gains so the only valid Range argument is 0.

32 BADFILENAME *Specified file name is not valid'.*

The FileName argument that was passed to a file vi is not valid. It is either an empty string or a NULL pointer.

33 DISKISFULL *Disk is full, could not complete operation.*

A file operation failed before completing because the disk that it was writing to is full. Try erasing some files from the disk. If this error occurred during either FileAInS.VI or FilePret.VI it indicates another problem. The disk space for these commands should have been previously allocated with MAKESTRM.EXE program. If this error is generated when data is being collected it indicates that you did not allocate a large enough file with MAKESTRM.EXE.

34 COMPATWARN *Board switch set to compatible mode - sampling speed may be limited.*

The board's switch is set for 'Compatible mode". When in "compatible mode" BLOCKIO transfers are not possible. BLOCKIO sampling was specified but it has automatically been changed to DMAIO transfers. The maximum smapling rate will be limited to the maximum rate for DMA transfers. Change the "compatible mode" switch on the board if you want to use BLOCKIO transfers.

35 BADPOINTER *Pointer is not valid.*

An invalid (NULL) pointer was passed as an argument to a vi.

37 RATEWARNING *Sample rate may be too fast for SINGLEIO mode.*

The specified sampling rate MAY be too high. The maximum allowable sampling rate depends very much on the computer that the program is running on. This warning is generated based on the slowest CPU speed. Your computer may be able to sustain faster rates, but, you should expect the computer to lock up (fail to respond to keyboard input) if you do exceed the sampling rate your computer can sustain.

38 CONVERTDMA *CONVERTDATA cannot be used with DMA I/O and BACKGROUND.*

The CONVERTDATA and BACKGROUND options can not be used together when the board is transferring data via DMA. Possible solutions include: Use ACvtData to convert the data after it is collected. Don't use BACKGROUND option. Use BLOCKIO Option if your A/D board supports it. Use INTIO Option if your computer is fast enough to support the selected sampling rate.

39 DTCONNECTERR *Board does not support DTCONNECT option.*

The DTCONNECT Option was passed to an analog input vi. The selected board does not support that option.

40 FORECONTINUOUS *CONTINUOUS can only be run with BACKGROUND.*

The CONTINUOUS Option was passed to a vi without also setting the BACKGROUND Option. This is not allowed. Any time you set the CONTINUOUS Option you must also set the BACKGROUND option.

41 BADBOARDTYPE This vi can not be used with this board'.

An attempt was made to call a vi for a board that does not support that vi.

42 WRONGDIGCONFIG *Digital port not configured correctly for requested operation.*

Some of the digital ports (FIRSTPORTA - EIGHTHPORTCH) must be configured as inputs OR outputs but not both. An attempt was made to use a digital input vi on a port that was configured as an output or vice versa. Use DCfgPort.VI to switch a port's direction. If you do not call DCfgPort.VI in your program then all of the configurable ports will be in in an unknown state (input or output).

43 NOTCONFIGURABLE *This digital port is not configurable (its an In/Out port).*
DCfgPort.VI was called for a port that is not configurable. Check the Port Number argument passed to DCfgPort.VI and make sure that it is in the range FIRSTPORTA - EIGHTHPORTCH. If not then there is no need to call this vi.

44 BADPORTCONFIG *Invalid digital port configuration.*
The Direction argument passed to DCfgPort.VI is invalid. It must be set to either DIGITALIN or DIGITALOUT.

45 BADFIRSTPOINT *First point number is not valid'.*
The First Point argument to FileRead.VI is invalid. It is either a negative number or it is larger then the number of points in the file.

46 ENDOFFILE *Attempted to read past the end of the file.*
FileRead.VI attempted to read beyond the end of the file. Check the file length with FileInfo.VI and make sure that the First Point and Count arguments to FileRead.VI are correct for that file length.

47 NOT8254CTR *This board does not have 8254 counter.*
The C8254Cfg.VI vi was called for a board that has a counter but not an 8254 counter. This vi can only be used with an 8254.

48 NOT9513CTR *This board does not have 9513 counter.*
A vi was called for a board that has a counter but not a 9513 counter. This vi can only be used with an 9513.

49 BADTRIGTYPE *Invalid TrigType.*
ATrig.VI was called with an invalid TrigType argument. It must be set to either TRIGABOVE or TRIGBELOW.

50 BADTRIGVALUE *Invalid TrigValue (> 4095).*
ATrig.VI was called with an invalid TrigValue argument. It must be in the range 0 - 4095.

52 BADOPTION *Invalid option specified for this vi.*
The Option argument contains an option that is not valid for this vi.

53 BADPRETRIGCOUNT *Invalid PreTrigCount specified.*
Either APretrxx.VI or FilePret.VI was called with an invalid preTrigCount argument. A valid pre-trigger count must be > 0 and less then TotalCount-512. It also must be less then 32000 for APretrxx.VI and less than 16,000 for FilePret.VI.

55 BADDIVIDER *Invalid Fout Divider value.*
The Fout Divider argument to C9513Ini.VI is not valid. It must be in the range 0 - 15.

56 BADSOURCE *Invalid Fout Source value.*
The Source argument to C9513Ini.VI is not valid. It must be one of the following values CTRINPUT1, CTRINPUT2, CTRINPUT3, CTRINPUT4, CTRINPUT5, GATE1, GATE2, GATE3, GATE4, GATE5, FREQ1, FERQ2, FREQ3, FREQ4, FREQ5 (i.e. 0 - 15)

57 BADCOMPARE *Invalid compare value.*
One or both of the compare arguments to C9513Ini.VI is not valid. They must be set to ENABLED or DISABLED.

58 BADTIMEOFDAY *Invalid Time Of Day value.*
The TimeOfDay argument to C9513Ini.VI is not valid. It must be set to either ENABLED or DISABLED.

59 BADGATEINTERVAL *Invalid Gate Interval value.*
The GateInterval argument to CFreqIn.VI is not valid. It must be greater then 0.

60 BADGATECNTRL *Invalid Gate Control value.*
The GateControl argument to C9513Cfg.VI is not valid. It must be in the range 0 -7.

61 BADCOUNTEREDGE *Invalid Counter Edge value*
The CounterEdge argument to C9513Cfg.VI is not valid. It must be set to either POSITIVEEDGE or NEGATIVEEDGE.

62 BADSPCLGATE *Invalid Special Gate value.*
The SpecialGate argument to C9513Cfg.VI is not valid. It must be set to either ENABLED or DISABLED.

63 BADRELOAD *Invalid Reload value.*
The Reload argument to C9513Cfg.VI is not valid. It must be set to either LOADREG or LOADANDHOLDREG

64 BADRECYCLEFLAG *Invalid Recycle Mode value.*
The RecycleMode argument to C9513Cfg.VI is not valid. It must be set to either ENABLED or DISABLED.

65 BADBCDFLAG *Invalid BCD Mode value.*
The BCDMode argument to C9513Cfg.VI is not valid. It must be set to either ENABLED or DISABLED.

66 BADDIRECTION *Invalid Count Direction value.*
The CountDirection argument to C9513Cfg.VI is not valid. It must be set to either COUNTUP or COUNTDOWN.

67 BADOUTCONTROL *Invalid Output Control value.*
The OutputControl argument to C9513Cfg.VI is not valid. It must be set to either ALWAYSLOW, HIGHPULSEONTC, TOGGLEONTC, DISCONNECTED or LOWPULSEONTC.

68 BADBITNUMBER *Invalid Bitnum specified.*
The BitNum argument to DBitIn.VI or DBitOut.VI is not valid. The valid range of bit numbers depends on the selected board. If it is a DIO24 compatible board the maximum bit number is 23. If it's a DIO96, the maximum bit number is 95.

69 NONEENABLED *None of the counter channels were enabled.*
None of the counter channels were marked as ENABLED in the CntrCntrl array that was passed to CStore.VI. At least one of the counter channels must be enabled.

70 BADCTRCONTROL *An element of Cntr Control array not set to DISABLED or ENABLED.*
One of the elements of the CntrControl array that was passed to cbCStoreOnInt.VI was set to something other then ENABLED or DISABLED. The array must have at least ten elements and the first ten elements must be set to either ENABLED or DISABLED.

71 BADEXPCHAN *Invalid EXP channel specified.*
An invalid channel was passed to one of the thermocouple input commands. The channel number must be >= 0. The maximum allowable channel number depends on which EXP board is being used. Refer to the board manual to find the number of channels.

72 WRONGADRANGE *Board set to wrong A/D range for reading thermocouples.*
A thermocouple input vi was called to read an EXP board input. The EXP board is connected to an A/D board that is set to the wrong range. The A/D must be set to the -5 to +5 volt range to work with the EXP board.

73 OUTOFRANGE *Temperature input is out of range.*
A thermocouple input vi returned an inavalid temperature. This usually indicates an open connection in the thermocouple or its connection to the mux board.

74 BADTEMPSCALE *Invalid temperature scale specified.*
The Scale argument to a thermocouple input vi is not valid. It must be set to either CELSIUS, FAHRENHEIT or KELVIN.

75 BADERRCODE *Invalid error code.*
The ErrCode argument to cbGetErrMsg.VI is invalid. It is not a known error code.

76 NOQUEUE *Specified board does not have channel/gain queue.*
The vi that was called requires that the board has a channel/gain queue. The specified board does not have a queue.

77 CONTINUOUSCOUNT *Count must be > 1023 to use recycle mode*

The count argument is not valid.

78 UNDERRUN *D/A FIFO went empty during output.*

The specified D/A output rate could not be sustained. This error should not normally occur.

79 BADMEMMODE *Invalid memory mode specified.*

The memory mode that was selected with cbMemSetDTMode is not one of the valid modes.

80 FREQOVERRUN *Measured frequency too high for selected gating interval.*

The gating interval used with cbFreqIn is too long to measure the frequency of the signal connected to the counter. The counter is overflowing. Decrease the gating interval to eliminate the error.

81 NOCJCCHAN *A CJC Channel must be configured to make temperature measurements.*

When the board was installed (with the InstaCal installation program) no CJC channel was selected. To use the temperature measurement vi's you must first select a CJC channel on the A/D board and then rerun the installation program.

82 BADCHIPNUM *Invalid Chipnum specified.*

An invalid ChipNum argument was used with the cbC9513Init.VI vi. If the board is CTR05 then ChipNum should be set to 0. If it is a CTR10, ChipNum should be either 0 or 1.

83 DIGNOTENABLED *The digital I/O on this board is not enabled.*

When the board was installed (with the InstaCal installation program), the expansion digital I/O was set to DISABLED. To use these digital I/O lines you must enable the digital I/O on the board (with a jumper) and then re-run the installation program and set the digital I/O to ENABLED.

84 CONVERT16BITS *CONVERT option can not be used with 16 bit A/D converters.*

When using a 16 bit A/D (DAS1600/16), if you try to use the CONVERT option with cbAInScan.VI or call cbAConvertData.VI you will get this error.

85 NOMEMBOARD *The EXTMEMORY option requires that a MEGA-FIFO be attached..*

Attempt to use a cbMem vi without a MEGA-FIFO board installed. Install MEGA-FIFO through InstaCal.

86 DTACTIVE *No memory read/write allowed while DT transfer in progress.*

A read or write to a memory board was attempted while data was being transferred via DT Connect.

87 NOTMEMCONF *Specified board is not a memory board.*

The specified board is not a memory board. This vi only works with memory boards.

88 ODDCHAN *The first channel in scan and number of channels must be even (0, 2, 4, etc).*

Some boards use a channel/gain queue that require the first channel in the queue and the number of channels in the queue always be an even channel. This error can occur even when you are not in the process of loading the queue. Some boards use the queue automatically with cbAInScan.VI. On those boards the low channel must be an even number.

89 CTRNOINIT *Counter was not configured or initialized.*

You attempted to use cbCLoad.VI or cbCIn.VI before inititalizing and configuring the counter.

90 NOT8536CTR *This board does not have an 8536 counter chip.*

Attempt to use 8536 initialization or configuration on board without 8536 chip.

91 FREERUNNING *Board doesn't time A/D sampling. Collecting at fastest possible speed.*

This board does not have an A/D pacer mechanism and you have called the vi cbAInScan.VI. The A/D will be sampled in a tight software loop as fast as the CPU can execute the instructions. The speed of sampling is dependent on the computer and the concurrent tasks.

92 INTERRUPTED *Operation interrupted with Ctrl-C key.*

A background operation was stopped before completion because either the Ctrl-C or Ctrl-Break keys were pressed.

93 NOSELECTORS *No selector could be allocated.*
A Windows selector required by the library could not be allocated. Close any unneeded open Windows applications and try again.

94 NOBURSTMODE *This board does not support burst mode.*
An attempt was made to use the BURSTMODE option on a board which does not support that option.

95 NOTWINDOWSFUNC *This vi is not available in Windows library.*
The library vi you called is not supported in the the current revision of Universal Library for Windows Languages. It may be supported in the future. Please call.

96 NOTSIMULCONF *Board not configured for SIMULTANEOUS option.*
The configuration file of the D/A board in InstaCal must be set for simultaneous update before you use the SIMULTANEOUS option of cbAOutScan.VI vi. Of course, it goes without saying that the jumpers on the D/A board must be set for simultaneous update before it will work.

97 EVENODDMISMATCH *An even channel is in an odd slot in the queue, or vice versa.*
The channel gain queue on some A/D boards has a restriction that the channel numbers must be in even queue positions and odd channel numbers must be in odd queue positions.

98 M1RATEWARNING *Sampling speed to system memory MAY be too fast.*
The A/D board sampling speed you have requested may be too fast for the computer system bus trasnfer to complete before the next packet is ready for transfer. If this is the case, data will overrun and sample data will be garbled. This warning is initiated whenever you request a sample rate over 625KHz AND the sample set is larger than the FIFO buffer on the board AND an external memory board, such as a MEGA-FIFO is not being used. Your system may be able to handle the rate requested but only experimentation will bear this out. Your system may be capable of the full 1MHz rate directly to system memory.

99 NOTRS485 *Selected board is not a RS-485 board*
An attempt was made to call cbRS485 with a board that is not RS485 compatible.

100 NOTDOSFUNC *This vi not available in DOS*
The vi that was called is not available in the DOS version of the Universal Library.

101 RANGEMISMATCH *Bipolar and unipolar ranges cannot be used together in A/D queue.*
The channel/gain queue should only be loaded (via cbALoadQueue) with all unipolar or bipolar ranges

102 CLOCKTOOSLOW *Sampling rate is too high for clockspeed, change clock umper on board.*
The sampling rate that you requested is too fast. The A/D board pacer might be capable of running at a higher rate. Check the board for an XTAL jumper and, if it is not, place the jumper in the position for the highest rate. Once the jumper is set, re-run InstaCal.

200 INTERNALERR *Internal error occurred in library.*
An internal error occurred in the library. This should not happen. Please report to Computer Board's Technical Support Dept.

301 CANTLOCKDMABUN *Could not lock DMA buffer into physical memory*
The requested scan vi could not execute because the data array could not be locked into physical low memory. To avoid this error try closing other running programs. If that doesn't work you need to use a smaller Count with the scan vi.

302 DMAINUSE *DMA in use. DMA is being controlled by another VxD*
Another running program is using the system's DMA controller. Either close that program or don't use DMA with the scan command - Try the SINGLEIO option.

303 BADMEMHANDLE *An invalid Windows memory handle passed to vi*
The Windows memory handle that was passed to a library vi is invalid. The handle should have been returned by a previous call to cbWinBufAlloc.VI.

304 NOENHANCEDMODE *Windows is not running in Enhanced mode*
Windows is not running in Enhanced mode. The Windows version of the Universal Library requires Windows Enhanced mode.

305 NOVDDINSTALLED *The CBUL.386 virtual device driver is not installed*
The Universal Library virtual device driver (CBUL.386) is not installed correctly. Your system's SYSTEM.INI file should contain the following line in the [386Enh] section. device=cbul.386

306 NOWINDOWSMEMORY *The requested size of Windows page-locked memory is not available*
The requested vi could not execute because the data array could not be locked into physical memory. To avoid this error try closing other running programs. If that doesn't work you need to use a smaller buffer or add more memory to your computer.

501 DOSBADFUNC *Invalid DOS vi number specified *** Internal Error.*
An internal error occurred in the library. This should not happen. Please report to Computer Board's Technical Support Dept.

502 DOSFILENOTFOUND *Cannot find file.*
A file name that was passed to cbFileGetInfo.VI or cbFileRead.VI does not exist.

503 DOSPATHNOTFOUND *Cannot find path.*
A file name that was passed to cbFileGetInfo.VI or cbFileRead.VI contained an invalid path (directory) name.

504 DOSNOHANDLES *No DOS handles left.*
An attempt to open a file failed because there were no more DOS file handles available. There are too many open files. If you need to have more files open simultaneously add a line to your system's \CONFIG.SYS file that says files = 20

505 DOSACCESSDENIED *File Access denied.*
An attempt was made to open a file that is locked. This error usualy indicates that another program (or another user on a network) has already opened this file. The file will remain locked until they close it.

506 DOSINVALIDHANDLE *Invalid handle - *** Internal Error.*
An internal error occurred in the library. This should not happen. Please report to Computer Board's Technical Support Dept.

507 DOSNOMEMORY *DOS has no more memory available.*
No memory is available. Try removing any TSR programs or device drivers that you are not using from memory.

515 DOSBADDRIVE *Invalid disk drive.*
An invalid drive was specified for a file operation.

518 DOSTOOMANYFILES *Too many disk files.*
There are too many files on your disk. This is an unlikely error. It will occur only if you are attempting to add greater than the number of entries allowed on the file allocation table.

519 DOSWRITEPROTECT *Disk is write protected.*
An attempt was made to write on a write-protected disk. This should only happen when writing to floppy disks. Remove the write-protect sticker from the disk and try again.

521 DOSDRIVENOTREADY *Disk drive is not ready.*
An attempt was made to write on a disk drive that isn't ready. This should only happen when writing to floppy disks. It indicates that there is no disk in the drive or the drive door is open.

525 DOSSEEKERROR *DOS seek error.*
This error indicates a serious error on the disk or drive.

529 DOSWRITEFAULT *DOS write error.*
This error indicates a serious error on the disk or drive.

530 DOSREADFAULT *DOS read error.*
This error indicates a serious error on the disk or drive.

531 DOSGENERALFAULT *DOS general fault.*
This error indicates a serious error on the disk or drive.

601 NODLL *DLL could not be loadedt.*
This error indicates there was a problem loading CBW.DLL.

602 NOFUN *Function not found in DLL.*
This error indicates that a function that was called is not located in the loaded DLL.

603 ARRAYTOOSMALL *User-defined LabVIEW array too small.*
This error indicates an array that was defined by the user is not large enough to hold the number of points selected.

607 WIN_UNK_INT *Program error: installing unknown interrupt handler*
This is an internal library program error. Check your U.L. installation. Verify that the DLL and VXD versions are synchronized.

608 WIN_CANNOT_SET_INT *The selected interrupt is not available.*
The interrupt number assigned to the selected board is not available for scan functions under Windows.

609 WIN_CANNOT_ENABLE_INT *The selected interrupt cannot be enabled.*
An internal Windows error occured while tring to unmask the interrupt line assigned to the selected board.

610 WIN_CANNOT_RESET_INT
The selected interrupt cannot be reset to the default handler.

611 WIN_CANNOT_DISABLE_INT *The selected interrupt cannot be disabled.*
An internal Windows error occured upon the completion of a scan function while disabling interrupts on the interrupt number assigned to the specific board.

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP