32-bit LabVIEW Drivers
User's Guide
(Windows 95&98/NT/2000 Version)

2st Edition
December,25 2000
S/N:2053000201
Advantech Corporation

Table of Contents

About this manual

Chap 1 Installation and Configuration
1.1 Installation
1.2 Configuration

Chap 2 Tutorials
2.1 Search for examples
2.2 Build a Virtual Instrument
2.3 Tutorials
2.3.1 Tutorial 1: Get voltage input from PCL-818
2.3.2 Tutorial 2: Get digital input from PCL-818
2.3.3 Tutorial 3: Programming with Advantech EASY I/O VIs
2.3.4 Tutorial 4: Error handling for error message

Chap 3 EASY I/O VIs
3.1 Select Pop
3.2 EasyFastAI

Chap 4 Advanced I/O VIs
4.1 Function flow overview
4.2 DeviceManager
4.3 SlowAI
4.4 SlowAO
4.5 DIO
4.6 CountTimer
4.7 Alarm
4.8 FastAI
4.9 FastAO
4.10 WatchDog
4.11 COM port
4.12 PWM
4.13 ADAMConfig

Appendix-A Card gain codes
Appendix-B Runtime error code listings
Appendix-C Reference

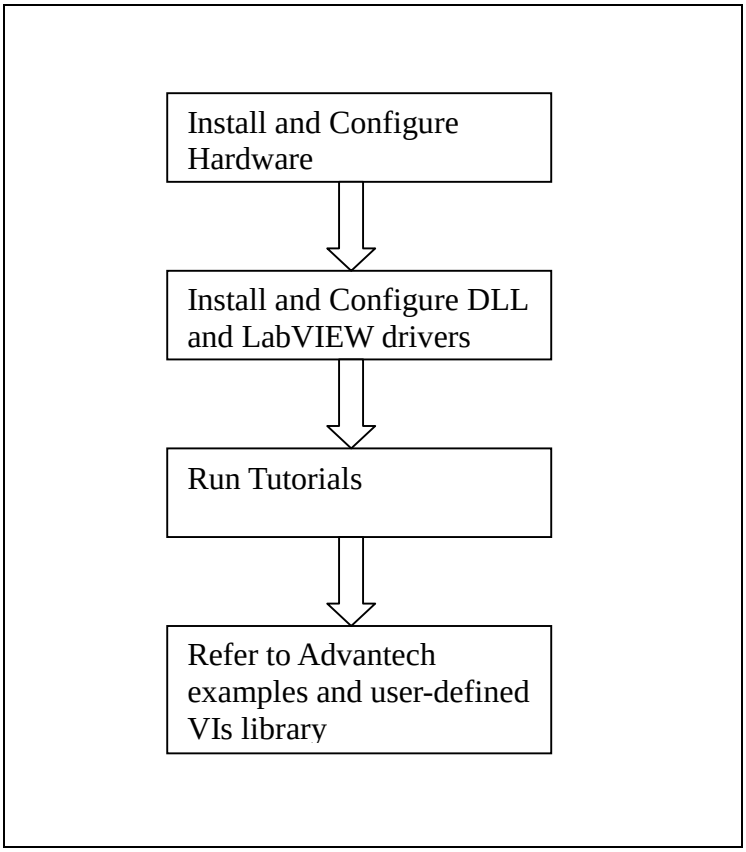
About This Manual

The LabView for Windows Data Acquisition and Control VI(Virtual Instrument) user manual describes the VIs that support Advantech plug-in data acquisition and control cards and field-bus data acquisition and control modules.

In order to efficiently link Advantech data acquisition and control (DA&C) devices with LabView to develop your DA&C application, you should be familiar with the operation of LabView, your computer, your computer's operating system, and your DA&C hardware.

How to use this Manual

Figure 1.1 is a flowchart that shows the four steps you should follow to make learning the DA&C VIs an easier process. These steps are explained in the following sections:



Install and Configure Hardware

Follow the instructions in the section **Installing Your DA&C Hardware** later in this chapter. Hardware installation procedures vary different DA&C for different module types (plug-in DA&C cards, field bus DA&C module).

After you have installed your DA&C hardware, use the **PCLTEST** utility to configure and test your hardware. The **PCLTEST** utility is bundled with Advantech DLL drivers.

Install DLL and LabVIEW drivers

The Tutorial sections will guide you through the several steps of installation and configuration to set up DLL drivers and LabVIEW drivers. DLL drivers should be installed and configured before LabVIEW drivers. For detailed information, please refer to chapter 1 Installation and Configuration.

Run tutorials

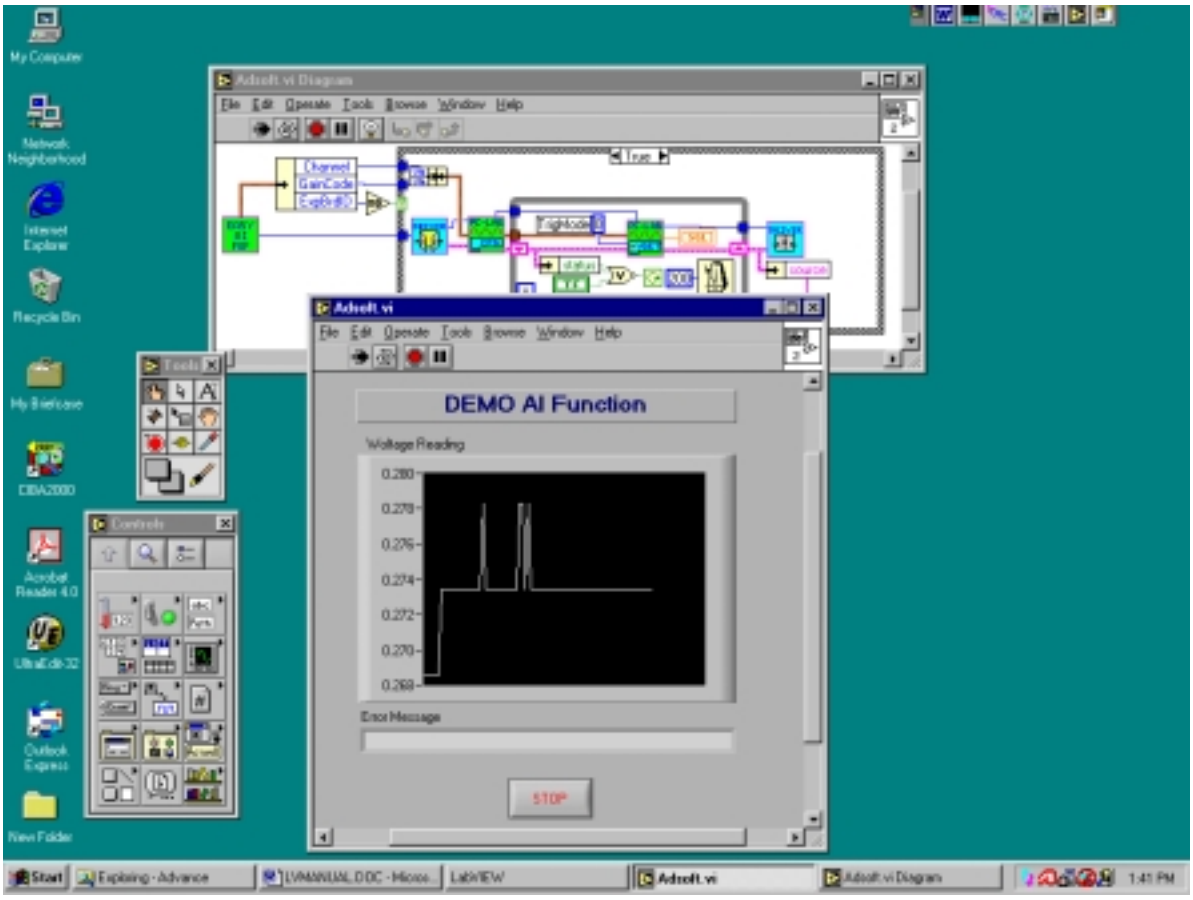
The Tutorial sections in this chapter will guide you through a simple example VIs for each functional area. The example will verify your hardware installation, illustrate basic data acquisition concepts, and give you a starting point for your application development. The **Getting Started** sections are as follows:

- Getting Started with Advanced VI: Analog Input
- Getting Started with Advanced VI: Digital Input
- Getting Started with Easy VI
- Getting Started with Error handling

Refer to Advantech user-defined VIs

Advantech provides EASY I/O and Advanced I/O user-defined VIs in LabVIEW drivers. For EASY I/O VIs, please refer to Chapter 3 EASY I/O VIs. For Advanced I/O VIs, please refer to chapter 4 Advanced I/O VIs.

Chapter 1 Installation and Configuration



This manual describes Advantech's 32-bit LabVIEW driver designed to integrate Advantech PC-LabCards and remote data acquisition and control devices with LabVIEW software. These drivers will be executed in Microsoft Windows 95 and Windows NT 4.0 environments. 32-bit LabVIEW drivers feature with both high speed and normal speed data acquisition functions and provide two different interfaces set for users. One is Easy I/O VIs (virtual instrument) for ease-to-use, another is Advanced I/O VIs (virtual instrument) for complex applications. Through 32-bit LabVIEW drivers, you can easily integrate Advantech DA&C devices with LabVIEW software to build a powerful laboratory automation solution.

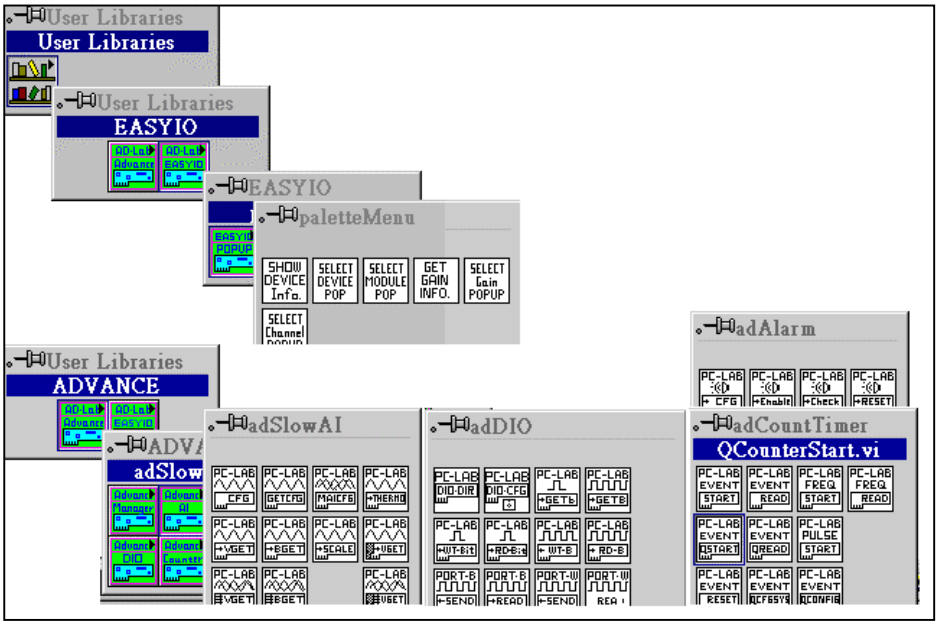
Functionality

Advantech 32-bit LabVIEW drivers enable you to use PC-LabCards and remote data acquisition and control modules with LabVIEW software. The LabVIEW driver forms an interface between Advantech DA&C device DLL drivers, which contain all the relevant functions to control the PC-LabCards or remote DA&C device, and the LabVIEW software. LabVIEW driver forms a VI (virtual instrument) in the LabVIEW package, which enables other applications to be used in conjunction with the PC-LabCards hardware.

Advantech's 32-bit LabVIEW drivers provide two different sets of VIs: Easy I/O VIs and Advanced I/O VIs. The Easy I/O VIs are a collection of I/O blocks which are used to perform basic I/O operations. The Advanced I/O VIs are more flexible than the Easy I/O VIs, at the cost of complexity.

32-bit LabVIEW drivers provide the following I/O VI functions:
Single Channel AI: single channel of analog input
Multi-Channel AI: multiple channel of analog input
Fast AI: high-speed analog input
AO: one channel of analog output
Fast AO: high-speed analog output
Temp Measure: temperature measurement
DI: digital input
DO: digital output
Event Counting: event counting
Frequency Measure: frequency measurement
Pulse Output: pulse output

The Advanced I/O VIs provides more hardware and software functionality, flexibility, and efficiency than the Easy I/O VIs. For functions of Advanced I/O VIs, please refer to Advantech DA&C device DLL drivers.



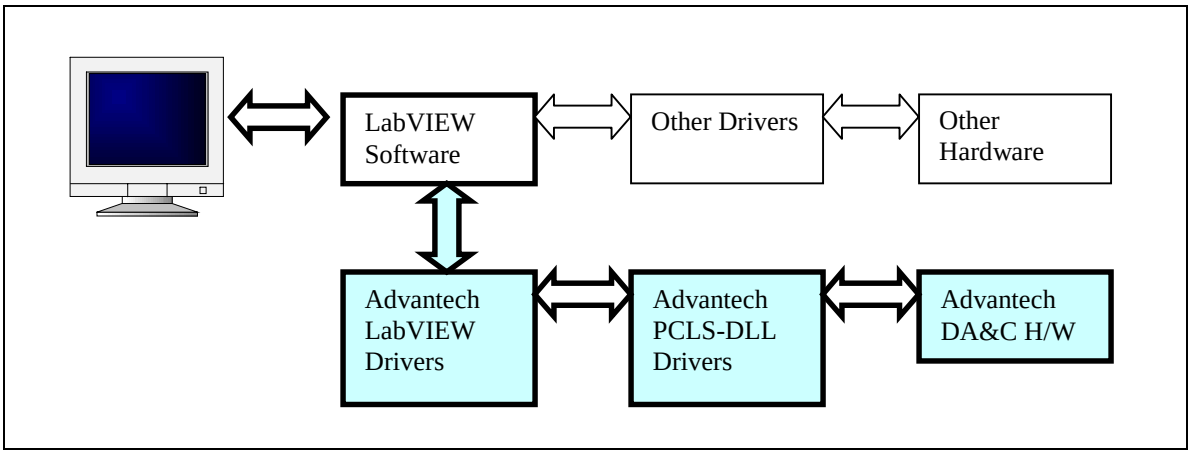
Hardware Supporting

32-bit LabVIEW drivers come in four kinds of packages according to the hardware functions and features, as listed below:

- **Plug-in DA&C cards**: PCL-1800, PCL-711S, PCL-720, PCL-722, PCL-724, PCL-725, PCL-726, PCL-727, PCL-728, PCL-730, PCL-731, PCL-733, PCL-734, PCL-735,

PCL-812PG, PCL-813B, PCL-816, PCL-818, PCL-818H/HD/HG/L/LS, PCL-833, PCL-836,PCI-1750/1751/1710/1710HG/1720/1760/1712/1713/1753/1752/1754/1756/1711/1731

- **PCM series modules:** PCM-3718, PCM-3724
- **MIC series modules:** MIC-2718/2728/2730/2732/2750/2752
- **ADAM-4000 and ADAM-5000/485 series modules**



1.1 Installation

Installing Advantech DLL (PCLS-DLL) drivers

1. Insert the **PCLS-DLL** driver installation disk in your PC's floppy drive (or Windows 95/NT DLL Drivers CD into CD-ROM device) and execute "**SETUP.EXE**" under Windows. The setup utility will expand the files on the installation disk and copy them to your system.

2. Add the device driver to the Windows system. Access the **Control Panel** in the Main Window and in the **Control Panel** window, click on the icon marked "**Drivers**". Choose the **ADD** button, and highlight the entry in the list box "**Unlisted or Updated Driver**" and press the **OK** button. Name the path to the location of the device driver e.g. "*C:\Program Files\Advantech\ADSAPI*" where you can find the **OEMSETUP.INF** file.

Once the driver is installed, you should see that it is now listed in the list box. There should be only one of each type of I/O device or board installed in WINDOWS. Once each device is installed, it remains installed until you remove it by using the **Remove** button in the Drivers dialog box.

Now you can enter the Device Installation Program ("**DEVINST.EXE**") to install and configure the setting of each I/O device driver you installed.

Installing the LabVIEW driver

1. Insert the Installation Disk in your PC's floppy drive A or Advantech Drivers Supporting CD in CD-ROM device.

2. From Program Manager, switch to LabVIEW driver directory and select “**SETUP.EXE**” file. Double click the mouse button to run it.

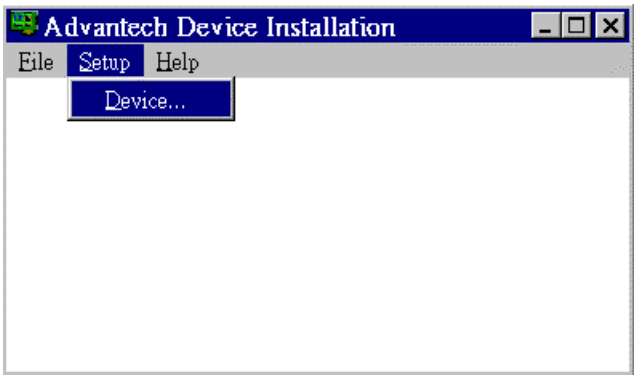
3. Supply the information required by the installation program. The setup program will install all required files to the HDD and the directory you specified.

An example program is also installed in a subdirectory of the LabVIEW driver software. The default directory is "*C:\LABVIEW\ EXAMPLES \ADVANTECH*".

4. LabVIEW Driver User's Manual is copied into the specified installation directory. You can use Microsoft Word 97 to read it and print it for reference.

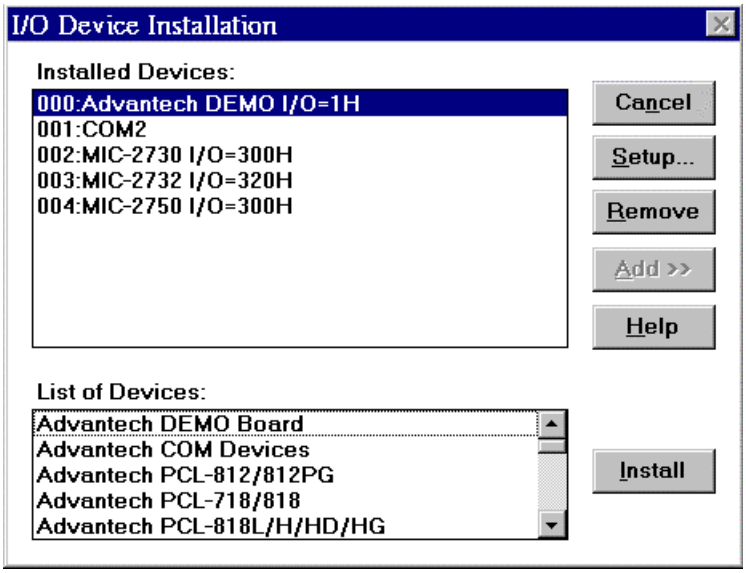
1.2 Configuration

To configure your hardware, you use the Advantech Device Installation Utility (“**DEVINST.EXE**”) that can be found in the PCLS-DLL driver directory (“*C:\Program Files\Advantech\ADSAPI”* etc.).



Installing each I/O device

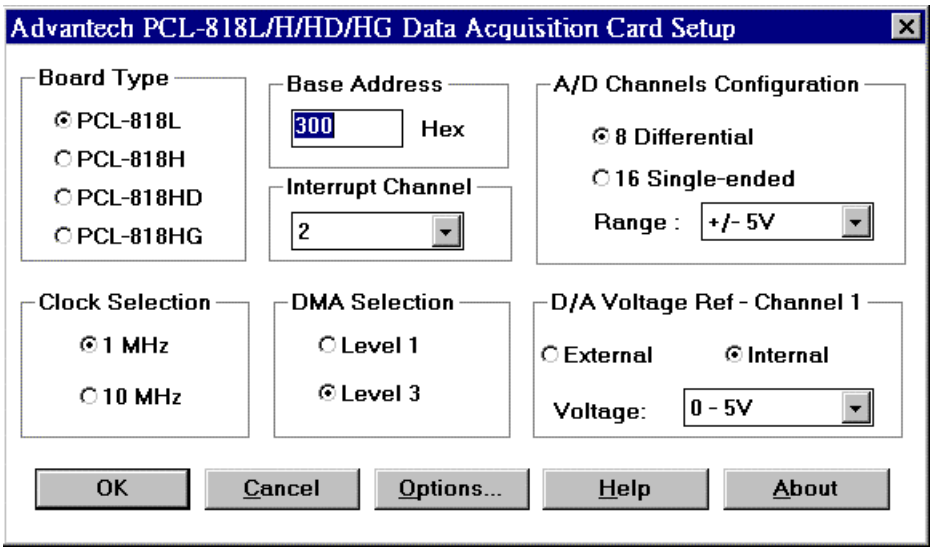
To Add and Set up each I/O device within the Advantech Device Installation program, go into the **SETUP** menu, and click **DEVICE**. You will notice a dialog box used to set up I/O devices. Choose the **ADD** button, and you should see a pop-up box (List of Devices) containing a list of the device driver type(s). Highlight the desired device and press **INSTALL**, or double click on the device. This will bring you to a device specific dialog box that allows you to software configure the device. Configure the device, and when you're satisfied with your entries, press **OK**. This will bring you back to the I/O Device Installation dialog box, where you can add another I/O device (double click, or **INSTALL**).



You can install as many I/O devices as you want. When you have completed your device installation and setup, there should be a listing reflecting your installation displayed in the **I/O Device Installation** dialog box. Once an I/O device installed, you may change its parameters by either double clicking on its entry in the **INSTALLED DEVICES** list box, or highlighting and pressing the Setup button.

Setting up or configuring a device

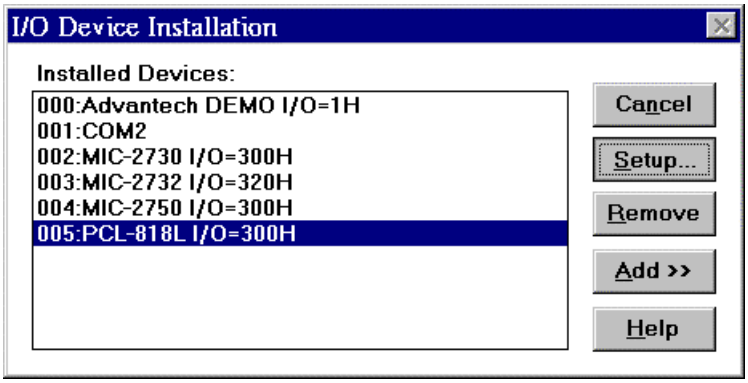
To Set up or Configure/re-configure each I/O device within the Device Installation program, go into the **SETUP** menu, and click on **DEVICE**. Highlight the previously installed I/O device you wish to set up, and press the **Setup** button. This will bring you to a device-specific dialog box that allows you to configure or re-configure the device.



Configure the device, and when you're satisfied with your entries, press **OK**. This will bring you back to the **I/O Device Installation** dialog box, where you can see a displayed list (Installed Devices) reflecting your I/O device configuration.

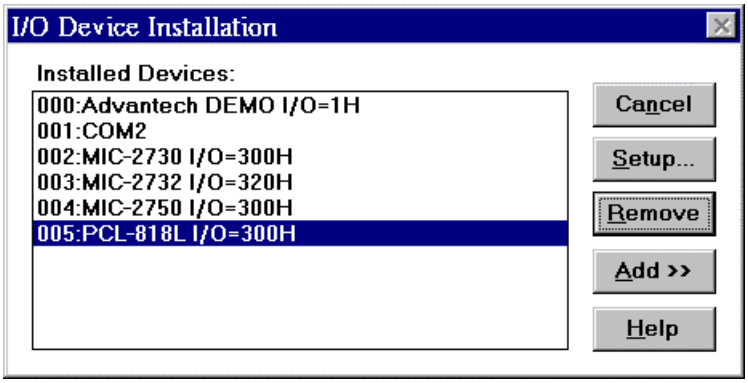
Adding a device

To Add and set up each I/O device in the Device Installation program, go into the **SETUP** menu, and click **DEVICE**. Choose the **Add** button, and you should see a pop-up box (List of Devices) containing a list displaying the device driver type(s) you have previously installed. Highlight the device and press **INSTALL**, or double click on it. This will bring you to a device-specific dialog box that allows you to configure the new device. Configure the device, and when you're satisfied with your entries, press **OK**, This will bring you back to the List of Devices box, where you can add another I/O device of the same type (double click, or **INSTALL**). When you have completed your device installation and setup, there should be a displayed list reflecting all installations in the **I/O Device Installation** dialog box.



Removing a device

To remove a previously set up I/O device within the Device Installation program, go into the **SETUP** menu, and click **DEVICE**. Highlight the device instance you wish to remove in the I/O Device Installation dialog box, and then press the **REMOVE** button. You should see that the device has been removed at this time. When you have completed your device installation and setup, there should be a displayed list reflecting all installations and removals. The **Remove** button in the I/O Device Installation dialog box will only remove each instance of the I/O device not the DLL driver itself.


I/O Device Installation
Installed Devices:
000:Advantech DEMO I/O=1H
001:COM2
002:MIC-2730 I/O=300H
003:MIC-2732 I/O=320H
004:MIC-2750 I/O=300H
005:PCL-818L I/O=300H
Cancel
Setup...
Remove
Add >>
Help

Chapter 2 Tutorials

This chapter teaches you step-by-step how to create an application in LabVIEW. It also guides you through the Search Examples feature to help you find examples in LabVIEW.

You will learn to do the following:
- Use the Search Examples feature to find and run an example
- Create a new program in LabVIEW

The VI you build generates data, analyzes it, then writes it to a file.

Tip: For more detailed information, please refer to examples in the directory path LabVIEW\Examples\Advantech. You can copy and paste the useful blocks into your programs to make programming faster and easier.

2.1 Search for Examples

In order to make you familiar with LabVIEW efficiently, the Advantech 32-bit DLL driver provides lots of examples describing how to design LabVIEW applications.

You can follow the steps below to run or view LabVIEW example programs:

Step 1. Click **Open VIs** to see a listing of LabVIEW examples.
Step 2. Click **Advantech** in the **Examples** directory.
Step 3. Double click **AI** to open this demonstration VI.
Step 4. Click the **Run** button on the front panel toolbar to see how a LabVIEW VI runs.
Step 5. Stop the program by clicking the **Stop** button.

2.2 Build a Virtual Instrument

This exercise demonstrates how simple it is to chart, analyze, and save a VI program in LabVIEW.

Create a User Interface
You can create a graphical user interface on the front panel using controls and indicators from the Controls palette.

1. Create a new VI by selecting **New VI** in the LabVIEW dialog box.
2. On the new front panel, choose **Edit>>Select Palette Set>>Basic**.
3. Select **Vertical Toggle Switch** from the **Controls>>Boolean** subpalette. Select icons from the palette in the same way you select items from a menu.
4. Move your pointer into the front panel and click to place the toggle switch on the front panel.
5. Type Power into the label for the toggle switch. If the label disappears, choose **Show Label** from the pop-up menu on the toggle switch. To access the pop-up menu, right-click on the toggle switch.

6. Create a *waveform chart* by selecting **Controls>>Graph>>Waveform Chart**. This chart plots data one point at a time.
7. Place the chart on the front panel and label it Random Plot.
8. To change the scale of the waveform chart, select the *Operating tool* from the Tools palette. If the Tools palette is not visible, select **Windows>>Show Tools Palette**. Double-click **10.0** on the Y-axis of the Random Plot indicator and enter new scale value.

Build the Block Diagram
Now you can create the source code in the block diagram of your VI.

1. Go to the block diagram by choosing **Show Diagram** from the **Windows** menu or clicking in the block diagram window. The two terminals on the block diagram correspond to the Power toggle switch and the Random Plot waveform chart on the front panel.
2. Select **Random Number (0-1)** from the **Functions>>Numeric** subpalette. If the Functions palette is not visible, select **Windows>>Show Functions Palette**. For this exercise, you use a random number generator to generate input data for your VI.
3. Place the Random Number function on the block diagram. This VI generates a random number ranging from zero to one.
4. Select **While Loop** from the **Functions>>Structures** subpalette. The *While Loop* runs all code within its boundaries while the value on its *conditional terminal* remains TRUE. When the value changes to FALSE, the While Loop exits.
5. Place your pointer in the position where you want to anchor the top left corner of the loop. Drag the loop diagonally to enclose the Random Number function, the Power toggle switch, and the Random Plot chart.

Wire and Run Your VI
You need to add wiring to define the data flow in your VI. After your VI is complete, you can run it from the front panel to see the data plot to your chart.
1. Select the *Wiring tool* from the Tools palette. If the Tools palette is not visible, select **Windows>>Show Tools Palette**.
2. Wire the Random Number function to the Random Plot chart terminal. To create a wire, click on the Random Number function, move to the Random Plot chart, and click again to terminate the wire.
3. Wire the Power toggle switch terminal to the conditional terminal on the While Loop.
4. Go to the front panel and select the **Operating** tool. Click the Power toggle switch to turn it to the TRUE position.
5. Click the **Run** button on the toolbar to run the VI.
6. To stop running the VI, click the Power toggle switch again to turn it to the FALSE position. Because the While Loop executes only as long as its conditional terminal remains TRUE, changing the value of the toggle switch to FALSE stops the loop.

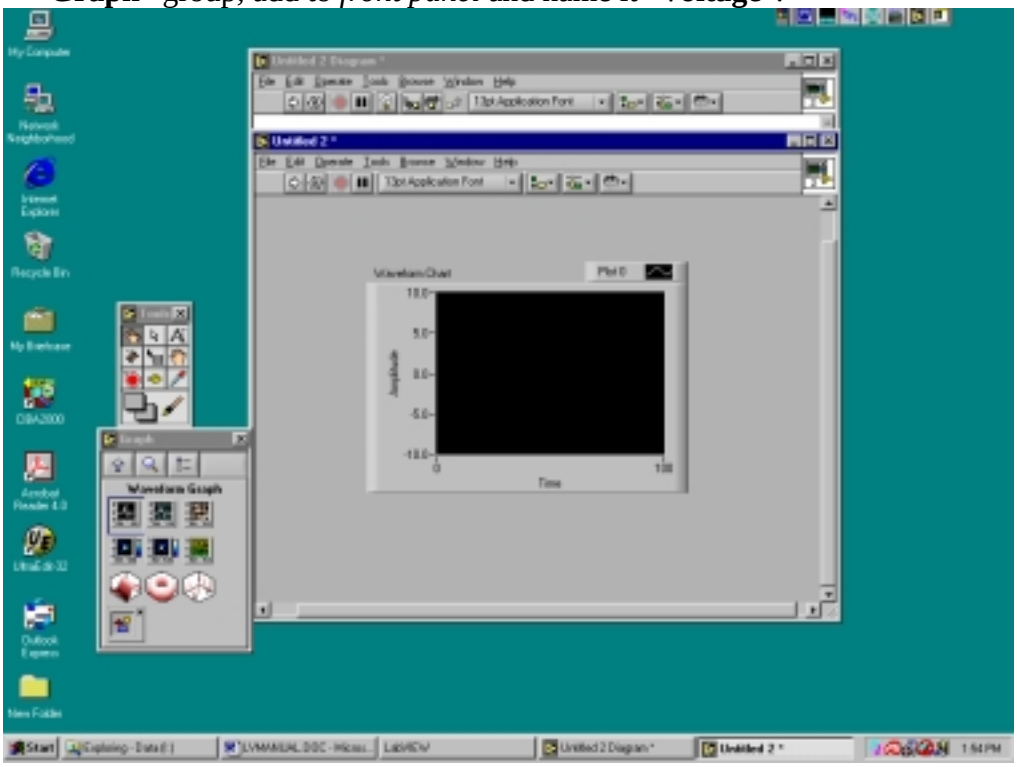
2.3 Tutorials

2.3.1 Tutorial 1 : Get voltage input from PCL-818L

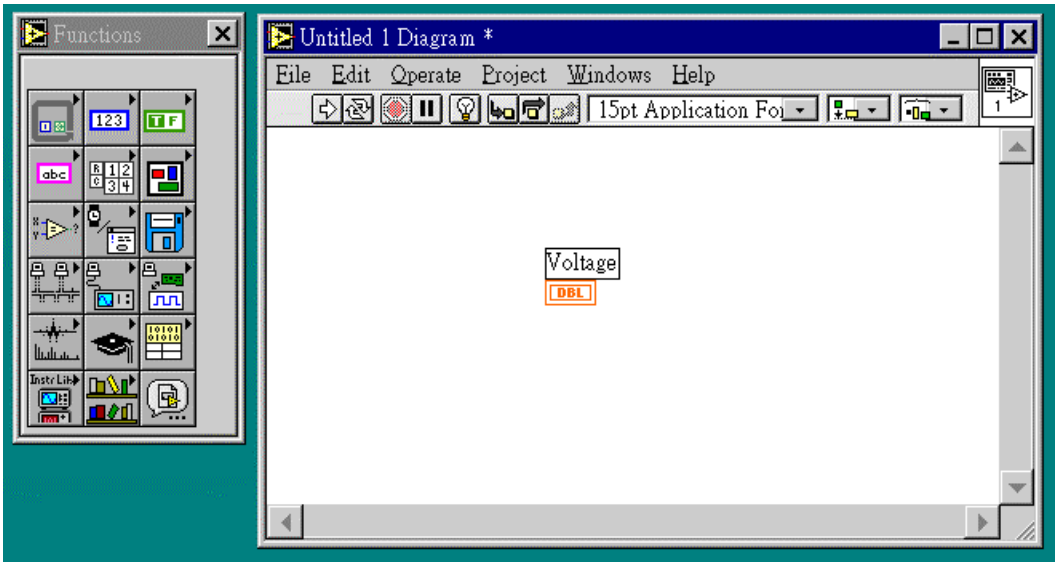
The purpose of this tutorial is to teach you how to build a LabVIEW application to retrieve voltage input values from Advantech PCL-818L data acquisition and control cards through Advantech 32-bit LabVIEW driver.

1. Click "**New VI**" button to create a new LabVIEW program

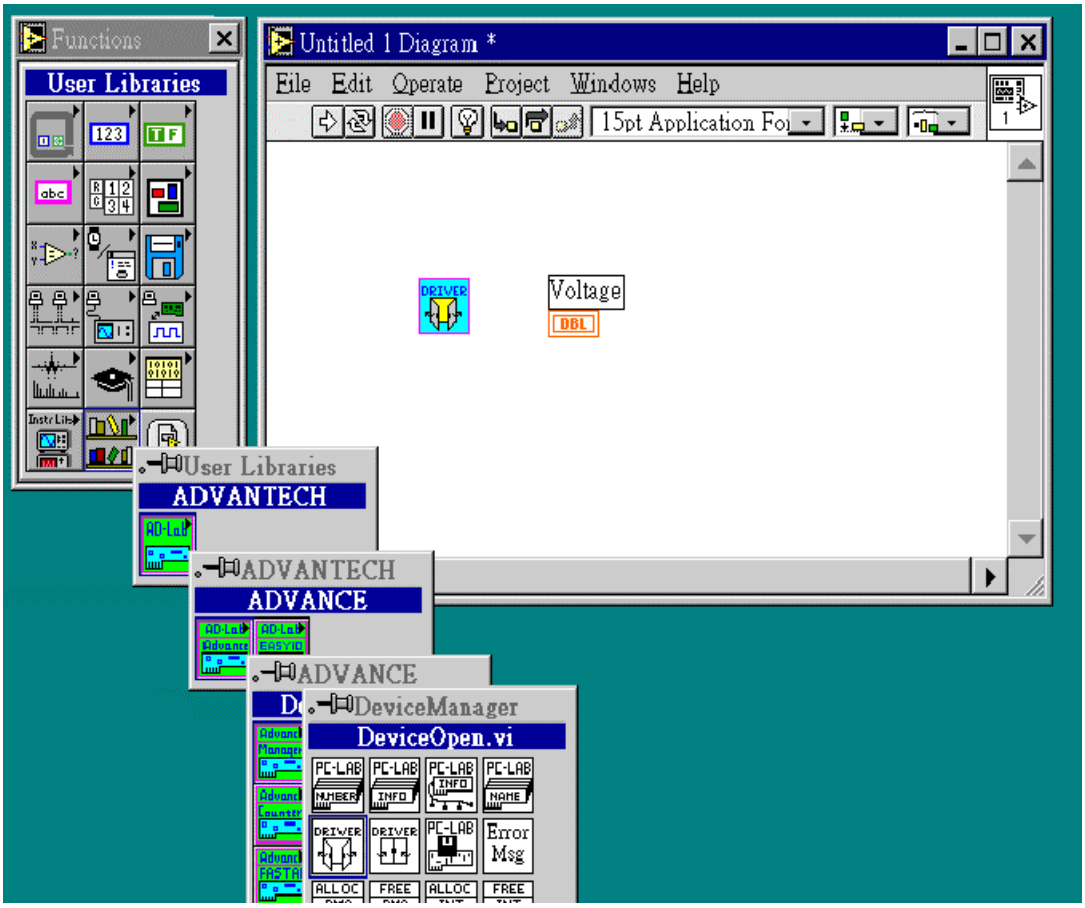
2. Select *Front Panel* and enable *Controls Palette* to choose a "**Waveform Chart**" indicator in "**Graph**" group, add to *front panel* and name it "**Voltage**".



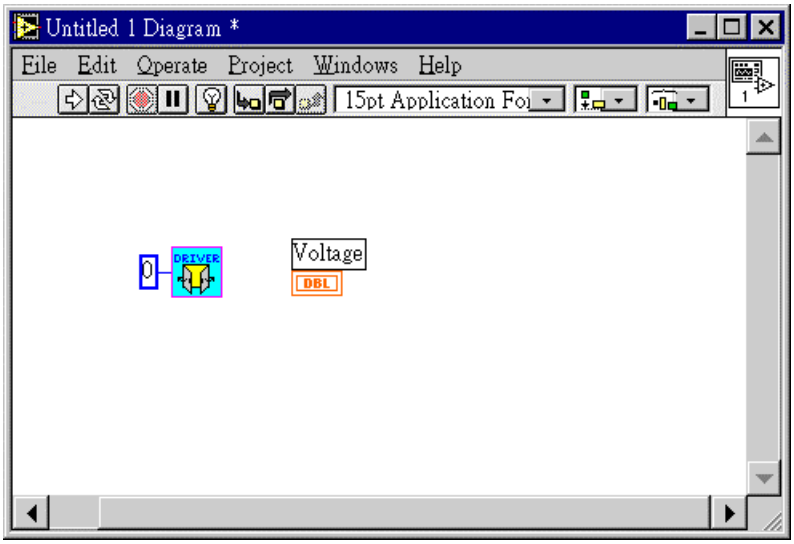
3. Switch to *Block Diagram* to verify that a new data terminal was created (named "**Voltage**").



4. Add a "**DeviceOpen**" VI from the Advantech 32-bit LabVIEW driver by clicking on *user defined* VIs. The path to get "**DeviceOpen**" VI is "**User Defined VIs" >> "Advantech" >> "Advance" >> "DeviceManager" >> "DeviceOpen"**.

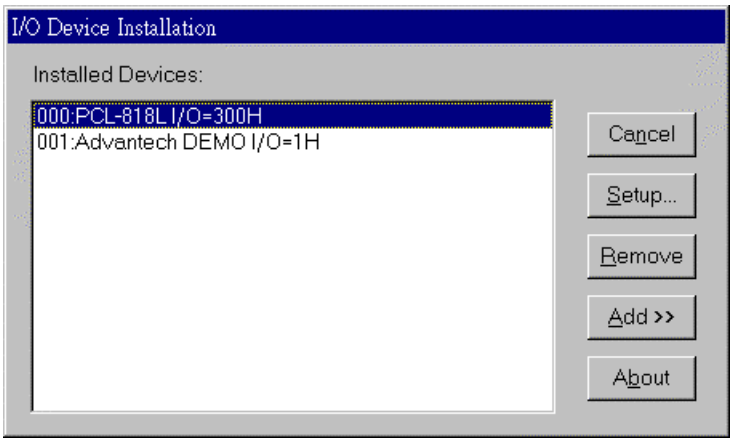


5. Move the mouse to the "**DeviceOpen**" VI and click on the mouse left button to invoke the "**Create Constant**" function. After creating a constant, a number will appear at the left of



"**DeviceOpen**" VI block.

6. Execute the Advantech *device installation* utility (within Advantech DLL driver) from the Windows environment to check with the device number of Advantech DEMO I/O virtual demo board. For example, the device number of Advantech DEMO I/O in the screen shown below is

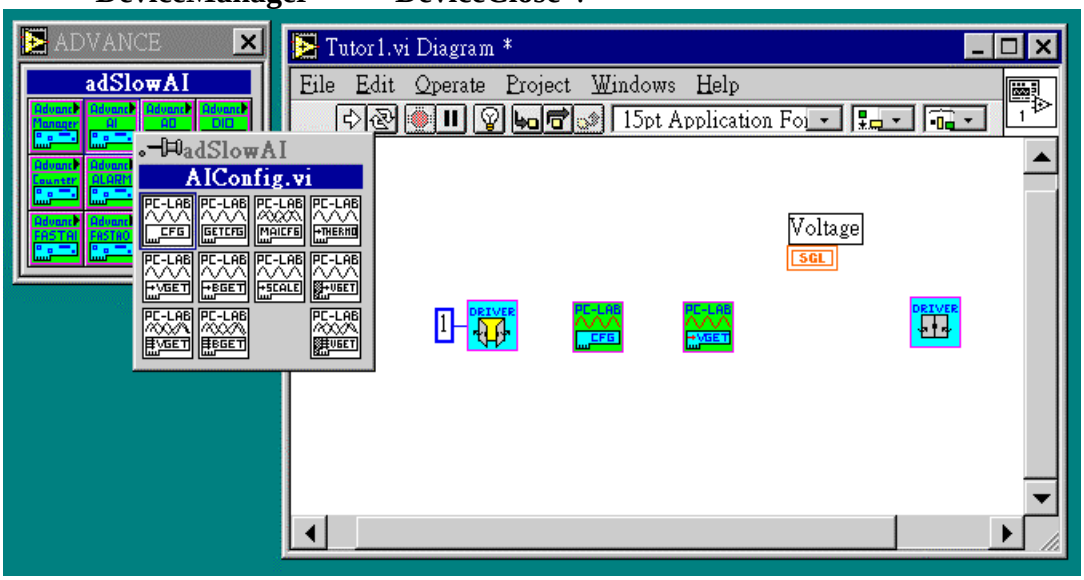


001.

7. Change the value of the constant next to the "**DeviceOpen**" VI to be **1**.

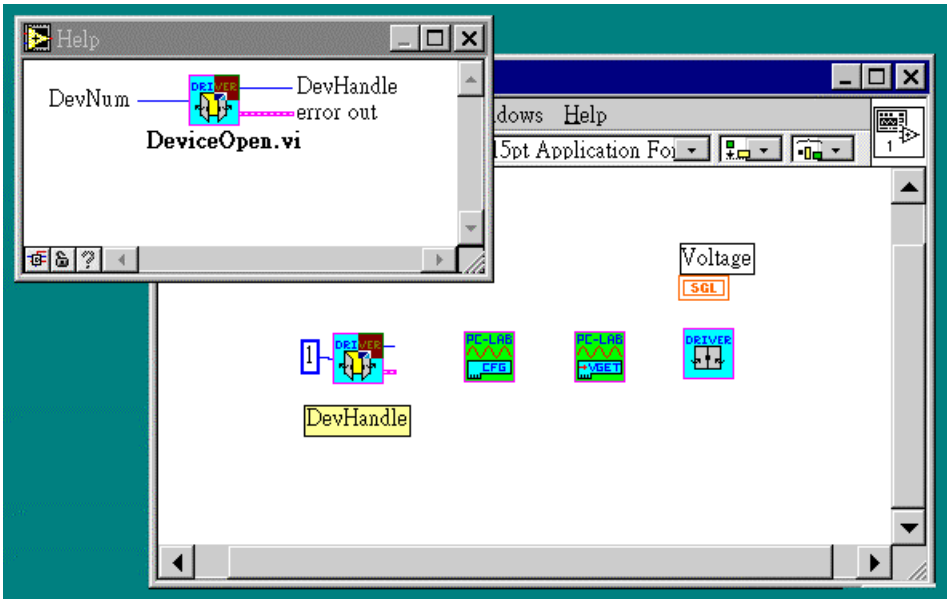
8. Add "**AIConfig**" and "**AIVoltageIn**" VIs from the Advantech 32-bit LabVIEW driver by clicking on user defined VIs. The path to get "**AIConfig**" and "**AIVoltageIn**" VI is **"User Defined VIs"** >> **"Advantech"** >> **"Advance"** >> **"adSlowAI"** >> **"AIConfig"** and **"AIVoltageIn"**.

9. Add “**DeviceClose**” VI from the Advantech 32-bit LabVIEW driver by clicking on User Defined VIs. The path to get “**DeviceClose**” VI is **“User Defined VIs”** >> **“Advantech”** >> **“Advance”** >> **“DeviceManager”** >> **“DeviceClose”**.

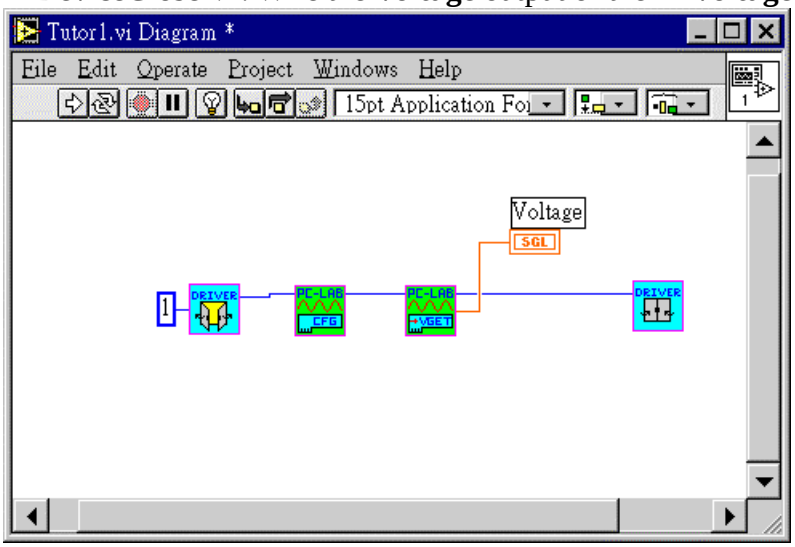


10. Change the mouse cursor for use with the “**Wiring**” pattern (move the cursor into *Tools Palette* and select the **Wiring** icon).

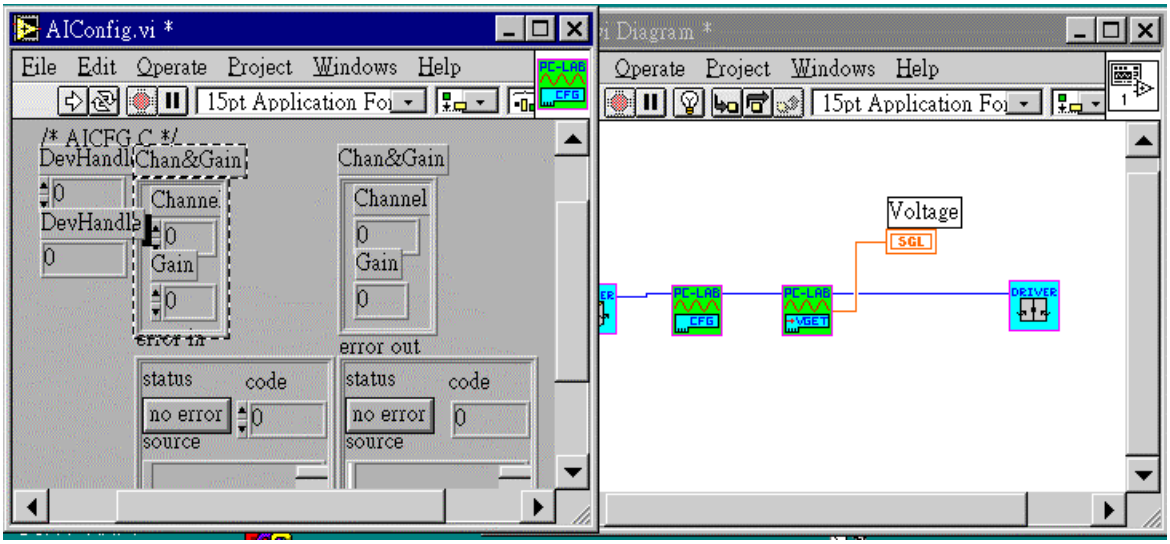
11. Wire the **DeviceHandle** output of the **DeviceOpen** VI to the **DeviceHandle** input of **AIConfig** VI. To make wiring easier, you can invoke the Help window to support it. The Help window shows input and output parameters of each selected VI. (The way to display the **Help** window is to click on the **Help** menu selection and select the “**Show Help**” option)

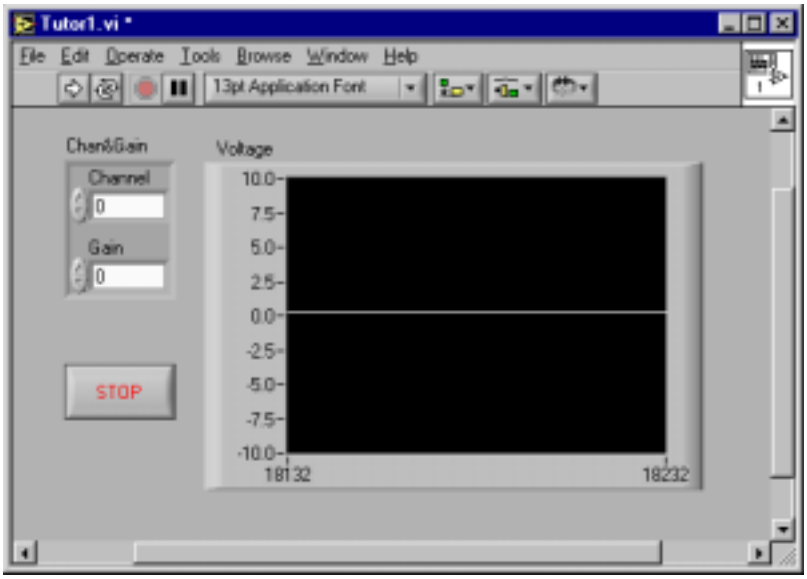


12.Wire the **DeviceHandle** output of the **AIConfig** VI to the **DeviceHandle** input of **AIVoltageIn** VI.Wire the **DeviceHandle** output of **AIVoltageIn** VI to the **DeviceHandle** input of the **DeviceClose** VI. Wire the **Voltage** output of the **AIVoltageIn** VI to the **Voltage** data terminal.

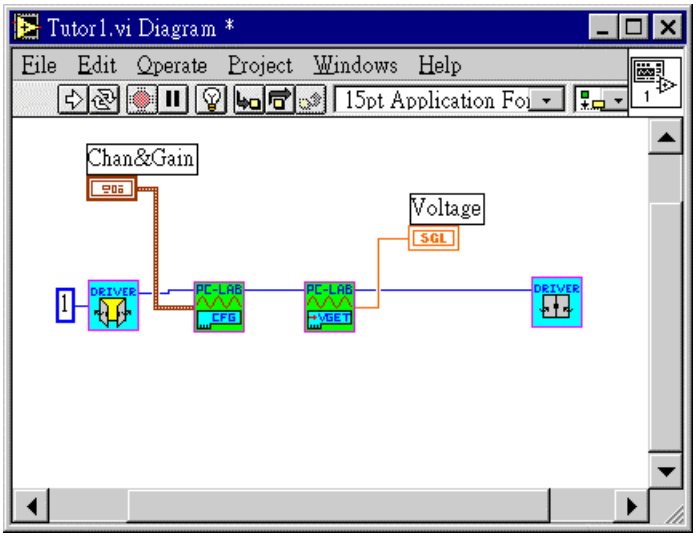


13.For the **AIConfig** VI, "**Channel**" and "**Gain**" are necessary input values. In order to provide "**Channel**" and "**Gain**" values for the **AIConfig** VI, you can add two controls items in *Front Panel* and pass their values to the **AIConfig** VI in *Block Diagram*. The simple way to do this is to copy existing control items from **AIConfig**. First, change the cursor from "**Wiring**" mode to "**Indicator**" mode. Then, move the cursor to the **AIConfig** VI and double-click on it. Another *Front Panel* for the **AIConfig** VI will appear on the screen. Move the cursor to the "**Chan&Gain**" field and draw a loop around the field. Use **"Edit">>"Copy"** to copy "**Channel**" and "**Gain**" control items and paste them on the *Front Panel*.

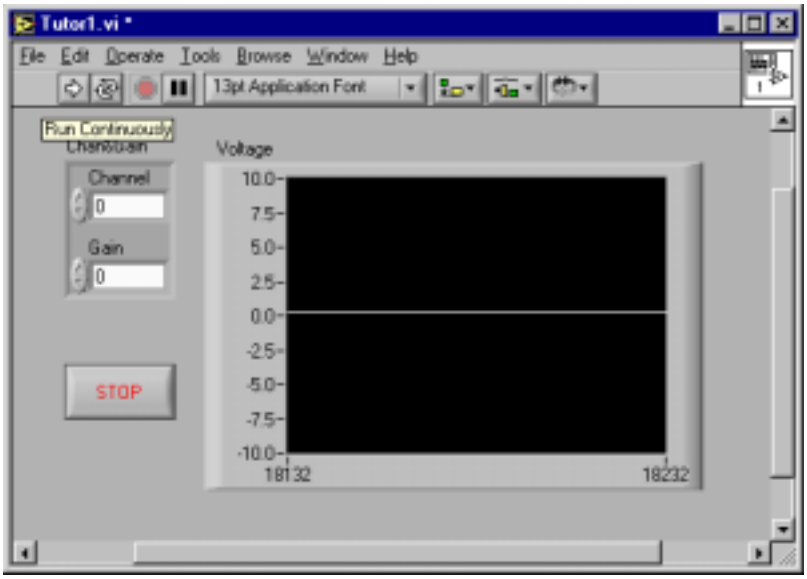




14. After enabling the *Block Diagram*, you will find a "**Chan&Gain**" data terminal displayed on the *Block Diagram*. Wire the output of the "**Chan&Gain**" data terminal to the input "**Chan&Gain**" of the **AIConfig** VI.

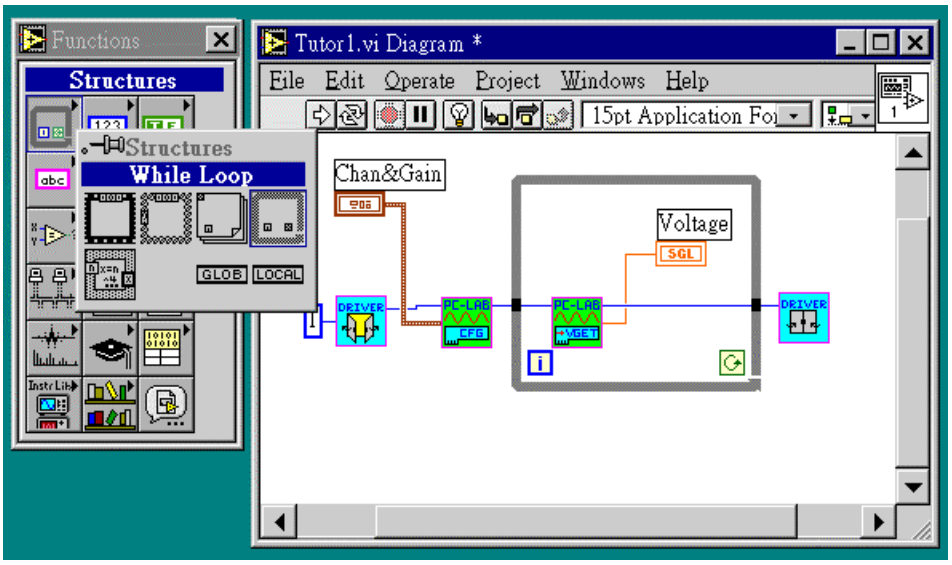


15.The basic steps to get **AIVoltageIn** data are now complete. You can change the window to the *Front Panel* window and click the "**Run**" or "**Run Continuously**" button to execute **AIVoltageIn** VI.

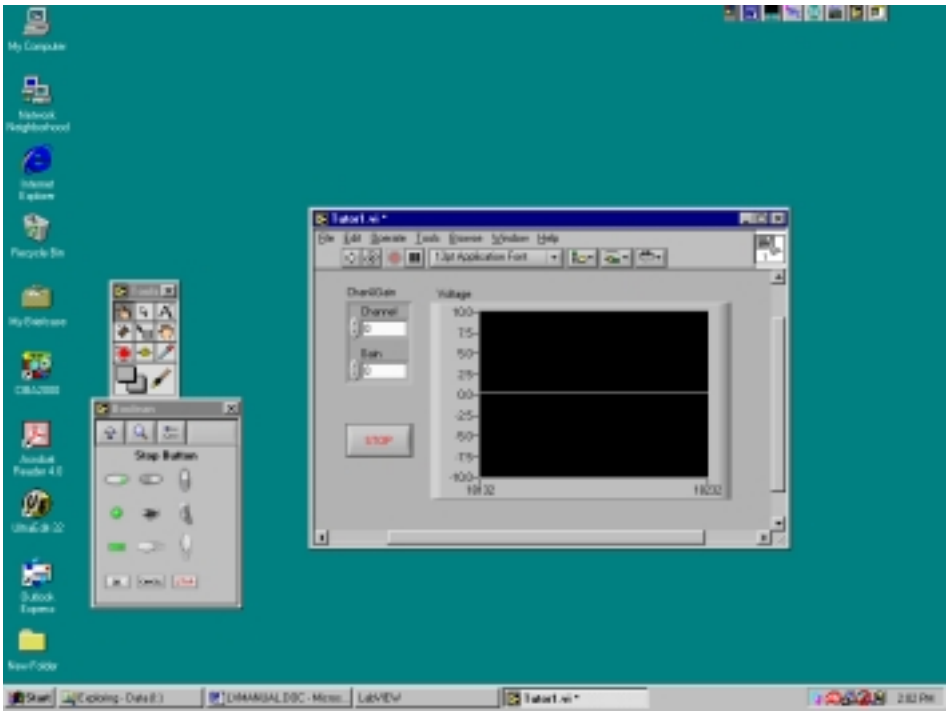


16.If your waveform chart does not show the whole graph, you can change the scale of the Y-axis. To change the Y axis scale on the waveform chart, move the cursor to the graphic origin, click the left mouse button, and type in the desired value.

17. The "**Run Continuously**" button executes the **AIVoltageIn** VI continuously. You can also add a "**While Loop**" to enhance your program. First, move mouse to *Functions Palette* and select the **Structures** group and the **While Loop** VI. Add the **While Loop** VI to the *Block Diagram* and enlarge it to include the **AIVoltageIn** VI and the **Voltage** data terminal.

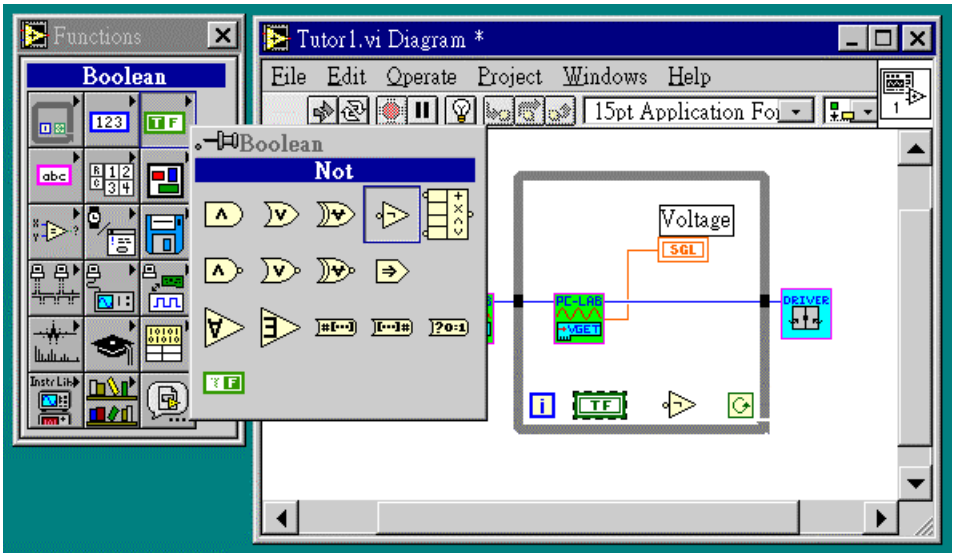


18. Add a **Stop** button to control the execution of the **While Loop**. Enable the *Front Panel* and move the mouse to *Controls Palette*. Select the **Boolean** and **Stop** Button to add a **STOP** button to the *Front Panel*.



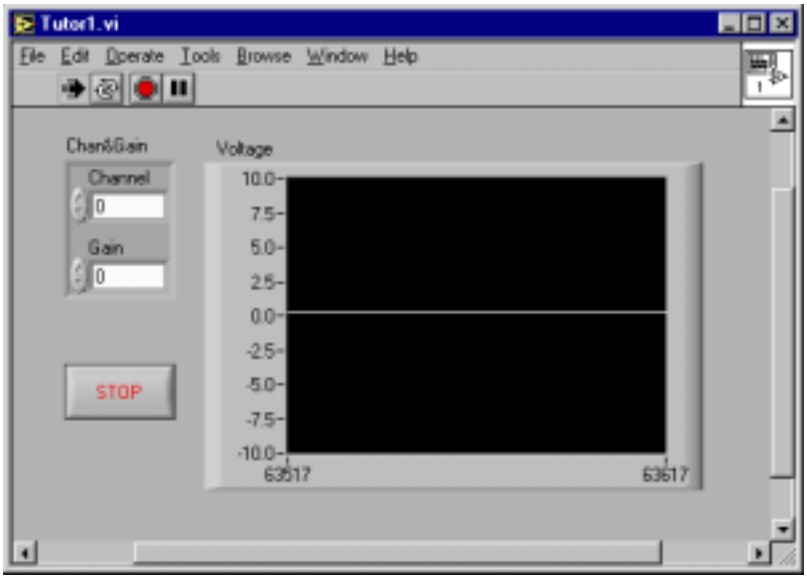
19. Switch to *Block Diagram*. A **TF** VI should be showing on the Diagram. Move the **TF** VI into the **While Loop** VI and add a **NOT** VI from *Functions Palette*. Move **NOT** VI to the location

between **TF** VI and **COND** VI.



20. Wire the output of the **TF** VI to the input of the **NOT** VI and wire the output of the **NOT** VI to the input of the **COND** VI. After wiring, switch to *Front Panel* and press the **RUN** button to execute.

21. If you want to get voltage data from a given device, just change the constant number of the **DeviceOpen** VI to be the device number of the device of interest. For example, the device number of the PCL-818 in this tutorial is **0**.

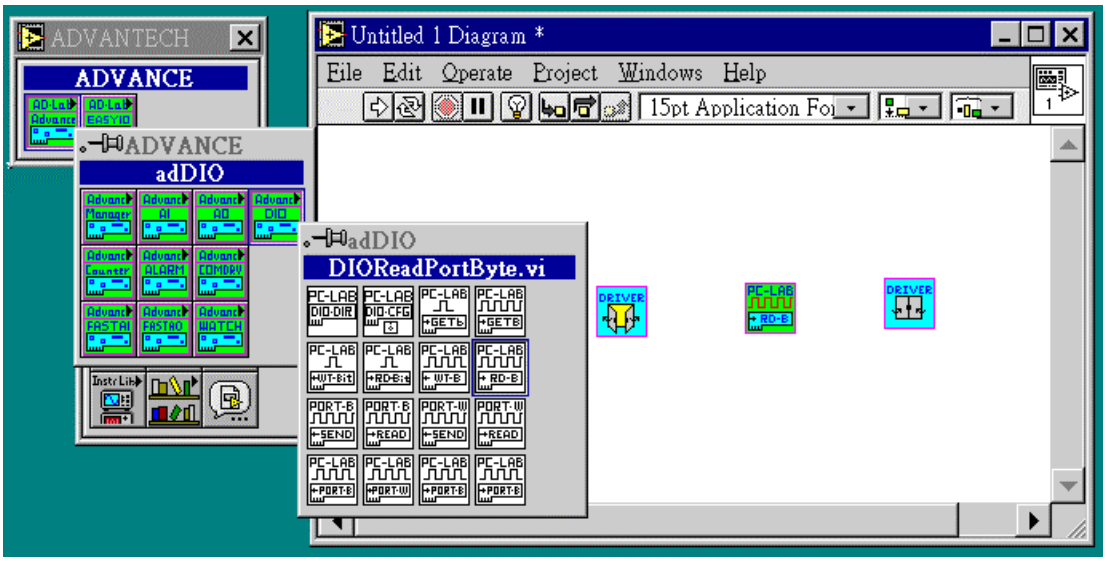


2.3.2 Tutorial 2: Get Digital Input Data from PCL-818

The purpose of this tutorial is to teach you how to build a LabVIEW application to retrieve digital input values from Advantech PCL-818L data acquisition and control cards through an Advantech 32-bit LabVIEW driver.

1. Click "**New VI**" button to create a new LabVIEW program

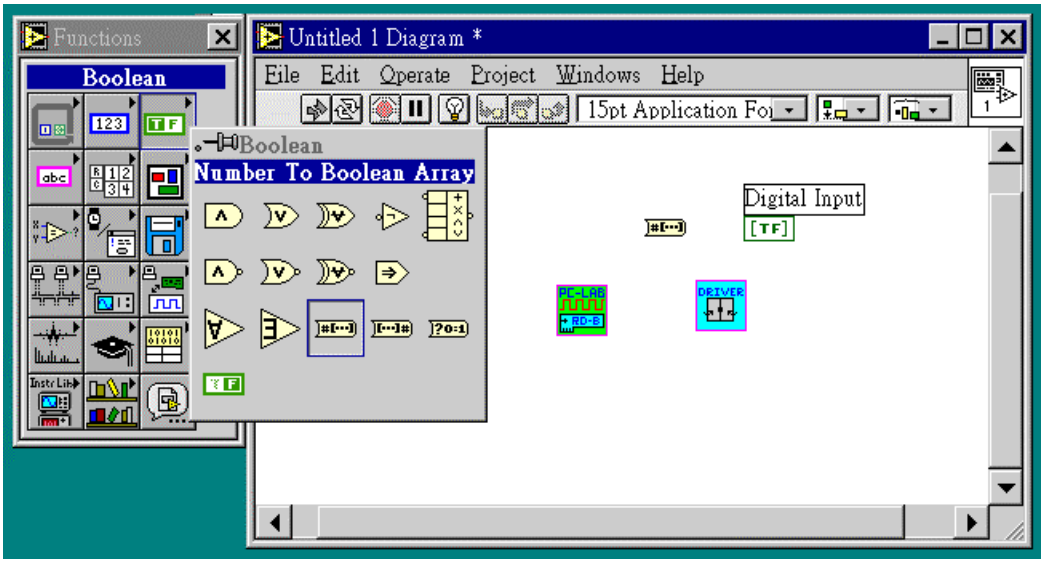
2. Switch to *Block Diagram*. Add the "**DeviceOpen**" and the "**DeviceClose**" VIs from Advanced VIs and the "**DeviceManager**" group. Add the "**DIOReadPortByte**" VI from Advanced VIs and "**adDIO**" group.



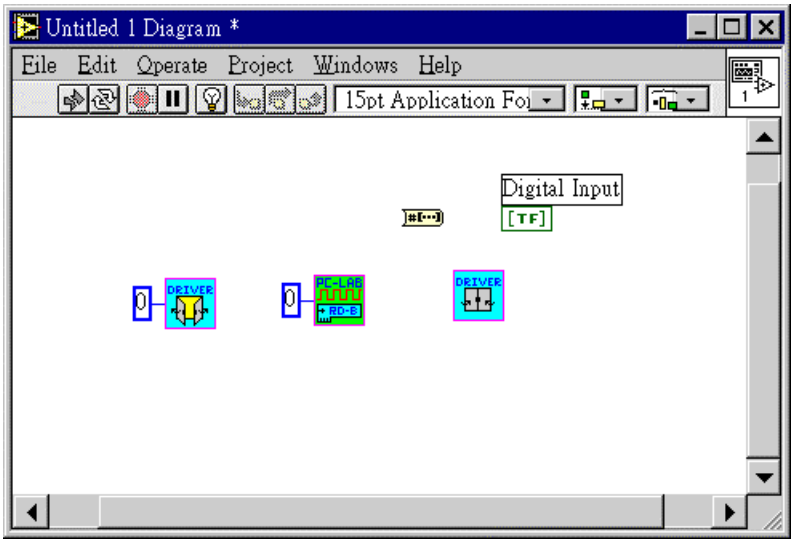
3. Switch to the *Front Panel*. Add a **Round LED** from **Boolean** group and add an **Array** from the **Array & Cluster** group.

4. Move the **Round LED** icon into the **Array** icon and enlarge the **Array** icon to contain 8 round LEDs.

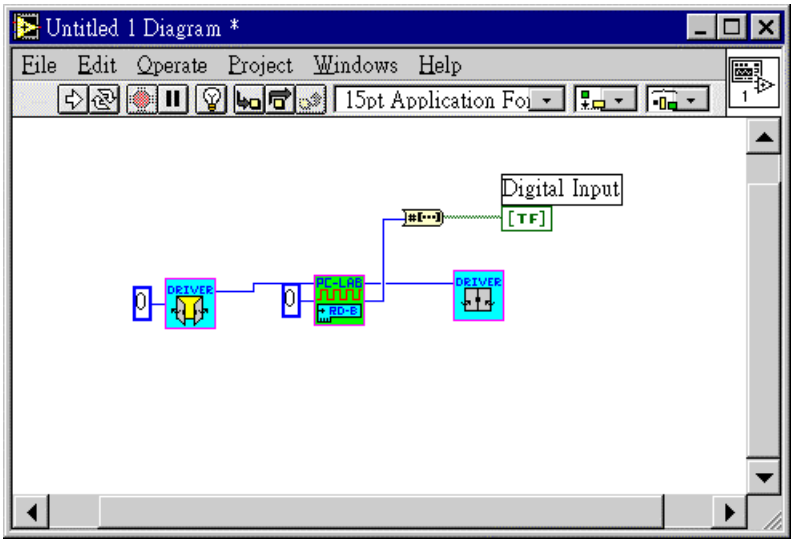
5. Switch to the *Block Diagram* window. You will find the **TF** data terminal (named **Digital Input**) mapped to the **Round LED** icon. Add a **NumbertoBinaryArray** VI from the **Boolean** group in the *Functions Palette*.



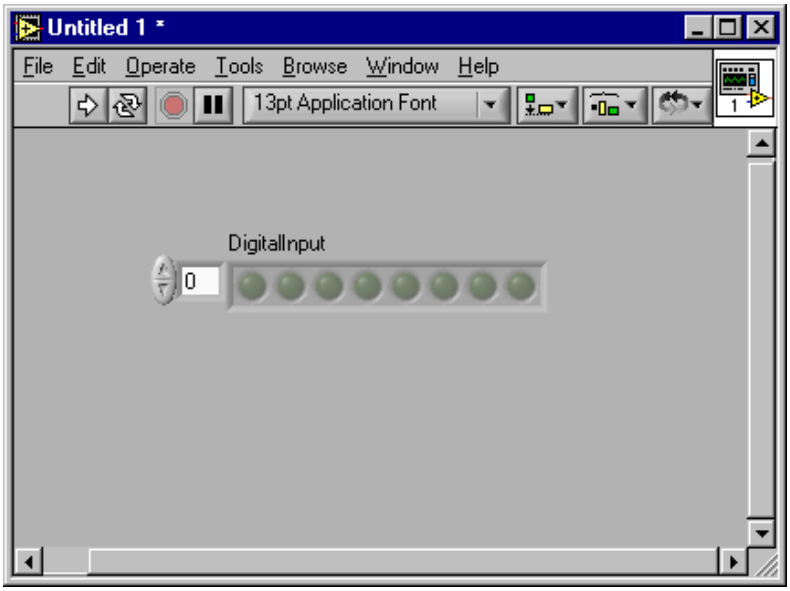
6. Create a constant for the device number of the **DeviceOpen** VI. Create another constant for the port of the **DIOReadPortByte** VI.



7. Wire the **DeviceHandle** output of the **DeviceOpen** VI to the **DeviceHandle** input of the **DIOReadPortByte** VI. Wire the **DeviceHandle** output of the **DIOReadPortByte** VI to the **DeviceHandle** input of **DeviceClose** VI. Wire the state output of the **DIOReadPortByte** VI to the input of **NumbertoBinaryArray** VI. Wire the output of **NumbertoBinaryArray** VI to the input of **TF** data terminal.



8. Compare the device number of the **Advantech DEMO I/O** with the device number value of the **DeviceOpen** VI. After checking, press the **Run** button to execute this tutorial.

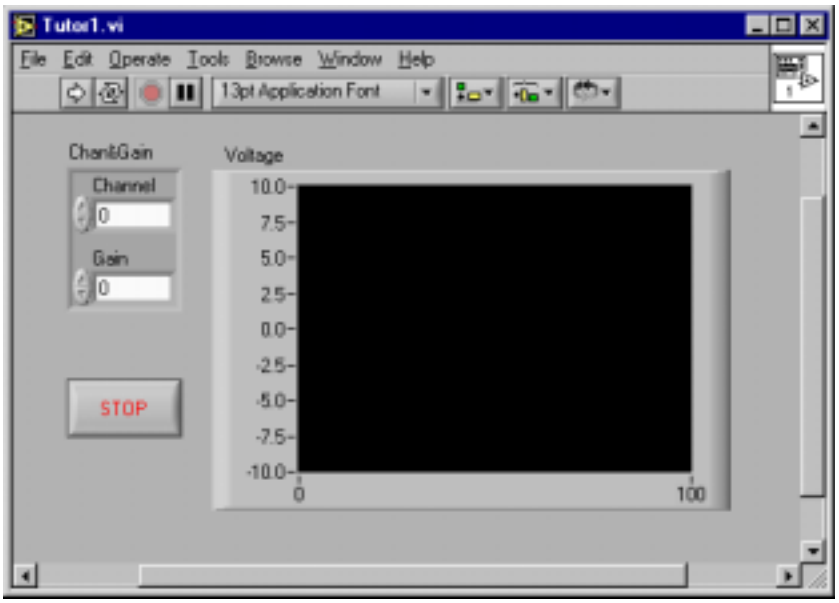


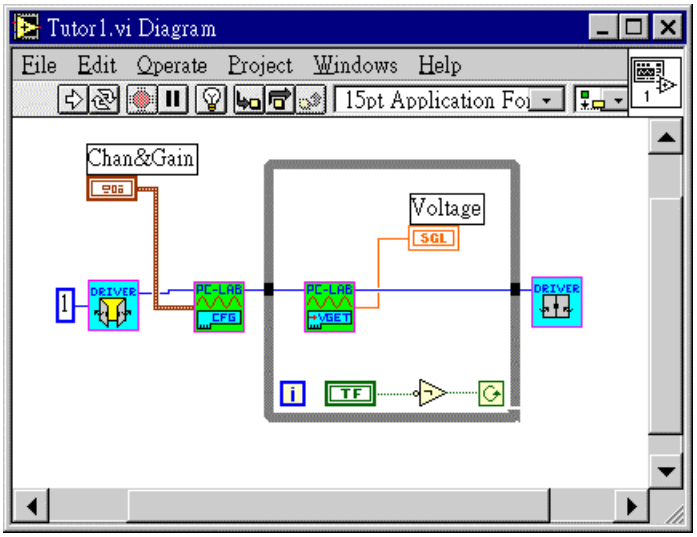
2.3.3 Tutorial 3: Programming with Advantech EASY I/O VIs

The purpose of this tutorial is to teach you how to build a LabVIEW application through EASY I/O VIs to retrieve digital input value from Advantech PCL-818L data acquisition and control cards.

1. First, you can open "**Tutor1.vi**" to continue with Tutorial 1 "*Get voltage input from PCL-818L*". If you did not do Tutorial 1, please refer to Tutorial 1 to implement "**Tutor1.vi**"

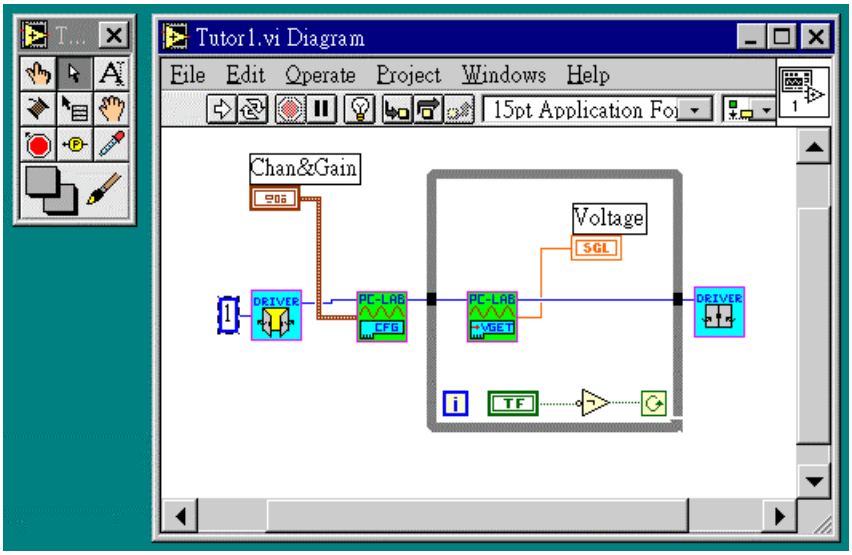
2. After opening "**Tutor1.vi**", the *Front Panel* and *Block Diagram* windows will appear as shown below:





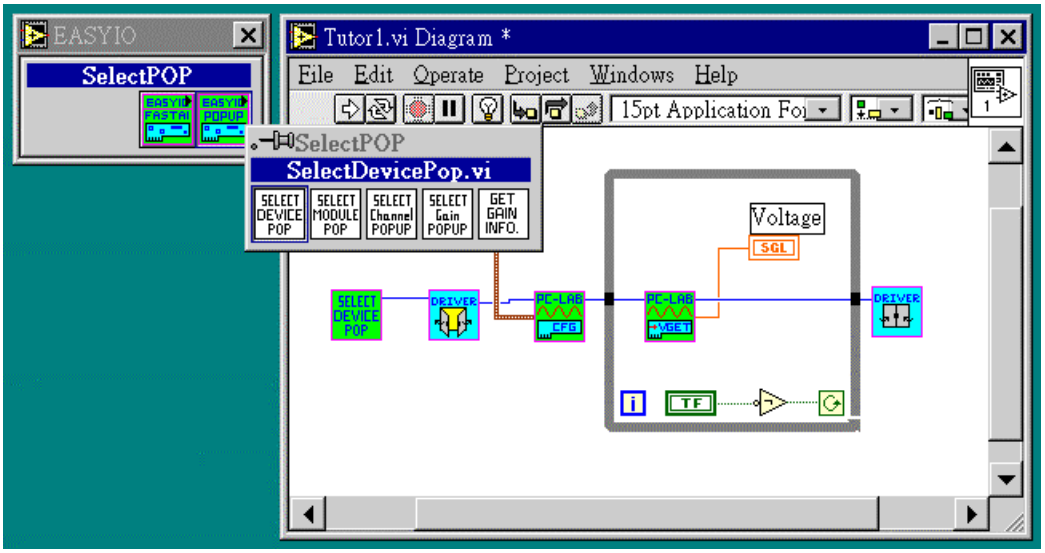
In **Tutorial 1**, you have to assign a constant value for the device number of the **DeviceOpen** VI before executing **Tutorial 1**. That means, you can not change the device while **Tutorial 1** is executing. Now, you can use the **SelectDevicePop** VI from EASY I/O VIs to select the device during execution.

3. Switch to the *Block Diagram* window. Move the mouse to the *Tools palette* and select indicator mode for the cursor.

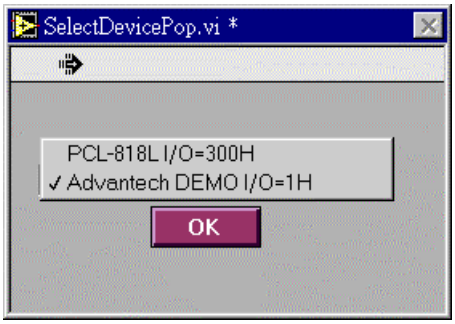


4. Delete the constant number next to the DeviceOpen VI.

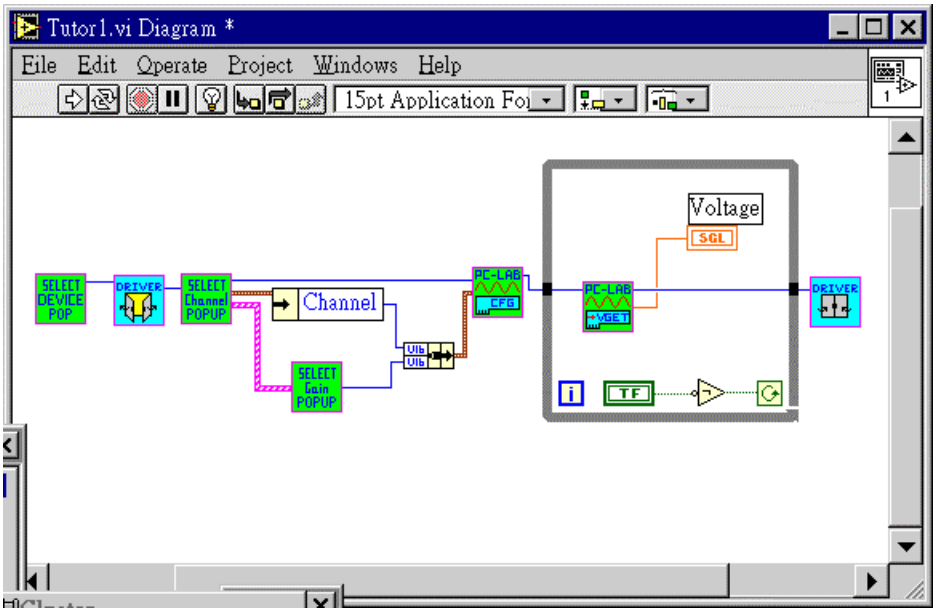
5. Move the mouse to the *Functions Palette* and select the **SelectDevPop** VI from the **SelectPop** group of the EASY I/O VIs. Add the **SelectDevPop** to the *Block Diagram* and wire it to the *DeviceOpen* VI.



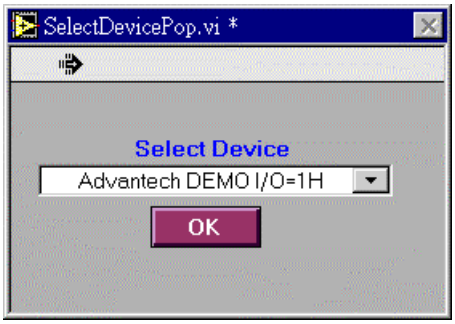
6. Now, you can run **Tutorial 1** immediately. A pop-up panel will be displayed on screen for selecting devices.

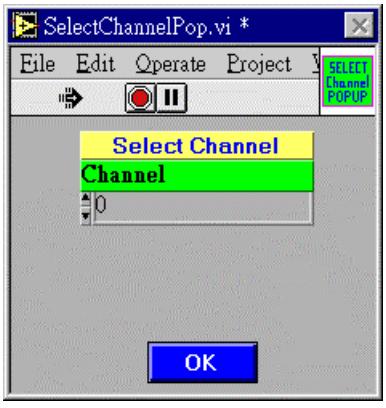


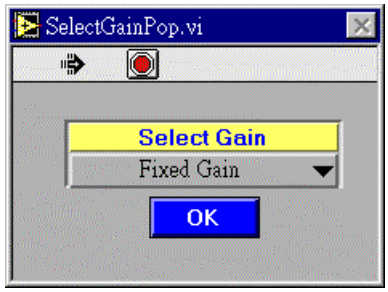
7. Proceed as in step 5 to add the **SelectChanPop** VI and the **SelectGainPop** VI. Due to parameter transformation of channel and gain data, you will need to add the "**Unbundled by Name**" VI for retrieving channel values from "**channel info**" parameter and a "**Bundled**" VI to combine channel and gain codes into a cluster "**chan&gain**" for the **AIConfig** VI.



8. Press the "**Run**" button to execute the tutorial 3 program. Three pop-up windows will appear for selecting device, channel and gain:





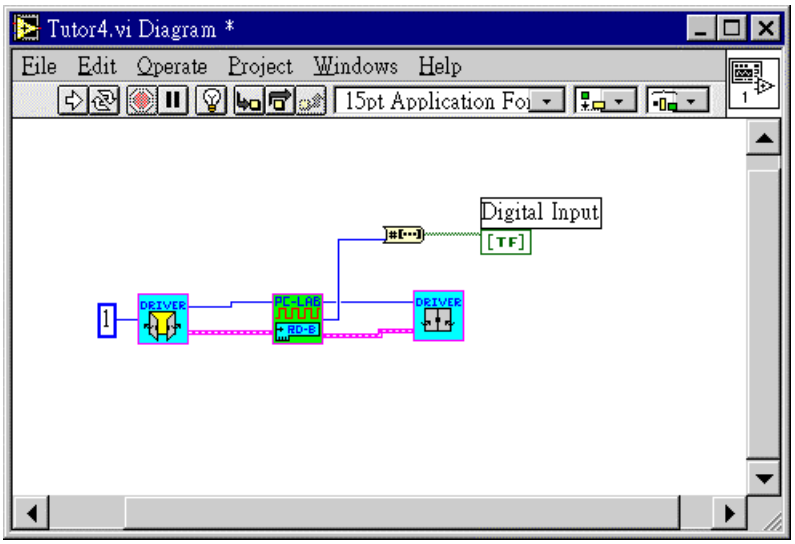

SelectGainPop.vi
Select Gain
Fixed Gain
OK

2.3.4 Tutorial 4. Error handling for error message

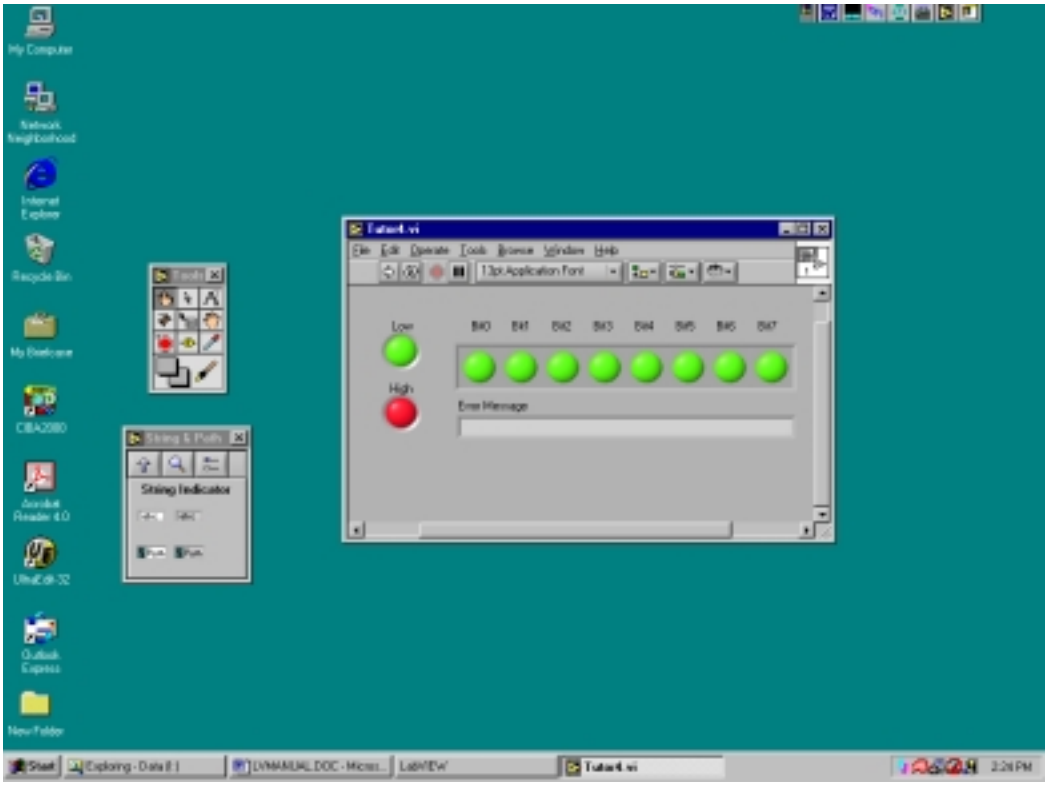
The purpose of this tutorial is to teach you how to apply the error handling function (error in and error out) of each Advantech VI to debug your program.

1. First, you can open "**Tutor2.vi**" to continue with Tutorial 2 "*Get voltage input from PCL-818L*". If you did not do Tutorial 2, please refer to Tutorial 2 to implement "**Tutor2.vi**".

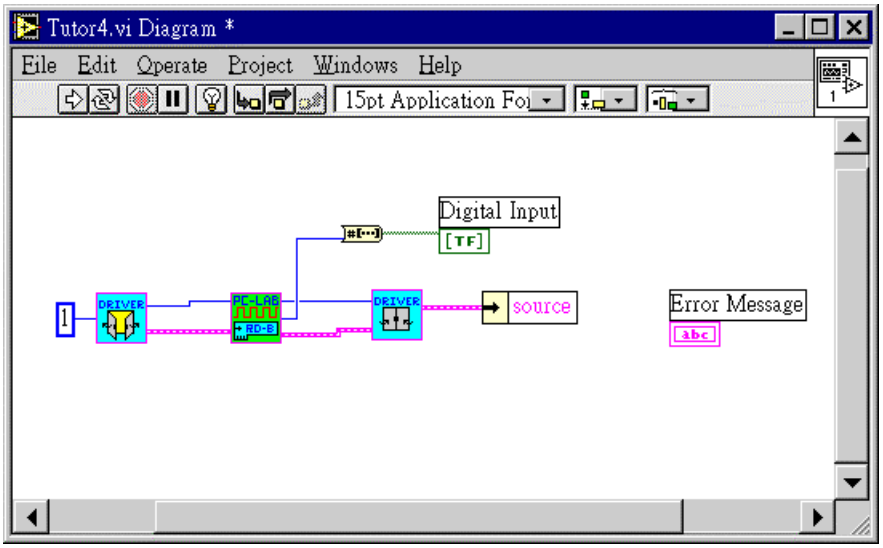
2. Wire the error out parameter of the **DeviceOpen** VI to the error in parameter of the **DIOReadPortByte** VI. Additionally, wire the error out parameter of the **DIOReadPortByte** VI to the error in parameter of the **DeviceClose** VI.



3. Switch to *Front Panel*. Add a **String Indicator** VI to *Front Panel* for displaying error messages during error handling.

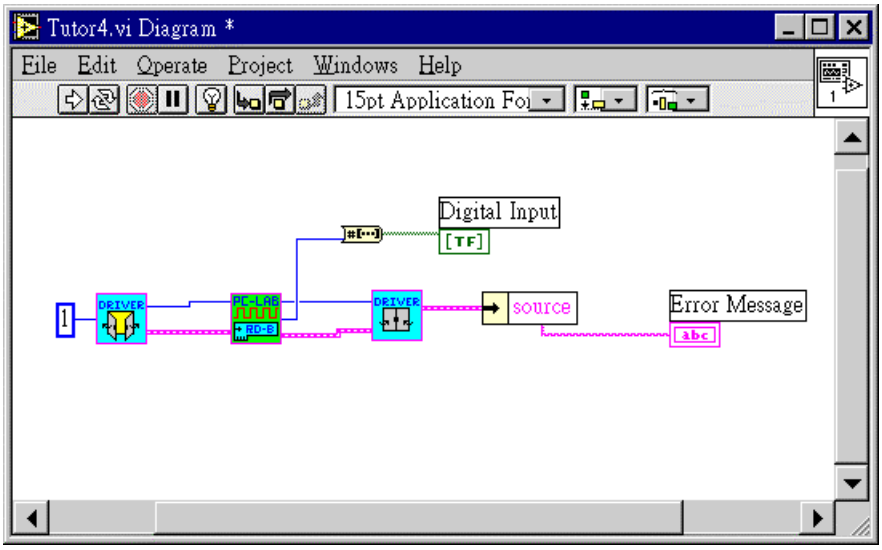


4. Switch to the *Block Diagram* window. You will find the **Error Message** data terminal added to the *Block Diagram*. Move the data terminal to the end of the **DeviceClose** VI. Insert an **UnbundledByName** VI between the **DeviceClose** VI and the **ErrorMessage** data terminal and wire the error out parameter to the input of the **UnbundledByName** VI.



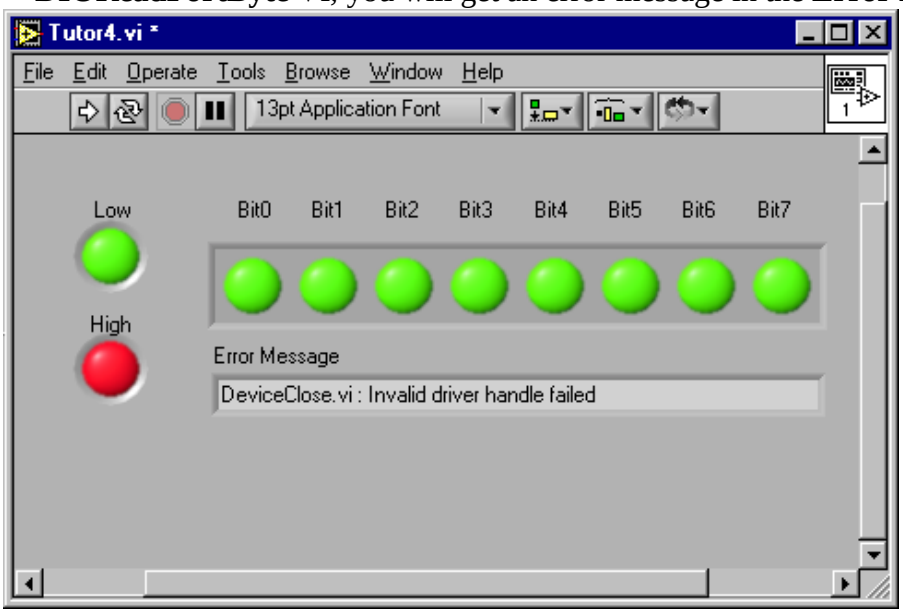
5. Move the cursor to the **UnbundledByName** VI and click on the right button of the mouse to invoke the **Selectitem** option. At the **Selectitem** option, choose "**source**" as the output item.

6. Wire the output of the **UnbundledByName** VI to the input of the **ErrorMassge** data terminal to



display messages in the *Front Panel*.

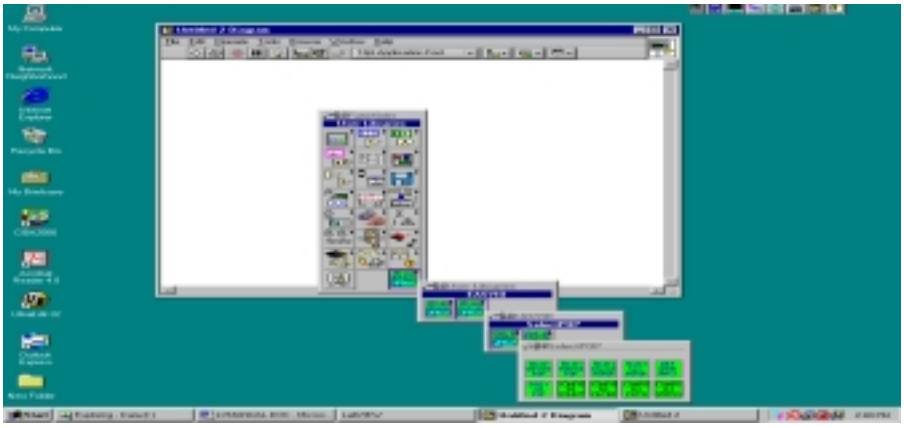
7. Press the **Run** button to execute Tutorial 4 program. You will find that the **Error Message** block at the *Front Panel* is blank. But if you delete the wiring between the **DeviceOpen** VI and the **DIOReadPortByte** VI, you will get an error message in the **Error Message** block.



Chapter 3 Easy I/O VIs

This chapter contains reference descriptions of the EASY I/O VIs, which perform simple channel/device operations and Fast AI channel operations. You can use these VIs from the front panel or use them as subVIs in basic applications.

The EASY I/O VIs are actually composed of intermediate VIs, which are in turn composed of advanced VIs. The EASY I/O VIs provide a simple, convenient interface with only the most commonly used inputs and outputs.

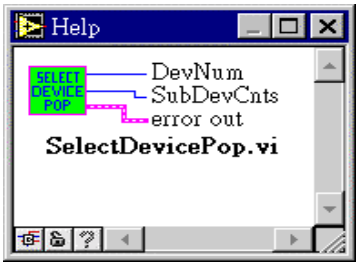


EASY I/O VIs	
Group	VI
SelectPOP	- SelectDevicePop -ConfigDICountPop - SelectSubDevicePop -ConfigDIFilterPop - SelectChannelPop -ConfigPatternPop - SelectGainPop -ConfigStatusPop - GetGainInfo - EasyAIPopup
EasyFastAI	- SingleChannelINTSetup - SingleChannelDMASetup - MultiChannelINTSetup - MultiChannelDMASetup - MultiChannelDMAExSetup - WaitFastAIEvent - EasyFAICheck - SingleChanDualDMASetup - MultiChanDualDMASetup - MultiChannelWatchdogINTSetup - MultiChannelWatchdogDMASetup - OverrunHandler - BufferChangeHandler - Split1DArray - EasyDualFAICheck

3.1 SelectPop

3.1.1 SelectDevicePop

Displays pop-up panel for selecting an installed device and outputting device number and the number of sub-devices linked to the device. If error occurs, error information will be passed through the ***error out*** parameter.



Input:

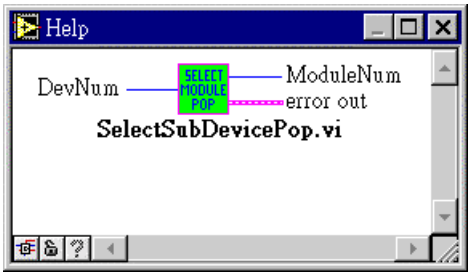
None

Output:

Data type	Parameter	Description
U32	***DevNum***	The device number of the hardware installed in the system by "**DEVINST.EXE**"
U16	***SubDevCnts***	Specifies the number of the expansion or daughterboards (if an expansion board has been installed). If no expansion board is installed, this value will be **zero**.
error	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

3.1.2 SelectSubDevicePop

Displays pop-up panel for selecting installed device by its device number, and outputting the module numbers of the expansion device or daughterboard. If error occurs, error information will be passed through ***error out*** parameter.



Input:

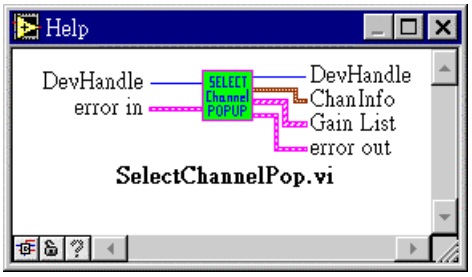
Data type	Parameter	Description
U32	***DevNum***	The device number of the hardware installed in the system by "**DEVINST.EXE**"

Output:

Data type	Parameter	Description
U16	***ModuleNum***	Specifies the module number of the expansion or daughterboard that's installed at the specified device (specified by input device number).
906	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

3.1.3 SelectChannelPop

Displays pop-up panel for selecting the channel of an installed device and outputting selected channel information and the gain list of the specified device. If error occurs, error information will be passed through the ***error out*** parameter.



Input:

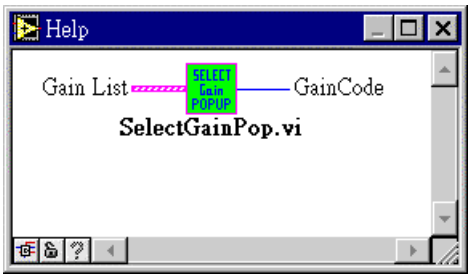
Data type	Parameter	Description
U32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
error	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
U32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program..
cluster	***Chan Info***	The ***Chan Info*** parameter includes **Channel number**, **ExpBrdID**, and **Expchannel** data. The **Channel number** specifies the digital or analog channel number. **ExpBrdID** specifies the ID of an expansion board. **Expchannel** specifies the channel number of the expansion board.
array	***Gain List***	An array of gain code for the device previously specified.
error	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

3.1.4 SelectGainPop

Display pop-up panel for selecting the gain code from a gain list for the device specified at a previous VI.



Input:

Data type	Parameter	Description
[ ]	***Gain List***	An array of gain code for the previously specified device.

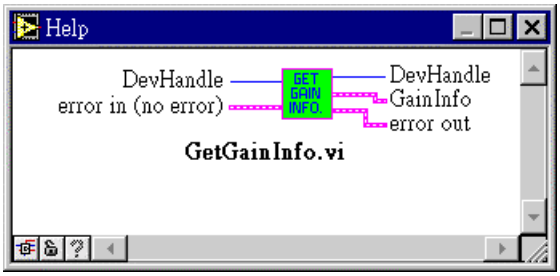
.

Output:

Data type	Parameter	Description
U8	***Gain Code***	Specifies the gain code for the channel specified above on the DAS I/O device. If the hardware does not support a programmable gain (gain setting by software), this value should be ignored.

3.1.5 GetGainInfo

Gets the gain information of the specified device. The device is specified by ***DevHandle***. (You can use **DeviceOpen** VI to get ***DevHandle***). Gain information includes a string list of gains (such as +/- 5V, +/- 10V, …) and gain codes.



Input:

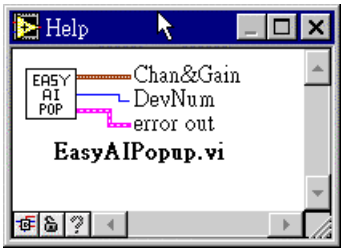
Data type	Parameter	Description
U32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
906	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
U32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
906	***Gain Info***	Gain information includes a string list of gains (such as +/- 5V, +/- 10V, …) and gain codes of the specified ***DevHandle***.
906	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

3.1.6 EasyAIPopup

Pops up a dialog windows for inputting Analog Input (AI) related configuration data and passes these data to next VI block.



Input:

None

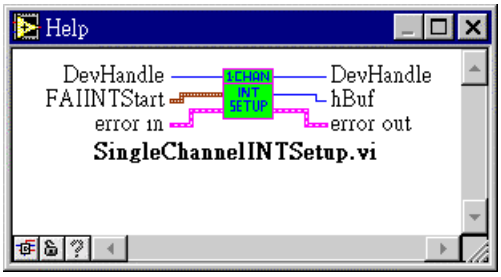
Output:

Data type	Parameter	Description
U32	***Chan&Gain***	A cluster data containing **Chan Info** parameter (includes **Channel number**, **ExpBrdID**, and **Expchannel** data) and **Gain code** for specified device.
I16	***DevNum***	The device number of the hardware installed in the system by "**DEVINST.EXE**"
I16	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

3.2 EasyFastAI

3.2.1 SingleChannelINTSetup

Initiates an asynchronous, single-channel data acquisition operation with interrupt transfer and stores its input in an array.



Input:

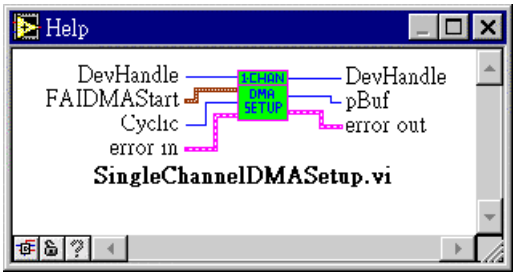
Data type	Parameter	Description
U32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
906	***FAIINTStart***	Initiates an asynchronous, single-channel data acquisition operation with interrupt transfer and stores its input in an array. ***FAIINTStart*** includes **trigsrc**, **samplerate, chan, gain, buffer, count, cyclic, Intrcount**.
906	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
U32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U32	***hBuf***	A data that stores the address of allocated memory.
906	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

3.2.2 SingleChannelDMASetup

Initiates an asynchronous, single-channel data acquisition operation with DMA and stores its input in an array.



Input:

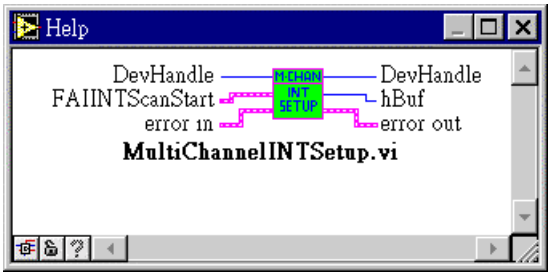
Data type	Parameter	Description
U32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
906	***FAIDMAStart***	***FAIDMAStart*** includes **trigsrc, samplerate, chan, gain, buffer, count, Intrcount.**
U8	***Cyclic***	Cyclic mode: ***cyclic (1), non cyclic (0)***
906	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
U32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U32	***pBuf***	A data that stores the address of allocated memory.
906	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

3.2.3 MultiChannelINTSetup

Initiates an asynchronous, multiple-channel data acquisition operation with interrupt transfer and stores its input in an array.



Input:

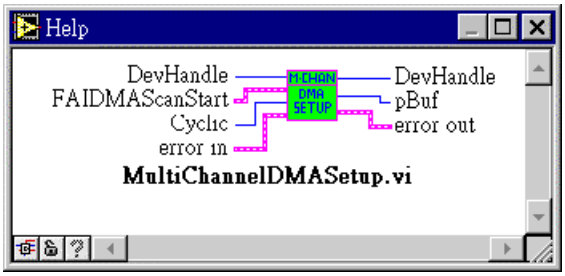
Data type	Parameter	Description
U32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
9D8	***FAIINTScanStart***	***FAIINTScanStart*** includes **TrigSrc, SampleRate, NumChans, StartChan, GainList, buffer, count, cyclic, FIFOSize.**
9D8	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
U32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U32	***hBuf***	A data that stores the address of allocated memory.
9D8	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

3.2.4 MultiChannelDMASetup

Initiates an asynchronous, multiple-channel data acquisition operation with DMA and stores its input in an array.



Input:

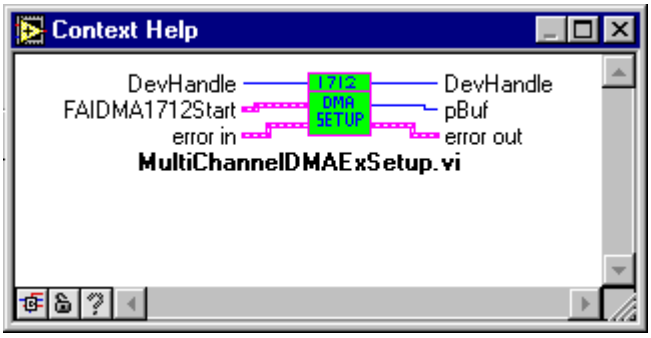
Data type	Parameter	Description
U32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
906	***FAIDMAScanStart***	***FAIDMAScanStart*** includes **TrigSrc, SampleRate, NumChans, StartChan, GainList, pBuf, count**
U8	***Cyclic***	Cyclic mode: **cyclic (1), non cyclic (0)**
906	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
U32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U32	***pBuf***	A data that stores the address of allocated memory.
906	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

3.2.5 MultiChannelDMAExSetup

Initiates an asynchronous, multiple-channel data acquisition operation with PCI-Bus mastering DMA and stores its input in an array(only for PCI-1712).



Input:

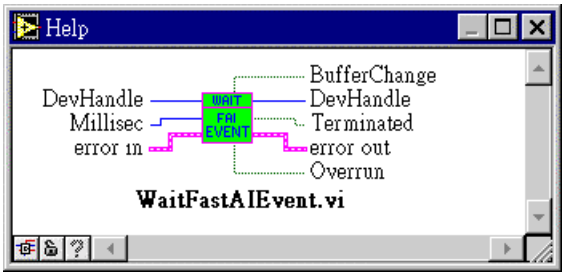
Data type	Parameter	Description
U32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
906	***FAIDMA1712Start***	***FAIDMA1712Start*** includes **TrigSrc, SampleRate, NumChans, StartChan, GainList, pBuf, count …**
906	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
U32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U32	***pBuf***	A data that stores the address of allocated memory.
906	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

3.2.6 WaitFastAIEvent

Waits for FastAI event until specified time has elapsed and outputs the status of the buffer change, terminate, or overrun.



Input:

Data type	Parameter	Description
U32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***Millisec***	Specifies the time to wait for a **FastAI** event. The unit of time is milliseconds.
error	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

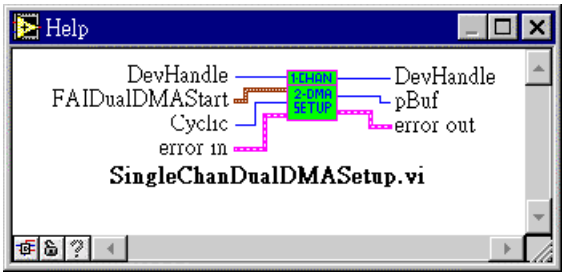
Output:

Data type	Parameter	Description
U32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***Buffer Change***	Indicates whether the data in the buffer has changed or not.
U16	***Terminated***	Indicates whether the waiting operation has terminated or not.
U16	***Overrun***	Indicates whether the specified time has overrun or not.
error	***error out***	A cluster containing error information. If ***error in***

indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

3.2.7 SingleChanDualDMASetup

Initiates an asynchronous, single-channel data acquisition operation with Dual DMA transfer and stores its input in an array. This function supports PCL-1800 only **.**



Input:

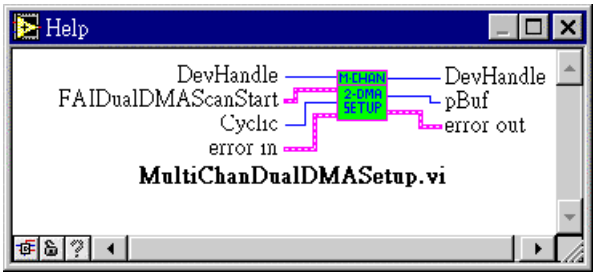
Data type	Parameter	Description
U32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
90s	***FAIDualDMAStart***	***FAIDualDMAStart*** includes **TrigSrc, SampleRate, NumChan, StartChan, GainList, BufferA, BufferB, count.**
U16	***Cyclic***	Cyclic mode: **cyclic (1), non cyclic (0)**
90s	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
U32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U32	***pBuf***	A data that stores the address of allocated memory.
90s	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

3.2.8 MultiChannelDualDMASetup

Initiates an asynchronous, multiple-channel data acquisition operation with Dual DMA transfer and stores its input in an array. This function supports PCL-1800 only .



Input:

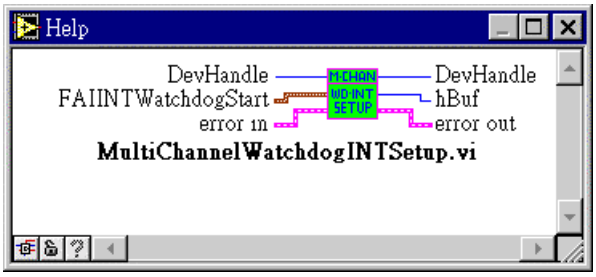
Data type	Parameter	Description
U32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
906	***FAIDualDMAScanStart***	***FAIDualDMAScanStart*** includes **TrigSrc, SampleRate, NumChan, StartChan, GainList, BufferA, BufferB, count.**
U8	***Cyclic***	Cyclic mode: **cyclic (1), non cyclic (0)**
906	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
U32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U32	***pBuf***	A data that stores the address of allocated memory.
906	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

3.2.9 MultiChannelWatchdogINTSetup

Initiates an asynchronous data acquisition operation with analog watchdog by interrupt transfer and stores its input in an array



Input:

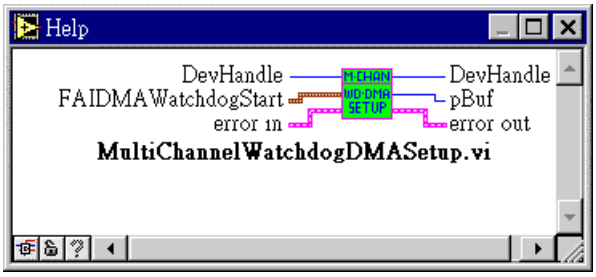
Data type	Parameter	Description
U32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
906	***FAIINTWatchdogStart***	***FAIINTWatchdogStart*** includes ***TrigSrc, SampleRate, buffer, BufferSize, count, cyclic, IntrCount***.
906	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
U32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U32	***hBuf***	A data that stores the address of allocated memory.
906	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

3.2.10 MultiChannelWatchdogDMASetup

Initiates an asynchronous data acquisition operation with analog watchdog by DMA transfer and stores its input in an array.



Input:

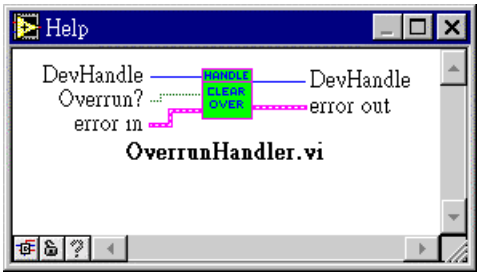
Data type	Parameter	Description
U32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
906	***FAIDMAWatchdogStart***	***FAIDMAWatchdogStart*** includes **TrigSrc, SampleRate, BufferA, BufferB, BufferSize, buffer, count, cyclic** .
906	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
U32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U32	***pBuf***	A data that stores the address of allocated memory.
906	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

3.2.11 OverrunHandler

Clears overrunning flag.



Input:

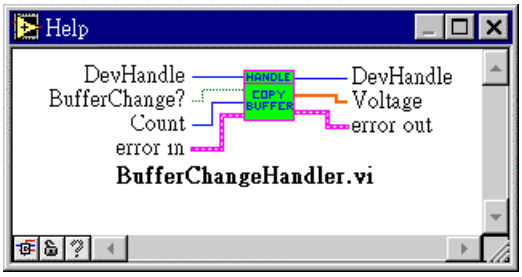
Data type	Parameter	Description
U32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***Overrun***	Overrun status: **overrun (1), no overrun (0)**
error	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
U32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
error	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

3.2.12 BufferChangeHandler

Transfers the data from the buffer being used for the data acquisition operation to the specified data buffer



Input:

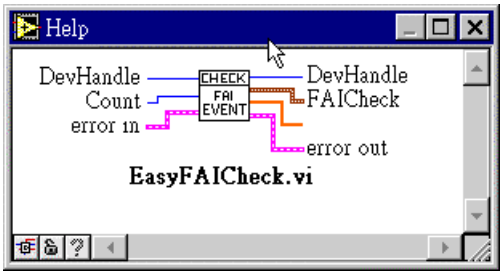
Data type	Parameter	Description
U32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***BufferChange***	Indicates whether the data in the buffer is changed or not.
U16	***Count***	Specifies the buffer size to be transferred
error	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
U32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***Voltage***	The value of voltage transferred from the buffer
error	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

3.2.13 EasyFAICheck

Checks whether the single channel data acquisition operation is complete or not and return current status.



Input:

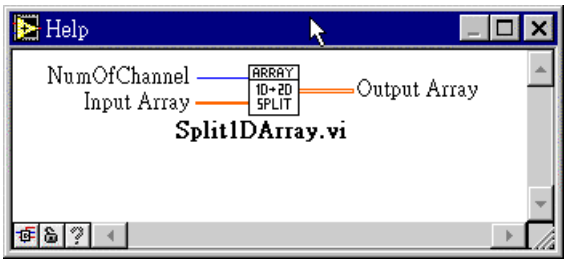
Data type	Parameter	Description
U32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***Count***	Specifies the buffer size to be transferred
cluster	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
U32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
cluster	***FAICheck***	A cluster containing **ActiveBuf, stopped, retrieved** and **overrun** data
cluster	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

3.2.14 SplitIDArray

Splits the input array to the output array based on the input ***NumOfChannel***.



Input:

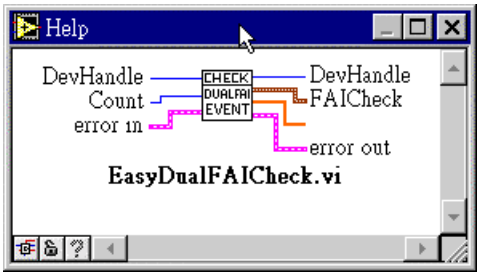
Data type	Parameter	Description
U16	***NumOfChannel***	While ***input array*** contains multiple-channel data, the ***NumOf Channel*** is used to specify which data will be output to ***output array.***
	Input Array	The buffer array stores multiple-channel data.

Output:

Data type	Parameter	Description
	Output Array	The buffer array stores multiple-channel data.

3.2.15 EasyDualFAICheck

Checks whether the asynchronous, multiple-channel data acquisition operation is complete or not and return current status.



Input:

Data type	Parameter	Description
U32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***Count***	Specifies the buffer size to be transferred
908	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
U32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
908	***FAICheck***	A cluster containing **ActiveBuf, stopped, retrieved** and **overrun** data
908	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

Chapter 4 Advanced I/O VIs

The Advanced I/O VIs provide more hardware and software functionality, flexibility, and efficiency than the Easy I/O VIs.

4.1 DLL Function Support

This chapter contains tables that show the VI functions support for each hardware device.

Table 4-1. VI Functions

	Device							
Function	PCL-818 Series	PCL-818HG	PCL-1800	PCL-816	PCL-812PG	PCL-711B	MIC-2718	PCM-3718
Device functions								
DeviceOpen	√	√	√	√	√	√	√	√
DeviceClose	√	√	√	√	√	√	√	√
DeviceGetFeatures	√	√	√	√	√	√	√	√
Analog input								
AIConfig	√	√	√	√	√	√	√	√
AIGetConfig	√	√	√	√	√	√	√	√
AIBinaryIn	√	√	√	√	√	√	√	√
AIScale	√	√	√	√	√	√	√	√
AIVoltageIn	√	√	√	√	√	√	√	√
AIVoltageInExp	√	√	√	√	√	√	√	√
MAIConfig	√	√	√	√	√	√	√	√
MAIBinaryIn	√	√	√	√	√	√	√	√
MAIVoltageIn	√	√	√	√	√	√	√	√
MAIVoltageInExp	√	√	√	√	√	√	√	√
Analog output								
AOConfig	√	√	√	√	√	√		
AOBinaryOut	√	√	√	√	√	√		
AOVoltageOut	√	√	√	√	√	√		
AOScale	√	√	√	√	√	√		
Port I/O functions								
WritePortByte	√	√	√	√	√	√	√	√
WritePortWord	√	√	√	√	√	√	√	√
ReadPortByte	√	√	√	√	√	√	√	√
ReadPortWord	√	√	√	√	√	√	√	√
Digital input/output								
DioGetConfig								
DioSetPortMode								
DioReadPortByte	√	√	√	√	√	√		√
DioWritePortByte	√	√	√	√	√	√		√
DioReadBit	√	√	√	√	√	√		√
DioWriteBit	√	√	√	√	√	√		√
DioGetCurrentDOByt	√	√	√	√	√	√		√
DioGetCurrentDOBit	√	√	√	√	√	√		√
Temperature								
TCMuxRead	√	√	√	√	√	√		√
High-speed								
FAIIntStart	√	√	√	√	√	√	√	√
FAIDMAStart	√	√	√	√	√	√		√

FAIIntScanStart	√	√	√	√	√	√	√	√
FAIDMAScanStart	√	√	√	√	√	√		√
FAIDualDMAStart			√					
FAIDualScanStart			√					
FAITransfer	√	√	√	√	√	√	√	√
FAICheck	√	√	√	√	√	√	√	√
FAIIntWatchdogStart			√					
FAIDMAWatchdogSta			√					
FAIWatchdogCheck			√					
FAIStop	√	√	√	√	√	√	√	√
AllocateDMABuffer	√	√	√	√	√		√	√
FreeDMABuffer	√	√	√	√	√		√	√
FAOIntStart			√	√				
FAODMAStart			√	√				
FAOLoad			√	√				
FAOScale			√	√				
FAOCheck			√	√				
FAOStop			√	√				
EnableEvent	√	√	√	√	√	√	√	√
CheckEvent	√	√	√	√	√	√	√	√
Counter functions								
CounterEventStart	√	√	√	√	√	√	√	√
CounterEventRead	√	√	√	√	√	√	√	√
CounterFreqStart	√	√	√	√	√	√	√	√
CounterFreqRead	√	√	√	√	√	√	√	√
CounterPulseStart	√	√	√	√	√	√	√	√
CounterReset	√	√	√	√	√	√	√	√
QCounterConfig								
QCounterConfigSys								
QCounterStart								
QcounterRead								

Table 4-1. VI Functions (Continued)

	Device					
Function	PCI-1710	PCI-1713	PCI-1711	PCI-1712	PCI-1731	PCI-1720
Device functions						
DeviceOpen	√	√	√	√	√	√
DeviceClose	√	√	√	√	√	√
DeviceGetFeatures	√	√	√	√	√	√
Analog input						
AIConfig	√	√	√	√	√	
AIGetConfig	√	√	√	√	√	
AIBinaryIn	√	√	√	√	√	
AIScale	√	√	√	√	√	
AIVoltageIn	√	√	√	√	√	
AIVoltageInExp						
MAIConfig	√	√	√	√	√	
MAIBinaryIn	√	√	√	√	√	
MAIVoltageIn	√	√	√	√	√	
MAIVoltageInExp	√	√	√	√	√	
Analog output						
AOConfig	√		√	√		√
AOBinaryOut	√		√	√		√

AOVoltageOut	√		√	√		√
AOScale	√		√	√		√
EnableSyncAO						√
WriteSyncAO						√
AOCurrentOut						√
Port I/O functions						
WritePortByte	√	√	√	√	√	√
WritePortWord	√	√	√	√	√	√
ReadPortByte	√	√	√	√	√	√
ReadPortWord	√	√	√	√	√	√
Digital input/output						
DioGetConfig						
DioSetPortMode						
DioReadPortByte	√		√	√	√	
DioWritePortByte	√		√	√	√	
DioReadBit	√		√	√	√	
DioWriteBit	√		√	√	√	
DioGetCurrentDOByte	√		√	√	√	
DioGetCurrentDOBit	√		√	√	√	
Temperature						
TCMuxRead						
High-speed functions						
FAIIntStart	√	√	√		√	
FAIDMAStart						
FAIIntScanStart	√	√	√		√	
FAIDMAScanStart						
FAIDualDMAStart						
FAIDualScanStart						
FAITransfer	√	√	√	√	√	
FAICheck	√	√	√	√	√	
FAICheckEvent			√			
FAIIntWatchdogStart						
FAIDMAWatchdogStart						
FAIWatchdogCheck				√		
FAIStop	√	√		√		
AllocateDMABuffer				√		
FreeDMABuffer				√		
FAOIntStart						
FAODMAStart				√		
FAOLoad				√		
FAOScale				√		
FAOCheck				√		
FAOStop				√		
EnableEvent	√	√	√	√	√	
CheckEvent	√	√	√	√	√	
Clearoverrun	√	√		√		
FAIDMAExStart				√		
Counter functions						
CounterEventStart	√		√	√	√	
CounterEventRead	√		√	√	√	
CounterFreqStart	√		√	√	√	
CounterFreqRead	√		√	√	√	
CounterPulseStart	√		√	√	√	
CounterReset	√		√	√	√	

OcounterRead				√		

Table 4-1. VI Functions (Continued)

	Device						
Function	PCL-813B	PCL-726/727	PCL-728 MIC2728	Demo Board	PCL-725 /730	PCL-733 MIC2730 /2732	PCL-722 /724/731 PCM-3724
Device functions							
DeviceOpen	√	√	√	√	√	√	√
DeviceClose	√	√	√	√	√	√	√
DeviceGetFeatures	√	√	√	√	√	√	√
Analog input							
AIConfig	√			√			
AIGetConfig	√			√			
AIBinaryIn	√						
AIScale	√						
AIVoltageIn	√			√			
AIVoltageInExp	√			√			
MAIConfig	√			√			
MAIBinaryIn	√			√			
MAIVoltageIn	√			√			
MAIVoltageInExp	√			√			
Analog output							
AOConfig		√	√				
AOBinaryOut		√	√				
AOVoltageOut		√	√				
AOScale		√	√				
Port I/O functions							
WritePortByte	√	√	√	√	√	√	√
WritePortWord	√	√	√	√	√	√	√
ReadPortByte	√	√	√	√	√	√	√
ReadPortWord	√	√	√	√	√	√	√
Digital input/output							
DioGetConfig							√
DioSetPortMode							√
DioReadPortByte		√			√	√	√
DioWritePortByte		√			√		√
DioReadBit		√			√	√	√
DioWriteBit		√			√		√
DioGetCurrentDOByt		√			√		√
DioGetCurrentDOBit		√			√		√
Temperature							
TCMuxRead							
High-speed							
EnableEvent							√
CheckEvent							√

Table 4-1. VI Functions (Continued)

	Device				
Function	PCL-734/735 MIC2750 /2752	PCL-833	PCL-720	PCL-721 /723	PCL-836
Device functions					
DeviceOpen	√	√	√	√	√

DeviceClose	√	√	√	√	√
DeviceGetFeatures	√	√	√	√	√
Port I/O functions					
WritePortByte	√	√	√	√	√
WritePortWord	√	√	√	√	√
ReadPortByte	√	√	√	√	√
ReadPortWord	√	√	√	√	√
Digital input/output					
DioGetConfig	√		√	√	√
DioSetPortMode					
DioReadPortByte		√	√	√	√
DioWritePortByte	√		√		√
DioReadBit		√	√	√	√
DioWriteBit	√		√	√	√
DioGetCurrentDOByte	√		√	√	
DioGetCurrentDOBit	√		√	√	
Counter functions					
CounterEventStart			√		√
CounterEventRead			√		√
CounterFreqStart			√		√
CounterFreqRead			√		√
CounterPulseStart			√		√
CounterReset			√		√
QCounterConfig		√			
QCounterConfigSys		√			
QCounterStart		√			
QCounterRead		√			
High-speed functions					
EnableEvent					√
CheckEvent					√

Table 4-1. VI Functions (Continued)

	Device						
Function	PCI-1750	PCI-1751	PCI-1760	PCI-1752	PCI-1753	PCI-1754	PCI-1756
Device functions							
DeviceOpen	√	√	√	√	√	√	√
DeviceClose	√	√	√	√	√	√	√
DeviceGetFeatures	√	√	√	√	√	√	√
Port I/O functions							
WritePortByte	√	√	√	√	√	√	√
WritePortWord	√	√	√	√	√	√	√
ReadPortByte	√	√	√	√	√	√	√
ReadPortWord	√	√	√	√	√	√	√
Digital input/output							
DioGetConfig		√					
DioSetPortMode		√			√		√
DioReadPortByte	√	√	√		√	√	√
DioWritePortByte	√	√	√		√	√	√
DioReadBit	√	√	√		√	√	√
DioWriteBit	√	√	√	√	√		
DioGetCurrentDOByt	√	√	√	√	√		
DioGetCurrentDOBit	√	√	√	√	√		

High-speed							
EnableEvent	√	√	√		√		√
CheckEvent	√	√			√		√
TimerCountSetting	√	√					
EnableEventEx			√		√		
FDITransfer			√				√
Counter functions							
CounterEventStart	√	√	√	√	√	√	√
CounterEventRead	√	√	√	√	√	√	√
CounterFreqStart	√	√	√	√	√	√	√
CounterFreqRead	√	√	√	√	√	√	√
CounterPulseStart	√	√	√	√	√	√	√
CounterReset	√	√	√	√	√	√	√
QCounterConfig							
QCounterConfigSys							
QCounterStart							
QcounterRead							

Table 4-1. VI Functions (Continued)

	Device							
Function	ADAM-4011/4011D	ADAM-4012	ADAM-4014D	ADAM-4018/4018M/5018	ADAM-4017/4013/5017	ADAM-4021/5024	ADAM-4016	ADAM-4052/4053/5051/5052
Device functions								
DeviceOpen	√	√	√	√	√	√	√	√
DeviceClose	√	√	√	√	√	√	√	√
DeviceGetFeatures	√	√	√	√	√	√	√	√
Analog input								
AIGetConfig	√	√	√	√	√		√	
AIVoltageIn	√	√	√	√	√		√	
AIVoltageInExp								
MAIConfig								
MAIVoltageIn	√	√	√	√	√		√	
MAIVoltageInExp								
Analog output								
AOVoltageOut						√		
Digital input/output								
DioGetConfig								
DioSetPortMode								
DioReadPortByte	√	√	√					√
DioWritePortByte	√	√	√				√	
DioReadBit	√	√	√					√
DioWriteBit	√	√	√				√	
DioGetCurrentDOByte	√	√	√				√	
DioGetCurrentDOBit	√	√	√				√	
Temperature								
TCMuxRead	√			√				
Counter functions								
CounterEventStart	√	√	√					
CounterEventRead	√	√	√					
CounterReset	√	√	√					
Alarm functions								
AlarmConfig	√	√	√	√	√			
AlarmEnable	√	√	√	√	√			
AlarmCheck	√	√	√	√	√			
AlarmReset	√	√	√	√	√			

Table 4-1. VI Functions (Continued)

Function	ADAM-4060/5056/5060	ADAM-4080D	ADAM-4530	ADAM-4521	ADAM-5050	ADAM-4050
Device functions						
DeviceOpen	√	√	√	√	√	√
DeviceClose	√	√	√	√	√	√
DeviceGetFeatures	√	√	√	√	√	√
Digital input/output						
DioGetConfig					√	
DioSetPortMode						
DioReadPortByte					√	√
DioWritePortByte	√	√			√	√
DioReadBit					√	√
DioWriteBit	√	√			√	√
DioGetCurrentDOByte	√	√			√	√
DioGetCurrentDOBit	√	√			√	√
Counter functions						
CounterEventStart		√				
CounterEventRead		√				
CounterReset		√				
Alarm functions						
AlarmConfig		√				
AlarmEnable		√				
AlarmCheck		√				
AlarmReset		√				
Comm. port						
COMOpen						
COMClose						
COMGetConfig						
COMSetConfig						
COMRead						
COMWrite						
COMWrite232						
COMEscape						

The advanced I/O VIs must be called in the following order:

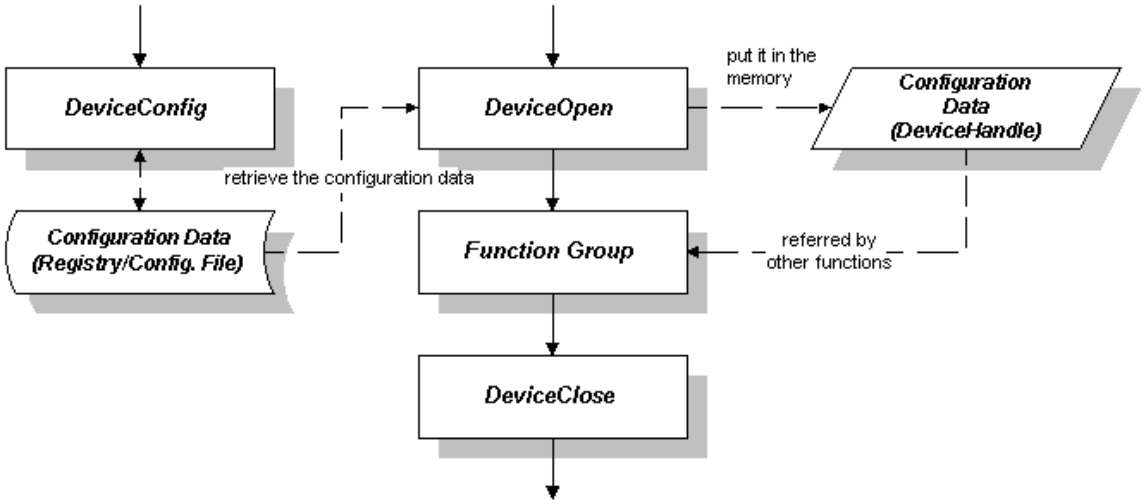


Fig 4-1. Function Flow Overview

Analog Input Flow Chart:

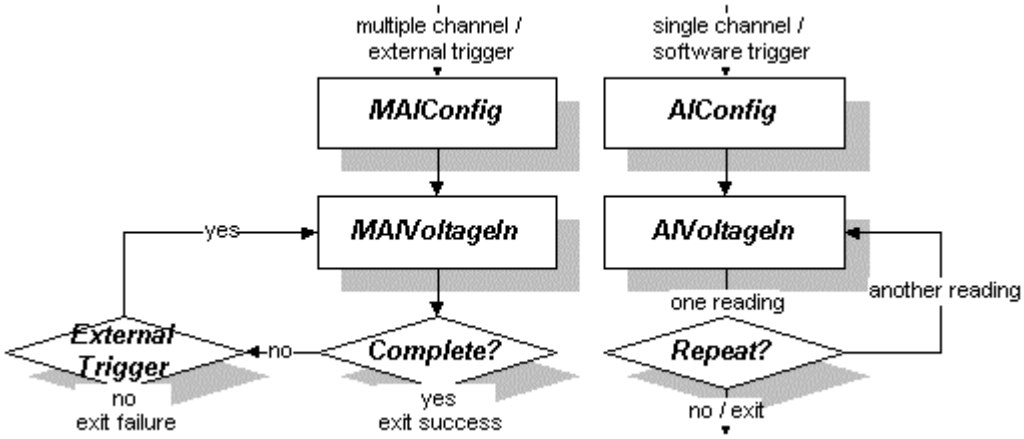


Fig 4-2. Function Flow Overview

Analog Output Flow Chart:

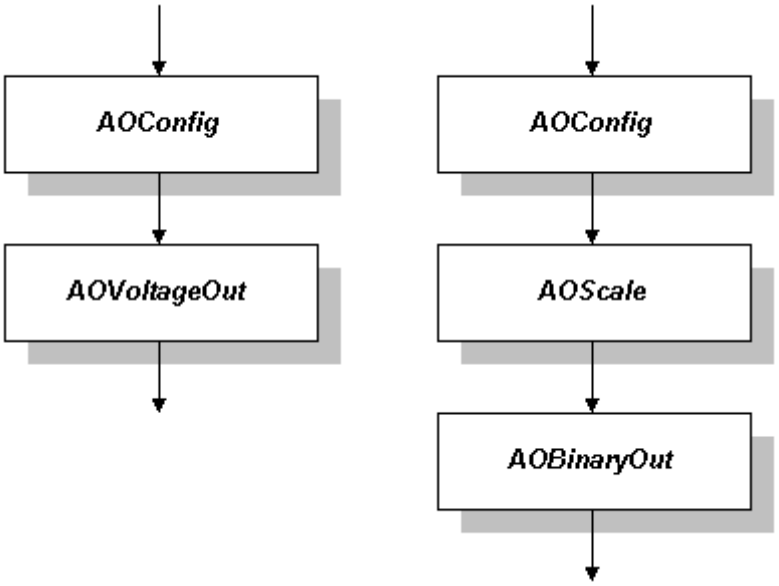


Fig 4-3. Function Flow Overview

Digital Input /Output Flow Chart:

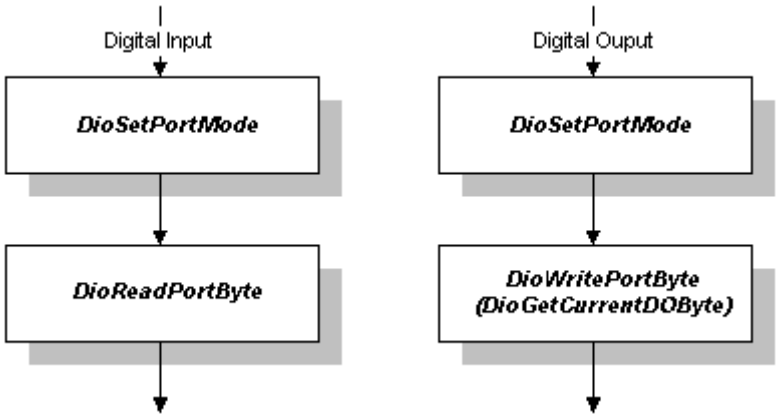


Fig 4-4. Function Flow Overview

Event Counter Flow Chart:

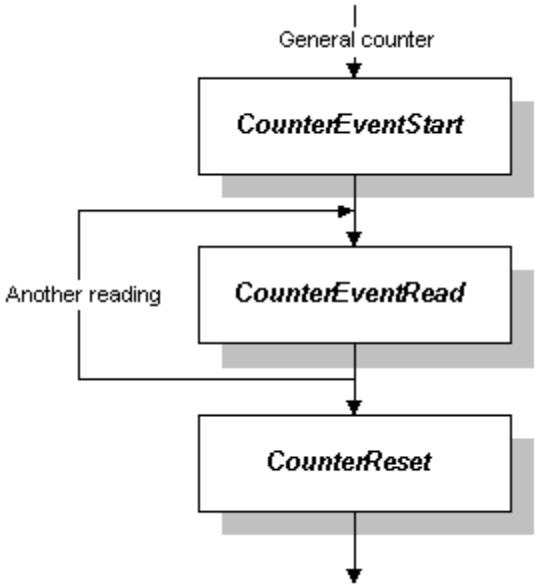


Fig 4-5. Function Flow Overview

Frequency Measurement Flow Chart:

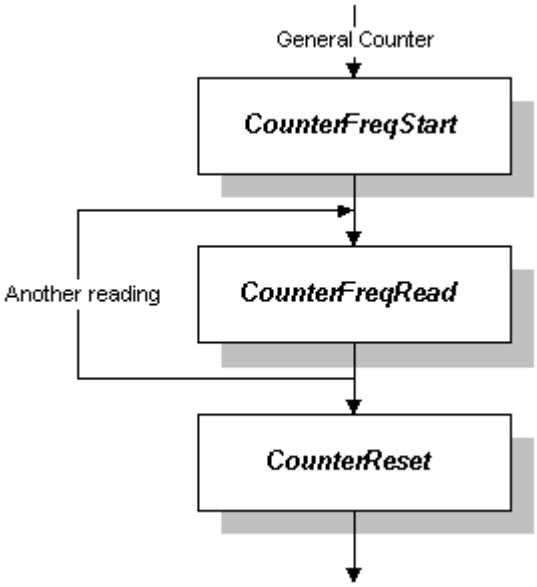


Fig 4-6. Function Flow Overview

Pulse Output Flow Chart:

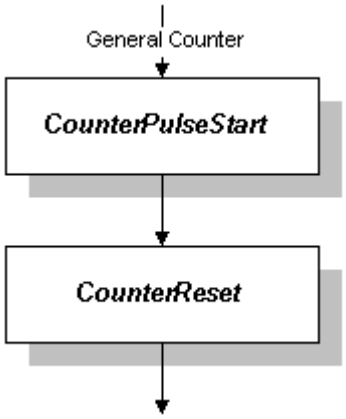


Fig 4-7. Function Flow Overview

Quadratic Counter Flow Chart:

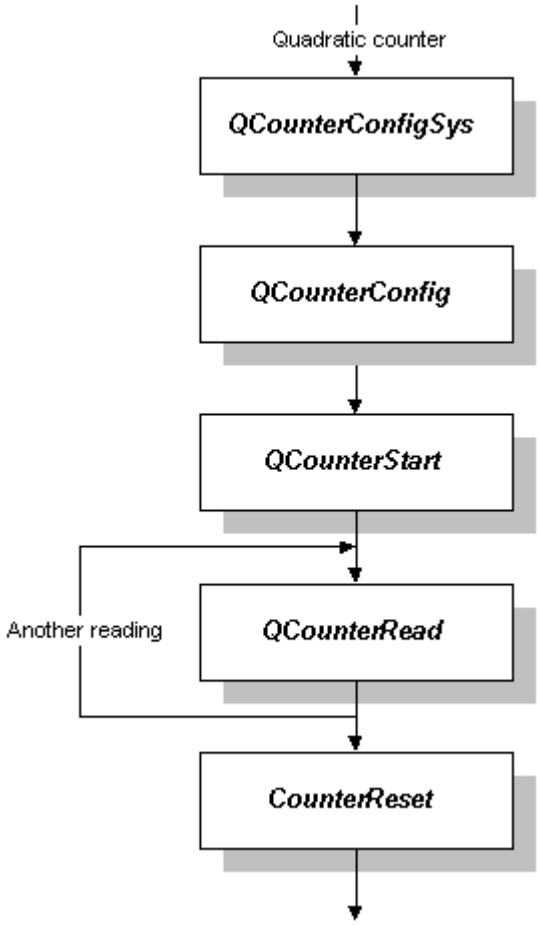


Fig 4-8. Function Flow Overview

Alarm Operations Flow Chart:

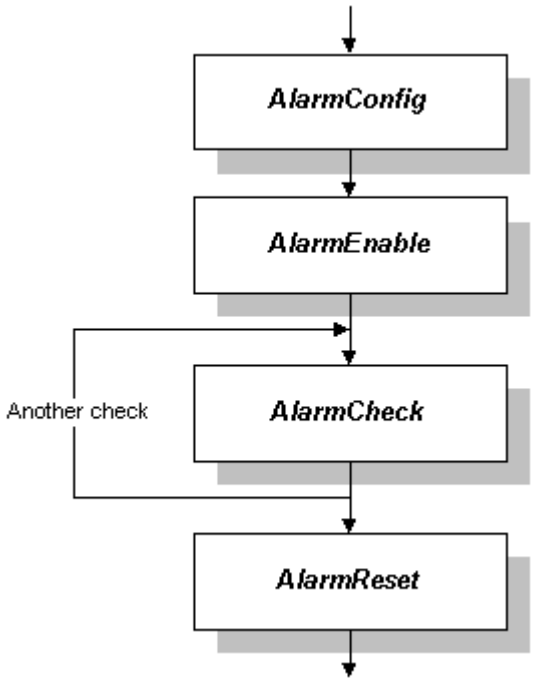


Fig 4-9. Function Flow Overview

High Speed Flow Chart:

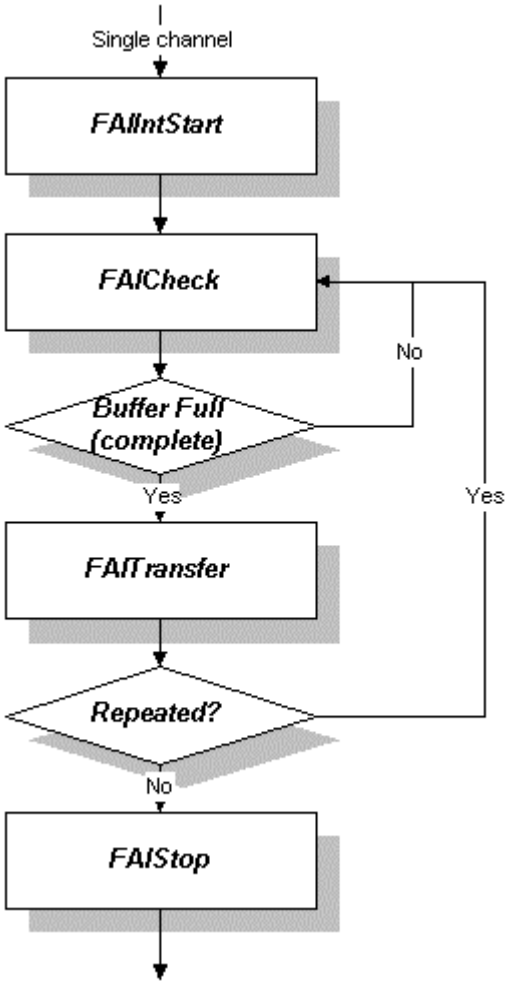


Fig 4-10. The Call Flow for Single-channel Data Acquisition with interrupt transfer

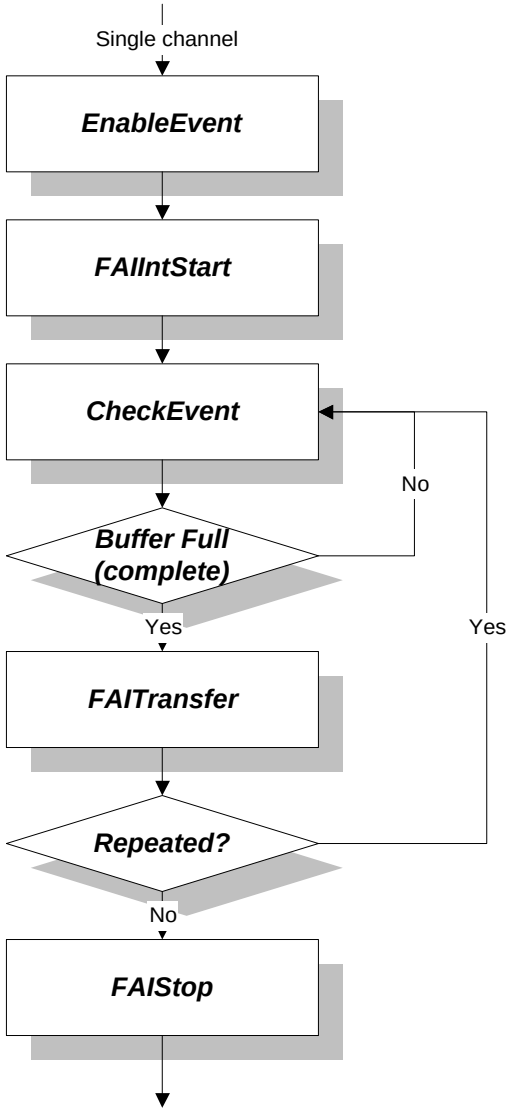


Fig 4-11. The Call Flow for Single-channel Data Acquisition with interrupt transfer and event function

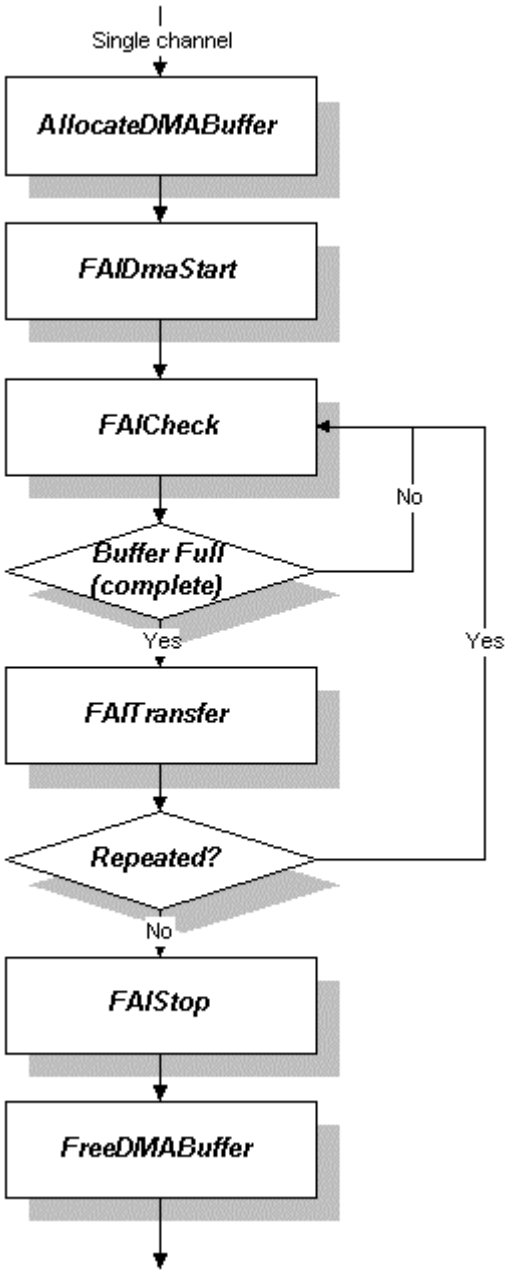


Fig 4-12. The Call Flow for Single-channel Data Acquisition with DMA transfer

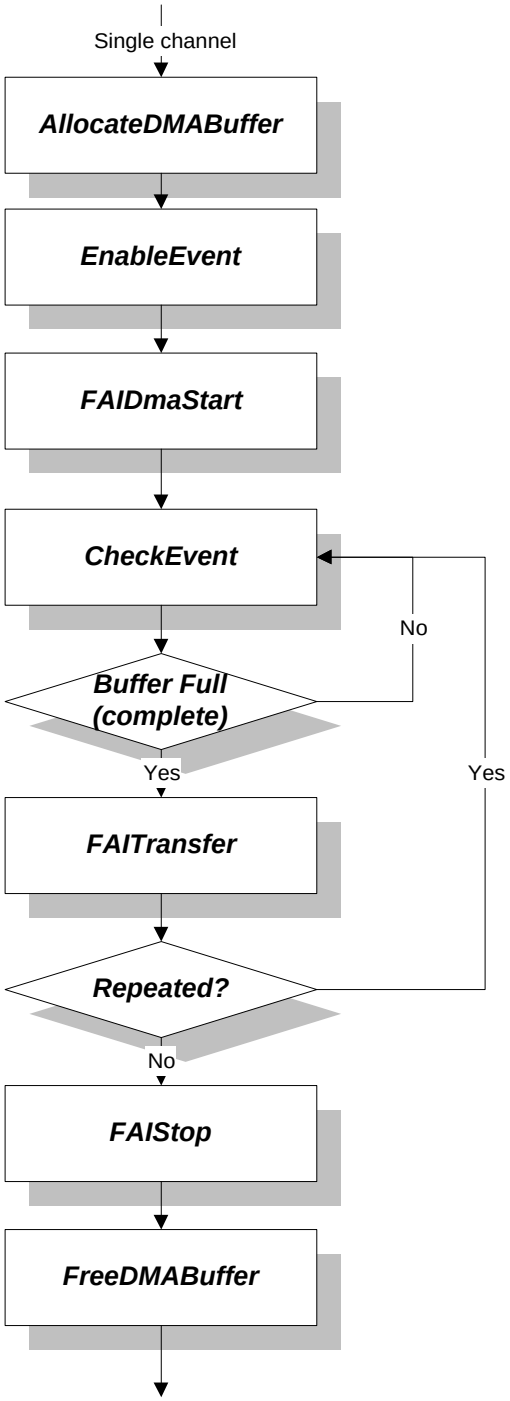


Fig 4-13. The Call Flow for Single-channel Data Acquisition with DMA transfer and event function

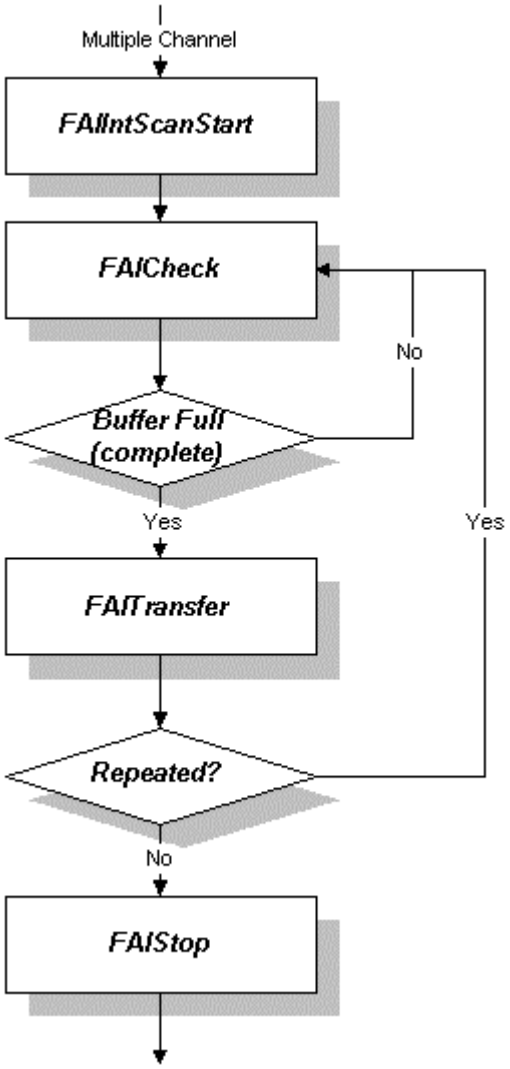


Fig 4-14. The Call Flow for Multiple-channel Data Acquisition with interrupt transfer

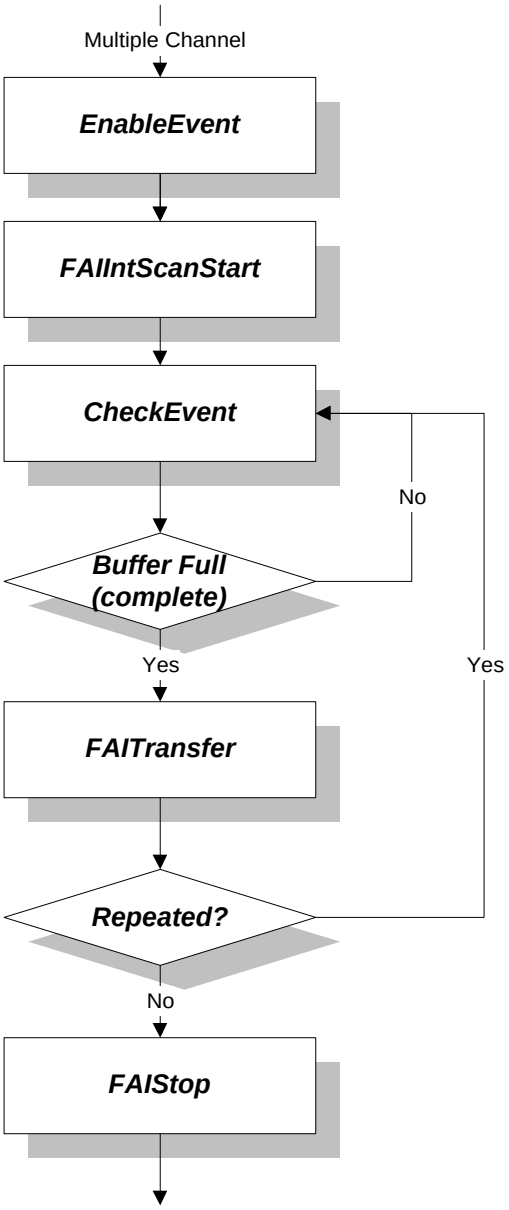


Fig 4-15. The Call Flow for Multiple-channel Data Acquisition with interrupt transfer and event function

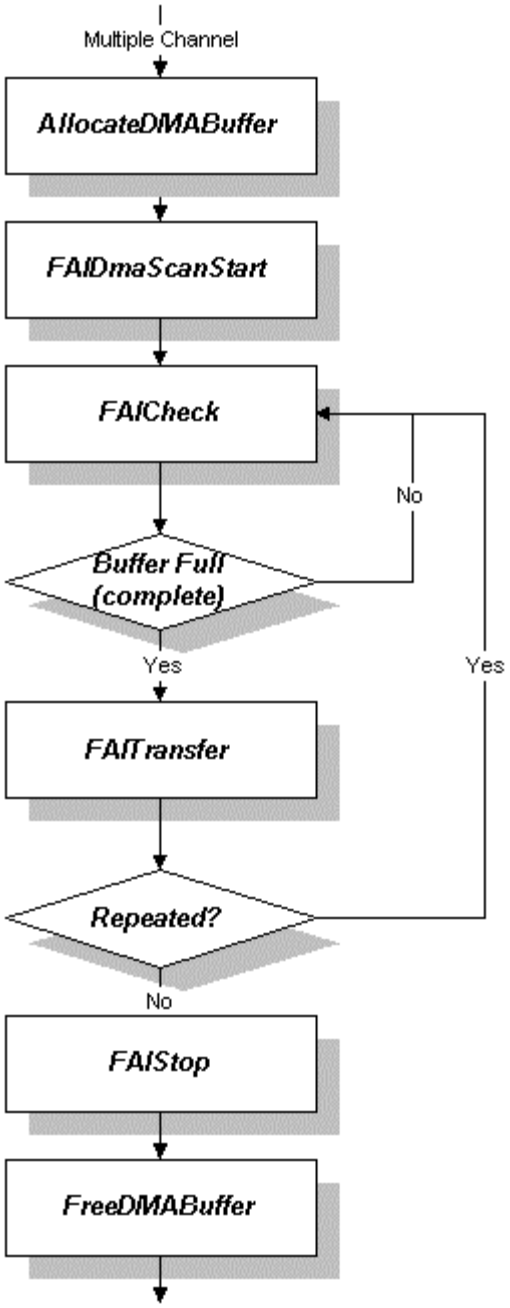


Fig 4-16. The Call Flow for Multiple-channel Data Acquisition with DMA transfer

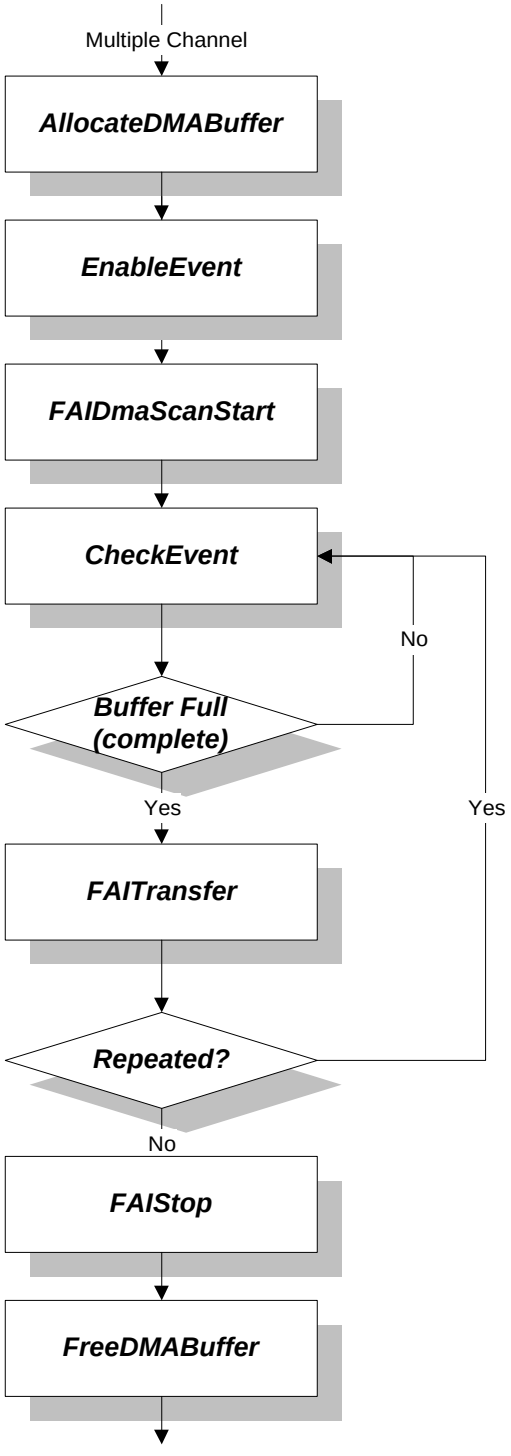


Fig 4-17. The Call Flow for Multiple-channel Data Acquisition with DMA transfer and event function

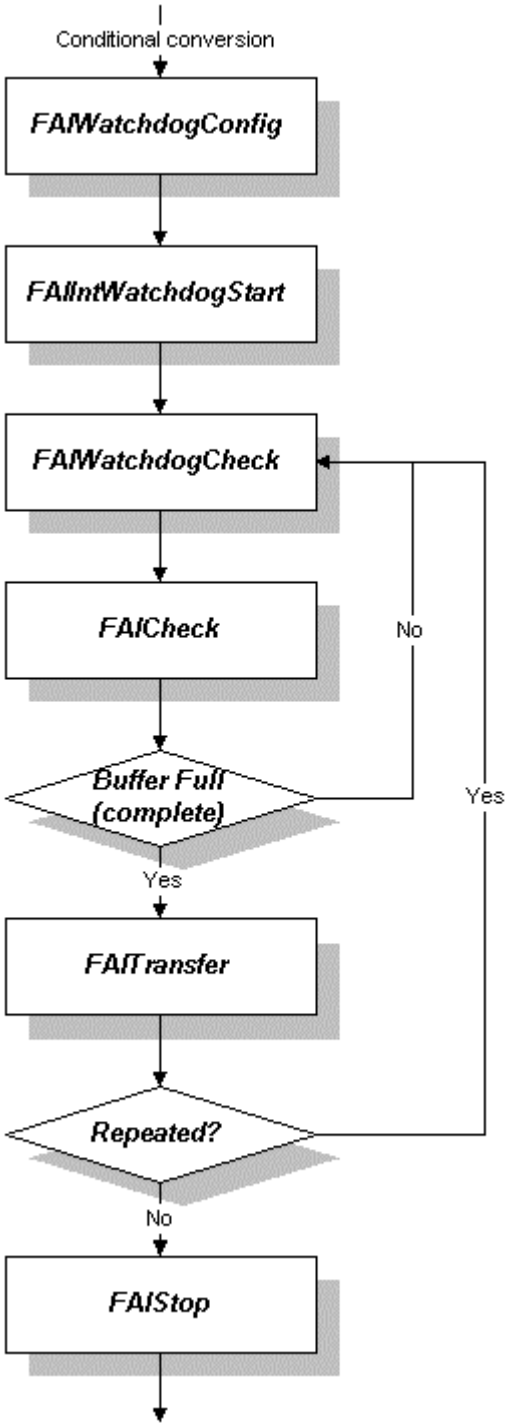


Fig 4-18. The Call Flow for Multiple-channel Watchdog Data Acquisition with interrupt transfer

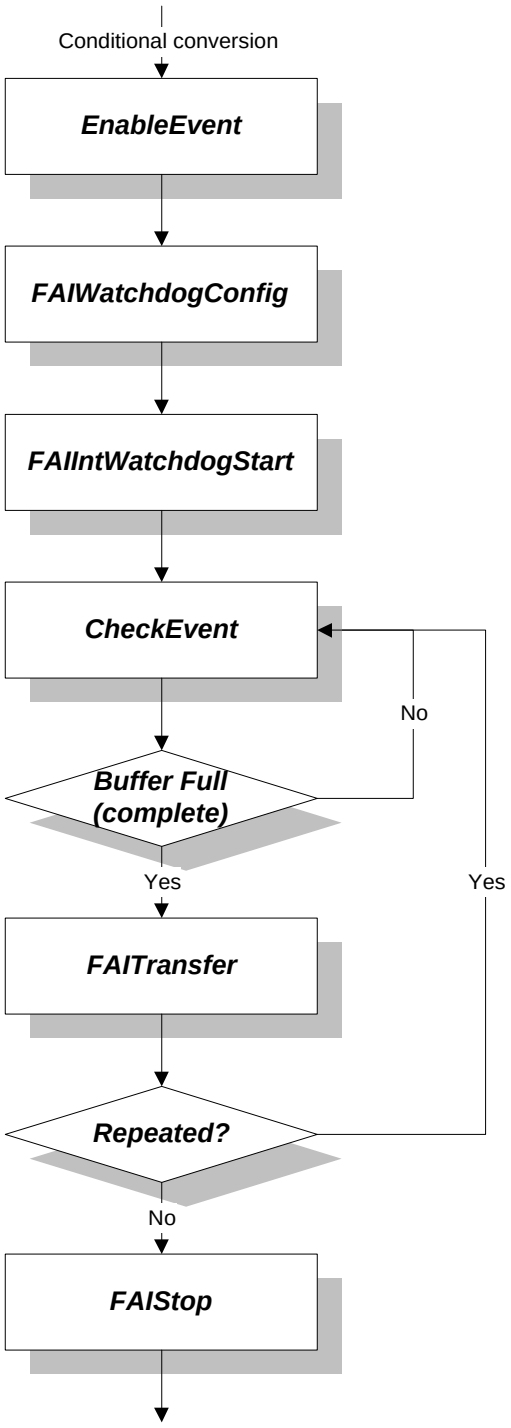


Fig 4-19. The Call Flow for Multiple-channel Watchdog Data Acquisition with interrupt transfer and event function

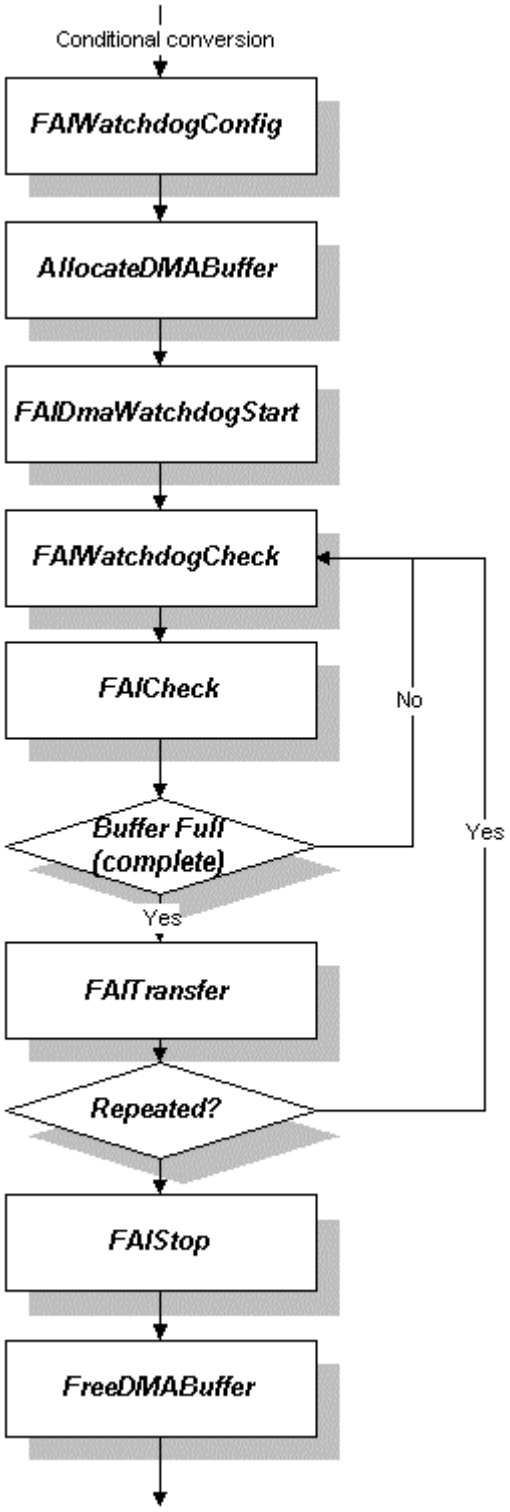


Fig 4-20. The Call Flow for Multiple-channel Watchdog Data Acquisition with DMA transfer

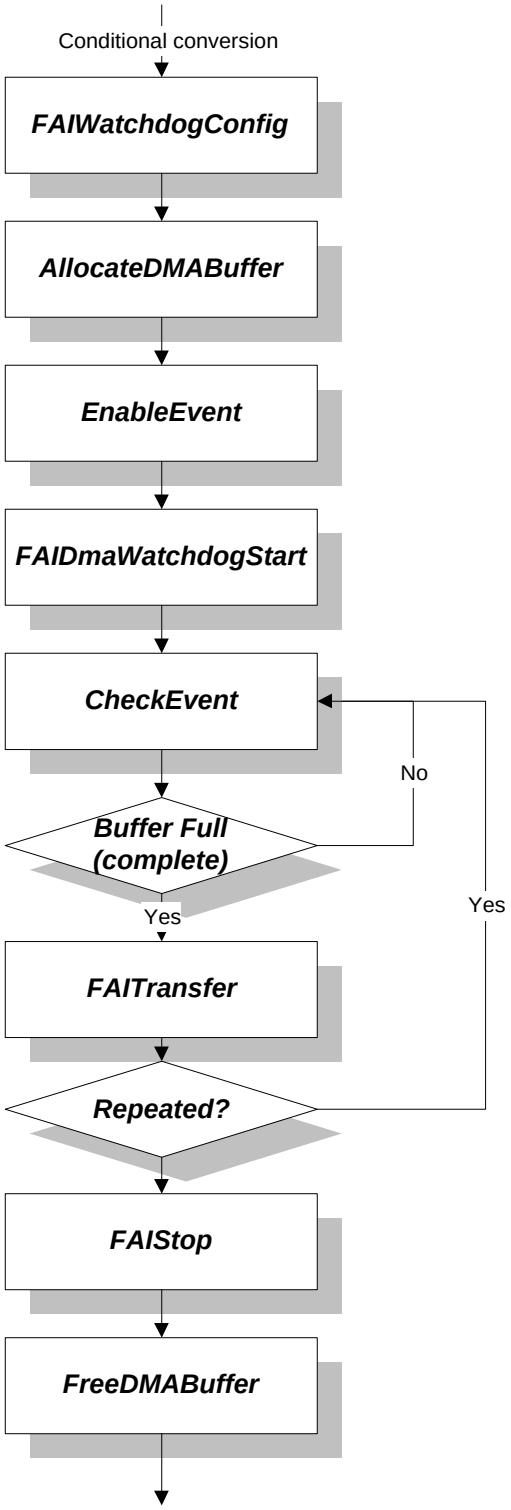


Fig 4-21. The Call Flow for Multiple-channel Watchdog Data Acquisition with DMA transfer and event function

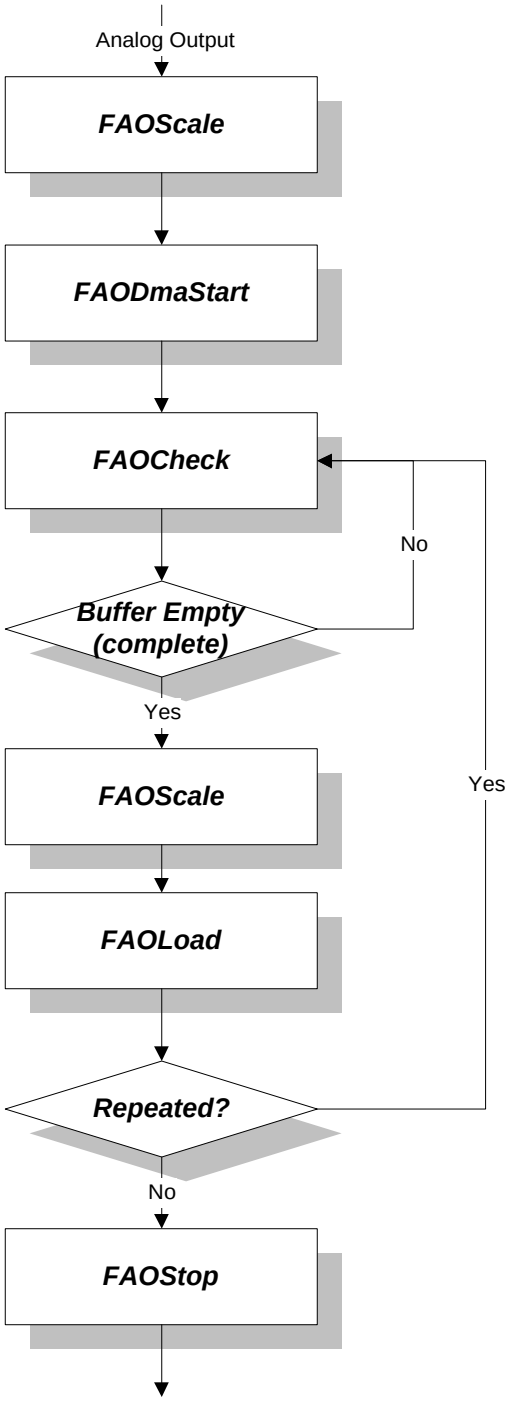


Fig 4-22. The Call Flow for Single-Channel Analog Output with DMA transfer

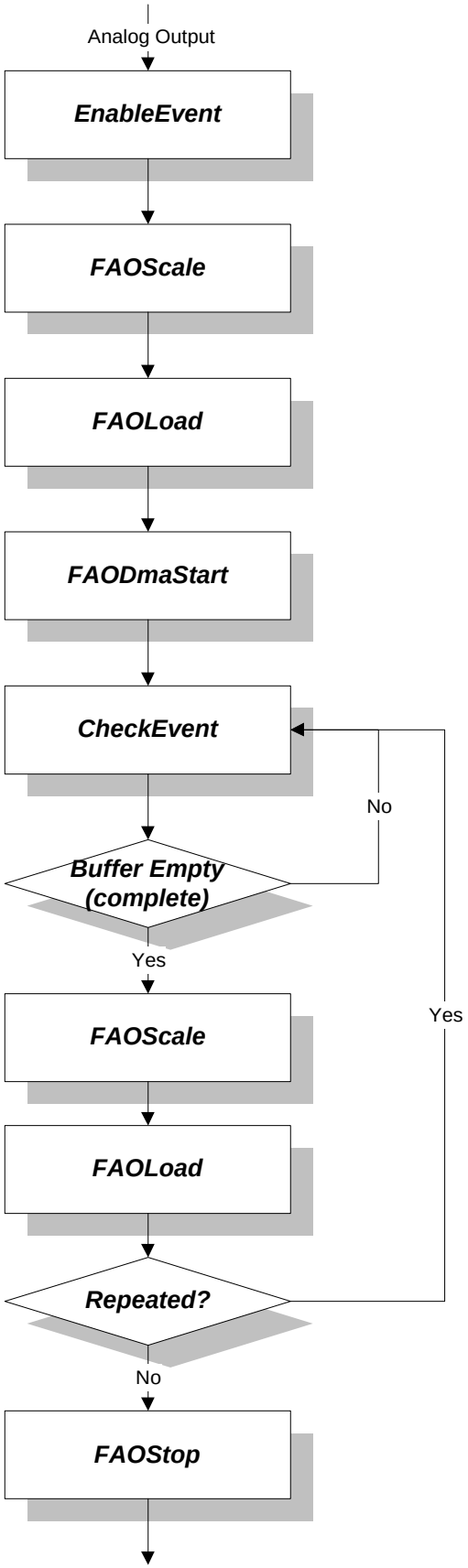

Analog Output
EnableEvent
FAOScale
FAOLoad
FAODmaStart
CheckEvent
No
Buffer Empty (complete)
Yes
FAOScale
FAOLoad
Repeated?
Yes
No
FAOStop

Fig 4-23 The Call Flow for Single-Channel Analog Output with DMA transfer and event function

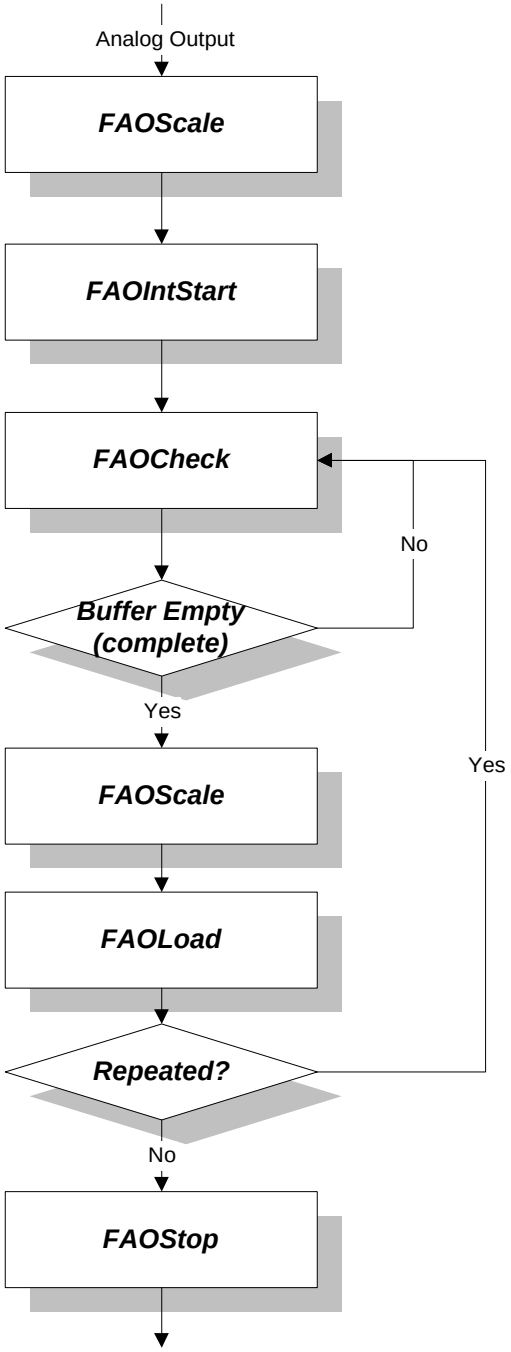


Fig 4-24. The Call Flow for Single-Channel Analog Output with interrupts transfer

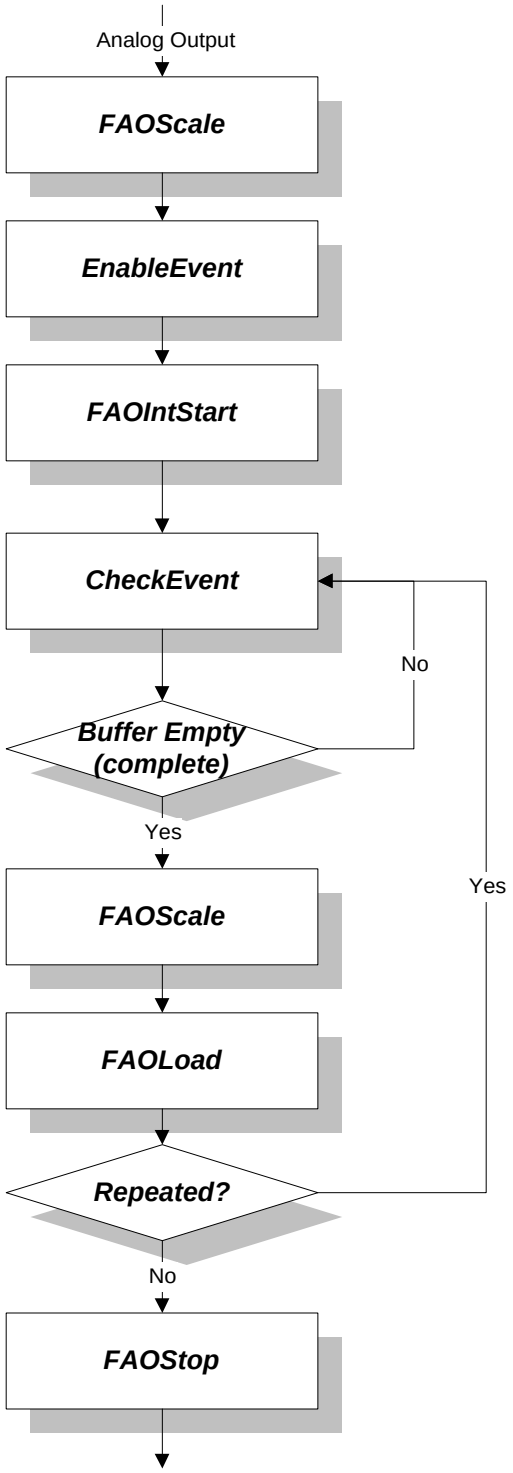


Fig 4-25. The Call Flow for Single-Channel Analog Output with interrupt transfer and event function

The Advanced I/O VIs are actually composed of intermediate VIs, which are in turn composed of advanced VIs.

Advanced I/O VIs		
Group	**VIs**	**Mapping 32-bit DLL API**
DeviceManager	- DeviceGetNumOfList - DeviceGetList - DeviceGetSubList - BoardName - DeviceOpen - DeviceClose - DeviceGetFeatures - GetErrorMessage - AllocDMABuf - FreeDMABuf - AllocINTBuf - FreeINTBuf - EnableEvent - CheckEvent - GetFIFOSize - DeviceNumToDeviceName	- DRV_DeviceGetNumOfList - DRV_DeviceGetList - DRV_DeviceGetSubList - DRV_BoardTypeMapBoardName - DRV_DeviceOpen - DRV_DeviceClose - DRV_DeviceGetFeatures - DRV_GetErrorMessage - DRV_AllocDMABuf - DRV_FreeDMABuf - DRV_EnableEvent - DRV_CheckEvent - DRV_GetFIFOSize - DRV_DeviceNumToDeviceName
adSlowAI	- AIConfig - AIGetConfig - MAIConfig - TCMuxRead - AIVoltageIn - AIBinaryIn - AIScale - AIVoltageInExp - MAIVolatgeIn - MAIBinaryIn - MAIVoltageInExp	- DRV_AIConfig - DRV_AIGetConfig - DRV_MAIConfig - DRV_TCMuxRead - DRV_AIVoltageIn - DRV_AIBinaryIn - DRV_AIScale - DRV_AIVoltageInExp - DRV_MAIVolatgeIn - DRV_MAIBinaryIn - DRV_MAIVoltageInExp
adSlowAO	- AOConfig - AOVoltageOut - AOScale - AOBinaryOut - AOCurrentOut - EnableSyncAO - WriteSyncAO	- DRV_AOConfig - DRV_AOVoltageOut - DRV_AOScale - DRV_AOBinaryOut - DRV_AOCurrentOut - DRV_EnableSyncAO - DRV_WriteSyncAO
adDIO	- DIOSetPortMode - DIOGetConfig - DIOGetCurrentDOBit - DIOGetCurrentDOByte - DIOWriteBit - DIOReadBit - DIOWritePortByte	- DRV_DIOSetPortMode - DRV_DIOGetConfig - DRV_DIOGetCurrentDOBit - DRV_DIOGetCurrentDOByte - DRV_DIOWriteBit - DRV_DIOReadBit - DRV_DIOWritePortByte

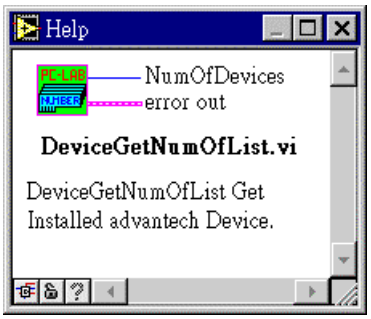
	- DIOReadPortByte - WritePortByte - ReadPortByte - WritePortWord - ReadPortWord - outp - outpw - inp - inpw	- DRV_DIOReadPortByte - DRV_WritePortByte - DRV_ReadPortByte - DRV_WritePortWord - DRV_ReadPortWord - DRV_outp - DRV_outpw - DRV_inp - DRV_inpw
adCountTimer	- CounterEventStart - CounterEventRead - CounterFregStart - CounterFregRead - QCounterStart - QCounterRead - CounterPulseStart - CounterReset - QCounterConfigSys - QcounterConfig - TimerCountSetting	- DRV_CounterEventStart - DRV_CounterEventRead - DRV_CounterFregStart - DRV_CounterFregRead - DRV_QCounterStart - DRV_QCounterRead - DRV_CounterPulseStart - DRV_CounterReset - DRV_QCounterConfigSys - DRV_QcounterConfig - DRV_TimerCountSetting
adAlarm	- AlarmConfig - AlarmEnable - AlarmCheck - AlarmReset	- DRV_AlarmConfig - DRV_AlarmEnable - DRV_AlarmCheck - DRV_AlarmReset
FastAI	- FAIINTStart - FAIINTScanStart - FAIDMAStart - FAIDMAScanStart - FAIDualDMAStart - FAIDualDMAScanStart - FAITransfer - FAICheck - ClearOverrun - FAIStop - FAIDmaExStart	- DRV_FAIINTStart - DRV_FAIINTScanStart - DRV_FAIDMAStart - DRV_FAIDMAScanStart - DRV_FAIDualDMAStart - DRV_FAIDualDMAScanStart - DRV_FAITransfer - DRV_FAICheck - DRV_ClearOverrun - DRV_FAIStop - DRV_FAIDmaExStart
FastAO	- FAOINTStart - FAODMAStart - FAOScale - FAOLoad - FAOCheck - FAOStop	- DRV_FAOINTStart - DRV_FAODMAStart - DRV_FAOScale - DRV_FAOLoad - DRV_FAOCheck - DRV_FAOStop
WatchDog	- FAIWatchdogCfg - FAIINTWatchdogStart - FAIDMAWatchdogStart - FAIWatchdogCheck	 - DRV_FAIINTWatchdogStart - DRV_FAIDMAWatchdogStart - DRV_FAIWatchdogCheck
COM Port	- COMOpen	- COMOpen

	- COMClose - COMGetConfig - COMSetConfig - COMRead - COMWrite - COMWrite232 - COMEscape	- COMClose - COMGetConfig - COMSetConfig - COMRead - COMWrite - COMWrite232 - COMEscape
PWM	- CounterPWMEnable - CounterPWMSetting - DICounterReset - EnableEventEx - FDITransfer - PWMStartRead	- DRV_CounterPWMEnable - DRV_CounterPWMSetting - DRV_DICounterReset - DRV_EnableEventEx - DRV_FDITransfer - DRV_PWMStartRead
ADAMConfig	- ADAMAIGetHWConfig - ADAMAISetHWConfig - ADAMAOGetHWConfig - ADAMAOSetHWConfig	- DRV_ADAMAIGetHWConfig - DRV_ADAMAISetHWConfig - DRV_ADAMAOGetHWConfig - DRV_ADAMAOSetHWConfig

4.2. DeviceManager

4.2.1 DeviceGetNumOfList

Gets number of the installed devices



Input:

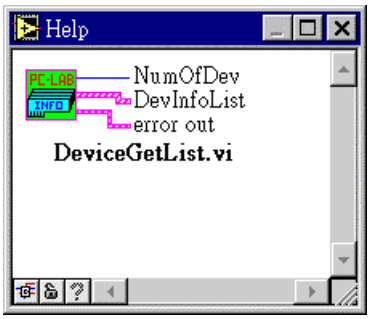
None

Output:

Data type	Parameter	Description
U16	***NumOfDevices***	The number of devices installed in the system by "**DEVINST.EXE**"
error	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.2.2 DeviceGetList

Gets a list of the installed devices not including the devices attached to COM ports.



Input:

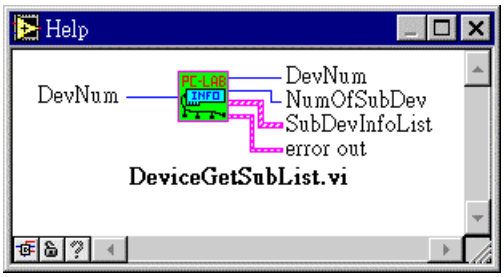
None

Output:

Data type	Parameter	Description
I32	***NumOfDev***	The number of devices installed in the system not including devices attached to COM ports.
[ ]	***DevInfoList***	Specifies the list of the installed devices.
906	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.2.3 DeviceGetSubList

Gets a list of the installed devices on COM port.



Input:

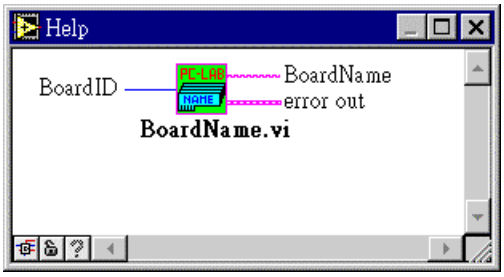
Data type	Parameter	Description
I32	***DevNum***	The device number of the hardware installed in the system by "DEVINST.EXE"

Output:

Data type	Parameter	Description
U16	***DevNum***	The device number of the hardware installed in the system by "**DEVINST.EXE**"
U16	***NumOfSubDev***	The number of devices expansion hardware installed in the system by "**DEVINST.EXE**"
[ ]	***SubDevInfoList***	Specifies the list of the installed expansion or daughterboard devices.
9 0 8	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.2.4 BoardName

Get the item name of a specified device.



Input:

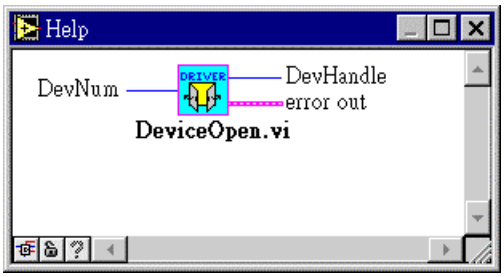
Data type	Parameter	Description
U16	***BoardID***	The device id of the hardware installed in the system by "**DEVINST.EXE**"

Output:

Data type	Parameter	Description
abc	***BoardName***	The device item name of the hardware installed in the system by "**DEVINST.EXE**"
	error out	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.2.5 DeviceOpen

Open the device specified by ***DevNum*** and installed in the system by "**DEVINST.EXE**".



Input:

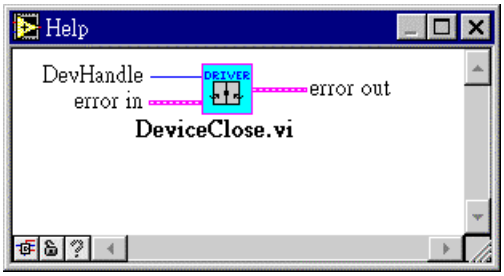
Data type	Parameter	Description
U32	***DevNum***	The device number of the hardware installed in the system by "**DEVINST.EXE**"

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
906	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.2.6 DeviceClose

Close the device specified by ***DevHandle*** and installed in the system by "**DEVINST.EXE**".



Input:

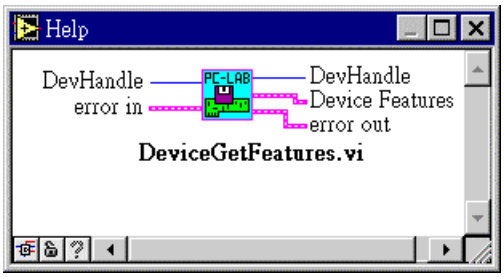
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	error in	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
	error out	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.2.7 DeviceGetFeatures

This function accepts a device handler id and returns the device features of the specified devices.



Input:

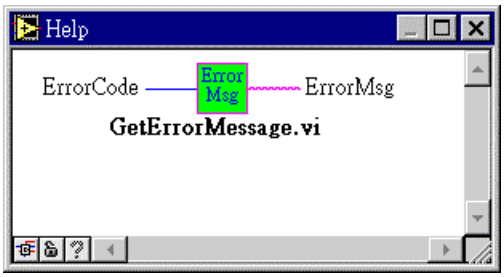
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
901	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
901	***Device Features***	Selected specifications of the device. These specifications are retrieved from a storage area and output by this VI..
901	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.2.8 GetErrorMessage

Retrieves an error message specified by an error code and returns it to the message buffer.



Input:

Data type	Parameter	Description
U16	***ErrorCode***	Specifies error code returned by the driver.

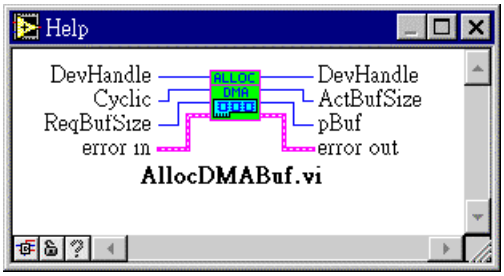
Output:

Data type	Parameter	Description
abc	***ErrorMsg***	Returns the error message string of ***ErrorCode***.

__Note__:For more information of __ErrorCode__ and __ErrorMsg,__ refer to DLL Driver user's manual __Appendix: Error Codes__.

4.2.9 AllocDMABuf

Allocates buffer for DMA data acquisition



Input:

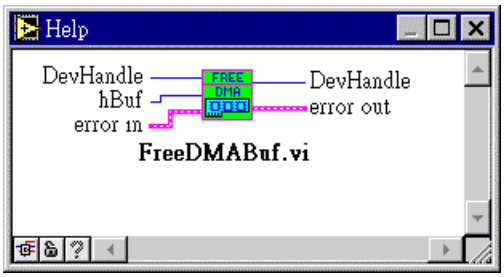
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***Cyclic***	Cyclic mode: **cyclic (1), non cyclic (0)**
U32	***ReqBufSize***	Specifies the size of requested DMA buffer
error	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U32	***ActBufSize***	Returns the size of actual allocated DMA buffer
I32	***pBuf***	A data that stores the allocated memory**.**
error	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.2.10 FreeDMABuf

Releases the buffer allocated by ***AllocateDMABuffer***.



Input:

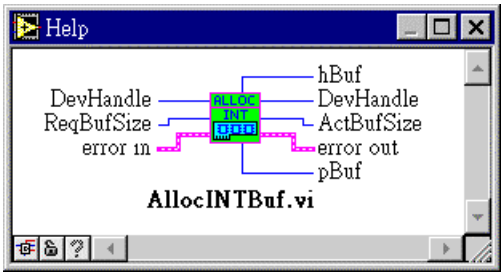
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
I32	***hBuf***	A data that stores the allocated memory**.**
error	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
error	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.2.11 AllocINTBuf

Allocates data buffer of Windows system for interrupt data acquisition.



Input:

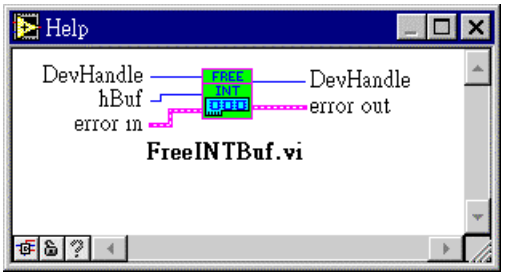
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U32	***ReqBufSize***	Specifies the size of requested INT buffer
808	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U32	***ActBufSize***	Returns the size of actual allocated INT buffer
U32	***pBuf***	A data that stores the allocated memory**.**
808	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.2.12 FreeINTBuf

Releases the buffer allocated by ***AllocateINTBuffer***.



Input:

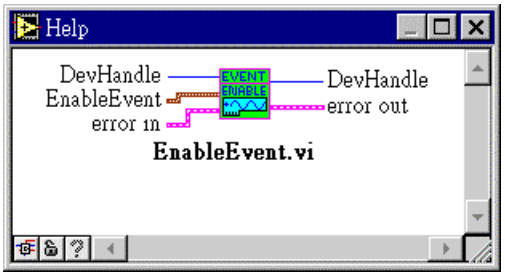
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U32	***hBuf***	A data that stores the allocated memory**.**
90	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
90	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.2.13 EnableEvent

Enables or disables events. This funtion supports both interrupt and DMA features



Input:

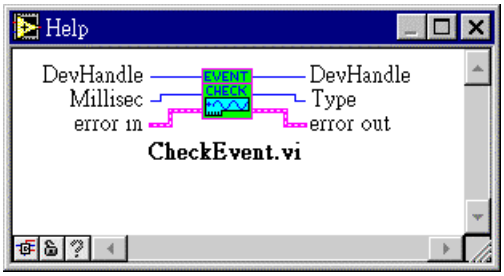
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	EnableEvent	A cluster containing **EventType, Enabled, and Count** information.
	error in	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	error out	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.2.14 CheckEvent

Clears events and reads current status.



Input:

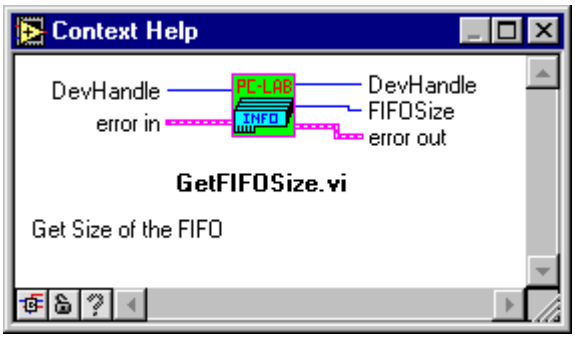
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U32	***Millisec***	Specifies the time to wait for an event. The unit of time is milliseconds.
90	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***Type***	Type of event
90	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.2.15 GetFIFOSize

Get size of the FIFO.



Input:

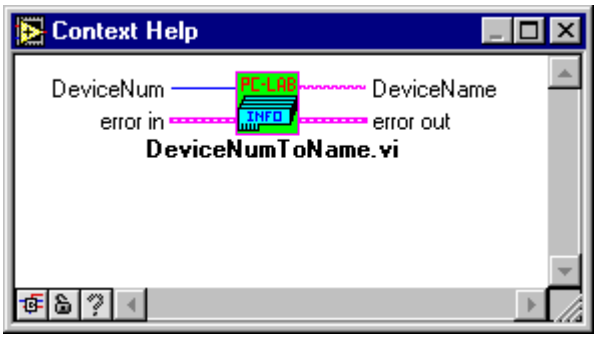
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
906	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
I32	***FIFOSize***	Return size of the FIFO.
906	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.2.116 DeviceNumToDeviceName

Return name of the specified device.



Input:

Data type	Parameter	Description
U32	***DeviceNum***	The device number of the hardware installed in the system by "**DEVINST.EXE**".
906	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

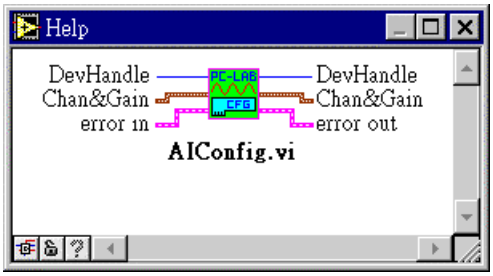
Output:

Data type	Parameter	Description
abc	***DeviceName***	The name of the specified device.
906	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.3. adSlowAI

4.3.1 AIConfig

Configures the input range for the specified analog input channel.



Input:

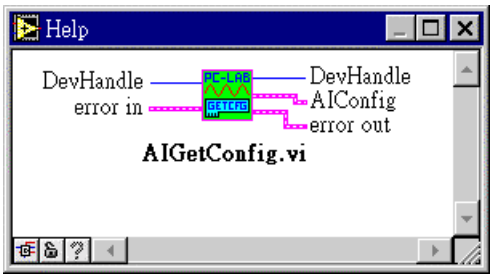
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	Chan&Gain	A cluster data containing **Chan Info** parameter (includes **Channel number**, **ExpBrdID**, and **Expchannel** data) and **Gain code** for specified device.
	error in	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	Chan&Gain	A cluster data containing **Chan Info** parameter (includes **Channel number**, **ExpBrdID**, and **Expchannel** data) and **Gain code** for specified device.
	error out	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.3.2 AIGetConfig

Retrieves analog input configuration data and returns it to a buffer.



Input:

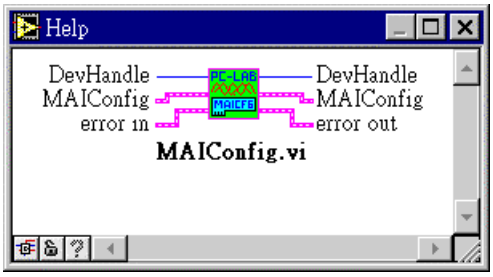
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	error in	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	AIConfig	A cluster of data containing sampled **channel** and **gain** code.
	error out	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.3.3 MAIConfig

Configures the input ranges for the specified analog input channels.



Input:

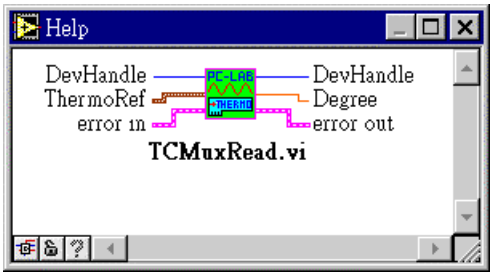
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	MAIConfig	A cluster of data containing **NumChan**, **StartChan** and **GainArray** value.
	error in	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	MAIConfig	A cluster of data containing **NumChan**, **StartChan** and **GainArray** value.
	error out	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.3.4 TCMuxRead

Measures a temperature using an expansion board, for example, PCLD-788/789/789D/8115/889.



Input:

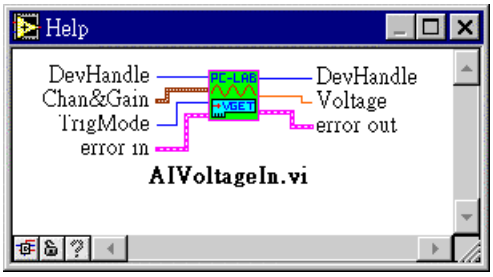
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
906	***ThermoRef***	A cluster of data containing **DasChan, DasGain, ExpChan, TCType, TempScale** and **temp** value.
906	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***Degree***	Returns value of temperature.
906	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.3.5 AIVoltageIn

Reads an analog input channel and returns the result scaled to a voltage (units = volts).



Input:

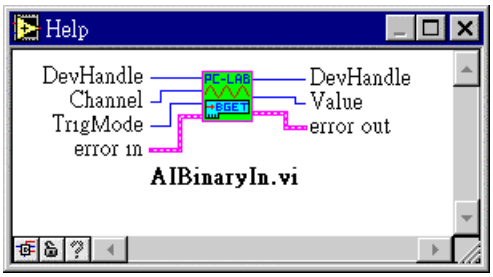
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
9D6	***Chan&Gain***	A cluster of data containing **Chan Info** parameter (includes **Channel number, ExpBrdID**, and **Expchannel** data) and **Gain codes** for a specified device.
U16	***TrigMode***	**0**-normal (software), **1**-external
9D6	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
SGL	***Voltage***	Returns the value of the voltage.
9D6	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.3.6 AIBinaryIn

Reads an analog input channel and returns the unscaled result.



Input:

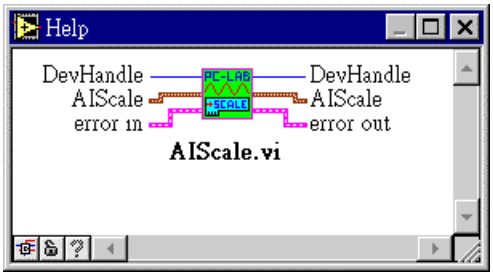
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***Channel***	**Channel** value specifies a digital or analog channel number
U16	***TrigMode***	**0**-normal (software), **1**-external
error	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***Value***	Return the value of the specified channel
error	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.3.7 AIScale

Convert the binary result from an ***AIBinaryIn*** call to the actual input voltage.



Input:

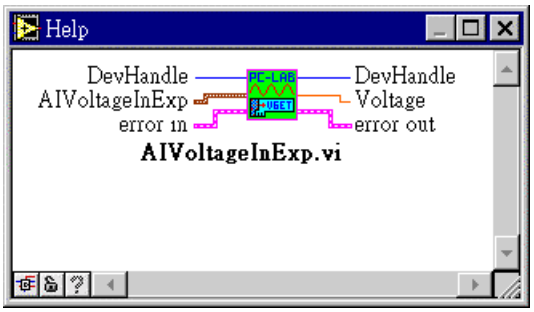
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	AIScale	A cluster containing **MaxVolt, MaxCount, offset** and **Voltage** values.
	error in	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	AIScale	A cluster containing **MaxVolt, MaxCount, offset** and **Voltage** values.
	error out	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.3.8 AIVoltageInExp

Reads an analog input channel using expansion board and returns the result scaled to a voltage (units = volts). This function supports the expansion boards: PCLD-770/779/789/789D/788.



Input:

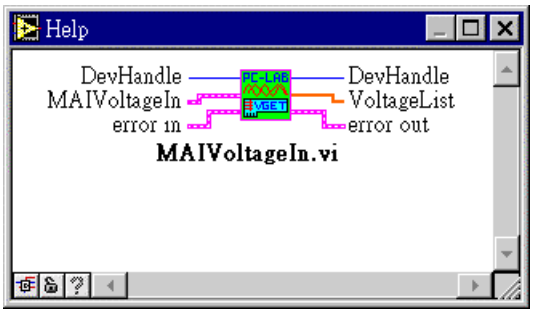
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	AIVoltageInExp	A cluster containing **DasChan, DasGain, ExpChan** and **voltage** values
	error in	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
SGL	***Voltage***	Returns the value of voltage.
	error out	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.3.9 MAIVolatgeIn

Reads analog input channels and returns the results scaled to voltages (units = volts).



Input:

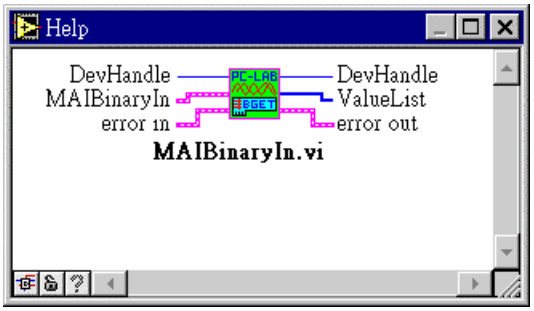
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	MAIVoltageIn	A cluster containing **NumChan, StartChan, GainArray, TrigMode** and **VoltageArray** values
	error in	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
[ ]	***VoltageList***	Returns the values of a list of voltages.
	error out	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.3.10 MAIBinaryIn

Reads analog input channels and returns the unscaled results.



Input:

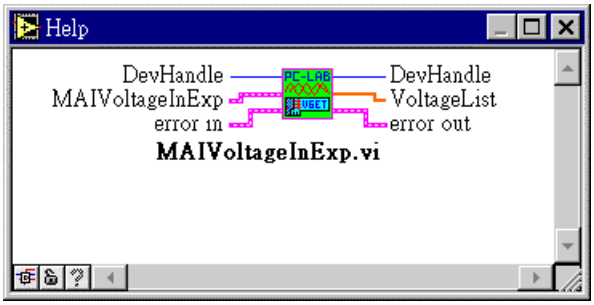
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	MAIBinaryIn	A cluster containing **NumChan, StartChan, TrigMode** and **ReadingArra**y values
	error in	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
[ ]	***ValueList***	Returns the values of a list of raw data.
	error out	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.3.11 MAIVoltageInExp

Reads an analog input channel with an expansion board and returns the result scaled to a voltage in units of volts. This function supports the expansion boards: PCLD-770/779/789/789D/788.



Input:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	MAIVoltageInExp	A cluster containing **NumChan, DasChanArray, DasGainArray, ExpChanArray** and **VoltageArray** values
	error in	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

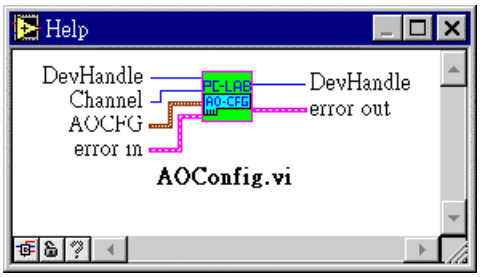
Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	VoltageList	Returns the value of a list of voltages.
	error out	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.4. adSlowAO

4.4.1 AOConfig

Records the output range and polarity for each analog output channel selected. Its use is optional.



Input:

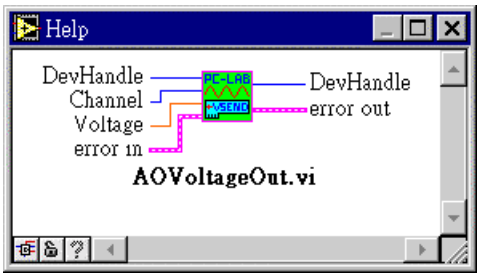
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***Channel***	The channel number of a specified device
	AOCFG	A cluster containing **RefSrc, MaxValue** and **MinValue** data
	error in	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	error out	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.4.2 AOVoltageOut

Accepts a floating-point voltage value, scales it to the proper binary number, and writes that number to an analog output channel to change the output voltage.



Input:

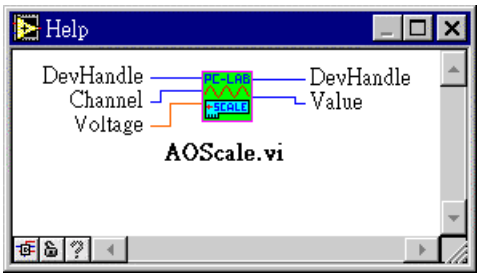
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***Channel***	The channel number of a specified device
SGL	***Voltage***	The value of output voltage.
error	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
error	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.4.3 AOScale

Scales a voltage to a binary value that, when written to one of the analog output channels, produces the specified voltage.



Input:

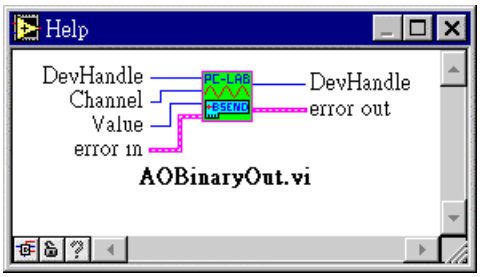
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***Channel***	The channel number of a specified device
SGL	***Voltage***	The voltage value needing to be changed to be a binary value.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***Value***	Binary output value scaled from input voltage.

4.4.4 AOBinaryOut

Writes a binary value to one of the analog output channels, changing the voltage produced at the channel.



Input:

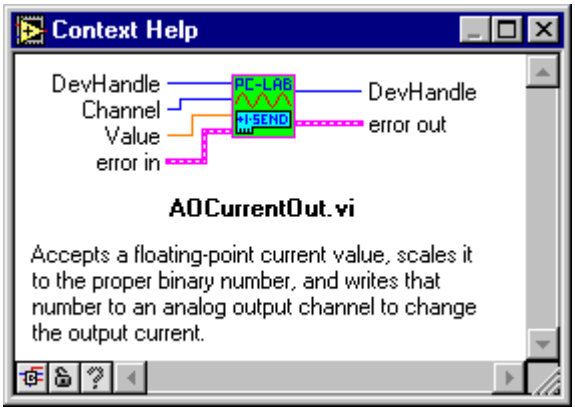
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***Channel***	The channel number of specified device
U16	***Value***	Output binary value, for use in analog output device.
	error in	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	error out	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.4.5 AOCurrentOut

Accepts a floating-point current value, scales it to the proper binary number, and writes that number to an analog output channel to change the output current.



Input:

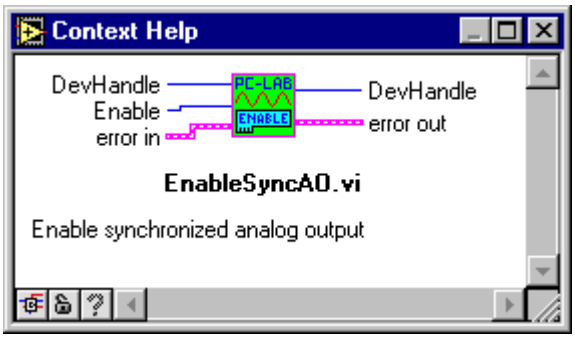
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***Channel***	The channel number of a specified device
SGL	***Value***	The current value needing to be changed to be a binary value.
error	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
error	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.4.6 EnableSyncAO

Enable synchronized analog output.



Input:

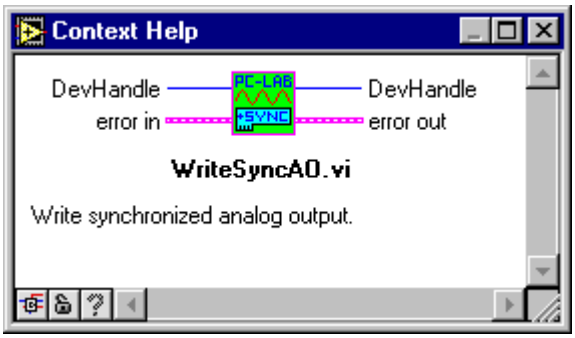
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U32	***Enable***	Ture is enable, false is disable
90	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
90	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.4.7 WriteSyncAO

Write synchronized analog output.



Input:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
90	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

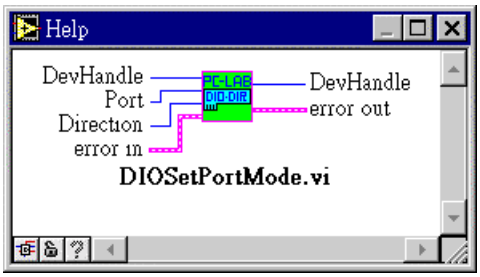
Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
90	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.5. adDIO

4.5.1 DIOSetPortMode

Configures the specified port for input or output. This function only supports PCL-722/724/731.



Input:

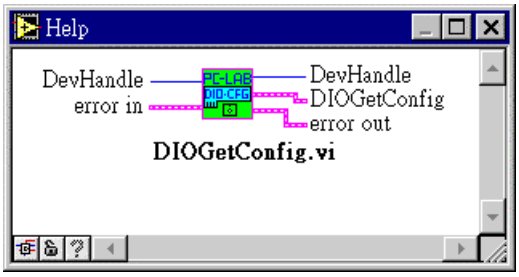
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***Port***	The digital port number
U16	***Direction***	Direction: **input (0)** or **output(1)**
9D8	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
9D8	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.5.2 DIOGetConfig

Reads digital input and output configuration data stored in the Registry or a configuration file. The application calls this function by passing the configuration data's memory address



Input:

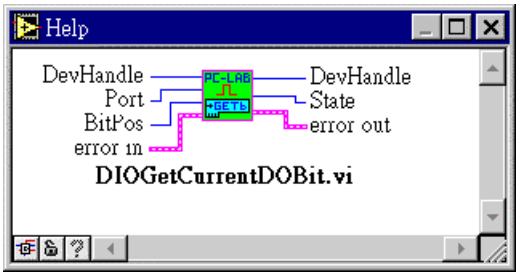
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program..
906	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
906	***DIOGetConfig***	A cluster containing **PortArray** and **NumOfPorts** data.
906	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.5.3 DIOGetCurrentDOBit

Returns the value stored at a specified bit position from the specified digital I/O port. The port is specified by port number which ranges from 0 to the maximum number of ports provided by the devices. For example, PCL-722 has up to 18 ports digital output. The port number of the board is from 0 to 17



Input:

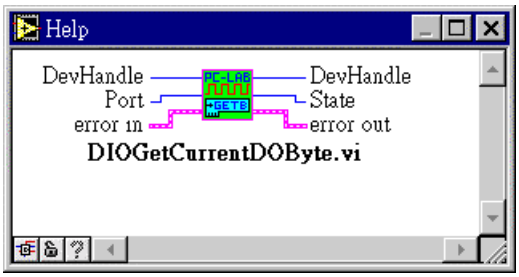
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***Port***	The port number of specified device
U16	***BitPos***	The specified position in the port.
	error in	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***State***	Bit data read from the specified port (0 or 1).
	error out	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.5.4 DIOGetCurrentDOByte

Returns byte data from the specified I/O port of a device. The port is specified by port number which ranges from 0 to the maximum number of ports provided by the device. For example, PCL-722 has up to 18 ports digital output. The port number of the board is from 0 to 17.



Input:

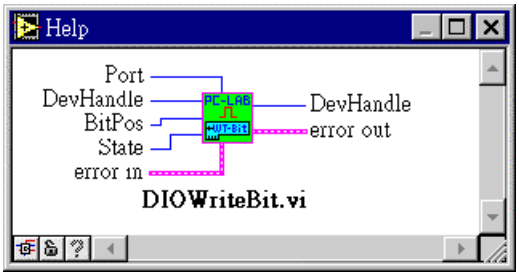
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***Port***	The port number of specified device
8-bit	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***State***	Byte data read from the specified port
8-bit	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.5.5 DIOWriteBit

Writes digital output bit data to the specified digital port. The port is specified by the port number which ranges from 0 to the maximum number of ports provided by the device. For example, PCL-730 has 4 ports of digital output. The port number of the board is from 0 to 3.



Input:

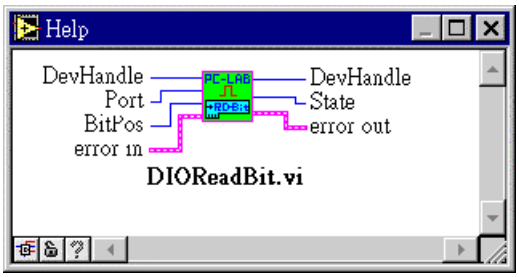
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***Port***	The port number of specified device
U16	***BitPos***	The specified bit position in the port.
U16	***State***	Bit value will be written to the specified bit position
error	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
error	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.5.6 DIOReadBit

Returns the bit state of digital input from the specified digital I/O port. The port is specified by port number which ranges from 0 to the maximum number of ports provided by the device. For example, PCL-722 has up to 18 ports digital output. The port number of the board is from 0 to 17.



Input:

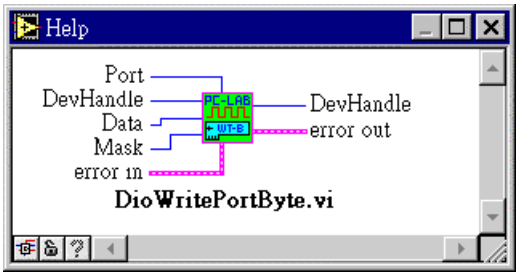
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***Port***	The port number of specified device
U16	***BitPos***	The specified bit position in the port.
9Db	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***State***	Data value read from the specified bit position.
9Db	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.5.7 DIOWritePortByte

Writes output byte data to the specified port.



Input:

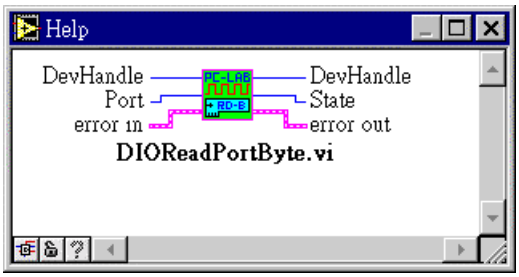
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***Port***	The port number of specified device
U16	***Data***	The byte data will be written to specified port.
U16	***Mask***	Specifies which bit(s) of specified byte data should be sent to the output port and which bits remain unchanged
error	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
error	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.5.8 DIOReadPortByte

Returns input byte data from the specified I/O port. The port is specified by port number which ranges from 0 to the maximum number of ports provided by the device. For example, PCL-722 has up to 18 digital output ports. The port number of the board is from 0 to 17.



Input:

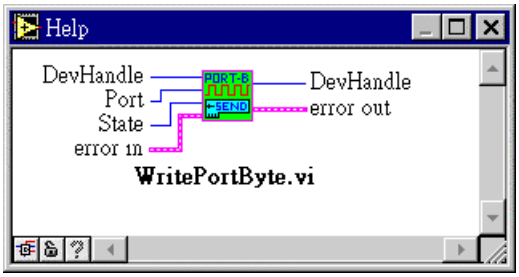
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***Port***	The port number of specified device
error	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***State***	Byte data read from the specified port
error	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.5.9 WritePortByte

Writes an 8-bit data to the specified I/O port. The port address is an I/O port address on the PC.



Input:

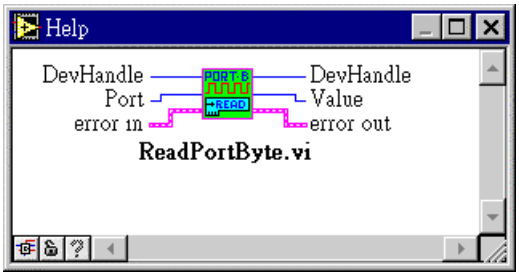
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***Port***	The port number of specified device
U16	***State***	Byte data written to the specified port
error	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program..
error	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.5.10 ReadPortByte

Reads a 8-bit of byte data from the specified I/O port. The port address is an I/O port address on the PC.



Input:

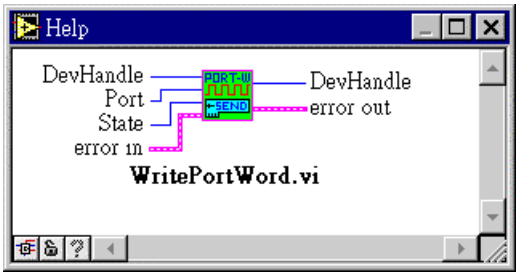
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program..
U16	***Port***	The port number of specified device
90	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program..
U16	***Value***	Data value read from specified port
90	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.5.11 WritePortWord

Writes a 16-bit of word data to the specified I/O port. The port address is an I/O port address on the PC.



Input:

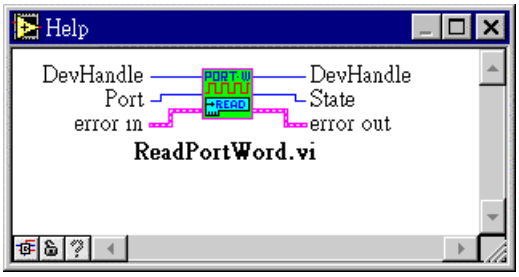
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program..
U16	***Port***	The port number of specified device
U16	***State***	Word data written to the specified port
error	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program..
error	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.5.12 ReadPortWord

Reads a 16-bit of word data from the specified I/O port. The port address is an I/O port address on the PC.



Input:

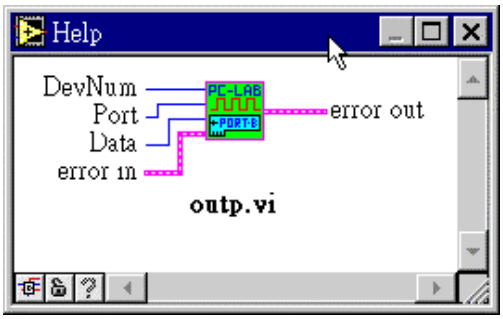
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program..
U16	***Port***	The port number of specified device
90	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program..
U16	***State***	Word data read from the specified port
90	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.5.13 Outp

Directly outputs byte data to a specified port of a defined device using the device number.



Input:

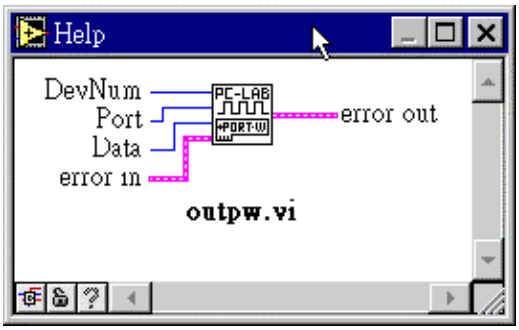
Data type	Parameter	Description
U32	***DevNum***	The device number of the hardware installed in the system by "**DEVINST.EXE**"
U16	***Port***	The port number of specified device
U16	***Data***	Byte data written to the specified port.
error	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
error	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.5.14 Outpw

Directly outputs word data to a specified port of a defined device using the device number.



Input:

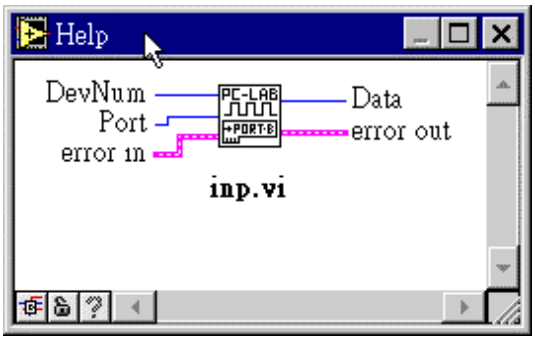
Data type	Parameter	Description
U32	***DevNum***	The device number of the hardware installed in the system by "**DEVINST.EXE**"
U16	***Port***	The port number of specified device
U16	***Data***	Word data written to the specified port.
error	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
error	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.5.15 Inp

Directly inputs byte data from a specified port of a defined device using the device number.



Input:

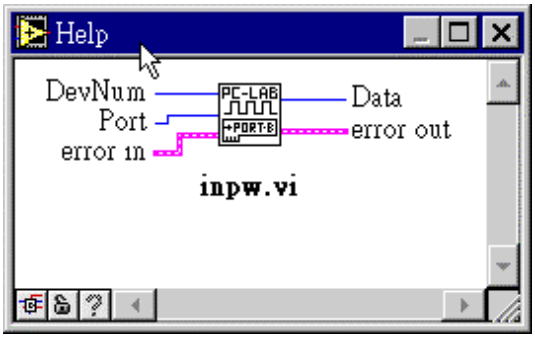
Data type	Parameter	Description
U32	***DevNum***	The device number of the hardware installed in the system by "**DEVINST.EXE**"
U16	***Port***	The port number of specified device
error	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
U16	***Data***	Byte data read from the specified port.
error	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.5.16 Inpw

Directly inputs word data from a specified port of a defined device using the device number.



Input:

Data type	Parameter	Description
U32	***DevNum***	The device number of the hardware installed in the system by "**DEVINST.EXE**"
U16	***Port***	The port number of specified device
error	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

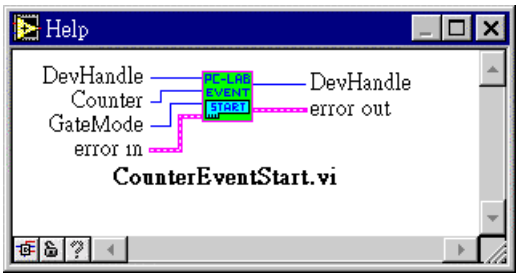
Output:

Data type	Parameter	Description
U16	***Data***	Word data read from the specified port.
error	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.6 adCountTimer

4.6.1 CounterEventStart

Configures the specified counter for an event-counting operation and starts the counter.



Input:

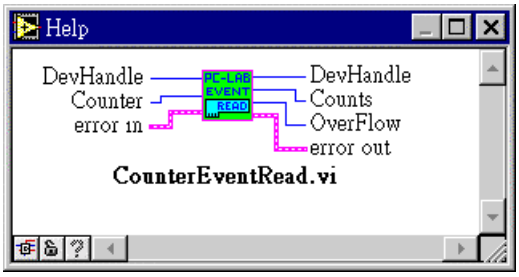
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program..
U16	***Counter***	Counter number of the specified device.
U16	***GateMode***	Gating mode to be used for AMD Am9513A
905	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program..
905	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.6.2 CounterEventRead

Reads the current counter total without disturbing the counting process and returns the count and overflow conditions.



Input:

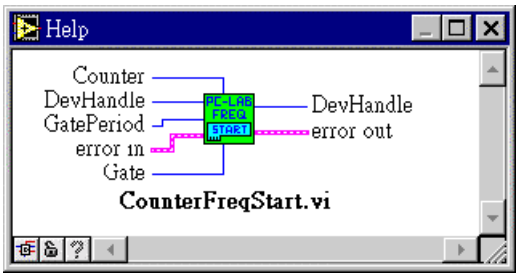
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program..
U16	***Counter***	Counter number of the specified device.
90	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program..
U16	***Counts***	The count value of the specified counter
U16	***OverFlow***	Overflow status of the counter, **1** means **overflow**, **0** means **no overflow.**
90	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.6.3 CounterFreqStart

Configures a specified counter for frequency measurement and starts the counter.



Input:

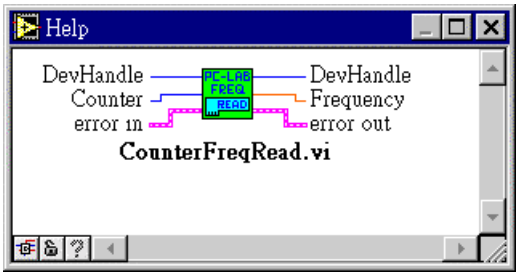
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program..
U16	***Counter***	Counter number of the specified device.
U16	***GatePeriod***	Gating period in seconds for AMD Am9513A
U16	***Gate***	Gating mode to be used for AMD Am9513A
error	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program..
error	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.6.4 CounterFreqRead

Reads the frequency measurement.



Input:

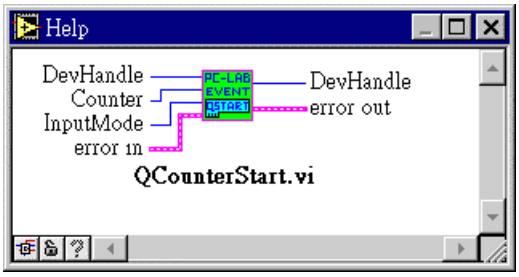
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program..
U16	***Counter***	Counter number of the specified device.
906	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program..
SGL	***Frequency***	Returned frequency value of the counter.
906	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.6.5 QCounterStart

Configures the specified counter for an event-counting operation and starts the counter. This function only supports PCL-833.



Input:

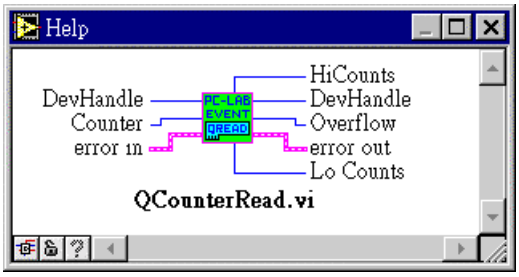
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***Counter***	Counter number of the specified device.
U16	***InputMode***	Input mode control : **DISABLE(0), ABPHASEX1(1), ABPHASEX2(2), ABPHASEX4(3), TWOPULSEIN(4), ONEPULSEIN(5)**
908	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
908	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.6.6 QCounterRead

Reads the current counter total without disturbing the counting process and returns the count and overflow conditions.



Input:

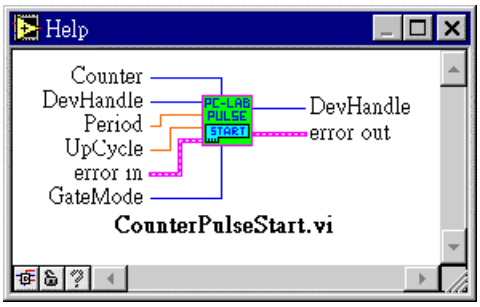
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***Counter***	Counter number of the specified device
error	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***Overflow***	Overflow state of the counter, **1** means **overflow, 0** means **no overflow**
U32	***HiCounts***	The high 32-bit of returned count value.
U32	***LoCounts***	The low 32-bit of returned count value.
error	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.6.7 CounterPulseStart

Configures the specified counter for pulse output and starts the counter.



Input:

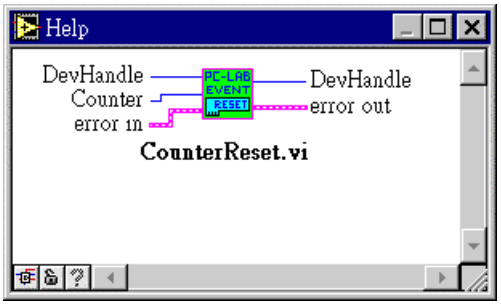
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***Counter***	Counter number of the specified device
SGL	***Period***	The period time of a pulse (Its unit is a second)
SGL	***UpCycle***	The first 1/2 cycle length in seconds for AMD Am9513A
U16	***GateMode***	Gating mode to be used for AMD Am9513A
906	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
906	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.6.8 CounterReset

Turns off the specified counter operation. This function supports boards with the timer/counter chip (i.e. Intel 8254 or AMD Am9513A) and PCL-833.



Input:

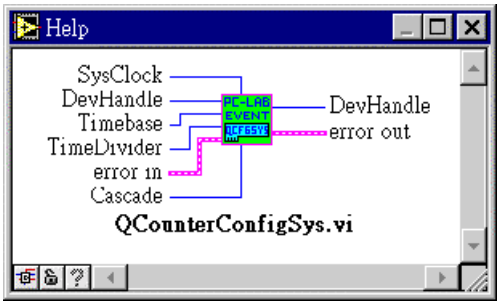
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***Counter***	Counter number of the specified device
906	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
906	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.6.9 QCounterConfigSys

Configures the system clock of the digital filter, time period for latching and cascade mode. This function only supports PCL-833.



Input:

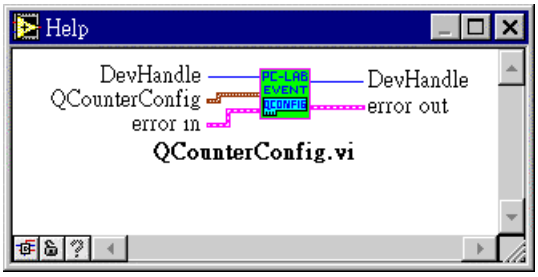
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***SysClockClock***	Frequency for digital filter. The value could be set to **SYS8MHZ**(0), **SYS4MHZ**(1), **SYS2MHZ**(2)
U16	***Timebase***	16C54 time base control.
U16	***TimeDivider***	Divider control value
U16	***Cascade***	Cascade mode **NOCASCADE**(0), **CASCADE**(1)
error	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
error	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.6.10 QCounterConfig

Configures the specified counter for an event-counting operation. This function only supports PCL-833.



Input:

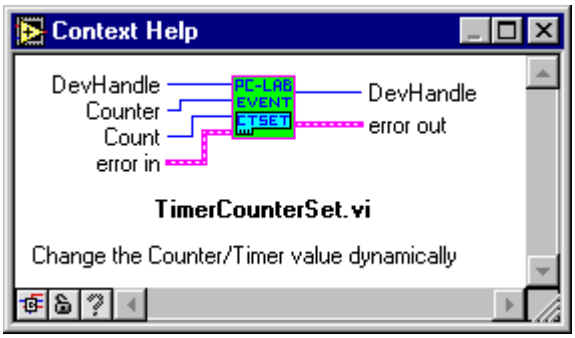
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	QCounterConfig	A cluster containing **Counter; LatchSrc, LatchOverflow, ResetOnLatch,** and **ResetValue** data
	error in	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	error out	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.6.11 TimerCountSetting

Change the Counter/Timer value dynamically.



Input:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***Counter***	Counter number of the specified device.
U32	***Count***	Input value of the specified counter.
error	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

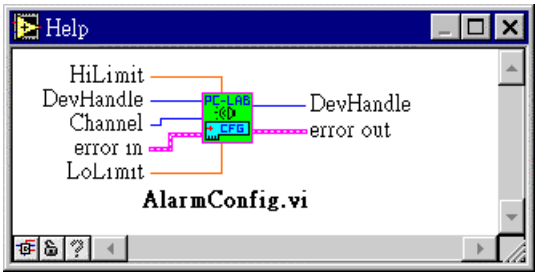
Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
error	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.7 adAlarm

4.7.1 AlarmConfig

Configures the high and low limit value of the specified channel for alarm monitoring. This function only supports ADAM modules.



Input:

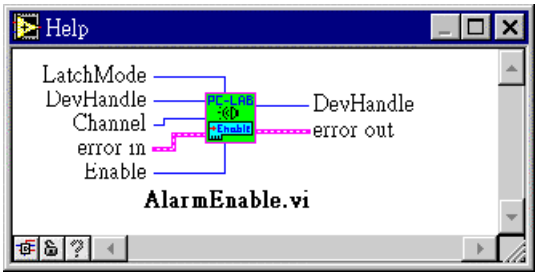
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***Channel***	The channel for alarm monitoring
SGL	***HiLimit***	High limit value for alarm monitoring
SGL	***LoLimit***	Low limit value for alarm monitoring
error cluster	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
error cluster	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.7.2 AlarmEnable

Enables the alarm in either momentary or latching mode. This function only supports ADAM modules.



Input:

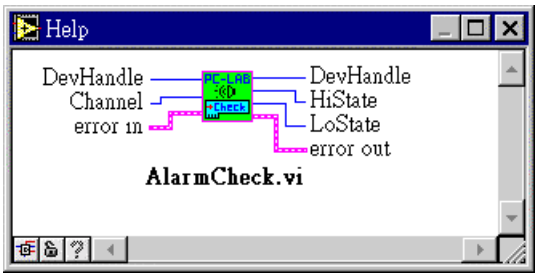
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***Channel***	The channel for alarm monitoring
U16	***LatchMode***	Momentary(0), Latching(1)
U16	***Enable***	Enable(1), Disable(0)
	error in	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	error out	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.7.3 AlarmCheck

Checks the alarm status of the specified channel.



Input:

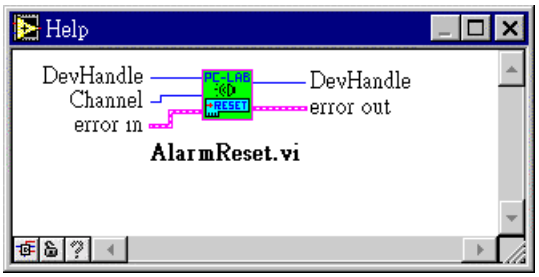
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***Channel***	The channel for alarm monitoring
error	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***LoState***	The current state of the low alarm limit
U16	***HiState***	The current state of the high alarm limit
error	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.7.4 AlarmReset

Resets the alarm monitoring of the specified channel.



Input:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***Channel***	The channel for alarm monitoring
error	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

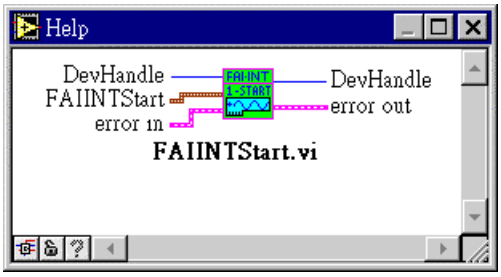
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
error	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.8 FastAI

4.8.1 FAIINTStart

Initiates an asynchronous, single-channel data acquisition operation with interrupt transfer and stores its input in an array.

Note: 1. While FIFO enabled, VI's count (number of conversion) must be a multiple of FIFO size
2. The FIFO size of PCL-818HD and PCL-818HG is 512.



Input:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	FAIINTStart	A cluster containing **TrigSrc, SampleRate, chan, gain, buffer, count, cyclic,** and **IntrCount** data
	error in	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

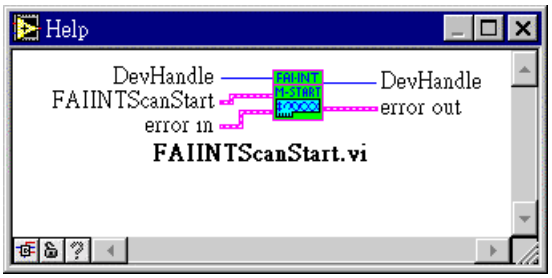
Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	error out	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.8.2 FAIINTScanStart

Initiates an asynchronous, multiple-channel data acquisition operation with Interrupt and stores its input in an array and the gain codes for the scan channels.

Note: 1. While FIFO enabled, VI's count (number of conversion) must be a multiple of FIFO size
2. The FIFO size of PCL-818HD and PCL-818HG is 512.



Input:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	FAIINTScanStart	A cluster containing **TrigSrc, SampleRate, NumChans, StartChan, buffer, count, cyclic,** and **IntrCount** data
	error in	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

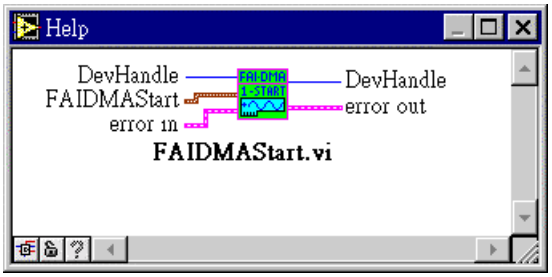
Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	error out	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.8.3 FAIDMAStart

Initiates an asynchronous, single-channel data acquisition operation with DMA and stores its input in an array.

Note: DMA buffer size must be bigger than 2048 while using DMA transfer



Input:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	FAIDMAStart	A cluster containing **TrigSrc, SampleRate, chan, gain, buffer** and **count** data
	error in	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

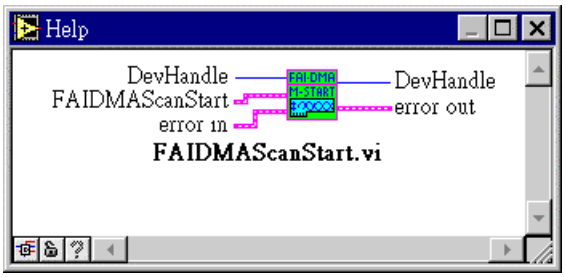
Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	error out	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.8.4 FAIDMAScanStart

Initiates an asynchronous, multiple-channel data acquisition operation with DMA and stores its input in an array and the gain codes for the scan channels.

Note: DMA buffer size must be bigger than 2048 while using DMA transfer



Input:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
906	***FAIDMAScanStart***	A cluster containing **TrigSrc, SampleRate, NumChans, StartChan, GainList, buffer,** and **count** data
906	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

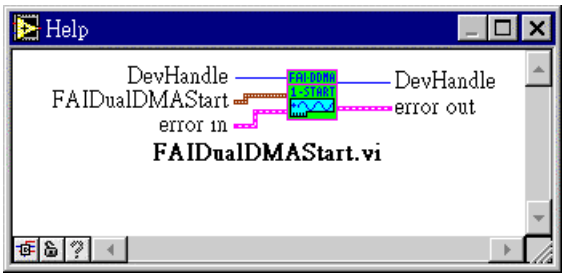
Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
906	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.8.5 FAIDualDMAStart

Initiates an asynchronous, single-channel data acquisition operation with Dual-DMA and stores its input in an array(for PCL-1800 only).

Note: DMA buffer size must be bigger than 2048 while using DMA transfer



Input:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	FAIDualDMAStart	A cluster containing **TrigSrc, SampleRate, Gain, Chan, Count, BufferA** and **BufferB** data
	error in	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

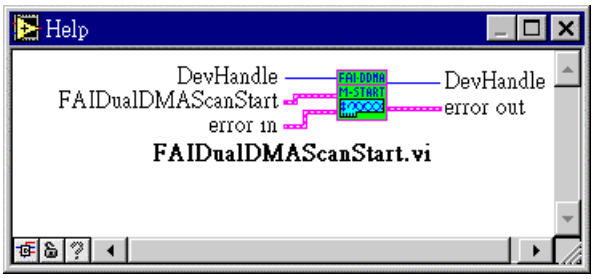
Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	error out	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.8.6 FAIDualDMAScanStart

Initiates an asynchronous, mutiple-channel data acquisition operation with Dual-DMA and stores its input in an array and the gain codes for the scan channels(for PCL-1800 only).

Note: DMA buffer size must be bigger than 2048 while using DMA transfer



Input:

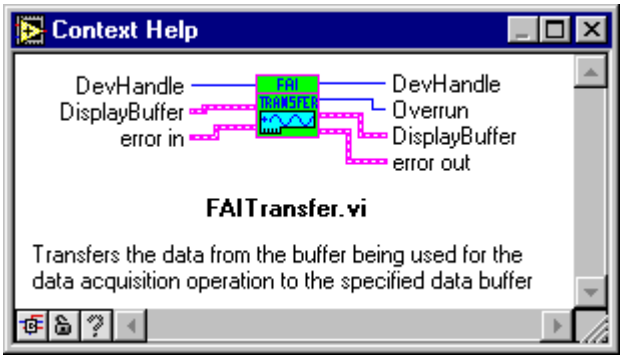
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	FAIDualDMAScanStart	A cluster containing **TrigSrc, SampleRate, NumChan, StartChan, GainList, BufferA, BufferB,** and **count** data
	error in	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	error out	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.8.7 FAITransfer

Transfers the data from the buffer being used for the data acquisition operation to the specified data buffer.



Input:

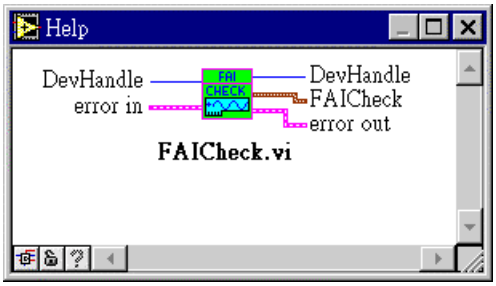
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
90	***DisplayBuffer***	A cluster containing **HalfBufSize, VoltageBuf, BinaryBuf …**
90	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
90	***DisplayBuffer***	A cluster containing **HalfBufSize, VoltageBuf, BinaryBuf …**
U16	***Overrun***	Overrun status: **overrun (1), no overrun (0)**
90	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.8.8 FAICheck

Checks if the current data acquisition operation is complete and return current status.



Input:

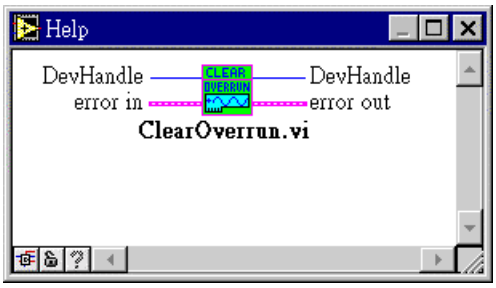
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	error in	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	FAICheck	A cluster containing **ActiveBuf, stopped, retrieved** and **overrun** data
	error out	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.8.9 ClearOverrun

Clears overrunning flag.



Input:

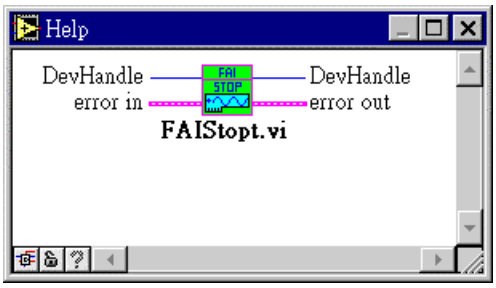
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
error cluster	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
error cluster	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.8.10 FAIStop

Cancels the current data acquisition operation and resets the hardware and software.



Input:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	error in	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

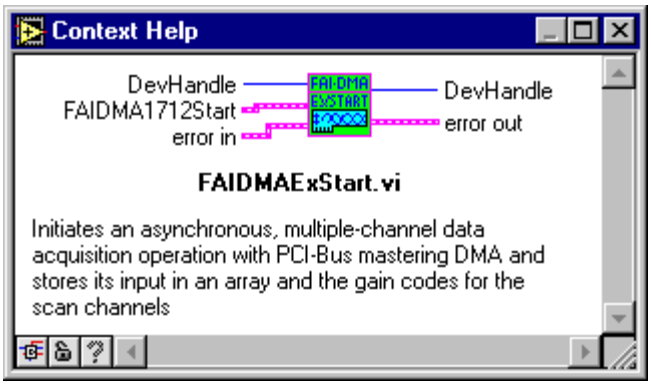
Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	error out	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.8.11 FAIDMAExStart

Initiates an asynchronous, multiple-channel data acquisition operation with PCI-Bus mastering DMA and stores its input in an array and the gain codes for the scan channels(for PCI-1712 only).

Note: DMA buffer size must be bigger than 2048 while using DMA transfer



Input:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
906	***FAIDMA1712Start***	A cluster containing **TrigSrc, SampleRate, NumChans, StartChan, GainList, buffer,** and **count** …
906	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

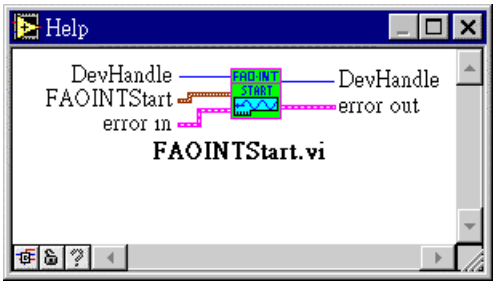
Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
906	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.9 FastAO

4.9.1 FAOINTStart

Initiates an asynchronous analog output operation with interrupt transfer.



Input:

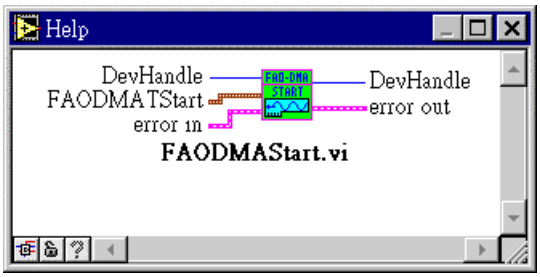
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	FAOINTStart	A cluster containing **StartChan, StopChan, buffer, count** and **cyclic** data
	error in	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	error out	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.9.2 FAODMAStart

Initiates an asynchronous analog output operation with DMA transfer.



Input:

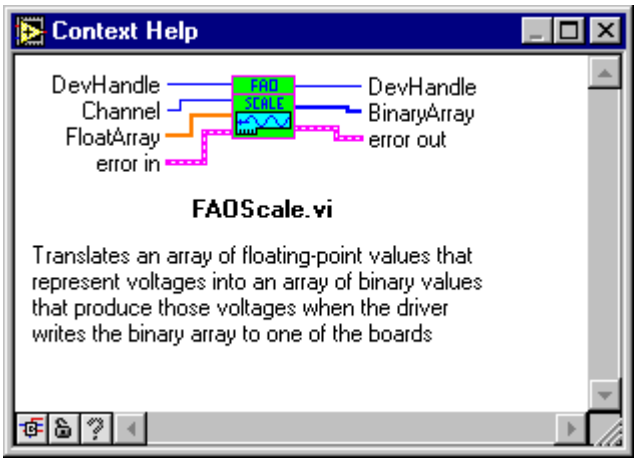
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	FAODMAStart	A cluster containing **StartChan, StopChan, buffer, count** and **cyclic** data
	error in	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	error out	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.9.3 FAOScale

Translates an array of floating-point values that represent voltages into an array of binary values that produce those voltages when the driver writes the binary array to one of the boards. This function uses the current analog output configuration settings to perform the conversions.



Input:

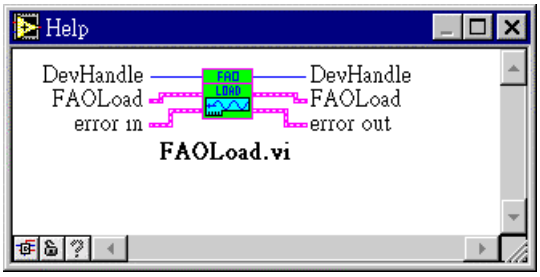
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***Channel***	Channel number
[ ]	***FloatArray***	Input float-point values of data buffer
606	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
[ ]	***BinaryArray***	Output binary values of data buffer
606	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.9.4 FAOLoad

Transfers the data from the buffer being used for the data acquisition operation to the specified data buffer.



Input:

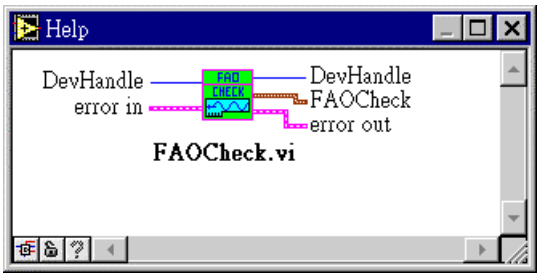
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	FAOLoad	A cluster containing **ActiveBuf, DataBuffer, start** and **count** data
	error in	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	FAOLoad	A cluster containing **ActiveBuf, DataBuffer, start** and **count** data
	error out	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.9.5 FAOCheck

Checks if the current analog output is complete and return current status.



Input:

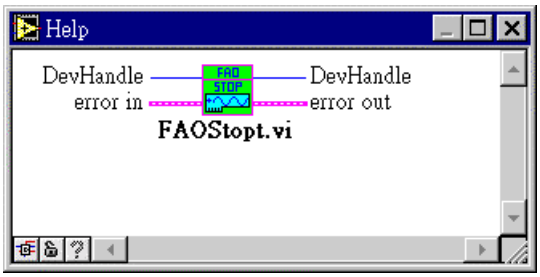
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	error in	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	FAOCheck	A cluster containing **ActiveBuf, stopped, CurrentCount,** and **HalfReady** data
	error out	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.9.6 FAOStop

Cancels the current analog output operation and resets the hardware and software.



Input:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
906	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

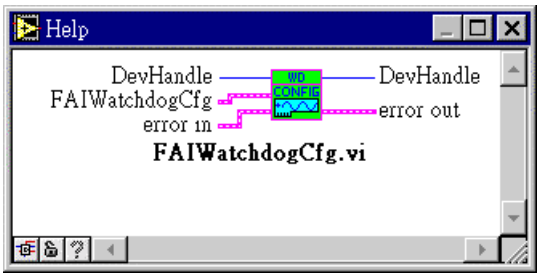
Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
906	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.10 WatchDog

4.10.1 FAIWatchdogCfg

Configures the hardware to acquire data before, before and after or after the signal triggers a analog watchdog. It also configures the condition and level of the analog watchdog for each channel. This function only supports PCL-1800.



Input:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	FAIWatchdogCfg	A cluster containing **TrigMode, NumChans, StartChan, GainList, CondList,** and **LevelList** data
	error in	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

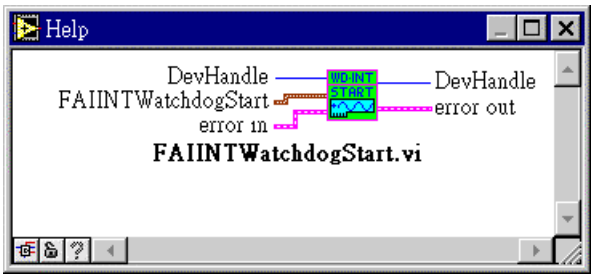
Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	error out	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.10.2 FAIINTWatchdogStart

Initiates an asynchronous data acquisition operation with analog watchdog by interrupt transfer and stores its input in an array.

Note: 1. While FIFO enabled, VI's count (number of conversion) must be a multiple of FIFO size
2. The FIFO size of PCL-818HD and PCL-818HG is 512.



Input:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
906	***FAIINTWatchdogStart***	A cluster containing **TrigSrc, SampleRate, buffer, BufferSize, count, cyclic,** and **IntrCount** data
906	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
906	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.10.3 FAIDMAWatchdogStart

Initiates an asynchronous data acquisition operation with analog watchdog by DMA transfer and stores its input in an array.

Note: DMA buffer size must be bigger than 2048 while using DMA transfer

Input:

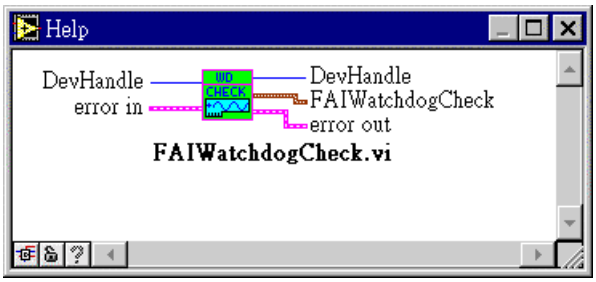
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	FAIDMAWatchdogStart	A cluster containing **TrigSrc, SampleRate, BufferA, BufferB, BufferSize, buffer, count,** and **cyclic** data
	error in	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	error out	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.10.4 FAIWatchdogCheck

Checks if the current data acquisition with watchdog is triggered.



Input:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	error in	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

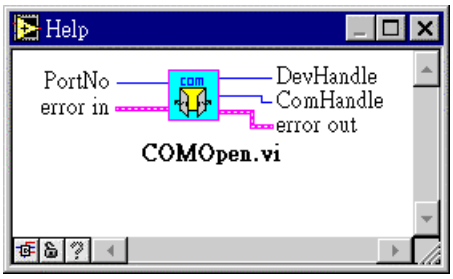
Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	FAIWatchdogCheck	A cluster containing **DataType,ActiveBuf, triggered, TrigChan, TrigIndex** and **TrigData** data
	error out	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.11 COM Port

4.11.1 COMOpen

Opens 1 of 4 serial communication ports (9 serial ports if SuperCom is installed). This function must be called before using any of the functions listed below.



Input:

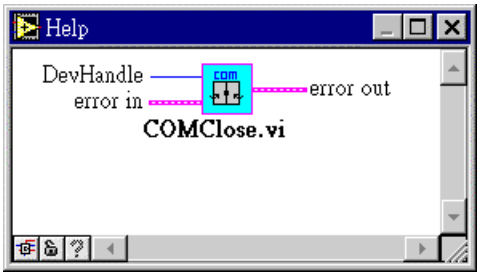
Data type	Parameter	Description
U16	***PortNo***	The number of the serial port
error	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
I32	***ComHandle***	The handler id of the serial port specified by PortNo
error	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.11.2 COMClose

Close the serial port that is opened by *COMOpen*.



Input:

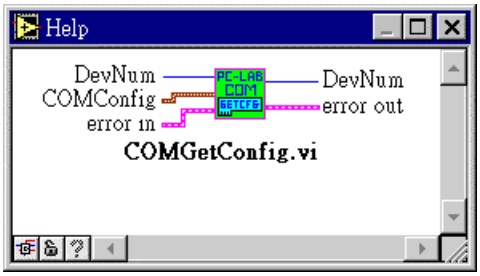
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
error	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
error	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.11.3 COMGetConfig

Retrieves the serial port settings; e.g. port number, baud rate, parity check.



Input:

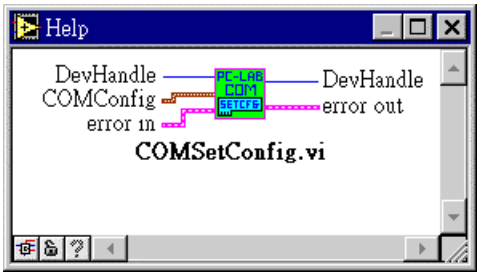
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
906	***COMConfig***	A cluster containing **CommPort, BaudRate, Parity, DataBits, StopBits, TxMode** and **PortAddress** data
906	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
906	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.11.4 COMSetConfig

Set the port communication configuration; e.g. port number, baud rate, parity check.



Input:

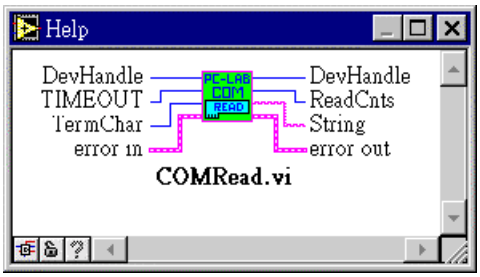
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
906	***COMConfig***	A cluster containing **CommPort, BaudRate, Parity, DataBits, StopBits, TxMode** and **PortAddress** data
906	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
906	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.11.5 COMRead

Reads data from the specified serial port.



Input:

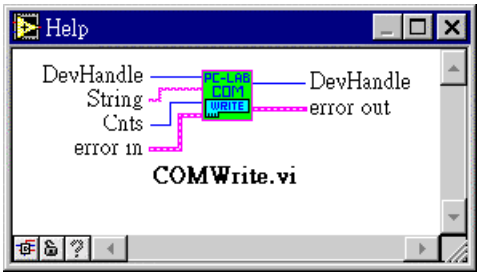
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***TIMEOUT***	The time interval elapsed in milliseconds.
U16	***TermChar***	The termination character for serial port communication.
error	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***ReadCnts***	The number of bytes read from the serial port.
abc	***String***	The data string read from the serial port.
error	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.11.6 COMWrite

Writes data to the specified serial port according to the previous configuration stored in Registry.



Input:

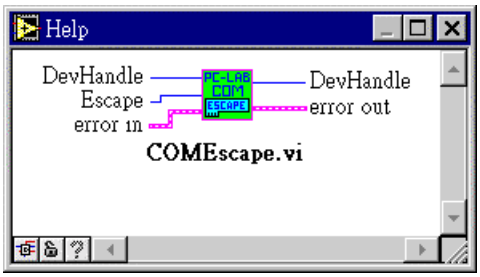
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
abc	***String***	The data string written to the serial port.
U16	***Cnts***	The number of bytes sent to the serial port.
error	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
error	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.11.7 COMEscape

This routine provides “escape” services to the callers.



Input:

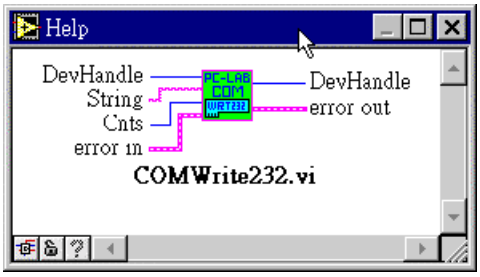
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***Escape***	The escape data for the serial port.
error	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
error	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.11.8 COMWrite232

Writes data to the specified RS-232 serial port according to the previous configuration stored in Registry.



Input:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
abc	***String***	The data string written to the serial port.
U16	***Cnts***	The number of bytes sent to the serial port.
90	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

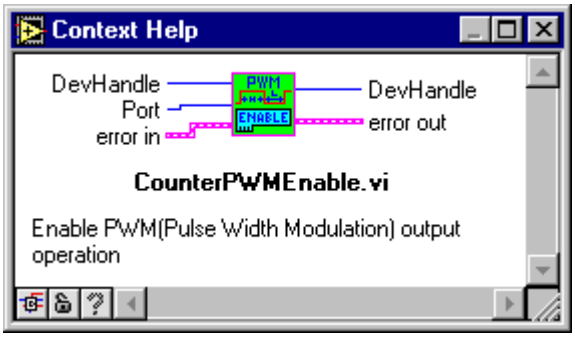
Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
90	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.12 PWM

4.12.1 CounterPWMEnable

Enable PWM(Pulse Width Modulation) output operation.



Input:

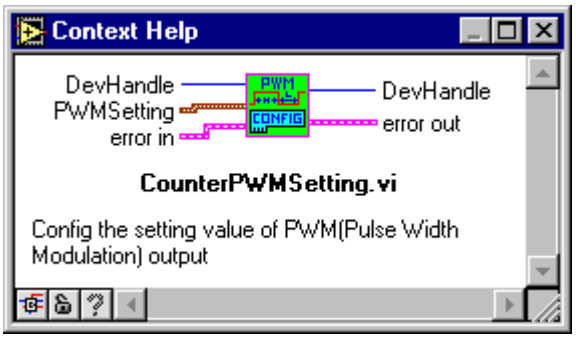
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
I32	***Port***	Enable/Disable port.If bit0 = 1, port0 is enabled. If bit1 = 1,port1 is enabled.
90	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
90	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.12.2 CounterPWMSetting

Config the setting value of PWM(Pulse Width Modulation) output.



Input:

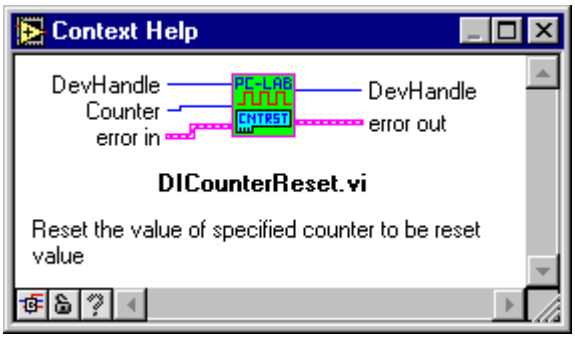
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	PWMSetting	A cluster containing ***Port, Period,HiPeriod, OutCount,*** and ***GateMode***
	error in	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	error out	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.12.3 DICounterReset

Reset the value of specified counter to be reset value.



Input:

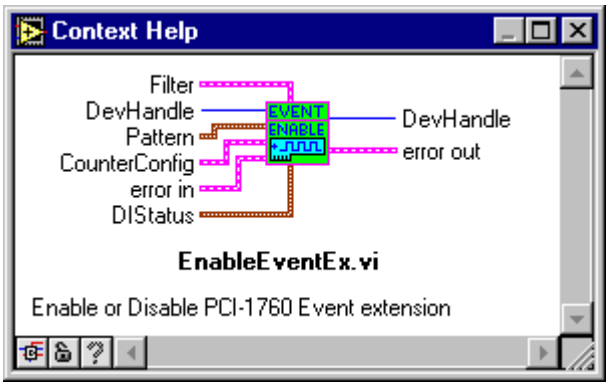
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
I32	***Counter***	Reset counter data.
error	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
error	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.12.4 EnableEventEx

Enable or Disable PCI-1760 Event extension.



Input:

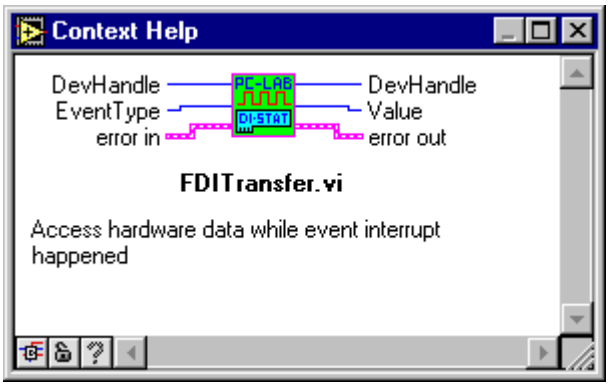
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	Filter	A cluster containing ”Digital Filter”.
	Pattern	A cluster containing ”Pattern Match”.
	CounterConfig	A cluster containing “Counter Match” and “Counter Overflow”
	DIStatus	A cluster containing ”Change of Input State”.
	error in	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	error out	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.12.5 FDITransfer

Access hardware data while event interrupt happened.



Input:

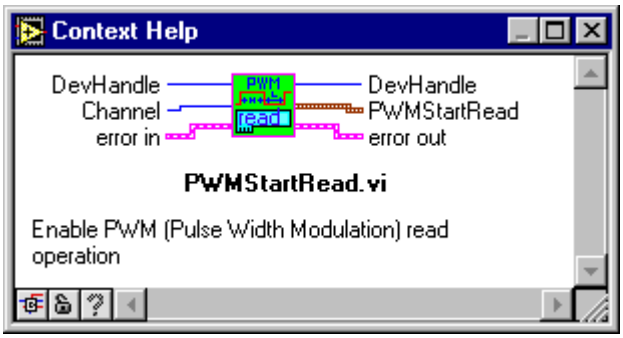
Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***EventType***	The type of event.
error	***error in***	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
I32	***Value***	The return data.
error	***error out***	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

4.12.6 PWMStartRead

Enable PWM (Pulse Width Modulation) read operation.



Input:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
U16	***Channel***	Specified the channel to be read.
	error in	A cluster containing error information, such as the **status**, **code**, and **source** elements that is passed from a previous linked VI.

Output:

Data type	Parameter	Description
I32	***DevHandle***	The handler id of the device specified by ***DevNum*** as assigned by the LabVIEW program.
	PWMStartRead	A cluster containing HighPeriod and Low Period.
	error out	A cluster containing error information. If ***error in*** indicated an error, the **status**, **code**, and **source** elements of ***error out*** have the same values as the corresponding elements of ***error in***

Appendix A Card Gain Codes

This appendix gives the card gains which correspond to the gain codes in the driver.

PCL-711B gain code table

Input range	Recommended gain	Gain code
±5 V	xl	0
±2.5 V	x2	1
±1.25 V	x4	2
±0.625 V	x8	3
±0.3125 V	x16	4

PCL-812PG gain code table (JP9 set to ±5 V)

Input range	Recommended gain	Gain code
±5 V	xl	0
±2.5 V	x2	1
±1.25 V	x4	2
±0.625 V	x8	3
±0.3125 V	x16	4

PCL-812PG gain code table (JP9 set to ±10V)

Input range	Recommended gain	Gain code
±10 V	xl	0
±5 V	x2	1
±2.5 V	x4	2
±1.25 V	x8	3
±0.625 V	x16	4

PCL-813 gain code table

Input range	Recommended gain	Gain code
±5 V	x1	0
±2.5 V	x2	1
±1.25 V	x4	2
±0.625 V	x8	3

PCL-818 gain code table

Input range	Recommended gain	Gain code
±10V	x0.5	8
±5 V	xl	0
±2.5 V	x2	1
±1.0 V	x5	2
±0.5 V	x10	3
0 to 10V	x1	4
0 to 5 V	x2	5
0 to 2 V	x5	6
0 to 1 V	x10	7

PCL-818H gain code table

Input range	Recommended gain	Gain code
±10V	x0.5	8
±5 V	xl	0
±2.5 V	x2	1
±1.25 V	x4	2
±0.625 V	x8	3
0 to 10V	x1	4
0 to 5 V	x2	5
0 to 2.5 V	x4	6
0 to 1.25 V	x8	7

PCL-818HD gain code table

Input range	Recommended gain	Gain code
±10V	x0.5	8
±5 V	xl	0
±2.5 V	x2	1
±1.25 V	x4	2
±0.625 V	x8	3
0 to 10V	x1	4
0 to 5 V	x2	5
0 to 2.5 V	x4	6
0 to 1.25 V	x8	7

PCL-818L gain code table (JP set to ±5V)

Input range	Recommended gain	Gain code
±5 V	xl	0
±2.5 V	x2	1
±1.25 V	x4	2
±0.625 V	x8	3

PCL-818L gain code table (JP set to ±10V)

Input range	Recommended gain	Gain code
±10V	x1	0
±5 V	x2	1
±2.5 V	x4	2
±1.25 V	x8	3

PCL-818HG gain code table

Input range	Recommended gain	Gain code
±5 V	xl	0
±0.5 V	x10	1
±0.05 V	x100	2
±0.005 V	x1000	3
0 to 10V	x1	4
0 to 1 V	x10	5
0 to 0.1 V	x100	6
0 to 0.01 V	x1000	7
±10V	x0.5	8
±1 V	x5	9
±0.1 V	x50	10
±0.01 V	x500	11

PCL-816 gain code table

Input range	Recommended gain	Gain code
±10V	x1	0
±5 V	x2	1
±2.5 V	x4	2
±1.25 V	x8	3
0 to 10V	x1	4
0 to 5 V	x2	5
0 to 2.5 V	x4	6
0 to 1.25 V	x8	7

PCL-1800 gain code table

Input range	Recommended gain	Gain code
±10V	x0.5	8
±5 V	xl	0
±2.5 V	x2	1
±1.25 V	x4	2
±0.625 V	x8	3
0 to 10V	x1	4
0 to 5 V	x2	5
0 to 2.5 V	x4	6
0 to 1.25 V	x8	7

MIC-2718 gain code table

Input range	Recommended gain	Gain code
±5 V	xl	0
±0.5 V	x10	1
±0.05 V	x100	2
±0.005 V	x1000	3
0 to 10V	x1	4
0 to 1 V	x10	5
0 to 0.1 V	x100	6
0 to 0.01 V	x1000	7
±10V	x0.5	8
±1 V	x5	9
±0.1 V	x50	10
±0.01 V	x500	11

PCM-3718 gain code table		
Input range	**Recommended gain**	**Gain code**
±10V	x0.5	8
±5 V	xl	0
±2.5 V	x2	1
±1.25 V	x4	2
±0.625 V	x8	3
0 to 10V	x1	4
0 to 5 V	x2	5
0 to 2.5 V	x4	6
0 to 1.25 V	x8	7

Appendix B Runtime Error Code Listings

The following is a list of possible errors and warnings that you may encounter during Runtime. These error messages can aid tremendously in troubleshooting various hardware problems when using the LabVIEW driver.

Runtime Error/Warning Codes are as follows:

A summary of the status codes is listed in Table B-1.

Table B-1. Status Code Summary

Error Code	Error ID	Description (Error Message)
1	MemoryAllocateFailed (*)	Not Enough Memory
2	ConfigDataLost (*)	Configuration Data Lost
3	InvalidDeviceHandle (*)	Invalid Device Handle
4	AIConversionFailed	Analog Input Failure On I/O=%XH
5	AIScaleFailed	Invalid Scaled Value On I/O=%XH
6	SectionNotSupported	Section Not Supported On I/O=%XH
7	InvalidChannel	Invalid Channel On I/O=%XH
8	InvalidGain	Invalid Gain Code On I/O=%XH
9	DataNotReady	Data Not Ready On I/O=%XH
10	InvalidInputParam	Invalid Input Parameter On I/O=%XH
11	NoExpansionBoardConfig	No Expansion Board Configuration in Registry/Configuration File On I/O=%XH
12	InvalidAnalogOutValue	Invalid Analog Output Value On I/O=%XH
13	ConfigIoPortFailed	Configure DIO Port Failure On I/O=%XH
14	CommOpenFailed	Open COM %d Failure
15	CommTransmitFailed	Unable to Transmit to COM %d Address %XH
16	CommReadFailed	Unable to Receive from COM %d Address %XH
17	CommReceiveFailed	Invalid Data Received from COM %d Address %XH
18	CommConfigFailed	Configure Communication Port Falied on COM %d
19	CommChecksumError	Checksum Error from COM %d Address %XH
20	InitError	Initialization Failure On I/O=%XH
21	DMABufAllocFailed (*)	No Buffer Allocated for DMA
22	IllegalSpeed	The Sample Rate Exceeds the Upper Limit On I/O=%XH
23	ChanConflict	Background Operation Is Still Running On I/O=%XH
24	BoardIDNotSupported	Board ID Is Not Supported On I/O=%XH
25	FreqMeasurementFailed	Time Interval For Frequency Measurement Is Too Small On I/O=%XH
26	CreateFileFailed (*)	Call CreateFile() Failed
27	FunctionNotSupported (*)	Function Not Supported
28	LoadLibraryFailed (*)	Load Library Failed
29	GetProcAddressFailed (*)	Call GetProcAddress() Failed
30	InvalidDriverHandle (*)	Invalid Driver Handle
31	InvalidModuleType	Module Type Not Existence On I/O=%XH
32	InvalidInputRange	The Value is Out of Range On I/O=%XH
33	InvalidWindowsHandle	Invalid Windows Handle of Destination on I/O=%XH
34	InvalidCountNumber	Invalid Numver of Conversion On I/O=%XH
35	InvalidInterruptCount	Invalid Number of Interrupt Count On I/O=%XH

36	InvalidEventCount	Invalid Number of Event Count On I/O=%XH
37	OpenEventFailed	Create or Open Event Failed On I/O=%XH
38	InterruptProcessFailed	Interrupt Process Failed On I/O=%XH
39	InvalidDOSetting	Invalid digital output direction setting COM %d Address %XH
40	InvalidEventType	Invalid Event Type On I/O=%XH
41	EventTimeOut	The Time-out Interval Elapsed in Milliseconds Parameter On I/O=%XH

Note: * means that the status code only includes error code.

Error Code	Error ID	Description (Error Message)
100	KeInvalidHandleValue	An error occured while starting the device
101	KeFileNotFound	The device has not been created
102	KeInvalidHandle	The handle passed to the function is not a valid
103	KeTooManyCmds	The logic commands have created an apparent endless loop
104	KeInvalidParameter	Passed to the driver contains an invalid parameter
105	KeNoAccess	Attempts to access a port which has not been defined in DEVINST
106	KeUnsuccessful	The operation was not successful
107	KeConInterruptFailure	The driver connects interrupt failure on I/O=%XH
108	KeCreateNoteFailure	The driver creates notification event failure On I/O=%XH
109	KeInsufficientResources	The system resource is insufficient OnI/O=%XH
110	KeHalGetAdapterFailure	An adapter object could not be created On I/O=%XH
111	KeOpenEventFailure	The driver opens notification event failure On I/O=%XH
112	KeAllocCommBufFailure	Allocate DMA buffer failure On I/O=%XH
113	KeAllocMdlFailure	Allocate MDL for DMA buffer failure On I/O=%XH
114	KeBufferSizeTooSmall	The buffer of requisition must be bigger that PAGE_SIZE On I/O=%XH

Error Code	Error ID	Description (Error Message)
201	DNInitFailed	DeviceNet Initialization Failed
202	DNSendMsgFailed	Send Message Failed On Port %d MACID %XH
203	DNRunOutOfMsgID	Run Out of Message ID
204	DNInvalidInputParam	Invalid Input Parameters
205	DNErrorResponse	Error Response On Port %d MACID %XH
206	DNNoResponse	No Response On Port %d MACID %XH
207	DNBusyOnNetwork	Busy On Network On Port %d MACID %XH
208	DNUnknownResponse	Unknown Response On Port %d MACID %XH
209	DNNotEnoughBuffer	Message Length Is Too Long on Port %d MACID %XH
210	DNFragResponseError	Fragment Response Error On Port %d MACID %XH
211	DNTooMuchDataAck	Too Much Data Acknowledge On Port %d MACID %XH
212	DNFragRequestError	Fragment Request Error On Port %d MACID %XH
213	DNEnableEventError	Event Enable/Disable Error On Port %d MACID %XH
214	DNCreateOrOpenEventError	Device Net Driver Create/Open Event Failed On Port %d MACID %XH
215	DNIORequestError	IO Message Request Error On Port %d MACID %XH
216	DNGetEventNameError	Get Event Name From CAN Driver Failed On Port %d MACID %XH
217	DNTimeOutError	Wait For Message Time Out Error On Port %d MACID %XH
218	DNOpenFailed	Open CAN Card Failed
219	DNCloseFailed	Close CAN Card Failed
220	DNResetFailed	DeviceNet Reset Failed

NOTE. To implement applications created on another system, the ADVANTEK.INI file from the initial system must be copied into the WINDOWS directory of the new system.

The ADVANTECH.INI file contains all of the driver information as configured in the initial system. For your application to function properly in another system, this file must accompany the previously-created applications.

Appendix C Reference

- Advantech 32-bit DLL Driver User's Manual for Windows NT/95 Ver. 1.xx, **MANUAL.PDF**[1]
- Advantech LabVIEW Driver User's Manual for Windows NT/95 Ver. 1.xx , **LVMANUAL.PDF**[2]
- **LabVIEW User Manual**
- **LabVIEW Code Interface Reference Manual**

[1] You can find MANUAL.PDF file in \Program Files\Advantech\Adsapi\ directory.

[2] You can find LVMANUAL.PDF in \Program Files\National Instruments\LabVIEW\ directory**.**