# PCIS-OCX

**32-bit ActiveX controls for**
**NuDAQ PCI Data Acquisition Cards**

# Programmer's Guide

Manual Rev: 2.30: November 15, 2001

# Getting Service from ADLINK

- **Customer Satisfaction is always the most important thing for ADLINK Tech Inc. If you need any help or service, please contact us and get it.**

| ADLINK Technology Inc. | | | |
|---|---|---|---|
| Web Site | http://www.adlink.com.tw<br>http://www.adlinktechnology.com | | |
| Sales & | service@adlink.com.tw | | |
| Technical | NuDAQ | nudaq@adlink.com.tw | |
| Support | NuDAM | nudam@adlink.com.tw | |
| | NuIPC | nuipc@adlink.com.tw | |
| | NuPRO | nupro@adlink.com.tw | |
| | Software | sw@adlink.com.tw | |
| | AMB | amb@adlink.com.tw | |
| TEL | +886-2-82265877 | FAX | +886-2-82265717 |
| Address | 9F, No. 166, Jian Yi Road, Chungho City, Taipei, 235 Taiwan, R.O.C. | | |

- **Please inform or FAX us of your detailed information for a prompt, satisfactory and constant service.**

| Detailed Company Information | | | |
|---|---|---|---|
| Company/Organizatio | | | |
| Contact Person | | | |
| E-mail Address | | | |
| Address | | | |
| Country | | | |
| TEL | | FAX | |
| Web Site | | | |
| **Questions** | | | |
| Product Model | | | |
| Environment to Use | ☐OS: _____<br>☐Computer Brand: _____<br>☐M/B: ☐CPU:<br>☐Chipset: ☐Bios:<br>☐Video Card:<br>☐Network Interface Card:<br>☐Other: | | |
| Challenge Description | | | |
| Suggestions for ADLINK | | | |

# Table of Contents


**NuDAQ PCI Configuration ..... 1**

Register NuDAQ cards for Windows NT ..... 2

Configure NuDAQ cards for Windows 98 or Windows 2000 ..... 3

Define device ..... 4

**ActiveX Controls Function Reference ..... 6**

Pci6208 ActiveX Control ..... 7

Properties ..... 7

Methods ..... 8

Events ..... 11

Pci6308 ActiveX Control ..... 13

Properties ..... 13

Methods ..... 16

Events ..... 19

Pci7200 ActiveX Control ..... 20

Properties ..... 20

Methods ..... 25

Events ..... 30

Pci7230 ActiveX Control ..... 32

Properties ..... 32

Methods ..... 33

Events ..... 35

Pci7233 ActiveX Control ..... 36

Properties ..... 36

Methods ..... 37

Events ..... 38

Pci7234 ActiveX Control ..... 40

Properties ..... 40

Methods ..... 40

Events ..... 41

Pci7248 ActiveX Control ..... 43

Properties ..... 43

Methods ..... 47

Events ..... 52

Pci7249 ActiveX Control ..... 53

Properties ..... 53

Methods ..... 57

Events ..... 62

Pci7250 ActiveX Control ..... 63

Properties ..... 63
Methods ..... 63
Events ..... 66
Pci7252 ActiveX Control ..... 68
Properties ..... 68
Methods ..... 68
Events ..... 71
Pci7296 ActiveX Control ..... 73
Properties ..... 73
Methods ..... 79
Events ..... 84
Pci7300 ActiveX Control ..... 86
Properties ..... 86
Methods ..... 93
Events ..... 98
Pci7396 ActiveX Control ..... 100
Properties ..... 100
Methods ..... 109
Events ..... 115
Pci7432 ActiveX Control ..... 116
Properties ..... 116
Methods ..... 117
Events ..... 119
Pci7433 ActiveX Control ..... 120
Properties ..... 120
Methods ..... 121
Events ..... 123
Pci7434 ActiveX Control ..... 124
Properties ..... 124
Methods ..... 124
Events ..... 125
Pci8554 ActiveX Control ..... 127
Properties ..... 127
Methods ..... 129
Events ..... 132
Pci9111 ActiveX Control ..... 134
Properties ..... 134
Methods ..... 139
Events ..... 143
Pci9112 ActiveX Control ..... 146

Properties ..... 146
Methods ..... 150
Events ..... 156
Pci9113 ActiveX Control ..... 158
Properties ..... 158
Methods ..... 162
Events ..... 165
Pci9114 ActiveX Control ..... 168
Properties ..... 168
Methods ..... 172
Events ..... 177
Pci9116 ActiveX Control ..... 180
Properties ..... 180
Methods ..... 191
Events ..... 197
Pci9118 ActiveX Control ..... 199
Properties ..... 199
Methods ..... 209
Events ..... 216
Pci9812 ActiveX Control ..... 218
Properties ..... 218
Methods ..... 224
Events ..... 226
Appendix Data File Format ..... 229

# How to Use This Guide

This manual is designed to help you use the NuDAQ PCI ActiveX controls to control NuDAQ PCI data acquisition cards.

The *Programmer's Guide* is organized as follows:

**Part 1**, *"NuDAQ Configuration"*, describes how you can use the NuDAQ Configuraton Utility to register NuDAQ cards on Widnows 98/NT/2000 and define local or remote devices.

**Part 2**, *"ActiveX Controls Function Reference"*, contains the detailed descriptions of each NuDAQ PCI ActiveX controls.

# NuDAQ PCI Configuration

Before you begin your NuDAQ PCI application development, you must configure your NuDAQ devices. NuDAQ PCI ActiveX controls need the device configuration information to program your hardware properly.

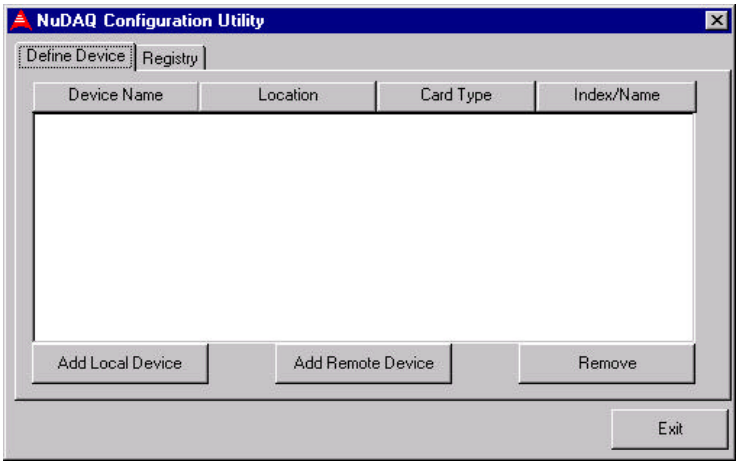
In most cases you follow the same general steps:

1. If your platform is Windows 98 or Windows 2000, you have to install DAQ hardware device when you play NuDAQ card and enter Windows. Please refer to *NuDAQ PCI and NuIPC CompactPCI DAQ cards software Installation Guide* for the detailed information.
2. Configure your device using the *NuDAQ Configuration Utility*.
3. Define your device using the *NuDAQ Configuration Utility*.
4. For the information of ActiveX programming, please refer to *DAQBench User's Guide*. You can find the manual on ADLINK All in One CD.

Using the *NuDAQ Configuration Utility*, you can:

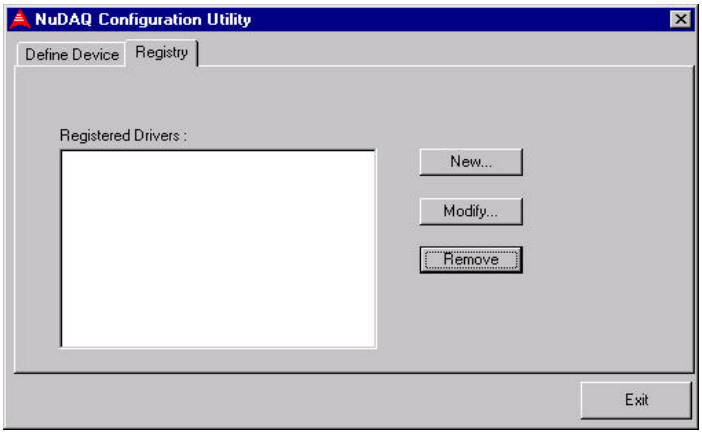
1. Registry NuDAQ device drivers to Windows in your system (NT only).
2. Configure the Continuous AI/AO/DI/DO buffers of NuDAQ cards.
3. Define NuDAQ devices that may be local or remote device in your system.
4. Save the NuDAQ device configuration to the configuration file.

The utility, NuDAQCfg.EXE, is installed in your PCIS-OCX\PCIDAQ directory.

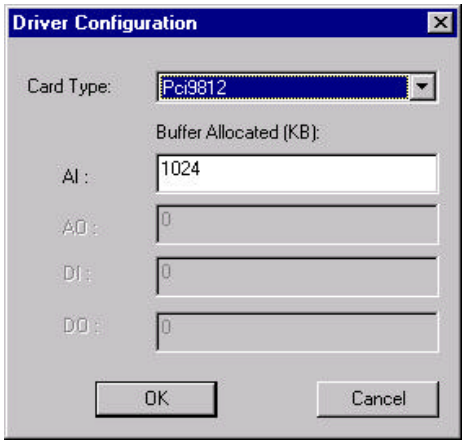



# Register NuDAQ cards for Windows NT

The NuDAQ devices must be registered at Window Registry before the NuDAQ applications are run. You can use *NuDAQ Configuration Utility* to do the registry of NuDAQ cards. On *NuDAQ Configuration Utility* window, Select "**Registry**" panel and view the window as below.

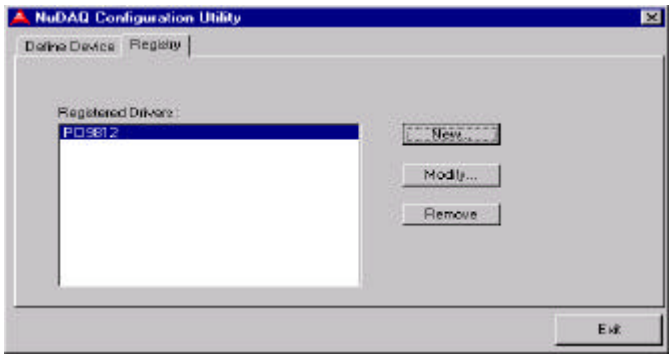


This Panel is used for users to ***make the registry*** of local NuDAQ PCI device drivers, ***remove*** installed drivers and ***modify*** the allocated buffer sizes of AI, AO, DI and DO. Click "**New**" or "**Modify**" button and popup a Driver Configuration dialog for specifying the allocated buffers as below.




The allocated buffer sizes of AI, AO, DI, DO represent the sizes of contiguous Initially Allocated memory for continuous analog input, analog output, digital input, digital output respectively. Its unit is *KB*, i.e. 1024 bytes. Device driver will try to allocate these sizes of memory at system startup time. The size of initially allocated memory is the maximum memory size that continuous AI/AO/DI/DO can be performed on this type of cards plugged on this local machine. It will induce an unexpected result in that the data size of continuous operation exceeds the initially allocated size.

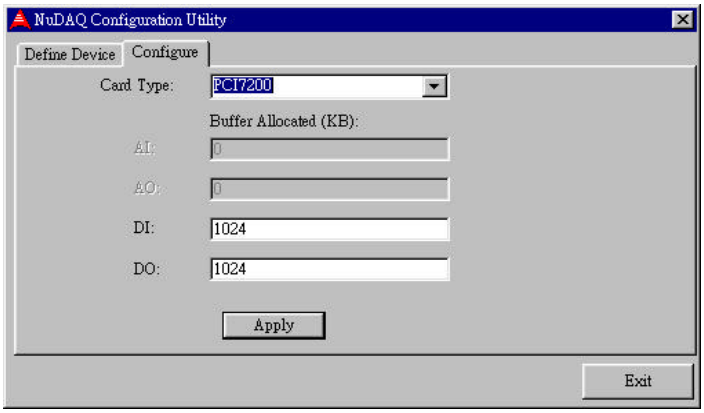
After the device configurations of the driver you select is finished, click "OK" to register the driver and return to the *NuDAQCfg* main window. The driver you just registered will be shown on the registered driver list as the following figure:




Then you can Click "Exit" button to exit the driver registry utility. To make the registered drivers work, you have to restart Windows NT system.

# Configure NuDAQ cards for Windows 98 or Windows 2000

Windows 98/2000 and NuDAQ PCI cards work very well together because Windows 98/2000 includes Plug and Play capabilities and standard drivers for PCI card devices. On Windows 98/2000, NuDAQ cards don't need to do registry work but they must allocate memory buffer for continuous operation. You can use NuDAQ Configuration Utility to specify the size of contiguous Initially Allocated Memory for analog input, analog output, digital input and digital output. On NuDAQ Configuration Utility window, Select "**Configure**" panel and view the window as below.

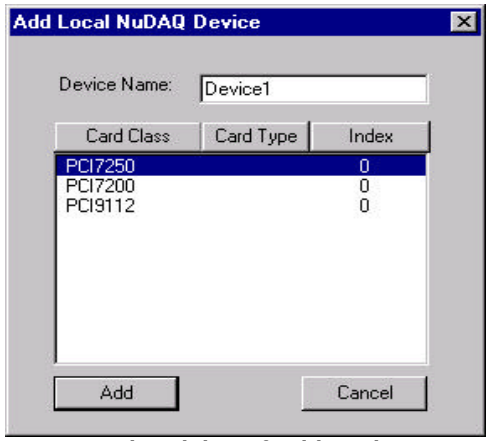



The allocated buffer sizes of AI, AO, DI, DO represent the sizes of contiguous Initially Allocated memory for continuous analog input, analog output, digital input, digital output respectively. Its unit is *KB*, i.e. 1024 bytes. Device driver will try to allocate these sizes of memory at system startup time. The size of initially allocated memory is the maximum memory size that continuous AI/AO/DI/DO can be performed on this type of cards plugged on this local machine. It will induce an unexpected result in that the data size of continuous operation exceeds the initially allocated size.

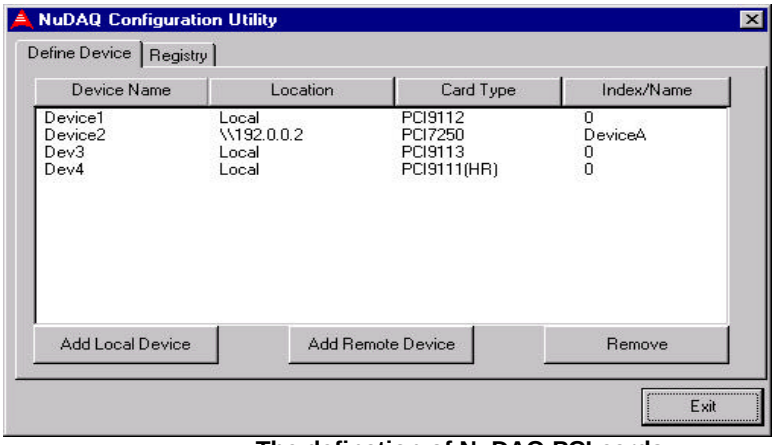
After the device configurations of the driver you select is finished, click “Apply” to register the driver.

# Define device

You can click **Add Local Device** button and will popup one dialog box. In the list box of Add Local NuDAQ Device dialog, you can find some ADLINK NuDAQ cards that are currently installed on this machine. Then, you can select one card and enter its device name. Click **Add** button and this local NuDAQ card is defined and added in the list box of Define Device Panel.




**The Dialog of Add Device**




**The defination of NuDAQ PCI cards**

# ActiveX Controls Function Reference

# Pci6208 ActiveX Control

The PCI-6208/16 is a PCI-bus analog output card. The Pci6208 ActiveX control is an software component that provides the interface for user to control PCI6208V / 6216V / 6208A cards.

# Properties

## *DeviceName Property*

Returns/sets a string that indicates the device name of a PCI-6208 card that was defined in NuDAQCfg Utility.

**Syntax**

*object***.DeviceName** [= string]

**Remarks**

The device name of NuDAQ cards must be defined in NuDAQCfg Utility. The indicated NuDAQ card by device name may be a local device or a remote device on remote machine. The device name of PCI6208 ActiveX Control must be specified to a PCI-6208 card defined in NuDAQCfg Utility.

**Data Type**

String

## *CardType Property*

Returns/sets a value that indicates the card type for programming Pci6208 ActiveX control.

**Syntax**

*object***.CardType** [= number]

**Settings**

| Number | Card Type | Analog output channel |
|---|---|---|
| 6 | PCI-6208V | 0 ~ 7 |
| 7 | PCI-6216V | 0 ~ 15 |
| 8 | PCI-6208A | 0 ~ 7 |

**Remarks**

This property will be automatically set value when the device name be specified and the device information be retrieved successfully.

**Data Type**

Integer

## *VoltageCurrentMode Property*

Returns/sets a value that indicates the output current for programming Pci6208 ActiveX control.

### Syntax

*object*.**VoltageCurrentMode** [= number]

### Settings

| Number | Output Current |
|---|---|
| 0 | 0 ~ 20 mA |
| 1 | 5 ~ 25 mA |
| 2 | 4 ~ 20 mA |

### Remarks

This will be valid only when the card type is PCI-6208A.

### Data Type

Integer

## *OpenMode Property*

Returns/sets a value that determines the mode of opening device .

### Syntax

*object*.**OpenMode** [= number]

### Settings

| Value | Description |
|---|---|
| 0 | Automatically open device when the control was created |
| 1 | Don't open device when the control was created. Must call the Open method to open device. (Manual) |

### Data Type

Integer.

# Methods

## *Open Method*

### Syntax

Function *object*.**Open (***[ErrMsgBox As Variant]***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*[ErrMsgBox As Variant]*

It is optional and boolean type. Default value is False

True, It will popup error message dialog box when the opening device is failed.

False, It will fire DAQError event instead of popping up dialog when the opening device is failed.

**Remarks**

This method will be use when the OpenMode property is Manual.

**Note**

In VC++, *ErrMsgBox* is a VARIANT of VT_I2**.**

# *ReadDILine Method*

**Syntax**

Function *object***.ReadDILine (***line As Integer, value As Variant***)** As Boolean

**Return Value**

True if the function is successful; otherwise False.

**Arguments**

*line As Integer*

Selects one line number from the indicated port: from 0 to 3(4-bit port).

*value As Variant*

Returns the data input from PCI-6208 card.

**Remarks**

Users can read data from the indicated digital input line of PCI-6208 card.

**Note**

In VC++, *value* is a VARIANT of VT_UI1

# *ReadDIPort Method*

**Syntax**

Function **ReadDIPort** (*value As Variant*) As Boolean

**Return Value**

True if the function is successful; otherwise False.

**Arguments**

*value As Variant*

The digital data read from the digital input port.

**Remarks**

You can read data from the digital input port.

**Note**

In VC++, *value* is a VARIANT of VT_I4

# *WriteDOPort Method*

### Syntax

Function *object*.**WriteDOPort (***value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*value as Variant*

4-bit data that will be written to the digital output port.

### Remarks

Users can write data to the PCI-6208 digital output port.

### Note

In VC++, *value* is a VARIANT of VT_I4.

## *WriteSingleAO Method*

### Syntax

Function object.**WriteSingleAO**(*channel as Integer, value As Variant*) As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*channel as integer*

6208V / 6208A: 0~7; 6216V: 0~15

*value As Variant*

The data will be written to analog output.

The range is 6208A: 0~10V;  6208V / 6216V: -10V~10V

### Note

In VC++, *value* is a VARIANT of VT_R8.

## *WriteDOLine Method*

### Syntax

Function *object*.**WriteDOLine(***line As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*line As Integer*

Selects one line number from the indicated port from 0 to 3.

*value As Variant*

Sets 0 or 1 to the indicated line.

### Note

In VC++, *value* is a VARIANT of VT_I4.

## ReadBackDOPort Method

Reads back data from the digital output port.

### Syntax

Function *object*.**ReadBackDOPort (***value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*value As Variant*

Data that is read back from the port.

### Note

In VC++, *value* is a VARIANT of VT_I4.

## ReadBackDOLine Method

Reads back data from the indicated digital output line of the digital output port.

### Syntax

Function *object*.**ReadBackDOLine(***line As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*line As Integer*

Selects one line number from the indicated port from 0 to 3.

*value As Variant*

Data that is read back from the indicated line.

### Note

In VC++, *value* is a VARIANT of VT_UI1.

# Events

## DAQError Event

### Syntax

sub *ControlName*_**DAQError (** *ErrString As String* **)**

### Arguments

*ErrString As String*

The string of error reasion

### Remarks

This event will occur when some error occur in control

# Pci6308 ActiveX Control

The PCI-6308 is a PCI-bus analog output card. The Pci6308 ActiveX control is an software component that provides the interface for user to control PCI6308V / 6308A cards.

# Properties

## *DeviceName Property*

Returns/sets a string that indicates the device name of a PCI-6308 card that was defined in NuDAQCfg Utility.

**Syntax**

*object***.DeviceName** [= string]

**Remarks**

The device name of NuDAQ cards must be defined in NuDAQCfg Utility. The indicated NuDAQ card by device name may be a local device or a remote device on remote machine. The device name of PCI6308 ActiveX Control must be specified to a PCI-6308 card defined in NuDAQCfg Utility.

**Data Type**

String

## *CardType Property*

Returns/sets a value that indicates the card type for programming Pci6308 ActiveX control.

**Syntax**

*object***.CardType** [= number]

**Settings**

| Number | Card Type | Analog output channel |
|---|---|---|
| 6 | PCI-6308V | 0 ~ 7 |
| 8 | PCI-6308A | 0 ~ 7 |

**Remarks**

This property will be automatically set value when the device name be specified and the device information be retrieved successfully.

**Data Type**

Integer.

## *CH03Polarity Property*

Returns/sets a value that indicates the voltage polarity of channel 0~3 in Pci6308 ActiveX control.

**Syntax**

*object*.**CH03Polarity** [= number]

**Settings**

| Number | Polarity |
|---|---|
| 0 | Unipolar |
| 1 | Bipolar |

**Remarks**

This property is available on PCI-6308V.

**Data Type**

Integer

## *CH47Polarity Property*

Returns/sets a value that indicates the voltage polarity of channel 4~7 in Pci6308 ActiveX control.

**Syntax**

*object*.**CH47Polarity** [= number]

**Settings**

| Number | Polarity |
|---|---|
| 0 | Unipolar |
| 1 | Bipolar |

**Remarks**

This property is available on PCI-6308V.

**Data Type**

Integer

## *CH03RefVoltage Property*

Returns/sets a value that indicates the reference voltage of channel 0~3 in Pci6308 ActiveX control.

**Syntax**

*object*.**CH03RefVoltage** [= double]

**Settings**

The range is 0 ~ 10V

**Remarks**

This property is available when the CH03Polarity property is unipolar on PCI-6308V.

**Data Type**

Double

## *CH47RefVoltage Property*

Returns/sets a value that indicates the reference voltage of channel 4~7 in Pci6308 ActiveX control.

**Syntax**

*object*.**CH47RefVoltage** [= double]

**Settings**

The range is 0 ~ 10V

**Remarks**

This property is available when the CH47Polarity property is unipolar on PCI-6308V.

**Data Type**

Integer

## *OpenMode Property*

Returns/sets a value that determines the mode of opening device .

**Syntax**

*object*.**OpenMode** [= number]

**Settings**

| Value | Description |
|---|---|
| 0 | Automatically open device when the control was created |
| 1 | Don't open device when the control was created. Must call the Open method to open device. (Manual) |

**Data Type**

Integer.

## *VoltageCurrentMode Property*

Returns/sets a value that indicates the output current for programming PCI6308 ActiveX control.

**Syntax**

*object*.**VoltageCurrentMode** [= number]

**Settings**

| Number | Output Current |
|---|---|
| 0 | 0 ~ 20 mA |
| 1 | 5 ~ 25 mA |
| 2 | 4 ~ 20 mA |

**Remarks**

This will be valid only when the card type is PCI-6308A.

**Data Type**

Integer

# Methods

## *Open Method*

### Syntax

Function *object*.**Open (***[ErrMsgBox As Variant]***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*[ErrMsgBox As Variant]*

It is optional and boolean type. Default value is False

True, It will popup error message dialog box when the opening device is failed.

False, It will fire DAQError event instead of popping up dialog when the opening device is failed.

### Remarks

This method will be use when the OpenMode property is Manual.

### Note

In VC++, *ErrMsgBox* is a VARIANT of VT_I2**.**

## *ReadDILine Method*

### Syntax

Function *object*.**ReadDILine (***line As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*line As Integer*

Selects one line number from the indicated port: from 0 to 3(4-bit port).

*value As Variant*

Returns the data input from PCI-6308 card.

### Remarks

Users can read data from the indicated digital input line of PCI-6308 card.

### Note

In VC++, *value* is a VARIANT of VT_UI1

## *ReadDIPort Method*

### Syntax

Function **ReadDIPort** (*value As Variant*) As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*value As Variant*

The digital data read from the digital input port.

### Remarks

You can read data from the digital input port.

### Note

In VC++, *value* is a VARIANT of VT_I4

## *WriteDOPort Method*

### Syntax

Function *object*.**WriteDOPort (***value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*value as Variant*

4-bit data that will be written to the digital output port.

### Remarks

Users can write data to the PCI-6308 digital output port.

### Note

In VC++, *value* is a VARIANT of VT_I4.

## *WriteSingleAO Method*

### Syntax

Function object.**WriteSingleAO**(*channel as Integer, value As Variant*) As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*channel as integer*

6308V / 6308A: 0~7.

*value As Variant*

The data will be written to analog output.

The range is 6308A: 0~10V;  6308V: -10V~10V

## *WriteDOLine Method*

### Syntax

Function *object.***WriteDOLine(***line As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*line As Integer*

Selects one line number from the indicated port from 0 to 3.

*value As Variant*

Sets 0 or 1 to the indicated line.

### Note

In VC++, *value* is a VARIANT of VT_I4.

## *ReadBackDOPort Method*

Reads back data from the digital output port.

### Syntax

Function *object.***ReadBackDOPort (***value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*value As Variant*

Data that is read back from the port.

### Note

In VC++, *value* is a VARIANT of VT_I4.

## *ReadBackDOLine Method*

Reads back data from the indicated digital output line of the digital output port.

### Syntax

Function *object.***ReadBackDOLine(***line As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*line As Integer*

Selects one line number from the indicated port from 0 to 3.

*value As Variant*

Data that is read back from the indicated line.

### Note

In VC++, *value* is a VARIANT of VT_UI1.

# Events

## *DAQError Event*

### Syntax

sub *ControlName*_**DAQError (** *ErrString As String* **)**

### Arguments

*ErrString As String*

The string of error reasion

### Remarks

This event will occur when some error occur in control

# Pci7200 ActiveX Control

The PCI-7200 is a PCI-bus high speed DI/O card. The Pci7200 ActiveX control is a software component that provides the interface for users to control PCI-7200 card.

# Properties

## *DeviceName Property*

Returns/sets a string that indicates the device name of a PCI-7200 card that is defined in NuDAQCfg Utility.

**Syntax**

*object*.**DeviceName** [= string]

**Remarks**

The device name of NuDAQ cards must be defined in NuDAQCfg Utility. The indicated NuDAQ card by device name may be a local device or a remote device on remote machine. The device name of PCI7200 ActiveX Control must be specified to a PCI-7200 card defined in NuDAQCfg Utility.

**Data Type**

String.

## *DIClockSource Property*

Returns/sets a value that determines the clock source of DI.

**Syntax**

*object*.**DIClockSource** [= number]

**Settings**

| Value | Description |
|---|---|
| 0 | Internal Trigger |
| 1 | External Trigger |
| 2 | Handshaking |

**Data Type**

Integer.

## *DIDoubleBufferMode Property*

Sets a value that determines whether it is double-buffer mode or not.

**Syntax**

*object*.**DIDoublebufferMode** [= boolean]

**Settings**

| Value | Description |
|---|---|
| True | During DMA Mode, the buffer will be divided into two parts. When either part is full of data, it will get DIHalfReady event. |
| False | The buffer is single, when it is full of data, it will get DIComplete event. |

**Data Type**

Boolean

## DINPatterns Property

Sets a value that indicates the total number of patterns of DMA digital input or circular buffer size.

**Syntax**

*object*.**DINPatterns** [= number]

**Remarks**

If double buffer mode is disabled, this value is the number of DMA digital input to be performed. If Double-buffer Mode value is True, this is the size (in patterns) of the circular buffer and its value must be a multiple of 2.

**Data Type**

Long.

## DIREQPolarity Property

Returns/sets a value that determines the I_REQ signal polarity.

**Syntax**

*object*.**DIREQPolarity** [= number]

**Settings**

| Value | Description |
|---|---|
| 0 | I_REQ is falling edge active. |
| 1 | I_REQ is rising edge active. |

**Data Type**

Integer.

## DIScanRate Property

Returns/sets a value that determines the patterns per second of DMA digital input.

**Syntax**

*object*.**DIScanRate** [= number]

**Settings**

The range of DIScanRate is between 0 and 2MHz. This property is used only when the DIClockSource

property is set to Internal Trigger.

### Data Type

Long.

## DITriggerPolarity Property

Returns/sets a value that determines the I_TRG trigger polarity.

### Syntax

*object*.**DITriggerPolarity** [= number]

### Settings

| Value | Description |
|---|---|
| 0 | Falling edge trigger. |
| 1 | Rising edge trigger. |

### Data Type

Integer.

## DITriggerWaiting Property

Returns/sets a value that determines the waiting status of the trigger of DI operation.

### Syntax

*object*.**DITriggerWaiting** [= Boolean]

### Settings

| Value | Description |
|---|---|
| False | The input samples will start immediately. |
| True | The input samples will wait rising or faling edge trigger I_TRG to start digital input. |

### Data Type

Boolean.

## DOClockSource Property

Returns/sets a value that determines the clock source of DO.

### Syntax

*object*.**DOClockSource** [= number]

### Settings

| Value | Description |
|---|---|
| 0 | Internal |
| 1 | Handshaking |

### Data Type

Integer.

## DOIterations Property

Returns/sets a value that indicates the iteration of data output.

### Syntax

*object*.**DOIterations** [= number]

### Remarks

The value range is 0 ~ 65535.

Iteration property means the number of times the data in buffer to output to channel in continuous DMA digital output operation. A value of 0 means that digital output operation proceeds indefinitely.

### Data Type

Integer.

## DONPatterns Property

Returns/sets a value that indicates the total number of patterns of DMA digital output.

### Syntax

*object*.**DONPatterns** [= number]

### Remarks

This value is the number of DMA digital output to be performed.

### Data Type

Long.

## DOREQEnable Property

Returns/sets a value that determines the output REQ status for the card.

### Syntax

*object*.**DOREQEnable** [= boolean]

### Settings

| Value | Description |
|---|---|
| True | O_REQ enable, an O_REQ strobe is generated after output data |
| False | O_REQ disable |

### Data Type

Boolean.

## DOScanRate Property

Returns/sets a value that determines the patterns per second of DMA digital output.

**Syntax**

*object*.**DOScanRate** [= number]

**Settings**

The range of DOScanRate is between 0 and 2MHz. This property is used only when the DOClockSource property is set to Internal.

**Data Type**

Long.

## DOTriggerSignal Property

Returns/sets a value that determines the O_TRIG signal.

**Syntax**

*object*.**DOTriggerSiginal** [= number]

**Settings**

| Value | Description |
|---|---|
| 0 | O_TRIG signal goes low |
| 1 | O_TRIG signal goes high |

**Data Type**

Integer.

## OpenMode Property

Returns/sets a value that determines the mode of opening device .

**Syntax**

*object*.**OpenMode** [= number]

**Settings**

| Value | Description |
|---|---|
| 0 | Automatically open device when the control was created |
| 1 | Don't open device when the control was created. You have to call the *Open* method to open device. |

**Data Type**

Integer.

## StreamToFile Property

Returns/sets a value that determines if the control is enabled the function of streaming data to disk file.

**Syntax**

*object*.**StreamToFile** [= boolean]

### Settings

| Value | Description |
|---|---|
| False | Disable the function of streaming data to disk file |
| True | Enable the function of streaming data to disk file |

### Data Type

Boolean.

# Methods

## *CheckContDI Method*

### Syntax

Function object**.CheckContDI(***AccessCount as long, stop as Integer***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*AccessCount as long*

Number of digital input data that has been transferred.

*stop as Integer*

Current state: 1 = stop, 0 = running

### Remarks

You can request DMA analog input status.

## *CheckContDO Method*

### Syntax

*Function object*.**CheckContDO(***AccessCount as long, stop as Integer***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*AccessCount as long*

Number of digital input data that has been transferred.

*stop as Integer*

Current state: 1 = stop, 0 = running

### Remarks

You can request DMA analog output status.

## *Open Method*

### Syntax

Function *object*.**Open (***[ErrMsgBox As Variant]***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*[ErrMsgBox As Variant]*

It is optional and boolean type. Default value is False

True, It will popup error message dialog box when the opening device is failed.

False, It will fire DAQError event instead of popping up dialog when the opening device is failed.

### Remarks

This method will be use when the OpenMode property is Manual.

### Note

In VC++, *ErrMsgBox* is a VARIANT of VT_I2.

## *ReadDILine Method*

### Syntax

Function *object*.**ReadDILine (***line As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*line As Integer*

Selects one line number from the indicated port: from 0 to 31 (32-bit port).

*value As Variant*

Returns the data input from PCI-7200 card.

### Remarks

Users can read data from the indicated digital input line of PCI-7200 card.

### Note

In VC++, *value* is a VARIANT of VT_UI1.

## *ReadDIPort Method*

### Syntax

Function object.**ReadDIPort** (*value As Variant*) As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*value As Variant*

Sets one value buffer for retrieving 32-bit input data from PCI-7200 card.

### Remarks

Users can read 32-bit digital input data from PCI-7200 card.

### Note

In VC++, *value* is a VARIANT of VT_I4.

## StartContDI Method

### Syntax

Function *object*.**StartContDI (***[FileName as Variant]***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*[FileName as String]*

It is optional and String type. Default value is NULL. FileName specified the file name of streaming data to disk.

### Remarks

You can use this method to start the DMA digital input function of PCI-7200 card. If the StreamToFile property is True then the DMA data will be write the file specified by FileName. Otherwise, the FileName parameter will be ignored.

The data file is written in binary format. DAQBench provides a convenient tool DAQCvt to convert the binary file to the file format read easily. See *DAQBench User's Guide* for the usage of the utility. If you want to handle the data by yourself, please refer to Appendix *Data File Format* for the file structure.

### Note

In VC++, FileName is a VARIANT of VT_BSTR.

## StartContDO Method

### Syntax

Function *object*.**StartContDO(***Buffer as Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*Buffer as Variant*

A buffer data or a array of buffer data, data type can be byte, integer, long, float, double

### Remarks

You can use this method to start the DMA digital output function of PCI-7200 card.

### Note

In VC++, Buffer is a VARIANT of VT_ARRAY | VT_I2, VT_ARRAY | VT_I4, VT_ARRAY | VT_R4, VT_ARRAY | VT_R8, VT_ARRAY | VT_UI1.

## *StopContDI Method*

### Syntax

Function *object*.**StopContDI ()** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

None

### Remarks

You can use this method to stop DMA digital input.

## *StopContDO Method*

### Syntax

Function *object*.**StopContDO ()** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

None

### Remarks

You can use this method to stop DMA digital output.

## *WriteDOPort Method*

### Syntax

Function *object*.**WriteDOPort (***value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*value as Variant*

32-bit data that will be written to the digital output port.

### Remarks

Users can write data to the PCI-7200 digital output port.

**Note**

In VC++, *value* is a VARIANT of VT_I4.

## *WriteDOLine Method*

**Syntax**

Function *object*.**WriteDOLine(***line As Integer, value As Variant***)** As Boolean

**Return Value**

True if the function is successful; otherwise False.

**Arguments**

*line As Integer*

Selects one line number from the indicated port from 0 to 31.

*value As Variant*

Sets 0 or 1 to the indicated line.

**Note**

In VC++, *value* is a VARIANT of VT_I4.

## *ReadBackDOPort Method*

Reads back data from the indicated digital output port.

**Syntax**

Function *object*.**ReadBackDOPort (***value As Variant***)** As Boolean

**Return Value**

True if the function is successful; otherwise False.

**Arguments**

*value As Variant*

Data that is read back from the port.

**Note**

In VC++, *value* is a VARIANT of VT_I4.

## *ReadBackDOLine Method*

Reads back data from the indicated digital output line of the digital output port.

**Syntax**

Function *object*.**ReadBackDOLine(***line As Integer, value As Variant***)** As Boolean

**Return Value**

True if the function is successful; otherwise False.

### Arguments

*line As Integer*

Selects one line number from the indicated port from 0 to 31.

*value As Variant*

Data that is read back from the indicated line.

### Note

In VC++, *value* is a VARIANT of VT_UI1.

# Events

## *DiComplete Event*

### Syntax

sub *ControlName*_**DiComplete (** *WaveForm As Variant* **)**

### Arguments

*WaveForm As Variant*

The data that retrieved from DMA buffer

### Remarks

This event will occur when digital input DMA function have completed

### Note

In VC++, *WaveForm* is a VARIANT of VT_ARRAY | VT_I4.

## *DiHalfReady Event*

### Syntax

sub *ControlName*_**DoHalfReady (** *WaveForm As Variant* **)**

### Arguments

*WaveForm As Variant*

The data that retrieved from one half-buffer of the circular buffer.

### Remarks

This event will occur when one half-buffer is full at double-buffered digital input DMA function.

### Note

In VC++, *WaveForm* is a VARIANT of VT_ARRAY | VT_I4.

## *DoComplete Event*

### Syntax

sub *ControlName*_**DoComplete ()**

### Arguments

None

### Remarks

This event will occur when digital output DMA function have completed

## *DAQError Event*

### Syntax

sub *ControlName*_**DAQError (** *ErrString As String* **)**

### Arguments

*ErrString As String*

The string of error reason

### Remarks

This event will occur when some error occur in control

# Pci7230 ActiveX Control

The PCI-7230 is a PCI-bus digital I/O card. The Pci7230 ActiveX control is a software component that provides the interface for user to control PCI-7230 card.

# Properties

## *DeviceName Property*

Returns/sets a string that indicates the device name of a PCI-7230 card that is defined in NuDAQCfg Utility.

**Syntax**

*object***.DeviceName** [= string]

**Remarks**

The device name of NuDAQ cards must be defined in NuDAQCfg Utility. The indicated NuDAQ card by device name may be a local device or a remote device on remote machine. The device name of PCI7230 ActiveX Control must be specified to a PCI-7230 card defined in NuDAQCfg Utility.

**Data Type**

String.

## *INT1Mode Property*

Returns/sets a value that determines the interrupt mode of INT1.

**Syntax**

*object***.INT1Mode** [= Mode]

**Settings**

| Value | Description |
|---|---|
| 0 | Disable |
| 1 | Enable |

**Remarks**

Enable**:** event occurred on the rising edge of digital input channel 0

**Data Type**

Integer.

## *INT2Mode Property*

Returns/sets a value that determines the interrupt mode of INT2.

**Syntax**

*object*.**INT2Mode** [= Mode]

**Settings**

| Value | Description |
|---|---|
| 0 | Disable |
| 1 | Enable |

**Remarks**

Enable**:** event occurred on the rising edge of digital input channel 1

**Data Type**

Integer.

## *OpenMode Property*

Returns/sets a value that determines the mode of opening device .

**Syntax**

*object*.**OpenMode** [= number]

**Settings**

| Value | Description |
|---|---|
| 0 | Automatically open device when the control was created |
| 1 | Don't open device when the control was created. Must call the Open method to open device. (Manual) |

**Data Type**

Integer.

# Methods

## *Open Method*

**Syntax**

Function *object*.**Open (***[ErrMsgBox As Variant]***)** As Boolean

**Return Value**

True if the function is successful; otherwise False.

**Arguments**

*[ErrMsgBox As Variant]*

It is optional and boolean type. Default value is False

True: It will popup error message dialog box when the opening device is failed.

False: It will fire DAQError event instead of popping up dialog when the opening device is failed.

**Remarks**

This method will be use when the OpenMode property is Manual.

**Note**

In VC++, ErrMsgBox is a VARIANT of VT_I2.

## *ReadDIPort Method*

**Syntax**

*Function object*.**ReadDIPort (***value as Variant***)** As Boolean

**Return Value**

True if the function is successful; otherwise False.

**Arguments**

*value As Variant*

Sets one value buffer for retrieving 16-bit input data from PCI-7230 card.

**Remarks**

Users can read 16-bit digital input data from PCI-7230 card.

**Note**

In VC++, *value* is a VARIANT of VT_I4.

## *ReadDILine Method*

**Syntax**

Function *object*.**ReadDILine (***line As Integer, value As Variant***)** As Boolean

**Return Value**

True if the function is successful; otherwise False.

**Arguments**

*line As Integer*

Selects one line number from the indicated port: from 0 to 15 (16-bit port).

*value As Variant*

Returns the data input from PCI-7230 card.

**Remarks**

Users can read data from the indicated digital input line of PCI-7230 card.

**Note**

In VC++, *value* is a VARIANT of VT_UI1.

## *WriteDOPort Method*

**Syntax**

Function *object*.**WriteDOPort (***value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*value as Variant*

16-bit data that will be written to the digital output port.

### Remarks

Users can write data to the PCI-7230 digital output port.

### Note

In VC++, *value* is a VARIANT of VT_I4.

# Events

## *Interrupt Event*

### Syntax

sub *ControlName*_**Interrupt (** *EvtNo as Integer* **)**

### Arguments

*EvtNo as Integer*

0: interrupt 1 occurred

1: interrupt 2 occurred

### Remarks

If an interrupt is generated by this card, this event will be generated by this ActiveX control.

## *DAQError Event*

### Syntax

sub *ControlName*_**DAQError (** *ErrString As String* **)**

### Arguments

*ErrString As String*

The string of error reason

### Remarks

This event will occur when some error occur in control

# Pci7233 ActiveX Control

The PCI-7233 is a 32-bit PCI-bus digital input card. The Pci7233 ActiveX control is a software component that provides the interface for user to control PCI-7233 card.

# Properties

## *DeviceName Property*

Returns/sets a string that indicates the device name of a PCI-7233 card that was defined in NuDAQCfg Utility.

**Syntax**

*object*.**DeviceName** [= string]

**Remarks**

The device name of NuDAQ cards must be defined in NuDAQCfg Utility. The indicated NuDAQ card by device name may be a local device or a remote device on remote machine. The device name of PCI7233 ActiveX Control must be specified to a PCI-7233 card defined in NuDAQCfg Utility.

**Data Type**

String.

## *INT1Mode Property*

Returns/sets a value that determines the interrupt mode of INT1.

**Syntax**

*object*.**INT1Mode** [= Mode]

**Settings**

| Value | Description |
|---|---|
| 0 | Disable |
| 1 | Enable |

**Remarks**

Enable**:** event occurred on the rising edge of digital input channel 0

**Data Type**

Integer.

## *INT2Mode Property*

Returns/sets a value that determines the interrupt mode of INT2.

### Syntax

*object*.**INT2Mode** [= Mode]

### Settings

| Value | Description |
|---|---|
| 0 | Disable |
| 1 | Enable |

### Remarks

Enable**:** event occurred on the rising edge of digital input channel 1

### Data Type

Integer.

## *OpenMode Property*

Returns/sets a value that determines the mode of opening device .

### Syntax

*object*.**OpenMode** [= number]

### Settings

| Value | Description |
|---|---|
| 0 | Automatically open device when the control was created |
| 1 | Don't open device when the control was created. Must call the Open method to open device. (Manual) |

### Data Type

Integer.

# Methods

## *Open Method*

### Syntax

Function *object*.**Open (***[ErrMsgBox As Variant]***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*[ErrMsgBox As Variant]*

It is optional and boolean type. Default value is False

True, It will popup error message dialog box when the opening device is failed.

False, It will fire DAQError event instead of popping up dialog when the opening device is failed.

### Remarks

This method will be used when the OpenMode property is Manual.

**Note**

In VC++, *ErrMsgBox* is a VARIANT of VT_I2.

## *ReadDIPort Method*

**Syntax**

*Function object***.ReadDIPort (***value as Variant***)** As Boolean

**Return Value**

True if the function is successful; otherwise False.

**Arguments**

*value As Variant*

Sets one value buffer for retrieving 32-bit input data from PCI-7233 card.

**Remarks**

Users can read 32-bit digital input data from PCI-7233 card.

**Note**

In VC++, *value* is a VARIANT of VT_I4

## *ReadDILine Method*

**Syntax**

Function *object***.ReadDILine (***line As Integer, value As Variant***)** As Boolean

**Return Value**

True if the function is successful; otherwise False.

**Arguments**

*line As Integer*

Selects one line number from the indicated port: from 0 to 31 (32-bit port).

*value As Variant*

Returns the data input from PCI-7233 card.

**Remarks**

Users can read data from the indicated digital input line of PCI-7233 card.

**Note**

In VC++, *value* is a VARIANT of VT_UI1

# Events

## *DAQError Event*

### Syntax

sub *ControlName*_**DAQError (** *ErrString As String* **)**

### Arguments

*ErrString As String*

The string of error reason

### Remarks

This event will occur when some error occur in control

# Pci7234 ActiveX Control

The PCI-7234 is a 32-bit PCI-bus digital output card. The Pci7234 ActiveX control is a software component that provides the interface for user to control PCI-7234 card.

# Properties

## *DeviceName Property*

Returns/sets a string that indicates the device name of a PCI-7234 card that was defined in NuDAQCfg Utility.

**Syntax**

*object***.DeviceName** [= string]

**Remarks**

The device name of NuDAQ cards must be defined in NuDAQCfg Utility. The indicated NuDAQ card by device name may be a local device or a remote device on remote machine. The device name of PCI7234 ActiveX Control must be specified to a PCI-7234 card defined in NuDAQCfg Utility.

**Data Type**

String

## *OpenMode Property*

Returns/sets a value that determines the mode of opening device .

**Syntax**

*object***.OpenMode** [= number]

**Settings**

| Value | Description |
|---|---|
| 0 | Automatically open device when the control was created |
| 1 | Don't open device when the control was created. Must call the Open method to open device. (Manual) |

**Data Type**

Integer

# Methods

## *Open Method*

### Syntax

Function *object*.**Open (***[ErrMsgBox As Variant]***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*[ErrMsgBox As Variant]*

It is optional and boolean type. Default value is False

True: It will popup error message dialog box when the opening device is failed.

False: It will fire DAQError event instead of popping up dialog when the opening device is failed.

### Remarks

This method will be used when the OpenMode property is Manual.

### Note

In VC++, *ErrMsgBox* is a VARIANT of VT_I2.

## *WriteDOPort Method*

### Syntax

Function *object*.**WriteDOPort (***value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*value As Variant*

32-bit data that will be written to the digital output port.

### Remarks

Users can write data to the PCI-7234 digital output port.

### Note

In VC++, *value* is a VARIANT of VT_I4.

# Events

## *DAQError Event*

### Syntax

sub *ControlName*_**DAQError (** *ErrString As String* **)**

### Arguments

*ErrString As String*

The string of error reason

### Remarks

This event will occur when some error occur in control

# Pci7248 ActiveX Control

The PCI-7248 is a PCI-bus digital I/O card. The Pci7248 ActiveX control is an software component that provides the interface for users to control PCI-7248 card.

# Properties

## *DeviceName Property*

Returns/sets a string that indicates the device name of a PCI-7248 card that is defined in NuDAQCfg Utility.

**Syntax**

*object*.**DeviceName** [= string]

**Remarks**

The device name of NuDAQ cards must be defined in NuDAQCfg Utility. The indicated NuDAQ card by device name may be a local device or a remote device on remote machine. The device name of PCI7248 ActiveX Control must be specified to a PCI-7248 card defined in NuDAQCfg Utility.

**Data Type**

String.

## *INT1Mode Property*

Returns/sets a value that determines the interrupt mode of INT1.

**Syntax**

*object*.**INT1Mode** [= Mode]

**Settings**

| Value | Description |
|---|---|
| 0 | Disable |
| 1 | falling edge of P1C0 |
| 2 | rising edge of P1C0 or P1C3 |
| 3 | event counter down to 0 |

**Data Type**

Integer.

## *INT2Mode Property*

Returns/sets a value that determines the interrupt mode of INT2.

**Syntax**

*object*.**INT2Mode** [= Mode]

**Settings**

| Value | Description |
|---|---|
| 0 | Disable |
| 1 | falling edge of P2C0 |
| 2 | rising edge of P2C0 or P2C3 |
| 3 | event counter down to 0 |

**Data Type**

Integer.

## *OpenMode Property*

Returns/sets a value that determines the mode of opening device .

**Syntax**

*object*.**OpenMode** [= number]

**Settings**

| Value | Description |
|---|---|
| 0 | Automatically open device when the control was created |
| 1 | Don't open device when the control was created. Must call the Open method to open device. (Manual) |

**Data Type**

Integer.

## *P1Adir Property*

Returns/sets a value that determines P1A port direction.

**Syntax**

*object*.**P1ADir** [= Direction]

**Settings**

| Value | Description |
|---|---|
| 0 | Direction: Input_Port. |
| 1 | Direction: Output_Port. |

**Data Type**

Integer.

## *P1Bdir Property*

Returns/sets a value that determines P1B port direction.

### Syntax

*object*.**P1BDir** [= Direction]

### Settings

| Value | Description |
|---|---|
| 0 | Direction: Input_Port. |
| 1 | Direction: Output_Port. |

### Data Type

Integer.

## *P1CLowerdir Property*

Returns/sets a value that determines P1C lower port direction.

### Syntax

*object*.**P1CLowerDir** [= Direction]

### Settings

| Value | Description |
|---|---|
| 0 | Direction: Input_Port. |
| 1 | Direction: Output_Port. |

### Data Type

Integer.

## *P1CUpperdir Property*

Returns/sets a value that determines P1C upper port direction.

### Syntax

*object*.**P1CUpperDir** [= Direction]

### Settings

| Value | Description |
|---|---|
| 0 | Direction: Input_Port. |
| 1 | Direction: Output_Port. |

### Data Type

Integer.

## *P2Adir Property*

Returns/sets a value that determines P2A port direction.

### Syntax

*object*.**P2ADir** [= Direction]

### Settings

| Value | Description |
|---|---|
| 0 | Direction: Input_Port |
| 1 | Direction: Output_Port |

### Data Type

Integer.

## P2Bdir Property

Returns/sets a value that determines P2B port direction.

### Syntax

*object*.**P2BDir** [= Direction]

### Settings

| Value | Description |
|---|---|
| 0 | Direction: Input_Port |
| 1 | Direction: Output_Port |

### Data Type

Integer.

## P2CLowerdir Property

Returns/sets a value that determines P2C lower port direction.

### Syntax

*object*.**P2CLowerDir** [= Direction]

### Settings

| Value | Description |
|---|---|
| 0 | Direction: Input_Port |
| 1 | Direction: Output_Port |

### Data Type

Integer.

## P2CUpperdir Property

Returns/sets a value that determines P2C upper port direction.

### Syntax

*object*.**P2CUpperDir** [= Direction]

### Settings

| Value | Description |
|---|---|
| 0 | Direction: Input_Port |
| 1 | Direction: Output_Port |

### Data Type

Integer.

# Methods

## *Open Method*

### Syntax

Function *object*.**Open (***[ErrMsgBox As Variant]***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*[ErrMsgBox As Variant]*

It is optional and boolean type. Default value is False

True: It will popup error message dialog box when the opening device is failed.

False: It will fire DAQError event instead of popping up dialog when the opening device is failed.

### Remarks

This method will be used when the OpenMode property is Manual.

### Note

In VC++, *ErrMsgBox* is a VARIANT of VT_I2.

## *ReadCounter Method*

### Syntax

Function *object*.**ReadCounter (***index As Integer, CtlValue As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*index As Integer*

Selects one counter number: 0~2.

*CtlValue As Variant*

Returns the content from the indicated counter.

### Remarks

You can read the current contents from the selected counter without disturbing the counting process.

### Note

In VC++, *CtlValue* is a VARIANT of VT_I4.

## *ReadDIPort Method*

### Syntax

*Function object*.**ReadDIPort (***port As Integer, value as Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*Port As Integer*

Selects one port for inputting digital data, number: 0~7.

| | | | |
|---|---|---|---|
| 0: P1A | 1: P1B | 2: P1CLower | 3: P1CUpper |
| 4: P2A | 5: P2B | 6: P2Clower | 7: P2CUpper |

*value As Variant*

The digital data read from the indicated digital input port.

### Remarks

You can read data from the digital input port.

### Note

In VC++, *value* is a VARIANT of VT_I2.

## *ReadDILine Method*

### Syntax

Function *object*.**ReadDILine (***port As Integer, line As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*Port As Integer*

Selects one port for inputting digital data, number: 0~7.

| | | | |
|---|---|---|---|
| 0: P1A | 1: P1B | 2: P1CLower | 3: P1CUpper |
| 4: P2A | 5: P2B | 6: P2Clower | 7: P2CUpper |

*line As Integer*

Selects one line number from the indicated port: from 0 to 7 (8-bit port).

*value As Variant*

Returns the data input from PCI-7248 card.

### Remarks

Users can read data from the indicated digital input line of PCI-7248 card.

### Note

In VC++, *value* is a VARIANT of VT_UI1.

## StartCounter Method

### Syntax

Function *object*.**StartCounter(*index As Integer, CtrMode As Integer, CtrValue As Variant, BinBcd As Intger*)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*index As Integer*

Selects one counter number: 0~2.

*CtrMode As Integer*

0: Toggle output from low to high on terminal count

1: Programmable one-shot

2: Rate generator

3: Square wave rate generator

4: Software-triggered strobe

5: Hardware-triggered strobe

*CtrValue as Variant*

Set the start value to the indicated counter.

*binbcd As Integer*

0: 16-bit binary counter, 1: 4-decade BCD counter.

### Remarks

You can start the indicated counter to operate in the specified mode.

### Note

In VC++, *CtrValue* is a VARIANT of VT_I4.

## StopCounter Method

### Syntax

Function *object*.**StopCounter(***index As Integer, State As Integer***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*index As Integer*

Selects one counter number: 0~2.

*State As Integer*

The logic state to which the counter is to be stopped, number: 0 or 1.

### Remarks

You can stop the indicated counter and set the specified state.

## *WriteDOPort Method*

### Syntax

Function *object*.**WriteDOPort (***port As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*port As Integer*

Selects one port for inputting digital data, number: 0~7.

| | | | |
|---|---|---|---|
| 0: P1A | 1: P1B | 2: P1CLower | 3: P1CUpper |
| 4: P2A | 5: P2B | 6: P2Clower | 7: P2CUpper |

*value as Variant*

8-bit data that will be written to the digital output port.

### Remarks

Users can write data to the PCI-7248 digital output port.

### Note

In VC++, *value* is a VARIANT of VT_I4.

## *WriteDOLine Method*

### Syntax

Function *object*.**WriteDOLine(***port As Integer, line As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*port As Integer*

Selects one port for outputting digital data, number: 0~7.

| | | | |
|---|---|---|---|
| 0: P1A | 1: P1B | 2: P1CLower | 3: P1CUpper |
| 4: P2A | 5: P2B | 6: P2Clower | 7: P2CUpper |

*line As Integer*

Selects one line number from the indicated port: from 0 to 7 (8-bit port).

*value As Variant*

Sets 0 or 1 to the indicated line.

### Note

In VC++, *value* is a VARIANT of VT_I4.

## ReadBackDOPort Method

Reads back data from the indicated digital output port.

### Syntax

Function *object*.**ReadBackDOPort (***port As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*port As Integer*

Selects one port for outputting digital data, number: 0~7.

| | | | |
|---|---|---|---|
| 0: P1A | 1: P1B | 2: P1CLower | 3: P1CUpper |
| 4: P2A | 5: P2B | 6: P2Clower | 7: P2CUpper |

*value As Variant*

Data that is read back from the indicated port.

### Note

In VC++, *value* is a VARIANT of VT_I4.

## ReadBackDOLine Method

Reads back data from the indicated digital output line of the indicated digital output port.

### Syntax

Function *object*.**ReadBackDOLine(***port As Integer, line As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*port As Integer*

Selects one port for outputting digital data, number: 0~7.

| | | | |
|---|---|---|---|
| 0: P1A | 1: P1B | 2: P1CLower | 3: P1CUpper |
| 4: P2A | 5: P2B | 6: P2Clower | 7: P2CUpper |

*line As Integer*

Selects one line number from the indicated port: from 0 to 7 (8-bit port).

*value As Variant*

Data that is read back from the indicated line.

### Note

In VC++, *value* is a VARIANT of VT_UI1.

# Events

## *Interrupt Event*

### Syntax

sub *ControlName*_**Interrupt (** *EvtNo as Integer* **)**

### Arguments

*EvtNo as Integer*

0: interrupt 1 occurred

1: interrupt 2 occurred

### Remarks

If an interrupt is generated by this card, this event will be generated by this ActiveX control.

## *DAQError Event*

### Syntax

sub *ControlName*_**DAQError (** *ErrString As String* **)**

### Arguments

*ErrString As String*

The string of error reason

### Remarks

This event will occur when some error occur in control

# Pci7249 ActiveX Control

The cPCI-7249 is a CompactPCI digital I/O card. The Pci7249 ActiveX control is an software component that provides the interface for users to control cPCI-7249 card.

# Properties

## *DeviceName Property*

Returns/sets a string that indicates the device name of a cPCI-7249 card that is defined in NuDAQCfg Utility.

### Syntax

*object*.**DeviceName** [= string]

### Remarks

The device name of NuDAQ cards must be defined in NuDAQCfg Utility. The indicated NuDAQ card by device name may be a local device or a remote device on remote machine. The device name of PCI7249 ActiveX Control must be specified to a cPCI-7249 card defined in NuDAQCfg Utility.

### Data Type

String

## *INT1Mode Property*

Returns/sets a value that determines the interrupt mode of INT1.

### Syntax

*object*.**INT1Mode** [= Mode]

### Settings

| Value | Description |
|---|---|
| 0 | Disable |
| 1 | falling edge of P1C0 |
| 2 | rising edge of P1C0 or P1C3 |
| 3 | event counter down to 0 |

### Data Type

Integer

## *INT2Mode Property*

Returns/sets a value that determines the interrupt mode of INT2.

**Syntax**

*object*.**INT2Mode** [= Mode]

**Settings**

| Value | Description |
|---|---|
| 0 | Disable |
| 1 | falling edge of P2C0 |
| 2 | rising edge of P2C0 or P2C3 |
| 3 | event counter down to 0 |

**Data Type**

Integer

## OpenMode Property

Returns/sets a value that determines the mode of opening device .

**Syntax**

*object*.**OpenMode** [= number]

**Settings**

| Value | Description |
|---|---|
| 0 | Automatically open device when the control was created |
| 1 | Don't open device when the control was created. Must call the Open method to open device. (Manual) |

**Data Type**

Integer

## P1Adir Property

Returns/sets a value that determines P1A port direction.

**Syntax**

*object*.**P1ADir** [= Direction]

**Settings**

| Value | Description |
|---|---|
| 0 | Direction: Input_Port. |
| 1 | Direction: Output_Port. |

**Data Type**

Integer

## P1Bdir Property

Returns/sets a value that determines P1B port direction.

### Syntax

*object*.**P1BDir** [= Direction]

### Settings

| Value | Description |
|---|---|
| 0 | Direction: Input_Port. |
| 1 | Direction: Output_Port. |

### Data Type

Integer

## P1CLowerdir Property

Returns/sets a value that determines P1C lower port direction.

### Syntax

*object*.**P1CLowerDir** [= Direction]

### Settings

| Value | Description |
|---|---|
| 0 | Direction: Input_Port. |
| 1 | Direction: Output_Port. |

### Data Type

Integer

## P1CUpperdir Property

Returns/sets a value that determines P1C upper port direction.

### Syntax

*object*.**P1CUpperDir** [= Direction]

### Settings

| Value | Description |
|---|---|
| 0 | Direction: Input_Port. |
| 1 | Direction: Output_Port. |

### Data Type

Integer

## P2Adir Property

Returns/sets a value that determines P2A port direction.

### Syntax

*object*.**P2ADir** [= Direction]

### Settings

| Value | Description |
|---|---|
| 0 | Direction: Input_Port. |
| 1 | Direction: Output_Port. |

### Data Type

Integer

## *P2Bdir Property*

Returns/sets a value that determines P2B port direction.

### Syntax

*object*.**P2BDir** [= Direction]

### Settings

| Value | Description |
|---|---|
| 0 | Direction: Input_Port. |
| 1 | Direction: Output_Port. |

### Data Type

Integer

## *P2CLowerdir Property*

Returns/sets a value that determines P2C lower port direction.

### Syntax

*object*.**P2CLowerDir** [= Direction]

### Settings

| Value | Description |
|---|---|
| 0 | Direction: Input_Port. |
| 1 | Direction: Output_Port. |

### Data Type

Integer

## *P2CUpperdir Property*

Returns/sets a value that determines P2C upper port direction.

### Syntax

*object*.**P2CUpperDir** [= Direction]

### Settings

| Value | Description |
|---|---|
| 0 | Direction: Input_Port. |
| 1 | Direction: Output_Port. |

### Data Type

Integer

# Methods

## *Open Method*

### Syntax

Function *object*.**Open (***[ErrMsgBox As Variant]***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*[ErrMsgBox As Variant]*

It is optional and boolean type. Default value is False

True: It will popup error message dialog box when the opening device is failed.

False: It will fire DAQError event instead of popping up dialog when the opening device is failed.

### Remarks

This method will be used when the OpenMode property is Manual.

### Note

In VC++, *ErrMsgBox* is a VARIANT of VT_I2.

## *ReadCounter Method*

### Syntax

Function *object*.**ReadCounter (***index As Integer, CtlValue As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*index As Integer*

Selects one counter number: 0~2.

*CtlValue As Variant*

Returns the content from the indicated counter.

### Remarks

You can read the current contents from the selected counter without disturbing the counting process.

### Note

In VC++, *CtlValue* is a VARIANT of VT_I4.

## ReadDIPort Method

### Syntax

Function *object*.**ReadDIPort (***port As Integer, value as Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*Port As Integer*

Selects one port for inputting digital data, number: 0~7.

0: P1A 1: P1B 2: P1CLower 3: P1CUpper

4: P2A 5: P2B 6: P2CLower 7: P2CUpper

*value As Variant*

The digital data read from the indicated digital input port.

### Remarks

You can read data from the digital input port.

### Note

In VC++, *value* is a VARIANT of VT_I2.

## ReadDILine Method

### Syntax

Function *object*.**ReadDILine (***port As Integer, line As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*Port As Integer*

Selects one port for inputting digital data, number: 0~7.

0: P1A 1: P1B 2: P1CLower 3: P1CUpper

4: P2A 5: P2B 6: P2CLower 7: P2CUpper

*line As Integer*

Selects one line number from the indicated port: from 0 to 7 (8-bit port).

*value As Variant*

Returns the data input from cPCI-7249 card.

### Remarks

Users can read data from the indicated digital input line of cPCI-7249 card.

### Note

In VC++, *value* is a VARIANT of VT_UI1.

## *StartCounter Method*

### Syntax

Function *object*.**StartCounter(*index As Integer, CtrMode As Integer, CtrValue As Variant, BinBcd As Intger*)**

As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*index As Integer*

Selects one counter number: 0~2.

*CtrMode As Integer*

0: Toggle output from low to high on terminal count

1: Programmable one-shot

2: Rate generator

3: Square wave rate generator

4: Software-triggered strobe

5: Hardware-triggered strobe

*CtrValue as Variant*

Set the start value to the indicated counter.

*binbcd As Integer*

0: 16-bit binary counter, 1: 4-decade BCD counter.

### Remarks

You can start the indicated counter to operate in the specified mode.

### Note

In VC++, *CtlValue* is a VARIANT of VT_I4.

## *StopCounter Method*

### Syntax

Function *object*.**StopCounter(***index As Integer, State As Integer***)** As

Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*index As Integer*

Selects one counter number: 0~2.

*State As Integer*

The logic state to which the counter is to be stopped, number: 0 or 1.

### Remarks

You can stop the indicated counter and set the specified state.

## *WriteDOPort Method*

### Syntax

Function *object*.**WriteDOPort (***port As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*port As Integer*

Selects one port for inputting digital data, number: 0~7.

0: P1A 1: P1B 2: P1CLower 3: P1CUpper

4: P2A 5: P2B 6: P2CLower 7: P2CUpper

*value as Variant*

8-bit data that will be written to the digital output port.

### Remarks

Users can write data to the cPCI-7249 digital output port.

### Note

In VC++, *value* is a VARIANT of VT_I4.

## *WriteDOLine Method*

### Syntax

Function *object.***WriteDOLine(***port As Integer, line As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*port As Integer*

Selects one port for outputting digital data, number: 0~7.

0: P1A 1: P1B 2: P1CLower 3: P1CUpper

4: P2A 5: P2B 6: P2CLower 7: P2CUpper

*line As Integer*

Selects one line number from the indicated port: from 0 to 7 (8-bit port).

*value As Variant*

Sets 0 or 1 to the indicated line.

### Note

In VC++, *value* is a VARIANT of VT_I4.

## ReadBackDOPort Method

Reads back data from the indicated digital output port.

### Syntax

Function *object*.**ReadBackDOPort (***port As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*port As Integer*

Selects one port for outputting digital data, number: 0~7.

0: P1A 1: P1B 2: P1CLower 3: P1CUpper

4: P2A 5: P2B 6: P2CLower 7: P2CUpper

*value As Variant*

Data that is read back from the indicated port.

### Note

In VC++, *value* is a VARIANT of VT_I4.

## ReadBackDOLine Method

Reads back data from the indicated digital output line of the indicated digital output port.

### Syntax

Function *object*.**ReadBackDOLine(***port As Integer, line As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*port As Integer*

Selects one port for outputting digital data, number: 0~7.

0: P1A 1: P1B 2: P1CLower 3: P1CUpper

4: P2A 5: P2B 6: P2CLower 7: P2CUpper

*line As Integer*

Selects one line number from the indicated port: from 0 to 7 (8-bit port).

*value As Variant*

Data that is read back from the indicated line.

### Note

In VC++, *value* is a VARIANT of VT_UI1.

# Events

## *Interrupt Event*

### Syntax

sub *ControlName*_**Interrupt (** *EvtNo as Integer* **)**

### Arguments

*EvtNo as Integer*

0: interrupt 1 occurred

1: interrupt 2 occurred

### Remarks

If an interrupt is generated by this card, this event will be generated by this ActiveX control.

## *DAQError Event*

### Syntax

sub *ControlName*_**DAQError (** *ErrString As String* **)**

### Arguments

*ErrString As String*

The string of error reason.

### Remarks

This event will occur when some error occur in control

# Pci7250 ActiveX Control

The PCI-7250/51 is a PCI-bus digital I/O card. The Pci7250 ActiveX control is an software component that provides the interface for user to control PCI-7250/51 card.

# Properties

## *DeviceName Property*

Returns/sets a string that indicates the device name of a PCI-7250 card that was defined in NuDAQCfg Utility.

**Syntax**

*object***.DeviceName** [= string]

**Remarks**

The device name of NuDAQ cards must be defined in NuDAQCfg Utility. The indicated NuDAQ card by device name may be a local device or a remote device on remote machine. The device name of PCI7250 ActiveX Control must be specified to a PCI-7250 card defined in NuDAQCfg Utility.

**Data Type**

String

## *OpenMode Property*

Returns/sets a value that determines the mode of opening device .

**Syntax**

*object***.OpenMode** [= number]

**Settings**

| Value | Description |
|---|---|
| 0 | Automatically open device when the control was created |
| 1 | Don't open device when the control was created. Must call the Open method to open device. (Manual) |

**Data Type**

Integer.

# Methods

## *Open Method*

### Syntax

Function *object*.**Open (***[ErrMsgBox As Variant]***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*[ErrMsgBox As Variant]*

It is optional and boolean type. Default value is False

True: It will popup error message dialog box when the opening device is failed.

False: It will fire DAQError event instead of popping up dialog when the opening device is failed.

### Remarks

This method will be use when the OpenMode property is Manual.

### Note

In VC++, *ErrMsgBox* is a VARIANT of VT_I2.

## *ReadDIPort Method*

### Syntax

Function *object*.**ReadDIPort (***port As Integer, value as Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*Port As Integer*

Selects one port number from 0 to 3.

*value As Variant*

Sets one value buffer for retrieving 8-bit input data from PCI-7250 card.

### Remarks

Users can read 8-bit digital input data from PCI-7250/51 card.

### Note

In VC++, *value* is a VARIANT of VT_I4.

## *ReadDILine Method*

### Syntax

Function *object*.**ReadDILine (***port As Integer, line As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*Port As Integer*

Selects one port number from 0 to 3.

*line As Integer*

Selects one line number from the indicated port: from 0 to 7 (8-bit port).

*value As Variant*

Returns the data input from PCI-7250/51 card.

### Remarks

Users can read data from the indicated digital input line of PCI-7250/51 card.

### Note

In VC++, *value* is a VARIANT of VT_UI1.

## *WriteDOPort Method*

### Syntax

Function *object*.**WriteDOPort (***port As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*port As Integer*

Selects one port number from 0 to 3.

*value as Variant*

8-bit data that will be written to the digital output port.

### Remarks

Users can write data to the PCI-7250/51 digital output port.

### Note

In VC++, *value* is a VARIANT of VT_I4.

## *WriteDOLine Method*

### Syntax

Function *object.***WriteDOLine(***port As Integer, line As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*port As Integer*

Selects one port number from 0 to 3.

*line As Integer*

Selects one line number from the indicated port from 0 to 7.

*value As Variant*

Sets 0 or 1 to the indicated line.

**Note**

In VC++, *value* is a VARIANT of VT_I4.

## *ReadBackDOPort Method*

Reads back data from the indicated digital output port.

**Syntax**

Function *object*.**ReadBackDOPort (***port As Integer, value As Variant***)** As Boolean

**Return Value**

True if the function is successful; otherwise False.

**Arguments**

*port As Integer*

Select one port number from 0 to 3.

*value As Variant*

Data that is read back from the indicated port.

**Note**

In VC++, *value* is a VARIANT of VT_I4.

## *ReadBackDOLine Method*

Reads back data from the indicated digital output line of the indicated digital output port.

**Syntax**

Function *object*.**ReadBackDOLine(***port As Integer, line As Integer, value As Variant***)** As Boolean

**Return Value**

True if the function is successful; otherwise False.

**Arguments**

*port As Integer*

Selects one port number from 0 to 3.

*line As Integer*

Selects one line number from the indicated port from 0 to 7.

*value As Variant*

Data that is read back from the indicated line.

**Note**

In VC++, *value* is a VARIANT of VT_UI1.

# Events

# *DAQError Event*

### Syntax

sub *ControlName*_**DAQError (** *ErrString As String* **)**

### Arguments

*ErrString As String*

The string of error reason

### Remarks

This event will occur when some error occur in control

# Pci7252 ActiveX Control

The cPCI-7252 is a CompactPCI digital I/O card. The Pci7252 ActiveX control is a software component that provides the interface for user to control cPCI-7252 card.

# Properties

## *DeviceName Property*

Returns/sets a string that indicates the device name of a cPCI-7252 card that was defined in NuDAQCfg Utility.

**Syntax**

*object***.DeviceName** [= string]

**Remarks**

The device name of NuDAQ cards must be defined in NuDAQCfg Utility. The indicated NuDAQ card by device name may be a local device or a remote device on remote machine. The device name of PCI7252 ActiveX Control must be specified to a cPCI-7252 card defined in NuDAQCfg Utility.

**Data Type**

String

## *OpenMode Property*

Returns/sets a value that determines the mode of opening device .

**Syntax**

*object***.OpenMode** [= number]

**Settings**

| Value | Description |
|---|---|
| 0 | Automatically open device when the control was created |
| 1 | Don't open device when the control was created. Must call the Open method to open device. (Manual) |

**Data Type**

Integer

# Methods

## *Open Method*

### Syntax

Function *object*.**Open (***[ErrMsgBox As Variant]***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*[ErrMsgBox As Variant]*

It is optional and boolean type. Default value is False

True, It will popup error message dialog box when the opening device is failed.

False, It will fire DAQError event instead of popping up dialog when the opening device is failed.

### Remarks

This method will be used when the OpenMode property is Manual.

### Note

In VC++, *ErrMsgBox* is a VARIANT of VT_I2.

## *ReadDIPort Method*

### Syntax

Function *object*.**ReadDIPort (***port As Integer, value as Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*Port As Integer*

This field can only be 0.

*value As Variant*

Sets one value buffer for retrieving 8-bit input data from cPCI-7252 card.

### Remarks

Users can read 16-bit digital input data from cPCI-7252 card.

### Note

In VC++, *value* is a VARIANT of VT_I2.

## *ReadDILine Method*

### Syntax

Function *object*.**ReadDILine (***port As Integer, line As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*Port As Integer*

This field can only be 0.

*line As Integer*

Selects one line number from the indicated port: from 0 to 15 (16-bit port).

*value As Variant*

Returns the data input from cPCI-7252 card.

### Remarks

Users can read data from the indicated digital input line of cPCI-7252 card.

### Note

In VC++, *value* is a VARIANT of VT_UI1.

## *WriteDOPort Method*

### Syntax

Function *object*.**WriteDOPort (***port As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*port As Integer*

This field can only be 0.

*value as Variant*

8-bit data that will be written to the digital output port.

### Remarks

Users can write data to the cPCI-7252 digital output port.

### Note

In VC++, *value* is a VARIANT of VT_I4.

## *WriteDOLine Method*

### Syntax

Function *object.***WriteDOLine(***port As Integer, line As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*port As Integer*

This field can only be 0.

*line As Integer*

Selects one line number from the indicated port from 0 to 7.

*value As Variant*

Sets 0 or 1 to the indicated line.

**Note**

In VC++, *value* is a VARIANT of VT_I4.

## *ReadBackDOPort Method*

Reads back data from the indicated digital output port.

**Syntax**

Function *object*.**ReadBackDOPort (***port As Integer, value As Variant***)** As Boolean

**Return Value**

True if the function is successful; otherwise False.

**Arguments**

*port As Integer*

This field can only be 0.

*value As Variant*

Data that is read back from the indicated port.

**Note**

In VC++, *value* is a VARIANT of VT_I2.

## *ReadBackDOLine Method*

Reads back data from the indicated digital output line of the indicated digital output port.

**Syntax**

Function *object*.**ReadBackDOLine(***port As Integer, line As Integer, value As Variant***)** As Boolean

**Return Value**

True if the function is successful; otherwise False.

**Arguments**

*port As Integer*

This field can only be 0.

*line As Integer*

Selects one line number from the indicated port from 0 to 7.

*value As Variant*

Data that is read back from the indicated line.

**Note**

In VC++, *value* is a VARIANT of VT_UI1

# Events

# *DAQError Event*

### Syntax

sub *ControlName*_**DAQError (** *ErrString As String* **)**

### Arguments

*ErrString As String*

The string of error reason

### Remarks

This event will occur when some error occur in control

# Pci7296 ActiveX Control

The PCI-7296 is a PCI-bus digital I/O card. The Pci7296 ActiveX control is an software component that provides the interface for users to control PCI-7296 card.

# Properties

## *DeviceName Property*

Returns/sets a string that indicates the device name of a PCI-7296 card that was defined in NuDAQCfg Utility.

**Syntax**

*object***.DeviceName** [= string]

**Remarks**

The device name of NuDAQ cards must be defined in NuDAQCfg Utility. The indicated NuDAQ card by device name may be a local device or a remote device on remote machine. The device name of PCI7296 ActiveX Control must be specified to a PCI-7296 card defined in NuDAQCfg Utility.

**Data Type**

String

## *INT1Mode Property*

Returns/sets a value that determines the interrupt mode of INT1.

**Syntax**

*object***.INT1Mode** [= Mode]

**Settings**

| Value | Description |
|---|---|
| 0 | Disable |
| 1 | falling edge of P1C0 |
| 2 | rising edge of P1C0 or P1C3 |
| 3 | event counter down to 0 |

**Data Type**

Integer

## *INT2Mode Property*

Returns/sets a value that determines the interrupt mode of INT2.

### Syntax

*object*.**INT2Mode** [= Mode]

### Settings

| Value | Description |
|---|---|
| 0 | Disable |
| 1 | falling edge of P2C0 |
| 2 | rising edge of P2C0 or P2C3 |
| 3 | event counter down to 0 |

### Data Type

Integer

## OpenMode Property

Returns/sets a value that determines the mode of opening device .

### Syntax

*object*.**OpenMode** [= number]

### Settings

| Value | Description |
|---|---|
| 0 | Automatically open device when the control was created |
| 1 | Don't open device when the control was created. Must call the Open method to open device. (Manual) |

### Data Type

Integer

## P1Adir Property

Returns/sets a value that determines P1A port direction.

### Syntax

*object*.**P1ADir** [= Direction]

### Settings

| Value | Description |
|---|---|
| 0 | Direction: Input_Port. |
| 1 | Direction: Output_Port. |

### Data Type

Integer

## P1Bdir Property

Returns/sets a value that determines P1B port direction.

### Syntax

*object*.**P1BDir** [= Direction]

### Settings

| Value | Description |
|---|---|
| 0 | Direction: Input_Port. |
| 1 | Direction: Output_Port. |

### Data Type

Integer

## P1CLowerdir Property

Returns/sets a value that determines P1C lower port direction.

### Syntax

*object*.**P1CLowerDir** [= Direction]

### Settings

| Value | Description |
|---|---|
| 0 | Direction: Input_Port. |
| 1 | Direction: Output_Port. |

### Data Type

Integer

## P1CUpperdir Property

Returns/sets a value that determines P1C upper port direction.

### Syntax

*object*.**P1CUpperDir** [= Direction]

### Settings

| Value | Description |
|---|---|
| 0 | Direction: Input_Port. |
| 1 | Direction: Output_Port. |

### Data Type

Integer

## P2Adir Property

Returns/sets a value that determines P2A port direction.

### Syntax

*object*.**P2ADir** [= Direction]

### Settings

| Value | Description |
|---|---|
| 0 | Direction: Input_Port. |
| 1 | Direction: Output_Port. |

### Data Type

Integer

## P2Bdir Property

Returns/sets a value that determines P2B port direction.

### Syntax

*object*.**P2BDir** [= Direction]

### Settings

| Value | Description |
|---|---|
| 0 | Direction: Input_Port. |
| 1 | Direction: Output_Port. |

### Data Type

Integer

## P2CLowerdir Property

Returns/sets a value that determines P2C lower port direction.

### Syntax

*object*.**P2CLowerDir** [= Direction]

### Settings

| Value | Description |
|---|---|
| 0 | Direction: Input_Port. |
| 1 | Direction: Output_Port. |

### Data Type

Integer

## P2CUpperdir Property

Returns/sets a value that determines P2C upper port direction.

### Syntax

*object*.**P2CUpperDir** [= Direction]

### Settings

| Value | Description |
|---|---|
| 0 | Direction: Input_Port. |
| 1 | Direction: Output_Port. |

### Data Type

Integer

## P3Adir Property

Returns/sets a value that determines P3A port direction.

### Syntax

*object*.**P3ADir** [= Direction]

### Settings

| Value | Description |
|---|---|
| 0 | Direction: Input_Port. |
| 1 | Direction: Output_Port. |

### Data Type

Integer

## P3Bdir Property

Returns/sets a value that determines P3B port direction.

### Syntax

*object*.**P3BDir** [= Direction]

### Settings

| Value | Description |
|---|---|
| 0 | Direction: Input_Port. |
| 1 | Direction: Output_Port. |

### Data Type

Integer

## P3CLowerdir Property

Returns/sets a value that determines P3C lower port direction.

### Syntax

*object*.**P3CLowerDir** [= Direction]

### Settings

| Value | Description |
|---|---|
| 0 | Direction: Input_Port. |

| | |
|---|---|
| 1 | Direction: Output_Port. |

**Data Type**

Integer

## *P3CUpperdir Property*

Returns/sets a value that determines P3C upper port direction.

**Syntax**

*object*.**P3CUpperDir** [= Direction]

**Settings**

| Value | Description |
|---|---|
| 0 | Direction: Input_Port. |
| 1 | Direction: Output_Port. |

**Data Type**

Integer

## *P4Adir Property*

Returns/sets a value that determines P4A port direction.

**Syntax**

*object*.**P4ADir** [= Direction]

**Settings**

| Value | Description |
|---|---|
| 0 | Direction: Input_Port. |
| 1 | Direction: Output_Port. |

**Data Type**

Integer

## *P4Bdir Property*

Returns/sets a value that determines P4B port direction.

**Syntax**

*object*.**P4BDir** [= Direction]

**Settings**

| Value | Description |
|---|---|
| 0 | Direction: Input_Port. |
| 1 | Direction: Output_Port. |

**Data Type**

Integer.

## *P4CLowerdir Property*

Returns/sets a value that determines P4C lower port direction.

### Syntax

*object*.**P4CLowerDir** [= Direction]

### Settings

| Value | Description |
|---|---|
| 0 | Direction: Input_Port. |
| 1 | Direction: Output_Port. |

### Data Type

Integer

## *P4CUpperdir Property*

Returns/sets a value that determines P4C upper port direction.

### Syntax

*object*.**P4CUpperDir** [= Direction]

### Settings

| Value | Description |
|---|---|
| 0 | Direction: Input_Port. |
| 1 | Direction: Output_Port. |

### Data Type

Integer

# Methods

## *Open Method*

### Syntax

Function *object*.**Open (***[ErrMsgBox As Variant]***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*[ErrMsgBox As Variant]*

It is optional and boolean type. Default value is False

True, It will popup error message dialog box when the opening device is failed.

False, It will fire DAQError event instead of popping up dialog when the opening device is failed.

### Remarks

This method will be use when the OpenMode property is Manual.

### Note

In VC++, *ErrMsgBox* is a VARIANT of VT_I2.

## ReadCounter Method

### Syntax

Function *object*.**ReadCounter (***index As Integer, CtlValue As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*index As Integer*

Selects one counter number: 0~2.

*CtlValue As Variant*

Returns the content from the indicated counter.

### Remarks

You can read the current contents from the selected counter without disturbing the counting process.

### Note

In VC++, *CtlValue* is a VARIANT of VT_I4.

## ReadDIPort Method

### Syntax

Function *object***.ReadDIPort (***port As Integer, value as Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*Port As Integer*

Selects one port for inputting digital data, number: 0~15.

| | | | |
|---|---|---|---|
| 0: P1A | 1: P1B | 2: P1CLower | 3: P1CUpper |
| 4: P2A | 5: P2B | 6: P2CLower | 7: P2CUpper |
| 8: P3A | 9: P3B | 10: P3CLower | 11: P3CUpper |
| 12: P4A | 13: P4B | 14: P4CLower | 15: P4CUpper |

*value As Variant*

The digital data read from the indicated digital input port.

### Remarks

You can read data from the digital input port.

### Note

In VC++, *value* is a VARIANT of VT_I2.

## *ReadDILine Method*

### Syntax

Function *object*.**ReadDILine (***port As Integer, line As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*Port As Integer*

Selects one port for inputting digital data, number: 0~15.

| | | | |
|---|---|---|---|
| 0: P1A | 1: P1B | 2: P1CLower | 3: P1CUpper |
| 4: P2A | 5: P2B | 6: P2CLower | 7: P2CUpper |
| 8: P3A | 9: P3B | 10: P3CLower | 11: P3CUpper |
| 12: P4A | 13: P4B | 14: P4CLower | 15: P4CUpper |

*line As Integer*

Selects one line number from the indicated port: from 0 to 7 (8-bit port).

*value As Variant*

Returns the data input from PCI-7296 card.

### Remarks

Users can read data from the indicated digital input line of PCI-7296 card.

### Note

In VC++, *value* is a VARIANT of VT_UI1.

## *StartCounter Method*

### Syntax

Function *object*.**StartCounter(*index** As Integer, CtrMode As Integer, CtrValue as Variant, BinBcd as Intger***)**

As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*index As Integer*

Selects one counter number: 0~2.

*CtrMode As Integer*

0: Toggle output from low to high on terminal count

1: Programmable one-shot

2: Rate generator

3: Square wave rate generator

4: Software-triggered strobe

5: Hardware-triggered strobe

*CtrValue as Variant*

Set the start value to the indicated counter.

*binbcd As Integer*

0: 16-bit binary counter, 1: 4-decade BCD counter.

### Remarks

You can start the indicated counter to operate in the specified mode.

### Note

In VC++, *CtrValue* is a VARIANT of VT_I4.

## StopCounter Method

### Syntax

Function *object*.**StopCounter(***index As Integer, State As Integer***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*index As Integer*

Selects one counter number: 0~2.

*State As Integer*

The logic state to which the counter is to be stopped, number: 0 or 1.

### Remarks

You can stop the indicated counter and set the specified state.

## WriteDOPort Method

### Syntax

Function *object*.**WriteDOPort (***port As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*port As Integer*

Selects one port for inputting digital data, number: 0~15.

| | | | |
|---|---|---|---|
| 0: P1A | 1: P1B | 2: P1CLower | 3: P1CUpper |
| 4: P2A | 5: P2B | 6: P2CLower | 7: P2CUpper |
| 8: P3A | 9: P3B | 10: P3CLower | 11: P3CUpper |
| 12: P4A | 13: P4B | 14: P4CLower | 15: P4CUpper |

*value as Variant*

8-bit data that will be written to the digital output port.

### Remarks

Users can write data to the PCI-7296 digital output port.

### Note

In VC++, *value* is a VARIANT of VT_I4.

## *WriteDOLine Method*

### Syntax

Function *object.***WriteDOLine(***port As Integer, line As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*port As Integer*

Selects one port for outputting digital data, number: 0~15.

| | | | |
|---|---|---|---|
| 0: P1A | 1: P1B | 2: P1CLower | 3: P1CUpper |
| 4: P2A | 5: P2B | 6: P2CLower | 7: P2CUpper |
| 8: P3A | 9: P3B | 10: P3CLower | 11: P3CUpper |
| 12: P4A | 13: P4B | 14: P4CLower | 15: P4CUpper |

*line As Integer*

Selects one line number from the indicated port: from 0 to 7 (8-bit port).

*value As Variant*

Sets 0 or 1 to the indicated line.

### Note

In VC++, *value* is a VARIANT of VT_I4.

## *ReadBackDOPort Method*

Reads back data from the indicated digital output port.

### Syntax

Function *object.***ReadBackDOPort (***port As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*port As Integer*

Selects one port for outputting digital data, number: 0~15.

| | | | |
|---|---|---|---|
| 0: P1A | 1: P1B | 2: P1CLower | 3: P1CUpper |
| 4: P2A | 5: P2B | 6: P2CLower | 7: P2CUpper |
| 8: P3A | 9: P3B | 10: P3CLower | 11: P3CUpper |
| 12: P4A | 13: P4B | 14: P4CLower | 15: P4CUpper |

*value As Variant*

Data that is read back from the indicated port.

### Note

In VC++, *value* is a VARIANT of VT_I4.

## *ReadBackDOLine Method*

Reads back data from the indicated digital output line of the indicated digital output port.

### Syntax

Function *object*.**ReadBackDOLine(***port As Integer, line As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*port As Integer*

Selects one port for outputting digital data, number: 0~15.

| | | | |
|---|---|---|---|
| 0: P1A | 1: P1B | 2: P1CLower | 3: P1CUpper |
| 4: P2A | 5: P2B | 6: P2CLower | 7: P2CUpper |
| 8: P3A | 9: P3B | 10: P3CLower | 11: P3CUpper |
| 12: P4A | 13: P4B | 14: P4CLower | 15: P4CUpper |

*line As Integer*

Selects one line number from the indicated port: from 0 to 7 (8-bit port).

*value As Variant*

Data that is read back from the indicated line.

### Note

In VC++, *value* is a VARIANT of VT_UI1.

# Events

## *Interrupt Event*

### Syntax

sub *ControlName*_**Interrupt (** *EvtNo as Integer* **)**

### Arguments

*EvtNo as Integer*

0: interrupt 1 occurred

1: interrupt 2 occurred

### Remarks

If an interrupt is generated by this card, this event will be generated by this ActiveX control.

## *DAQError Event*

### Syntax

sub *ControlName*_**DAQError (** *ErrString As String* **)**

### Arguments

*ErrString As String*

The string of error reason

### Remarks

This event will occur when some error occur in control

# Pci7300 ActiveX Control

The PCI-7300 is a PCI-bus high speed DI/O card. The Pci7300 ActiveX control is an software component that provides the interface for users to control PCI-7300 card.

# Properties

## *DeviceName Property*

Returns/sets a string that indicates the device name of a PCI-7300 card that was defined in NuDAQCfg Utility.

**Syntax**

*object***.DeviceName** [= string]

**Remarks**

The device name of NuDAQ cards must be defined in NuDAQCfg Utility. The indicated NuDAQ card by device name may be a local device or a remote device on remote machine. The device name of PCI7300 ActiveX Control must be specified to a PCI-7300A card defined in NuDAQCfg Utility.

**Data Type**

String

## *DIClearFIFO Property*

Returns/sets a value that determines if clear FIFO function of digital input

**Syntax**

*object***.DIClearFIFO** [= boolean]

**Settings**

| Value | Description |
|---|---|
| False | not clear FIFO at digital input |
| True | clear FIFO at digital input |

**Data Type**

Boolean

## *DIClockSource Property*

Returns/sets a value that determines the clock source of digital input.

**Syntax**

*object***.DIClockSource** [= number]

### Settings

| Value | Description |
|---|---|
| 0 | Internal pacer timer 0 |
| 1 | External Trigger |
| 2 | Handshaking |
| 3 | Internal 10MHz clock |
| 4 | Internal 20MHz clock |

### Data Type

Integer

## *DIDisableAfterDMA Property*

Returns/sets a value that determines if disable DI after DMA complete.

### Syntax

*object*.**DIDisableAfterDMA** [= boolean]

### Settings

| Value | Description |
|---|---|
| False | not disable DI after DMA complete |
| True | disable DI after DMA complete |

### Data Type

Boolean

## *DINPatterns property*

Sets a value that indicates the total number of patterns of DMA digital input or circular buffer size.

### Syntax

*object*.**DINPatterns** [= number]

### Remarks

If double buffer mode is disabled, this value is the number of DMA digital input to be performed. If Double-buffer Mode value is True, this is the size (in samples) of the circular buffer and its value must be a multiple of 8.

### Data Type

Long

## *DIScanRate Property*

Returns/sets a value that determines the samples per second of DMA digital input.

### Syntax

*object*.**DIScanRate** [= number]

### Settings

The range of DIScanRate is between 0 and 20MHz.

### Remarks

This value is used only when the DIClockSource property is set to “Internal pacer timer0”

### Data Type

Long

## *DISignalPolarity Property*

Returns/sets a value that determines the I_REQ , I_ACK, I_TRIG signal polarity. This function only works with PCI-7300A Revision B.

### Syntax

*object*.**DISignalPolarity** [= number]

### Settings

The setting is a value formed by addition from one or more of the following values.

| Value | Description |
|---|---|
| 0 | I_REQ is rising edge active. |
| 1 | I_REQ is falling edge active. |
| 0 | I_ACK is rising edge active. |
| 2 | I_ACK is falling edge active. |
| 0 | I_TRIG is rising edge active. |
| 4 | I_TRIG is falling edge active. |

### Data Type

Integer

## *DITerminator property*

Returns/sets a value that determines the PortA Terminator On/Off,.

### Syntax

*object*.**DITerminator** [= boolean]

### Settings

| Value | Description |
|---|---|
| True | PortA terminator on. |
| False | PortA terminator off. |

### Data Type

Boolean

# DITriggerWait Property

Returns/sets a value that determines the DI Wait Trigger Status.

### Syntax

*object*.**DITriggerWait** [= number]

### Settings

| Value | Description |
|---|---|
| 0 | No Wait, input sampling starts immediately. |
| 1 | Wait Trigger, input sampling waits rising or falling edge of I_TRG to start DI. |

### Data Type

Integer

# DOClockSource Property

Returns/sets a value that determines the clock source of digital output.

### Syntax

*object*.**DOClockSource** [= number]

### Settings

**Card revision A:**

| Value | Description |
|---|---|
| 0 | Internal pacer timer 1 |
| 1 | External Trigger |
| 2 | Internal 10MHz clock |
| 3 | Internal 20MHz clock |

**Card revision B:**

| Value | Description |
|---|---|
| 0 | Internal pacer timer 1 |
| 1 | External Trigger |
| 2 | Internal 10MHz clock |
| 3 | Internal 20MHz clock |
| 4 | Burst handshaking mode by using timer1 output as output clock |
| 5 | Burst handshaking mode by using 10MHz clock as output clock |
| 6 | Burst handshaking mode by using 20MHz clock as output clock |

### Data Type

Integer

# DOFIFOThreshold Property

Returns/sets a value that determines the FIFO threshold value of DO

**Syntax**

*object*.**DOFIFOThreshold** [= number]

**Settings**

The range is between 0 and the size of FIFO

**Data Type**

Long

## DOIterations Property

Returns/sets a value that indicates the iterations of digital output.

**Syntax**

*object*.**DOIterations** [= number]

**Remarks**

The value range is 0 ~ 65535.

Iterations property means the number of times the data in buffer to output to channel in continuous DMA digital output operation. A value of 0 means that digital output operation proceeds indefinitely.

**Data Type**

Integer

## DONPatterns Property

Sets a value that indicates the total number of patterns of DMA digital output.

**Syntax**

*object*.**DONPatterns** [= number]

**Remarks**

This value is the total number of patterns of DMA digital output to be performed.

**Data Type**

Long

## DOScanRate Property

Returns/sets a value that determines the samples per second of DMA digital output.

**Syntax**

*object*.**DOScanRate** [= number]

**Settings**

The range of DOScanRate is between 0 and 20MHz.

**Remarks**

This value is used only when the DOClockSource property is set to “Internal pacer timer1”

### Data Type

Long

## DOSignalPolarity Property

Returns/sets a value that determines the O_REQ, O_ACK, O_TRIG signal polarity. This function only works with PCI-7300A Revision B.

### Syntax

*object*.**DOSignalPolarity** [= number]

### Settings

The setting is a value formed by addition from one or more of the following values.

| Value | Description |
|---|---|
| 0 | O_REQ is rising edge active. |
| 8 | O_REQ is falling edge active. |
| 0 | O_ACK is rising edge active. |
| 16 | O_ACK is falling edge active. |
| 0 | O_TRIG is rising edge active. |
| 32 | O_TRIG is falling edge active. |

### Data Type

Integer

## DOTerminator Property

Returns/sets a value that determines the PortB Terminator On/Off,.

### Syntax

*object*.**DOTerminator** [= boolean]

### Settings

| Value | Description |
|---|---|
| True | PortB terminator on. |
| False | PortB terminator off. |

### Data Type

Boolean

## DOTriggerWait Property

Returns/sets a value that determines the DO Wait Trigger Status.

### Syntax

*object*.**DOTriggerWait** [= number]

### Settings

| Value | Description |
|---|---|
| 0 | No Wait, digital output starts immediately. |
| 1 | Wait Trigger, digital output waits rising or falling edge of O_TRG to start. |

### Data Type

Integer

## OpenMode Property

Returns/sets a value that determines the mode of opening device .

### Syntax

*object*.**OpenMode** [= number]

### Settings

| Value | Description |
|---|---|
| 0 | Automatically open device when the control was created |
| 1 | Don't open device when the control was created. Must call the Open method to open device. (Manual) |

### Data Type

Integer

## PortWidth Property

Returns/sets a value that determines the width of digital input and digital output port.

### Syntax

*object*.**PortWidth** [= number]

### Settings

| Value | Description |
|---|---|
| 0 | DI32 |
| 1 | DO32 |
| 2 | DI8DO8 |
| 3 | DI16DO16 |
| 4 | DI8DO16 |
| 5 | DI16DO8 |

### Data Type

Integer.

## StreamToFile Property

Returns/sets a value that determines if the control is enabled the function of streaming data to disk file.

### Syntax

*object*.**StreamToFile** [= boolean]

### Settings

| Value | Description |
|---|---|
| False | Disable the function of streaming data to disk file |
| True | Enable the function of streaming data to disk file |

### Data Type

Boolean.

# Methods

## *CheckContDI Method*

### Syntax

Function *object*.**CheckContDI(***AccessCount as long, stop as Integer***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*AccessCount as long*

Number of digital input data that has been transferred.

*stop as Integer*

Current state: 1 = stop, 0 = running

### Remarks

You can request DMA analog input status.

## *CheckContDO Method*

### Syntax

Function *object*.**CheckContDO(***AccessCount as long, stop as Integer***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*AccessCount as long*

Number of digital input data that has been transferred.

*stop as Integer*

Current state: 1 = stop, 0 = running

### Remarks

You can request DMA analog output status.

## Open Method

### Syntax

Function *object*.**Open (***[ErrMsgBox As Variant]***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*[ErrMsgBox As Variant]*

It is optional and boolean type. Default value is False

True, It will popup error message dialog box when the opening device is failed.

False, It will fire DAQError event instead of popping up dialog when the opening device is failed.

### Remarks

This method will be used when the OpenMode property is Manual.

### Note

In VC++, *ErrMsgBox* is a VARIANT of VT_I2.

## ReadAuxDILine Method

### Syntax

Function *object*.**ReadAuxDILine (***line As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*line As Integer*

Selects one line number from the indicated port: from 0 to 7 (8-bit port).

*value As Variant*

Returns the data input from PCI-7300 card.

### Remarks

Users can read data from the indicated digital input line of PCI-7300 card.

### Note

In VC++, *value* is a VARIANT of VT_UI1

## ReadAuxDIPort Method

### Syntax

Function **ReadAuxDIPort** (*value As Variant*) As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*value As Variant*

The digital data read from the digital input port.

### Remarks

You can read data from the digital input port.

### Note

In VC++, *value* is a VARIANT of VT_UI1

## *StartContDI Method*

### Syntax

Function *object*.**StartContDI ((***[FileName as Variant]***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*[FileName as String]*

It is optional and String type. Default value is NULL

FileName specified the file name of streaming data to disk.

### Remarks

Start the DMA digital input function of PCI-7300 card. If the StreamToFile property is True then the DMA data will be write the file specified by FileName. Otherwise, the FileName parameter will be ignored.

The data file is written in binary format. DAQBench provides a convenient tool DAQCvt to convert the binary file to the file format read easily. See DAQBench User's Guide for the usage of the utility. If you want to handle the data by yourself, please refer to Appendix Data File Format for the file structure.

### Note

In VC++, FileName is a VARIANT of VT_BSTR.

## *StartContDO Method*

### Syntax

Function *object*.**StartContDO(***Buffer as Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*Buffer as Variant*

A buffer data or a array of buffer data, data type can be byte, integer, long, float, double

**Remarks**

You can use this method to start the DMA digital output function of PCI-7300 card.

**Note**

In VC++, Buffer is a VARIANT of VT_ARRAY | VT_I2, VT_ARRAY | VT_I4, VT_ARRAY | VT_R4, VT_ARRAY | VT_R8, VT_ARRAY | VT_UI1.

## *StopContDI Method*

**Syntax**

Function *object*.**StopContDI ()** As Boolean

**Return Value**

True if the function is successful; otherwise False.

**Arguments**

None

**Remarks**

You can use this method to stop DMA digital input.

## *StopContDO Method*

**Syntax**

Function *object*.**StopContDO ()** As Boolean

**Return Value**

True if the function is successful; otherwise False.

**Arguments**

None

**Remarks**

You can use this method to stop DMA digital output.

## *WriteAuxDOPort Method*

**Syntax**

Function *object*.**WriteAuxDOPort (***value As Variant***)** As Boolean

**Return Value**

True if the function is successful; otherwise False.

**Arguments**

*value as Variant*

8-bit data that will be written to the digital output port.

### Remarks

Users can write data to the PCI-7300 digital output port.

### Note

In VC++, *value* is a VARIANT of VT_I4.

## *WriteAuxDOLine Method*

### Syntax

Function *object.***WriteAuxDOLine(***line As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*line As Integer*

Selects one line number from the auxiliary port: from 0 to 3.

*value As Variant*

Sets 0 or 1 to the indicated line.

### Note

In VC++, *value* is a VARIANT of VT_I4.

## *ReadBackAuxDOPort Method*

Reads back data from the auxiliary digital output port.

### Syntax

Function *object.***ReadBackDOPort (***value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*value As Variant*

Data that is read back from the auxiliary ouput port.

### Note

In VC++, *value* is a VARIANT of VT_I4.

## *ReadBackAuxDOLine Method*

Reads back data from the indicated digital output line of the auxiliary digital output port.

### Syntax

Function *object.***ReadBackAuxDOLine(***line As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*line As Integer*

Selects one line number from the indicated port: from 0 to 3.

*value As Variant*

Data that is read back from the indicated line.

### Note

In VC++, *value* is a VARIANT of VT_UI1.

# Events

## *DiComplete Event*

### Syntax

sub *ControlName*_**DiComplete (** *WaveForm As Variant* **)**

### Arguments

*WaveForm As Variant*

The data that retrieved from DMA buffer

### Remarks

This event will occur when digital input DMA function have completed

### Note

In VC++, *WaveForm* is a VARIANT of VT_ARRAY | VT_UI1, VT_ARRAY | VT_I2, VT_ARRAY | VT_I2, according to the PortWidth property.

.

## *DoComplete Event*

### Syntax

sub *ControlName*_**DoComplete ()**

### Arguments

None

### Remarks

This event will occur when digital output DMA function have completed

## *DAQError Event*

### Syntax

sub *ControlName*_**DAQError (** *ErrString As String* **)**

### Arguments

*ErrString As String*

The string of error reasion

### Remarks

This event will occur when some error occur in control

# Pci7396 ActiveX Control

The PCI-7396 is a PCI-bus digital I/O card. The Pci7396 ActiveX control is a software component that provides the interface for users to control PCI-7396 card.

# Properties

## *DeviceName Property*

Returns/sets a string that indicates the device name of a PCI-7396 card that was defined in NuDAQCfg Utility.

**Syntax**

*object***.DeviceName** [= string]

**Remarks**

The device name of NuDAQ cards must be defined in NuDAQCfg Utility. The indicated NuDAQ card by device name may be a local device or a remote device on remote machine. The device name of PCI7396 ActiveX Control must be specified to a PCI-7396 card difined in NuDAQCfg Utility.

**Data Type**

String

## *COSINTP1A Property*

Returns/sets a value that determines COSINT function of port 1A.

**Syntax**

*object***.COSINTP1A** [= boolean]

**Settings**

| Value | Description |
|---|---|
| False | Disable |
| True | Enable |

**Remarks**

Change of state of port 1A will rise INT1 notification.

**Data Type**

Boolean

## *COSINTP1B Property*

Returns/sets a value that determines COSINT function of port 1B.

### Syntax

*object*.**COSINTP1B** [= boolean]

### Settings

| Value | Description |
|---|---|
| False | Disable |
| True | Enable |

### Remarks

Change of state of port 1B will rise INT1 notification.

### Data Type

Boolean

## *COSINTP1C Property*

Returns/sets a value that determines COSINT function of port 1C.

### Syntax

*object*.**COSINTP1C** [= boolean]

### Settings

| Value | Description |
|---|---|
| False | Disable |
| True | Enable |

### Remarks

Change of state of port 1C will rise INT1 notification.

### Data Type

Boolean

## *COSINTP2A Property*

Returns/sets a value that determines COSINT function of port 2A.

### Syntax

*object*.**COSINTP2A** [= boolean]

### Settings

| Value | Description |
|---|---|
| False | Disable |
| True | Enable |

### Remarks

Change of state of port 2A will rise INT1 notification.

### Data Type

Boolean

## *COSINTP2B Property*

Returns/sets a value that determines COSINT function of port 2B.

**Syntax**

*object***.COSINTP2B** [= boolean]

**Settings**

| Value | Description |
|---|---|
| False | Disable |
| True | Enable |

**Remarks**

Change of state of port 2B will rise INT1 notification.

**Data Type**

Boolean

## *COSINTP2C Property*

Returns/sets a value that determines COSINT function of port 2C.

**Syntax**

*object***.COSINTP2C** [= boolean]

**Settings**

| Value | Description |
|---|---|
| False | Disable |
| True | Enable |

**Remarks**

Change of state of port 2C will rise INT1 notification.

**Data Type**

Boolean

## *COSINTP3A Property*

Returns/sets a value that determines COSINT function of port 3A.

**Syntax**

*object***.COSINTP3A** [= boolean]

**Settings**

| Value | Description |
|---|---|
| False | Disable |
| True | Enable |

**Remarks**

Change of state of port 3A will rise INT2 notification.

**Data Type**

Boolean

## *COSINTP3B Property*

Returns/sets a value that determines COSINT function of port 3B.

**Syntax**

*object*.**COSINTP3B** [= boolean]

**Settings**

| Value | Description |
|---|---|
| False | Disable |
| True | Enable |

**Remarks**

Change of state of port 3B will rise INT2 notification.

**Data Type**

Boolean

## *COSINTP3C Property*

Returns/sets a value that determines COSINT function of port 3C.

**Syntax**

*object*.**COSINTP3C** [= boolean]

**Settings**

| Value | Description |
|---|---|
| False | Disable |
| True | Enable |

**Remarks**

Change of state of port 3C will rise INT2 notification.

**Data Type**

Boolean

## *COSINTP4A Property*

Returns/sets a value that determines COSINT function of port 4A.

**Syntax**

*object*.**COSINTP4A** [= boolean]

**Settings**

| Value | Description |
|---|---|
| False | Disable |
| True | Enable |

**Remarks**

Change of state of port 4A will rise INT2 notification.

**Data Type**

Boolean

## *COSINTP4B Property*

Returns/sets a value that determines COSINT function of port 4B.

**Syntax**

*object*.**COSINTP4B** [= boolean]

**Settings**

| Value | Description |
|---|---|
| False | Disable |
| True | Enable |

**Remarks**

Change of state of port 4B will rise INT2 notification.

**Data Type**

Boolean

## *COSINTP4C Property*

Returns/sets a value that determines COSINT function of port 4C.

**Syntax**

*object*.**COSINTP4C** [= boolean]

**Settings**

| Value | Description |
|---|---|
| False | Disable |
| True | Enable |

**Remarks**

Change of state of port 4C will rise INT2 notification.

**Data Type**

Boolean

## *INT1Mode Property*

Returns/sets a value that determines the interrupt mode of INT1.

### Syntax

*object*.**INT1Mode** [= Mode]

### Settings

| Value | Description |
|---|---|
| 0 | Disable |
| 1 | falling edge of P1C0 |
| 2 | rising edge of P1C0 or P1C3 |
| 3 | event counter down to 0 |

### Data Type

Integer

## *INT2Mode Property*

Returns/sets a value that determines the interrupt mode of INT2.

### Syntax

*object*.**INT2Mode** [= Mode]

### Settings

| Value | Description |
|---|---|
| 0 | Disable |
| 1 | falling edge of P2C0 |
| 2 | rising edge of P2C0 or P2C3 |
| 3 | event counter down to 0 |

### Data Type

Integer

## *OpenMode Property*

Returns/sets a value that determines the mode of opening device .

### Syntax

*object*.**OpenMode** [= number]

### Settings

| Value | Description |
|---|---|
| 0 | Automatically open device when the control was created |
| 1 | Don’t open device when the control was created. Must call the Open method to open device. (Manual) |

### Data Type

Integer

## P1Adir Property

Returns/sets a value that determines P1A port direction.

### Syntax

*object*.**P1ADir** [= Direction]

### Settings

| Value | Description |
|---|---|
| 0 | Direction: Input_Port. |
| 1 | Direction: Output_Port. |

### Data Type

Integer

## P1Bdir Property

Returns/sets a value that determines P1B port direction.

### Syntax

*object*.**P1BDir** [= Direction]

### Settings

| Value | Description |
|---|---|
| 0 | Direction: Input_Port. |
| 1 | Direction: Output_Port. |

### Data Type

Integer.

## P1Cdir Property

Returns/sets a value that determines P1C port direction.

### Syntax

*object*.**P1CDir** [= Direction]

### Settings

| Value | Description |
|---|---|
| 0 | Direction: Input_Port. |
| 1 | Direction: Output_Port. |

### Data Type

Integer

## P2Adir Property

Returns/sets a value that determines P2A port direction.

### Syntax

*object*.**P2ADir** [= Direction]

### Settings

| Value | Description |
|---|---|
| 0 | Direction: Input_Port. |
| 1 | Direction: Output_Port. |

### Data Type

Integer

## P2Bdir Property

Returns/sets a value that determines P2B port direction.

### Syntax

*object*.**P2BDir** [= Direction]

### Settings

| Value | Description |
|---|---|
| 0 | Direction: Input_Port. |
| 1 | Direction: Output_Port. |

### Data Type

Integer

## P2Cdir Property

Returns/sets a value that determines P2C port direction.

### Syntax

*object*.**P2CDir** [= Direction]

### Settings

| Value | Description |
|---|---|
| 0 | Direction: Input_Port. |
| 1 | Direction: Output_Port. |

### Data Type

Integer

## P3Adir Property

Returns/sets a value that determines P3A port direction.

### Syntax

*object*.**P3ADir** [= Direction]

### Settings

| Value | Description |
|---|---|
| 0 | Direction: Input_Port. |
| 1 | Direction: Output_Port. |

### Data Type

Integer

## P3Bdir Property

Returns/sets a value that determines P3B port direction.

### Syntax

*object*.**P3BDir** [= Direction]

### Settings

| Value | Description |
|---|---|
| 0 | Direction: Input_Port. |
| 1 | Direction: Output_Port. |

### Data Type

Integer

## P3Cdir Property

Returns/sets a value that determines P3C port direction.

### Syntax

*object*.**P3CDir** [= Direction]

### Settings

| Value | Description |
|---|---|
| 0 | Direction: Input_Port. |
| 1 | Direction: Output_Port. |

### Data Type

Integer

## P4Adir Property

Returns/sets a value that determines P4A port direction.

### Syntax

*object*.**P4ADir** [= Direction]

### Settings

| Value | Description |
|---|---|
| 0 | Direction: Input_Port. |
| 1 | Direction: Output_Port. |

### Data Type

Integer

## *P4Bdir Property*

Returns/sets a value that determines P4B port direction.

### Syntax

*object*.**P4BDir** [= Direction]

### Settings

| Value | Description |
|---|---|
| 0 | Direction: Input_Port. |
| 1 | Direction: Output_Port. |

### Data Type

Integer

## *P4Cdir Property*

Returns/sets a value that determines P4C port direction.

### Syntax

*object*.**P4CDir** [= Direction]

### Settings

| Value | Description |
|---|---|
| 0 | Direction: Input_Port. |
| 1 | Direction: Output_Port. |

### Data Type

Integer

# Methods

## *Open Method*

### Syntax

Function *object*.**Open (***[ErrMsgBox As Variant]***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*[ErrMsgBox As Variant]*

It is optional and boolean type. Default value is False

True, It will popup error message dialog box when the opening device is failed.

False, It will fire DAQError event instead of popping up dialog when the opening device is failed.

### Remarks

This method will be use when the OpenMode property is Manual.

### Note

In VC++, *ErrMsgBox* is a VARIANT of VT_I2.

## ReadCounter Method

### Syntax

Function *object*.**ReadCounter (***index As Integer, CtlValue As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*index As Integer*

Selects one counter number: 0~2.

*CtlValue As Variant*

Returns the content from the indicated counter.

### Remarks

You can read the current contents from the selected counter without disturbing the counting process.

### Note

In VC++, *CtlValu*e is a VARIANT of VT_I4.

## ReadDIPort Method

### Syntax

Function *object***.ReadDIPort (***port As Integer, value as Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*Port As Integer*

Selects one port for inputting digital data, number: 0~11.

0: P1A 1: P1B 2: P1C

3: P2A 4: P2B 5: P2C

6: P3A 7: P3B 8: P3C

9: P4A 10: P4B 11: P4C

*value As Variant*

The digital data read from the indicated digital input port.

**Remarks**

You can read data from the digital input port.

**Note**

In VC++, *valu*e is a VARIANT of VT_I2.

# *ReadDILine Method*

**Syntax**

Function *object*.**ReadDILine (***port As Integer, line As Integer, value As Variant***)** As Boolean

**Return Value**

True if the function is successful; otherwise False.

**Arguments**

*Port As Integer*

Selects one port for inputting digital data, number: 0~11.

0: P1A 1: P1B 2: P1C

3: P2A 4: P2B 5: P2C

6: P3A 7: P3B 8: P3C

9: P4A 10: P4B 11: P4C

*line As Integer*

Selects one line number from the indicated port: from 0 to 7 (8-bit port).

*value As Variant*

Returns the data input from PCI-7396 card.

**Remarks**

Users can read data from the indicated digital input line of PCI-7396 card.

**Note**

In VC++, *valu*e is a VARIANT of VT_UI1.

# *StartCounter Method*

**Syntax**

Function *object*.**StartCounter(*index** As Integer, CtrMode As Integer, CtrValue as Variant, BinBcd as Intger***)**

As Boolean

**Return Value**

True if the function is successful; otherwise False.

### Arguments

*index As Integer*

Selects one counter number: 0~2.

*CtrMode As Integer*

0: Toggle output from low to high on terminal count

1: Programmable one-shot

2: Rate generator

3: Square wave rate generator

4: Software-triggered strobe

5: Hardware-triggered strobe

*CtrValue as Variant*

Set the start value to the indicated counter.

*binbcd As Integer*

0: 16-bit binary counter, 1: 4-decade BCD counter.

### Remarks

You can start the indicated counter to operate in the specified mode.

### Note

In VC++, *CtrValue* is a VARIANT of VT_I4.

## *StopCounter Method*

### Syntax

Function *object*.**StopCounter(***index As Integer, State As Integer***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*index As Integer*

Selects one counter number: 0~2.

*State As Integer*

The logic state to which the counter is to be stopped, number: 0 or 1.

### Remarks

You can stop the indicated counter and set the specified state.

## *WriteDOPort Method*

### Syntax

Function *object*.**WriteDOPort (***port As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*Port As Integer*

Selects one port for inputting digital data, number: 0~11.

| | | |
|---|---|---|
| 0: P1A | 1: P1B | 2: P1C |
| 3: P2A | 4: P2B | 5: P2C |
| 6: P3A | 7: P3B | 8: P3C |
| 9: P4A | 10: P4B | 11: P4C |

*value as Variant*

8-bit data that will be written to the digital output port.

### Remarks

Users can write data to the PCI-7396 digital output port.

### Note

In VC++, *valu*e is a VARIANT of VT_I4

## *WriteDOLine Method*

### Syntax

Function *object.***WriteDOLine(***port As Integer, line As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*port As Integer*

Selects one port for outputting digital data, number: 0~11.

| | | |
|---|---|---|
| 0: P1A | 1: P1B | 2: P1C |
| 3: P2A | 4: P2B | 5: P2C |
| 6: P3A | 7: P3B | 8: P3C |
| 9: P4A | 10: P4B | 11: P4C |

*line As Integer*

Selects one line number from the indicated port: from 0 to 7 (8-bit port).

*value As Variant*

Sets 0 or 1 to the indicated line.

### Note

In VC++, *value* is a VARIANT of VT_I4.

## *ReadBackDOPort Method*

Reads back data from the indicated digital output port.

### Syntax

Function *object*.**ReadBackDOPort (***port As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*port As Integer*

Selects one port for outputting digital data, number: 0~11.

| | | |
|---|---|---|
| 0: P1A | 1: P1B | 2: P1C |
| 3: P2A | 4: P2B | 5: P2C |
| 6: P3A | 7: P3B | 8: P3C |
| 9: P4A | 10: P4B | 11: P4C |

*value As Variant*

Data that is read back from the indicated port.

### Note

In VC++, *value* is a VARIANT of VT_I4.

## *ReadBackDOLine Method*

Reads back data from the indicated digital output line of the indicated digital output port.

### Syntax

Function *object*.**ReadBackDOLine(***port As Integer, line As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*port As Integer*

Selects one port for outputting digital data, number: 0~11.

| | | |
|---|---|---|
| 0: P1A | 1: P1B | 2: P1C |
| 3: P2A | 4: P2B | 5: P2C |
| 6: P3A | 7: P3B | 8: P3C |
| 9: P4A | 10: P4B | 11: P4C |

*line As Integer*

Selects one line number from the indicated port: from 0 to 7 (8-bit port).

*value As Variant*

Data that is read back from the indicated line.

### Note

In VC++, *value* is a VARIANT of VT_UI1.

# Events

## *Interrupt Event*

### Syntax

sub *ControlName*_**Interrupt (** *EvtNo as Integer* **)**

### Arguments

*EvtNo as Integer*

0: interrupt 1 occurred

1: interrupt 2 occurred

### Remarks

If an interrupt is generated by this card, this event will be generated by this ActiveX control.

## *DAQError Event*

### Syntax

sub *ControlName*_**DAQError (** *ErrString As String* **)**

### Arguments

*ErrString As String*

The string of error reason

### Remarks

This event will occur when some error occur in control

# Pci7432 ActiveX Control

The PCI-7432 is a PCI-bus digital I/O card. The Pci7432 ActiveX control is an software component that provides the interface for user to control PCI-7432 card.

# Properties

## *DeviceName Property*

Returns/sets a string that indicates the device name of a PCI-7432 card that was defined in NuDAQCfg Utility.

**Syntax**

*object***.DeviceName** [= string]

**Remarks**

The device name of NuDAQ cards must be defined in NuDAQCfg Utility. The indicated NuDAQ card by device name may be a local device or a remote device on remote machine. The device name of PCI7432 ActiveX Control must be specified to a PCI-7432 card defined in NuDAQCfg Utility.

**Data Type**

String

## *INT1Mode Property*

Returns/sets a value that determines the interrupt mode of INT1.

**Syntax**

*object***.INT1Mode** [= Mode]

**Settings**

| Value | Description |
|---|---|
| 0 | Disable |
| 1 | Enable |

**Remarks**

Enable**:** event occurred on the rising edge of digital input channel 0

**Data Type**

Integer

## *INT2Mode Property*

Returns/sets a value that determines the interrupt mode of INT2.

### Syntax

*object*.**INT2Mode** [= Mode]

### Settings

| Value | Description |
|---|---|
| 0 | Disable |
| 1 | Enable |

### Remarks

Enable**:** event occurred on the rising edge of digital input channel 1

### Data Type

Integer

## *OpenMode Property*

Returns/sets a value that determines the mode of opening device .

### Syntax

*object*.**OpenMode** [= number]

### Settings

| Value | Description |
|---|---|
| 0 | Automatically open device when the control was created |
| 1 | Don't open device when the control was created. Must call the Open method to open device. (Manual) |

### Data Type

Integer

# Methods

## *Open Method*

### Syntax

Function *object*.**Open (***[ErrMsgBox As Variant]***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*[ErrMsgBox As Variant]*

It is optional and boolean type. Default value is False

True, It will popup error message dialog box when the opening device is failed.

False, It will fire DAQError event instead of popping up dialog when the opening device is failed.

### Remarks

This method will be use when the OpenMode property is Manual.

### Note

In VC++, *ErrMsgBox* is a VARIANT of VT_I2**.**

## *ReadDIPort Method*

### Syntax

Function *object***.ReadDIPort (***value as Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*value As Variant*

Sets one value buffer for retrieving 32-bit input data from PCI-7432 card.

### Remarks

Users can read 32-bit digital input data from PCI-7432 card.

### Note

In VC++, *value* is a VARIANT of VT_I4.

## *ReadDILine Method*

### Syntax

Function *object***.ReadDILine (***line As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*line As Integer*

Selects one line number from the indicated port: from 0 to 31 (32-bit port).

*value As Variant*

Returns the data input from PCI-7432 card.

### Remarks

Users can read data from the indicated digital input line of PCI-7432 card.

### Note

In VC++, *value* is a VARIANT of VT_UI1.

## *WriteDOPort Method*

### Syntax

Function *object*.**WriteDOPort (***value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*value as Variant*

32-bit data that will be written to the digital output port.

### Remarks

Users can write data to the PCI-7432 digital output port.

### Note

In VC++, *value* is a VARIANT of VT_I4.

# Events

## *Interrupt Event*

### Syntax

sub *ControlName*_**Interrupt (** *EvtNo as Integer* **)**

### Arguments

*EvtNo as Integer*

0: interrupt 1 occurred

1: interrupt 2 occurred

### Remarks

If an interrupt is generated by this card, this event will be generated by this ActiveX control.

## *DAQError Event*

### Syntax

sub *ControlName*_**DAQError (** *ErrString As String* **)**

### Arguments

*ErrString As String*

The string of error reason

### Remarks

This event will occur when some error occur in control

# Pci7433 ActiveX Control

The PCI-7433 is a 64-bit PCI-bus digital input card. The Pci7433 ActiveX control is an software component that provides the interface for user to control PCI-7433 card.

# Properties

## *DeviceName Property*

Returns/sets a string that indicates the device name of a PCI-7433 card that was defined in NuDAQCfg Utility.

**Syntax**

*object***.DeviceName** [= string]

**Remarks**

The device name of NuDAQ cards must be defined in NuDAQCfg Utility. The indicated NuDAQ card by device name may be a local device or a remote device on remote machine. The device name of PCI7433 ActiveX Control must be specified to a PCI-7433 card defined in NuDAQCfg Utility.

**Data Type**

String

## *INT1Mode Property*

Returns/sets a value that determines the interrupt mode of INT1.

**Syntax**

*object***.INT1Mode** [= Mode]

**Settings**

| Value | Description |
|---|---|
| 0 | Disable |
| 1 | Enable |

**Remarks**

Enable**:** event occurred on the rising edge of digital input channel 0

**Data Type**

Boolean

## *INT2Mode Property*

Returns/sets a value that determines the interrupt mode of INT2.

### Syntax

*object*.**INT2Mode** [= Mode]

### Settings

| Value | Description |
|---|---|
| 0 | Disable |
| 1 | Enable |

### Remarks

Enable**:** event occurred on the rising edge of digital input channel 1

### Data Type

Boolean

## *OpenMode Property*

Returns/sets a value that determines the mode of opening device .

### Syntax

*object*.**OpenMode** [= number]

### Settings

| Value | Description |
|---|---|
| 0 | Automatically open device when the control was created |
| 1 | Don't open device when the control was created. Must call the Open method to open device. (Manual) |

### Data Type

Integer

# Methods

## *Open Method*

### Syntax

Function *object*.**Open (***[ErrMsgBox As Variant]***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*[ErrMsgBox As Variant]*

It is optional and boolean type. Default value is False

True, It will popup error message dialog box when the opening device is failed.

False, It will fire DAQError event instead of popping up dialog when the opening device is failed.

### Remarks

This method will be use when the OpenMode property is Manual.

### Note

In VC++, *ErrMsgBox* is a VARIANT of VT_I2**.**

## *ReadDIPort Method*

### Syntax

Function *object***.ReadDIPort (***port as integer, value as Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*port as integer*

Selects one port number from 0 to 1.

*value As Variant*

Sets one value buffer for retrieving 32-bit input data from PCI-7433 card.

### Remarks

Users can read 32-bit digital input data from PCI-7433 card.

### Note

In VC++, *value* is a VARIANT of VT_I4**.**

## *ReadDILine Method*

### Syntax

Function *object***.ReadDILine(***port as integer, line As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*port as integer*

Selects one port number from 0 to 1.

*line As Integer*

Selects one line number from the indicated port: from 0 to 31 (32-bit port).

*value As Variant*

Returns the data input from PCI-7433 card.

### Remarks

Users can read data from the indicated digital input line of PCI-7433 card.

### Note

In VC++, *value* is a VARIANT of VT_UI1**.**

# Events

## *Interrupt Event*

### Syntax

sub *ControlName*_**Interrupt (** *EvtNo as Integer* **)**

### Arguments

*EvtNo as Integer*

0: interrupt 1 occurred

1: interrupt 2 occurred

### Remarks

If an interrupt is generated by this card, this event will be generated by this ActiveX control.

## *DAQError Event*

### Syntax

sub *ControlName*_**DAQError (** *ErrString As String* **)**

### Arguments

*ErrString As String*

The string of error reason

### Remarks

This event will occur when some error occur in control

# Pci7434 ActiveX Control

The PCI-7434 is a 64-bits PCI-bus digital output card. The Pci7434 ActiveX control is an software component that provides the interface for user to control PCI-7434 card.

# Properties

## *DeviceName Property*

Returns/sets a string that indicates the device name of a PCI-7434 card that was defined in NuDAQCfg Utility.

**Syntax**

*object***.DeviceName** [= string]

**Remarks**

The device name of NuDAQ cards must be defined in NuDAQCfg Utility. The indicated NuDAQ card by device name may be a local device or a remote device on remote machine. The device name of PCI7434 ActiveX Control must be specified to a PCI-7434 card defined in NuDAQCfg Utility.

**Data Type**

String

## *OpenMode Property*

Returns/sets a value that determines the mode of opening device .

**Syntax**

*object***.OpenMode** [= number]

**Settings**

| Value | Description |
|---|---|
| 0 | Automatically open device when the control was created |
| 1 | Don't open device when the control was created. Must call the Open method to open device. (Manual) |

**Data Type**

Integer

# Methods

## *Open Method*

### Syntax

Function *object*.**Open (***[ErrMsgBox As Variant]***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*[ErrMsgBox As Variant]*

It is optional and boolean type. Default value is False

True, It will popup error message dialog box when the opening device is failed.

False, It will fire DAQError event instead of popping up dialog when the opening device is failed.

### Remarks

This method will be use when the OpenMode property is Manual.

### Note

In VC++, *ErrMsgBox* is a VARIANT of VT_I2**.**

## *WriteDOPort Method*

### Syntax

Function *object*.**WriteDOPort (***port as integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*port as integer*

Selects one port number from 0 to 1.

*value As Variant*

32-bit data that will be written to the digital output port.

### Remarks

Users can write data to the PCI-7434 digital output port.

### Note

In VC++, value is a VARIANT of VT_I4.

# Events

## *DAQError Event*

### Syntax

sub *ControlName*_**DAQError (** *ErrString As String* **)**

### Arguments

*ErrString As String*

The string of error reason

### Remarks

This event will occur when some error occur in control

# Pci8554 ActiveX Control

The PCI-8554 is a PCI-bus data general-purpose counter / timer and digital I/O card. The Pci8554 ActiveX control is a software component that provides the interface for users to control PCI-8554 card.

# Properties

## *DeviceName Property*

Returns/sets a string that indicates the device name of a PCI-8554 card that is defined in NuDAQCfg Utility.

**Syntax**

*object*.**DeviceName** [= string]

**Remarks**

The device name of NuDAQ cards must be defined in NuDAQCfg Utility. The indicated NuDAQ card by device name may be a local device or a remote device on remote machine. The device name of PCI8554 ActiveX Control must be specified to a PCI-8554 card defined in NuDAQCfg Utility.

**Data Type**

String

## *CK1Source Property*

Returns/sets a value that determines CK1 source.

**Syntax**

*object*.**CK1Source** [= number]

**Settings**

| Value | Description |
|---|---|
| 0 | Fixed 8MHz |
| 1 | Output of COUT11 |

**Data Type**

Integer

## *CounterClockSource Property*

Returns/sets a array value that determines clock source of count1 ~ counter10.

**Syntax**

*object*.**CounterClockSource(***index As Integer***)** [= number]

**Parameter**

*index As Integer*

Counter number is between 0 and 9

### Settings

| Value | Description |
|---|---|
| 0 | external clock source |
| 1 | cascaded counter output of COUT(n-1) |
| 2 | CK1 |
| 3 | output of COUT10 |

### Data Type

Integer

## DeBounceSource Property

Returns/sets a value that determines DeBounce source.

### Syntax

*object*.**DeBounceSource** [= number]

### Settings

| Value | Description |
|---|---|
| 0 | output of COUT11 |
| 1 | 2MHz |

### Data Type

Integer

## INT1Mode Property

Returns/sets a value that determines the interrupt mode of INT1.

### Syntax

*object*.**INT1Mode** [= Mode]

### Settings

| Value | Description |
|---|---|
| 0 | Disable |
| 1 | Enable: COUT12 |

### Remarks

Enable**:** event occurred on the clock of COUT12

### Data Type

Integer

## INT2Mode Property

Returns/sets a value that determines the interrupt mode of INT2.

### Syntax

*object*.**INT2Mode** [= Mode]

### Settings

| Value | Description |
|---|---|
| 0 | Disable |
| 1 | Enable: External signal EXTINT |

### Remarks

Enable**:** event occurred on the clock of EXTINT

### Data Type

Integer

## *OpenMode Property*

Returns/sets a value that determines the mode of opening device .

### Syntax

*object*.**OpenMode** [= number]

### Settings

| Value | Description |
|---|---|
| 0 | Automatically open device when the control was created |
| 1 | Don't open device when the control was created. Must call the Open method to open device. (Manual) |

### Data Type

Integer.

# Methods

## *Open Method*

### Syntax

Function *object*.**Open (***[ErrMsgBox As Variant]***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*[ErrMsgBox As Variant]*

It is optional and boolean type. Default value is False

True, It will popup error message dialog box when the opening device is failed.

False, It will fire DAQError event instead of popping up dialog when the opening device is failed.

### Remarks

This method will be use when the OpenMode property is Manual.

### Note

In VC++, *ErrMsgBox* is a VARIANT of VT_I2**.**

## *ReadCounter Method*

### Syntax

Function *object*.**ReadCounter (***index As Integer, CtlValue As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*index As Integer*

Selects one counter number: 0~9.

*CtlValue As Varinant*

Returns the content from the indicated counter.

### Remarks

You can read the current contents from the selected counter without disturbing the counting process.

### Note

In VC++, *CtlValue* is a VARIANT of VT_I4**.**

## *ReadDILine Method*

### Syntax

Function *object***.ReadDILine (** *line As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*line As Integer*

Selects one line number from the indicated port: from 0 to 7 (8-bit port).

*value As Variant*

Returns the data input from PCI-8554 card.

### Remarks

Users can read data from the indicated digital input line of PCI-8554card.

### Note

In VC++, *value* is a VARIANT of VT_UI1**.**

# *ReadDIPort Method*

### Syntax

Function *object*.**ReadDIPort (***value as Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*value As Variant*

The digital data read from the indicated digital input port.

### Remarks

You can read data from the digital input port.

### Note

In VC++, *value* is a VARIANT of VT_I4**.**

# *StartCounter Method*

### Syntax

Function *object*.**StartCounter(*index** *As Integer, CtrMode As Integer, CtrValue As Variant, BinBcd As Intger***)**

As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*index As Integer*

Selects one counter number: 0~9.

*CtrMode As Integer*

0: Toggle output from low to high on terminal count

1: Programmable one-shot

2: Rate generator

3: Square wave rate generator

4: Software-triggered strobe

5: Hardware-triggered strobe

*CtrValue as Variant*

Set the start value to the indicated counter.

*binbcd As Integer*

0: 16-bit binary counter, 1: 4-decade BCD counter.

### Remarks

You can start the indicated counter to operate in the specified mode.

### Note

In VC++, *CtrValue* is a VARIANT of VT_I4**.**

## *StopCounter Method*

### Syntax

Function *object*.**StopCounter(***index As Integer, State As Integer***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*index As Integer*

Selects one counter number: 0~9.

*State As Integer*

The logic state to which the counter is to be stopped, number: 0 or 1.

### Remarks

You can stop the indicated counter and set the specified state.

## *WriteDOPort Method*

### Syntax

Function *object***.WriteDOPort (** *value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*value as Variant*

8-bit data that will be written to the digital output port.

### Remarks

Users can write data to the PCI-8554 digital output port.

### Note

In VC++, *value* is a VARIANT of VT_I4**.**

# Events

## *Interrupt Event*

### Syntax

sub *ControlName*_**Interrupt (** *EvtNo as Integer* **)**

### Arguments

*EvtNo as Integer*

0: interrupt 1 occurred

1: interrupt 2 occurred

### Remarks

If an interrupt is generated by this card, this event will be generated by this ActiveX control.

## *DAQError Event*

### Syntax

sub *ControlName*_**DAQError (** *ErrString As String* **)**

### Arguments

*ErrString As String*

The string of error reason

### Remarks

This event will occur when some error occur in control

# Pci9111 ActiveX Control

The PCI-9111 is a PCI-bus multi-function data acquisition card. The Pci9111 ActiveX control is a software component that provides the interface for users to control PCI-9111 card. The properties and methods allow users to perform analog I/O and digital I/O.

# Properties

## *DeviceName Property*

Returns/sets a string that indicates the device name of a PCI-9111 card that was defined in NuDAQCfg Utility.

**Syntax**

*object*.**DeviceName** [= string]

**Remarks**

The device name of NuDAQ cards must be defined in NuDAQCfg Utility. The indicated NuDAQ card by device name may be a local device or a remote device on remote machine. The device name of PCI9111 ActiveX Control must be specified to a PCI-9111 card defined in NuDAQCfg Utility.

**Data Type**

String

## *AIRange Property*

Sets a range for interrupt analog input.

**Syntax**

*object*.**AIRange** [= number]

**Settings**

| Value | Description |
|---|---|
| 0 | ±10V |
| 1 | ±5V |
| 2 | ±2.5V |
| 3 | ±1.25V |
| 4 | ±0.625V |

**Data Type**

Integer

## *AutoScan Property*

Returns/sets a value that determines whether the analog input operate in auto scan mode or not.

**Syntax**

*object***.AutoScan** [= boolean]

**Settings**

| Value | Description |
|---|---|
| True | automatically scan channels to read analog data. The scan is in ascending order. For example, if Channel property set to 3, the channels scanned are 0, 1, 2, 3, 0, 1, 2, 3, … |
| False | only read data from the indicated channel. |

**Data Type**

Boolean

# *CardType Property*

Returns/sets a value that indicates the card type for programming Pci9111 ActiveX control.

**Syntax**

*object***.CardType** [= number]

**Settings**

| Number | Card Type | Resolution |
|---|---|---|
| 1 | PCI-9111DG | 12-bit signed data |
| 2 | PCI-9111HR | 16-bit signed data |

**Remarks**

This property will be automatically set when the device name be specified and the device information be retrieved successfully.

**Data Type**

Integer

# *Channel Property*

Sets the selected analog input channel for interrupt data transfer.

**Syntax**

*object***.Channel** [= number]

**Settings**

Channel number is from 0 to 15.

**Data Type**

Integer

# *ClockSource Property*

Returns/sets a value that determines the A/D clock source.

### Syntax

*object*.**ClockSource** [= number]

### Settings

| Value | Setting |
|---|---|
| 0 | Internal timer pacer |
| 1 | External signal trigger |

### Data Type

Integer

## DoubleBufferMode Property

Returns/Sets a value that determines whether it is double-buffer mode or not.

### Syntax

*object*.**DoublebufferMode** [= boolean]

### Settings

| Value | Description |
|---|---|
| True | During interrupt analog input mode, the buffer is divided into two parts. When either part is full of data, it will get AiHalfReady event. |
| False | The buffer is single, when it is full of data, it will get AiComplete Event. |

### Data Type

Boolean

## EDOMode Property

Returns/sets a value that determines the mode of the extended digital output channels.

### Syntax

*object*.**EDOMode** [= number]

### Settings

| Value | Desciption |
|---|---|
| 0 | Input |
| 1 | Output EDO |
| 2 | Output channel |

### Data Type

Integer

## NumOfScan Property

Sets a value that indicates the total number of scans to acquired.

### Syntax

*object*.**NumOfScan** [= number]

### Remarks

**Non-double-buffer mode**

This value multiply the total number of scan channels is the total number of A/D conversions to be performed. It must be the multiple of 512.

**Double-buffer-mode**

This value multiply the total number of scan channels is the size (in sample) of the circular buffer. It must be the multiple of 1024.

### Data Type

Long

## *OpenMode Property*

Returns/sets a value that determines the mode of opening device .

### Syntax

*object*.**OpenMode** [= number]

### Settings

| Value | Description |
|---|---|
| 0 | Automatically open device when the control was created |
| 1 | Don't open device when the control was created. Must call the Open method to open device. (Manual) |

### Data Type

Integer

## *PreTrigger Property*

Returns/sets a value that determines the pre-trigger mode.

### Syntax

*object*.**PreTrigger** [= boolean]

### Settings

| Value | Description |
|---|---|
| True | Pre-trigger enable |
| False | Pre-trigger disable |

### Remarks

When PreTrigger is True, the hardware is continuously acquiring A/D data and waiting for Pre-Trigger Signal. Before the Pre-Trigger is inserted, the software must read the FIFO data to prevent the FIFO full. When

Pre-Trigger is inserted, the counter is starting to count down from the initial counter value N (ScansAfterPreTrigger Property). The A/D trigger will be disable automatically when the counter value reaches zero.

### Data Type

Boolean

## ReturnType Property

Returns/sets a value that determines the return data type of analog input when AiComplete or AiHalfReady event would occur.

### Syntax

*object*.**ReturnType** [= number]

### Settings

**PCI-9111DG**

| Value | Description |
|---|---|
| 0 | Scaled data only |
| 1 | Binary codes with channel only |
| 2 | Binary codes without channel only |
| 3 | Scaled data and binary codes with channel |
| 4 | Scaled data and binary codes without channel |
| 5 | No data return |

**PCI-9111HR**

| Value | Description |
|---|---|
| 0 | Scaled data only |
| 1 | Binary codes only |
| 2 | Scaled data and binary codes |
| 3 | No data return |

### Data Type

Integer

## ScanRate Property

Returns/sets a value that determines the scan rate (scans per second) of continuous analog input.

### Syntax

*object*.**ScanRate** [= number]

### Settings

The range of (ScanRate * total number of scan channels) must be between 0 and 100KHz.

### Remarks

This property is used only when the Clocksource property is set to Internal timer pacer.

**Data Type**

Double

## *ScansAfterPreTrigger Property*

Returns/sets a value that determines the total number of scans will be acquired after the trigger event happens.

**Syntax**

*object*.**ScansAfterPreTrigger** [= number]

**Remarks**

The value that ScansAfterPreTrigger multiply the total number of scan channels must be less than 65535.

The timer #0 is used as the pre-trigger counter after the pre-trigger is inserted. The lock source of counter 0 is from A/D trigger source so that 8254 can count the A/D trigger number.

**Data Type**

Long

## *StreamToFile Property*

Returns/sets a value that determines if the control is enabled the function of streaming data to disk file.

**Syntax**

*object*.**StreamToFile** [= boolean]

**Settings**

| Value | Description |
|---|---|
| False | Disable the function of streaming data to disk file |
| True | Enable the function of streaming data to disk file |

**Data Type**

Boolean

# Methods

## *CheckContAI Method*

**Syntax**

Function *object*.**CheckContAI(***AccessCount as long, stop as Integer***)** As Boolean

**Return Value**

True if the function is successful; otherwise False.

**Arguments**

*AccessCount as long*

Number of analog input data that has been transferred.

*stop as Integer*

Current state: 1 = stop, 0 = running

### Remarks

You can request DMA analog input status.

## Open Method

### Syntax

Function *object*.**Open (***[ErrMsgBox As Variant]***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*[ErrMsgBox As Variant]*

It is optional and boolean type. Default value is False

True, It will popup error message dialog box when the opening device is failed.

False, It will fire DAQError event instead of popping up dialog when the opening device is failed.

### Remarks

This method will be use when the OpenMode property is Manual.

### Note

In VC++, *ErrMsgBox* is a VARIANT of VT_I2**.**

## ReadDILine Method

### Syntax

Function *object*.**ReadDILine (***line As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*line As Integer*

Selects one line number from the indicated port: from 0 to 15 (16-bit port).

*value As Variant*

Returns the data input from PCI-9111 card.

### Remarks

Users can read data from the indicated digital input line of PCI-9111 card.

### Note

In VC++, *value* is a VARIANT of VT_UI1

## ReadDIPort Method

### Syntax

Function object.**ReadDIPort** (*value As Variant*) As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*value As Variant*

Sets one value buffer for retrieving 16-bit input data from PCI-9111 card.

### Remarks

Users can read 16-bit digital input data from PCI-9111 card.

### Note

In VC++, *value* is a VARIANT of VT_I4

## ReadSingleAI Method

### Syntax

Function object.**ReadSingleAI** (*channel As Integer, range As Integer, value As Variant*) As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*channel As Integer*

Analog input channel number 0 ~ 15.

*range As Integer*

Range of analog input.

| Value | Description |
|---|---|
| 0 | ±10V |
| 1 | ±5V |
| 2 | ±2.5V |
| 3 | ±1.25V |
| 4 | ±0.625V |

*value As Variant*

The analog data read from analog input channel (already scaled to voltage).

### Remarks

You can read one data from one analog input channel.

### Note

In VC++, *value* is a VARIANT of VT_R8

## StartContAI Method

### Syntax

Function object.**StartContAI** (**(***[FileName as Variant]*) As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*[FileName as String]*

It is optional and String type. Default value is NULL

FileName specified the file name of streaming data to disk.

### Remarks

Start the continuous analog input function of PCI-9111 card. If the StreamToFile property is True then the data will be written to the file specified by FileName. Otherwise, the FileName parameter will be ignored.

The data file is written in binary format, with the lower byte first (little endian). Data type is "Binary codes with channel". DAQBench provides a convenient tool DAQCvt to convert the binary file to the file format read easily. See DAQBench User's Guide for the usage of the utility. If you want to handle the data by yourself, please refer to Appendix Data File Format for the file structure.

### Note

In VC++, FileName is a VARIANT of VT_BSTR

## StopContAI Method

### Syntax

Function object.**StopContAI** () As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

None

### Remarks

You can use this method to stop continuous analog input.

## WriteDOPort Method

### Syntax

Function *object***.WriteDOPort (***value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*value as Variant*

16-bit data that will be written to the digital output port.

### Remarks

Users can write data to the PCI-9111 digital output port.

### Note

In VC++, *value* is a VARIANT of VT_I4.

## *WriteSingleAO Method*

### Syntax

Function *object*.**WriteSingleAO (***range As integer***,** *value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*range As Integer*

Range of analog output.

| Value | Description |
|---|---|
| 0 | 0 ~ 10V |
| 1 | -10 ~ 10V |

*value As Variant*

Sets one value to writing to analog output of PCI-9111 card, the range will depend on *range* parameter.

### Remarks

Users can write data to analog output channel of PCI-9111 card. The range value must be same with the AO jumper setting of PCI-9111 card.

### Note

In VC++, *value* is a VARIANT of VT_R8.

# Events

## *AiComplete Event*

### Syntax

sub *ControlName*_**AiComplete(** *ScaledData as Variant, BinaryCodes As Variant* **)**

### Arguments

*ScaledData as Variant*

The analog input data array that have been translated to the engineering data (voltage) according to

AIRange property.

*BinaryCodes As Variant*

The analog input data array with digital format.

### Remarks

This event occurs when continuous analog input function is completed.

Whether *ScaledData* or *BinaryCodes* contains data depends on *ReturnType* property setting.

**PCI-9111DG**

"*BinaryCodes with channel*" format:

Range -32768 to +32767. The least significant 4 bits are channel number.

"*BinaryCodes without channel*" format:

Range -2048 to +2047.

**PCI-9111HR**

"*BinaryCodes*" format:

Range -32768 to +32767.

### Note

In VC++, *ScaledData* is a VARIANT of VT_ARRAY | VT_R4, *BinaryCodes* is a VARIANT of VT_ARRAY | VT_I4 (with channel) or VT_ARRAY | VT_I2 (without channel).

## *AiHalfReady Event*

### Syntax

sub *ControlName*_**AiHalfReady(** *ScaledData as Variant, BinaryCodes As Variant* **)**

### Arguments

*ScaledData as Variant*

The analog input data array that have been translated to the engineering data (voltage) according to AIRange property.

*BinaryCodes As Variant*

The analog input data array with digital format.

### Remarks

This event occurs when one half-buffer of the circular buffer is full at continuous analog input operation.

Whether *ScaledData* or *BinaryCodes* contains data depends on *ReturnType* property setting.

**PCI-9111DG**

"*BinaryCodes with channel*" format:

Range -32768 to +32767. The least significant 4 bits are channel number.

"*BinaryCodes without channel*" format:

Range -2048 to +2047.

**PCI-9111HR**

"*BinaryCodes*" format:

Range -32768 to +32767.

### Note

In VC++, *ScaledData* is a VARIANT of VT_ARRAY | VT_R4, *BinaryCodes* is a VARIANT of VT_ARRAY | VT_I4 (with channel) or VT_ARRAY | VT_I2 (without channel).

## *DAQError Event*

### Syntax

sub *ControlName*_**DAQError (** *ErrString As String* **)**

### Arguments

*ErrString As String*

The string of error reasion

### Remarks

This event will occur when some error occur in control

# Pci9112 ActiveX Control

The PCI-9112 is a PCI-bus multi-function data acquisition card. The Pci9112 ActiveX control is a software component that provides the interface for users to control PCI-9112 card. The properties and methods allow users to perform analog I/O and digital I/O.

# Properties

## *DeviceName Property*

Returns/sets a string that indicates the device name of a PCI-9112 card that was defined in NuDAQCfg Utility.

**Syntax**

*object*.**DeviceName** [= string]

**Remarks**

The device name of NuDAQ cards must be defined in NuDAQCfg Utility. The indicated NuDAQ card by device name may be a local device or a remote device on remote machine. The device name of PCI9112 ActiveX Control must be specified to a PCI-9112 card defined in NuDAQCfg Utility.

**Data Type**

String

## *AIRange Property*

Sets a range for interrupt analog input.

**Syntax**

*object*.**AIRange** [= number]

**Settings**

| Value | Description |
|---|---|
| 0 | ±5V |
| 1 | ±2.5V |
| 2 | ±1.25V |
| 3 | ±0.625V |
| 4 | 0~10V |
| 5 | 0~5V |
| 6 | 0~2.5V |
| 7 | 0~1.25V |
| 8 | ±10V |

**Data Type**

Integer

## AutoScan Property

Returns/sets a value that determines whether the analog input operate in auto scan mode or not.

**Syntax**

*object*.**AutoScan** [= boolean]

**Settings**

| Value | Description |
|---|---|
| True | automatically scan channels to read analog data. The scan is in descending order. For example, if Channel property set to 3, the channels scanned are 3, 2, 1, 0, 3, 2, 1, 0, … |
| False | only read data from the indicated channel. |

**Data Type**

Boolean

## Channel Property

Sets the selected analog input channel for continuous analog input operation.

**Syntax**

*object*.**Channel** [= number]

**Settings**

Channel number is from 0 to 15.

**Data Type**

Integer

## ClockSource Property

Returns/sets a value that determines the A/D clock source.

**Syntax**

*object*.**ClockSource** [= number]

**Settings**

| Value | Setting |
|---|---|
| 0 | Internal timer pacer |
| 1 | External signal trigger |

**Data Type**

Integer

## DoubleBufferMode Property

Returns/Sets a value that determines whether it is double-buffer mode or not.

### Syntax

*object*.**DoublebufferMode** [= boolean]

### Settings

| Value | Description |
|---|---|
| True | During interrupt analog input mode, the buffer is divided into two parts. When either part is full of data, it will get AiHalfReady event. |
| False | The buffer is single, when it is full of data, it will get AiComplete Event. |

### Data Type

Boolean

## NumOfScan Property

Sets a value that indicates the total number of scans to acquired.

### Syntax

*object*.**NumOfScan** [= number]

### Remarks

**Non-double-buffer mode**

This value multiply the total number of scan channels is the total number of A/D conversions to be performed. It must be the multiple of 2.

**Double-buffer-mode**

This value multiply the total number of scan channels is the size (in sample) of the circular buffer. It must be the multiple of 4.

### Data Type

Long

## OpenMode Property

Returns/sets a value that determines the mode of opening device .

### Syntax

*object*.**OpenMode** [= number]

### Settings

| Value | Description |
|---|---|
| 0 | Automatically open device when the control was created |
| 1 | Don't open device when the control was created. Must call the Open method to open device. |

(Manual)

### Data Type

Integer.

## ReturnType Property

Returns/sets a value that determines the return data type of analog input when AiComplete or AiHalfReady event would occur.

### Syntax

*object*.**ReturnType** [= number]

### Settings

| Value | Description |
|---|---|
| 0 | Scaled data only |
| 1 | Binary codes with channel only |
| 2 | Binary codes without channel only |
| 3 | Scaled data and binary codes with channel |
| 4 | Scaled data and binary codes without channel |
| 5 | No data return |

### Data Type

Integer

## ScanRate Property

Returns/sets a value that determines the scan rate (scans per second) of continuous analog input.

### Syntax

*object*.**ScanRate** [= number]

### Settings

The range of (ScanRate * total number of scan channels) must be between 0 and 110KHz.

### Remarks

This property is used only when the ClockSource Property is set to Internal timer pacer.

### Data Type

Double

## StreamToFile Property

Returns/sets a value that determines if the control is enabled the function of streaming data to disk file.

### Syntax

*object*.**StreamToFile** [= boolean]

### Settings

| Value | Description |
|---|---|
| False | Disable the function of streaming data to disk file |
| True | Enable the function of streaming data to disk file |

### Data Type

Boolean

# Methods

## *CheckContAI Method*

### Syntax

Function *object*.**CheckContAI(***AccessCount as long, stop as Integer***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*AccessCount as long*

Number of analog input data that has been transferred.

*stop as Integer*

Current state: 1 = stop, 0 = running

### Remarks

You can request DMA analog input status.

## *Open Method*

### Syntax

Function *object*.**Open (***[ErrMsgBox As Variant]***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*[ErrMsgBox As Variant]*

It is optional and boolean type. Default value is False

True, It will popup error message dialog box when the opening device is failed.

False, It will fire DAQError event instead of popping up dialog when the opening device is failed.

### Remarks

This method will be use when the OpenMode property is Manual.

### Note

In VC++, *ErrMsgBox* is a VARIANT of VT_I2**.**

## *ReadCounter0 Method*

### Syntax

Function *object*.**ReadCounter0 (***value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*value As Variant*

Returns the current counter value from PCI-9112 card.

### Remarks

You can read the value from Timer Counter #0.

### Note

In VC++, *value* is a VARIANT of VT_I4**.**

## *ReadDILine Method*

### Syntax

Function *object*.**ReadDILine (***line As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*line As Integer*

Selects one line number from the indicated port: from 0 to 15 (16-bit port).

*value As Variant*

Returns the data input from PCI-9112 card.

### Remarks

Users can read data from the indicated digital input line of PCI-9112 card.

### Note

In VC++, *value* is a VARIANT of VT_UI1**.**

## *ReadDIPort Method*

### Syntax

Function *object*.**ReadDIPort** (*value As Variant*) As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*value As Variant*

Sets one value buffer for retrieving 16-bit input data from PCI-9112 card.

### Remarks

Users can read 16-bit digital input data from PCI-9112 card.

### Note

In VC++, *value* is a VARIANT of VT_I4**.**

## *ReadSingleAI Method*

### Syntax

Function *object*.**ReadSingleAI** (*channel As Integer, range As Integer, value As Variant*) As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*channel As Integer*

Analog input channel number 0 ~ 15.

*range As Integer*

Range of analog input.

| Value | Description |
|---|---|
| 0 | ±5V |
| 1 | ±2.5V |
| 2 | ±1.25V |
| 3 | ±0.625V |
| 4 | 0~10V |
| 5 | 0~5V |
| 6 | 0~2.5V |
| 7 | 0~1.25V |
| 8 | ±10V |

*value As Variant*

The analog data read from analog input channel (already scaled to voltage).

### Remarks

You can read one data from one analog input channel.

### Note

In VC++, *value* is a VARIANT of VT_R8**.**

## *StartContAI Method*

### Syntax

Function *object*.**StartContAI** ( *[FileName as Variant]* ) As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*[FileName as String]*

It is optional and String type. Default value is NULL

FileName specified the file name of streaming data to disk.

### Remarks

You can use this method to start the continuous analog input function of PCI-9112 card. If the StreamToFile property is True then the data will be written to the file specified by FileName. Otherwise, the FileName parameter will be ignored.

The data file is written in binary format, with the lower byte first (little endian). Data type is “Binary codes with channel”. DAQBench provides a convenient tool DAQCvt to convert the binary file to the file format read easily. See DAQBench User’s Guide for the usage of the utility. If you want to handle the data by yourself, please refer to Appendix Data File Format for the file structure.

### Note

In VC++, FileName is a VARIANT of VT_BSTR.

## *StartCounter0 Method*

### Syntax

Function *object*.**StartCounter0** (*CtrMode As Integer, CtrValue As Variant, BinBcd As Integer*) As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*CtrMode As Integer*

0: Toggle output from low to high on terminal count

1: Programmable one-shot

2: Rate generator

3: Square wave rate generator

4: Software-triggered strobe

5: Hardware-triggered strobe

*CtrValue as Variant*

Set the start value to the indicated counter.

*binbcd As Integer*

0: 16-bit binary counter, 1: 4-decade BCD counter.

### Remarks

You can start the indicated counter to operate in the specified mode.

**Note**

In VC++, *CtrValue* is a VARIANT of VT_I4**.**

## *StopContAI Method*

**Syntax**

Function *object*.**StopContAI** () As Boolean

**Return Value**

True if the function is successful; otherwise False.

**Arguments**

None

**Remarks**

You can use this method to stop continuous analog input.

## *StopCounter0 Method*

**Syntax**

Function *object*.**StopCounter0**(*State As Integer*) As Boolean

**Return Value**

True if the function is successful; otherwise False.

**Arguments**

*State As Integer*

The logic state to which the timer counter0 is to be stopped, number: 0 or 1.

**Remarks**

You can stop the timer counter0 and set the specified state.

## *WriteDOPort Method*

**Syntax**

Function *object***.WriteDOPort (***value As Variant***)** As Boolean

**Return Value**

True if the function is successful; otherwise False.

**Arguments**

*value as Variant*

16-bit data that will be written to the digital output port.

**Remarks**

Users can write data to the PCI-9112 digital output port.

### Note

In VC++, *value* is a VARIANT of VT_I4**.**

## *WriteSingleAO Method*

### Syntax

Function *object***.WriteSingleAO (***channel As Integer, range As integer***,** *value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*channel As Integer*

Analog output channel number 0 ~ 1.

*range As Integer*

Range of analog output.

| Value | Description |
|---|---|
| 0 | 0 ~ 5V |
| 1 | 0 ~ 10V |

*value As Variant*

Sets one value to writing to analog output of PCI-9112 card, the range will depend on *range* parameter.

### Remarks

Users can write data to analog output channel of PCI-9112 card. The range value must be same with the D/A jumper setting of internal reference at PCI-9112 card.

### Note

In VC++, *value* is a VARIANT of VT_R8**.**

## *WriteSingleAOBinary Method*

### Syntax

Function *object***.WriteSingleAOBinary (***channel As Integer***,** *value As Integer***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*channel As Integer*

Analog output channel number 0 ~ 1.

*value As Integer*

Sets the binary value to writing to analog output of PCI-9112 card, the range is between 0 and 4095.

### Remarks

Users can write binary data to analog output channel of PCI-9112 card.

# Events

## *AiComplete Event*

### Syntax

sub *ControlName*_**AiComplete(** *ScaledData as Variant, BinaryCodes As Variant* **)**

### Arguments

*ScaledData as Variant*

The analog input data array that have been translated to the engineering data (voltage) according to AIRange property.

*BinaryCodes As Variant*

The analog input data array with digital format.

### Remarks

This event occurs when continuous analog input function is completed.

Whether *ScaledData* or *BinaryCodes* contains data depends on *ReturnType* property setting.

Although the auto-scan is in descending order, the data in ScaledData and BinaryData have been re-ordered to ascending order.

"*BinaryCodes with channel*" format:

Range 0 to +65535. The least significant 4 bits are channel number.

"*BinaryCodes without channel*" format:

Range 0 to +4095.

### Note

In VC++, *ScaledData* is a VARIANT of VT_ARRAY | VT_R4, *BinaryCodes* is a VARIANT of VT_ARRAY | VT_I4 (with channel) or VT_ARRAY | VT_I2 (without channel).

## *AiHalfReady Event*

### Syntax

sub *ControlName*_**AiHalfReady(** *ScaledData as Variant, BinaryCodes As Variant* **)**

### Arguments

*ScaledData as Variant*

The analog input data array that have been translated to the engineering data (voltage) according to AIRange property.

*BinaryCodes As Variant*

The analog input data array with digital format.

### Remarks

This event occurs when one half-buffer of the circular buffer is full at continuous analog input operation.

Whether *ScaledData* or *BinaryCodes* contains data depends on *ReturnType* property setting.

Although the auto-scan is in descending order, the data in ScaledData and BinaryData have been re-ordered to ascending order.

"*BinaryCodes with channel*" format:

Range 0 to +65535. The least significant 4 bits are channel number.

"*BinaryCodes without channel*" format:

Range 0 to +4095.

### Note

In VC++, *ScaledData* is a VARIANT of VT_ARRAY | VT_R4, *BinaryCodes* is a VARIANT of VT_ARRAY | VT_I4 (with channel) or VT_ARRAY | VT_I2 (without channel).

## *DAQError Event*

### Syntax

sub *ControlName*_**DAQError (** *ErrString As String* **)**

### Arguments

*ErrString As String*

The string of error reason

### Remarks

This event will occur when some error occur in control

# Pci9113 ActiveX Control

The PCI-9113 is a PCI-bus multi-function data acquisition card. The Pci9113 ActiveX control is a software component that provides the interface for users to control PCI-9113 card. The properties and methods allow users to perform analog I/O and digital I/O.

# Properties

## *DeviceName Property*

Returns/sets a string that indicates the device name of a PCI-9113 card that was defined in NuDAQCfg Utility.

**Syntax**

*object*.**DeviceName** [= string]

**Remarks**

The device name of NuDAQ cards must be defined in NuDAQCfg Utility. The indicated NuDAQ card by device name may be a local device or a remote device on remote machine. The device name of PCI9113 ActiveX Control must be specified to a PCI-9113 card defined in NuDAQCfg Utility.

**Data Type**

String

## *AIRange Property*

Sets a range for interrupt analog input.

**Syntax**

*object*.**AIRange** [= number]

**Settings**

| Value | Description |
|---|---|
| 0 | ±10V |
| 1 | ±5V |
| 2 | ±0.5V |
| 3 | ±0.05V |
| 4 | ±1V |
| 5 | ±0.1V |
| 6 | 0~10V |
| 7 | 0~1V |
| 8 | 0~0.1V |

### Data Type

Integer

## AutoScan Property

Returns/sets a value that determines whether the analog input operate in auto scan mode or not.

### Syntax

*object*.**AutoScan** [= boolean]

### Settings

| Value | Description |
|---|---|
| True | automatically scan channels to read analog data. The scan is in ascending order. For example, if Channel property set to 3, the channels scanned are 0, 1, 2, 3, 0, 1, 2, 3, … |
| False | only read data from the indicated channel. |

### Data Type

Boolean

## Channel Property

Sets the selected analog input channel for interrupt data transfer.

### Syntax

*object*.**Channel** [= number]

### Settings

Channel number is from 0 to 31.

### Data Type

Integer

## ClockSource Property

Returns/sets a value that determines the A/D clock source.

### Syntax

*object*.**ClockSource** [= number]

### Settings

| Value | Setting |
|---|---|
| 0 | Internal timer pacer |
| 1 | External signal trigger |

### Data Type

Integer

# DoubleBufferMode Property

Returns/Sets a value that determines whether it is double-buffer mode or not.

### Syntax

*object*.**DoublebufferMode** [= boolean]

### Settings

| Value | Description |
|---|---|
| True | During interrupt analog input mode, the buffer is divided into two parts. When either part is full of data, it will get AiHalfReady event. |
| False | The buffer is single, when it is full of data, it will get AiComplete Event. |

### Data Type

Boolean

# NumOfScan Property

Sets a value that indicates the total number of scans to acquired.

### Syntax

*object*.**NumOfScan** [= number]

### Remarks

**Non-double-buffer mode**

This value multiply the total number of scan channels is the total number of A/D conversions to be performed. It must be the multiple of 512.

**Double-buffer-mode**

This value multiply the total number of scan channels is the size (in sample) of the circular buffer. It must be the multiple of 1024.

### Data Type

Long

# OpenMode Property

Returns/sets a value that determines the mode of opening device .

### Syntax

*object*.**OpenMode** [= number]

### Settings

| Value | Description |
|---|---|
| 0 | Automatically open device when the control was created |
| 1 | Don't open device when the control was created. Must call the Open method to open device. (Manual) |

### Data Type

Integer.

## ReturnType Property

Returns/sets a value that determines the return data type of analog input when AiComplete or AiHalfReady event would occur.

### Syntax

*object*.**ReturnType** [= number]

### Settings

| Value | Description |
|---|---|
| 0 | Scaled data only |
| 1 | Binary codes with channel only |
| 2 | Binary codes without channel only |
| 3 | Scaled data and binary codes with channel |
| 4 | Scaled data and binary codes without channel |
| 5 | No data return |

### Data Type

Integer

## ScanRate Property

Returns/sets a value that determines the scan rate (scans per second) of continuous analog input.

### Syntax

*object*.**ScanRate** [= number]

### Settings

The range of (ScanRate * total number of scan channels) must be between 0 and 60KHz.

### Remarks

This property is used only when the ClockSource property os set to Internal timer pacer.

### Data Type

Double

## StreamToFile Property

Returns/sets a value that determines if the control is enabled the function of streaming data to disk file.

### Syntax

*object*.**StreamToFile** [= boolean]

### Settings

| Value | Description |
|---|---|
| False | Disable the function of streaming data to disk file |
| True | Enable the function of streaming data to disk file |

**Data Type**

Boolean

# Methods

## *CheckContAI Method*

**Syntax**

Function *object*.**CheckContAI(***AccessCount as long, stop as Integer***)** As Boolean

**Return Value**

True if the function is successful; otherwise False.

**Arguments**

*AccessCount as long*

Number of analog input data that has been transferred.

*stop as Integer*

Current state: 1 = stop, 0 = running

**Remarks**

You can request DMA analog input status.

## *Open Method*

**Syntax**

Function *object*.**Open (***[ErrMsgBox As Variant]***)** As Boolean

**Return Value**

True if the function is successful; otherwise False.

**Arguments**

*[ErrMsgBox As Variant]*

It is optional and boolean type. Default value is False

True, It will popup error message dialog box when the opening device is failed.

False, It will fire DAQError event instead of popping up dialog when the opening device is failed.

**Remarks**

This method will be use when the OpenMode property is Manual.

**Note**

In VC++, *ErrMsgBox* is a VARIANT of VT_I2**.**

# ReadCounter0 Method

### Syntax

Function *object*.**ReadCounter0 (***value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*value As Variant*

Returns the current counter value from PCI-9112 card.

### Remarks

You can read the value from Timer Counter #0.

### Note

In VC++, *value* is a VARIANT of VT_I4**.**

# ReadSingleAI Method

### Syntax

Function object.**ReadSingleAI** (*channel As Integer, range As Integer, value As Variant*) As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*channel As Integer*

Analog input channel number 0 ~ 31.

*range As Integer*

Range of analog input.

| Value | Description |
|---|---|
| 0 | ±10V |
| 1 | ±5V |
| 2 | ±0.5V |
| 3 | ±0.05V |
| 4 | ±1V |
| 5 | ±0.1V |
| 6 | 0~10V |
| 7 | 0~1V |
| 8 | 0~0.1V |

*value As Variant*

The analog data read from analog input channel (already scaled to voltage).

### Remarks

You can read one data from one analog input channel.

### Note

In VC++, *ErrMsgBox* is a VARIANT of VT_R8**.**

## *StartContAI Method*

### Syntax

Function object.**StartContAI** (**(***[FileName as Variant]*) As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*[FileName as String]*

It is optional and String type. Default value is NULL

FileName specified the file name of streaming data to disk.

### Remarks

You can use this method to start the continuous analog input function of PCI-9113 card. If the StreamToFile property is True then the data will be written to the file specified by FileName. Otherwise, the FileName parameter will be ignored.

The data file is written in binary format, with the lower byte first (little endian). Data type is “Binary codes with channel” (4 bytes). DAQBench provides a convenient tool DAQCvt to convert the binary file to the file format read easily. See DAQBench User’s Guide for the usage of the utility. If you want to handle the data by yourself, please refer to Appendix Data File Format for the file structure.

### Note

In VC++, FileName is a VARIANT of VT_BSTR.

## *StartCounter0 Method*

### Syntax

Function object.**StartCounter0**(*CtrMode As Integer, CtrValue As Variant, BinBcd As Integer*) As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*CtrMode As Integer*

0: Toggle output from low to high on terminal count

1: Programmable one-shot

2: Rate generator

3: Square wave rate generator

4: Software-triggered strobe

5: Hardware-triggered strobe

*CtrValue as Variant*

Set the start value to the indicated counter.

*binbcd As Integer*

0: 16-bit binary counter, 1: 4-decade BCD counter.

**Remarks**

You can start the timer counter0 to operate in the specified mode.

**Note**

In VC++, *CtrValue* is a VARIANT of VT_I4**.**

## *StopContAI Method*

**Syntax**

Function object.**StopContAI** () As Boolean

**Return Value**

True if the function is successful; otherwise False.

**Arguments**

None

**Remarks**

You can use this method to stop DMA analog input.

## *StopCounter0 Method*

**Syntax**

Function object.**StopCounter0**(*State As Integer*) As Boolean

**Return Value**

True if the function is successful; otherwise False.

**Arguments**

*State As Integer*

The logic state to which the timer counter0 is to be stopped, number: 0 or 1.

**Remarks**

You can stop the timer counter0 and set the specified state.

# Events

## *AiComplete Event*

### Syntax

sub *ControlName*_**AiComplete(** *ScaledData as Variant, BinaryCodes As Variant* **)**

### Arguments

*ScaledData as Variant*

The analog input data array that have been translated to the engineering data (voltage) according to AIRange property.

*BinaryCodes As Variant*

The analog input data array with digital format.

### Remarks

This event occurs when continuous analog input function is completed.

Whether *ScaledData* or *BinaryCodes* contains data depends on *ReturnType* property setting.

"*BinaryCodes with channel*" format:

B31… B21 C4 C3 C2 C1 C0 B15 … B12 D11 D10 ... D1 D0

where D11 D10 ... D0 : A/D converted data (range 0 to +4095)

C4 C3 C2 C1 C0 : converted channel no.

B31 ~ B21 & B15 ~ B12: don't care

"*BinaryCodes without channel*" format:

Range 0 to +4095.

### Note

In VC++, *ScaledData* is a VARIANT of VT_ARRAY | VT_R4, *BinaryCodes* is a VARIANT of VT_ARRAY | VT_I4 (with channel) or VT_ARRAY | VT_I2 (without channel).

## *AiHalfReady Event*

### Syntax

sub *ControlName*_**AiHalfReady(** *ScaledData as Variant, BinaryCodes As Variant* **)**

### Arguments

*ScaledData as Variant*

The analog input data array that have been translated to the engineering data (voltage) according to AIRange property.

*BinaryCodes As Variant*

The analog input data array with digital format.

### Remarks

This event occurs when one half-buffer of the circular buffer is full at continuous analog input operation.

Whether *ScaledData* or *BinaryCodes* contains data depends on *ReturnType* property setting.

"*BinaryCodes with channel*" format:

B31… B21 C4 C3 C2 C1 C0 B15 … B12 D11 D10 ... D1 D0

where D11 D10 ... D0 : A/D converted data (range 0 to +4095)

C4 C3 C2 C1 C0 : converted channel no.

B31 ~ B21 & B15 ~ B12: don’t care

“*BinaryCodes without channel*” format:

Range 0 to +4095.

### Note

In VC++, *ScaledData* is a VARIANT of VT_ARRAY | VT_R4, *BinaryCodes* is a VARIANT of VT_ARRAY | VT_I4 (with channel) or VT_ARRAY | VT_I2 (without channel).

## *DAQError Event*

### Syntax

sub *ControlName*_**DAQError (** *ErrString As String* **)**

### Arguments

*ErrString As String*

The string of error reason

### Remarks

This event will occur when some error occur in control

# Pci9114 ActiveX Control

The PCI-9114 is a PCI-bus multi-function data acquisition card. The Pci9114 ActiveX control is a software component that provides the interface for users to control PCI-9114 card. The properties and methods allow users to perform analog I/O and digital I/O.

# Properties

## *DeviceName Property*

Returns/sets a string that indicates the device name of a PCI-9114 card that was defined in NuDAQCfg Utility.

**Syntax**

*object*.**DeviceName** [= string]

**Remarks**

The device name of NuDAQ cards must be defined in NuDAQCfg Utility. The indicated NuDAQ card by device name may be a local device or a remote device on remote machine. The device name of PCI9114 ActiveX Control must be specified to a PCI-9114 card defined in NuDAQCfg Utility.

**Data Type**

String

## *AIRange Property*

Sets a range for interrupt analog input.

**Syntax**

*object*.**AIRange** [= number]

**Settings**

PCI-9114HG:

| Value | Description |
|---|---|
| 0 | ±10V |
| 1 | ±1V |
| 2 | ±100mV |
| 3 | ±10mV |

PCI-9114DG:

| Value | Description |
|---|---|
| 0 | ±10V |
| 1 | ±5V |

| | |
|---|---|
| 2 | ±2.5V |
| 3 | ±1.25V |

**Data Type**

Integer

## AutoScan Property

Returns/sets a value that determines whether the analog input operate in auto scan mode or not.

**Syntax**

*object*.**AutoScan** [= boolean]

**Settings**

| Value | Description |
|---|---|
| True | automatically scan channels to read analog data. The scan is in ascending order. For example, if Channel property set to 3, the channels scanned are 0, 1, 2, 3, 0, 1, 2, 3, … |
| False | only read data from the indicated channel. |

**Data Type**

Boolean

## CardType Property

Returns/sets a value that indicates the card type for programming Pci9114 ActiveX control.

**Syntax**

*object*.**CardType** [= number]

**Settings**

| Number | Card Type | Analog input range |
|---|---|---|
| 1 | PCI-9114DG | ±10V, ±5V, ±2.5V, ±1.25V |
| 2 | PCI-9114 HG | ±10V, ±1V, ±100mV, ±10mV |

**Remarks**

This property will be automatically set value when the device name be specified and the device information be retrieved successfully.

**Data Type**

Integer

## Channel Property

Sets the selected analog input channel for continuous analog input.

**Syntax**

*object*.**Channel** [= number]

### Settings

Channel number is from 0 to 31.

### Data Type

Integer

## *ClockSource Property*

Returns/sets a value that determines the A/D clock source.

### Syntax

*object*.**ClockSource** [= number]

### Settings

| Value | Setting |
|---|---|
| 0 | Internal timer pacer |
| 1 | External signal trigger |

### Data Type

Integer

## *DoubleBufferMode Property*

Returns/Sets a value that determines whether it is double-buffer mode or not.

### Syntax

*object*.**DoublebufferMode** [= boolean]

### Settings

| Value | Description |
|---|---|
| True | During interrupt analog input mode, the buffer is divided into two parts. When either part is full of data, it will get AiHalfReady event. |
| False | The buffer is single, when it is full of data, it will get AiComplete Event. |

### Data Type

Boolean

## *NumOfScan Property*

Sets a value that indicates the total number of scans to acquired.

### Syntax

*object*.**NumOfScan** [= number]

### Remarks

**Non-double-buffer mode**

This value multiply the total number of scan channels is the total number of A/D conversions to be performed.

It must be the multiple of 512.

**Double-buffer-mode**

This value multiply the total number of scan channels is the size (in sample) of the circular buffer. It must be the multiple of 1024.

### Data Type

Long

## *OpenMode Property*

Returns/sets a value that determines the mode of opening device .

### Syntax

*object*.**OpenMode** [= number]

### Settings

| Value | Description |
|---|---|
| 0 | Automatically open device when the control was created |
| 1 | Don't open device when the control was created. Must call the Open method to open device. (Manual) |

### Data Type

Integer

## *ReturnType Property*

Returns/sets a value that determines the return data type of analog input when AiComplete or AiHalfReady event would occur.

### Syntax

*object*.**ReturnType** [= number]

### Settings

| Value | Description |
|---|---|
| 0 | Scaled data only |
| 1 | Binary codes with channel only |
| 2 | Binary codes without channel only |
| 3 | Scaled data and binary codes with channel |
| 4 | Scaled data and binary codes without channel |
| 5 | No data return |

### Data Type

Integer

## *ScanRate Property*

Returns/sets a value that determines the scan rate (scans per second) of continuous analog input.

### Syntax

*object*.**ScanRate** [= number]

### Settings

The range of (ScanRate * total number of scan channels) must be between 0 and 100KHz.

### Remarks

This property is used only when the ClockSource property is set to Internal timer pacer.

### Data Type

Double

## *StreamToFile Property*

Returns/sets a value that determines if the control is enabled the function of streaming data to disk file.

### Syntax

*object*.**StreamToFile** [= boolean]

### Settings

| Value | Description |
|---|---|
| False | Disable the function of streaming data to disk file |
| True | Enable the function of streaming data to disk file |

### Data Type

Boolean

# Methods

## *CheckContAI Method*

### Syntax

Function *object*.**CheckContAI(***AccessCount as long, stop as Integer***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*AccessCount as long*

Number of analog input data that has been transferred.

*stop as Integer*

Current state: 1 = stop, 0 = running

### Remarks

You can request DMA analog input status.

## Open Method

### Syntax

Function *object*.**Open (***[ErrMsgBox As Variant]***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*[ErrMsgBox As Variant]*

It is optional and boolean type. Default value is False

True, It will popup error message dialog box when the opening device is failed.

False, It will fire DAQError event instead of popping up dialog when the opening device is failed.

### Remarks

This method will be use when the OpenMode property is Manual.

### Note

In VC++, *ErrMsgBox* is a VARIANT of VT_I2**.**

## ReadCounter0 Method

### Syntax

Function *object*.**ReadCounter0 (***value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*value As Variant*

Returns the current counter value from PCI-9114 card.

### Remarks

You can read the value from Timer Counter #0.

### Note

In VC++, *value* is a VARIANT of VT_I4**.**

## ReadDILine Method

### Syntax

Function *object*.**ReadDILine (***line As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*line As Integer*

Selects one line number from the indicated port: from 0 to 15 (16-bit port).

*value As Variant*

Returns the data input from PCI-9114 card.

### Remarks

Users can read data from the indicated digital input line of PCI-9114 card.

### Note

In VC++, *value* is a VARIANT of VT_UI1**.**

## *ReadDIPort Method*

### Syntax

Function object.**ReadDIPort** (*value As Variant*) As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*value As Variant*

Sets one value buffer for retrieving 16-bit input data from PCI-9114 card.

### Remarks

Users can read 16-bit digital input data from PCI-9114 card.

### Note

In VC++, *value* is a VARIANT of VT_I4**.**

## *ReadSingleAI Method*

### Syntax

Function object.**ReadSingleAI** (*channel As Integer, range As Integer, value As Variant*) As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*channel As Integer*

Analog input channel number 0 ~ 31.

*range As Integer*

Range of analog input.

PCI-9114HG:

| Value | Description |
|---|---|
| 0 | ±10V |
| 1 | ±1V |

| | |
|---|---|
| 2 | ±100mV |
| 3 | ±10mV |

PCI-9114DG:

| Value | Description |
|---|---|
| 0 | ±10V |
| 1 | ±5V |
| 2 | ±2.5V |
| 3 | ±1.25V |

*value As Variant*

The analog data read from analog input channel (already scaled to voltage).

**Remarks**

You can read one data from one analog input channel.

**Note**

In VC++, *value* is a VARIANT of VT_R8**.**

# *StartContAI Method*

**Syntax**

Function object.**StartContAI (***[FileName as Variant]*) As Boolean

**Return Value**

True if the function is successful; otherwise False.

**Arguments**

*[FileName as String]*

It is optional and String type. Default value is NULL

FileName specified the file name of streaming data to disk.

**Remarks**

You can use this method to start the continuous analog input function of PCI-9114 card. If the StreamToFile property is True then the data will be wirrer to the file specified by FileName. Otherwise, the FileName parameter will be ignored.

The data file is written in binary format, with the lower byte first (little endian). Data type is “Binary codes with channel” (4 bytes). DAQBench provides a convenient tool DAQCvt to convert the binary file to the file format read easily. See DAQBench User’s Guide for the usage of the utility. If you want to handle the data by yourself, please refer to Appendix Data File Format for the file structure.

**Note**

In VC++, FileName is a VARIANT of VT_BSTR.

# *StartCounter0 Method*

### Syntax

Function object.**StartCounter0**(*CtrMode As Integer, CtrValue As Variant, BinBcd As Integer*) As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*CtrMode As Integer*

0: Toggle output from low to high on terminal count

1: Programmable one-shot

2: Rate generator

3: Square wave rate generator

4: Software-triggered strobe

5: Hardware-triggered strobe

*CtrValue as Variant*

Set the start value to the indicated counter.

*binbcd As Integer*

0: 16-bit binary counter, 1: 4-decade BCD counter.

### Remarks

You can start the timer counter0 to operate in the specified mode.

### Note

In VC++, *CtrValue* is a VARIANT of VT_I4**.**

## *StopContAI Method*

### Syntax

Function object.**StopContAI** () As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

None

### Remarks

You can use this method to stop continuous analog input.

## *StopCounter0 Method*

### Syntax

Function object.**StopCounter0**(*State As Integer*) As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*State As Integer*

The logic state to which the timer counter0 is to be stopped, number: 0 or 1.

### Remarks

You can stop the timer counter0 and set the specified state.

## *WriteDOPort Method*

### Syntax

Function *object*.**WriteDOPort (***value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*value as Variant*

16-bit data that will be written to the digital output port.

### Remarks

Users can write data to the PCI-9114 digital output port.

### Note

In VC++, *value* is a VARIANT of VT_I4**.**

# Events

## *AiComplete Event*

### Syntax

sub *ControlName*_**AiComplete(** *ScaledData as Variant, BinaryCodes As Variant* **)**

### Arguments

*ScaledData as Variant*

The analog input data array that have been translated to the engineering data (voltage) according to AIRange property.

*BinaryCodes As Variant*

The analog input data array with digital format.

### Remarks

This event occurs when continuous analog input function is completed.

Whether *ScaledData* or *BinaryCodes* contains data depends on *ReturnType* property setting.

"*BinaryCodes with channel*" format:

B31… B21 C4 C3 C2 C1 C0  D15 D14... D1 D0

where D15 D14... D1 D0 : A/D converted data (signed, range -32768 to +32767)

C4 C3 C2 C1 C0 : converted channel no.

B31 ~ B21: don't care

"*BinaryCodes without channel*" format:

Range -32768 to +32767.

**Note**

In VC++, *ScaledData* is a VARIANT of VT_ARRAY | VT_R4, *BinaryCodes* is a VARIANT of VT_ARRAY | VT_I4 (with channel) or VT_ARRAY | VT_I2 (without channel).

## *AiHalfReady Event*

**Syntax**

sub *ControlName*_**AiHalfReady(** *ScaledData as Variant, BinaryCodes As Variant* **)**

**Arguments**

*ScaledData as Variant*

The analog input data array that have been translated to the engineering data (voltage) according to AIRange property.

*BinaryCodes As Variant*

The analog input data array with digital format.

**Remarks**

This event occurs when one half-buffer of the circular buffer is full at continuous analog input operation.

Whether *ScaledData* or *BinaryCodes* contains data depends on *ReturnType* property setting.

"*BinaryCodes with channel*" format:

B31… B21 C4 C3 C2 C1 C0 D15 D14... D1 D0

where D15 D14... D1 D0 : A/D converted data (signed, range -32768 to +32767)

C4 C3 C2 C1 C0 : converted channel no.

B31 ~ B21: don't care

"*BinaryCodes without channel*" format:

Range -32768 to +32767.

**Note**

In VC++, *ScaledData* is a VARIANT of VT_ARRAY | VT_R4, *BinaryCodes* is a VARIANT of VT_ARRAY | VT_I4 (with channel) or VT_ARRAY | VT_I2 (without channel).

## *DAQError Event*

**Syntax**

sub *ControlName*_**DAQError (** *ErrString As String* **)**

**Arguments**

*ErrString As String*

The string of error reason

### Remarks

This event will occur when some error occur in control

# Pci9116 ActiveX Control

The cPCI-9116 is a CompactPCI multi-function data acquisition card. The Pci9116 ActiveX control is a software component that provides the interface for users to control cPCI-9116 card. The properties and methods allow users to perform analog I/O and digital I/O.

# Properties

## *DeviceName Property*

Returns/sets a string that indicates the device name of a PCI-9116 card that is defined in NuDAQCfg Utility.

**Syntax**

*object*.**DeviceName** [= string]

**Remarks**

The device name of NuDAQ cards must be defined in NuDAQCfg Utility. The indicated NuDAQ card by device name may be a local device or a remote device on remote machine. The device name of PCI9116 ActiveX Control must be specified to a PCI-9116 card defined in NuDAQCfg Utility.

**Data Type**

String

## *AutoMode Property*

Returns/sets a value that determines whether the scan order of channel was contiguous or discrete.

**Syntax**

*object*.**AutoMode** [= number]

**Settings**

| Value | Description |
|---|---|
| 0 | Contiguous: The scan order of channels is from *StartChannel* to *EndChannel*. For example, if *StartChannel* set to 1 and *EndChannel*. Set to 3, the channels scanned are 1, 2, 3, 1, 2, 3, … |
| 1 | Discrete: The scan order of channel can be customized by users. You can customize the *DiscreteChannel* array and *DiscreteRange* array and *DiscreteNum*. |

**Data Type**

Integer

## *ClockSource Property*

Returns/sets a value that determines the A/D clock source.

### Syntax

*object*.**ClockSource** [= number]

### Settings

| Value | Setting |
|---|---|
| 0 | Internal timer pacer |
| 1 | External signal trigger |

### Data Type

Integer

## CommonMode Property

Returns/sets a value that determines the common mode.

### Syntax

*object*.**CommonMode** [= number]

### Settings

| Value | Setting |
|---|---|
| 0 | Local Ground of PCI-9116 |
| 1 | User defined Common Mode |

### Data Type

Integer

## DelaySource Property

Returns/sets a value that determines delay source.

### Syntax

*object*.**CommonMode** [= number]

### Settings

| Value | Setting |
|---|---|
| 0 | Sample Interval |
| 1 | Internal time base |

### Data Type

Integer

## DiscreteChannel Property

Returns/sets a array value that indicates channels for discrete scanning of Pci9116 ActiveX control.

### Syntax

*object*.**DiscreteChannel(***index As Integer***)** [= number]

**Parameter**

*index As Integer*

the index of Discrete channel array

**Settings**

Channel number is from 0 to 63.

**Data Type**

Integer

# *DiscreteRange Property*

Returns/sets a array value that indicates ranges of channel for discrete scanning of Pci9116 ActiveX control.

**Syntax**

*object*.**DiscreteRange(***index As Integer***)** [= number]

**Parameter**

*index As Integer*

the index of Discrete range array

**Settings**

Polarity is Bipolar

| Value | Description |
|---|---|
| 0 | ±5V |
| 1 | ±2.5V |
| 2 | ±1.25V |
| 3 | ±0.625V |

Polarity is Unipolar

| Value | Description |
|---|---|
| 0 | 0~10V |
| 1 | 0~5V |
| 2 | 0~2.5V |
| 3 | 0~1.25V |

**Data Type**

Integer

# *DiscreteNum Property*

Returns/sets a value that indicates the number of Discrete channel array.

**Syntax**

*object*.**DiscreteNum** [= number]

### Settings

Number of Discrete channel array from 1 to 512

### Data Type

Integer

## *DoubleBufferMode Property*

Returns/Sets a value that determines whether it is double-buffer mode or not.

### Syntax

*object*.**DoublebufferMode** [= boolean]

### Settings

| Value | Description |
|---|---|
| True | During interrupt analog input mode, the buffer is divided into two parts. When either part is full of data, it will get AiHalfReady event. |
| False | The buffer is single, when it is full of data, it will get AiComplete Event. |

### Data Type

Boolean

## *EndChannel Property*

Returns/sets a value that indicates end channel of contiguous scanning of Pci9116 ActiveX control.

### Syntax

*object*.**EndChannel** [= number]

### Settings

channel number is from 0 to 63.

### Data Type

Integer

## *GateSource Property*

Returns/sets a value that determines the counter gate source.

### Syntax

*object*.**GateSource** [= number]

### Settings

| Value | Setting |
|---|---|
| 0 | gate is controlled by software |
| 1 | gate is controlled by GP_TC_GATE pin |

### Data Type

Integer

## GPTCEnable Property

Enabled/Disables GPTC(general purpose time/counter).

### Syntax

*object*.**GPTCMode** [= boolean]

### Settings

| Value | Setting |
|---|---|
| True | Enables GPTC |
| False | Disables GPTC |

### Data Type

Boolean

## GPTCMode Property

Returns/sets a value that determines GPTC(general purpose time/counter) mode.

### Syntax

*object*.**GPTCMode** [= number]

### Settings

| Value | Setting |
|---|---|
| 0 | General counter |
| 1 | Generation of pulse |

### Data Type

Integer

## GPTCSource Property

Returns/sets a value that determines GPTC(general purpose time/counter) source.

### Syntax

*object*.**GPTCSource** [= number]

### Settings

| Value | Setting |
|---|---|
| 0 | internal time base |
| 1 | external time base from GP_TC_CLK pin |

### Data Type

Integer

## *InputMode Property*

Returns/sets a value that determines the mode of analog input.

### Syntax

*object*.**InputMode** [= number]

### Settings

| Value | Description |
|---|---|
| 0 | Single ended |
| 1 | Differential |

### Data Type

Integer

## *NumOfScan Property*

Sets a value that indicates the total number of scans to acquired.

### Syntax

*object*.**NumOfScan** [= number]

### Remarks

**Non-double-buffer mode**

This value multiply the total number of scan channels is the total number of A/D conversions to be performed. It must be the multiple of 2.

**Double-buffer-mode**

This value multiply the total number of scan channels is the size (in sample) of the circular buffer. It must be the multiple of 4.

### Data Type

Long

## *MCount Property*

The counter value of MCounter . This argument is only valid for Pre-trigger and Middle trigger mode

### Syntax

*object*.**MCount** [= number]

### Settings

The value range is 0~65535.

### Data Type

Long

## *MCounterEnable Property*

Enables/Disables MCounter.

### Syntax

*object*.**MCounter** [= boolean]

### Settings

| Value | Setting |
|---|---|
| True | Enables Mcounter and the trigger signal is ignore before M terminal count is reached(see Mcount). |
| False | Disables MCounter |

### Data Type

Boolean

## *OpenMode Property*

Returns/sets a value that determines the mode of opening device .

### Syntax

*object*.**OpenMode** [= number]

### Settings

| Value | Description |
|---|---|
| 0 | Automatically open device when the control was created |
| 1 | Don't open device when the control was created. Must call the Open method to open device. (Manual) |

### Data Type

Integer.

## *Polarity Property*

Returns/sets a value that determines the polarity type of analog input.

### Syntax

*object*.**Polarity** [= number]

### Settings

| Value | Description |
|---|---|
| 0 | Bipolar |
| 1 | Unipolar |

### Remarks

Polarity is Bipolar:

Input range is ±5V, ±2.5V, ±1.25V, ±0.625V

Polarity is Unipolar:

Input range is 0~10V, 0~5V, 0~2.5V, 0~1.25V

### Data Type

Integer

## PostCount Property

Returns/sets a value that determines the number of sample data that will be acquired after the trigger event happens.

### Syntax

*object*.**PostCount** [= number]

### Settings

The value range is 0~65535.

### Data Type

Long

## Range Property

Sets a range for interrupt analog input at contiguous scanning of Pci9116.

### Syntax

*object*.**Range** [= number]

### Settings

Polarity is Bipolar

| Value | Description |
|---|---|
| 0 | ±5V |
| 1 | ±2.5V |
| 2 | ±1.25V |
| 3 | ±0.625V |

Polarity is Unipolar

| Value | Description |
|---|---|
| 0 | 0~10V |
| 1 | 0~5V |
| 2 | 0~2.5V |
| 3 | 0~1.25V |

### Data Type

Integer

## RetrigCount Property

The accepted trigger times in an acquisition. This argument is only valid for Delay trigger and Post trigger

mode.

### Syntax

*object*.**RetrigCount** [= number]

### Settings

The value range is 0~65535.

### Data Type

Long

## *RetrigEnable Property*

Returns/Sets re-trigger.

### Syntax

*object*.**RetrigEnable** [= boolean]

### Settings

| Value | Setting |
|---|---|
| True | Enables re-trigger |
| False | Disables re-trigger |

### Data Type

Boolean

## *ReturnType Property*

Returns/sets a value that determines the return data type of analog input when AiComplete or AiHalfReady event would occur.

### Syntax

*object*.**ReturnType** [= number]

### Settings

| Value | Description |
|---|---|
| 0 | Scaled data only |
| 1 | Binary codes only |
| 2 | Scaled data and binary codes |
| 3 | No data return |

### Data Type

Integer

## *SampleRate Property*

Returns/sets a value that determines the sample rate (the frequency of each A/D conversion within a scan

sequence) of contiguous analog input

### Syntax

*object*.**SampleRate** [= number]

### Settings

The range is between 384 and 200KHz.

### Remarks

This property is used only when the ClockSource property is set to Internal timer pacer.

### Data Type

Double

## ScanRate Property

Returns/sets a value that determines the scan rate (scans per second) of contiguous analog input

### Syntax

*object*.**ScanRate** [= number]

### Settings

The range of (ScanRate * total number of scan channels) is between 2 and 200KHz.

### Remarks

This property is used only when the ClockSource property is set to Internal timer pacer.

### Data Type

Double

## StartChannel Property

Returns/sets a value that indicates start channel of contiguous scanning of Pci9116 ActiveX control.

### Syntax

*object*.**StartChannel** [= number]

### Settings

channel number is from 0 to 63.

### Data Type

Integer

## StreamToFile Property

Returns/sets a value that determines if the control is enabled the function of streaming data to disk file.

### Syntax

*object*.**StreamToFile** [= boolean]

### Settings

| Value | Description |
|---|---|
| False | Disable the function of streaming data to disk file |
| True | Enable the function of streaming data to disk file |

**Data Type**

Boolean

## *TriggerMode Property*

Returns/sets a value that determines trigger mode of analog input

**Syntax**

*object*.**TriggerMode** [= number]

**Settings**

| Value | Description |
|---|---|
| 0 | Disable |
| 1 | Post trigger |
| 2 | Delay trigger |
| 3 | Pre trigger |
| 4 | Middle trigger |

**Remarks**

Use post trigger acquisition in application where you want to collect data after the start condition.

Use delay trigger acquisition in application where you want to collect data after the start condition plus specified time.

Use pre trigger acquisition in application where you want to collect data before the start condition.

Use middle trigger acquisition in application where you want to collect data before and after a specific trigger event.

**Data Type**

Integer

## *TriggerPolarity Property*

Returns/sets a value that determines the active type of digital trigger.

**Syntax**

*object*.**TriggerPolarity** [= number]

**Settings**

| Value | Description |
|---|---|
| 0 | Positive |
| 1 | Negative |

**Remarks**

TriggerPolarity is valid only when TriggerMode is not disable(software trigger).

### Data Type

Integer

## *UpdownSource Property*

Returns/sets a value that determines the counter up-down source.

### Syntax

*object*.**UpdownSource** [= number]

### Settings

| Value | Setting |
|---|---|
| 0 | Up/Down controlled by software |
| 1 | Up/Down controlled by GP_TC_UPDN pin |

### Data Type

Integer

## *UpdownControl Property*

Returns/sets a value that determines the counter up-down control source.

### Syntax

*object*.**UpdownControl** [= number]

### Settings

| Value | Setting |
|---|---|
| 0 | counting direction is down |
| 1 | counting direction is up |

### Data Type

Integer

# Methods

## *CheckContAI Method*

### Syntax

Function *object*.**CheckContAI(***AccessCount as long, stop as Integer***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*AccessCount as long*

Number of analog input data that has been transferred.

*stop as Integer*

Current state: 1 = stop, 0 = running

### Remarks

You can request DMA analog input status.

## Open Method

### Syntax

Function *object*.**Open (***[ErrMsgBox As Variant]***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*[ErrMsgBox As Variant]*

It is optional and boolean type. Default value is False

True, It will popup error message dialog box when the opening device is failed.

False, It will fire DAQError event instead of popping up dialog when the opening device is failed.

### Remarks

This method will be use when the OpenMode property is Manual.

### Note

In VC++, *ErrMsgBox* is a VARIANT of VT_I2**.**

## ReadDILine Method

### Syntax

Function *object*.**ReadDILine (***line As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*line As Integer*

Selects one line number from the indicated port: from 0 to 7 (8-bit port).

*value As Variant*

Returns the data input from PCI-9116 card.

### Remarks

Users can read data from the indicated digital input line of PCI-9116 card.

### Note

In VC++, *value* is a VARIANT of VT_UI1**.**

# ReadDIPort Method

### Syntax

Function object.**ReadDIPort** (*value As Variant*) As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*value As Variant*

Sets one value buffer for retrieving 8-bit input data from PCI-9116 card.

### Remarks

Users can read 8-bit digital input data from PCI-9116 card.

### Note

In VC++, *value* is a VARIANT of VT_I4**.**

# ReadSingleAI Method

### Syntax

Function object.**ReadSingleAI** (*channel As Integer, range As Integer, value As Variant*) As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*channel As Integer*

Analog input channel number 0 ~ 63.

*range As Integer*

Range of analog input.

Polarity is Bipolar

| Value | Description |
|---|---|
| 0 | ±5V |
| 1 | ±2.5V |
| 2 | ±1.25V |
| 3 | ±0.625V |

Polarity is Unipolar

| Value | Description |
|---|---|
| 0 | 0~10V |
| 1 | 0~5V |
| 2 | 0~2.5V |
| 3 | 0~1.25V |

*value As Variant*

The analog data read from analog input channel (already scaled to voltage).

### Remarks

You can read one data from one analog input channel.

### Note

In VC++, *value* is a VARIANT of VT_R8**.**

## *StartContAI Method*

### Syntax

Function object.**StartContAI** (*[FileName as Variant]*) As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*[FileName as String]*

It is optional and String type. Default value is NULL

FileName specified the file name of streaming data to disk.

### Remarks

You can use this method to start the continuous analog input function of PCI-9116 card. If the StreamToFile property is True then the data will be written to the file specified by FileName. Otherwise, the FileName parameter will be ignored.

The data file is written in binary format, with the lower byte first (little endian). Data type is “Binary codes with channel”. DAQBench provides a convenient tool DAQCvt to convert the binary file to the file format read easily. See DAQBench User’s Guide for the usage of the utility. If you want to handle the data by yourself, please refer to Appendix Data File Format for the file structure.

### Note

In VC++, FileName is a VARIANT of VT_BSTR.

## *StopContAI Method*

### Syntax

Function object.**StopContAI** () As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

None

### Remarks

You can use this method to stop DMA analog input.

## *WriteDOLine Method*

### Syntax

Function *object*.**WriteDOLine(***line As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*line As Integer*

Selects one line number from the indicated port from 0 to 7.

*value As Variant*

Sets 0 or 1 to the indicated line.

### Note

In VC++, *value* is a VARIANT of VT_I4.

## *WriteDOPort Method*

### Syntax

Function *object***.WriteDOPort (***value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*value as Variant*

8-bit data that will be written to the digital output port.

### Remarks

Users can write data to the PCI-9116 digital output port.

### Note

In VC++, *value* is a VARIANT of VT_I4**.**

## *ReadBackDOLine Method*

Reads back data from the indicated digital output line of the digital output port.

### Syntax

Function *object*.**ReadBackDOLine(***line As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*line As Integer*

Selects one line number from the indicated port from 0 to 7.

*value As Variant*

Data that is read back from the indicated line.

### Note

In VC++, *value* is a VARIANT of VT_UI1.

## ReadBackDOPort Method

Reads back data from the digital output port.

### Syntax

Function *object*.**ReadBackDOPort (***value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*value As Variant*

Data that is read back from the port.

### Note

In VC++, *value* is a VARIANT of VT_I4.

## ReadGCTR0 Method

Reads the GCTR (global counter) value.

### Syntax

Function *object*.**ReadGCTR0(***value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*value As Variant*

the counter value of the specified general-purpose timer/counter.

Range: 0 through 65536

### Note

In VC++, *value* is a VARIANT of VT_I4.

## StartGCTR0 Method

Start the GCTR (global counter) value.

### Syntax

Function *object*.**StartGCTR0(***value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*value As Variant*

the counter value of the specified general-purpose timer/counter.

Range: 0 through 65536

### Note

In VC++, *value* is a VARIANT of VT_I4.

## *StopGCTR0 Method*

Stops the GCTR (global counter) .

### Syntax

Function *object*.**StopGCTR0** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

None

# Events

## *AiComplete Event*

### Syntax

sub *ControlName*_**AiComplete(** *ScaledData as Variant, BinaryCodes As Variant* **)**

### Arguments

*ScaledData as Variant*

The analog input data array that have been translated to the engineering data (voltage) according to AIRange property.

*BinaryCodes As Variant*

The analog input data array with digital format.

### Remarks

This event occurs when continuous analog input function is completed.

Whether *ScaledData* or *BinaryCodes* contains data depends on *ReturnType* property setting.

"*BinaryCodes*" format:

Range -32768 to +32767.

**Note**

In VC++, *ScaledData* is a VARIANT of VT_ARRAY | VT_R4, *BinaryCodes* is a VARIANT of VT_ARRAY | VT_I4 (with channel) or VT_ARRAY | VT_I2 (without channel).

# *AiHalfReady Event*

**Syntax**

sub *ControlName*_**AiHalfReady(** *ScaledData as Variant, BinaryCodes As Variant* **)**

**Arguments**

*ScaledData as Variant*

The analog input data array that have been translated to the engineering data (voltage) according to AIRange property.

*BinaryCodes As Variant*

The analog input data array with digital format.

**Remarks**

This event occurs when one half-buffer of the circular buffer is full at continuous analog input operation.

Whether *ScaledData* or *BinaryCodes* contains data depends on *ReturnType* property setting.

"*BinaryCodes*" format:

Range -32768 to +32767.

**Note**

In VC++, *ScaledData* is a VARIANT of VT_ARRAY | VT_R4, *BinaryCodes* is a VARIANT of VT_ARRAY | VT_I4 (with channel) or VT_ARRAY | VT_I2 (without channel).

# *DAQError Event*

**Syntax**

sub *ControlName*_**DAQError (** *ErrString As String* **)**

**Arguments**

*ErrString As String*

The string of error reason

**Remarks**

This event will occur when some error occur in control

# Pci9118 ActiveX Control

The PCI-9118 is a PCI-bus multi-function data acquisition card. The Pci9118 ActiveX control is a software component that provides the interface for users to control PCI-9118 card. The properties and methods allow users to perform analog I/O and digital I/O.

# Properties

## *DeviceName Property*

Returns/sets a string that indicates the device name of a PCI-9118 card that is defined in NuDAQCfg Utility.

**Syntax**

*object*.**DeviceName** [= string]

**Remarks**

The device name of NuDAQ cards must be defined in NuDAQCfg Utility. The indicated NuDAQ card by device name may be a local device or a remote device on remote machine. The device name of PCI9118 ActiveX Control must be specified to a PCI-9118 card defined in NuDAQCfg Utility.

**Data Type**

String

## *AutoMode Property*

Returns/sets a value that determines whether the scan order of channel was contiguous or discrete.

**Syntax**

*object*.**AutoMode** [= number]

**Settings**

| Value | Description |
|---|---|
| 0 | Contiguous: The scan order of channels is from *StartChannel* to *EndChannel*. For example, if *StartChannel* set to 1 and *EndChannel*. Set to 3, the channels scanned are 1, 2, 3, 1, 2, 3, … |
| 1 | Discrete: The scan order of channel can be customized by users. You can customize the *DiscreteChannel* array and *DiscreteRange* array and *DiscreteNum*. |

**Data Type**

Integer

## *BurstCount Property*

Returns/sets a value that determines the valid burst number when Burst Mode is Enable.

### Syntax

*object*.**BurstCount** [= number]

### Settings

The value range is 0~65535.

### Data Type

Long

## *BurstMode Property*

Returns/sets a value that determines the A/D burst mode.

### Syntax

*object*.**BurstMode** [= number]

### Settings

| Value | Description |
|---|---|
| 0 | Disable |
| 1 | Burst |
| 2 | Burst with sample & hold |

### Remarks

This property is for you to accurately control the period between conversions of individual channels in a scan, and the period between conversions of the entire scan.

### Data Type

Integer

## *CardType Property*

Returns/sets a value that indicates the card type for programming Pci9118 ActiveX control.

### Syntax

*object*.**CardType** [= number]

### Settings

| Number | Card Type | Resolution |
|---|---|---|
| 1 | PCI-9118DG | 12-bits |
| 2 | PCI-9118HG | 12-bits |
| 3 | PCI-9118HR | 16-bits |

### Remarks

This property will be automatically set when the device name be specified and the device information be retrieved successfully.

### Data Type

Integer

## ClockSource Property

Returns/sets a value that determines the A/D clock source.

### Syntax

*object*.**ClockSource** [= number]

### Settings

| Value | Setting |
|---|---|
| 0 | Internal timer pacer |
| 1 | External signal trigger |

### Data Type

Integer

## DiscreteChannel Property

Returns/sets a array value that indicates channels for discrete scanning of Pci9118 ActiveX control.

### Syntax

*object*.**DiscreteChannel(***index As Integer***)** [= number]

### Parameter

*index As Integer*

the index of Discrete channel array

### Settings

Channel number is from 0 to 15.

### Data Type

Integer

## DiscreteRange Property

Returns/sets a array value that indicates ranges of channel for discrete scanning of Pci9118 ActiveX control.

### Syntax

*object*.**DiscreteRange(***index As Integer***)** [= number]

### Parameter

*index As Integer*

the index of Discrete channel array

### Settings

PCI-9118DG/HR:

Polarity is Bipolar

| Value | Description |
|---|---|
| 0 | ±5V |
| 1 | ±2.5V |
| 2 | ±1.25V |
| 3 | ±0.625V |

Polarity is Unipolar

| Value | Description |
|---|---|
| 0 | 0~10V |
| 1 | 0~5V |
| 2 | 0~2.5V |
| 3 | 0~1.25V |

PCI-9118HG:

Polarity is Bipolar

| Value | Description |
|---|---|
| 0 | ±5V |
| 1 | ±0.5V |
| 2 | ±0.05V |
| 3 | ±0.005V |

Polarity is Unipolar

| Value | Description |
|---|---|
| 0 | 0~10V |
| 1 | 0~1V |
| 2 | 0~0.1V |
| 3 | 0~0.01V |

### Data Type

Integer

## *DiscreteNum Property*

Returns/sets a value that indicates the number of Discrete channel array.

### Syntax

*object*.**DiscreteNum** [= number]

### Settings

Number of Discrete channel array from 1 to 255.

### Data Type

Integer

## DoubleBufferMode Property

Returns/Sets a value that determines whether it is double-buffer mode or not.

### Syntax

*object*.**DoublebufferMode** [= boolean]

### Settings

| Value | Description |
|---|---|
| True | During interrupt analog input mode, the buffer is divided into two parts. When either part is full of data, it will get AiHalfReady event. |
| False | The buffer is single, when it is full of data, it will get AiComplete Event. |

### Data Type

Boolean

## EndChannel Property

Returns/sets a value that indicates end channel of contiguous scanning of Pci9118 ActiveX control.

### Syntax

*object*.**EndChannel** [= number]

### Settings

channel number is from 0 to 15.

### Data Type

Boolean

## ExtGate Propeerty

Returns/Sets a value that determines whether external gate is enable or disable.

### Syntax

*object*.**ExtGate**[= boolean]

### Settings

| Value | Description |
|---|---|
| True | Enable **t**he function of external gate. |
| False | Disable **t**he function of external gate. |

### Remarks

Users can specify the A/D control as external gate control or software gate control.

### Data Type

Boolean

## ExtTriggerPolarity Property

Returns/Sets a value that determines whether external trigger polarity is positive or negative.

### Syntax

*object*.**ExtTriggerPolarity**[= number]

### Settings

| Value | Description |
|---|---|
| 0 | positive trigger. |
| 1 | negative trigger. |

### Remark

ExtTriggerPolarity is valid only when *ClockSource* is External

### Data Type

Integer

## *InputMode Property*

Returns/sets a value that determines the mode of analog input.

### Syntax

*object*.**InputMode** [= number]

### Settings

| Value | Description |
|---|---|
| 0 | Single ended |
| 1 | Differential |

### Data Type

Integer

## *NumOfScan Property*

Sets a value that indicates the total number of scans to acquired.

### Syntax

*object*.**NumOfScan** [= number]

### Remarks

**Non-double-buffer mode**

This value multiply the total number of scan channels is the total number of A/D conversions to be performed. It must be the multiple of 2.

**Double-buffer-mode**

This value multiply the total number of scan channels is the size (in sample) of the circular buffer. It must be the multiple of 4.

### Data Type

Long

# OpenMode Property

Returns/sets a value that determines the mode of opening device .

**Syntax**

*object***.OpenMode** [= number]

**Settings**

| Value | Description |
|---|---|
| 0 | Automatically open device when the control was created |
| 1 | Don't open device when the control was created. Must call the Open method to open device. (Manual) |

**Data Type**

Integer.

# Polarity Property

Returns/sets a value that determines the polarity type of analog input.

**Syntax**

*object***.Polarity** [= number]

**Settings**

| Value | Description |
|---|---|
| 0 | Bipolar |
| 1 | Unipolar |

**Remarks**

Polarity is Bipolar:

Input range of 9118DG/HR is ±5V, ±2.5V, ±1.25V, ±0.625V

Input range of 9118HG is ±5V, ±0.5V, ±0.05V, ±0.005V

Polarity is Unipolar:

Input range of 9118DG/HR is 0~10V, 0~5V, 0~2.5V, 0~1.25V

Input range of 9118HG is 0~10V, 0~1V, 0~0.1V, 0~0.01V

**Data Type**

Integer

# PostCount Property

Returns/sets a value that determines the number of sample data that will be acquisited, after the trigger event happens.

**Syntax**

*object*.**PostCount** [= number]

### Settings

The value range is 0~65535.

### Data Type

Long

## *Range Property*

Sets a range for interrupt analog input at contiguous scanning of Pci9118.

### Syntax

*object*.**Range** [= number]

### Settings

PCI-9118DG/HR:

Polarity is Bipolar

| Value | Description |
|---|---|
| 0 | ±5V |
| 1 | ±2.5V |
| 2 | ±1.25V |
| 3 | ±0.625V |

Polarity is Unipolar

| Value | Description |
|---|---|
| 0 | 0~10V |
| 1 | 0~5V |
| 2 | 0~2.5V |
| 3 | 0~1.25V |

PCI-9118HG:

Polarity is Bipolar

| Value | Description |
|---|---|
| 0 | ±5V |
| 1 | ±0.5V |
| 2 | ±0.05V |
| 3 | ±0.005V |

Polarity is Unipolar

| Value | Description |
|---|---|
| 0 | 0~10V |
| 1 | 0~1V |
| 2 | 0~0.1V |
| 3 | 0~0.01V |

### Data Type

Integer

## *ReturnType Property*

Returns/sets a value that determines the return data type of analog input when AiComplete or AiHalfReady event would occur.

### Syntax

*object*.**ReturnType** [= number]

### Settings

**PCI-9118DG/HG**

| Value | Description |
|---|---|
| 0 | Scaled data only |
| 1 | Binary codes with channel only |
| 2 | Binary codes without channel only |
| 3 | Scaled data and binary codes with channel |
| 4 | Scaled data and binary codes without channel |
| 5 | No data return |

**PCI-9118HR**

| Value | Description |
|---|---|
| 0 | Scaled data only |
| 1 | Binary codes only |
| 2 | Scaled data and binary codes |
| 3 | No data return |

### Data Type

Integer

## *ScanRate Property*

Returns/sets a value that determines the scan rate (scans per second) of confinuous analog input

### Syntax

*object*.**ScanRate** [= number]

### Settings

For PCI-9118DG/HG, The range of (ScanRate * total number of scan channels) is between 0 and 330KHz.

For PCI-9118HR, The range of (ScanRate * total number of scan channels) is between 0 and 100KHz.

### Remarks

This property is used only when the ClockSource property is set to Internal timer pacer.

### Data Type

Double

## StartChannel Property

Returns/sets a value that indicates start channel of contiguous scanning of Pci9118 ActiveX control.

### Syntax

*object*.**StartChannel** [= number]

### Settings

channel number is from 0 to 15.

### Data Type

Integer

## StreamToFile Property

Returns/sets a value that determines if the control is enabled the function of streaming data to disk file.

### Syntax

*object*.**StreamToFile** [= boolean]

### Settings

| Value | Description |
|---|---|
| False | Disable the function of streaming data to disk file |
| True | Enable the function of streaming data to disk file |

### Data Type

Boolean

## TriggerMode Property

Returns/sets a value that determines trigger mode of analog input

### Syntax

*object*.**TriggerMode** [= number]

### Settings

| Value | Description |
|---|---|
| 0 | Disable |
| 1 | Post trigger |
| 2 | About trigger |

### Remarks

Use post trigger acquisition in application where you want to collect data after the start condition.

Use about trigger acquisition in application where you want to collect data before and after a specific trigger event.

### Data Type

Integer

## *TriggerPolarity Property*

Returns/sets a value that determines the active type of digital trigger.

### Syntax

*object*.**TriggerPolarity** [= number]

### Settings

| Value | Description |
|---|---|
| 0 | Positive |
| 1 | Negative |

### Remark

TriggerPolarity is valid only when TriggerMode is Post or About

### Data Type

Integer

# Methods

## *CheckContAI Method*

### Syntax

Function *object*.**CheckContAI(***AccessCount as long, stop as Integer***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*AccessCount as long*

Number of analog input data that has been transferred.

*stop as Integer*

Current state: 1 = stop, 0 = running

### Remarks

You can request DMA analog input status.

## *Open Method*

### Syntax

Function *object*.**Open (***[ErrMsgBox As Variant]***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*[ErrMsgBox As Variant]*

It is optional and boolean type. Default value is False

True, It will popup error message dialog box when the opening device is failed.

False, It will fire DAQError event instead of popping up dialog when the opening device is failed.

### Remarks

This method will be use when the OpenMode property is Manual.

### Note

In VC++, *ErrMsgBox* is a VARIANT of VT_I2**.**

## *ReadCounter0 Method*

### Syntax

Function *object***.ReadCounter0 (***value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*value As Variant*

Returns the current counter value from PCI-9118 card.

### Remarks

You can read the value from Timer Counter #0.

### Note

In VC++, *value* is a VARIANT of VT_I4**.**

## *ReadDILine Method*

### Syntax

Function *object***.ReadDILine (***line As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*line As Integer*

Selects one line number from the indicated port: from 0 to 3 (4-bit port).

*value As Variant*

Returns the data input from PCI-9118 card.

### Remarks

Users can read data from the indicated digital input line of PCI-9118 card.

### Note

In VC++, *value* is a VARIANT of VT_UI1**.**

## *ReadDIPort Method*

### Syntax

Function object.**ReadDIPort** (*value As Variant*) As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*value As Variant*

Sets one value buffer for retrieving 4-bit input data from PCI-9118 card.

### Remarks

Users can read 4-bit digital input data from PCI-9118 card.

### Note

In VC++, *value* is a VARIANT of VT_I4**.**

## *ReadSingleAI Method*

### Syntax

Function object.**ReadSingleAI** (*channel As Integer, range As Integer, value As Variant*) As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*channel As Integer*

Analog input channel number 0 ~ 15.

*range As Integer*

Range of analog input.

PCI-9118DG/HR

Polarity is Bipolar

| Value | Description |
|---|---|
| 0 | ±5V |
| 1 | ±2.5V |
| 2 | ±1.25V |
| 3 | ±0.625V |

Polarity is Unipolar

| Value | Description |
|---|---|

| | |
|---|---|
| 0 | 0~10V |
| 1 | 0~5V |
| 2 | 0~2.5V |
| 3 | 0~1.25V |

PCI-9118HG

Polarity is Bipolar

| Value | Description |
|---|---|
| 0 | ±5V |
| 1 | ±0.5V |
| 2 | ±0.05V |
| 3 | ±0.005V |

Polarity is Unipolar

| Value | Description |
|---|---|
| 0 | 0~10V |
| 1 | 0~1V |
| 2 | 0~0.1V |
| 3 | 0~0.01V |

*value As Variant*

The analog data read from analog input channel (already scaled to voltage).

### Remarks

You can read one data from one analog input channel.

### Note

In VC++, *value* is a VARIANT of VT_R8**.**

## *StartContAI Method*

### Syntax

Function object.**StartContAI** (*[FileName as Variant]*) As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*[FileName as String]*

It is optional and String type. Default value is NULL

FileName specified the file name of streaming data to disk.

### Remarks

You can use this method to start the continuous analog input function of PCI-9118 card. If the StreamToFile property is True then the data will be written to the file specified by FileName. Otherwise, the FileName parameter will be ignored.

The data file is written in binary format, with the lower byte first (little endian). Data type is “Binary codes with channel”. DAQBench provides a convenient tool DAQCvt to convert the binary file to the file format read easily. See DAQBench User’s Guide for the usage of the utility. If you want to handle the data by yourself, please refer to Appendix Data File Format for the file structure.

**Note**

In VC++, FileName is a VARIANT of VT_BSTR.

## *StartCounter0 Method*

**Syntax**

Function object.**StartCounter0**(*CtrMode As Integer, CtrValue As Variant, BinBcd As Integer*) As Boolean

**Return Value**

True if the function is successful; otherwise False.

**Arguments**

*CtrMode As Integer*

0: Toggle output from low to high on terminal count

1: Programmable one-shot

2: Rate generator

3: Square wave rate generator

4: Software-triggered strobe

5: Hardware-triggered strobe

*CtrValue as Variant*

Set the start value to the indicated counter.

*binbcd As Integer*

0: 16-bit binary counter, 1: 4-decade BCD counter.

**Remarks**

You can start the timer counter0 to operate in the specified mode.

**Note**

In VC++, *CtrValue* is a VARIANT of VT_I4**.**

## *StopContAI Method*

**Syntax**

Function object.**StopContAI** () As Boolean

**Return Value**

True if the function is successful; otherwise False.

**Arguments**

None

### Remarks

You can use this method to stop DMA analog input.

## StopCounter0 Method

### Syntax

Function object.**StopCounter0**(*State As Integer*) As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*State As Integer*

The logic state to which the timer counter0 is to be stopped, number: 0 or 1.

### Remarks

You can stop the timer counter0 and set the specified state.

## WriteDOPort Method

### Syntax

Function *object***.WriteDOPort (***value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*value as Variant*

4-bit data that will be written to the digital output port.

### Remarks

Users can write data to the PCI-9118 digital output port.

### Note

In VC++, *value* is a VARIANT of VT_I4**.**

## WriteSingleAO Method

### Syntax

Function *object***.WriteSingleAO (***channel As integer***,** *value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*channel As Integer*

Channel number is 0 or 1.

*value As Variant*

Sets one value to writing to analog output of PCI-9118 card, the value range is between –10V and 10V.

### Remarks

Users can write data to analog output channel of PCI-9118 card.

### Note

In VC++, *value* is a VARIANT of VT_R8**.**

## *WriteDOLine Method*

### Syntax

Function *object.***WriteDOLine(***line As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*line As Integer*

Selects one line number from the indicated port from 0 to 3.

*value As Variant*

Sets 0 or 1 to the indicated line.

### Note

In VC++, *value* is a VARIANT of VT_I4.

## *ReadBackDOPort Method*

Reads back data from the digital output port.

### Syntax

Function *object.***ReadBackDOPort (***value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*value As Variant*

Data that is read back from the port.

### Note

In VC++, *value* is a VARIANT of VT_I4.

## *ReadBackDOLine Method*

Reads back data from the indicated digital output line of the digital output port.

### Syntax

Function *object*.**ReadBackDOLine(***line As Integer, value As Variant***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*line As Integer*

Selects one line number from the indicated port from 0 to 3.

*value As Variant*

Data that is read back from the indicated line.

### Note

In VC++, *value* is a VARIANT of VT_UI1.

# Events

## *AiComplete Event*

### Syntax

sub *ControlName*_**AiComplete(** *ScaledData as Variant, BinaryCodes As Variant* **)**

### Arguments

*ScaledData as Variant*

The analog input data array that have been translated to the engineering data (voltage) according to AIRange property.

*BinaryCodes As Variant*

The analog input data array with digital format.

### Remarks

This event occurs when continuous analog input function is completed.

Whether *ScaledData* or *BinaryCodes* contains data depends on *ReturnType* property setting.

**PCI-9118DG/HG**

"*BinaryCodes with channel*" format:

Range 0 to +65535. The least significant 4 bits are channel number.

"*BinaryCodes without channel*" format:

Range 0 to +4095.

**PCI-9118HR**

"*BinaryCodes*" format:

Range -32768 to +32767.

### Note

In VC++, *ScaledData* is a VARIANT of VT_ARRAY | VT_R4, *BinaryCodes* is a VARIANT of VT_ARRAY | VT_I4 (with channel) or VT_ARRAY | VT_I2 (without channel).

# AiHalfReady Event

### Syntax

sub *ControlName*_**AiHalfReady(** *ScaledData as Variant, BinaryCodes As Variant* **)**

### Arguments

*ScaledData as Variant*

The analog input data array that have been translated to the engineering data (voltage) according to AIRange property.

*BinaryCodes As Variant*

The analog input data array with digital format.

### Remarks

This event occurs when one half-buffer of the circular buffer is full at continuous analog input operation.

Whether *ScaledData* or *BinaryCodes* contains data depends on *ReturnType* property setting.

**PCI-9118DG/HG**

"*BinaryCodes with channel*" format:

Range 0 to +65535. The least significant 4 bits are channel number.

"*BinaryCodes without channel*" format:

Range 0 to +4095.

**PCI-9118HR**

"*BinaryCodes*" format:

Range -32768 to +32767.

### Note

In VC++, *ScaledData* is a VARIANT of VT_ARRAY | VT_R4, *BinaryCodes* is a VARIANT of VT_ARRAY | VT_I4 (with channel) or VT_ARRAY | VT_I2 (without channel).

# DAQError Event

### Syntax

sub *ControlName*_**DAQError (** *ErrString As String* **)**

### Arguments

*ErrString As String*

The string of error reason

### Remarks

This event will occur when some error occur in control

# Pci9812 ActiveX Control

The PCI-9812 is a PCI-bus multi-function data acquisition card. The Pci9812 ActiveX control is a software component that provides the interface for users to control PCI-9812 card. The properties and methods allow users to perform analog I/O and digital I/O.

# Properties

## *DeviceName Property*

Returns/sets a string that indicates the device name of a PCI-9812 card that was defined in NuDAQCfg Utility.

**Syntax**

*object*.**DeviceName** [= string]

**Remarks**

The device name of NuDAQ cards must be defined in NuDAQCfg Utility. The indicated NuDAQ card by device name may be a local device or a remote device on remote machine. The device name of PCI9812 ActiveX Control must be specified to a PCI-9812 card defined in NuDAQCfg Utility.

**Data Type**

String

## *AIRange Property*

Sets a range for interrupt analog input.

**Syntax**

*object*.**AIRange** [= number]

**Settings**

| Value | Description |
|---|---|
| 0 | ±1V |
| 1 | ±5V |

**Data Type**

Integer

## *CardType Property*

Returns/sets a value that indicates the card type for programming Pci9812 ActiveX control.

**Syntax**

*object*.**CardType** [= number]

**Settings**

| Number | Card Type | Resolution |
|---|---|---|
| 4 | PCI-9812 | 12-bits |
| 5 | PCI-9810 | 10-bits |

**Remarks**

This property will be automatically set value when the device name be specified and the device information be retrieved successfully.

**Data Type**

Integer

# *Channel Property*

Returns/sets a value that determines the analog input channel for DMA data transfer.

**Syntax**

*object*.**Channel** [= number]

**Settings**

| Value | Description |
|---|---|
| 1 | DMA analog input will only read data from channel 0. |
| 2 | DMA analog input will read data from channel 0 and channel 1. The sequence of channel scan is 0, 1, 0, 1,…… |
| 3 | It is invalid. |
| 3 | DMA analog input will read data from aoo four channels. The seguence of channel scan is 0, 1, 2, 3, 0, 1, 2, 3,…... |

**Data Type**

Integer

# *ClockFreq Property*

Returns/sets a value that determines the clock frequence.

**Syntax**

*object*.**ClockFreq** [= number]

**Settings**

| Value | Description |
|---|---|
| 0 | The frequence is smaller than PCI Bus frequency. |
| 1 | The frequence is bigger than PCI Bus frequency. |

**Data Type**

Integer

## *ClockSource Property*

Returns/sets a value that determines the A/D clock source.

### Syntax

*object*.**ClockSource** [= number]

### Settings

| Value | Setting |
|---|---|
| 0 | Internal timer pacer |
| 1 | External – Sin |
| 2 | External – Square |

### Remarks

If the ClockSource property is set to External Sin or External Square, the frequency divider is set to 2 by Pci9812 ActiveX control. Hence, the scan rate is: Frequency of external trigger source/2)

### Data Type

Integer

## *DoubleBufferMode Property*

Returns/Sets a value that determines whether it is double-buffer mode or not.

### Syntax

*object*.**DoublebufferMode** [= boolean]

### Settings

| Value | Description |
|---|---|
| True | During interrupt analog input mode, the buffer is divided into two parts. When either part is full of data, it will get AiHalfReady event. |
| False | The buffer is single, when it is full of data, it will get AiComplete Event. |

### Data Type

Boolean

## *NumOfScan Property*

Sets a value that indicates the total number of scans to be acquired.

### Syntax

*object*.**NumOfScan** [= number]

### Remarks

**Non-double-buffer mode**

This value multiply the total number of scan channels is the total number of A/D conversions to be performed.

It must be the multiple of 2.

**Double-buffer-mode**

This value multiply the total number of scan channels is the size (in sample) of the circular buffer. It must be the multiple of 4.

### Data Type

Long

## *OpenMode Property*

Returns/sets a value that determines the mode of opening device .

### Syntax

*object*.**OpenMode** [= number]

### Settings

| Value | Description |
|---|---|
| 0 | Automatically open device when the control was created |
| 1 | Don't open device when the control was created. Must call the Open method to open device. (Manual) |

### Data Type

Integer.

## *PostCount Property*

Returns/sets a value that determines the number of sample data that will be acquisited, after the trigger event happens.

### Syntax

*object*.**PostCount** [= number]

### Settings

The value range is 0~65535.

### Remarks

PostCount is valid only when the *TriggerMode* is Delay trigger or Middle trigger

### Data Type

Integer

## *ReturnType Property*

Returns/sets a value that determines the return data type of analog input when AiComplete or AiHalfReady event would occur.

### Syntax

*object*.**ReturnType** [= number]

### Settings

| Value | Description |
|---|---|
| 0 | Scaled data only |
| 1 | Binary codes with miscellaneous data |
| 2 | Binary codes without miscellaneous data |
| 3 | Scaled data and binary codes with miscellaneous data |
| 4 | Scaled data and binary codes without miscellaneous data |
| 5 | No data return |

### Data Type

Integer

## ScanRate Property

Returns/sets a value that determines the scan rate (scans per second)of continuous analog input.

### Syntax

*object*.**ScanRate** [= number]

### Settings

The range of (ScanRate * total number of scan channels) must be between 0 and 20MHz.

### Remarks

This property is used only when the ClockSource property is set to Internal timer pacer.

### Data Type

Double

## StreamToFile Property

Returns/sets a value that determines if the control is enabled the function of streaming data to disk file.

### Syntax

*object*.**StreamToFile** [= boolean]

### Settings

| Value | Description |
|---|---|
| False | Disable the function of streaming data to disk file |
| True | Enable the function of streaming data to disk file |

### Data Type

Boolean

## TriggerLevel Property

Returns/sets a value that determines the value for the trigger level.

### Syntax

*object*.**TriggerLevel** [= number]

### Settings

The value range is 0~255.

### Remarks

The trigger event occurs when the trigger signal changes from a voltage. It could be less than the trigger level to a voltage or greater than the specified trigger level.

The trigger level is set as a reference voltage for trigger event.

### Data Type

Integer

## *TriggerMode Property*

Returns/sets a value that determines trigger mode of analog input

### Syntax

*object*.**TriggerMode** [= number]

### Settings

| Value | Description |
|---|---|
| 0 | Software Trigger |
| 1 | Post Trigger |
| 2 | Pre-Trigger |
| 3 | Delay Trigger |
| 4 | Middle Trigger |

### Remarks

Use post trigger acquisition in application where you want to collect data after the start condition.

### Data Type

Integer

## *TriggerPolarity Property*

Returns/sets a value that determines the active type of digital trigger.

### Syntax

*object*.**TriggerPolarity** [= number]

### Settings

| Value | Description |
|---|---|
| 0 | Positive |
| 1 | Negative |

### Remark

TriggerPolarity is valid only when TriggerMode is not Software trigger

### Data Type

Integer

## *TriggerSource Property*

Returns/sets a value that determines the trigger source for retrieving the trigger signal.

### Syntax

*object*.**TriggerSource** [= number]

### Settings

| Value | Description |
|---|---|
| 0 | Channel 0 |
| 1 | Channel 1 |
| 2 | Channel 2 |
| 3 | Channel 3 |
| 4 | Ext-Digital |

### Data Type

Integer

# Methods

## *CheckContAI Method*

### Syntax

Function *object*.**CheckContAI(***AccessCount as long, stop as Integer***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*AccessCount as long*

Number of analog input data that has been transferred.

*stop as Integer*

Current state: 1 = stop, 0 = running

### Remarks

You can request DMA analog input status.

## *Open Method*

### Syntax

Function *object*.**Open (***[ErrMsgBox As Variant]***)** As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*[ErrMsgBox As Variant]*

It is optional and boolean type. Default value is False

True, It will popup error message dialog box when the opening device is failed.

False, It will fire DAQError event instead of popping up dialog when the opening device is failed.

### Remarks

This method will be use when the OpenMode property is Manual.

### Note

In VC++, *ErrMsgBox* is a VARIANT of VT_I2**.**

## *StartContAI Method*

### Syntax

Function object.**StartContAI** (*[FileName as Variant]*) As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

*[FileName as String]*

It is optional and String type. Default value is NULL

FileName specified the file name of streaming data to disk.

### Remarks

You can use this method to start the DMA analog input function of PCI-9812 card. If the StreamToFile property is True then the DMA data will be written to the file specified by FileName. Otherwise, the FileName parameter will be ignored.

The data file is written in binary format, with the lower byte first (little endian). Data type is “Binary codes with miscellaneous data”. DAQBench provides a convenient tool DAQCvt to convert the binary file to the file format read easily. See DAQBench User’s Guide for the usage of the utility. If you want to handle the data by yourself, please refer to Appendix Data File Format for the file structure.

### Note

In VC++, FileName is a VARIANT of VT_BSTR.

## *StopContAI Method*

### Syntax

Function object.**StopContAI** () As Boolean

### Return Value

True if the function is successful; otherwise False.

### Arguments

None

### Remarks

You can use this method to stop DMA analog input.

# Events

## *AiComplete Event*

### Syntax

sub *ControlName*_**AiComplete(** *ScaledData as Variant, BinaryCodes As Variant* **)**

### Arguments

*ScaledData as Variant*

The analog input data array that have been translated to the engineering data (voltage) according to AIRange property.

*BinaryCodes As Variant*

The analog input data array with digital format.

### Remarks

This event occurs when continuous analog input function is completed.

Whether *ScaledData* or *BinaryCodes* contains data depends on *ReturnType* property setting.

**PCI-9812**

"*BinaryCodes with miscellaneous data*" format:

Range -32768 to +32767.

bit 15~4: A/D data (-2048 to 2047)

bit 3: trigger detection flag

bit 2~0: digital input data from DI2, DI1, DI0

"*BinaryCodes without miscellaneous data*" format:

Range -2048 to +2047.

**PCI-9810**

"*BinaryCodes with miscellaneous data*" format:

Range -32768 to +32767.

bit 15~6: A/D data (-512 to 511)

bit 5~4: don't care

bit 3: trigger detection flag

bit 2~0: digital input data from DI2, DI1, DI0

"*BinaryCodes without miscellaneous data*" format:

Range -512 to +511.

### Note

In VC++, *ScaledData* is a VARIANT of VT_ARRAY | VT_R4, *BinaryCodes* is a VARIANT of VT_ARRAY | VT_I4 (with miscellaneous data) or VT_ARRAY | VT_I2 (without miscellaneous data).

## *AiHalfReady Event*

### Syntax

sub *ControlName*_**AiHalfReady(** *ScaledData as Variant, BinaryCodes As Variant* **)**

### Arguments

*ScaledData as Variant*

The analog input data array that have been translated to the engineering data (voltage) according to AIRange property.

*BinaryCodes As Variant*

The analog input data array with digital format.

### Remarks

This event occurs when one half-buffer of the circular buffer is full at continuous analog input operation.

Whether *ScaledData* or *BinaryCodes* contains data depends on *ReturnType* property setting.

**PCI-9812**

"*BinaryCodes with miscellaneous data*" format:

Range -32768 to +32767.

bit 15~4: A/D data (-2048 to 2047)

bit 3: trigger detection flag

bit 2~0: digital input data from DI2, DI1, DI0

"*BinaryCodes without miscellaneous data*" format:

Range -2048 to +2047.

**PCI-9810**

"*BinaryCodes with miscellaneous data*" format:

Range -32768 to +32767.

bit 15~6: A/D data (-512 to 511)

bit 5~4: don't care

bit 3: trigger detection flag

bit 2~0: digital input data from DI2, DI1, DI0

"*BinaryCodes without miscellaneous data*" format:

Range -512 to +511.

### Note

In VC++, *ScaledData* is a VARIANT of VT_ARRAY | VT_R4, *BinaryCodes* is a VARIANT of VT_ARRAY | VT_I4 (with miscellaneous data) or VT_ARRAY | VT_I2 (without miscellaneous data).

## *DAQError Event*

### Syntax

sub *ControlName*_**DAQError (** *ErrString As String* **)**

### Arguments

*ErrString As String*

The string of error reason

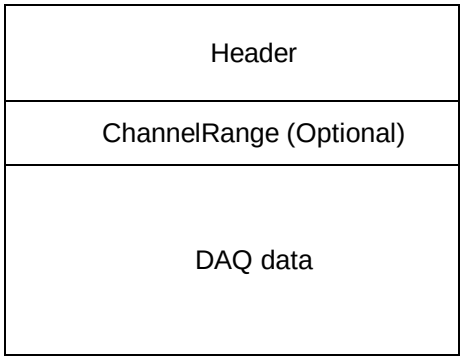
### Remarks

This event will occur when some error occur in control

# Appendix Data File Format

This appendix describes the file format of the data files generated by the stream to file functions.

The data file includes three parts, Header, ChannelRange (optional) and Data block. The file structure is as the figure below:




**Header**

The *header* part records the information related to the stored data and its total length is 60 bytes. The data structure of the file header is as follows:

| **Header** | | | *Total Length: 60 bytes* |
|---|---|---|---|
| **Elements** | **Type** | **Size (bytes)** | **Comments** |
| ID | char | 10 | file ID<br>*ex. ADLinkDAQ1* |
| card_type | short | 2 | card Type<br>*ex. Pci7250, Pci9112* |
| num_of_channel | short | 2 | number of scanned channels<br>*ex. 1, 2* |
| channel_no | unsigned char | 1 | channel number where the data read from (only available as the num_of_channel is 1)<br>*ex. 0, 1* |
| num_of_scan | long | 4 | the number of scan for each channel (total count / num_of_channel) |
| data_width | short | 2 | the data width<br>0: 8 bits, 1: 16 bits, 2: 32 bits |
| channel_order | short | 2 | the channel scanned sequence<br>0: normal (ex. 0-1-2-3) |

| | | | |
|---|---|---|---|
| | | | 1: reverse (ex. 3-2-1-0)<br>2: custom* (ex. 0, 1, 3) |
| ad_range | short | 2 | the AI range code<br>Please refer to Appexdix B<br>*ex. 0 (AD_B_5V)* |
| scan_rate | double | 8 | The scanning rate of each channel<br>(total sampling rate / num_of_channel) |
| num_of_channel_range | short | 2 | The number of ChannelRange* structure |
| start_date | char | 8 | The starting date of data acquisition<br>*ex. 12/31/99* |
| start_time | char | 8 | The starting time of data acquisition<br>*ex. 18:30:25* |
| start_millisec | char | 3 | The starting millisecond of data acquisition<br>*ex. 360* |
| reserved | char | 6 | not used |

∗ If the *num_of_channel_range* is 0, the *ChannelRange* block won't be included in the data file.

∗ The *channel_order* is set to "custom" only when the card supports variant channel scanning order.

**ChannelRange**

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

Total Length = *2* * num_of_channel_range bytes

The data structure of each ChannelRange unit is as follows:

| **ChannelRange Unit**<br>*Length: 2 bytes* | | | |
|---|---|---|---|
| **Elements** | **Type** | **Size (bytes)** | **Comments** |
| channel | char | 1 | scanned channel number<br>*ex. 0, 1* |
| range | char | 1 | the AI range code of *channel*<br>Please refer to Appexdix B<br>*ex. 0 (AD_B_5V)* |

**Data Block**

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