

# **ActiveDAS Control**

## **User Manual**

@Copyright 2001 EVOC Technology Co., Ltd.  
Copyright Reserved

April 20, 2001, Version 1

Except for the accessories attached to the product as specified herein, what is contained in this User Manual does not represent the commitments of EVOC Company. EVOC Company reserves the right to revise this User Manual, without prior notice, and will not be held liable for any direct, indirect, intended or unintended losses and/or hidden dangers due to installation or improper operation.

The information in this user manual is protected by copyright. No part of this manual may be reproduced in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without permission in writing from EVOC.

#### Trademark

ActiveDAS is a registered trademark of EVOC Technology Co., Ltd. Windows 95/98/NT, Visual Basic, and Visual C++ are the registered trademarks of Microsoft Company. Intel is a registered trademark of Intel Group. Other product names mentioned in this Manual are only used for identification here.

# Table of Contents

<b>TABLE OF CONTENTS</b> .....	<b>0</b>
<b>CHAPTER 1 INTRODUCTION</b> .....	<b>3</b>
1.1 DEFINITION .....	3
1.2 INSTALL EVOC ACTIVE <span style="font-variant: small-caps;">DAS</span> CONTROL .....	3
1.2.1 <i>System Requirements</i> .....	4
1.2.2 <i>Directions for Install</i> .....	4
1.2.3 <i>Install Files</i> .....	6
1.3 SET PROPERTY .....	6
1.3.1 Set “ <i>Property Page</i> ” .....	6
1.3.2 <i>Program Setup</i> .....	8
1.4 METHOD FOR USING CONTROL .....	8
1.5 EVENTS IN CONTROL .....	8
AFTER CONFIGURING A CONTROL IN FORM, YOU CAN COMPILE ITS EVENTS, E.G. IN VISUAL BASIC: PRIVATE SUB GRAPH1_CLICK() .....	8
<b>CHAPTER 2 USE OF EVOC ACTIVE<span style="font-variant: small-caps;">DAS</span> IN VISUAL BASIC</b> .....	<b>9</b>
2.1 DEVELOP VISUAL BASIC PROGRAM .....	9
2.1.1 <i>Add EVOC Active<span style="font-variant: small-caps;">DAS</span> to Toolbox</i> .....	9
2.1.2 <i>Use EVOC Active<span style="font-variant: small-caps;">DAS</span> to establish UI</i> .....	9
2.1.3 <i>Use “Property Page” in Visual Basic</i> .....	10
2.1.4 <i>Use Visual Basic Program to Set Property</i> .....	12
2.1.5 <i>Use Control in Visual Basic</i> .....	12
2.1.6 <i>Events in Visual Basic</i> .....	12
<b>CHAPTER 3 USE OF EVOC ACTIVE<span style="font-variant: small-caps;">DAS</span> IN VISUAL C++</b> .....	<b>13</b>
<b>CHAPTER 3 USE OF EVOC ACTIVE<span style="font-variant: small-caps;">DAS</span> IN VISUAL C++</b> .....	<b>14</b>

3.1 DEVELOP VISUAL C++ PROGRAM .....	14
3.1.1 Establish Application Program in Visual C++ .....	14
3.1.2 Add EVOC ActiveDAS to Toolbox.....	14
3.1.3 Use EVOC ActiveDAS to Establish UI.....	15
3.1.5 Set Control Property in Visual C++ .....	16
3.1.6 Set Control in Visual C++, such as .....	17
3.1.7 Set control Events in Visual C++.....	17
<b>CHAPTER 4 USE OF EVOC ACTIVE DAS IN DELPHI.....</b>	<b>18</b>
4.1 DEVELOP DELPHI PROGRAM .....	18
4.1.1 Add EVOC ActiveDAS Control to Controls Panel .....	18
<b>CHAPTER 5 DETAILED INTRODUCTION OF EVOC</b>	
<b>ACTIVE DAS .....</b>	<b>19</b>
5.1 E7SEGMENT CONTROL .....	19
5.2 BOOLEAN CONTROL .....	20
5.3 KNOB CONTROL.....	22
5.4 SLIDE CONTROL.....	26
5.5 MOTOR CONTROL .....	26
5.6 PIPE CONTROL .....	27
5.7 PUMP CONTROL .....	28
5.8 TANK CONTROL .....	29
5.9 VALVE CONTROL.....	29
5.10 GRAPH CONTROL.....	29
5.11 CHART CONTROL .....	35
5.13 DASAI CONTROL .....	36
5.14 DASAO CONTROL.....	38
5.15 DASDI CONTROL .....	39
5.16 DASDO CONTROL.....	40

5.17	DASCOUNTER CONTROL.....	41
5.18	DASPULSE CONTROL .....	42
5.19	DASALARM CONTROL .....	42
5.20	PCL731A CONTROL .....	43
5.21	PCL733 CONTROL.....	43
5.22	PCL734 CONTROL.....	44
5.23	PCL836A CONTROL .....	44
<b>CHAPTER 6 SAMPLES AND POINTS FOR ATTENTION.....</b>		<b>46</b>
6.1	THE SAMPLE OF DEMOUI10DIO .....	46
6.2	THE SAMPLE OF DEMOUI10DA.....	48
6.3	THE SAMPLE OF DEMOUI10AD.....	50
6.4	THE SAMPLE OF GRAPHDEMO .....	54
6.5	THE SAMPLE OF CHARTDEMO .....	56
6.6	THE COMPREHENSIVE SAMPLE OF INSTRUMENT .....	58
6.7	THE COMPREHENSIVE SAMPLE OF SYNTHESIZE .....	61
6.8	THE SAMPLE OF VCGRAPHDEMO.....	61
6.9	THE SAMPLE OF DASAiDEMO.....	62
6.10	THE SAMPLE OF DASAoDEMO .....	64
6.11	THE SAMPLE OF DASDiDEMO .....	64
6.12	THE SAMPLE OF DASDoDEMO .....	66
6.13	OTHER POINTS FOR ATTENTION .....	67
<b>CHAPTER 7 DEVICES SUPPORTED BY HARDWARE</b>		
<b>OPERATION AND THEIR INSTALLATION.....</b>		<b>68</b>
7.1	TYPES OF CARDS SUPPORTED BY VARIOUS CONTROLS .....	68
7.2	INSTALLATION .....	69
<b>CHAPTER 8 USAGE OF DEVINST.EXE.....</b>		<b>71</b>

## Chapter 1 Introduction

This Chapter mainly introduces the basic information about EVOC ActiveDAS control and its installation method.

### 1.1 Definition

EVOC ActivDAS control is an aggregation of the controls in industrial control industry and used for any program supporting ActiveX control. EVOC ActivDAS control can help develop complex UI (User Interface) program which is used for data display and acquisition. EVOC ActivDAS control falls into three categories:

- (1) EVOCU110.ocx—This 32-bit control is used for data acquisition and device control, including Boolean, E7Segment, Knob, and Slide.
- (2) EVOCEquipment.ocx—This control is used for the design of multimedia program, including Motor, Pipe, Pump, Tank, and Valve.
- (3) EVOCCart10.ocx—It includes Graph and Chart.
- (4) Hardware.ocx—This control is mainly used for EVOC hardware, including eight controls and one DevInst.exe. They are DASDevice, DASAI, DASAO, DASDI, DASDO, DASCounter, DASPulse, and DASAlarm.

### 1.2 Install EVOC ActiveDAS Control

It will take several minutes to install EVOC ActiveDAS. Users need to register MFC, such as `regsvr32 C:\windows\system\mfc42.dll` before using a control. Before using a card control, you need to configure the resources of card including port, interrupt and DMA via DevInst.exe.

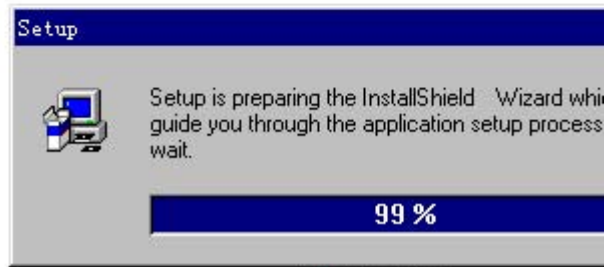
### 1.2.1 System Requirements

When using EVOC ActiveDAS control, the following system requirements are needed:

- (1) Microsoft Windows95/98 or Windows NT OS
- (2) PC 66MHz 80486 or higher configuration
- (3) VGA resolution adapter card or higher configuration
- (4) Set over 16 bits enhanced color for the screen, otherwise the image of control will not be clear.
- (5) ActiveX containers, such as Visual Basic (32-bit version) , Visual C++ or Delphi (32-bit version)
- (6) Memory with no less than 16M
- (7) Free disk space with no less than 10M
- (8) Mouse compatible with Windows
- (9) Register MFC before using a control, such as regsvr32 C:\windows\system\mfc42.dll
- (10) Before using a card control, you need to configure the resources of card including port, interrupt and DMA via DevInst.exe. For the configuration method, please refer to the Chapter 8 of this Manual.

### 1.2.2 Directions for Install

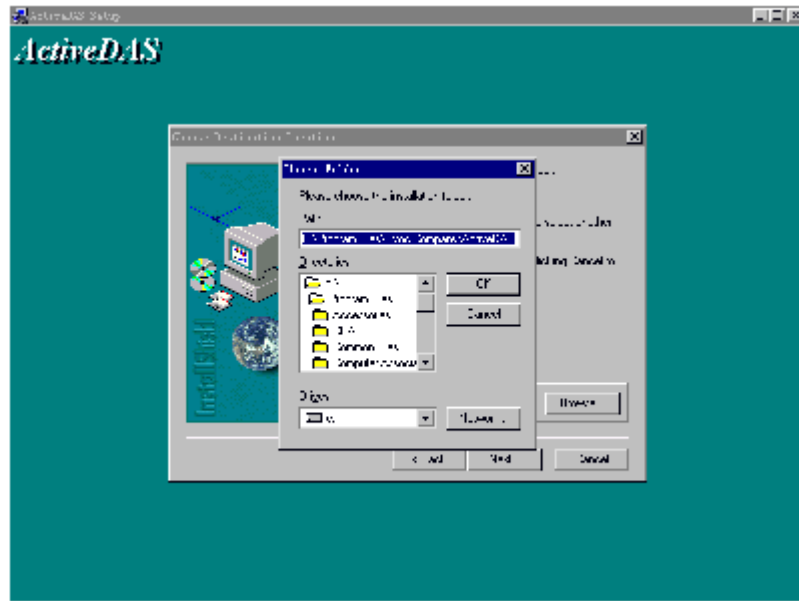
This part introduces the installation method of EVOC ActiveDAS control. Execute "Setup.exe" in CD, and follow the on-screen instructions to install it. Of course, you can copy it to the disk to install it. First you can see the following image:





### 1.2.3 Install Files

The route of file can be changed during installation, shown as follows:



Install files are automatically copied into \Windows\System, and then registered automatically. The programs of install files will be copied into the directory that you select.

### 1.3 Set Property

You can set the property of a control via “Property Page” or programming.

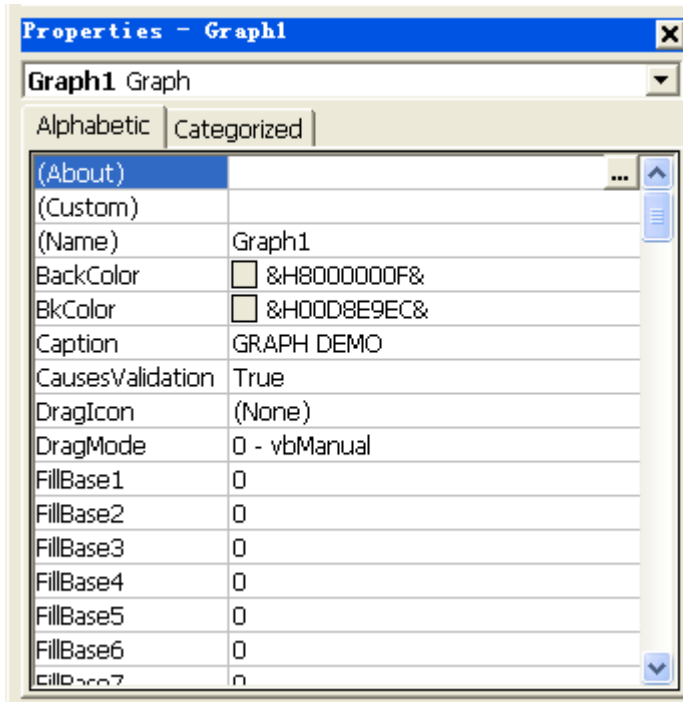
#### 1.3.1 Set “Property Page”

“Property Page” is a common user interface in Windows98 and

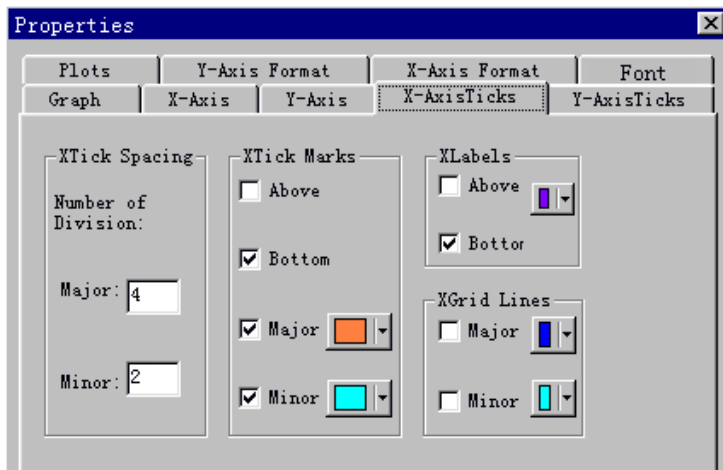
Windows NT. When you want to change the property of any object, you can right click the object and then click “property”.

“Property Page” of some programs (such as Visual Basic and Delphi), includes two types: “Default Property Page” and “User-defined Property Page”.

The “Default Property Page” of Visual Basic of Graph control is shown as follows:



“User-defined Property Page” has different appearances and functions in different controls. The “User-defined Property Page” of Graph control is shown below:



### 1.3.2 Program Setup

e.g. changing Boolean value in Visual Basic: `Boolean1.Value=0`

We advise you not to use program to set properties unless you are very familiar with them. It is simple to set properties via “Property Page”.

### 1.4 Method for Using Control

ActiveX control has its own method or function. You can use these methods in your program, e.g. in VB: `Graph1.PlotGraphs value`.

### 1.5 Events in Control

After configuring a control in Form, you can compile its events, e.g. in Visual Basic: `Private Sub Graph1_Click()`

End Sub

## Chapter 2 Use of EVOC ActiveDAS in Visual Basic

This chapter introduces that how to use EVOC ActiveDAS in Visual Basic, how to put it into Visual Basic, how to set its properties, methods, and events, and how to operate it.

Before using a card control, you need to configure the resources of card including port, interrupt and DMA via DevInst.exe. You need to register MFC before using a control, such as regsvr32 C:\windows\system\mfc42.dll.

### 2.1 Develop Visual Basic Program

#### 2.1.1 Add EVOC ActiveDAS to Toolbox

Before establishing an application program, EVOC ActiveDAS control must be added to toolbox. EVOC ActiveDAS falls into three categories: EVOCU110.ocx, EVOCEquipment.ocx, and EVOCCart10.ocx.

Method for adding it to toolbox:

- (1) In toolbox, right click the blank and click “components”
- (2) You can see a dialog box that shows EVOCCart10 ActiveX Control module, EVOCEquipment ActiveX Control module and EVOCU110 ActiveX Control module.
- (3) Select a control upon your demand and click “OK”.

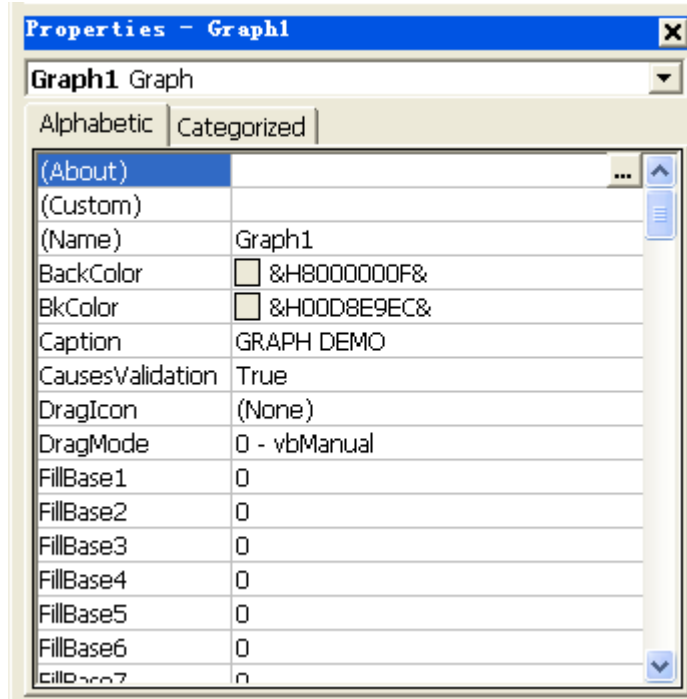
#### 2.1.2 Use EVOC ActiveDAS to establish UI

After adding EVOC ActiveDAS in toolbox, you can establish your program. Select corresponding control icon in toolbox and drag it to

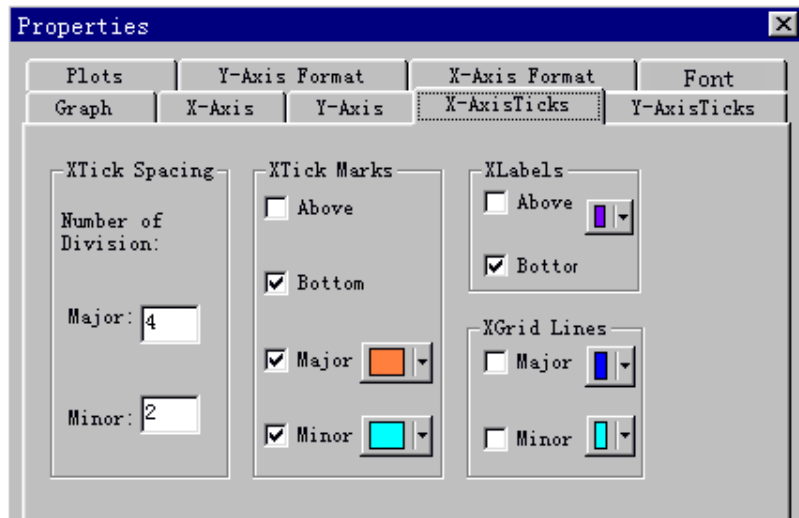
Form. And set its properties via “Property Page”.

### 2.1.3 Use “Property Page” in Visual Basic

Visual Basic can automatically set some default values after the control is put into Form. The property setup of the control has two methods. One is “default”, shown as below:



The other is “user-defined”, shown as below:



#### 2.1.4 Use Visual Basic Program to Set Property

You can use program to set the control property, e.g.: Boolean1.value=0. We advise you not to use program to set properties unless you are very familiar with them.

#### 2.1.5 Use Control in Visual Basic

ActiveX control has its own method or function. You can directly use these methods, e.g.: Graph1.PlotGraphs value

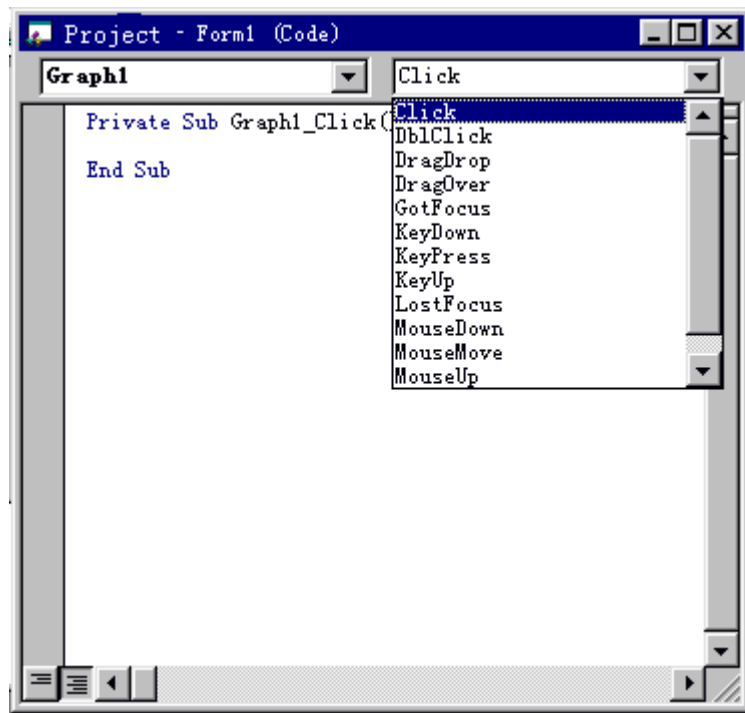
#### 2.1.6 Events in Visual Basic

After configuring the control in Form, you can compile its events, e.g. in Visual Basic:

```
Private Sub Graph1_Click()
```

```
End Sub
```

Please see the following image: the left part is controls and the right part is its events.





## Chapter 3 Use of EVOC ActiveDAS in Visual C++

This chapter introduces how to use EVOC ActiveDAS in Visual C++, how to add it to Visual C++, how to establish a wrapper class, and how to establish an application program via MFC and MFC ClassWizard. You should well know C++, Visual C++, and ActiveX control.

Before using a card control, you need to configure the resources of card including port, interrupt and DMA via DevInst.exe. You need to register MFC before using a control, such as regsvr32 C:\windows\system\mfc42.dll.

### 3.1 Develop Visual C++ Program

Method for developing Visual C++ program:

- (1) Establish a new working space or project in Visual C++
- (2) You can use MFC application guide when establishing a program.
- (3) Add EVOC ActiveDAS control to Tool Bar of the application program.
- (4) Set properties of the control through “Property Page”.
- (5) Use control method to operate the control; use control events to respond to the action of control.

#### 3.1.1 Establish Application Program in Visual C++

Mark “ActiveX Control” with “√” when establishing a program.

#### 3.1.2 Add EVOC ActiveDAS to Toolbox

After the control is put into Component Library, there appears a series

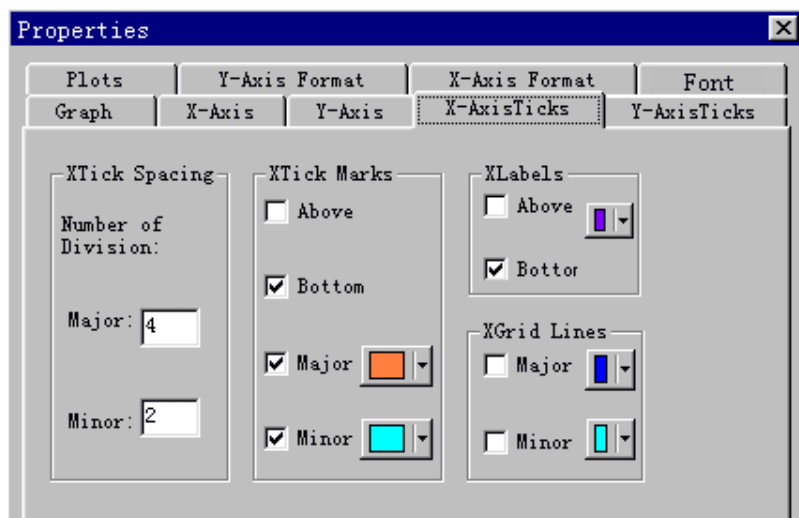
of C++ wrapper classes in the project. The controls can be executed through these wrapper classes. Tool Bar of the control appears after activating “Visual C++” dialog box.

Steps for adding to project:

- (1) Select Project>>Add To Project...>Components and Controls, and double click the registered ActiveX control.
- (2) Select your needed control and click “Insert”.

### 3.1.3 Use EVOC ActiveDAS to Establish UI

After adding to Tool Bar, you should drag the control to the dialog box. Right click to open “User-defined Property Page”, shown as follows:



### 3.1.4 Use EVOC ActiveDAS to Program

You can use Visual C++ wrapper class to set the properties, methods, events of a control. Input a variable for a control, and you can operate this control through the variable. “Variant” type is used in Graph and Chart control. You can refer to the sample of VCGraphDemo.

(1) First define the VARIANT, such as data and SAFEARRAY FAR\* such as psa.

(2) Set VariantInit(&data) during program initialization.

(3) When the program is destroyed, please set:

```
if(psa != NULL)
    SafeArrayDestroy(psa);
VariantClear(&data);
```

(4) Code in OnButton1(): first define an array, such as double array[1024], and define a safe array that shows the number of dimensions, such as SAFEARRAYBOUND rgsabound[1], which means it is a single-dimensional array.

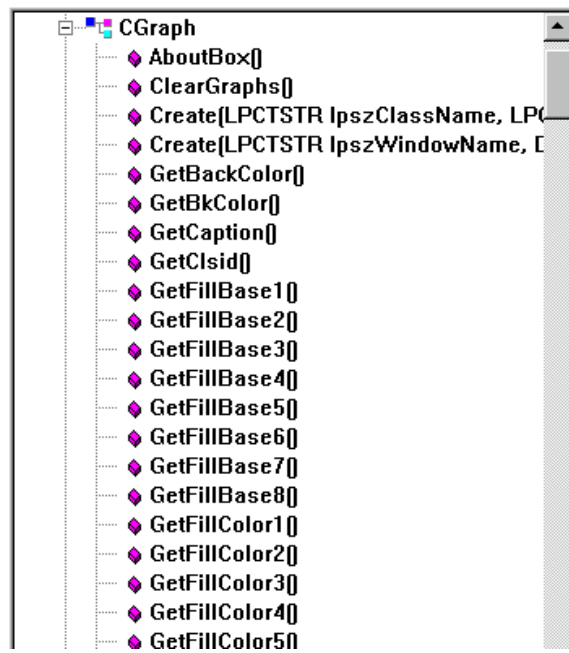
“SafeArrayCreate” is used to establish a safe array.

“SafeArrayPutElement” is used to input values for a safe array.

### 3.1.5 Set Control Property in Visual C++

You cannot directly read and write the property of a control. You need to set the property of a control via the function of a wrapped class, such as:

```
m_graph1.GetCaption()
```



3.1.6 Set Control in Visual C++, such as  
`m_graph1.PlotGraphs` value

3.1.7 Set control Events in Visual C++

Steps for adding events:

- (1) Right click a control, and select "Establish MFC ClassWizard".
- (2) Find out "Message Maps"
- (3) Select "Object ID".
- (4) Select its events.
- (5) Add events and program them.

## Chapter 4 Use of EVOC ActiveDAS in Delphi

This chapter introduces how to use EVOC ActiveDAS in Delphi. You should well know Delphi.

Before using a card control, you need to configure the resources of card including port, interrupt and DMA via DevInst.exe. You need to register MFC before using a control, such as regsvr32 C:\windows\system\mfc42.dll.

### 4.1 Develop Delphi Program

#### 4.1.1 Add EVOC ActiveDAS Control to Controls Panel

Before establishing an application program, you must add the control to the controls panel of Delphi. In Delphi, it is OK to add one time while in Visual Basic and Visual C++, you need to add it many times when establishing many programs. After adding it, the “import units” of “pascal” has been established. It declares the properties, methods, and events of a control.

Steps for adding the control to the panel:

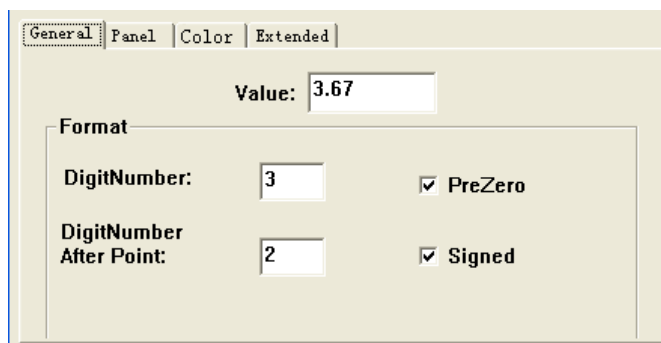
- (1) In Delphi, select “Component” menu, and select “Import ActiveX Control.....”, and then a dialog box for all registered controls appear.
- (2) Select controls upon your demand and click “Install.....”
- (3) Under path \Import, head files are established.
- (4) Click “OK” in “Install” dialog box
- (5) In the dialog box below, click “YES”. Delphi can automatically compile the controls. Please remember register the controls first.

## Chapter 5 Detailed Introduction of EVOC ActiveDAS

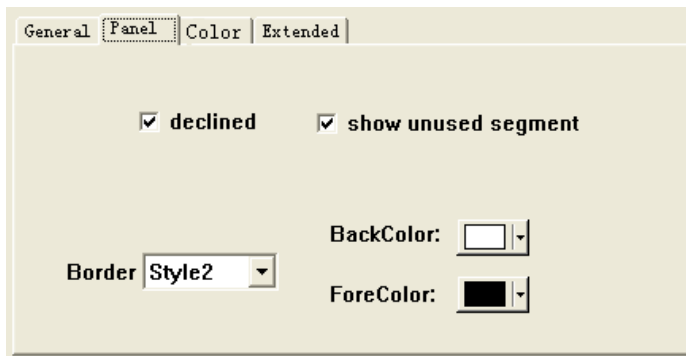
Before using a card control, you need to configure the resources of card including port, interrupt and DMA via DevInst.exe. You need to register MFC before using a control, such as regsvr32 C:\windows\system\mfc42.dll.

### 5.1 E7Segment Control

7Segment is used for digit display.



“Value” represents the display value. “DigitNum” represents the digit number of value. “DigitNum After Point” represents the digit number after point. “Signed” is used to enable symbol display.

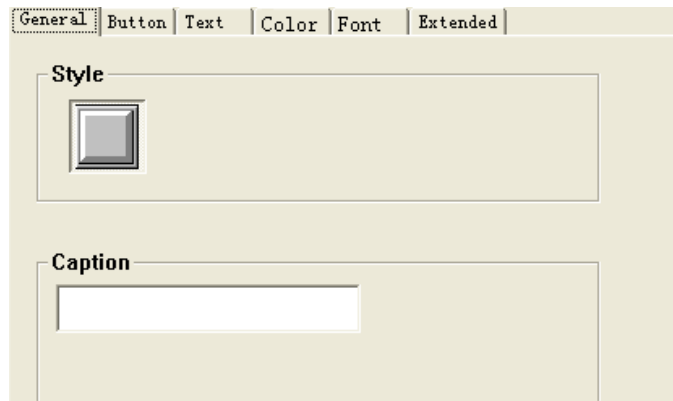


“show unused segment” represents showing unused segment with shadow. “Border” is used to set control border. If you want to change the display value of this control, you can directly change the property of “Value”. The control can automatically refresh the display value.

### 5.2 Boolean Control



Boolean control is used for the display and operation of switch. It is convenient for digital input and output. “Caption” is the title of control.



General | Button | **Text** | Color | Font | Extended

**State**

StateDisplay

**Index**

Index Display

Index Sequence: From Right ▾

It reaches up to 50 digits and is set by Button Num. “Distance” represents the distance between buttons. “Direction” represents the sequence direction of button. “Only One On Flag” represents only one button stay “open” status, and others stay “close”.

General | **Button** | Text | Color | Font | Extended

Button Number: 1

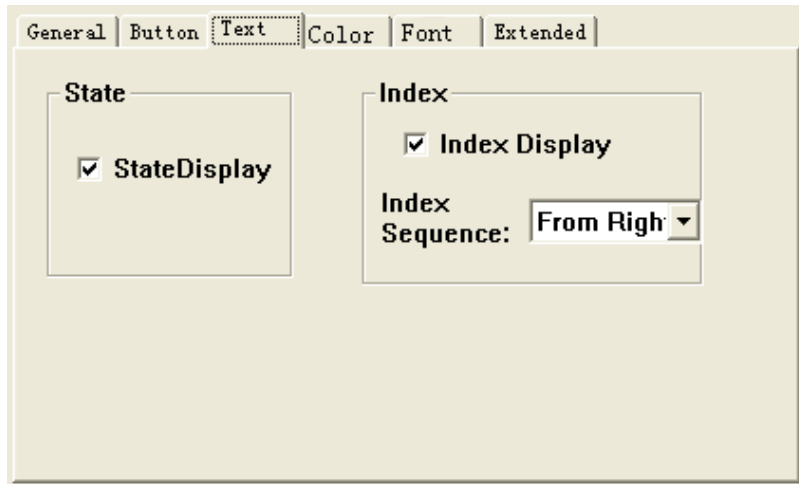
Distance: 0

Direction: Horizor ▾

Only One On Flag



“Index Sequence” represents the direction of index. “Index Display” is used to display the index. “State Display” is used to display on/off status.

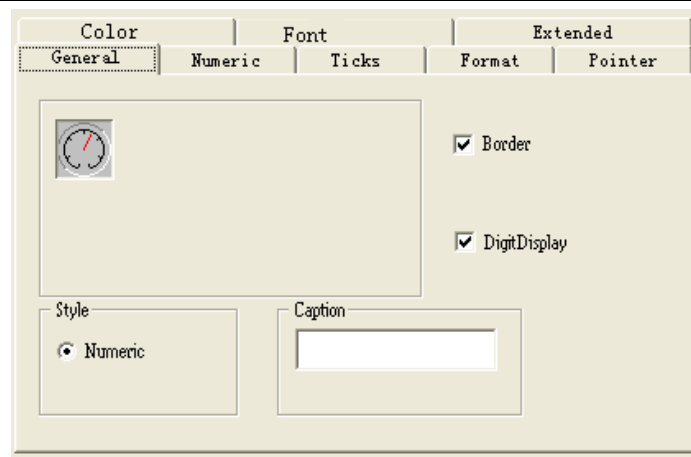


When clicking the button of the control, the value will be changed. You can add “click” event to realize the function of control and data acquisition. For the code, please refer to the sample of DemoUI10DIO. This sample is based on PLG-812PG (Multifunction Card) of our company.

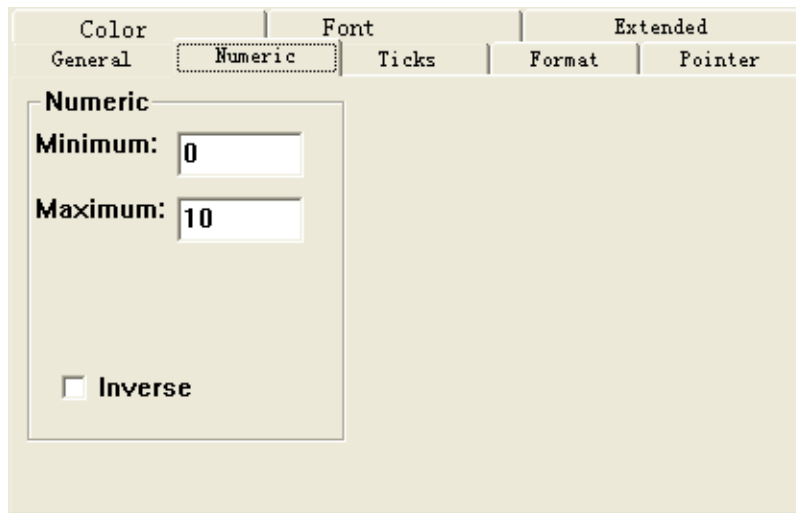


### 5.3 Knob Control

Knob control is used for data output and acquisition, including analog and digital



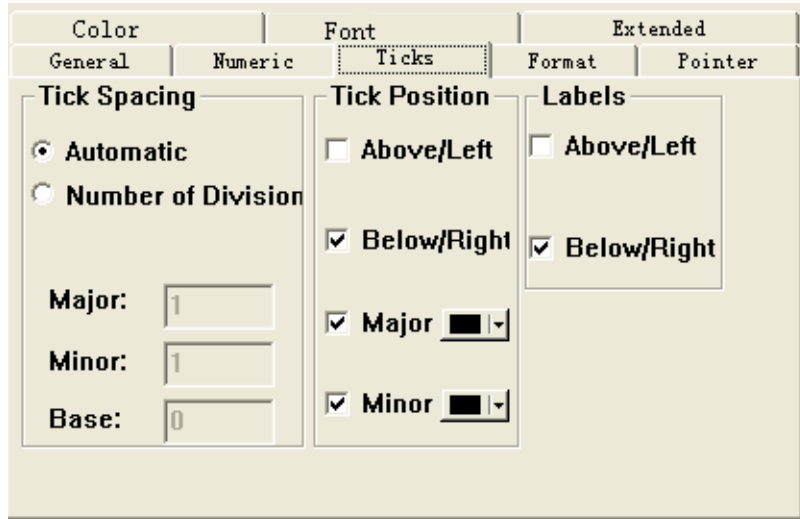
“Caption” is the title of control. “Border” is used to set control border.  
 “DigitDisplay” represents the current value of control.



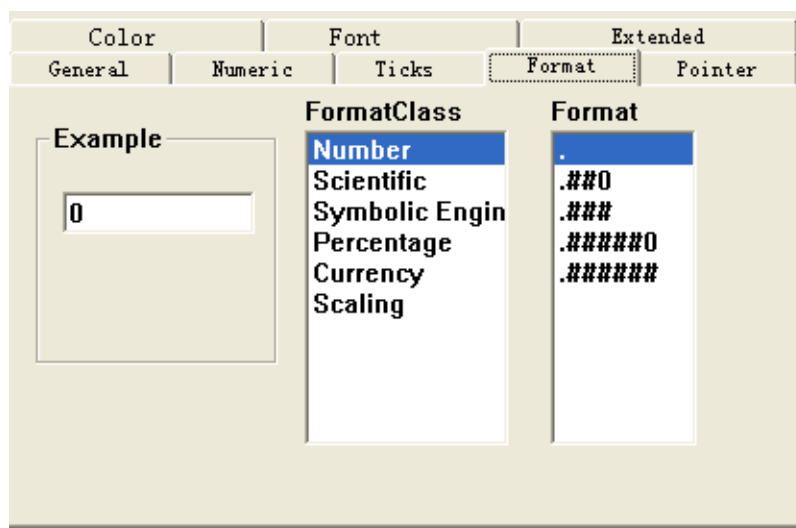
“Minimum” is the min. value of scale. “Maximum” is the max. value

of scale. “Inservse” represents the interchange between max. value and min. value.

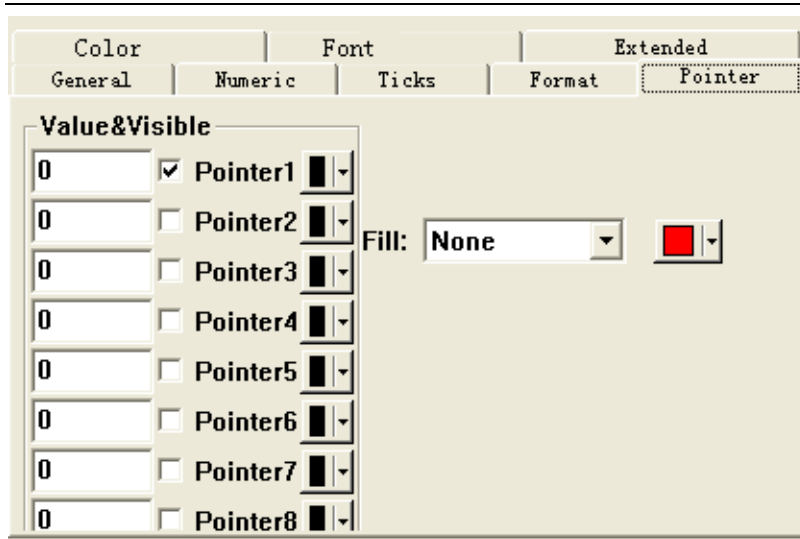
The scale has two types: Automatic and Number of Division. Under “Automatic”, the system will automatically allocate the scale; under “Number of Division”, you need to define it. “Static” represents the location and color of scale mark. “Labels” represents the location of the scale label.



“Format” defines many kinds of data format. The scale label and current value are displayed depending on the format that you select. “Example” is used for checking the instant effect of the format. The format falls into six categories: Number, Scientific, Symbolic Engineer, Percentage, Currency, and Scaling. Every category has many types. Here we do not introduce the formats of every category.



It can reach up to eight pointers (eight channels). The color and value of every pointer can be set. You had better set the value of every pointer rather than set the value of the control. The value of control equals to that of the pointer you set.



For its use method, you can refer to the sample of DemoUI10DA and DemoUI10AD. “Format” is used for the format of data display. “Numeric” is used to set the range of the dial. “Tick Spacing” is in “Ticks Property Page”.

#### 5.4 Slide Control

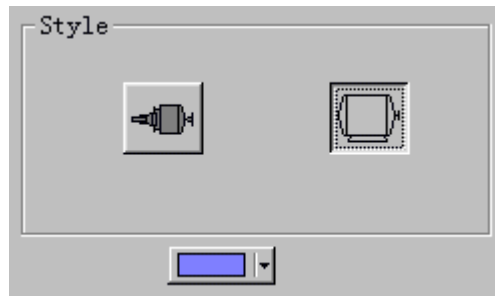


The property of Slide control is the same as that of Knob control. For its use method, you can refer to the sample of DemoUI10DA and DemoUI10AD.

#### 5.5 Motor Control



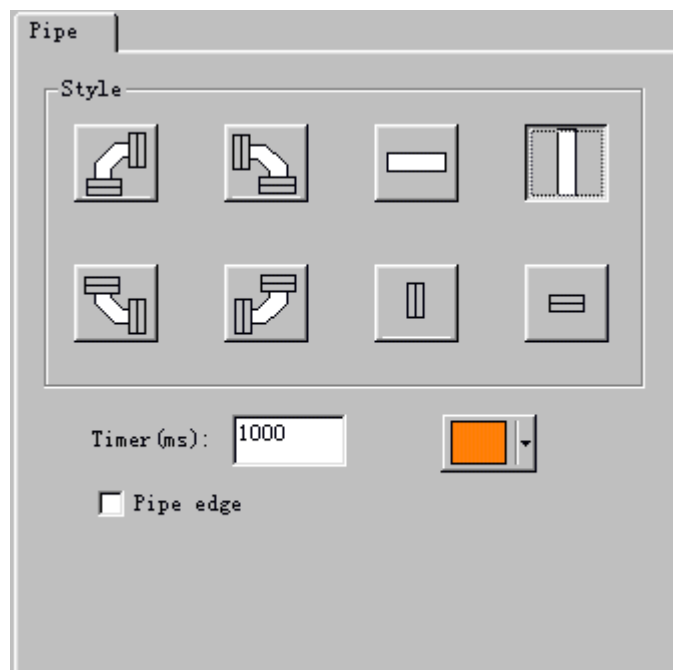
There are two kinds of Motor control:



The color can be set. For its use method, you can refer to the sample of Synthesize.

### 5.6 Pipe Control

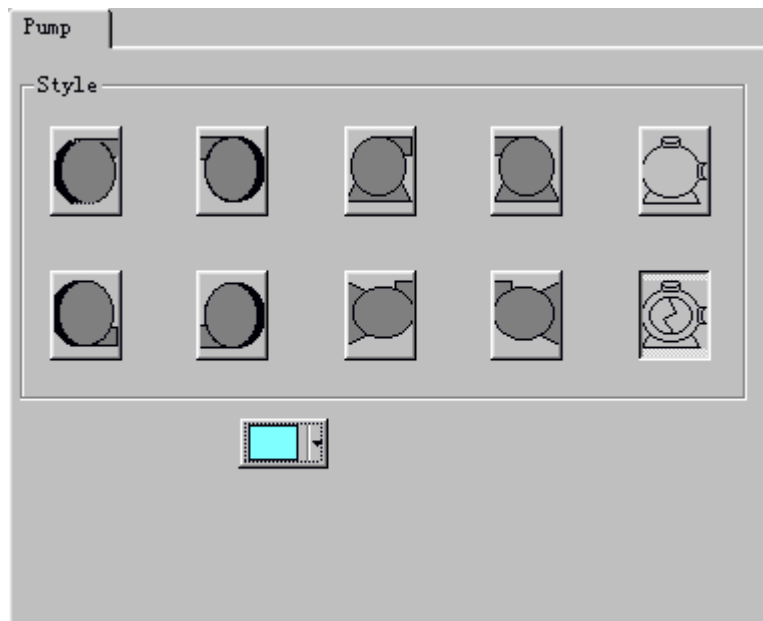
There are eight kinds of Pipe control. “Timer” is used to set its color and the glittering frequency (time unit: millisecond). For its use method, you can refer to the sample of Synthesize.



### 5.7 Pump Control



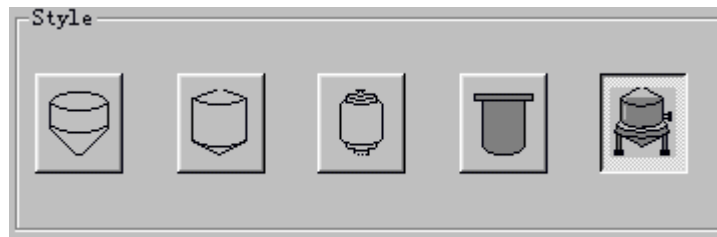
Pump control has ten types. The color can be set. For its use method, you can refer to the sample of Synthesize.



### 5.8 Tank Control



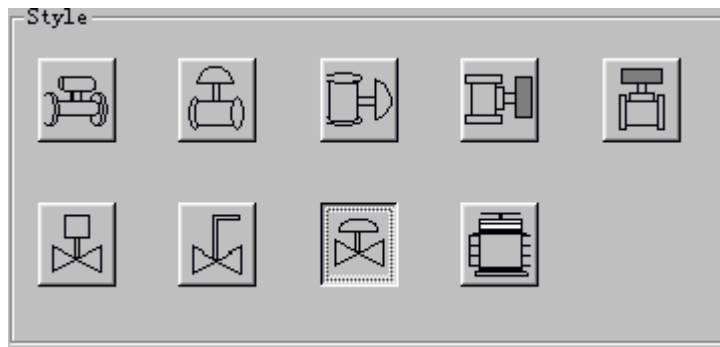
Tank control has five types. The color can be set. For its use method, you can refer to the sample of Synthesize.



### 5.9 Valve Control



Valve control has nine types. The color can be set. For its use method, you can refer to the sample of Synthesize.



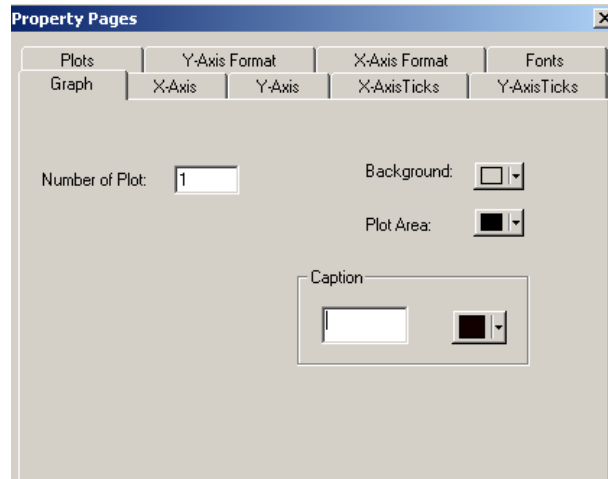
### 5.10 Graph Control



Graph is a complex control that outputs the historic data. "Number of Plot" represents the number of output. "Caption" is the title of control. "Background" is the background color of control. "Plot Area" is the

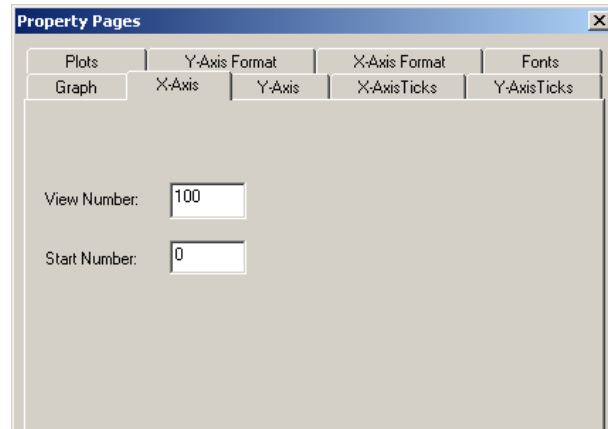


color of the output area.

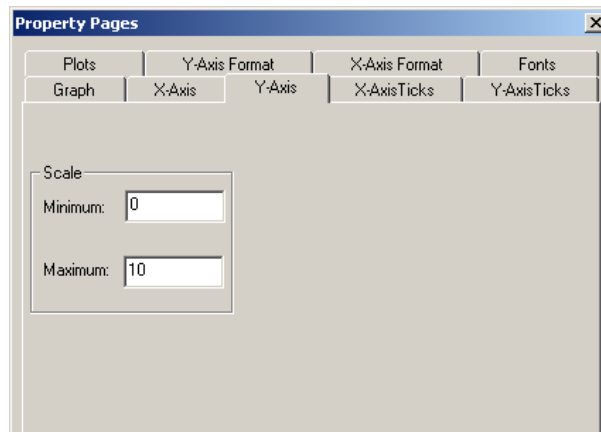


“View Number” represents the number of data showing on the X-Axis.

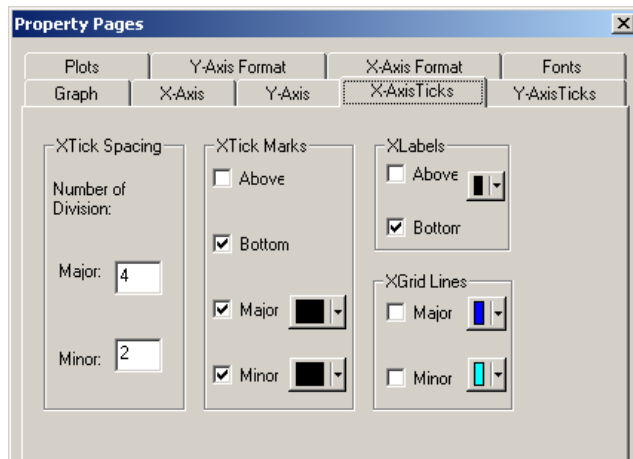
“Start Number” represents the start of data display.



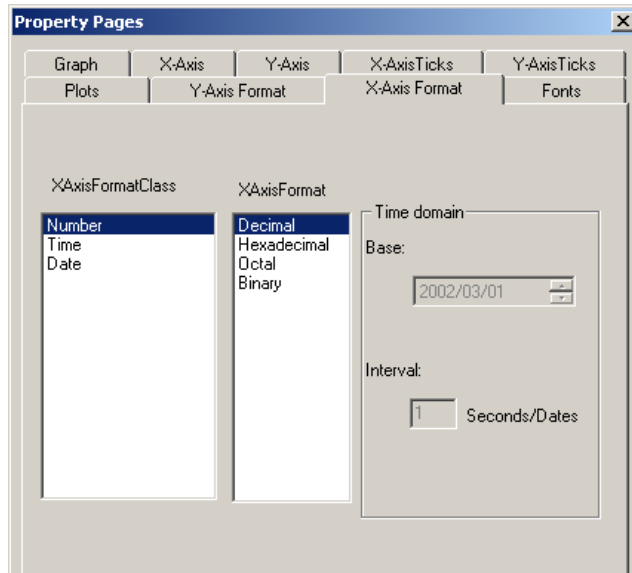
“Minimum” and “Maximum” represents the min. value and max. value on Y-Axis.



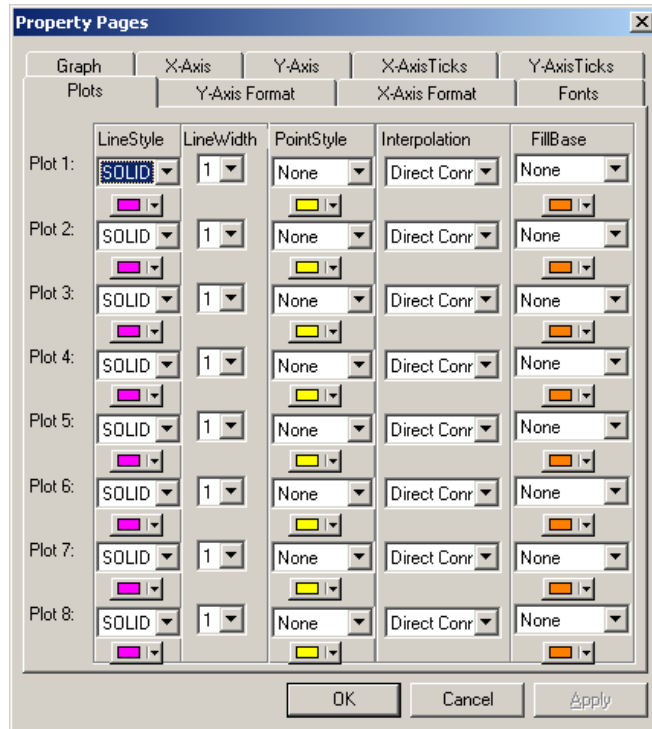
“X-AxisTicks” represents the characteristics of scale on the X-Axis. “XTick Marks” represents the position and color of scale mark. “Xlabels” represents the position of scale label. “Xgrid Lines” is used to set the grid lines and its color. “Y-AxisTicks” represents the characteristics of scale on Y-Axis. For the setup of Y-Axis, you can refer to that of X-Axis.



The setup of “Y-Axis Format” is the same as that of “Format” in Knob control. “X-Axis Format” represents the format of scale on X-Axis. It falls into three categories: Number (including decimal, hex, octal, and binary), Time (including two types), and Date (including two types). “Base” represents the base time. “Interval” represents the number of data interval (Unit: second or day).



It can reach up to eight outputs. The output data must be kept in a same group. “Plots” is used to define the output styles which include LineStyle, LineWidth, PointStyle, Interpolation, FillBase, and Color.



For program, you can refer to “GraphDemo”.

Steps:

1. Establish double-type array “Gdata”.
2. Input value
3. Output data, Graph1.PlotGraphs GData.

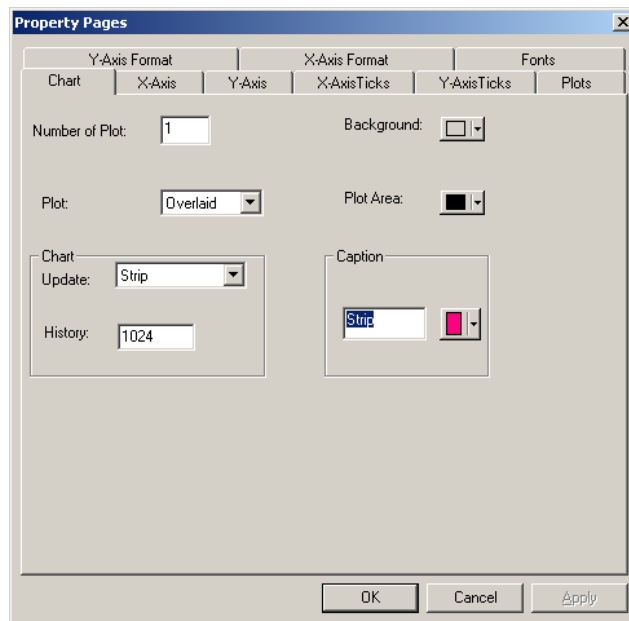
You can see the output through “ScrollBar”. You can set many outputs.

“ClearGraphs” is used to clear the output.

### 5.11 Chart Control



Chart control is used to draw the real-time curve and can reach up to eight outputs. The setup of Chart control is the same as that of Graph control. In Chart control, “Plot” has two types: Overlaid and Stacked. The update modes include Strip, Scope, and Sweep. “History” represents the buffer capacity for data storage. “Caption” is the title of control. “Background” represents the background color of control. “Plot Area” is the color of output area. “Number of Plot” represents the number of output line.



---

For program, you can refer to the sample of ChartDemo.

The steps:

1. Set double-type array value. The length of the array equals to the number of output. Even if the length is “1”, users need to set the form of array.
2. Use “InitControl” to initialize the control, and then initialize the array.
3. Output data in the event “Timer” (leave space for memory), such as Chart1.PlotCharts Value
4. Re-input value in the event “Timer”, please note the order of (3) and (4).
5. After program, please use “ReleaseControl” to release the memory.

#### 5.12 DASDevice Control

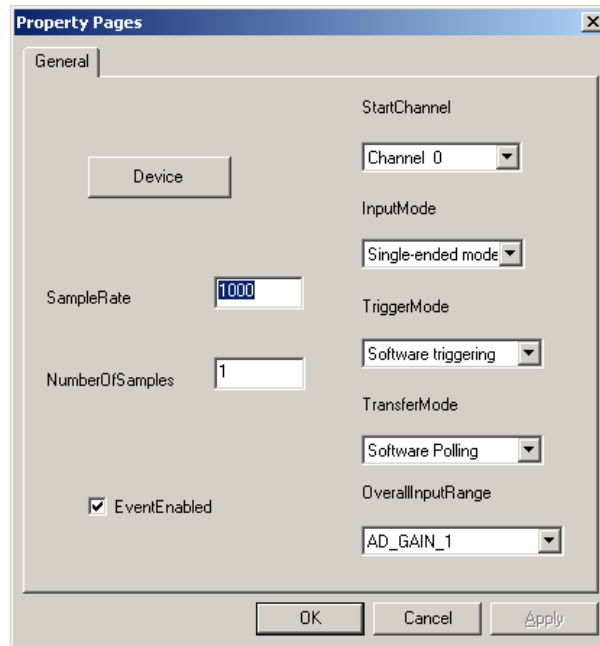


DASDevice control is used to select device and can obtain the device number. It has two properties: DeviceName (read-only) and DeviceNumber. Before using devices, you should configure their resources via DevInst.exe. When configuring, please pay attention to resource interrupts. The resource of ISA-HG16AD card should be configured by adding new hardware to “Devices Manager” in “Control Panel”.

#### 5.13 DASAI Control



DASAI control is used for analog input.



The above image is “Property Page” of DASAI control. You can use “Device” or “DASDevice” to select devices. “EventEnable” is used to set the trigger events after data acquisition. “TransferMode” has three modes for transfer: Software Polling, Interrupt, and DMA. Different devices have different modes. The trigger mode has three types: Software, External, and Clocking. “OverallInputRange” or “AdGain” is used to set the input range of card. For the detailed information, please refer to User Manual of the card. “StartChannel” represents the channel for acquisition. “InputMode” has two kinds of properties: single input and double input. Before data acquisition in large



quantities, please first set the SampleRate and NumberOfSamples.

It has two events. When “EventEnable” is true, “OnEventRaw” event occurs after data acquisition. The original data can be obtained by “Data”. “DataCount” represents the number of data acquisition. When stopping acquisition or after finishing acquisition, “OnTerminated” event occurs.

It has four methods. After all properties setup, use “OpenDevice” to open the device. If “OpenDevice” is enable, it returns True, otherwise returns False. “RawInput” represents the input function of original data. Only one data can be obtained by using “RawInput”. For data acquisition in large quantities, please use “AcquireStart” and “AcquireStop” and obtain a number of data in event “OnEventRaw”. For its use method, you can refer to DASAIDemo.

Steps:

1. Add EVOC DASAI and EVOC DASDevice to it.
2. Right click DASAI control and select “Characteristic”
3. Set the properties and use “Device” button or DASDevice control to select the device.
4. Open DASAI.
5. For one acquisition, you should use “RawInput”.
6. For acquisitions in large quantities, you should use “AcquireStart”.

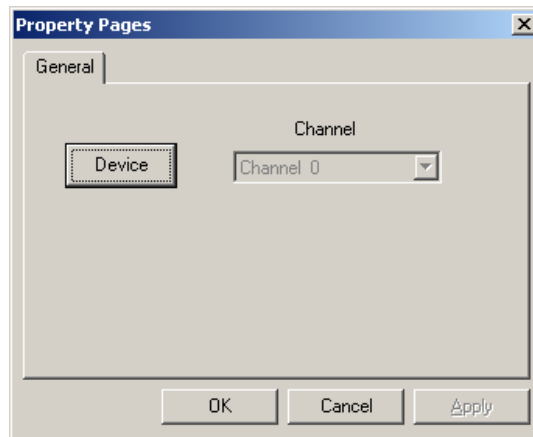
#### 5.14 DASAO Control



It is used for analog output.

Before using a control, you need to configure the resources of card including port, interrupt and DMA via DevInst.exe.

“Property Page” is shown below:



You can use DASDevic control or button on “Property Page” to select device. “Channel” represents the channel of analog and can be set via “Property Page”. “RawOutput” represents the output of original data. Use analog value 0~44095 as a parameter of function to output. If output is executed, it returns True, otherwise returns False.

You can refer to the sample of DASAODemo.

For types of card supported by this control, please refer to Chapter 7.

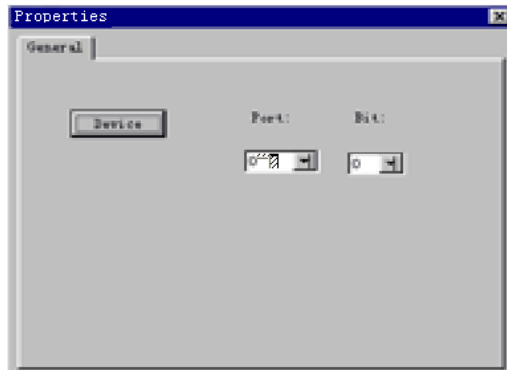
### 5.15 DASDI Control



It is used for digital input.

Before using a control, you need to configure the resources of card including port, interrupt and DMA via DevInst.exe.

“Property Page” is shown below:

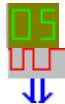


It has two important properties: Port and Bit, e.g. 720 has 32 channels (32-bit); the port is "4" (0~3), each port has 8 bits, 4 x 8=32.

After properties setup, use "OpenDevice" to start program. "OpenDevice" has two methods: "BitInput" and "ByteInput". Under "BitInput", you need to set Port and Bit. It corresponds to a certain channel of card, e.g. Port=0, Bit=4, which means it corresponds to the fourth channel of card. If Bit=1, "BitInput" returns True, otherwise returns False. Under "ByteInput", you need to set Port only. One channel corresponds to one byte, that is 8-bit (8-channel).

You can refer to the sample of DASDIDemo.

For types of card supported by this control, please refer to Chapter 7.



#### 5.16 DASDO Control

It is used for digital output.

Before using a control, you need to configure the resources of card including port, interrupt and DMA via DevInst.exe. The setup of its properties is the same as that of DASDI control.

“ByteOutput” represents the digital output of byte. Please set the Port. If output is executed, it returns True, otherwise returns False. “BitOutput” is the digital output of bit. Please set the Port and Bit. If output value is “1”, the Boolean parameter is True, otherwise it is False. If output is executed, BitOutput returns True, otherwise returns False. This control also has two properties: “BitReadBack” and “ByteReadBack”. Their setups are the same as those of “BitOutput” and “ByteOutput”. Maybe some cards cannot support read back. For more information, please refer to the User Manual of card. “BitReadBack” and “ByteReadBack” are not applied for all cards with digital output.

You can refer to the sample of DASDODemo.

For types of card supported by this control, please refer to Chapter 7.

#### 5.17 DASCounter Control



Before using a control, you need to configure the resources of card including port, interrupt and DMA via DevInst.exe.

It is used to support ISA cards. It has two important properties: “Channel” and “PresetValue”. “Channel” represents the channel of a counter. 8254 chipset has three counters (channels). “PresetValue” represents the preset value of a counter. “EnableCounter” is used to enable or disable counter. It is enabled when the parameter is True; it is disabled when the parameter False. When “CounterValue” is read, the

---

control will show the value of “CounterValue”.

### 5.18 DASPulse Control

Before using a control, you need to configure the resources of card including port, interrupt and DMA via DevInst.exe.

It is used for pulse output of counter. The property of “Channel” is the same as that of DASCounter control. “PulseWidth” provides the number of period of clock signal. If clock signal is 10KHz, the period is 0.1 millisecond. If “PulseWidth” is 4, the period is 0.4 millisecond. Correspondingly, the output frequency is 2.5KHz. For detailed information, please refer to 8254 chipset. “Channel” and “PulseWidth” can be set via “Property Page”. After setup, you can use “EnablePulseOut” to enable or disable output.

### 5.19 DASAlarm Control




It is a control for alarm. “ResetAlarm” is used for reset alarm control. “AlarmMode” has two options: “momentary” and “Latched”. Under “Momentary”, when it outstrips “HiLimit” or “LoLimit”, the control stays the alarm status. As long as “EnableAlarm” is True at this time, it can activate some events in OnHiAlarm, OnHiToNormal, OnLoAlarm, and OnLoToNormal. When the value returns to its normal status, the control will automatically get out of alarm status. Under “Latched”, it still stays the alarm status even if it returns to normal value. But if the “RetriggerAlarm” is True, the alarm can be closed. “Value” represents the real-time data value for DASAlarm. From normal status to “HiLimit”, the event “OnHiAlarm” occurs. From “HiLimit” to normal status, the event “OnHiToNormal” occurs. From normal status to

“LoLimit”, the event “OnLoAlarm” occurs. From “LoLimit” to normal status, the event “OnLoToNormal” occurs.

#### 5.20 PCL731A Control



It is used for digital input/output of PCL731 (A) card. Its properties are shown below:

“Init” represents the initialized function. “InData” represents the value of a certain channel. “OutData” represents output digital. “ConfigCHN” represents the configuration  a certain channel. It has two parameters including “Channel” and “config\_value”. “config\_value” represents the configuration value of register. “SetINTop” represents the operation mode of interrupt. “Directory” represents the direction of input or output. “Buffer” is the buffer of data storage. “INTStart” represents the start of interrupt. “INTStop” represents the stop of interrupt. “GetINTStatus” represents the status of getting interrupt. “interrupt\_count” represents the number of transfer data.

#### 5.21 PCL733 Control

It is used for digital input of PCL733 card. Its properties are shown below:

“Init” represents initialized function. “InData” represents the value of a certain channel. “SetINTop” is used to set interrupt. “Directory” represents the direction of input or output. “Buffer” is the buffer of data storage. “INTStart” represents the start of interrupt; “INTStop” represents the stop of interrupt. “GetINTStatus” represents the status of getting interrupt. “interrupt\_count” represents the number of transfer data.

### 5.22 PCL734 Control



It is used for digital output of PCL734 card. Its properties are shown below:

“Init” represents the initialized function. “OutData” represents the output digital.

### 5.23 PCL836A Control



Counter or timer, it is used for PCL836 (A) card.

“DeviceOpen” is used to open a device, supporting multi devices. It has two parameters: device number and base address. Please first enable “DeviceOpen” among all the parameters. “DeviceClose” is used to set the close of device. “Counter” represents counter number 0-5. “MaxInFreq” represents max. frequency of counter. When its value is “zero”, the counter is invalid. Its max. value is 312kHz. “CounterEventStart” represents the start value of external events. CounterEventRead is used to read out the measure value of external events. “Counter” is counter number, 0~5. “Count” is the counter value. “CounterFreqStart” represents the start frequency of counter. “Counter” is the counter number, 0~5. “FreqLevel” represents the level of frequency. When its value is “0”, the MAX is 1Hz~65kHz; when its value is “1”, the MAX 10kHz~650kHz; when value is “2”, MAX 100kHz~6500kHz. “CounterPulseStart” represents the start of pulse output. The time unit of “Period” is millisecond. “Upcycle” is the length of high level (time unit: millisecond). “FreqOutStart” represents the start of frequency output. “Counter” represents the counter number, 0-5. “Fout” represents the frequency value of output. “FreqOutReset” represents the reset of frequency output. “CounterReset” represents the

reset of counter. “DioWritePortByte” represents the byte of digital write port. “State” represents the write value. “DioWriteBit” represents the output of digital write. “State” represents the output value, “0” or “1”. “DioGetCurrentDOByte” represents the byte of the current output. “DioGetCurrentDOBit” represents the bit value of the current output. “State” represents the readback value, “0” or “1”. “DioReadPortByte” represents the byte of digital read port. “State” represents the read value. “DioReadBit” represents the write input of digital. “State” represents the read value, “0” or “1”.



## Chapter 6 Samples and Points for Attention

### 6.1 The sample of DemoUI10DIO

This sample is based on the multi-function card PLC-812PG to realize the I/O of digital quantity. To get it running, you have to install its driver DLL2 first. The upper Boolean control is for output and has 8 channels. The initial value is 00000000 (binary system). The nether Boolean control is for data input and also has 8 channels.

1) Add the header file Dll2.bas of the driver PLC-812PG to the process.

2) Define a global variable Dim PCL\_BaseAddress As Long

3) Initialize 812PG

```
Private Sub Form_Load()  
    'Initialize 812PG  
    PCL_BaseAddress = &H220  
    If W_812PG_Initial(CARD_1, PCL_BaseAddress)  
    <> ERR_NoError Then  
        MsgBox "PCL-812PG Initialization Error!"  
        Unload Form1  
    End If  
End Sub
```

4) The programming of collection is simple. In the Timer event:

```
Private Sub Timer1_Timer()  
    Dim di_data As Byte
```

```
W_812PG_DI di_low_byte, di_data 'collect digital
quantity
```

```
If Boolean2.Value <> di_data Then
```

```
    Boolean2.Value = di_data    'make the control
display digital quantity
```

```
End If
```

```
End Sub
```

5) The Click event of Boolean has the code of output digital quantity

```
Private Sub Boolean1_Click()
```

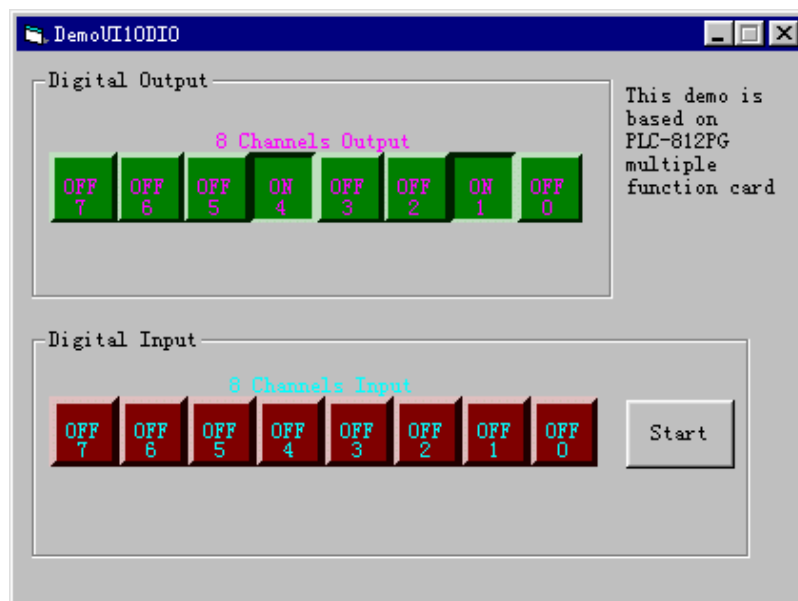
```
    Dim do_data As Byte
```

```
    do_data = Boolean1.Value
```

```
    W_812PG_DO DO_Low_BYTE, do_data 'Output the
digital quantity according to the control value
```

```
End Sub
```

This sample mainly describes the usage of Boolean control. Users can refer to it while programming. The following is an interface of it:



## 6.2 The sample of DemoUI10DA

This sample is based on the multi-function card PLC-812PG to realize the output of analog, i.e. digital-analog conversion. To get it running, you have to install its driver DLL2 first. It has two channels. Only by using the mouse to drag the pointer of the Knob control or the Slide control will the analog be output through the corresponding channel. The drag to a certain pointer should be written in MouseMove event.

1) Add the header file Dll2.bas of the driver PLC-812PG to the process.

2) Define a global variety Dim PCL\_BaseAddress As Long

3) Initialize 812PG

```
Private Sub Form_Load()
```

```
'Initialize 812PG
```

```
PCL_BaseAddress = &H220
```

```
If W_812PG_Initial(CARD_1, PCL_BaseAddress) <>
```

```
ERR_NoError Then
```

```
MsgBox "PCL-812PG Initialization Error!"
```

```
Unload Form1
```

```
End If
```

```
End Sub
```

4) Convert digital and analog via moving the pointer in Knob

```
Private Sub Knob1_MouseMove(Button As Integer,  
Shift As
```

```

Integer, x As Single, y As Single)
If volt1 <> Knob1.Pointer1Value Then
    volt1 = Knob1.Pointer1Value
    Slide1.Pointer1Value = Knob1.Pointer1Value
    da_data1 = volt1 / 5 * 4095 ' D/A Range == -5V
to +5V
    W_812PG_DA DA_CH_1, da_data1 'Output
the analog in channel 1 of 812PG
End If
If volt2 <> Knob1.Pointer2Value Then
    volt2 = Knob1.Pointer2Value
    Slide2.Pointer1Value = Knob1.Pointer2Value
    da_data2 = volt2 / 5 * 4095 ' D/A Range == -5V
to +5V
    W_812PG_DA DA_CH_2, da_data2 'Output the
analog in channel 2 of 812PG
End If
End Sub
5) Convert digital and analog via moving the pointer in
Slide control
Private Sub Slide1_MouseMove(Button As Integer,
Shift As
Integer, x As Single, y As Single)
If volt1 <> Slide1.Pointer1Value Then
    volt1 = Slide1.Pointer1Value
    Knob1.Pointer1Value = Slide1.Pointer1Value
    da_data1 = volt1 / 5 * 4095 ' D/A Range ==
-5V to +5V

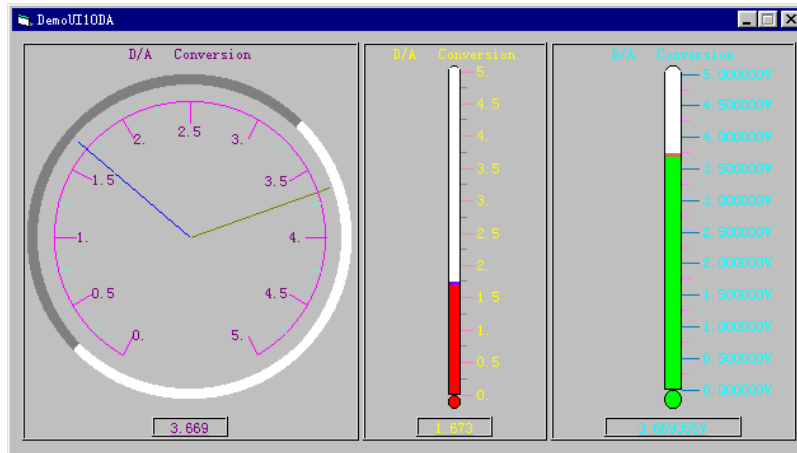
```

```

W_812PG_DA DA_CH_1, da_data1 'Output
the analog in channel 1 of 812PG
End If
End Sub

```

Users can refer to it while programming. The following is an interface:



### 6.3 The sample of DemoUI10AD

This sample is based on the multi-function card PLC-812PG to realize the conversion between analog and digital and collected data. To get it running, you have to install its driver DLL2 first. It has two channels.

- 1) Add the header file Dll2.bas of the driver PLC-812PG to the process.
- 2) Define a global variety Dim PCL\_BaseAddress As Long
- 3) Initialize 812PG

---

```
Private Sub Form_Load()
    'Initialize 812PG
    PCL_BaseAddress = &H220
    If W_812PG_Initial(CARD_1, PCL_BaseAddress)
```

◇

```
    ERR_NoError Then
        MsgBox "PCL-812PG Initialization Error!"
        Unload Form1
```

```
End If
```

```
W_812PG_AD_Set_Gain AD_GAIN_1
W_812PG_AD_Set_Mode AD_MODE_1
ad_count = 1
ad_start_ch = 0
ad_stop_ch = 1
End Sub
```

4) Write the code of collected data in Timer event.

```
Private Sub Timer1_Timer()
    Dim ch, ad_data As Long
    Dim err As Integer
    Dim volt As Single
    Dim buffer(15, 99) As Integer
```

```
W_812PG_CLR_IRQ 'Clear the interruption
```

```
For ch = ad_start_ch To ad_stop_ch
```

```
W_812PG_AD_Set_Channel ch
```

```
err = W_812PG_AD_Aquire(ad_data)
```

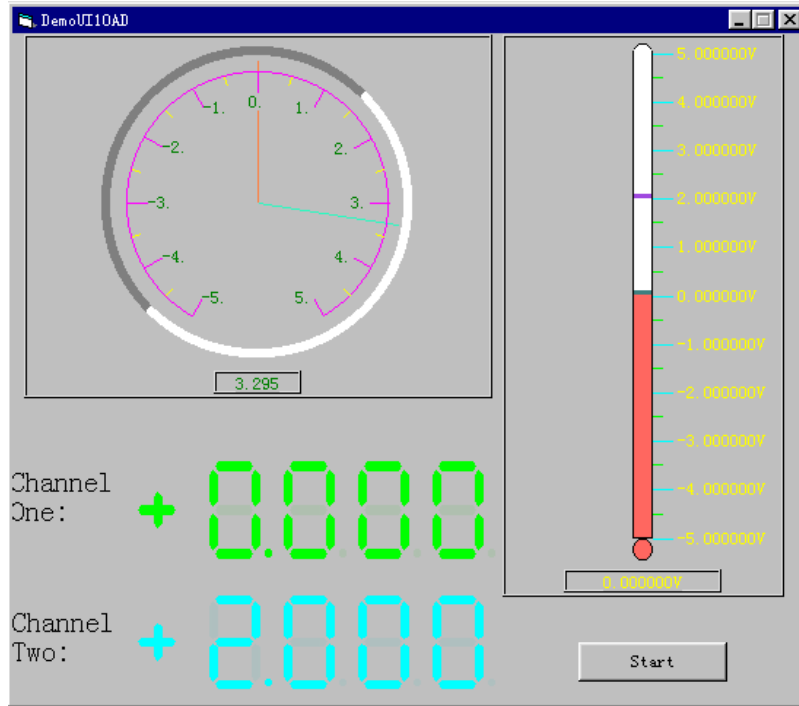
```
If err <> ERR_NoError Then
```

```
    MsgBox "PCL-812PG Error Number :" + Str(err)
```

```
Else
    volt = ad_data
    volt = 10 * volt / 4095 + (-5)
    If ch = 0 Then      'Display the collected data of
channel 1 via control
        Knob1.Pointer1Value = volt
        Slide1.Pointer1Value = volt
        E7Segment1.Value = volt
    Else
        If ch = 1 Then      'Display the collected data of
channel 2 via control
            Knob1.Pointer2Value = volt
            Slide1.Pointer2Value = volt
            E7Segment2.Value = volt
        End If
    End If
End If
Next ch

End Sub
```

Users can refer to it while programming. The following is an interface:





#### 6.4 The sample of GraphDemo

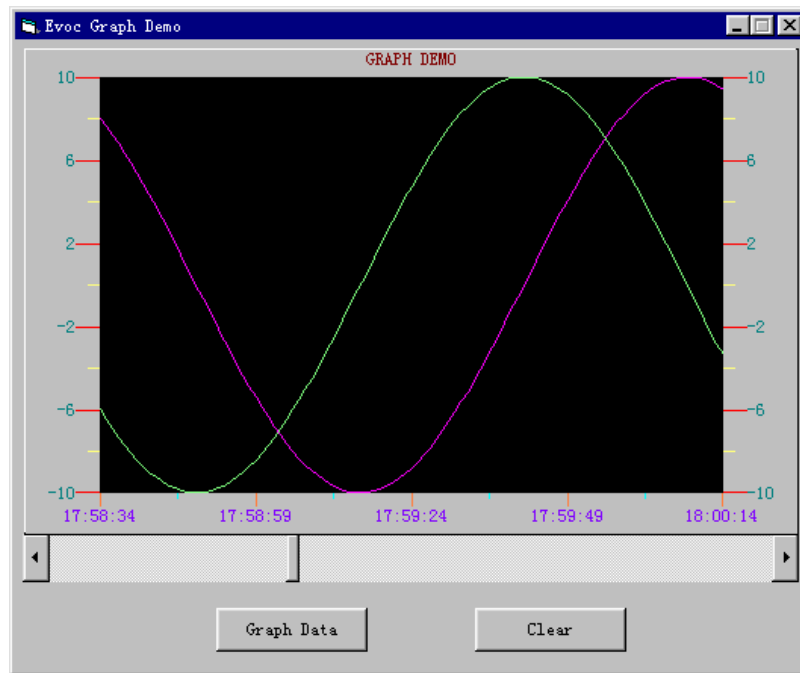
It simulates the output of two sinusoids.

1) Generate the data of sinusoid in Form\_Load().

```
Private Sub Form_Load()  
    Dim k As Integer  
    deltaX = 3.14159256 * 4 / 210  
    For i = 0 To 4095 Step 2  
        k = i / 2  
        GData(i) = (Sin(k * deltaX) * 10) 'Generate  
        sinusoidal data and reserve it in array  
        GData(i + 1) = (Cos(k * deltaX) * 10)  
        'Generate cosine curve data and reserve it in array  
    Next i  
End Sub
```

2) Graph1.PlotGraphs GData 'Output the curve according to the data in the array

The curve can be checked or cleared via scroll bar and the programming of this control is very convenient. Users can refer to it while programming. The following is an interface:



## 6.5 The sample of ChartDemo

Five Chart controls are listed and different modes of controls are displayed via stimulating sinusoids.

- 1) Define the value of one-dimension array of 2 after setting each property. Dim Value(0 To 1) As Double
- 2) Then use it in Form\_Load().

Initialize the control via “InitControl” (making room for memory) and assign an initial value to the array

```
Private Sub Form_Load()
```

```
'Initialize the control
```

```
Chart_Strip.InitControl
```

```
Chart_Scope.InitControl
```

```
Chart_Sleep.InitControl
```

```
Chart_Overlaid.InitControl
```

```
Chart_Stacked.InitControl
```

```
icount = 0
```

```
deltaX = 3.14159256 * 4 / 51
```

```
'Initialize the array
```

```
Value(0) = 5 + (Sin(icount * deltaX) * 5)
```

```
Value(1) = 5 + (Cos(icount * deltaX) * 5)
```

```
End Sub
```

- 3) Don't forget to release control (release memory) in Form\_Unload() via ReleaseControl.

```
Private Sub Form_Unload(Cancel As Integer)
```

```
'Make sure to release the memory in Chart control when the program is over
```

```
Chart_Strip.ReleaseControl
```

```
Chart_Scope.ReleaseControl  
Chart_Sleep.ReleaseControl  
Chart_Overlaid.ReleaseControl  
Chart_Stacked.ReleaseControl
```

```
End Sub
```

- 4) In the Timer event output the data from the array first and then vest the array with new value. Do not transpose the order.

```
Private Sub Timer1_Timer()  
Dim Scope_Value(0) As Double  
'Output data  
Chart_Strip.PlotCharts Value  
Scope_Value(0) = Value(0) - 5  
Chart_Scope.PlotCharts Scope_Value  
Chart_Sleep.PlotCharts Value  
Chart_Overlaid.PlotCharts Value  
Chart_Stacked.PlotCharts Value
```

```
Value(0) = 5 + (Sin(icount * deltaX) * 5)
```

'Generate the real-time data of sinusoid and reserve it in the array.

```
Value(1) = 5 + (Cos(icount * deltaX) * 5)
```

'Generate real-time data of cosine curve and reserve it in the array.

```
icount = icount + 1
```

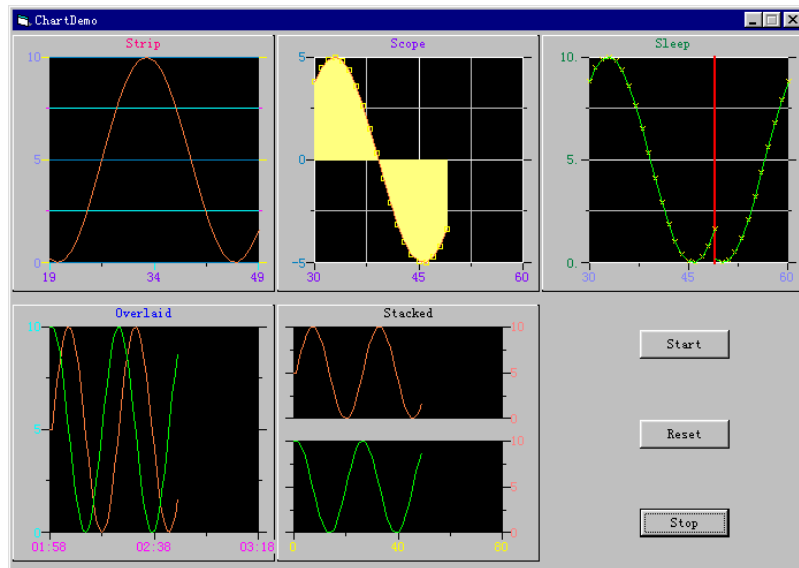
```
If icount = 102 Then
```

```
icount = 0
```

```
End If
```

End Sub

The programming of this control is very convenient and users can refer to it while programming. The following is an interface:



## 6.6 The comprehensive sample of Instrument

The sample uses analog temperature. The two knobs are used for adjusting the upper limit and lower limit of the scope and the former should be higher than the latter while adjusting. The thermometer shows the real-time temperature and the curve control shows the changing temperature.

- 1) Define a global variable array value of 3. The

Value(0) is measured value and Value(1) and Value(2) are upper limit and lower limit respectively.

Dim Value(2) As Double 'Define an array

2) Don't forget to initialize the control and array.

Private Sub Form\_Load()

Chart1.InitControl 'Initialize the Chart control

Value(0) = 24

Value(1) = 24

Value(2) = 27

End Sub

3) Don't forget to release the control.

Private Sub Form\_Unload(Cancel As Integer)

Chart1.ReleaseControl 'Release control

End Sub

4) In the Timer event, output the data first and then assign a value. Pay attention to the match between various data and transfer the data when necessary.

Private Sub Timer1\_Timer()

Chart1.PlotCharts Value

Slide1.Pointer1Value = Value(0)

Randomize Second(Time) 'Initialize the random seeds

Dim temp As Double

temp = ((35 \* Rnd) + 20)

While (Abs(temp - (Value(2) + Value(1)) / 2) > Abs(Value(2) - Value(1)))

temp = ((35 \* Rnd) + 20)

Wend

Value(0) = temp 'Generate data at random and simulate the temperature reading.

If Value(0) > 35 Then

Value(0) = 35

Else

If Value(0) < 20 Then

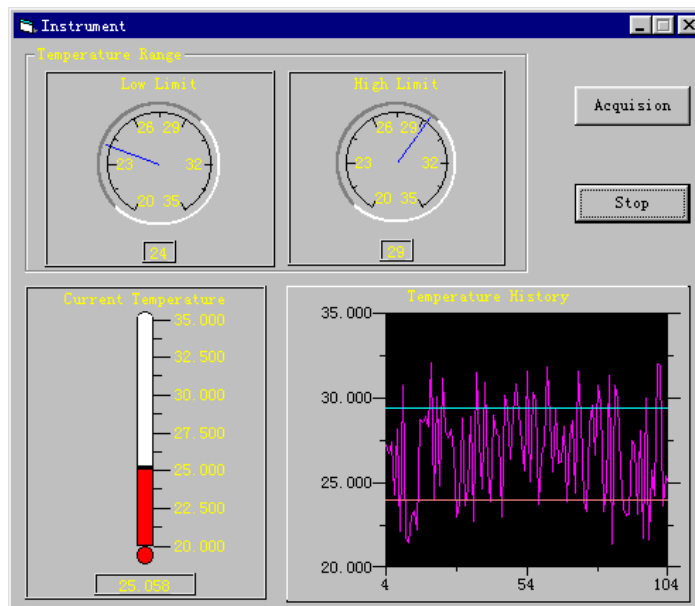
Value(0) = 20

End If

End If

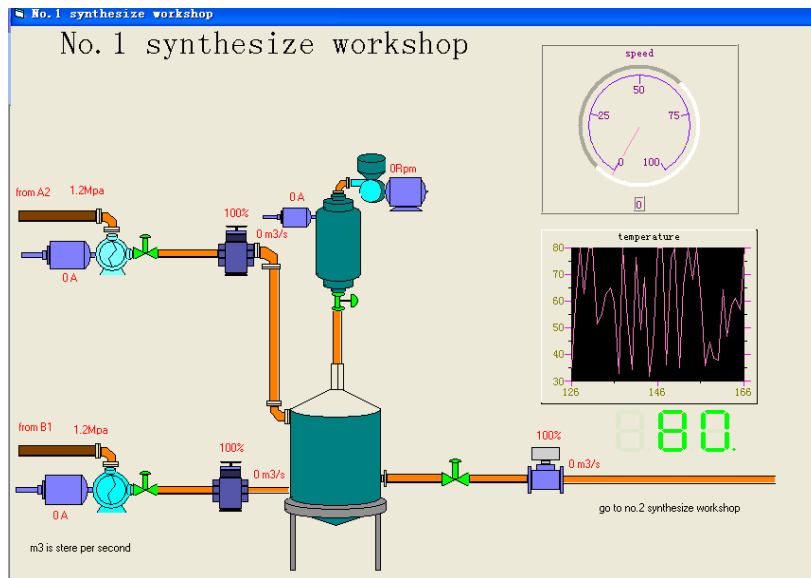
End Sub

Its programming is very simple and users can refer to it while programming. The following is an interface:



### 6.7 The comprehensive sample of Synthesize

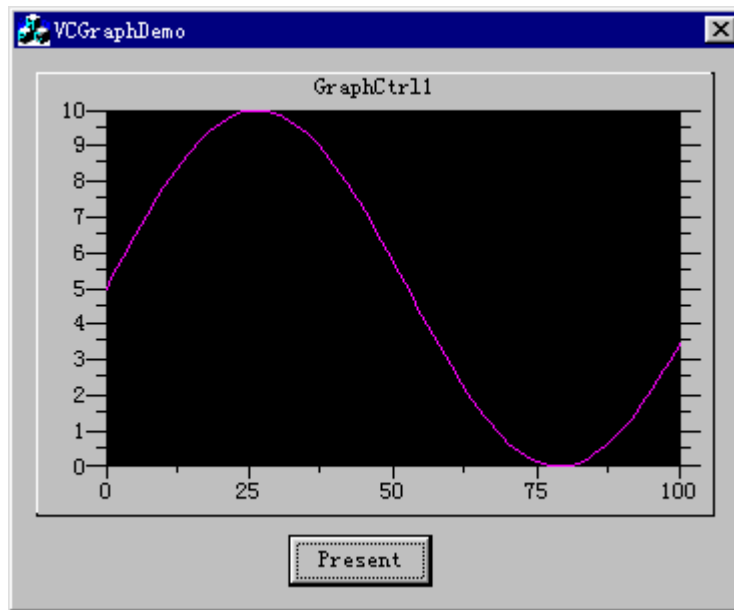
It is a simulating graph of a chemical synthesis workshop mainly using EvocEquipment serial control. Its programming is very convenient and users can refer to it while programming. The following is an interface:



### 6.8 The sample of VCGraphDemo

This sample is about how to vest value via VARIANT in the Visual C++ and mainly used in the controls of Graph and Chart. Users can refer to it while programming. The following is an interface:

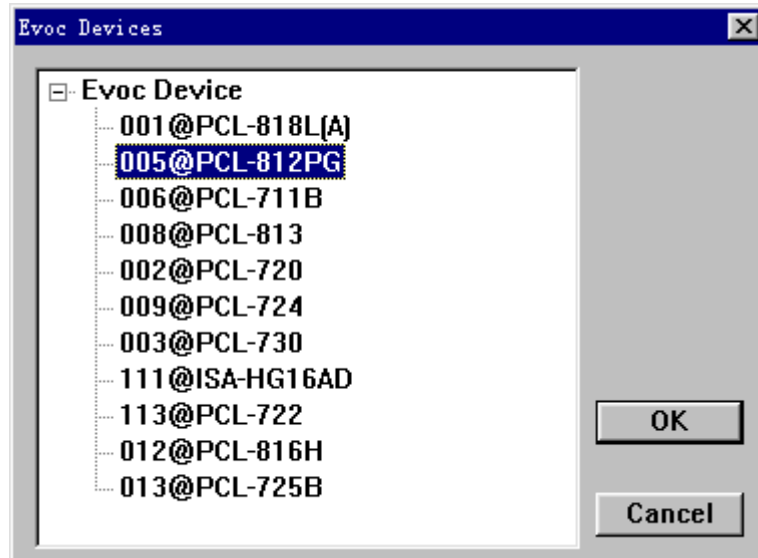




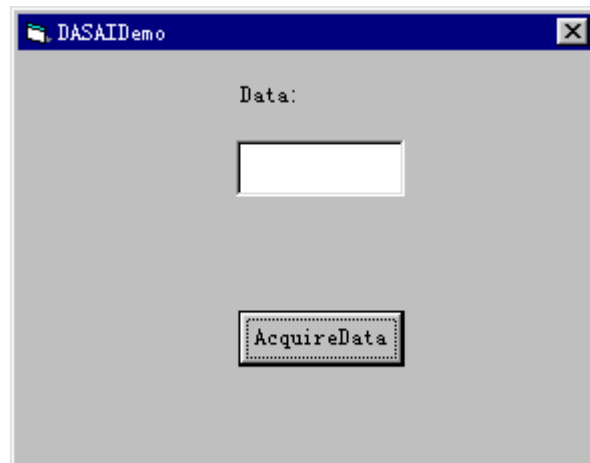
#### 6.9 The sample of DASAI Demo

It uses DASAI control to input analog and get the transferred digital quantity. At least one card supported by this control should be inserted and the detail can be referred to Chapter 7. Users have to have some basic knowledge on the hardware setting of the card and configure the card resources via DevInst.exe, including the port, interruption and DMA. You can find the detailed configuration method in Chapter 8, or this sample will not work normally.

First, select a supported card in the following way:

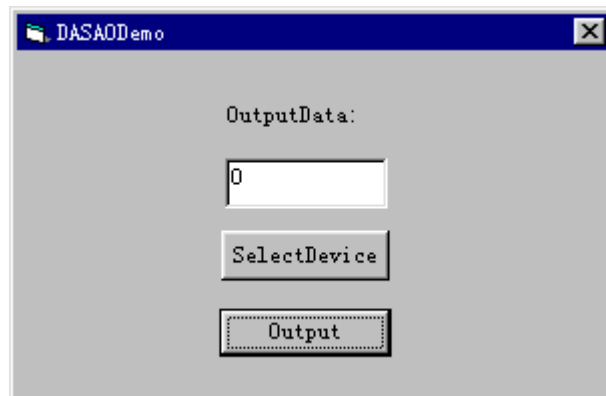


Here is another interface of this sample:



### 6.10 The sample of DASAO Demo

The analog can be output via DASAO control in this sample and at least one supported card should be inserted. The detail can be referred to Chapter 7. Users have to get some basic knowledge on the hardware setting of the card and configure the card resources via DevInst.exe, including the port, interruption and DMA. You can find the detailed configuration method in Chapter 8. Here is another interface of the sample:

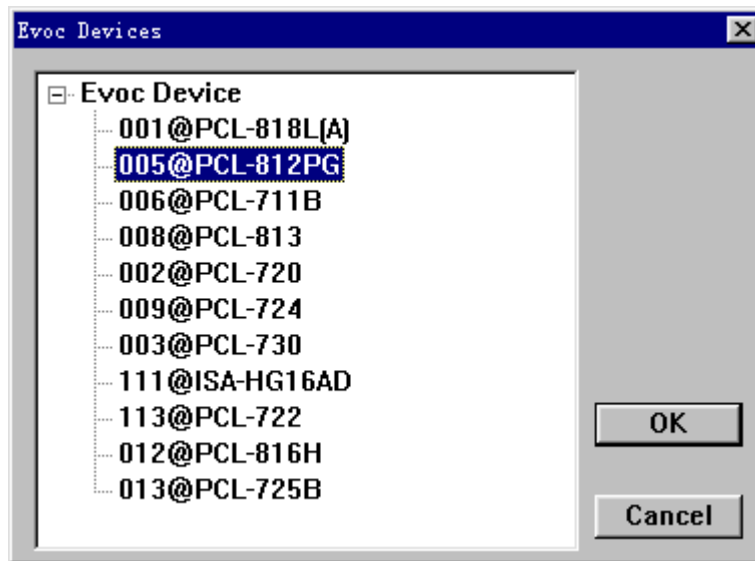


### 6.11 The sample of DASDi Demo

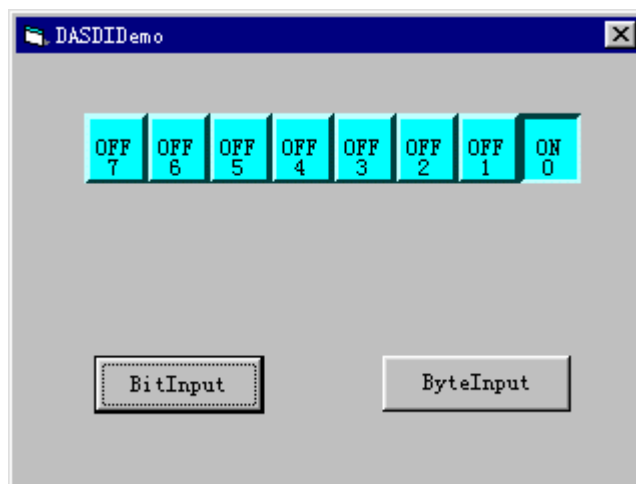
The output of shaped switch quantity is carried out by DASD I control and at least one supported card should be inserted. The detail can be referred to Chapter 7. Users have to get some basic knowledge on the hardware setting of the card and configure the card resources via DevInst.exe, including the port, interruption and DMA. You can find the detailed configuration method in Chapter 8. Here is another

interface of the sample:

First, choose a supported card in the following way:



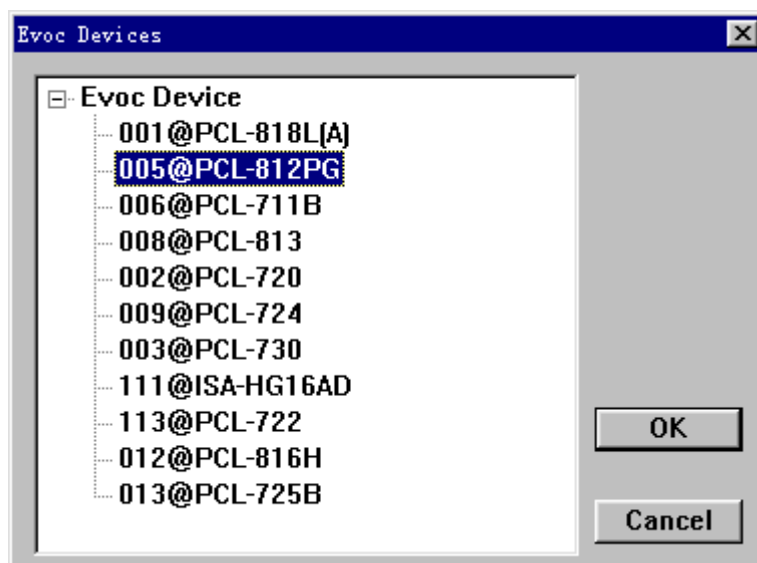
Here is an interface of this sample:



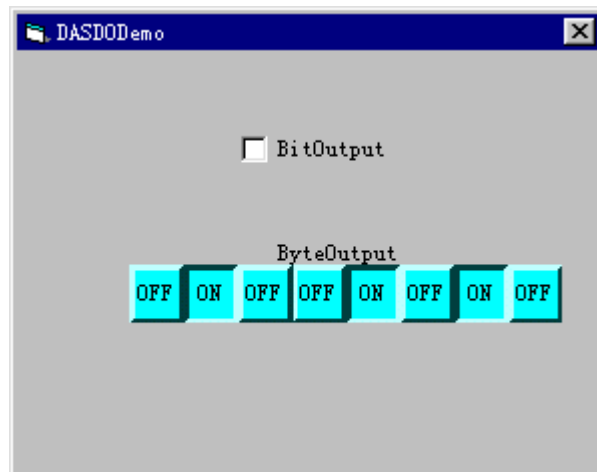
## 6.12 The sample of DASDoDemo

The output of shaped switch quantity is carried out by DASDO control and at least one supported card should be inserted. The detail can be referred to Chapter 7. Users have to get some basic knowledge on the hardware setting of the card and configure the card resources via DevInst.exe, including the port, interruption and DMA. You can find the detailed configuration method in Chapter 8. Here is an interface of the sample:

First, choose a supported card in the following way:



Here is an interface of the sample:



### 6.13 Other points for attention

Login MFC before using the control, e.g. regsvr32

C:\windows\system\mfc42.dll

For most controls, it's better to set their properties through the property page and don't program directly. But the property values of Knob, Slide, E7Segment and Boolean can be set directly by programming. Don't assign value to the value properties of Knob and Slide via program, and just assign values to the corresponding pointer. The control method can be used via programming.

In addition, "Enhanced color" is suggested for the resolution (over 16 bits).

## **Chapter 7 Devices supported by hardware operation and their installation**

### 7.1 Types of cards supported by various controls

1. DASAI: PCL-711B, PCL-818L, PCL-813, PCL-812PG, PCL-816H, ISA-HG16AD
2. DASAO: PCL-711B, PCL-818L, PCL-812PG, PCL-726, PCL-728, ISA-HG16AD
3. DASDI: PCL-720, PCL-722, PCL-724, PCL-725, PCL-725/16, PCL-730, PCL-711B, PCL-818L, PCL-812PG, PCL-816H, PCL-726, ISA-HG16AD
4. DASDO: PCL-720, PCL-722, PCL-724, PCL-725, PCL-725/16, PCL-730, PCL-711B, PCL-818L, PCL-812PG, PCL-816H, PCL-726, ISA-HG16AD
5. DASCounter : PCL-720 , PCL-730 , PCL-818L , PCL812PG, PCL-816H
6. DASPulse: PCL-720, PCL730, PCL-818L, PCL-812PG, PCL-816H
7. PCL731A: PCL-731(A)
8. PCL733: PCL-733
9. PCL734: PCL-734
10. PCL836A: PCL-836(A)

---

## 7.2 Installation

Install the corresponding driver of your ISA card first and the control packet ActiveDAS next.

Win98 version:

First, please install the driver of the ISA card for 95&98, and then install the control packet of ActiveDAS for Win98. Before using the control, you have to distribute the card resources via the DevInst.exe, including port, interruption and DMA. You can find the detailed configuration in Chapter 8. Before using the control, you have to login MFC, e.g. regsvr32

C:\windows\system\mfc42.dll

1. The cards support by DLL1 for Win95&98: PCL-720, PCL-722, PCL-724, PCL-725, PCL-725/16, PCL-730.
2. The cards supported by DLL2 for Win95&98: PCL-711B, PCL-813, PCL-816H, PCL-818L, PCL-818HG, PCL-812PG, PCL-726, PCL-728.
3. The cards supported by DLL8 for Win95&98: PCL-816H
4. The cards supported by 73x for Win98: PCL-731(A), PCL-733, PCL-734,



5. The card supported by 836A for Win98&WinNT:  
PCL-836(A)

6. The card supported by ISA-HG16AD for 98:  
ISA-HG16AD

WinNT:

Please install the driver of ISA card for NT, and then install the control packet of ActiveDAS for WinNT. Before using the card control, you have to distribute the card resources via the DevInst.exe, including port, interruption and DMA. You can find the detailed configuration in Chapter 8. Before using the control, you have to login MFC, e.g. regsvr32

C:\windows\system\mfc42.dll。

1. The cards supported by DLL1 for WinNT: PCL-720, PCL-722, PCL-724, CL-725, PCL-725/16, PCL-730.
2. The cards supported by DLL2 for WinNT: PCL-711B, PCL-813, PCL-816H, PCL-818L, PCL-818HG, PCL-812PG, PCL-726, PCL-728.

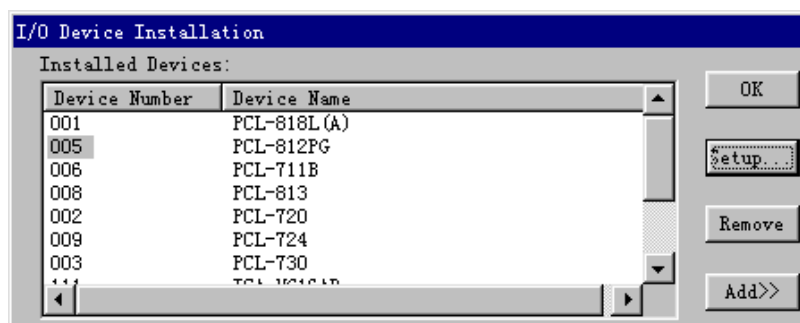
3. The card supported by DLL8 for WinNT: PCL-816H

4. The card supported by 836A for Win98&WinNT:  
PCL-836(A)

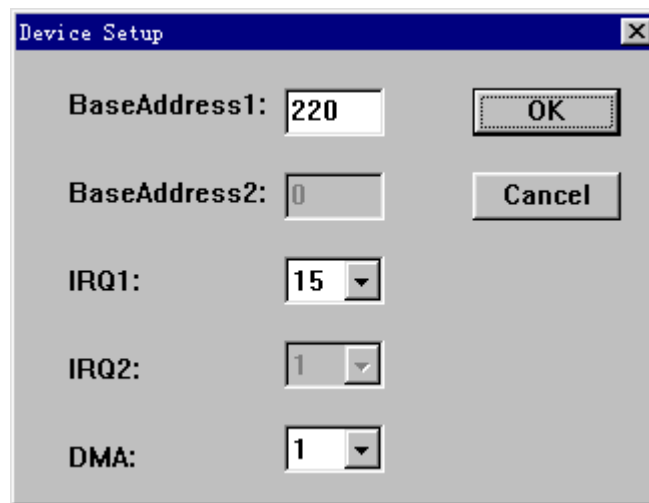
## Chapter 8 Usage of DevInst.EXE

This configuration tool is especially used for distributing the hardware resources that need to be driven by the controls of hardware operation. You have to register the resources (including port, interruption and DMA) for the ISA card that doesn't have the function of plug and play, thus corresponding hardware operation can be realized. The jumper should be in coherence with the configured information, i.e. the configuration of software should be in coherence with that of hardware, or the hardware will fail to be driven correctly and the system may be damaged

You can find the detailed information on the controls of hardware operation and supported cards in Chapter 7. After the configuration, the information will be always in effect as long as no change is made to the hardware of the computer. Once the number of cards is increased or decreased, or the jumper is changed, a re-configuration should be conducted. The already configured hardware can be found through the software. The following picture is the newly starting up of the program and the hardware shown in the list box is the card of the computer.



Choose the device number of the card with the mouse and click “Setup” on the right side. You will see two base addresses of the port, two interruptions and two DMA, and the non-necessary resources will be shown in grey as un-editable status. The edition is subject to hex.



The list box shows to be blank if this configuration tool

is used for the first time and you may click “Add” to add new devices, but don’t repeat the DeviceNumber. The device will be added into the above list box after “Add it” is clicked. Click “Setup” to configure it.

