Software Introduction and Testing
Agenda

- Introduction
- 32-bit DLL Driver
- Software Support List
- ActiveDAQ Pro Driver
- LabVIEW DAQ Driver
- Conclusion
Introduction
Introduction

- To have a complete DA&C system, we have to learn how to programming DA&C modules

- Advantech plug-in DA&C products is consumed by engineers who use any program development tool
  → Software support could be really difficult

- The easier to use, the more successful drivers are
32-bit DLL Driver
Advantech’s 32-bit DLL Driver

- Supported operation systems: Win2000, WinXP
- Is a MUST for DA&C cards
- Same programs can be easily adopted by different cards with the same functions
  → Hardware independent
- Immunity against version changes of hardware
- Is the base of most 3rd-party drivers
Driver Hierarchy

- API is controlling DA&C cards through ADSAPI32 drivers
Comprehensive software support is based on DLL drivers
DOS Support

- Can be found in DA&C companion CD folder “CD\DOS”
- Basically, all control of the cards are accessing the register
- Register information can be found in the bottom of HW manual
- One can use register programming for the followings:
  - some advanced function
  - real time application, like RTX
- DOS support could be
  1. Register (direct I/O) Programming
  2. DOS driver
DOS Support Example – PCL-818HG

- **Register Programming Solution**
  1. Find DEMO example in folder:
     DA&C 2.5D\DOS\PCL\PCL818HG.150\DIRECT.IO\C
  2. Check “Appendix C. Register Structure & Format” of PCL-818HG’s HW manual

<table>
<thead>
<tr>
<th>Base Addr.</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>R</strong></td>
<td>R</td>
<td>A/D low byte data and channels</td>
<td>AD3</td>
<td>AD2</td>
<td>AD1</td>
<td>AD0</td>
<td>C3</td>
<td>C2</td>
</tr>
<tr>
<td><strong>W</strong></td>
<td></td>
<td>Software A/D trigger</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>R</strong></td>
<td>R</td>
<td>A/D high byte data and channels</td>
<td>AD11</td>
<td>AD10</td>
<td>AD9</td>
<td>AD8</td>
<td>AD7</td>
<td>AD6</td>
</tr>
<tr>
<td><strong>W</strong></td>
<td></td>
<td>A/D range control</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MUX scan channel status</td>
<td>G3</td>
<td>G2</td>
<td>G1</td>
<td>G0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3. Compile the program and make execution file
DOS Support Example – PCL-818HG

- DOS Driver Solution

1. Find examples in folder:
   DA&C 2.5D\DOS\PCL\PCL818HG.150\C

2. Compile the program and make execution file

```c
/*
 * Program : ADTRIG.C
 * Description : Demo program for PCL-818HG pacemaker trigger A/D conversion with software data transfer.
 * Revision : 1.00
 * Date : 12/27/1993
 * Advantech Co., Ltd.
 */

#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <dos.h>

extern pcl818HG(int, unsigned int *)

unsigned int param[10];
unsigned int data[100];
unsigned int far * dat;

main()
{
    unsigned int i;
    float DataBuf;

    clrscr();

    dat = data;

    param[0] = 0; /* Board number */
    param[1] = 0x300; /* Base I/O address */
    param[5] = 50; /* Pacer rate = 3M / (50 * 100) = 300 Hz */

    Use driver's functions

    // Code to use the driver's functions
}

Include DOS driver
```
Software Support List
Supported Software for DA&C Cards

- Advantech Device Manager
  - Utility for installation and testing of DA&C cards
  - Provided functional tests: AI, AO, DI, DO, Counter
Supported Software for DA&C Cards

- Advantech GeniDAQ
  - SCADS software for DA&C cards and ADAM modules
  - Fit in low-speed data monitoring and control applic.
Supported Software for DA&C Cards

- **Advantech WaveScan 2.0**
  - Support PCI cards & USB modules
  - Real-time monitoring
  - Data logging
Supported Software for DA&C Cards

- 32-bit DLL Driver Examples
  - Include VB, VC, BCB, Delphi examples

- ActiveDAQ & ActiveDAQ Pro Examples
  - ActiveDAQ Pro supports PCI and USB modules
  - All supports VB.NET and C#.NET

- Third-party Drivers & Examples
  - LabVIEW (LabVIEW & LabVIEW DAQ)
  - MATLAB
  - DASYLAB

- DOS Examples
- Linux Examples
Additional Support Lists

- **DAQ Support List**
  - Details support list for ActiveDAQ Pro, LabVIEW, LabVIEW DAQ, Linux and WinCE drivers

- **MATLAB Support List**

- **DASYLAB Support List**
  - [http://www.dasylab.com/content/driver.php?action=filelist&sid=24](http://www.dasylab.com/content/driver.php?action=filelist&sid=24)
Hands-on Practice – WaveScan 2.0

- Based on ActiveDAQ Pro control
- Integrate DI function with AI (for low speed AI only)
- Extend Hardware support to PCI cards
- Real-time monitoring
- History checking
Hands-on Practice – WaveScan 2.0

1. Open WaveScan in the folder:
   - C:\Program Files\Advantech\ActiveDAQ Pro\WaveScan
   - Start menu → Programs → Advantech Automation
     → WaveScan 2.0 → WaveScan 2.0
Hands-on Practice – WaveScan 2.0

2. Set device parameters
   - Please add a Demo module and select it in WaveScan 2.0
Hands-on Practice – WaveScan 2.0

- 3. Select DI or DO and Save2Disk and Run
Hands-on Practice – WaveScan 2.0

- 4. Click “Save As” and create a folder for binary data
Hands-on Practice – WaveScan 2.0

- 5. Click “History” and select saved project
Hands-on Practice – WaveScan 2.0

6. Click “Play” to start history checking
Hands-on Practice – WaveScan 2.0

- 7. Click “Convert” and create a folder for analog data
Hands-on Practice – WaveScan 2.0

8. Open the converted .xlt file
   - in C:\Program Files\Advantech\WaveScan 2.0\Converted Data
Hands-on Practice – WaveScan 2.0

- 9. Create a chart to plot the data
ActiveDAQ Pro Control
Introduction

- A newly developed driver for USB and PCI modules
- Revision of ActiveDAQ
  → ActiveDAQ doesn’t support high speed function of PCI-1714, PCI-1712, PCI-1721 and PCI-1755
- OCX(ActiveX) structure
- Efficient programming
  - 20% of parameter setting
  - 80% of programming
- Supports more interfaces
  - VB, VC, BCB, Delphi, VB.NET, C#.Net
Features of ActiveDAQ Pro

- Straightforward user interface
- **Property page** offers access to all settings
- Default settings for immediate execution
- Can be adopted by software that supports OCX component
- Satisfy all high speed functions
  - Modification of ActiveDAQ
- GUI component support
Structure of ActiveDAQ Pro

- Based on 32-bit DLL driver
- Object-oriented
- Consists of Property, Method and Event

Active DAQ PRO

Win32-DLL

Properties
- DeviceName
- DeviceNumber
- Resolution

Method
- SelectDevice
- ReadDiChannel
- ReadMeasurementValue

Event
- OnDiInterrupt
- OnTerminate
- OnTimeOut
ActiveDAQ Pro Controls

- **Device Control**
  - AdvDevice: Set & get device information

- **Analog Control**
  - AdvAI: High speed and low speed AI
  - AdvAO: High speed and low speed AO

- **Digital Control**
  - AdvDIO: Digital input and output

- **Thermo Control**
  - AdvThermo: Thermocouple measurement

- **Counter Control**
  - AdvCounter: Read counter input

- **Pulse Control**
  - AdvPulse: Generate pulse output
Comprehensive Examples

- **Examples are available in**
  
  C:\Program Files\Advantech\ActiveDAQ Pro\Examples
Hands-on Practice – Using AdvAI Control

- **Demand**
  - Implementation of AI control in a C#.NET project

- **Material**
  - USB-4716 (or any other AI module)
  - DevMgr, 32-bit DLL driver, ActiveDAQ Pro driver
  - Visual Studio .Net environment

- **Goal**
Hands-on Practice – Using AdvAI Control

- Step 1. New a C# Windows appl. project
Hands-on Practice – Using AdvAI Control

- Step 2. Add ActiveDAQ Pro AdvAI Control
Hands-on Practice – Using AdvAI Control

- Step 2. Add ActiveDAQ Pro AdvAI Control (Cnt’d)
Hands-on Practice – Using AdvAI Control

- Step 3. Check control’s property
Hands-on Practice – Using AdvAI Control

- Step 4. Design the form
Hands-on Practice – Using AdvAI Control

- **Step 5. Start programming**
  5.1- Create a button to select device (optional)
    Create a textbox(textBox1) to show device name
    ```c#
    Demo.SelectDevice();
    textBox1.Text=Demo.DeviceName.ToString();
    ```
  5.2- Create a button to read the AI
    Create a textbox(textBox2) to show AI reading
    ```c#
    Demo.ChannelNow=0;
    textBox2.Text=Demo.DataAnalog.ToString();
    ```

- **Step 6. Test the program**
Hands-on Practice – Using AdvAI Control

- Comparison with DLL driver programming
  - Steps of operation are the same
  - The program is more complicated

```vbnet
Private Sub Command1_Click()
    Dim DeviceNum As Long
    Dim DeviceName As String * 256
    Dim Response As Long
    ErrCde = DRV_SelectDevice(Me.hWnd, False, DeviceNum, DeviceName)
    ErrCde = DRV_DeviceOpen(DeviceNum, DeviceHandle)
    Text1.Text = DeviceName
End Sub

Private Sub Command2_Click()
    Dim lpAIConfig As PT_AIConfig
    lpAIConfig.DasChan = 0
    ErrCde = DRV_AIConfig(DeviceHandle, lpAIConfig)

    Dim lpAIVoltageIn As PT_AIVoltageIn
    Dim AIreading As Single
    lpAIVoltageIn.chan = 0
    lpAIVoltageIn.voltage = DRV_GetAddress(AIreading)
    ErrCde = DRV_AIVoltageIn(DeviceHandle, lpAIVoltageIn)
    Text2.Text = Format(AIreading, "###0.000000")
End Sub
```
Hands-on Practice – AdvAI & AdvAO

- **Goal:** Connect AI to AO for self-testing
  - AO: Output AO as will
  - AI: Read high speed AI value coming from AO ch.
    → Can check the functionality of both AI & AO

- Different controls can coexist at the same time

- Please refer to ActiveDAQ Pro software manual in C:Program Files\Advantech\ActiveDAQ Pro

- New error-handling mechanism
Hands-on Practice – AdvAI & AdvAO

- 1. Connect AO to AI
- 2. Open VB and create a new project
- 3. Design a form for your application
   (the one below is for your reference)
Hands-on Practice – AdvAI & AdvAO

4. Open ActiveDAQ Pro manual and check AdvAO/Properties/DataAnalog

Open eAutomation, Boundless Integration
Hands-on Practice – AdvAI & AdvAO

- 5. Double-click the output button and paste the example code from the manual

```vbscript
Private Sub Command2_Click()

' Set the device number of AdvAO1 to 0
AdvAO1.DeviceNumber = 0

' Set the ChannelNow of AdvAO1 to 0
AdvAO1.ChannelNow = 0

' Set the ChannelNow as a voltage output
AdvAO1.DataPhysics = 0

' Set the value range
AdvAO1.SetValueRange AdvAO1.ChannelNow, 0, 5

' Returns the value range string.
labDataValueRange.Caption = AdvAI1.DataValueRange

' Output a value from DataAnalog
AdvAO1.DataAnalog = 1.65

End Sub
```
Hands-on Practice – AdvAI & AdvAO

6. Modify the posted code as shown below:

```vbnet
Private Sub Command2_Click()
    On Error GoTo ErrorHandler
    'Sets the device number of AdvAO1 to 0
    AdvAO1.DeviceNumber = 0

    'Set the ChannelNow of AdvAO1 to 0
    AdvAO1.ChannelNow = 0

    'Set the ChannelNow as a voltage output
    AdvAO1.DataPhysics = 0

    'Set the value range
    AdvAO1.SetValuerange AdvAO1.ChannelNow, 0, 5

    'Returns the value range string.
    labDataValueRange.Caption = AdvAO1.DataValueRange

    'Output a value from DataAnalog
    AdvAO1.DataAnalog = Val(txt1.Text)
    The AO value is given in Text1

Exit Sub
ErrorHandler:
    MsgBox AdvAO1.LastErrorMessage
End Sub
```
Hands-on Practice – AdvAI & AdvAO

7. Open ActiveDAQ Pro manual and check AdvAI/Properties/AcquireBulkDataToMemory
8. Double-click the AI Scan button and paste the example code from manual

Private Sub Command1_Click()

'Sets the device number of AdvAI1 to 0
AdvAI1.DeviceNumber = 0
AdvAI1.ChannelScanStart = 0
AdvAI1.ChannelScanCount = 4

'Sets DataSampleRate
AdvAI1.DataSampleRate = 40000

'Starts to acquire bulk data
Dim DigitalValue As Variant
AdvAI1.AcquireBulkDataToMemory 20000, DigitalValue, 0, False, True

'Converts the Digital Value to Analog Value
AdvAI1.ConvertDigitalValueToAnalog DigitalValue, AnalogValue, 20000

'Displays the data
Dim i As Long
frmDisplay.lstRawdata.Clear
frmDisplay.lstVoltage.Clear
For i = 0 To 5
    frmDisplay.lstRawdata.AddItem (Hex(digitalArray(i)) & " E")
    frmDisplay.lstVoltage.AddItem (Format(analogArray(i), "0.000000"))
Next i
End Sub
Hands-on Practice – AdvAI & AdvAO

9. Modify the posted code as shown below:

```vbnet
Private Sub Command1_Click()

On Error GoTo ErrorHandler

' Sets the device number of AdvAI to 0
AdvAI.DeviceNumber = 0
AdvAI.ChannelScanStart = 0
AdvAI.ChannelScanCount = 1

' Only scan channel 0

' Sets DataSampleRate
AdvAI.DataSampleRate = 40000

' Starts to acquire bulk data
Dim DigitalValue As Variant
Dim AnalogValue As Variant
AdvAI.AcquireBulkDataToMemory 20000, DigitalValue, -1, False, True
AdvAI.ConvertDigitalValueToAnalog DigitalValue, AnalogValue, 20000

' Displays the data
For i As Long = 1 To 5
List1.Clear
List2.Clear

For i = 0 To 5
List1.AddItem Hex(DigitalValue(i)) & “H”
List2.AddItem Format(AnalogValue(i), “0.000000”) Next i

Exit Sub
ErrHandler:
MsgBox AdvAI.LastErrorMessage
AdvAI.StopAcquireBulkData (0)

End Sub
```
Hands-on Practice – AdvAI & AdvAO

- 10. Compile and run the program
GUI Control ActiveDAQ Pro
Introduction

- Graphic User Interface
- ActiveX component
- Will be released in September 2007
- Must pay to have Serial Number for installation
- Consists of two parts:
  - ActiveDAQ Pro Device Control
  - ActiveDAQ Pro GUI Control
Introduction

- Totally 7 different controls

Advantech ActiveDAQ Pro AdvButton Control 1.0
Advantech ActiveDAQ Pro AdvGraph Control
Advantech ActiveDAQ Pro AdvIntensity Control
Advantech ActiveDAQ Pro AdvKnob Control
Advantech ActiveDAQ Pro AdvLED Control
Advantech ActiveDAQ Pro AdvNumEdit Control
Advantech ActiveDAQ Pro AdvSlider Control
Installation

- Contents in Companion CD

ActiveDAQ Pro is a set of OCX components for data acquisition & control and data representation. The data acquisition & control components support a wide variety of Advantech's data acquisition & control hardware just like the previous version of ActiveDAQ 1.6x; what's more, they are improved to a totally new generation to satisfy all of the high speed, user-friendly acquisition. The data representation components, such as Graph and Button, are a set of bran-new OCX for data representation.
How to add component (in VS 2005)

→ Add control (Toolbox → Choose Item → COM Component)
Operation

- How to add component (in VS 2005)
  -> Add control (Toolbox -> Choose item -> COM component)
Graph Control

- **Includes**
  - Axes control
  - Cursor control
  - Annotation of data
  - Multi-Y axes
  - X vs. Y graph
Graph Control

- Property page for parameter setting
Hands-on Practice - AdvGraph

- **Goal: Add data to graph**
  - Among all the GUI controls, Graph control is the most powerful one
  
  - The first thing of utilizing graph control is to plot data to graph
    → AI, DI or some calculated results may be represented graphically (through GUI)
  
  - Annotation helps to highlight points in a graph
Hands-on Practice - AdvGraph

- Get Started – 1. Add control to the form
Hands-on Practice - AdvGraph

- Get Started – 2. Create a button to plot graph

```vba
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Click
    Dim ArrData() As Double = {1.0, 5.0, 3.0, 7.0}
    Graph.Axes.Item(0).AutoScale = True
    Graph.Plots.Item(0).PlotY(ArrData)
End Sub
```
Hands-on Practice - AdvGraph

- Get Started – 3. Create a button to add annotation

```vba
Dim AddNote As AdvGraphLib.AdvAnnotation

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Dim ArrData() As Double = {1.0, 5.0, 3.0, 7.0}
    Graph.Axes.Item(0).AutoScale = True
    Graph.Plots.Item(0).Plot(YArrData)
End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    AddNote = Graph.Annotations.Add()
    AddNote.ier_Plot(Graph.Plots.Item(0))
    AddNote.PointIndex = 2
    AddNote SnapNode = AdvGraphLib.AdvCursorSnapNode.advSnapanchoredToPoint
    AddNote.Caption = "The third one"
End Sub
```
Hands-on Practice – AdvGraph & AdvNumEdit

- **Goal: Enhance graph with cursor**
  - NumEdit control to replace traditional number control
  - Add cursor using the same way like adding annotation
  - Cursor provides capability of making scope application
Hands-on Practice – AdvGraph & AdvNumEdit

1. Add a button and a AdvNumEdit control
   - Button is for adding cursor
   - AdvNumEdit is for cursor control
Hands-on Practice – AdvGraph & AdvNumEdit

2. Program for adding cursor
   - the same way as adding annotation
   - the snap mode should be changed to “advSnapNearestYForFixedX”

3. Program for X-axis control
   - Double-click to enter the programming of NumEdit_value_change

```vbnet
Private Sub AxAdvNumEdit1_ValueChanged(ByVal sender As Object, ByVal e As System.EventArgs)
    AddCursor.XPosition = CInt(XNum.Value)
End Sub
```

4. Run and test
Demo of ActiveDAQ Pro

- With ActiveDAQ Pro and its GUI component, DAQ card can address in many more application
LabVIEW DAQ Driver
Introduction

- Short for Laboratory Virtual Instrument Engineering Workbench
- LabVIEW is a visually-oriented programming development tool
- Programs are called VI
- Every VI in LabVIEW has three main parts:
  1. The front panel
  2. The block (or wiring) diagram window
  3. Icon & connector
Advantech LabVIEW DAQ Driver

- The latest generation of LabVIEW driver
- Wrapper of Advantech 32-bit DLL driver for DA&C cards
- Working with LabVIEW version 7, 7.1, 8.0, 8.2
- Provide different interfaces:
  - Easy I/O VIs
  - Intermediate I/O VIs
  - Advanced I/O VIs
  - Utility I/O VIs
- LabVIEWDAQ VI can replace NI’s DAQ VI
Structure of LabVIEWDAQ Driver

- Through the structure, driver becomes NI Style
Driver Installation

- **Where to find the driver**
  - in CD folder: ..\LabVIEW\LabVIEW.exe
  - on DA&C module’s supporting web page

- **Steps of operation**
  1. Install Advantech device manager
  2. Install 32-bit DLL driver
  3. Install LabVIEW DAQ driver

- **Software Manual**
  Can be found, after installation, in
  C:\Program Files\National Instruments\LabVIEW X.X\help\Advantech
Where to find LabVIEW DAQ VI?

- Right-click on the “block diagram window” and select Functions → All Functions → User Libraries → Advatech DAQ
Hands-on Practice – Easy VI for polling AI

- **Goal:** To read software polling AI w/ easy AI VI
  - Only one VI is needed
1. Draw VIs out from the block diagram window
   - By just one VI, normal speed AI can be read

   "Build array" is used to separate multi-ch. data
Hands-on Practice – Easy VI for polling AI

- 2. Create a “Wait Clock”

- 3. Wire and run the program
Hands-on Practice – Replace NI DAQ VI

- **Goal:** Use existing LV project based on NI DAQ VI
  - The pin assign of Advantech LabVIEW DAQ VI is identical to NI DAQ VI
  - By replacing the VI, existing programs based on NI DAQ driver can be used to control Advantech DAQ
  - A demo video is available on the web
  - All examples NI DAQ driver provides can be replaced in the same manner
Hands-on Practice – Replace NI DAQ VI

1. Find NI DAQ “Cont Acq&Chart (buffered)” example in C:\Program Files\National Instruments\LabVIEW 8.0\examples\daq\anlogin
Hands-on Practice – Replace NI DAQ VI

- 2. Run & Error (Doesn’t support the HW)
Hands-on Practice – Replace NI DAQ VI

3. Right click the VI and replace it by LabVIEWDAQ VI
- Watch out the Device Number connection of AI_Start
Hands-on Practice – Replace NI DAQ VI

- 4. Finish the replacement of all VI
Hands-on Practice – Replace NI DAQ VI

- 5. Run again to test
Demo Operation - FFT Analysis of AI

- Goal: Do FFT Analysis on signal sampled by PCI-1714
  - We can make use of LV’s VI to do lots of things
Demo Operation - FFT Analysis of AI

- We can use any level of VI to complete the DAQ part
- **All functions** that are available in LabVIEW can be used with LabVIEW DAQ drivers
- The FFT part is done by “Spectral Measurement VI”

![Spectral Measurements Diagram](attachment:image.png)

Performs FFT-based spectral measurements, such as the averaged magnitude spectrum, power spectrum, and phase spectrum on a signal.
Conclusion
Conclusion

- 32-bit DLL driver maintains all software support
- ActiveDAQ Pro provides a better interface of controlling Advantech DA&C cards
- Development tools that support OCX can also support ActiveDAQ Pro
- GUI control improves ActiveDAQ Pro
- WaveScan 2.0 is a good tool for data monitoring and data logging applications with any programming
- Existing LV programs can be saved with LabVIEW DAQ