User Manual

APAX-557X

Software Manual

ADVANTECH

eAutomation

Copyright

Acknowledgements

Notes on the Manual

This is the Software Manual for the Advantech APAX-557X product. This manual will help guide the end user through implementation and use of the software portion of this product.

What is covered in this manual:

This manual will give a general overview of the Windows XP Embedded operating system, most of the applications that are included with Windows XP Embedded as well as the applications added and/or created by Advantech Corporation in the Windows XP Embedded image. This manual will also cover installation and use of development and utility software that is needed. It will also reference optional software that can be used by the end user with the Windows XP Embedded Operating system.

What is not covered in this manual:

This manual will reference the hardware but does not contain hardware setup information, wiring information, electrical specifications or any detailed hardware information. Please refer to the hardware manual for this information.

Edition 1

September 2009

Contents

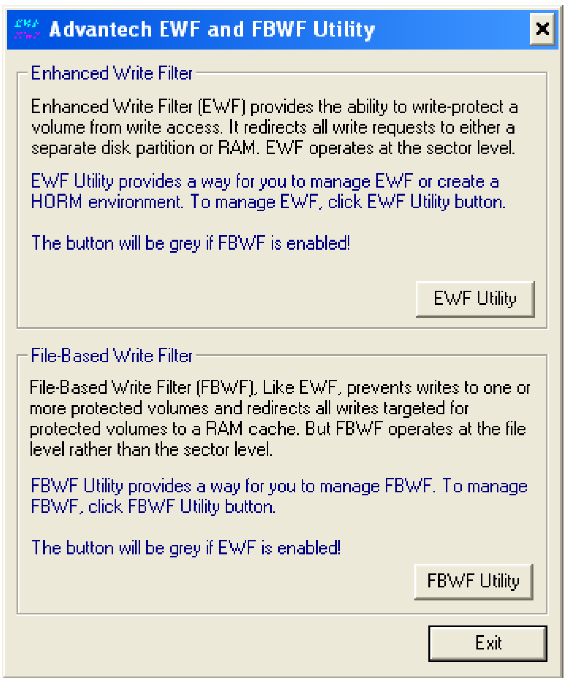
Chapter 1

Windows XP Embedded

1.1 Windows XP Embedded

The APAX-557X comes with Windows XP Embedded as the operating system. XP Embedded is the componentized version of XP Professional. When XP Professional was developed, almost every driver and application that makes up Windows XP Professional is in component form and can be built into an image. This allows a developer to build a custom version of XP with only the needed components to make it leaner. The XP Embedded target designer contains over 10,000 components. For those not familiar with XP Embedded, it is the same as XP Professional with only a couple of major differences.

1. **Enhanced Write Filter** - There are two types of enhanced write filters and Advantech offers both types with a utility EWF (enhanced write filter) and FBWF (file based write filter).
   - EWF protects the entire volume from writes. For example, if a change is made such as copying a file to the C drive, it will appear to be there. But actually the file is written to a layer in RAM. Once the power is cycled on the unit, the file will be gone because the RAM will not have the layer anymore. This protects the whole image from corruption. If something happens to corrupt the OS, the unit just needs to be reset and the original image will boot again. This can be a problem if the end user wants to save data to the CF card.
   - FBWF is similar to EWF where it uses a RAM overlay to store current changes. The difference is that FBWF does not protect the whole volume, it only protects based on directory. So it is possible to protect the Windows directory while writing data to a different directory on the CF card. This is important for a user that wants to do data collection or has an application that needs to save changes.
   - Obviously only one type of Write filter can be used at a time.



Enhanced Write Filter (EWF) provides the ability to write-protect a volume from write access. It redirects all write requests to either a separate disk partition or RAM. EWF operates at the sector level.

EWF Utility provides a way for you to manage EWF or create a HORM environment. To manage EWF, click EWF Utility button.

The button will be grey if FBWF is enabled!

File-Based Write Filter (FBWF), Like EWF, prevents writes to one or more protected volumes and redirects all writes targeted for protected volumes to a RAM cache. But FBWF operates at the file level rather than the sector level.

FBWF Utility provides a way for you to manage FBWF. To manage FBWF, click FBWF Utility button.

The button will be grey if EWF is enabled!

2. **Microsoft Updates -** XP Embedded does not support Microsoft updates the way that XP Professional does. The end user cannot connect to the Microsoft updates page and get the latest patches from Microsoft. It does support Windows Server Update Services WSUS. This allows updates to be pushed out from a central server the same as network administrators would do in a supported environment.

3. **Pagefile System -** The APAX-557X comes with the operating system on a Compact Flash card. Compact flash cards have limited write capabilites to the cards. To lengthen the life of the compact flash, it has built in firmware that does "wear leveling" to not over use any one sector. Also to lengthen the life of the card, the XP Embedded Pagefile system (virtual memory) is disabled. This CAN be enabled, but it is not recommended for systems using solid state drives due to limited lifetime writes of SSD's.

1.1.1 Administrator Password

The XP Embedded OS comes with automatic logon already enabled. For automatic logon to work, the login name must also have a password. Below is the default login name and password. This can be changed via the registry or the TweekUI tool offered by Microsoft.

- **Login:** Administrator
- **Password:** password

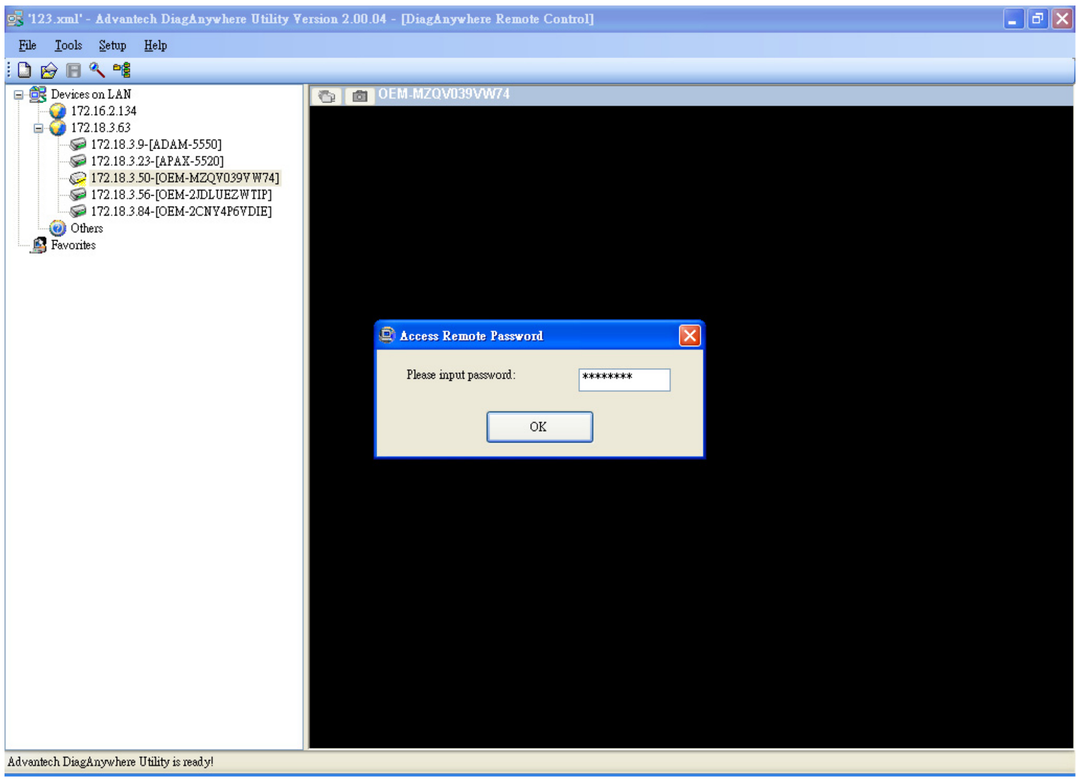
1.1.2 Remote Administration

DiagAnywhere

"DiagAnywhere" tool, an abbreviation of "Diagnostic Anywhere", is a networking solution for remotely monitoring and controlling other Windows based devices. It is very similar to a remote desktop application with some additional features. Currently, DiagAnywhere includes the utility on client side, and the server on the other. The main technology is based on Microsoft .NET Framework for the client. For this reason, the PCs using this solution must have the Microsoft .NET Framework installed for Win32 platform. You can find the .NET Framework and DiagAnywhere client trial version on the CD APAX-557X provides. DiagAnywhere has an advantage over the Windows Remote Desktop described below as it allows remote control while the local user can still see the desktop. Withi Windows Remote Desktop, it locks out the local user.

APAX-557X offers DiagAnywhere client trial version in the CD. Select the **Other Software** button and click the **DiagAnywhere** button to install the trial version. Then, you can use it to connect to DiagAnywhere server on APAX-557X. After you complete the installation, you can launch DiagAnywhere client by selecting Start >> All Programs >> Advantech Automation >> DiagAnywhere >> Advantech DiagAnywhere Utility. At the left side, expand the **Device on LAN** item by double click it to see all network on your PC. Select the IP which is used to connect with APAX-557X module and then click the search the **LAN** button. All devices connected to that local network will appear. Double click the item represent the APAX-557X to connect to it, a window will pop-up asking you to type password. After you enter the correct password, you can see the desktop on APAX-557X.

Note! *There is no default password.*



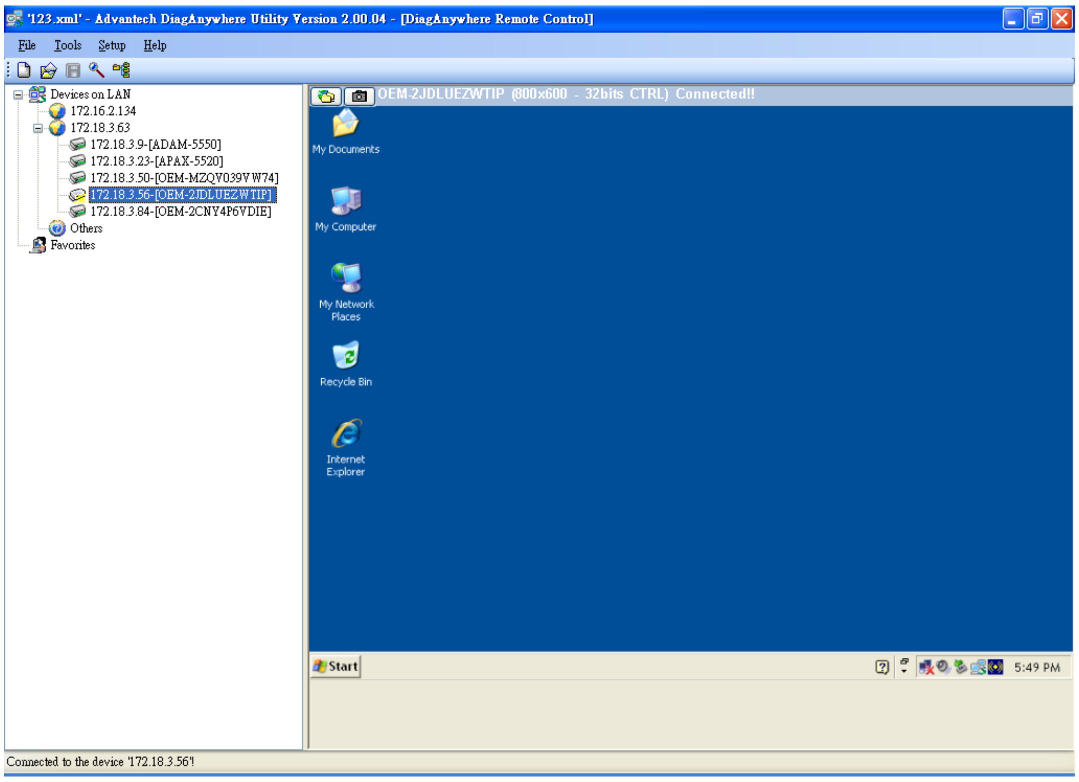
'123.xml' - Advantech DiagAnywhere Utility Version 2.00.04 - [DiagAnywhere Remote Control]

- Devices on LAN
  - 172.16.2.134
  - 172.18.3.63
    - 172.18.3.9-[ADAM-5550]
    - 172.18.3.23-[APAX-5520]
    - 172.18.3.50-[OEM-MZQV039VW74]
    - 172.18.3.56-[OEM-2JDLUEZWTIP]
    - 172.18.3.84-[OEM-2CNY4P6VDIE]
  - Others
- Favorites

Access Remote Password

Please input password: ********

Advantech DiagAnywhere Utility is ready!



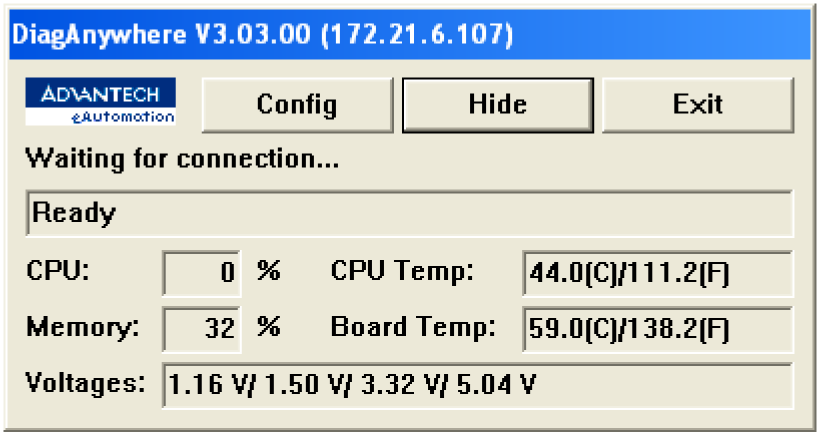
'123.xml' - Advantech DiagAnywhere Utility Version 2.00.04 - [DiagAnywhere Remote Control]

- Devices on LAN
  - 172.16.2.134
  - 172.18.3.63
    - 172.18.3.9-[ADAM-5550]
    - 172.18.3.23-[APAX-5520]
    - 172.18.3.50-[OEM-MZQV039VW74]
    - 172.18.3.56-[OEM-2JDLUEZWTIP]
    - 172.18.3.84-[OEM-2CNY4P6VDIE]
  - Others
- Favorites

OEM-2JDLUEZWTIP (800x600 - 32bits CTRL) Connected!!

Connected to the device '172.18.3.56'!

DiagAnywhere server can only run on Advantech's TPC, UNO, AMAX, APAX and ADAM Windows based devices. The supported platforms include Windows XP and Windows XPe.


DiagAnywhere V3.03.00 (172.21.6.107)

Waiting for connection...

Ready

CPU: 0 % CPU Temp: 44.0(C)/111.2(F)

Memory: 32 % Board Temp: 59.0(C)/138.2(F)

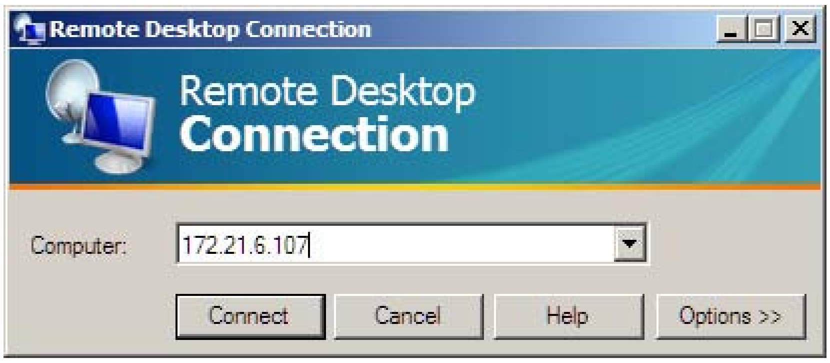
Voltages: 1.16 V/ 1.50 V/ 3.32 V/ 5.04 V

However, the server can accept only one connection from the utility at a time, and other connection attempts will be rejected if there is a live connection. APAX-557X has built-in DiaAnywhere server and the server will launch automatically after the system boots.

Remote Desktop

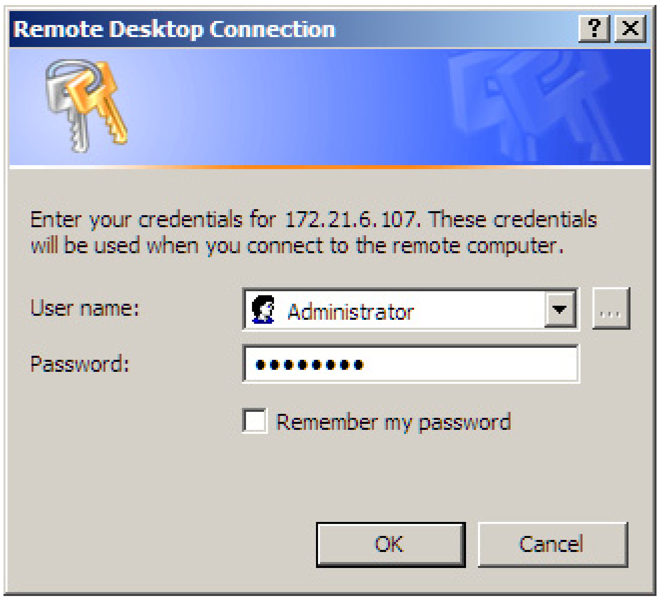
Windows XP embedded offers one remote desktop connection for maintenance purposes. This can be used as a remote management tool. The remote desktop should already be enabled in the image and another user only needs to start the remote desktop connection from another computer on the same network and login with an administrator login and password. When this login happens, it will lock out the local user on the APAX unit.

From the remote computer, use the Remote Desktop connection provided by Windows XP to log in to the APAX-557X.



Computer: 172.21.6.107

Log on using the Administrator account or another setup user account.


Remote Desktop Connection

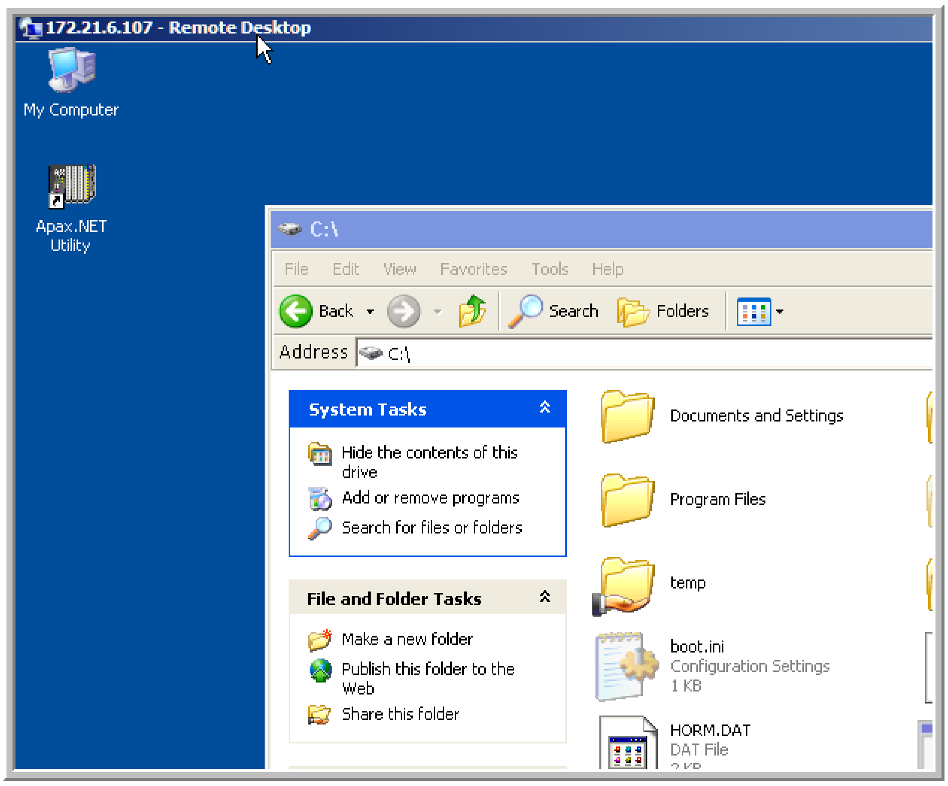
Enter your credentials for 172.21.6.107. These credentials will be used when you connect to the remote computer.

User name: Administrator

Password: ••••••••

Remember my password

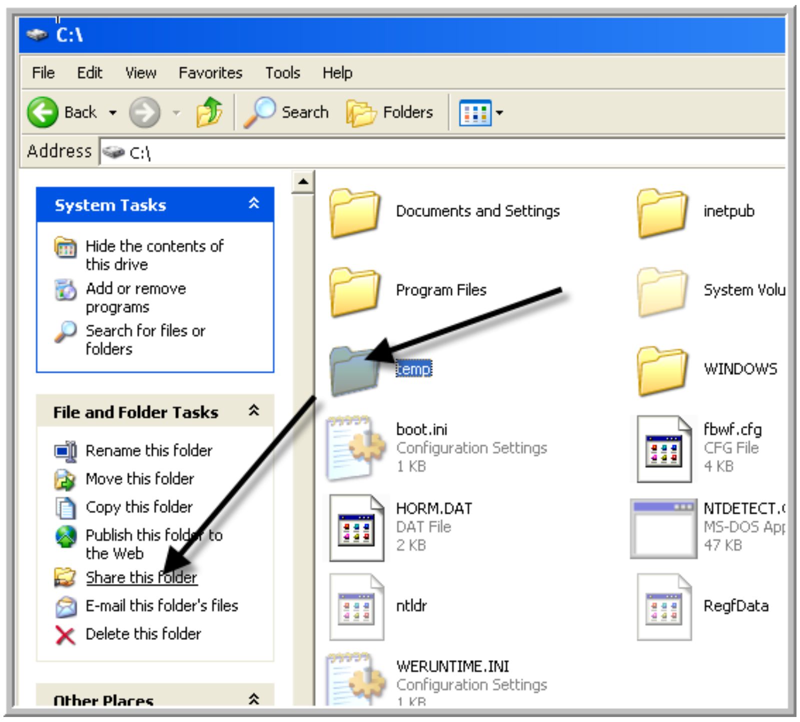
The remote desktop can then be controlled remotely but NOT locally. The local user will be logged out.


172.21.6.107 - Remote Desktop

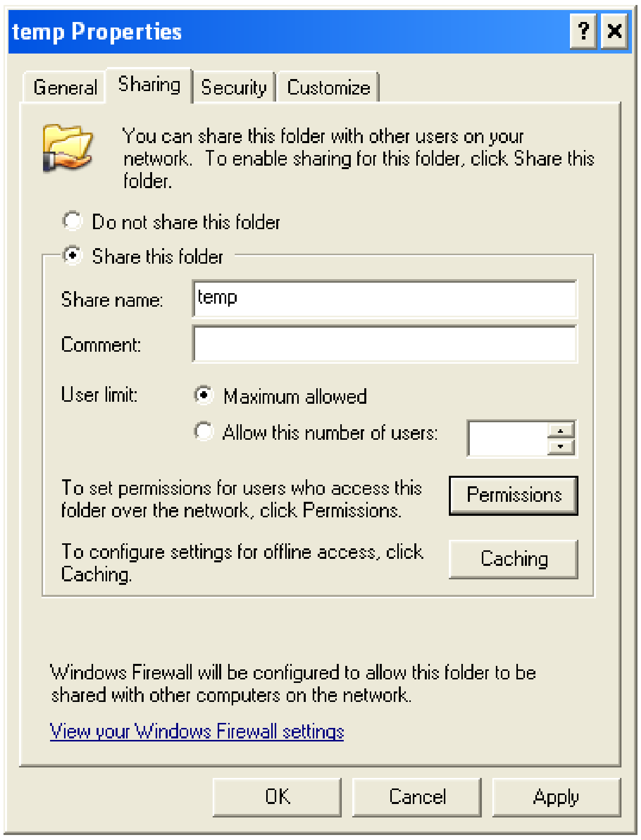
For further information on the Microsoft remote desktop see the Microsoft web site.

Remote File Sharing

To share files over a network, first enable a directory or multiple directories for sharing. In this example we will share the "temp" directory. First select the folder in Windows explorer and then click "Share this folder" from the left window pane.



From the "temp" properties select the share tab, click the share this folder radio button and then press the properties button.



You can share this folder with other users on your network. To enable sharing for this folder, click Share this folder.

Do not share this folder [✓ Share this folder]

Share name: temp

Comment:

User limit: [✓ Maximum allowed] Allow this number of users:

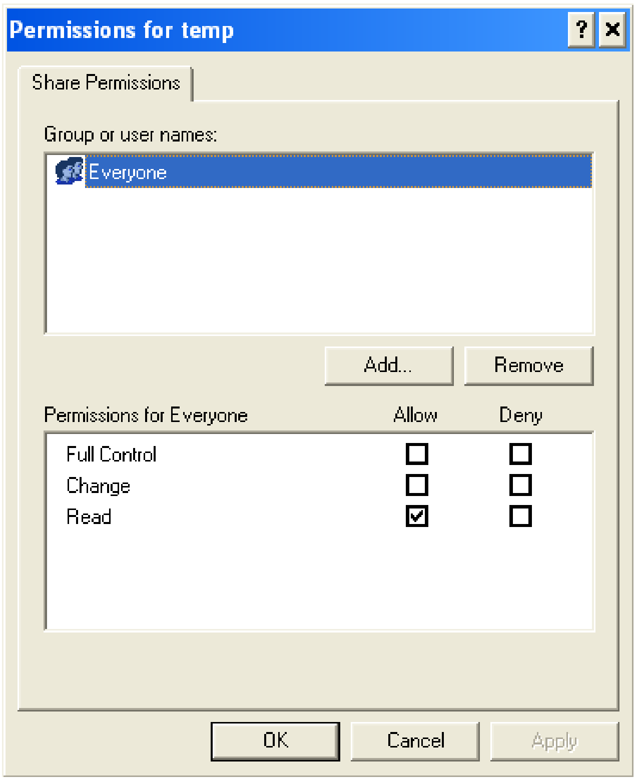
To set permissions for users who access this folder over the network, click Permissions.

To configure settings for offline access, click Caching.

Windows Firewall will be configured to allow this folder to be shared with other computers on the network.

View your Windows Firewall settings

From here select the access rights for sharing.



Permissions for temp

Share Permissions

Group or user names:

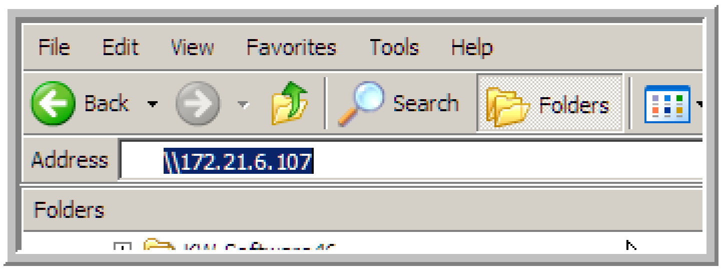
Everyone

Add... Remove

Permissions for Everyone	Allow	Deny
Full Control	☐	☐
Change	☐	☐
Read	☑	☐

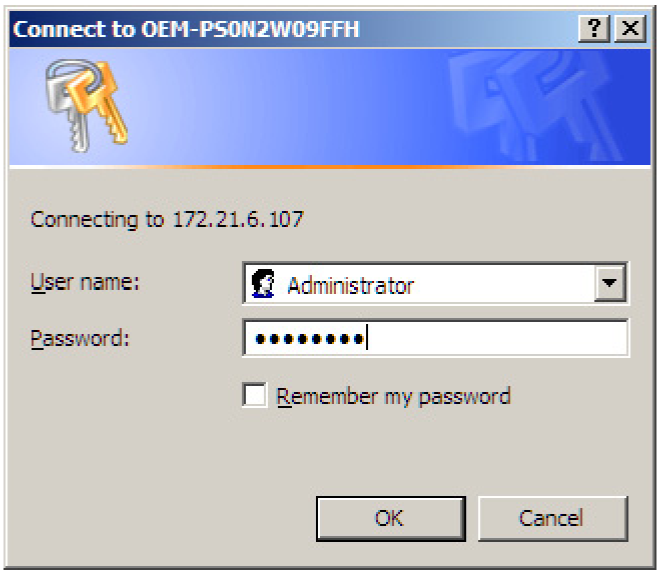
OK Cancel Apply

To access the folder from another computer, type in the IP address with the two slashes preceding.



\\172.21.6.107

Log in with the Administrator user name and password, or another user name and password that has been set up in the APAX-557X system.


Connect to OEM-PS0N2W09FFH

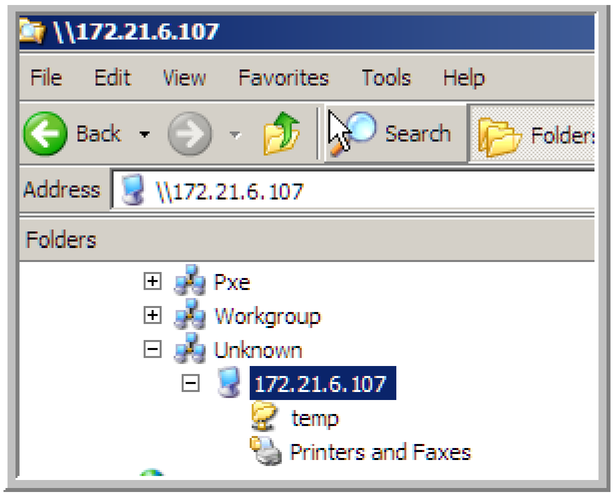
Connecting to 172.21.6.107

User name: Administrator

Password: ••••••••

Remember my password

The folder is now available via Windows explorer over the network.


\\172.21.6.107

Address \\172.21.6.107

Folders
- Pxe
- Workgroup
- Unknown
  - 172.21.6.107
    - temp
    - Printers and Faxes

1.1.3 XP Embedded Custom Image

Some end users may need components added or removed from the standard image of the APAX-557X. It is possible to have Advantech create a custom XP Embedded image to fit the needs of the user.

1.1.4 .Net Framework

The APAX-557X comes with .NET Framework version 2.0 installed. It is up to the end user to install later versions. Since there are so many different languages to support it is best left to the developer to decide what language of the .NET Redistributable to install. This will allow the image to stay at a reasonable size.

Setting the IP address

The APAX-557X comes with a default IP address of 10.0.0.1. This can be changed via the normal network settings of Windows XP.

1.1.5 Internet Information Server

The Advantech XP Embedded image comes with Microsoft Internet Information Server installed. This is helpful to take advantage of serving up web pages from the APAX. To access the control for IIS, right click My computer from the desktop and select Manage. This will open the Computer Management consol. For more information on how to use this server, please see the Microsoft web site.

Chapter 2

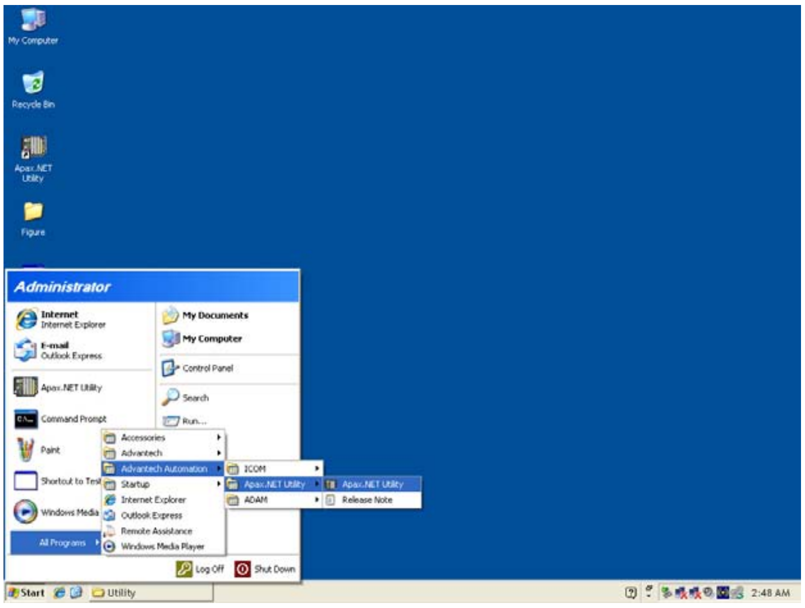
Utilities

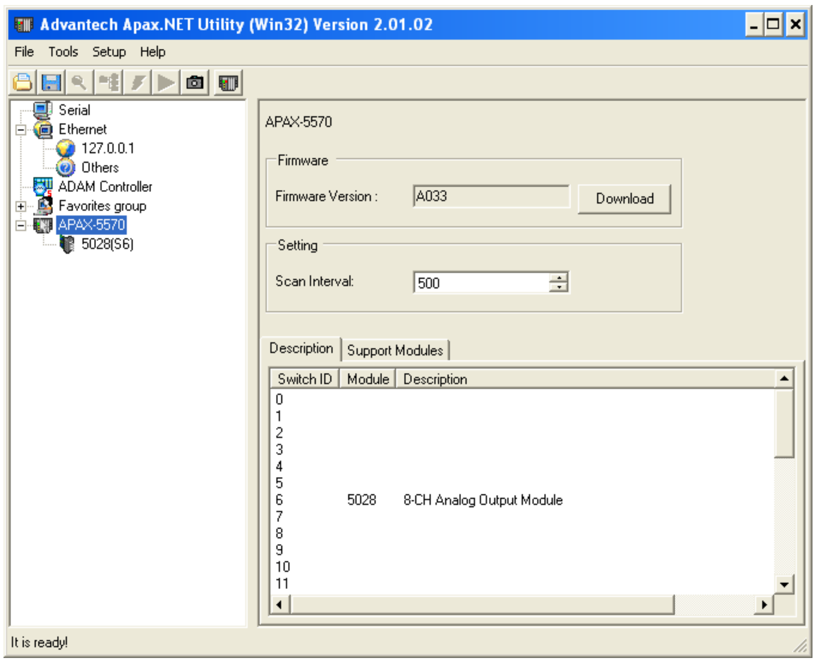
2.1 APAX.NET Utility

Advantech provides the APAX.NET utility which allows the developer/end user to interrogate the APAX bus, see connected modules, and do simple testing of the I/O. This software can be helpful when checking wiring inputs prior to installing the runtime project. It is also able to detect and test other Advantech supported hardware for this product such as Ethernet or Serial I/O.

The installation file is contained in the CD. When you launch the CD, select the **APAX Software** button and click the **APAX.NET Utility** button to find the installation file. After you complete the installation, you can launch APAX.NET utility by selecting Start >> All Programs >> Advantech Automation >> Apax.NET Utility >> Apax.NET Utility. Besides, you also can link to the web site: http://www.advantech.com and click into the Download Area under the Support site to get the latest version of the APAX.NET utility.

Detailed operation for APAX.NET utility can be found in Appendix B.





APAX-5570

Firmware

Firmware Version : A033 Download

Setting

Scan Interval: 500

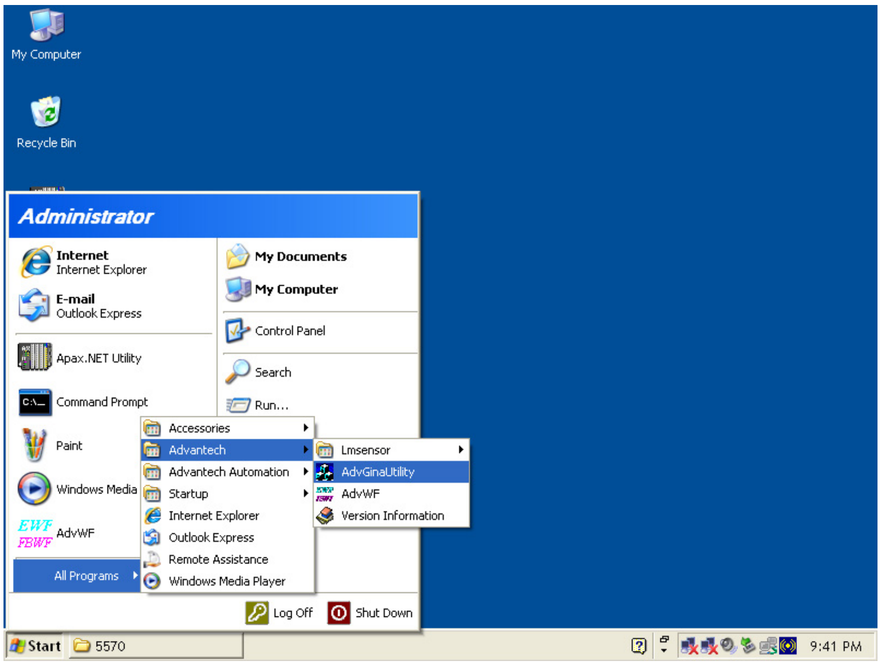
Description | Support Modules

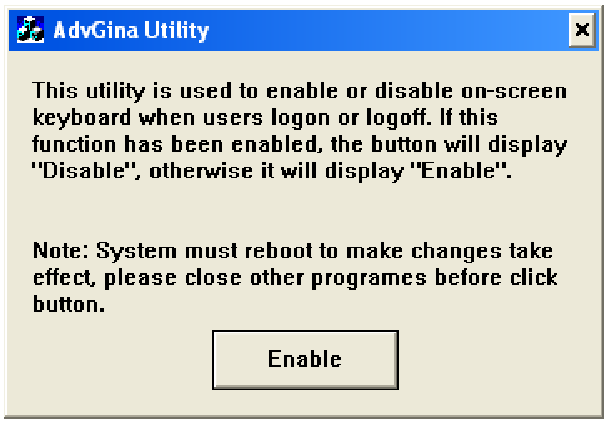
Switch ID	Module	Description
0		
1		
2		
3		
4		
5		
6	5028	8-CH Analog Output Module
7		
8		
9		
10		
11		

It is ready!

2.2 AdvGinaUtility

This utility is provided by Advantech to enable the On-Screen Keyboard during login for touch screen systems that don't have a keyboard connected. Launch the AdvGinaUtility by selecting Start >> All Programs >> Advantech >> AdvGinaUtility.



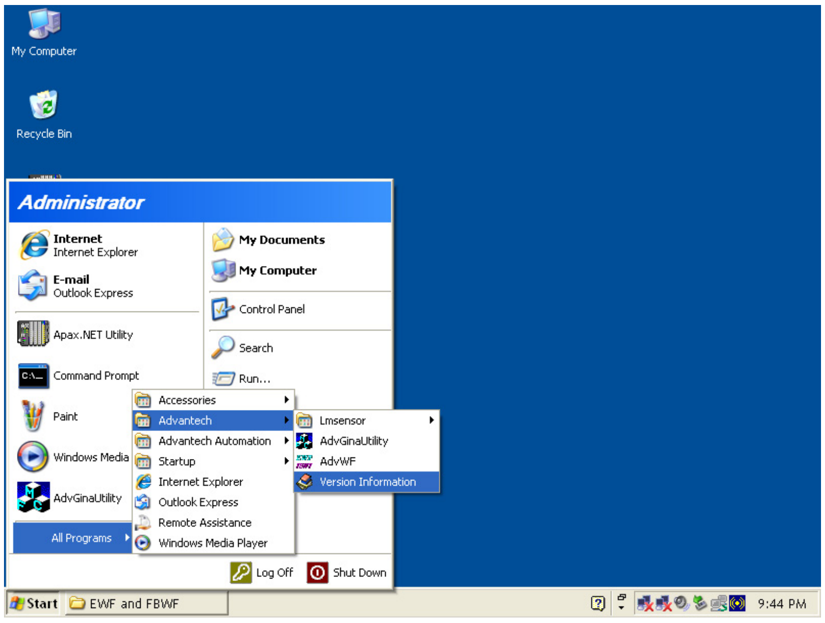


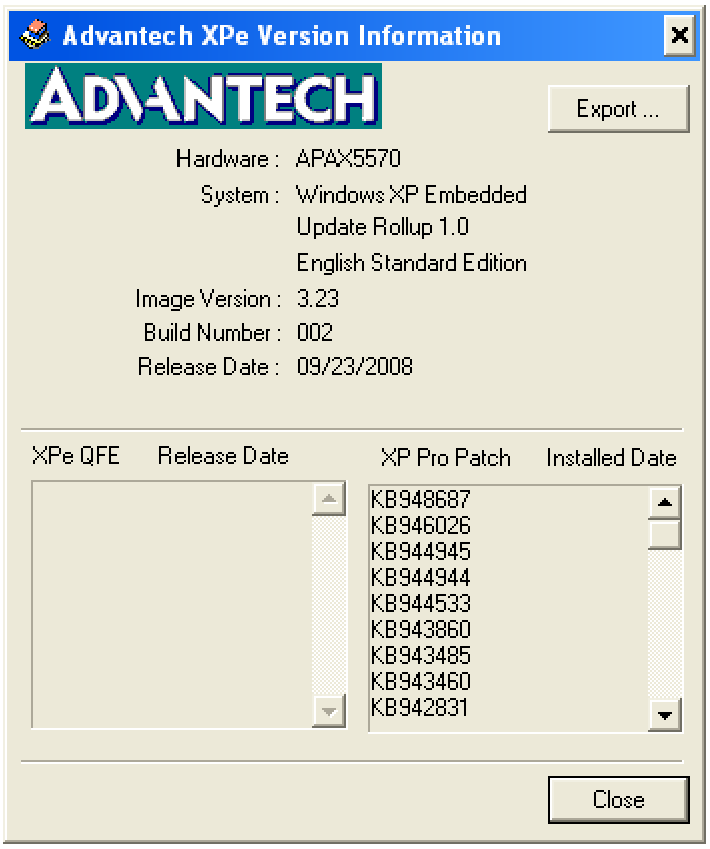
This utility is used to enable or disable on-screen keyboard when users logon or logoff. If this function has been enabled, the button will display "Disable", otherwise it will display "Enable".

Note: System must reboot to make changes take effect, please close other programes before click button.

2.3 Advantech Version Information Tool

Advantech provides a simple reporting tool that will provide necessary version information for the XP Embedded operating system. This is an important tool for determining what version of XPe is on the APAX-557X and may help during troubleshooting. Launch the Version Information Tool by selecting Start >> All Programs >> Advantech >> Version Information.





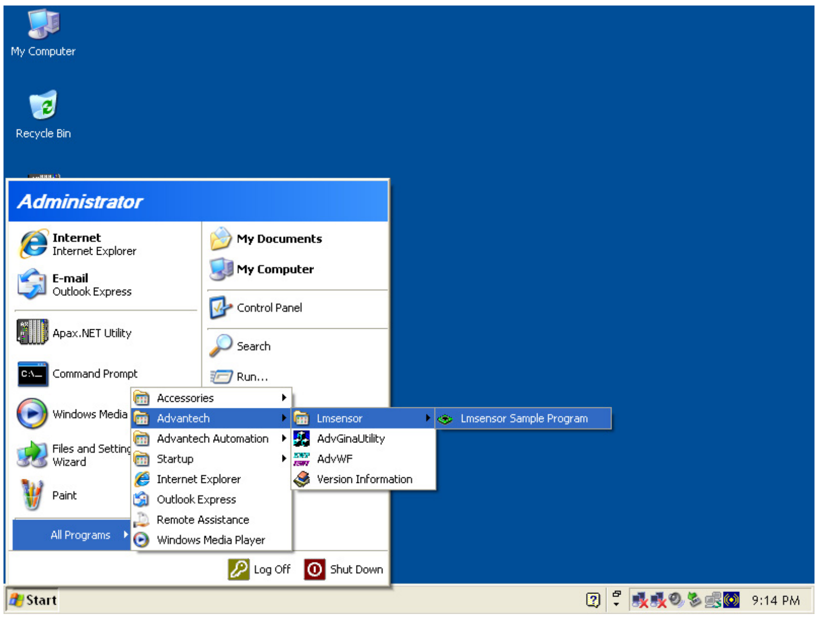
Hardware : APAX5570
System : Windows XP Embedded
Update Rollup 1.0
English Standard Edition
Image Version : 3.23
Build Number : 002
Release Date : 09/23/2008

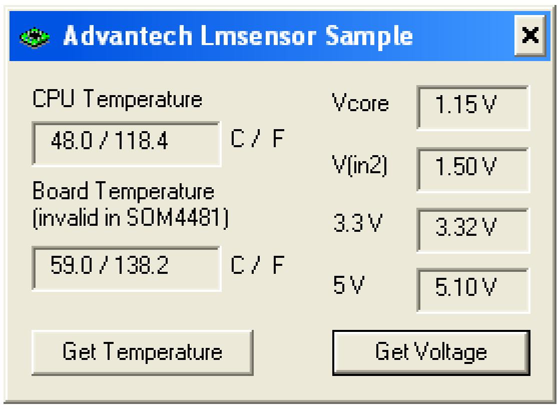
XPe QFE	Release Date

XP Pro Patch	Installed Date
KB948687	
KB946026	
KB944945	
KB944944	
KB944533	
KB943860	
KB943485	
KB943460	
KB942831	

2.4 Lmsensor Sample

The Lmsensor sample is a sample program that shows how to use the Lmsensor driver to read diagnostic information from the CPU board. It will read CPU internal temperature, CPU module board temperature and CPU operating voltage's by hardware sensors. With lmsensor, you can monitor the system status easily with your own application. The full source code is provided to allow a user to integrate this functionality into another application. Launch the Lmsensor Sample by selecting Start >> All Programs >> Advantech >> Lmsensor >> Lmsensor sample program.




Advantech Lmsensor Sample

CPU Temperature
48.0 / 118.4 C / F

Board Temperature
(invalid in SOM4481)
59.0 / 138.2 C / F

Vcore 1.15 V
V(in2) 1.50 V
3.3 V 3.32 V
5 V 5.10 V

Note!

Vcore: voltage on CPU
V(in2): voltage on North Bridge
3.3V: Voltage transferred from power input to 3.3 V onboard
5V: Voltage transferred from power input to 5 V onboard

Chapter 3

API Programming

3.1 VC++ API

Advantech provides a VC++ API for C/C++ development environment to control APAX-5000 I/O modules. The API has already been installed in APAX-557X. You also can install the VC++ API by the CD offered by APAX-557X. The installation file is contained in the CD. When you launch the CD, select the **APAX Software** button and click the **I/O Drivers** button to find the installation file.

In order to save your development time, Advantech provides several examples that you can use it as reference to build your own C or C++ application program. These examples can also be found in the CD offered by APAX-557X, or from the Advantech website at *http://www.advantech.com* in the download area under Support page. When you launch the CD, select the **APAX Software** button and click the **VC++ Example** button to find these examples.

3.1.1 ADAMDrvOpen

LONG ADS_API ADAMDrvOpen(LONG* handle);

Purpose:

Initialize the driver

Parameters:

handle = driver handle

Return

1. ERR_SUCCESS, Driver initialization succeeded, the handle will be valid for function use until closed.
2. ERR_INTERNAL_FAILED, Call GetLastError to get extended error information.

3.1.2 ADAMDrvClose

LONG ADS_API ADAMDrvClose(LONG* handle);

Purpose:

Initialize the driver

Parameters:

handle = driver handle

Return

1. ERR_SUCCESS, Driver termination succeeded
2. ERR_INTERNAL_FAILED, Call GetLastError to get extended error information.

3.1.3 SYS_SetInnerTimeout

LONG ADS_API SYS_SetInnerTimeout(LONG handle, WORD i_wTimeout);

Purpose:

Set the inner-timeout of the configuration functions that use internal communication channel. All functions with the exception of Get/Set values, use the internal communication network. When using any of those functions, they must wait for completion before returning. This sets the timeout value for returning.

Parameters:

handle = driver handle

i_wTimeout = inner-timeout, in millisecond. Default is 50 milliseconds.

Return

1. ERR_SUCCESS, Set timer succeeded
2. ERR_INTERNAL_FAILED, Call GetLastError to get extended error information.

3.1.4 SYS_GetModuleID

LONG ADS_API SYS_GetModuleID (LONG handle, WORD i_wSlot, DWORD* o_dwModuleID);

Purpose:

Get the module ID of the indicated slot.

Parameters:

handle = driver handle

i_wSlot = the slot ID which is ranged from 0 to 31.

o_dwModuleID = returned module ID

Return

1. ERR_SUCCESS, Module ID was found and returned
2. ERR_INTERNAL_FAILED, Call GetLastError to get extended error information.

3.1.5 SYS_GetSlotInfo

LONG ADS_API SYS_GetSlotInfo (LONG handle, WORD i_wSlot, struct SlotInfo* o_stSlotInfo);

Purpose:

Get the module information of the indicated slot.

Parameters:

handle = driver handle

i_wSlot = the slot ID which is ranged from 0 to 31.

o_stSlotInfo = returned SlotInfo structure.

Return

1. ERR_SUCCESS, o_stSlotInfo contains slot information.
2. ERR_INTERNAL_FAILED, Call GetLastError to get extended error information.

3.1.6 SYS_GetAllSlotErrorFlag

LONG ADS_API SYS_GetAllSlotErrorFlag(LONG handle,

DWORD* o_wError);

Purpose:

Get the presence of a module for each slot.

Parameters:

handle = driver handle

o_wError = Return value for all slots status. From LSB to MSB of the value indicates the slot-0 to slot-31 status. If the bit is 1, it means that the slot has no module present.

Return

1. ERR_SUCCESS,
2. ERR_INTERNAL_FAILED, Call GetLastError to get extended error information.

3.1.7 AIO_GetValue

LONG ADS_API AIO_GetValue(LONG handle,

WORD i_wSlot,

WORD i_wChannel,

WORD* o_wValue);

Purpose:

Get a single analog input or output value from the indicated slot and channel.

Parameters:

handle = driver handle

i_wSlot = the slot ID which has a range of 0 to 31.

i_wChannel = the channel ID which has a range of 0 to 31.

o_wValue = the variable to hold the returned AIO value.

Return

1. ERR_SUCCESS, o_wValue contains AIO value.
2. ERR_INTERNAL_FAILED, Call GetLastError to get extended error information.

3.1.8 AIO_GetValues

LONG ADS_API AIO_GetValues(LONG handle,

WORD i_wSlot,

WORD* o_wValues);

Purpose:

Get the all analog input or output values of the indicated slot.

Parameters:

handle = driver handle

i_wSlot = the slot ID which is ranged from 0 to 31.

o_wValues = the variables array to hold the returned AIO values. The size of this array must be at least 32 WORD's.

Return

1. ERR_SUCCESS, o_wValue contains AIO values from channel-0 to the last channel.
2. ERR_INTERNAL_FAILED, Call GetLastError to get extended error information.

3.1.9 AIO_SetRanges

LONG ADS_API AIO_SetRanges(LONG handle,

WORD i_wSlot,

WORD i_wChannelTotal,

WORD* i_wRanges);

Purpose:

Set the channel ranges of the indicated slot.

Parameters:

handle = driver handle

i_wSlot = the slot ID which is ranged from 0 to 31.

i_wChannelTotal = the channel total of the module in the indicated slot.

i_wRanges = the ranges to be set. The size of this array must be i_wChannelTotal WORDs. See APPENDIX A for valid range settings.

Return

1. ERR_SUCCESS, setting ranges succeeded.
2. ERR_INTERNAL_FAILED, Call GetLastError to get extended error information.

3.1.10 AIO_SetZeroCalibration

LONG ADS_API AIO_SetZeroCalibration(LONG handle,

WORD i_wSlot,

WORD i_wChannel,

WORD i_wType);

Purpose:

Run the zero calibraion of the indicated slot and channel.

Parameters:

handle = driver handle

i_wSlot = the slot ID which is ranged from 0 to 31.

i_wChannel = the channel ID which is ranged from 0 to 31.

i_wType = the type value to be set. Currently, it is ingnored.

Return

1. ERR_SUCCESS, setting zero calibration succeeded.
2. ERR_INTERNAL_FAILED, Call GetLastError to get extended error information.

3.1.11 AIO_SetSpanCalibration

LONG ADS_API AIO_SetSpanCalibration(LONG handle,
WORD i_wSlot,
WORD i_wChannel,
WORD i_wType);

Purpose:

Run the span calibraion of the indicated slot and channel.

Parameters:

handle = driver handle

i_wSlot = the slot ID which is ranged from 0 to 31.

i_wChannel = the channel ID which is ranged from 0 to 31.

i_wType = the type value to be set. Currently, it is ingnored.

Return

1. ERR_SUCCESS, setting span calibration succeeded.
2. ERR_INTERNAL_FAILED, Call GetLastError to get extended error information.

3.1.12 AIO_GetChannelStatus

LONG ADS_API AIO_GetChannelStatus(LONG handle,
WORD i_wSlot,
BYTE* o_byStatus);

Purpose:

Get all channels status of the indicated slot.

Parameters:

handle = driver handle

i_wSlot = the slot ID which is ranged from 0 to 31.

o_byStatus = the array to hold the returned channels status. The size of this array must be at least 32 BYTEs.

Return

1. ERR_SUCCESS, channel status succeeded.

The value of o_byStatus indicates:

0: None

1: Normal

2: Over current

3: Under current

4: Burn out

5: Open loop

6: Not ready

2. ERR_INTERNAL_FAILED, Call GetLastError to get extended error information.

3.1.13 AI_SetChannelMask

LONG ADS_API AI_SetChannelMask(LONG handle,

WORD i_wSlot,

DWORD i_dwMask);

Purpose:

Set enabled AI channel mask of the indicated slot.

Parameters:

handle = driver handle

i_wSlot = the slot ID which is ranged from 0 to 31.

i_dwMask = the enabled AI channel mask to be set. From LSB to MSB of the value indicate the slot-0 to slot-31 enabled mask. If the bit is 1, it means that the channel is enabled.

Return

1. ERR_SUCCESS, setting channel mask succeeded.
2. ERR_INTERNAL_FAILED, Call GetLastError to get extended error information.

3.1.14 AI_SetIntegrationTime

LONG ADS_API AI_SetIntegrationTime(LONG handle,

WORD i_wSlot,

DWORD i_dwIntegration);

Purpose:

Set AI integration time of the indicated slot.

Parameters:

handle = driver handle

i_wSlot = the slot ID which is ranged from 0 to 31.

i_dwIntegration = the AI integration time to be set. Two settings are availabled

0 = 60Hz

1 = 50Hz

Return

1. ERR_SUCCESS, setting integration time succeeded.
2. ERR_INTERNAL_FAILED, Call GetLastError to get extended error information.

3.1.15 AI_SetAutoCalibration

LONG ADS_API AI_SetAutoCalibration(LONG handle,

WORD i_wSlot);

Purpose:

Set to run the auto calibraion of the indicated slot.

Parameters:

handle = driver handle

i_wSlot = the slot ID which is ranged from 0 to 31.

Return

1. ERR_SUCCESS, setting auto calibration succeeded.
2. ERR_INTERNAL_FAILED, Call GetLastError to get extended error information.

3.1.16 AO_SetCalibrationMode

LONG ADS_API AO_SetCalibrationMode(LONG handle,
WORD i_wSlot);

Purpose:

Set to switch to the AO calibraion mode of the indicated slot.

Parameters:

handle = driver handle

i_wSlot = the slot ID which is ranged from 0 to 31.

Return

1. ERR_SUCCESS, setting calibration mode succeeded.
2. ERR_INTERNAL_FAILED, Call GetLastError to get extended error information.

3.1.17 AO_GetStartupValues

LONG ADS_API AO_GetStartupValues(LONG handle,
WORD i_wSlot,
WORD i_wChannelTotal,
WORD* o_wValues);

Purpose:

Get the AO startup values of the indicated slot.

Parameters:

handle = driver handle

i_wSlot = the slot ID which is ranged from 0 to 31.

i_wChannelTotal = the channel total of the module in the indicated slot.

o_wValues = the variables array to hold the AO startup values. The size of this array must be at least 32 WORDs.

Return

1. ERR_SUCCESS, Geting values succeeded. o_wValues contains AO startup values from channel-0 to the last channel.
2. ERR_INTERNAL_FAILED, Call GetLastError to get extended error information.

3.1.18 AO_SetStartupValues

LONG ADS_API AO_SetStartupValues(LONG handle,
WORD i_wSlot,
WORD i_wChannelTotal,
WORD* i_wValues);

Purpose:

Set the AO startup values of the indicated slot. These values are stored in onboard flash and are initialized to the slot upon boot up of the hardware.

Parameters:

handle = driver handle

i_wSlot = the slot ID which is ranged from 0 to 31.

i_wChannelTotal = the channel total of the module in the indicated slot.

i_wValues = the values array to be set. The size of this array must be i_wChannelTotal WORDs.

Return

1. ERR_SUCCESS, setting values succeeded.
2. ERR_INTERNAL_FAILED, Call GetLastError to get extended error information.

3.1.19 AO_SetValue

LONG ADS_API AO_SetValue(LONG handle,
WORD i_wSlot,
WORD i_wChannel,
WORD i_wValue);

Purpose:

Set a single AO value of the indicated slot and channel.

Parameters:

handle = driver handle

i_wSlot = the slot ID which is ranged from 0 to 31.

i_wChannel = the channel ID which is ranged from 0 to 31.

i_wValue = the AO value to be set.

Return

1. ERR_SUCCESS,
2. ERR_INTERNAL_FAILED, Call GetLastError to get extended error information.

3.1.20 AO_SetValues

LONG ADS_API AO_SetValues(LONG handle,

WORD i_wSlot,

DWORD i_dwMask,

WORD* i_wValues);

Purpose:

Set multiple AO values of the indicated slot.

Parameters:

handle = driver handle

i_wSlot = the slot ID which is ranged from 0 to 31.

i_dwMask = the channels mask. From LSB to MSB of the value indicate the slot-0 to slot-31 mask. If the bit is 1, it means that the channel must change value.

i_wValues = the AO values to be set. This is a pointer to an array of 32 words.

Return

1. ERR_SUCCESS, setting values succeeded.
2. ERR_INTERNAL_FAILED, Call GetLastError to get extended error information.

3.1.21 AO_BufValues

LONG ADS_API AO_BufValues(LONG handle,

WORD i_wSlot,

DWORD i_dwMask,

WORD* i_wValues);

Purpose:

Buffer the AO values of the indicated slot. This function is used along with OUT_FlushBufValues for a synchronized write Output. Once all slots are buffered, then OUT_FlushBufValues function triggers the synchronized buffer write of all masked slots.

Parameters:

handle = driver handle

i_wSlot = the slot ID which is ranged from 0 to 31.

i_dwMask = the channels mask. From LSB to MSB of the value indicate the slot-0 to slot-31 mask. If the bit is 1, it means that the channel must buffer value.

i_wValues = the AO values to be bufferred.

Return

1. ERR_SUCCESS, bufferring values succeeded.
2. ERR_INTERNAL_FAILED, Call GetLastError to get extended error information.

3.1.22 DIO_GetValue

LONG ADS_API DIO_GetValue(LONG handle,
WORD i_wSlot,
WORD i_wChannel,
BOOL* o_bValue);

Purpose:

Get a single DIO value of the indicated slot and channel.

Parameters:

handle = driver handle

i_wSlot = the slot ID which is ranged from 0 to 31.

i_wChannel = the channel ID which is ranged from 0 to 31.

o_bValue = the variable to hold the DIO value.

Return

1. ERR_SUCCESS, geting value succeeded o_wValue contains DIO value.
2. ERR_INTERNAL_FAILED, Call GetLastError to get extended error information.

3.1.23 DIO_GetValues

LONG ADS_API DIO_GetValues(LONG handle,
WORD i_wSlot,
DWORD* o_dwHighValue,
DWORD* o_dwLowValue);

Purpose:

Get the all DIO values of the indicated slot.

Parameters:

handle = driver handle

i_wSlot = the slot ID which is ranged from 0 to 31.

o_dwHighValue = the variable to hold the returned DIO values from channel 32 to 63. The LSB indicates the channel-32.

o_dwLowValue = the variable to hold the returned DIO values from channel 0 to 31. The LSB indicates the channel-0.

Return

1. ERR_SUCCESS, Get values succeeded. The o_dwHighValue and o_dwLowValue contain DIO values from channel-0 to the last channel.
2. ERR_INTERNAL_FAILED, Call GetLastError to get extended error information.

3.1.24 DI_GetFilters

LONG ADS_API DI_GetFilters(LONG handle,

WORD i_wSlot,

DWORD* o_dwHighMask,

DWORD* o_dwLowMask,

DWORD* o_dwWidth);

Purpose:

Get the DI filter mask and width of the indicated slot. All channels use the same filter.

Parameters:

handle = driver handle

i_wSlot = the slot ID which is ranged from 0 to 31.

o_dwHighMask = RESERVED

o_dwLowMask = If set to zero, filter is disabled. Non-zero indicates that filter is applied.

o_dwWidth = the variable to hold the DI filter width. Filter is in .1msec units and value of filter width must be in multiples of 5.

Return

1. ERR_SUCCESS, get filters succeeded.
2. ERR_INTERNAL_FAILED, Call GetLastError to get extended error information.

3.1.25 DI_SetFilters

LONG ADS_API DI_SetFilters(LONG handle,

WORD i_wSlot,

DWORD i_dwHighMask,

DWORD i_dwLowMask,

DWORD i_dwWidth);

Purpose:

Set the DI filter mask and width of the indicated slot. Filter is amount of time needed to verify a change of state. This is to reduce noise.

Parameters:

handle = driver handle

i_wSlot = the slot ID which is ranged from 0 to 31.

i_dwHighMask = RESERVED

i_dwLowMask = If set to zero, filter is disabled. Non-zero indicates that filter is applied.

i_dwWidth = the variable to hold the DI filter width. Filter is in .1msec units and value of filter width must be in multiples of 5.

Return

1. ERR_SUCCESS, setting filter succeeded.
2. ERR_INTERNAL_FAILED, Call GetLastError to get extended error information.

3.1.26 DO_SetValue

LONG ADS_API DO_SetValue(LONG handle,
WORD i_wSlot,
WORD i_wChannel,
BOOL i_bValue);
Purpose:
Set a single DO value of the indicated slot and channel.

Parameters:
handle = driver handle
i_wSlot = the slot ID which is ranged from 0 to 31.
i_wChannel = the channel ID which is ranged from 0 to 31.
i_bValue = the DO value to be set.

Return
1) ERR_SUCCESS, setting value succeeded.
2) ERR_INTERNAL_FAILED, Call GetLastError to get extended error information.

3.1.27 DO_SetValues

LONG ADS_API DO_SetValues(LONG handle,
WORD i_wSlot,
DWORD i_dwHighValue,
DWORD i_dwLowValue);

Purpose:

Set all DO values of the indicated slot.

Parameters:

handle = driver handle
i_wSlot = the slot ID which is ranged from 0 to 31.
i_dwHighValue = the DI values from channel 32 to 63 to be set. The LSB indicates the channel-32.
i_dwLowValue = the DI values from channel 0 to 31 to be set. The LSB indicates the channel-0.

Return

1. ERR_SUCCESS, setting values succeeded.
2. ERR_INTERNAL_FAILED, Call GetLastError to get extended error information.

3.1.28 DO_BufValues

LONG ADS_API DO_BufValues(LONG handle,
WORD i_wSlot,
DWORD i_dwHighValue,
DWORD i_dwLowValue);

Purpose:

Buffer the DO values of the indicated slot.

Parameters:

handle = driver handle

i_wSlot = the slot ID which is ranged from 0 to 31.

i_dwHighValue = the DI values from channel 32 to 63 to be bufferred. The LSB indicates the channel-32.

i_dwLowValue = the DI values from channel 0 to 31 to be bufferred. The LSB indicates the channel-0.

Return

1. ERR_SUCCESS, bufferring values succeeded.
2. ERR_INTERNAL_FAILED, Call GetLastError to get extended error information.

3.1.29 OUT_FlushBufValues

LONG ADS_API OUT_FlushBufValues(LONG handle,
DWORD i_dwSlotMask);

Purpose:

Flush the bufferred values. This triggers all buffered values to write simultaniously.

Parameters:

handle = driver handle

i_dwSlotMask = the flush slot enable mask. The LSB indicates the slot-0.

Return

1. ERR_SUCCESS, flushing values succeeded.
2. ERR_INTERNAL_FAILED, Call GetLastError to get extended error information.

3.1.30 Modbus Functions

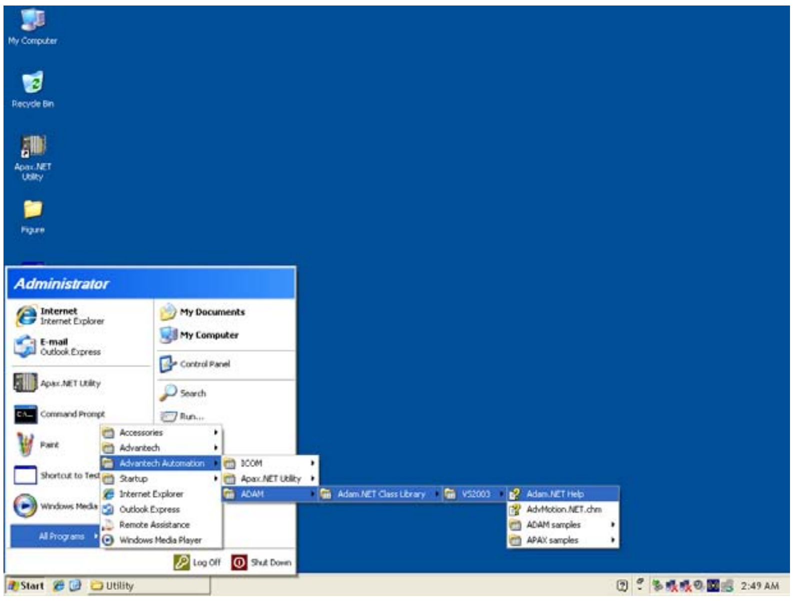
The Modbus functions' reference manual is located on the CD that comes with the APAX-557X. When you launch the CD, click the **Browser Manual** button and the you can see the document **APAX Modbus Library Manual.pdf** there.

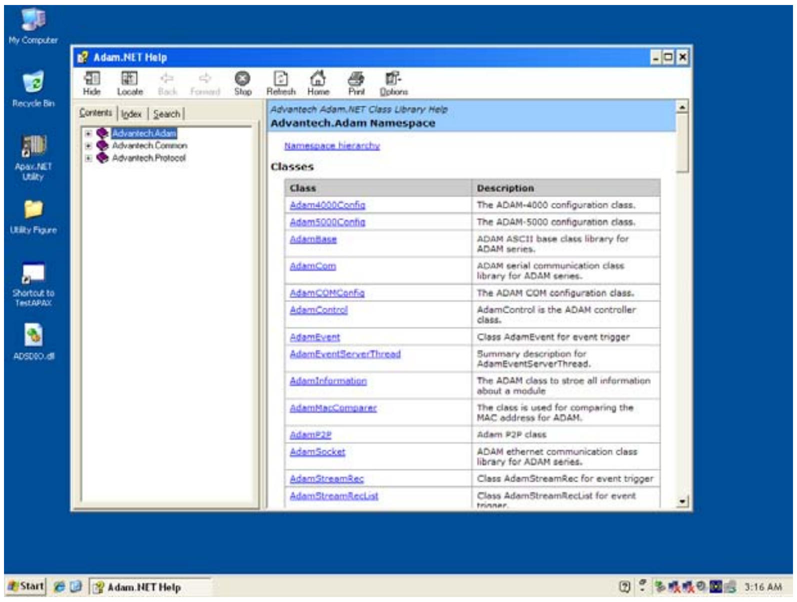
3.2 .NET API (Adam .NET Class Library)

Advantech provides a .NET API for developing .NET applications for many Advantech products. This API interface is called Adam .NET Class library. All the functions described in Section 3.1 are supported by Adam .NET class library. You can leverage Advantech Adam .NET class library to develop application controlling APAX-5000 I/O modules under Microsoft Visual Studio .NET environment such as VB .NET or C#.

The installation file is contained in the CD. When you launch the CD, select the APAX Software button and click the .NET Class Library button to find the installation file. Besides, you also can link to the website: http://www.advantech.com and click into the Download Area under the Support site to get the latest version of the Adam.NET class library.

After you complete the installation, you can find Adam .NET class library help document by selecting Start >> All Programs >> Advantech Automation >> ADAM >> Adam.NET Class Library >> VS2003 >> Adam.NET Help.





Advantech Adam.NET Class Library Help

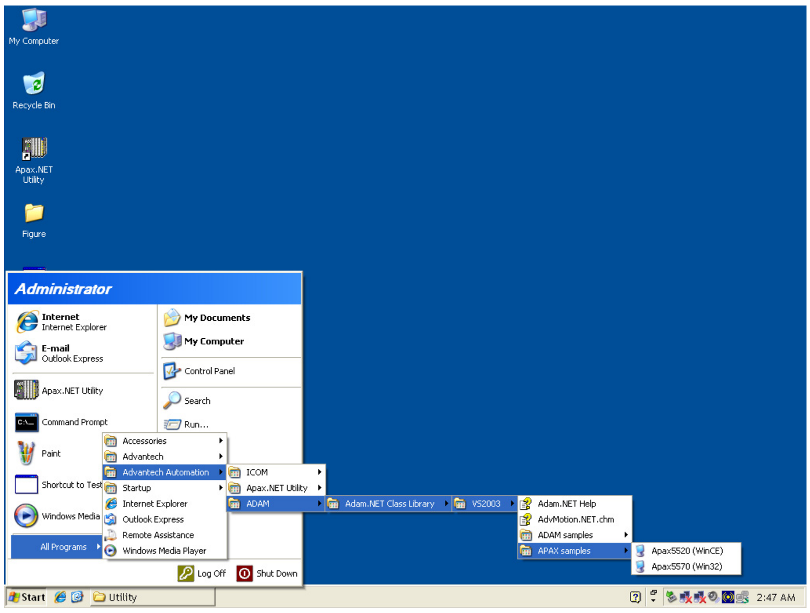
Advantech.Adam Namespace

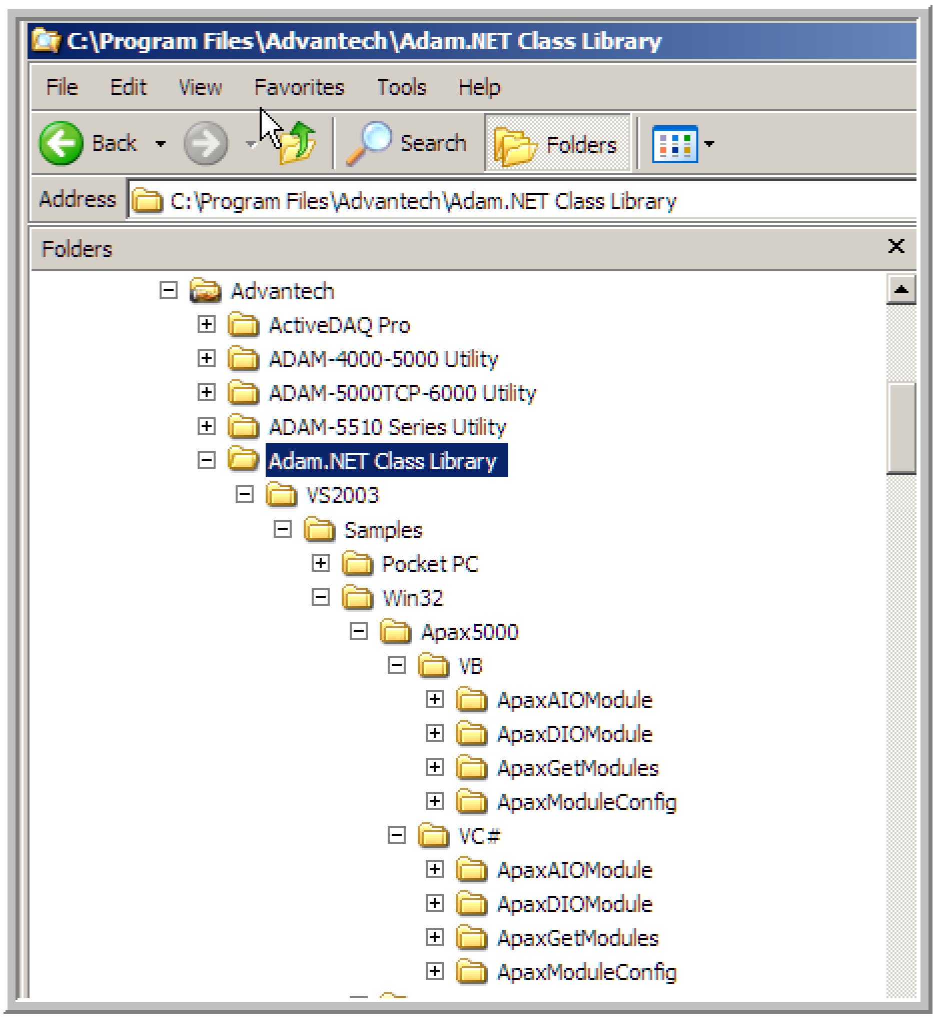
Namespace hierarchy

Classes

Class	Description
Adam4000Config	The ADAM-4000 configuration class.
Adam5000Config	The ADAM-5000 configuration class.
AdamBase	ADAM ASCII base class library for ADAM series.
AdamCom	ADAM serial communication class library for ADAM series.
AdamCOMConfig	The ADAM COM configuration class.
AdamControl	AdamControl is the ADAM controller class.
AdamEvent	Class AdamEvent for event trigger
AdamEventServerThread	Summary description for AdamEventServerThread.
AdamInformation	The ADAM class to stroe all information about a module
AdamMacComparer	The class is used for comparing the MAC address for ADAM.
AdamP2P	Adam P2P class
AdamSocket	ADAM ethernet communication class library for ADAM series.
AdamStreamRec	Class AdamStreamRec for event trigger
AdamStreamRecList	Class AdamStreamRecList for event trigger.

Besides, there are many examples offered that you can use it as reference to build your own application program. These examples can also be found by selecting Start >> All Programs >> Advantech Automation >> ADAM >> Adam.NET Class Library >> VS2003 >> APAX samples >> Apax5570 (Win32) after you have installed Adam.NET Class library. Or you can find these examples by C:\Program Files\Advantech\Adam.NET Class Library\.





Appendix A

Analog I/O Board Settings

A.1 Analog I/O Board Settings

Range Settings for Analog I/O Boards. These ranges are provided for reference. Not all boards support all ranges. Please see hardware manual for valid ranges for a particular board.

	Setting Type	Value (Hex)
Millivolts DC (mV)	+/- 15mV	0x0100
	+/- 50mV	0x0101
	+/- 100mV	0x0102
	+/- 150mV	0x0103
	+/- 500mV	0x0104
	0~150mV	0x0105
	0~500mV	0x0106
Volts DC (V)	+/- 1V	0x0140
	+/- 2.5V	0x0141
	+/- 5V	0x0142
	+/- 10V	0x0143
	+/- 15V	0x0144
	0~1V	0x0145
	0~2.5V	0x0146
	0~5V	0x0147
	0~10V	0x0148
	0~15V	0x0149
Milliamps (mA)	4~20mA	0x0180
	+/-20mA	0x0181
	0~20mA	0x0182
Counter settings	Pulse/DIR	0x01C0
	Up/Down	0x01C1
	Up	0x01C2
	Frequency	0x01C3
	AB 1X	0x01C4
	AB 2X	0x01C5
	AB 4X	0x01C6
Pt-100 (3851)	Pt-100 (3851) -200~850 'C	0x0200
	Pt-100 (3851) -120~130 'C	0x0201
	Pt-100 (3851) -200~200 'C	0x0202
	Pt-100 (3851) -100~100 'C	0x0203
	Pt-100 (3851) -50~150 'C	0x0204
	Pt-100 (3851) 0~100 'C	0x0205
	Pt-100 (3851) 0~200 'C	0x0206
	Pt-100 (3851) 0~400 'C	0x0207
	Pt-100 (3851) 0~600 'C	0x0208
Pt-200 (3851)	Pt-200 (3851) -200~850 'C	0x0220
	Pt-200 (3851) -120~130 'C	0x0221
Pt-500 (3851)	Pt-500 (3851) -200~850 'C	0x0240
	Pt-500 (3851) -120~130 'C	0x0241

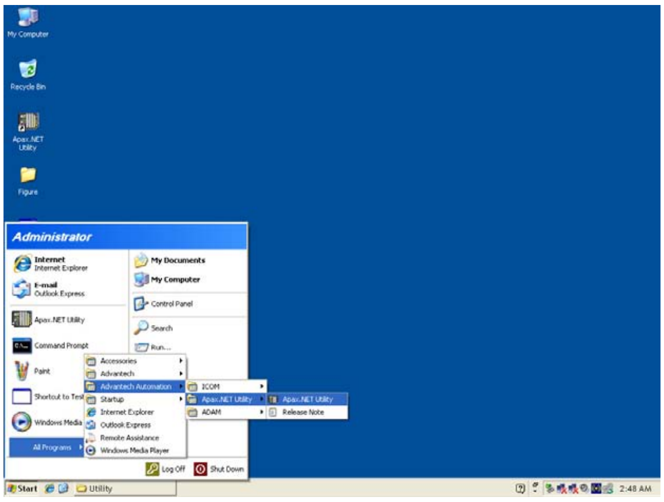
Pt-1000 (3851)	Pt-1000 (3851) -200~850 'C	0x0260
	Pt-1000 (3851) -120~130 'C	0x0261
	Pt-1000 (3851) -40~160 'C	0x0262
Pt-100 (3916)	Pt-100 (3916) -200~850 'C	0x0280
	Pt-100 (3916) -120~130 'C	0x0281
	Pt-100 (3916) -200~200 'C	0x0282
	Pt-100 (3916) -100~100 'C	0x0283
	Pt-100 (3916) -50~150 'C	0x0284
	Pt-100 (3916) 0~100 'C	0x0285
	Pt-100 (3916) 0~200 'C	0x0286
	Pt-100 (3916) 0~400 'C	0x0287
	Pt-100 (3916) 0~600 'C	0x0288
Pt-200 (3916)	Pt-200 (3916) -200~850 'C	0x02A0
	Pt-200 (3916) -120~130 'C	0x02A1
Pt-500 (3916)	Pt-500 (3916) -200~850 'C	0x02C0
	Pt-500 (3916) -120~130 'C	0x02C1
Pt-1000 (3916)	Pt-1000 (3916) -200~850 'C	0x02E0
	Pt-1000 (3916) -120~130 'C	0x02E1
	Pt-1000 (3916) -40~160 'C	0x02E2
Balco 500	Balcon(500) -30~120	0x0300
Ni 518	Ni(518) -80~100 'C	0x0320
	Ni(518) 0~100 'C	0x0321
Ni 508	Ni(508) 0~100 'C	0x0340
	Ni(508) -50~200 'C	0x0341
Thermistor 3K	Thermistor 3K 0~100 'C	0x0360
Thermistor 10K	Thermistor 10K 0~100 'C	0x0380
	Thermistor 10K -50~100 'C	0x0381
T/C TypeJ	T/C TypeJ 0~760 'C	0x0400
	T/C TypeJ -200~1200 'C	0x0401
T/C TypeK	T/C TypeK 0~1370 'C	0x0420
	T/C TypeK -270~1372 'C	0x0421
T/C TypeT	T/C TypeT -100~400 'C	0x0440
	T/C TypeT -270~400 'C	0x0441
T/C TypeE	T/C TypeE 0~1000 'C	0x0460
	T/C TypeE -270~1000 'C	0x0461
T/C TypeR	T/C TypeR 500~1750 'C	0x0480
	T/C TypeR 0~1768	0x0481
T/C TypeS	T/C TypeS 500~1750 'C	0x04A0
	T/C TypeS 0~1768 'C	0x04A1
T/C TypeB	T/C TypeB 500~1800 'C	0x04C0
	T/C TypeB 300~1820 'C	0x04C1

Appendix B

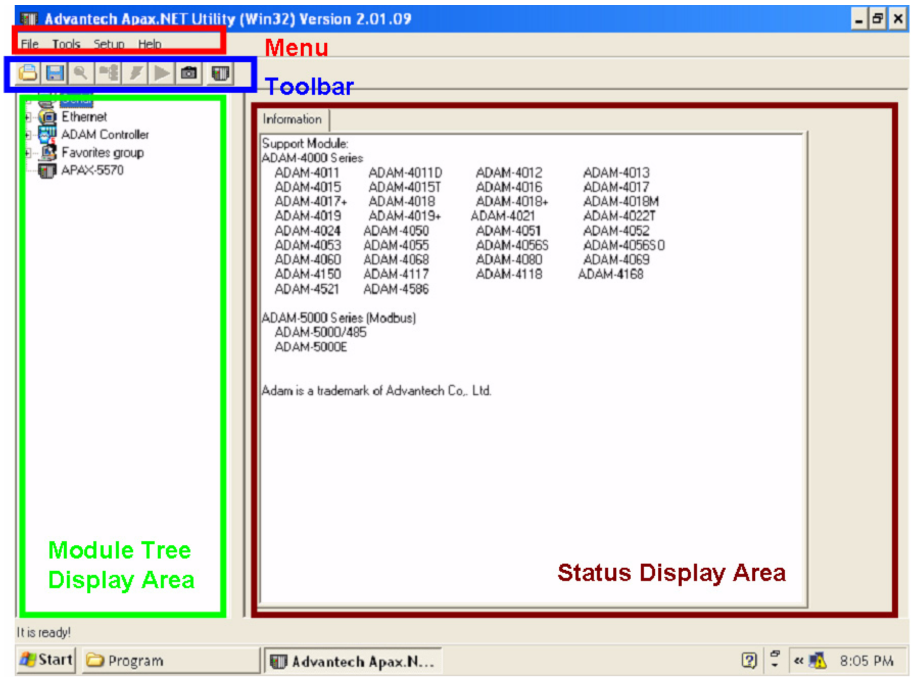
APAX.NET Utility Operation

B.1 APAX.NET Utility General Window

After you install the APAX.NET utility, you can launch it by selecting Start >> All Programs >> Advantech Automation >> Apax.NET Utility >> Apax.NET Utility. Refer to Section 2.1 for installation information.



After you launch the utility, you should see the operation window as figure below. Except APXA-5000 I/O modules, other devices such as ADAM-4000, ADAM-5000 and ADAM-6000 modules can also be searched and configured in this utility.



```
Support Module:
ADAM-4000 Series
    ADAM-4011     ADAM-4011D    ADAM-4012     ADAM-4013
    ADAM-4015     ADAM-4015T    ADAM-4016     ADAM-4017
    ADAM-4017+    ADAM-4018     ADAM-4018+    ADAM-4018M
    ADAM-4019     ADAM-4019+    ADAM-4021     ADAM-4022T
    ADAM-4024     ADAM-4050     ADAM-4051     ADAM-4052
    ADAM-4053     ADAM-4055     ADAM-4056S    ADAM-4056SO
    ADAM-4060     ADAM-4068     ADAM-4080     ADAM-4069
    ADAM-4150     ADAM-4117     ADAM-4118     ADAM-4168
    ADAM-4521     ADAM-4586

ADAM-5000 Series (Modbus)
    ADAM-5000/485
    ADAM-5000E

Adam is a trademark of Advantech Co,. Ltd.
```

The operation window consists of four areas --- the **Menu**, the **Toolbar**, the **Module Tree Display Area** and the **Status Display Area**.

B.1.1 Menu

The menu at the top of the operation window contains:

- The **File** menu

1. **Open Favorite Group** - You can configure your favorite group and save the configuration into one file. Using this option, you can load your configuration file for favorite group.
2. **Save Favorite Group** - You can configure your favorite group and save the configuration into one file. Using this option, you can save your favorite group into one configuration file.
3. **Auto-Initial Group** - If you want to have the same favorite group configuration when you exit APAX.NET utility and launch it again, you need to check this option.
4. **Exit** - Exit APAX.NET Utility.

- The **Tools** menu

1. **Search** - Search if there are any remote I/O modules connected. For I/O modules communicated by serial (such as ADAM-4000 modules), click the **COM1** item (COM 2 is an internal COM port) under **Serial** item in the **Module Tree Display Area** first before you click this button. For I/O modules communicated by Ethernet (such as ADAM-6000 modules), click the **Ethernet** item in the **Module Tree Display Area** first before you click this button.
2. **Add Devices to Group** - You can add any I/O modules to your favorite group by this option. You need to select the device you want to add in the **Module Tree Display Area** (it will be described below) first, and then select this option to add.
3. **Terminal for Command Testing** - ADAM modules support ASCII commands and Modbus as communication protocol. You can launch the terminal to communicate with remote module by these two kinds of protocols directly. Refer to ADAM-4000, ADAM-5000 and ADAM-6000 manual for ASCII and Modbus command.
4. **Monitor Stream/ Event Data** - ADAM-6000 modules support Data Stream function. You can use this to configure related setting for the connected ADAM-6000 modules connected. Refer to ADAM-6000 manual for more detail.

- The **Setup** menu

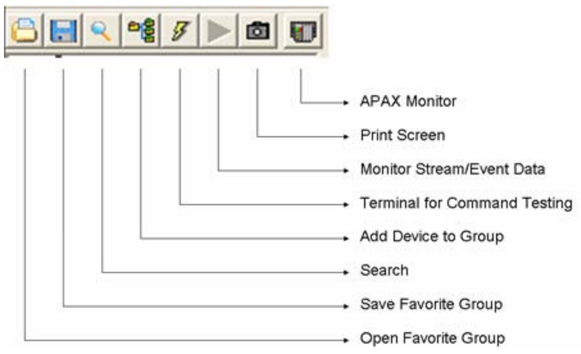
1. **Favorite Group** - You can configure your favorite group including add one new device (only for remote device), modify or delete one current device, sort current devices and diagnose connection to one device.
2. **Refresh COM and LAN node** - APAX.NET utility will refresh the serial and LAN network connection situation.
3. **ShowTreeView** - Check this option to display the **Module Tree Display Area** or not.
4. **Add COM Port Tree Nodes** - This option is used to add serial COM ports in APAX.NET Utility.
5. **Delete the COM Port** - This option is used to delete serial COM ports in APAX.NET Utility.

- The **Help** menu

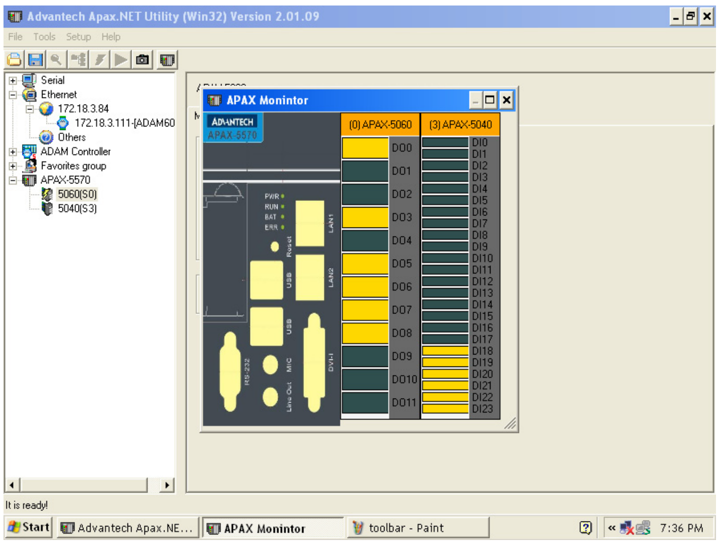
1. **Check Up-to-Date on the Web** - Choose this option, it will automatically connect to Advantech download website. You can download the latest utility there.
2. **About Apax.NET Utility** - Choose this option, you can see version of APAX.NET Utility installed on your computer.

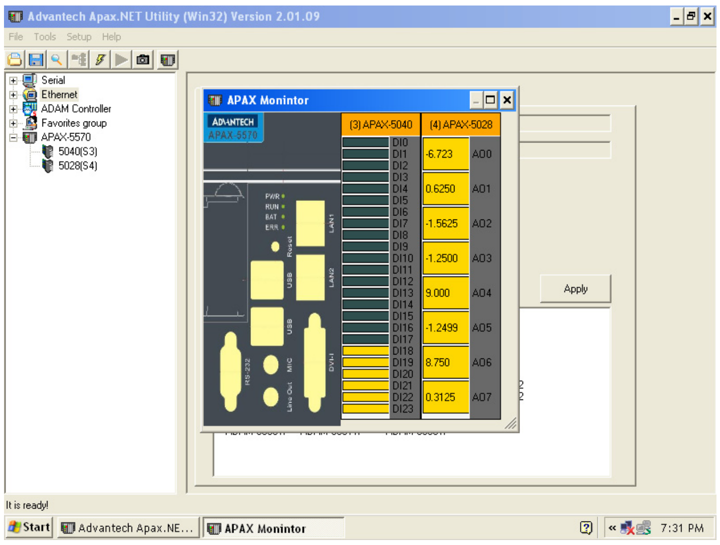
B.1.2 Toolbar

The eight buttons on toolbar represent the six common used items from the **Menu** and two new functions: Print Screen and APAX Monitor. Refer to figure below for the definition of each button:

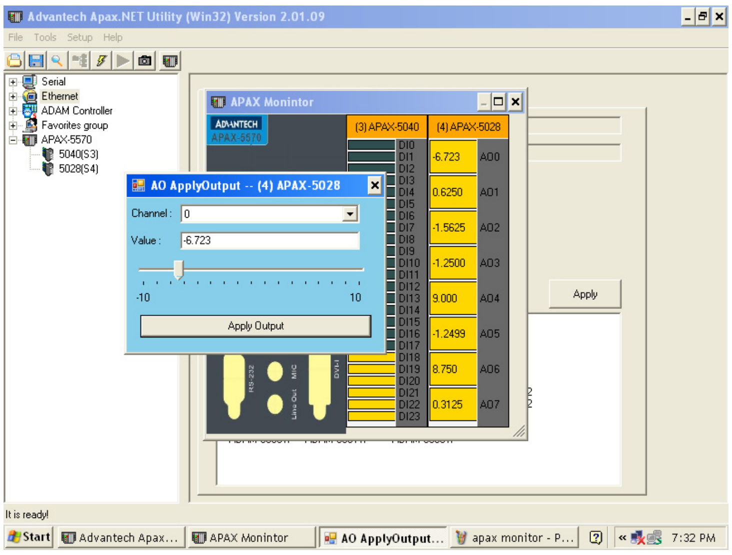


Click the **Print Screen** button, you can capture current window image of APAX.NET utility and save it to an image file. Click the **APAX Monitor** button, there will be one pop-up window showing status of all APAX-5000 I/O modules connected with APAX-557X, including ID number and channel value.





DI and AI channel values are displayed on the window. Click the DO channels to change its value. Clicking the AO channels, another pop-up window will let you configure the output value. (Refer to figure below)



Channel : 0

Value : -6.723

-6.723	AO0
0.6250	AO1
-1.5625	AO2
-1.2500	AO3
9.000	AO4
-1.2499	AO5
8.750	AO6
0.3125	AO7

B.1.3 Module Tree Display Area

APAX.NET Utility is one complete software tool that all APAX and ADAM I/O module can be configure and operated in this utility. The **Module Tree Display Area** is on the left part of the utility operation window. There are four categories in the **Module Tree Display Area**:

- **Serial**

All serial remote I/O Modules connected to APAX-557X will be listed in this category. You also can configure COM port parameter (such as baud rate, parity, stop bit) here.

- **Ethernet**

All Ethernet remote I/O Modules connected to APAX-557X will be listed in this category.

- **ADAM Controller**

All ADAM-5000 controllers connected to APAX-557X through serial interface in the same system, such as ADAM-5510 or ADAM-4501, will be listed in this category. Simply click this item all related modules will be displayed automatically.

- **Favorite Group**

You can define which devices listed in **Serial** or **Ethernet** categories above into your personal favorite group. This will make you easier to find your interested modules. Click on the **ADAM device** item under **Favorite group** item, and select **Favorite** >> **New** in Setup menu to create a new group. After you create your own group, click on your group and select **Favorite >> New** in **Setup** menu to add any remote devices into your group. You can also select **Diagnose connection** to check the communication.

- **APAX-557X**

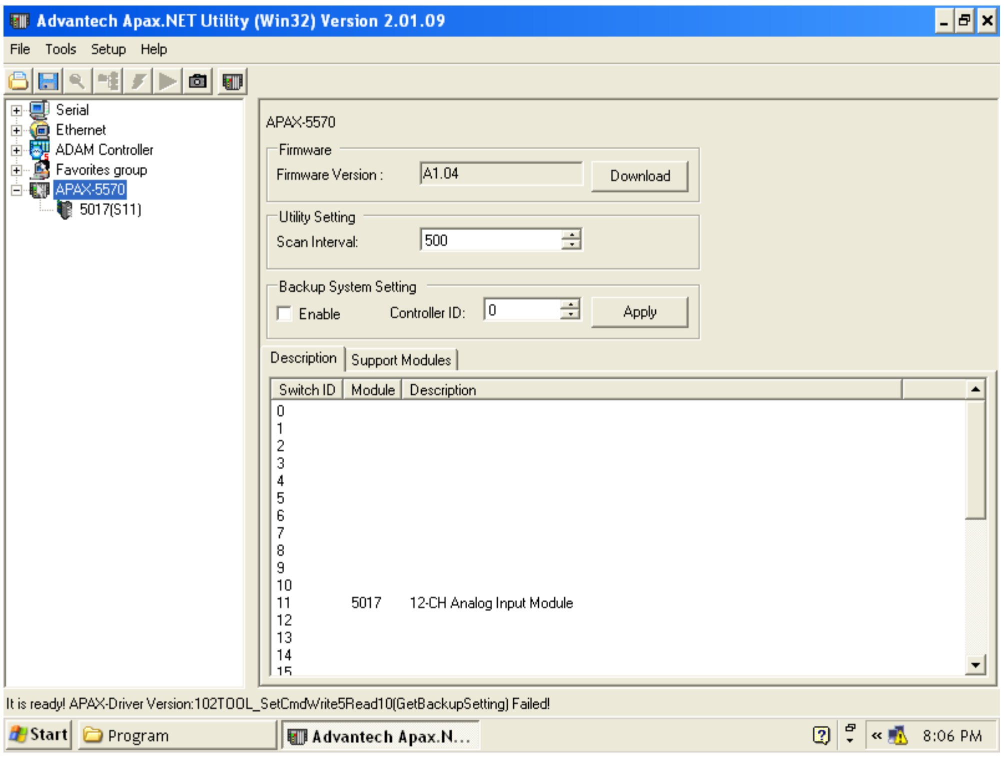
All APAX-5000 local I/O modules in the same system will be listed in this category. Simply click this item all related modules will be displayed automatically.

B.1.4 Status Display Area

Status Display Area, on the right part of utility operation window, is the main screen for operation. When you select different items in **Modules Tree Display Area**, **Status Display Area** will change dependently. You can do all configurations and tests on this area.

B.2 General Configuration

If you click the **APAX-5570** item in the **Module Tree Display Area**, the **Status Display Area** should looks as figure below:



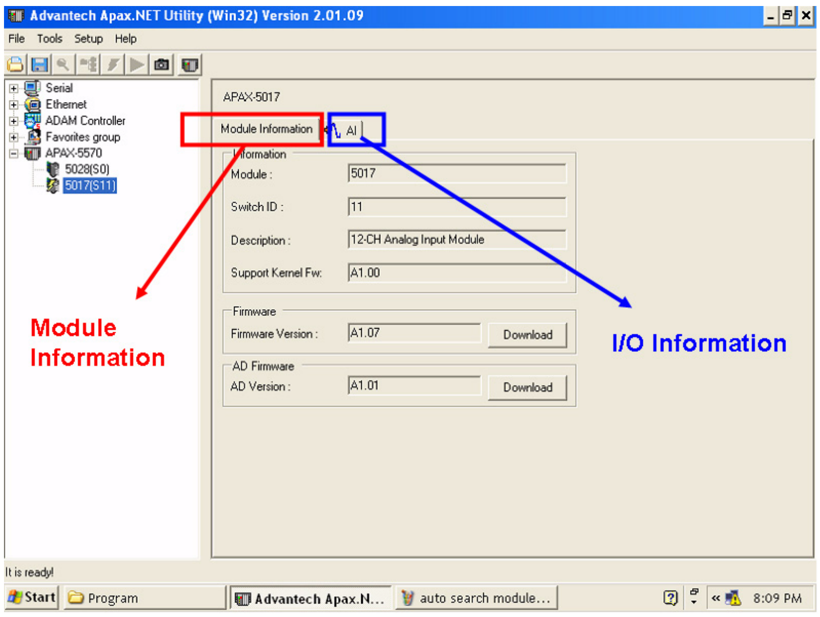
Switch ID	Module	Description
0		
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11	5017	12-CH Analog Input Module
12		
13		
14		
15		

It is ready! APAX-Driver Version:102TOOL_SetCmdWrite5Read10(GetBackupSetting) Failed!

All I/O modules with its ID number are listed in the **Description** tab in the **Module Tree Display Area** (the left tab) and **Description** tab on **Status Display Area** (the right tab). You can see all I/O modules supported by APAX-557X by the **Support Modules** tab on **Status Display Area**. The **Backup Setting** check box is used to enable or disable APAX-557X backup function. Refer to Appendix C for more detail about backup functionality.

B.3 I/O Modules Configuration

When you click any I/O module in the **Module Tree Display Area**, the **Status Display Area** at the right side will automatically change to show the module's information. There will be two tabs displayed: **Module Information** and **I/O Information**. (Refer to the figure below)

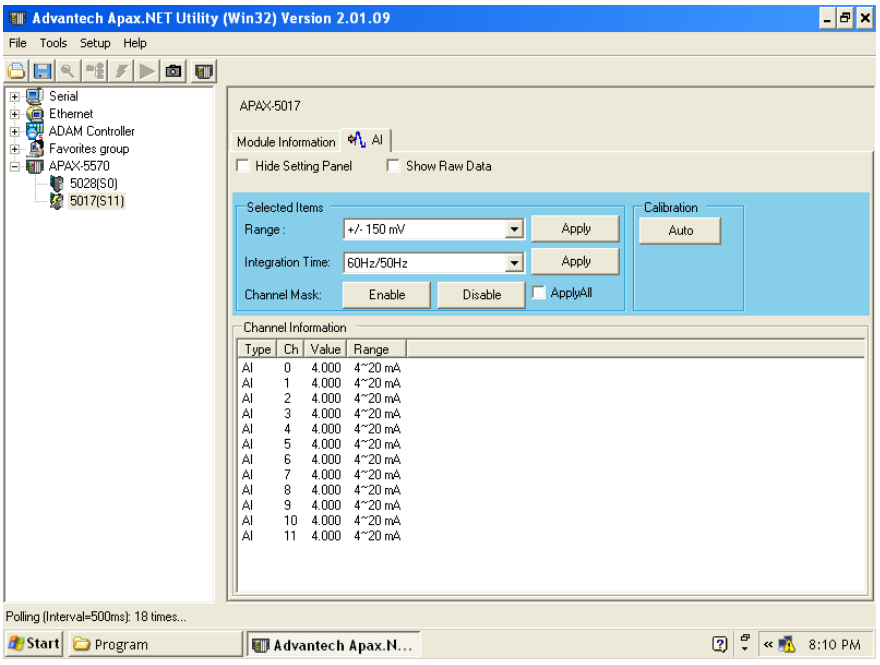


Module : 5017
Switch ID : 11
Description : 12-CH Analog Input Module
Support Kernel Fw: A1.00
Firmware Version : A1.07
AD Version : A1.01

On the **Module Information** tab, module information (such as module name, switch ID, module description, and firmware version) is displayed. You also can update related firmware to the specific module by the **Download** button.

On the **I/O Information** tab, you can write or read all channels' status and perform related configuration and calibration. Refer to sections below for more detail.

B.3.1 Analog Input Modules



Range : +/- 150 mV
Integration Time: 60Hz/50Hz
Channel Mask:

Type	Ch	Value	Range
AI	0	4.000	4~20 mA
AI	1	4.000	4~20 mA
AI	2	4.000	4~20 mA
AI	3	4.000	4~20 mA
AI	4	4.000	4~20 mA
AI	5	4.000	4~20 mA
AI	6	4.000	4~20 mA
AI	7	4.000	4~20 mA
AI	8	4.000	4~20 mA
AI	9	4.000	4~20 mA
AI	10	4.000	4~20 mA
AI	11	4.000	4~20 mA

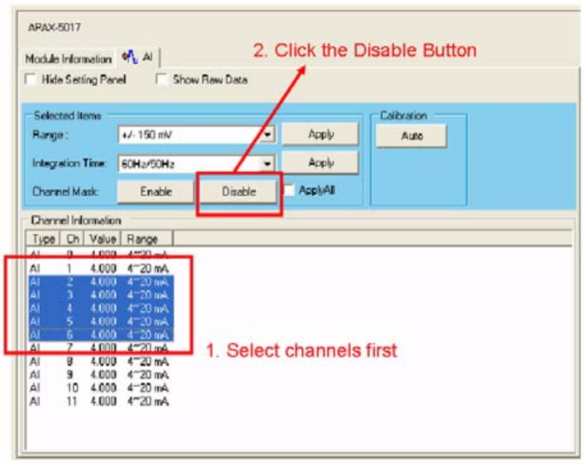
Polling (Interval=500ms): 18 times...

There are two parts for the **I/O Informaion** tab of APAX-5000 AI module. At the bottom is the **Channel Information** Area. You can see all channels' type, value, and range. Above the **Channel Information** Area is the **Setting Panel** Area. If you don't want see the **Setting Panel** Area, you can click the **Hide Setting Panel** check box to hide the **Setting Panel** Area. If you want to see the raw data (presented in Hexadecimal format) from the input channels, click the **Show Raw Data** check box.

If you want to configure specific input channels' range or integration time, select the channels in the **Channel Information** Area (use the "Shift" or "Ctrl" key on keyboard to select multiple channels at the same time). Choose appropriate range and integration time for the selected channels by the **Range** and **Integration Time** combo boxes in the **Setting Panel** Area and then click the **Apply** button to save the configuration. If you want to save the same range setting for all channels, click the **ApplyAll** check box before you click the **Apply** button.

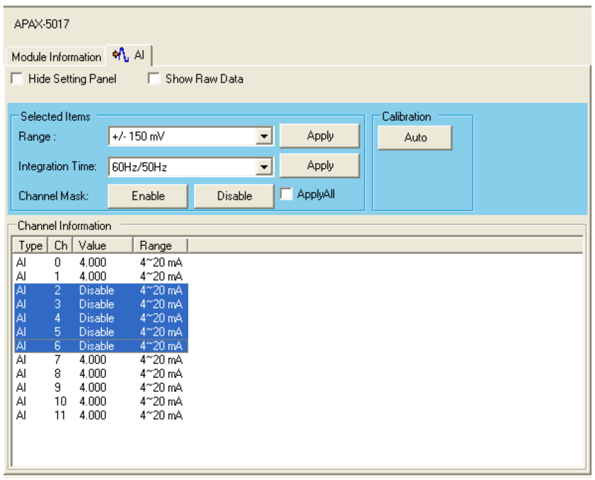
Note! *In order to remove the noise from the power supply, APAX AI modules feature built-in filter. Filters are used to remove noise generated from environment. The integration time is used to configure the filter frequency.*

You can define specific channels reading or not by the **Enable** and **Disable** button. Refer to figure below, channel 2 ~ 6 are disabled that no data will be read.



Range: +/- 150 mV
Integration Time: 60Hz/50Hz

Type	Ch	Value	Range
AI	0	4.000	4~20 mA
AI	1	4.000	4~20 mA
AI	2	4.000	4~20 mA
AI	3	4.000	4~20 mA
AI	4	4.000	4~20 mA
AI	5	4.000	4~20 mA
AI	6	4.000	4~20 mA
AI	7	4.000	4~20 mA
AI	8	4.000	4~20 mA
AI	9	4.000	4~20 mA
AI	10	4.000	4~20 mA
AI	11	4.000	4~20 mA

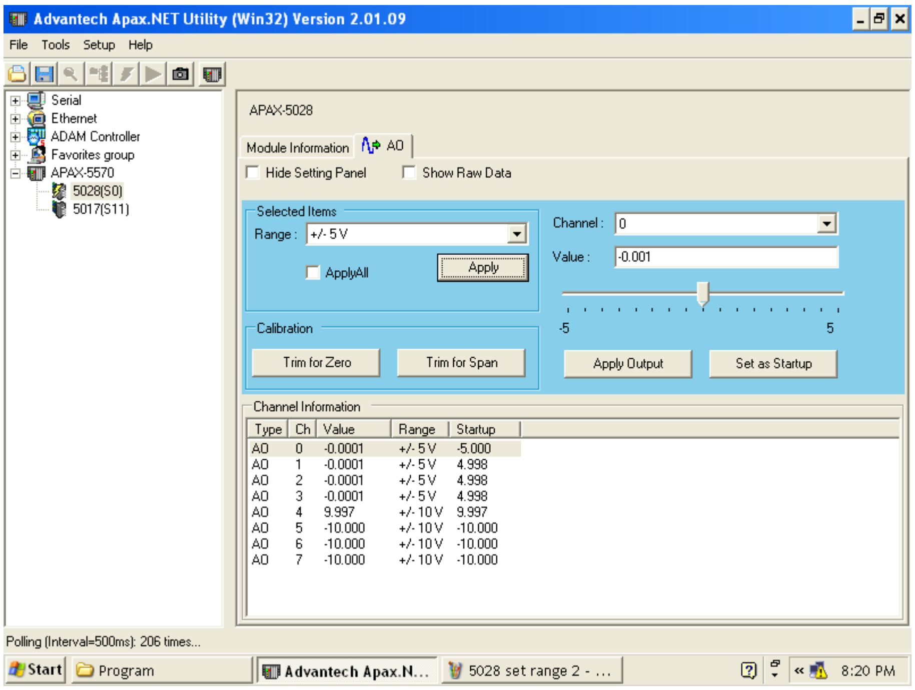


Range : +/- 150 mV
Integration Time: 60Hz/50Hz

Type	Ch	Value	Range
AI	0	4.000	4~20 mA
AI	1	4.000	4~20 mA
AI	2	Disable	4~20 mA
AI	3	Disable	4~20 mA
AI	4	Disable	4~20 mA
AI	5	Disable	4~20 mA
AI	6	Disable	4~20 mA
AI	7	4.000	4~20 mA
AI	8	4.000	4~20 mA
AI	9	4.000	4~20 mA
AI	10	4.000	4~20 mA
AI	11	4.000	4~20 mA

By clicking on the **Auto** button, you can perform auto calibration to the AI module. The module will automatically calibrate itself. You don’t need to connect any external devices or instruments.

B.3.2 Analog Output Module



Hide Setting Panel [ ] Show Raw Data [ ]

Range : +/- 5 V

ApplyAll [ ]

Channel : 0

Value : -0.001

Channel Information

Type	Ch	Value	Range	Startup
AO	0	-0.0001	+/- 5 V	-5.000
AO	1	-0.0001	+/- 5 V	4.998
AO	2	-0.0001	+/- 5 V	4.998
AO	3	-0.0001	+/- 5 V	4.998
AO	4	9.997	+/- 10 V	9.997
AO	5	-10.000	+/- 10 V	-10.000
AO	6	-10.000	+/- 10 V	-10.000
AO	7	-10.000	+/- 10 V	-10.000

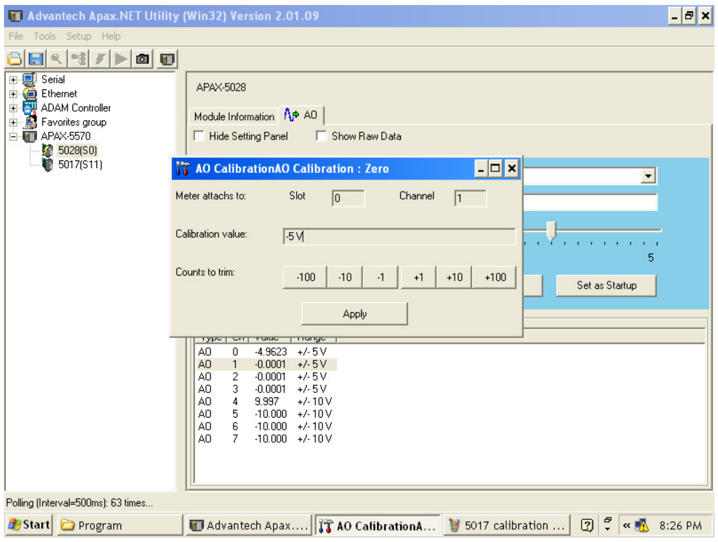
Polling (Interval=500ms): 206 times...

There are two parts for the **I/O Information** tab of APAX-5000 AO module. At the bottom is the **Channel Information** Area. You can see all channels' type, value, range and startup value (the initial value when the AO module is power-on). Above the **Channel Information** Area is the **Setting Panel** Area. If you don't want see the **Setting Panel** Area, you can click the **Hide Setting Panel** check box to hide the **Setting Panel** Area. If you want to see the raw data (presented in Hexadecimal format) from the output channels, click the **Show Raw Data** check box.

If you want to configure specific output channels' range, select the channels in the **Channel Status** Area. Choose appropriate range by the **Range** combo box in the **Setting Panel** Area and then click the **Apply** button to save the configuration. If you want to save the same range setting for all channels, click the **ApplyAll** check box before you click the **Apply** button.

If you want to change specific output channel' output value, select that channel by clicking the channel in the **Channel Information** Area or choosing it from **Channel** combo box in the **Setting Panel** Area. Then define the output value by the **Value** text box or the horizontal slide below in the **Setting Panel** Area. Then, click the **Apply** button to save the configuration. You can see the channel output value changed in the **Channel Information** Area. Similarly, you can save the value in the **Value** text box to become the startup value by the **Set as Startup** button. And you also can see the startup value changed in the **Channel Information** Area.

By clicking the **Trim to Span** button and **Trim to Zero** button, you can perform span calibration and zero calibration, separately. When you click the **Trim to Zero** button, you will see a dialog popping-up as figure below. The specific channel will generate output signal using the minimum value within range which is shown in the **Calibration Value** text box. Connect that channel to an external accurate instrument and measure the output signal. Using the **Counts to trim** buttons to adjust until the output value real matches the value in the **Calibration Value** text box. Then click the **Apply** button to save the calibration configuration.



Meter attachs to: Slot 0 Channel 1

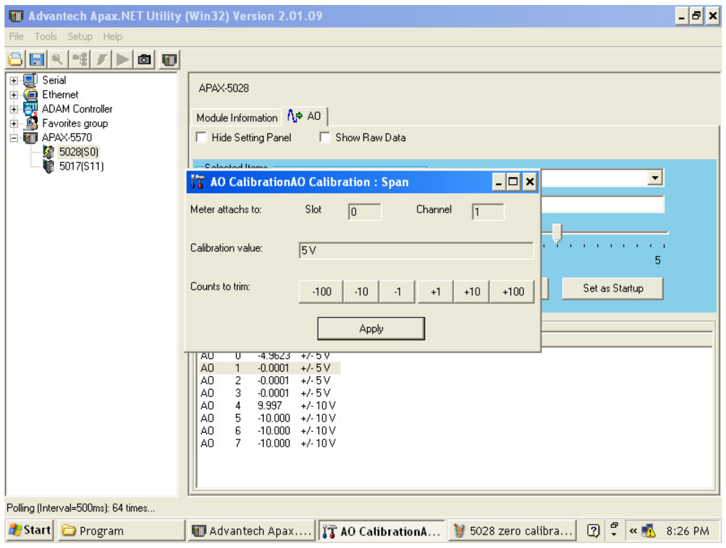
Calibration value: -5 V

Counts to trim:

```
AO  0  -4.9623  +/- 5 V
AO  1  -0.0001  +/- 5 V
AO  2  -0.0001  +/- 5 V
AO  3  -0.0001  +/- 5 V
AO  4  9.997    +/- 10 V
AO  5  -10.000  +/- 10 V
AO  6  -10.000  +/- 10 V
AO  7  -10.000  +/- 10 V
```

Polling (Interval=500ms): 63 times...

When you click the **Trim to Span** button, you will see a dialog popping-up as figure below. The specific channel will generate output signal using the maximum value within range which is shown in the **Calibration Value** text box. Connect that channel to an external accurate instrument and measure the output signal. Using the **Counts to trim** buttons to adjust until the output value real matches the value in the **Calibration Value** text box. Then click the **Apply** button to save the calibration configuration.



Meter attachs to: Slot 0 Channel 1

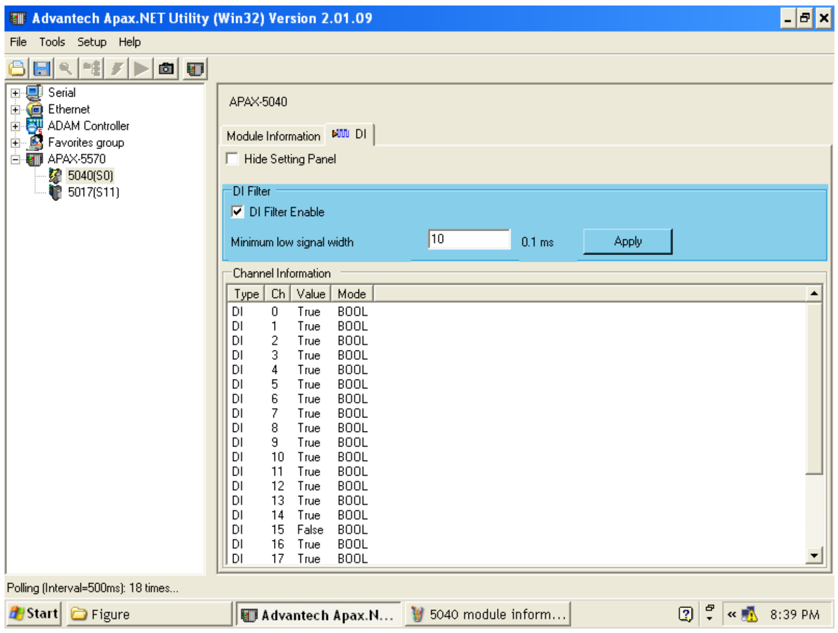
Calibration value: 5 V

Counts to trim:

```
AO  0  -4.9623  +/- 5 V
AO  1  -0.0001  +/- 5 V
AO  2  -0.0001  +/- 5 V
AO  3  -0.0001  +/- 5 V
AO  4  9.997    +/- 10 V
AO  5  -10.000  +/- 10 V
AO  6  -10.000  +/- 10 V
AO  7  -10.000  +/- 10 V
```

Polling (Interval=500ms): 64 times...

B.3.3 Digital Input Module



APAX-5040

☐ Hide Setting Panel

DI Filter

☑ DI Filter Enable

Minimum low signal width 10 0.1 ms Apply

Channel Information

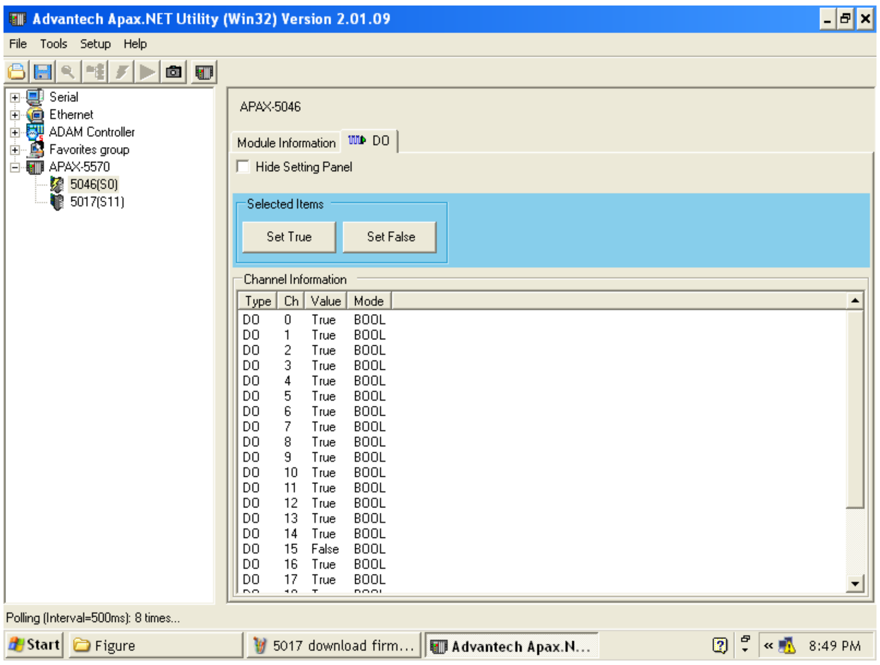
Type	Ch	Value	Mode
DI	0	True	BOOL
DI	1	True	BOOL
DI	2	True	BOOL
DI	3	True	BOOL
DI	4	True	BOOL
DI	5	True	BOOL
DI	6	True	BOOL
DI	7	True	BOOL
DI	8	True	BOOL
DI	9	True	BOOL
DI	10	True	BOOL
DI	11	True	BOOL
DI	12	True	BOOL
DI	13	True	BOOL
DI	14	True	BOOL
DI	15	False	BOOL
DI	16	True	BOOL
DI	17	True	BOOL

Polling (Interval=500ms): 18 times...

There are two parts for the **I/O Information** tab of APAX-5000 DI module. At the bottom is the **Channel Information** Area. You can see all channels' type, value, and mode. Above the **Channel Information** Area is the **Setting Panel** Area. If you don't want see the **Setting Panel** Area, you can click the **Hide Setting Panel** check box to hide the **Setting Panel** Area.

If you want to configure specific output channels' filter status or minimum acceptable pulse width, select the channels in the **Channel Information** Area. Click the **DI Filter Enable** check box in the **Setting Panel** Area to enable filter for that channel. Type the appropriate value (unit: 0.1 ms) into the **Minimum signal width** text box to configure acceptable minimum pulse width in the **Setting Panel** Area. After you complete the configuration, click the **Apply** button to save the configuration.

B.3.4 Digital Output Module



Type	Ch	Value	Mode
DO	0	True	BOOL
DO	1	True	BOOL
DO	2	True	BOOL
DO	3	True	BOOL
DO	4	True	BOOL
DO	5	True	BOOL
DO	6	True	BOOL
DO	7	True	BOOL
DO	8	True	BOOL
DO	9	True	BOOL
DO	10	True	BOOL
DO	11	True	BOOL
DO	12	True	BOOL
DO	13	True	BOOL
DO	14	True	BOOL
DO	15	False	BOOL
DO	16	True	BOOL
DO	17	True	BOOL

There are two parts for the **I/O Information** tab of APAX-5000 DO module. At the bottom is the **Channel Information** Area. You can see all channels' type, value, and mode. Above the **Channel Information** Area is the **Setting Panel** Area. If you don't want see the **Setting Panel** Area, you can click the **Hide Setting Panel** check box to hide the **Setting Panel** Area.

If you want to change specific output channels' output value, select those channels by clicking the channel in the **Channel Information** Area. Then define the output value by the **Set True** button or **Set False** button in the **Setting Panel** Area. Then, click the **Apply** button to save the configuration. You can see the channel output value changed in the **Channel Information** Area.

Appendix C

System Backup Functionality

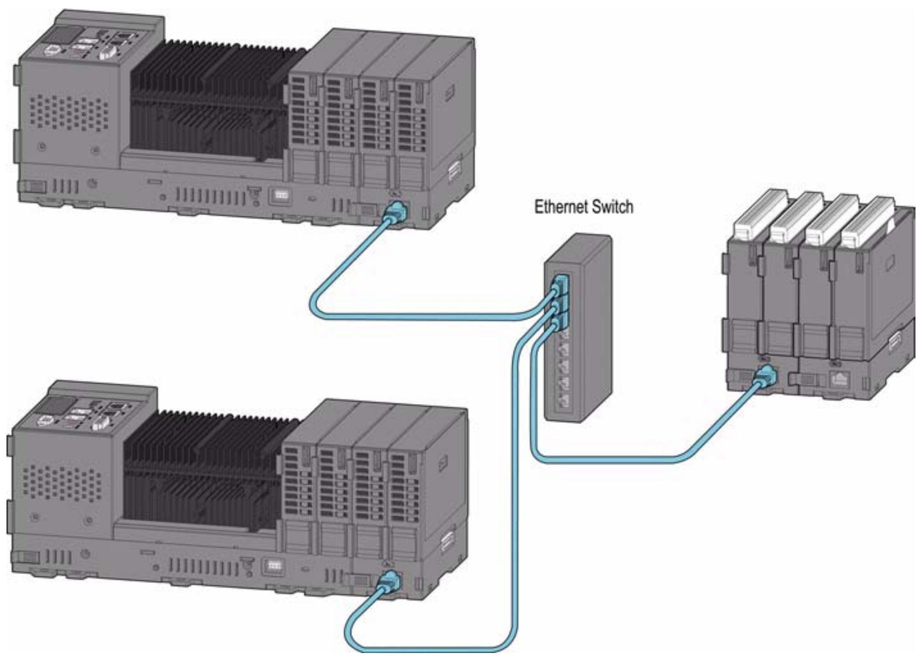
C.1 Introduction

APAX-5000 series delivers system backup functionality. To leverage this functionality, two CPU modules (controllers), with the same control program, are installed in one system. After both controllers' backup function is enabled, the APAX-5000 system will automatically delegate one of the two controllers as the master controller.

The master controller will run the control program to execute the control process, while another controller (the backup controller) is put on standby. The master controller will periodically send living message to the backup controller. If the backup controller dose not receive living message from master controller over 500 milliseconds, it will automatically become master controller and take the control responsibility and restarts the control process execution. The maximum operation time for the backup controller to become master controller (the take over time) won't be greater than 1.5 second.

Changing master controller means there is something wrong for the previous master controller. Therefore, engineers can check or change the previous master controller with a new one and enable it to have backup functionality, becoming second backup controller. Then if the new master controller fails again, the second backup controller will automatically take the control responsibility. This mechanism ensures the control system will continuously run the control process. And the system won't be stopped even if controller fails.

C.2 Configuration



For each APAX-557X module, one APAX-5002 backplane is stacked backward for the expansion. APAX-5000 I/O modules which will be controlled by the system are inserted on the backplanes. Use an unmanaged industrial Ethernet switches (such as Advantech’s EKI-2528) with 100 Mbps transmission speed and standard Ethernet cable to connect the two APAX-557X modules and APAX-5000 I/O modules. APAX-5000 series will automatically decide which one is the master controller. Be aware that two APAX-557X modules **MUST** be configured with different controller ID numbers for the system to distinguish. The next paragraph will explain how to configure ID number for APAX-557X modules.

Note! *The maximum length for the Ethernet cable between two backplanes is 100 m.*

Warning! *1. DO NOT use managed switch, hub or router between backplanes for expansion.*

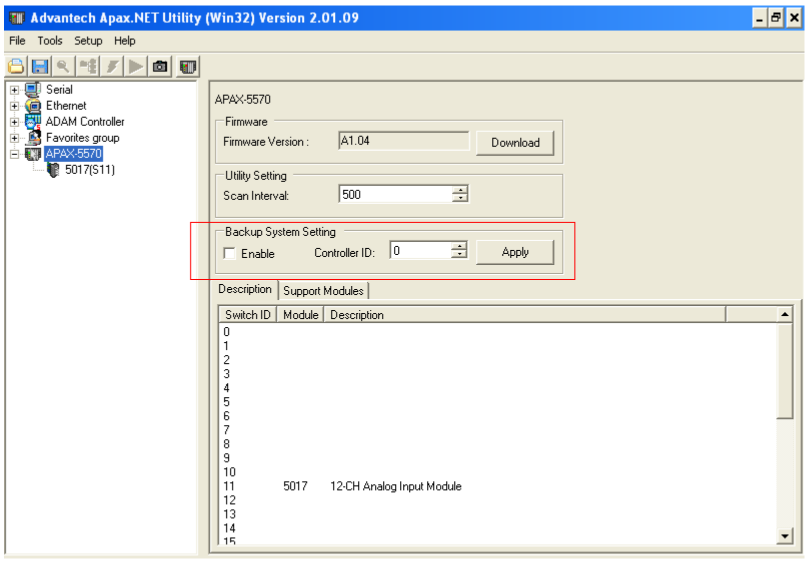
2. The network for the expansion should be a local network, NOT to connect with other external network (such as public network in enterprise network, including Internet).

3. Shielded industrial Ethernet cable MUST be used instead of standard Ethernet cable when the system is used in harsh environment, such as factory automation. Cat 6 Ethernet cable is strongly recommended for better data transmission quality.

4. It is suggested to power on all the I/O modules and APAX-557x together to avoid any unpredictable situation.

Warning! *If the APAX-557X modules' controller IDs are the same, how the system execute the backup process will be unpredictable, and unexpected execution may happen.*

Backup functionality needs to be enabled in the APAX utility for the two APAX-557X modules. Refer to figure below. In the **Backup System Setting** area, click the **Enable** check box to enable backup function. Define the controller ID for the APAX-557X modules by the **Controller ID** selector. (The ID can only be "0" or "1"). Remember to have different ID number for the two APAX-557X modules in one system. After you complete the configuration, click the **Apply** button to save the configuration.



Firmware Version : A1.04

Scan Interval: 500

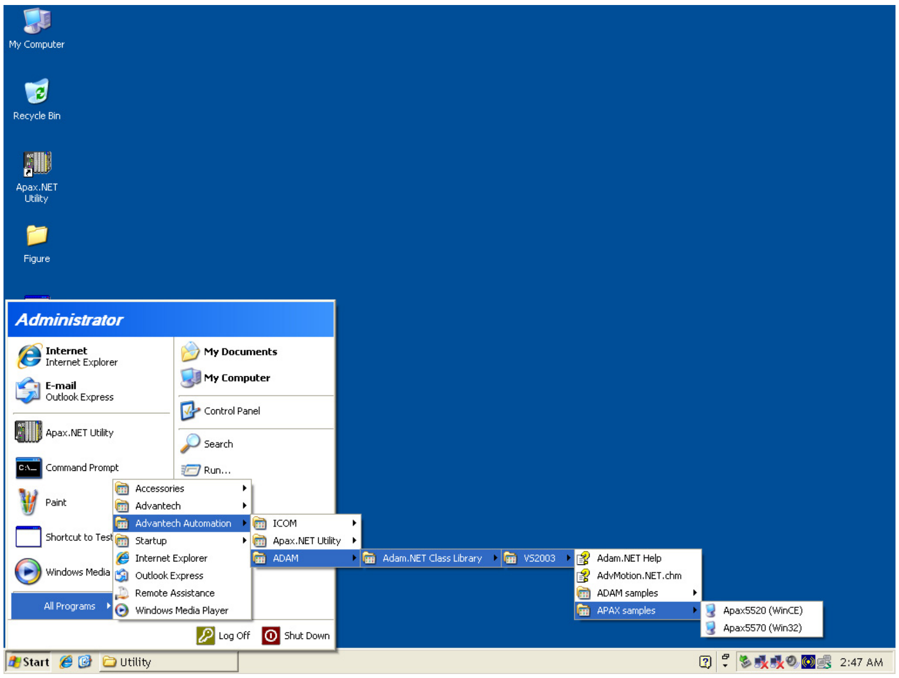
Enable ☐ Controller ID: 0

Switch ID	Module	Description
0		
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11	5017	12-CH Analog Input Module
12		
13		
14		
15		

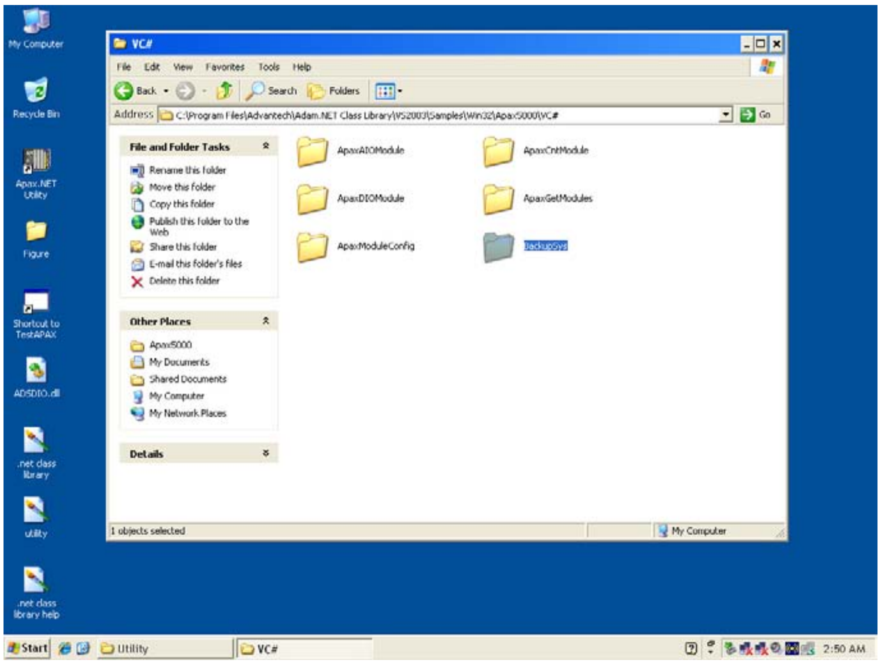
Note! *After applying the configuration for the backup system, remember to power cycle the whole system to run the backup functionality.*

C.3 Programming in Visual Studio .NET

After you enable backup functionality by utility, you can leverage the backup functionality in your application written in Microsoft Visual Studio .NET program. Related libraries are provided with Advantech Adam .NET class libraries. After you have installed the Advantech Adam .NET class libraries (Refer to Section 3.2 for the installation procedure), you can find related example programs by selecting Start >> All Programs >> Advantech Automation >> ADAM >> Adam.NET Class Library >> VS2003 >> APAX samples >> Apax5570 (Win32).



Double click the VC# folder, you can find related example code in BackupSys folder.


VC#
C:\Program Files\Advantech\Adam.NET Class Library\VS2003\Samples\Win32\Apax5000\VC#
ApaxAIOModule
ApaxCntModule
ApaxDIOModule
ApaxGetModules
ApaxModuleConfig
BackupSys

ADVANTECH

*e*Automation

www.advantech.com