WinPAC-8000 User Manual

Version 1.5, December 2008

Service and usage information for

WP-8141

WP-8441

WP-8841

Written by Kyon Huang

Edited by Anna Huang

Important Notices

Warranty

All products manufactured by ICP DAS are under warranty regarding defective materials for a period of one year, beginning from the date of delivery to the original purchaser.

Warning

ICP DAS assumes no liability for any damage resulting from the use of this product. ICP DAS reserves the right to change this manual at any time without notice. The information furnished by ICP DAS is believed to be accurate and reliable. However, no responsibility is assumed by ICP DAS for its use, not for any infringements of patents or other rights of third parties resulting from its use.

Copyright

Trademark

The names used for identification only may be registered trademarks of their respective companies.

Table of Contents

CHAPTER

1

Introduction

WinPAC-8000 is the second generation PAC of ICPDAS. It equips a PXA270 CPU (520MHz) running a Windows CE.NET 5.0 operating system, variant connectivities (VGA, USB, Ethernet, RS-232/485) and 1/4/8 slots for high performance parallel I/O modules (high profile I-8K series) and serial-type I/O modules (high profile I-87K I/O modules).

Its operating system,Windows CE 5.0, has many advantages, including hard real-time capability, small core size, fast boot speed, interrupt handling at a deeper level, achievable deterministic control and low cost. Using Windows CE.Net 5.0 in the WinPAC-8000 gives it the ability to run PC-based Control software such as Visual Basic.NET, Visual C#, Embedded Visual C++, SCADA software, Soft PLC …etc.

Comparing with the first generation WinCon-8000, it not only improves the CPU performance (from 206 MHz to 520 MHz) and upgrading OS (from CE 4.1 to CE 5.0), but also adds many reliability features, such as dual LAN, redundant power input, dual battery backup SRAM, etc. It gives you all of the best features of both traditional PLCs and Windows capable PCs.

1.1. Features

Software Features

Windows CE .Net 5.0

Most of the popular features in MS software are included, such as

1. FTP Server
2. HTTP Server
3. ASP (Java script, VB script)
4. SQL Server Compact Edition 3.5
5. Compact .Net Framework 2.0

Remote Maintenance via FTP Server and VCEP Software

The FTP server is used for uploading applications or downloading data. And VCEP is designed for managing the WinPAC-8000. VCEP can synchronize every movements (screen, keyboard and mouse actions) between the PC and the WinPAC-8000 via the Ethernet. By using the FTP server and VCEP, you can update and manage the WinPAC-8000 remotely via the Ethernet.

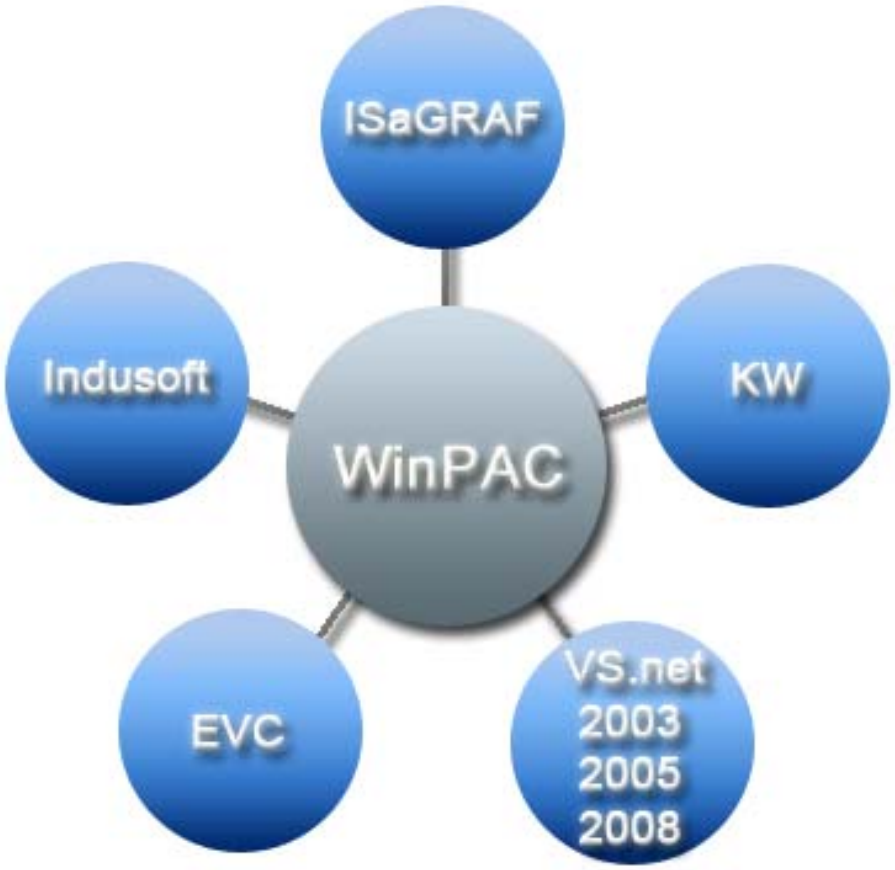
Built-In OPC Server (Quicker)

Quicker is an OPC server, and SCADA software can easily integrate I/O modules through it. Furthermore, it also provides a library which users can use to develop their AP by eVC, C# or VB.Net.

Quicker not only supports I/O modules in local slots, but also supports remote I/O modules with the following protocols via the RS-232/485 or Ethernet:

1. Modbus/RTU
2. Modbus/ASCII
3. Modbus/TCP
4. DCON

Rich Software Solutions



On the WinPAC-8000, ICP DAS provides the following software solutions to fit in different applications.

1. **Visual Studio .Net 2003/2005/2008 and eVC solution:**
   SDK as well as demo programs for C#, VB.Net and eVC are provided.
2. **SoftPLC solution:**
   A. **ISaGRAF** supports IEC61131-3 languages, Ladder Diagram (LD), Structured Text (ST), Function Block Diagram (FBD), Sequential Function Chart (SFC), Instruction List (IL), and Flow Chart (FC).
   B. **KW-software** supports IEC61131-3 languages and HMI features.
3. **SCADA solution: Indusoft** provides simple “drag and drop”, “point and click” developing environment for HMI and SCADA applications.

Upgrading applications from WinCon to WinPAC just copy and play

Users can upgrade their controller from WinCon-8000 to WinPAC-8000, and meanwhile still keep most of their old applications (.exe file) running smoothly on WinPAC-8000 without any modification. The only thing that users have to do is just to copy the .exe file from WinCon-8000, paste it to WinPAC-8000, and then play it. How simple it is. And for exceptions that old applications (.exe file) are not able to follow the copy-and-paste rule, users can still finish the upgrading by recompiling the old applications.

Hardware Features

Power CPU module

The most important features of the CPU module are

1. PXA270 or compatible CPU (32-bit and 520MHz)
2. 128MB SDRAM
3. 15MB Built-In Flash Disk

Built-in VGA Port

A built-in VGA port can be directly connected to a regular LCD display. Users can operate the HMI or SCADA software (running on the WinPAC-8000) with display, keyboard and mouse just as how they usually did on regular PCs.

64-bit Hardware Serial Number

The 64-bit hardware serial number is unique and individual. Every serial number of WinPAC-8000 is different. Users can add a checking mechanism to their AP to prevent software from pirating.

I/O Module Hot Swap Ability (Will be available)

(For High Profile I-87K Modules Only)

The WinPAC-8000 features hot swap which means that there is no need to power off the WinPAC-8000 for replacing high profile I-87K modules. The OS provides a function sending plug-in and removing messages to user's applications. Using this feature, users can design its own plug-and-play applications.

Rich I/O Expansion Ability (RS-232/485, Ethernet, FRnet, CAN)

Beside the local I/O slots, WinPAC-8000 also equips several RS-232/485 ports, two Ethernet ports to connect serial I/O and Ethernet I/O. And with FRnet and CAN communication module in local slot, FRnet I/O and CAN devices are easy to be integrated.

Built-In Flash Disk (15MB)

In normal situation, users can store their AP or data to the Micro SD card or USB Flash disk. But in some vibrational environment (for example, like driving ships), the two storage media would be bad connection. Then the built-in Flash disk will be the best storage media in such the vibrational environment.

Dual Watchdog Timer

A system could be hanged up when the OS or the AP fails. There are two watchdogs (OS watchdog and AP watchdog) designed to automatically reset the CPU when the situations happen. The design will increase the reliability of the system.

Dual Battery-Backup SRAM (512KB)

To maintain important data while power off, non-volatile memory is the ideal design. The WinPAC-8000 equips a 512KB SRAM with two Li-batteries to maintain data while power off. The two Li-batteries can continually supply power to the 512KB SRAM to retain the data for 10 years; and the dual-battery design can avoid data lost while replacing a new battery.

Dual Ethernet Ports

WinPAC-8000 provides two Ethernet ports. The two Ethernet ports can be used to implement redundant Ethernet communication and separate Ethernet communication (one for global Internet, one for private Ethernet).

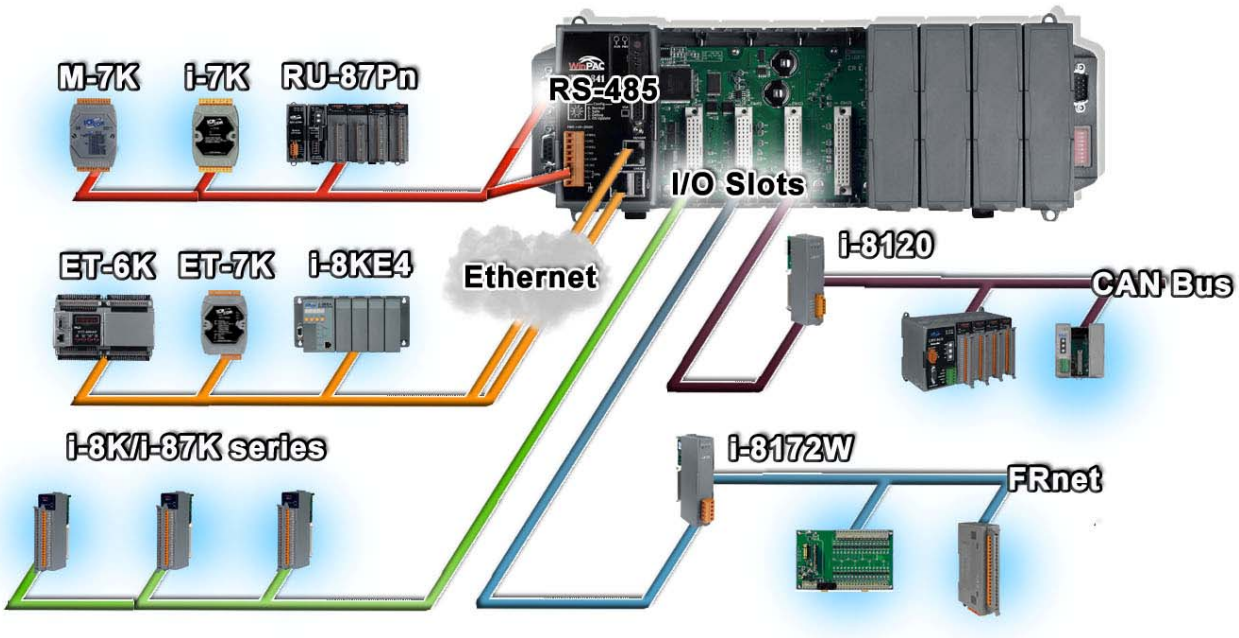
Redundant Power Input

To prevent the WinPAC-8000 from failing by the power loss, the power module is designed with two input connectors. Once a power input fails, the power module switches to the other power input. And there is a relay output for informing the power failure.

Ventilated Housing Design Allows Operation Between -25℃ ~ +75℃

Each WinPAC-8000 is housed in a plastic-based box with a column-like ventilator that can help to cool the working environment inside the box and allow the WinPAC-8000 operating between -25℃ and +75℃.

I/O expansion



WinPAC-8000 provides the following I/O expansion bus

Local I/O Slot:

There are 1/4/8 slot options to expand local I/O. And the I/O modules can be parallel bus type **(high profile I-8K series)** and serial bus type **(high profile I-87K series)**.

The difference between them is

Item	I-8K Series	I-87K Series
Microprocessor	No	Yes (8051)
Communication interface	Parallel bus	Serial bus
Communication speed	Fast	Slow
DI latched function	No	Yes
Counter input (for digital input module)	No	Yes (100 Hz)
Power on value	No	Yes
Safe value	No	Yes
Programmable slew-rate for AO module	No	Yes

Ethernet:

The available Ethernet I/O devices are ET-6000, ET-7000, I-8KE4/8 and I-8KE4/8-MTCP. WinPAC-8000 can access them with Modbus/TCP or DCON protocol.

RS-485:

WinPAC-8000 is equipped with two RS485 COM ports which enable the transmission rate up to a maximum of 115.2 Kbps. The medium for connection is a twisted-pair, multi-drop, 2-wire RS-485 network that can link i-7000, M-7000, RU-87Pn and high profile I-87K modules.

FRnet:

FRnet is an innovative industrial field bus that has many special features, such as high-speed deterministic I/O control, real I/O synchronization capabilities, non-protocol communication, and easy programming. Pluging in a FRnet communication module (I-8172W), the WinPAC-8000 can link FRnet I/O modules to implement high-speed distributed I/O.

CAN Bus:

The Controller Area Network (CAN) is a serial communication way, which efficiently supports distributed real-time control with a very high level of security. It provides the error-processing mechanisms and concepts of message priority. These features can improve the network reliability and transmission efficiency. Furthermore, CAN supplies the multi-master capabilities, and is especially suited for networking “intelligent” devices as well as sensors and actuators within a system or a sub-system. With I-8120, I-8123, I-8124, I-87120, I-87123, and I-87124, WinPAC-8000 is able to demonstrate every feature of CAN, CANopen, and DeviceNet.

1.2. Specifications

This section provides specifications and supplemental information for the WinPAC-8000 controller.

System Software	
OS	WinCE.Net 5.0
.Net Compact Framework	2.0
Embedded Service	FTP server, Web server (supports VB script, JAVA script), Embedded SQL server
SDK Provided	Dll for eVC, Dll for Visual Studio.Net 2003/2005/2008

CPU Module	
CPU	PXA 270 or compatible (32-bit and 520MHz)
SDRAM	128MB
Dual Battery Backup SRAM	512KB (for 5 years data retain)
Flash	48MB (32MB for OS image, 15MB for built-in Flash disk, 1MB for registry) 100,000 erase/write cycles
Expansion Flash Memory	Micro SD socket with one 1GB Micro SD card
RTC (Real Time Clock)	Year-2000 compliance; seconds, minutes, hours, date of the month; month, year, valid up from 1980 to 2079
EEPROM	16KB Data retention: 40 years; 1,000,000 erase/write cycles.
64-bit Hardware Serial Number	Yes
Dual Watchdog Timer	Yes

CPU Module	
Programmable LED Indicator	1
Rotary Switch	Yes (0~9)
DIP Switch	Yes (8 bits); for (WP-8441 and WP-8841 only)

Power	
Input Range	+10 ~ +30 V_{DC}
Isolation	1KV
Redundant Power Inputs	Yes (with one Relay output for alarm warning)
Capacity	WP-8141: 1.0A, 5V supply to CPU and backplane, 0.6A, 5V supply to I/O expansion slots, total 30W
	WP-8441: 1.1A, 5V supply to CPU and backplane, 4.9A, 5V supply to I/O expansion slots, total 30W
	WP-8841: 1.2A, 5V supply to CPU and backplane, 4.8A, 5V supply to I/O expansion slots, total 30W
Consumption	WP-8141: 7.3W (0.3A@24V)
	WP-8441: 9.1W (0.38A@24V)
	WP-8841: 9.6W (0.4A@24V)

Operating Environment	
Operating Temperature	–25°C ~ +75°C
Storage Temperature	–30°C ~ +85°C
Humidity	5 ~ 90% RH, Non-condensing

I/O Expansion Slots (Supports high profile I-8K and I-87K modules only)	
WP-8141	1 Slot
WP-8441	4 Slots
WP-8841	8 Slots

Connectivity	
VGA	1 (800x600 resolution)
Ethernet Port	2, 10/100 Base-TX Ethernet Controller (Auto-negotiating, Auto_MDIX, LED indicator)
USB 1.1 (host)	1

COM0 (Internal RS-485 interface communication with I-87K I/O modules in slots)

Baud Rate – 115200 bps

Data Bits – 8

Parity – None, Even, Odd

Stop Bits – 1

FIFO – 1 byte

Note: CPU internal uart

COM1 (RS-232 used to update firmware)

Baud Rate – 115200, 57600, 38400, 19200, 9600, 4800, 2400, 1200 bps

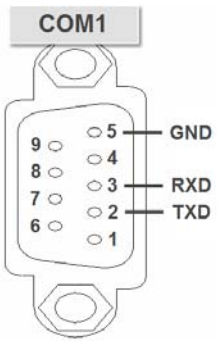
Data Bits – 7, 8

Parity – None, Even, Odd

Stop Bits – 1

FIFO – 1 byte

Note: CPU internal uart



Connectivity

COM2 (RS-485)

Baud Rate – 115200, 57600, 38400, 19200, 9600, 4800, 2400, 1200 bps

Data Bits – 5, 6, 7, 8

Parity – None, Even, Odd, Mark (Always 1), Space (Always 0)

Stop Bits – 1, 2

FIFO – 16 bytes

Note: 16C550 compatible

COM3 (RS-232/RS-485) (For WP-8441 and WP-8841 only)

Baud Rate – 115200, 57600, 38400, 19200, 9600, 4800, 2400, 1200 bps

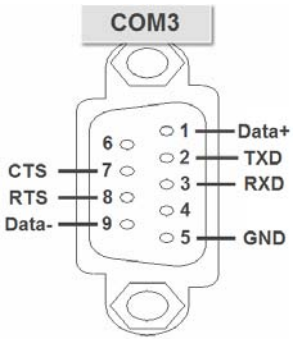
Data Bits – 5, 6, 7, 8

Parity – None, Even, Odd, Mark (Always 1), Space (Always 0)

Stop Bits – 1, 2

FIFO – 16 bytes

Note: 16C550 compatible



COM3 can be configured as either RS-232 or RS-485, and the configuration depends on the pin connections as follows:

- RS-232 (RXD, TXD, CTS, RTS and GND)
- RS-485 (Data+ and Data-)

There is no software configuration or hardware jumper needed.

Connectivity

COM4 (RS-232) (For WP-8441 and WP-8841 only)

Baud Rate – 115200, 57600, 38400, 19200, 9600, 4800, 2400, 1200 bps

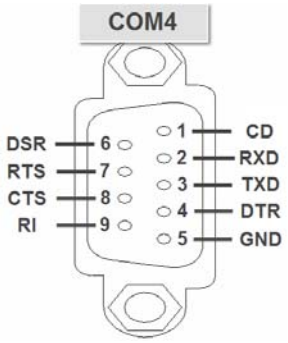
Data Bits – 5, 6, 7, 8

Parity – None, Even, Odd, Mark (Always 1), Space (Always 0)

Stop Bits – 1, 2

FIFO – 16 bytes

Note: 16C550 compatible



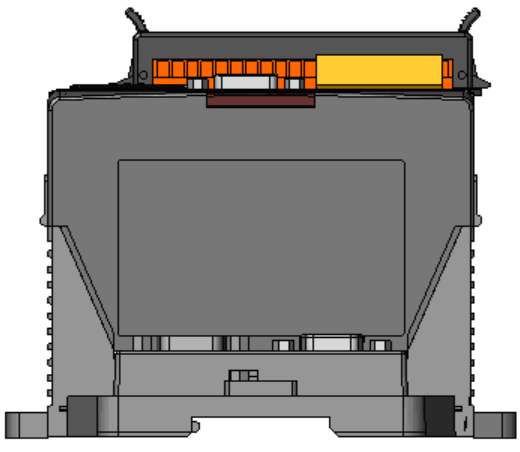
Dimensions

WP-8141	95 x 132 x 110 mm
WP-8441	230 x 132 x 110 mm
WP-8841	354 x 132 x 110 mm

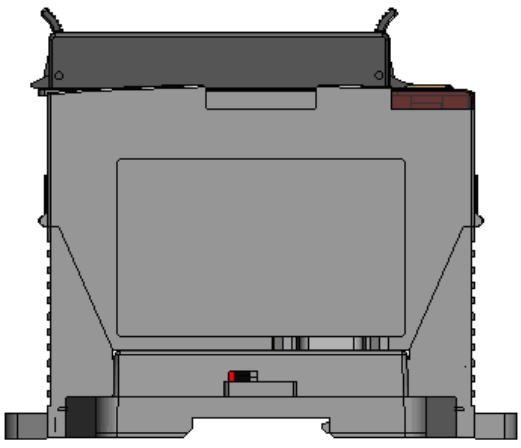
1.3. Dimensions

This section describes the dimension of the WinPAC-8000 controller.

WP-8141:

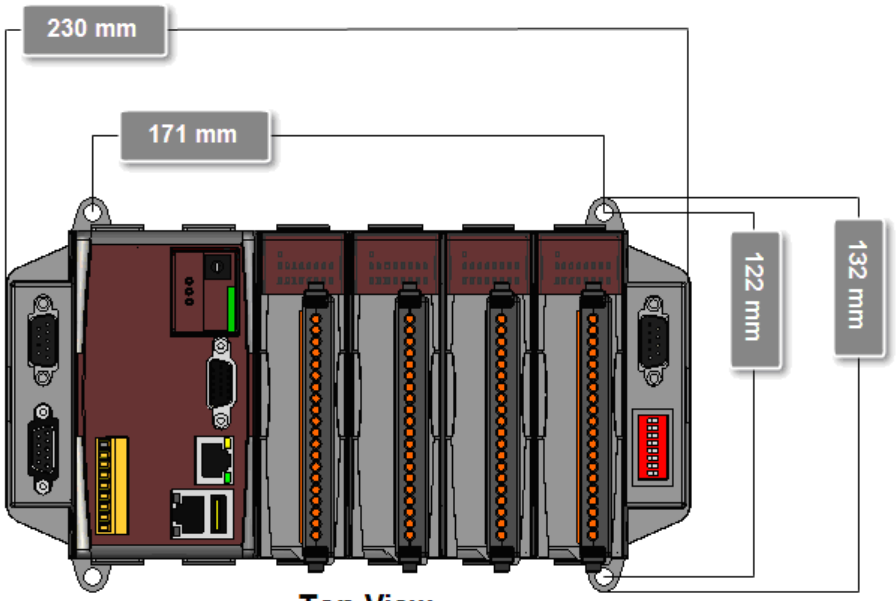


Left Side View

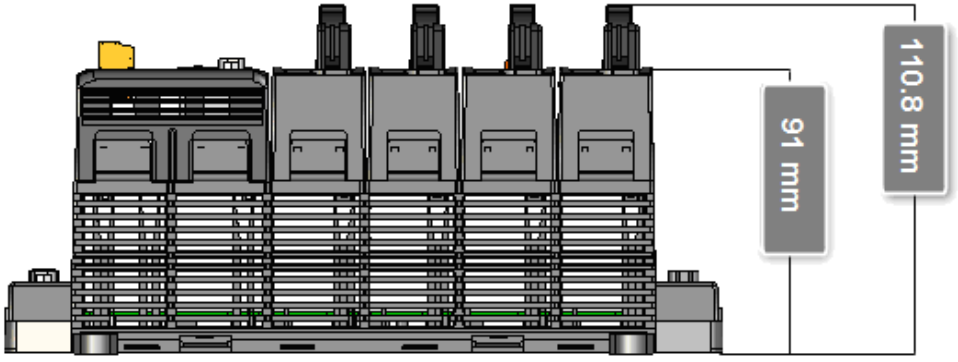


Right Side View

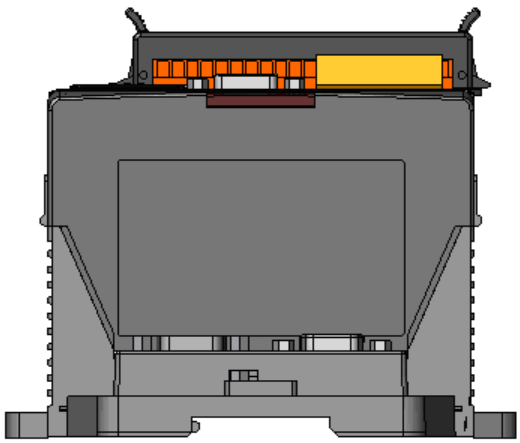
WP-8441:



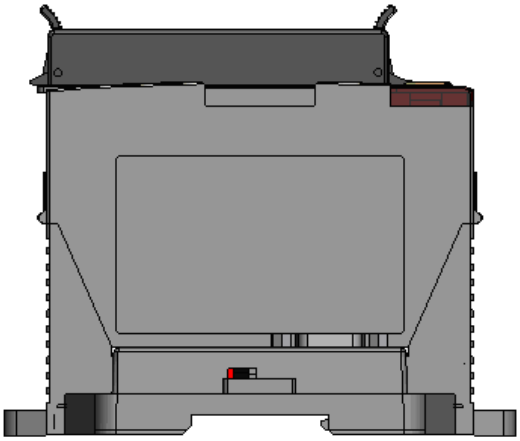
Top View



Front View

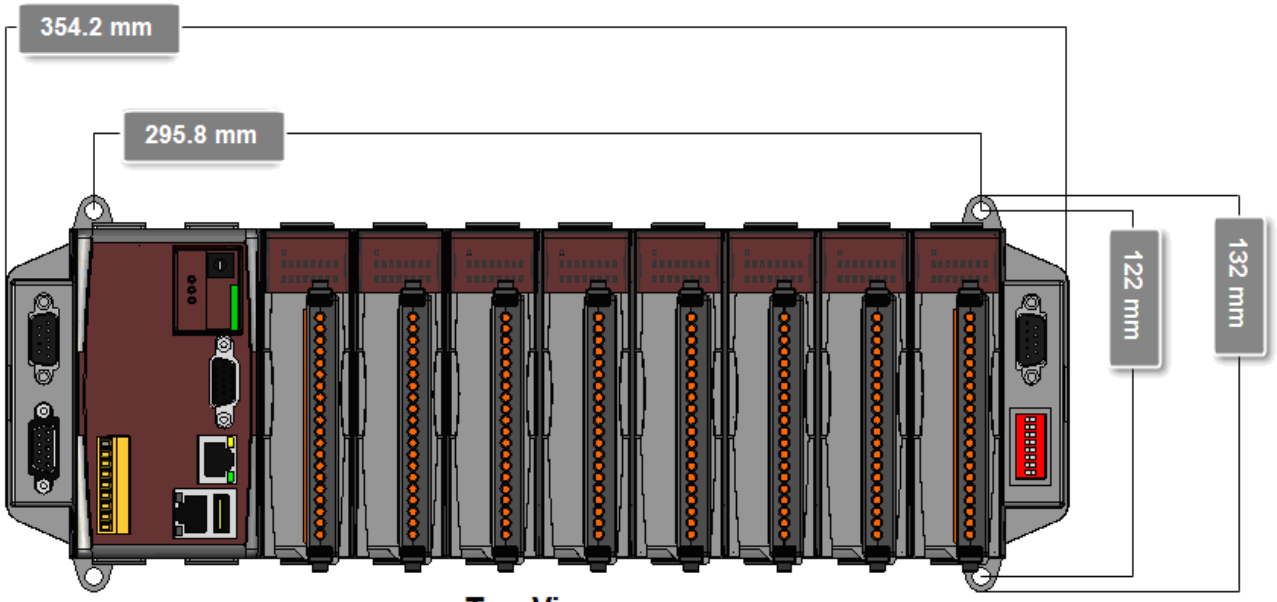


Left Side View

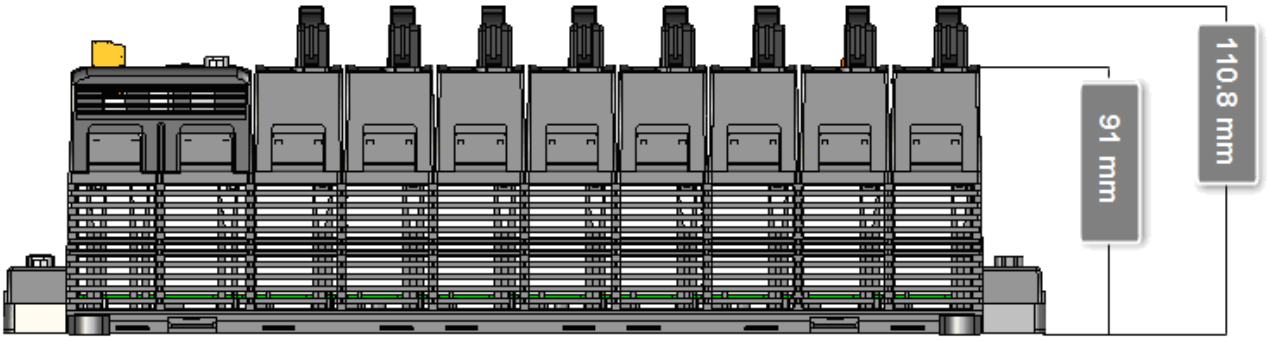


Right Side View

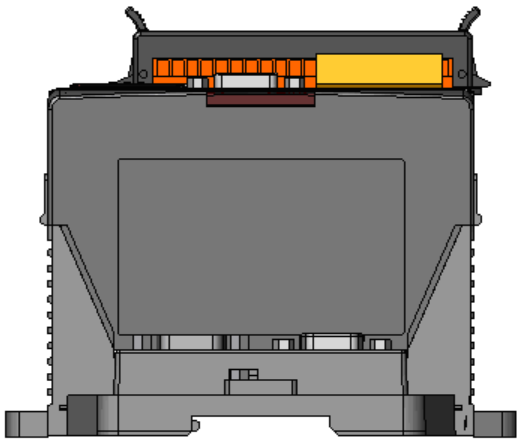
WP-8811:



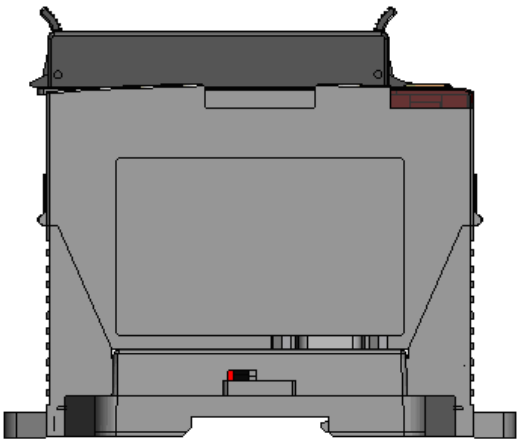
Top View



Front View



Left Side View



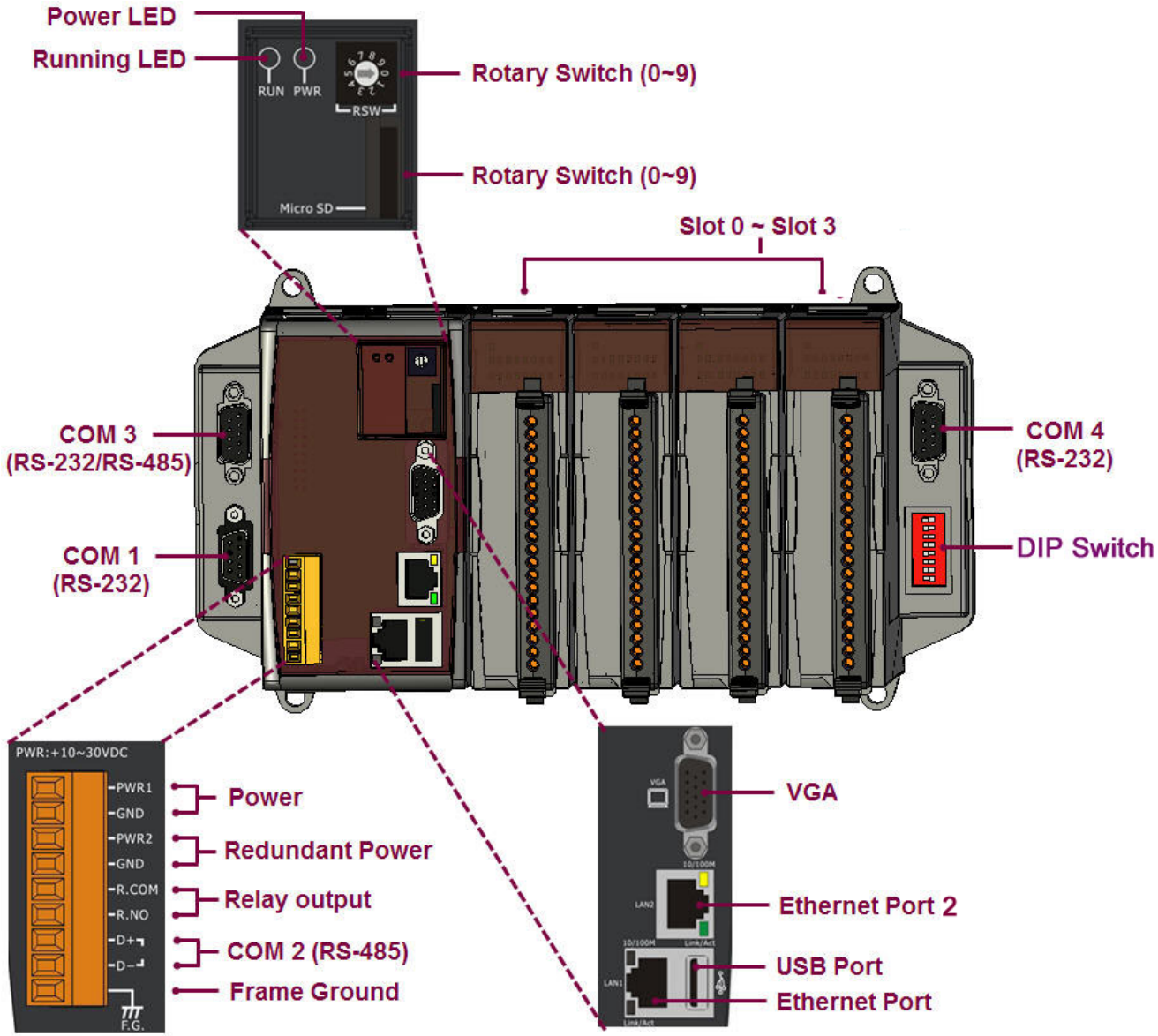
Right Side View

1.4. Overview

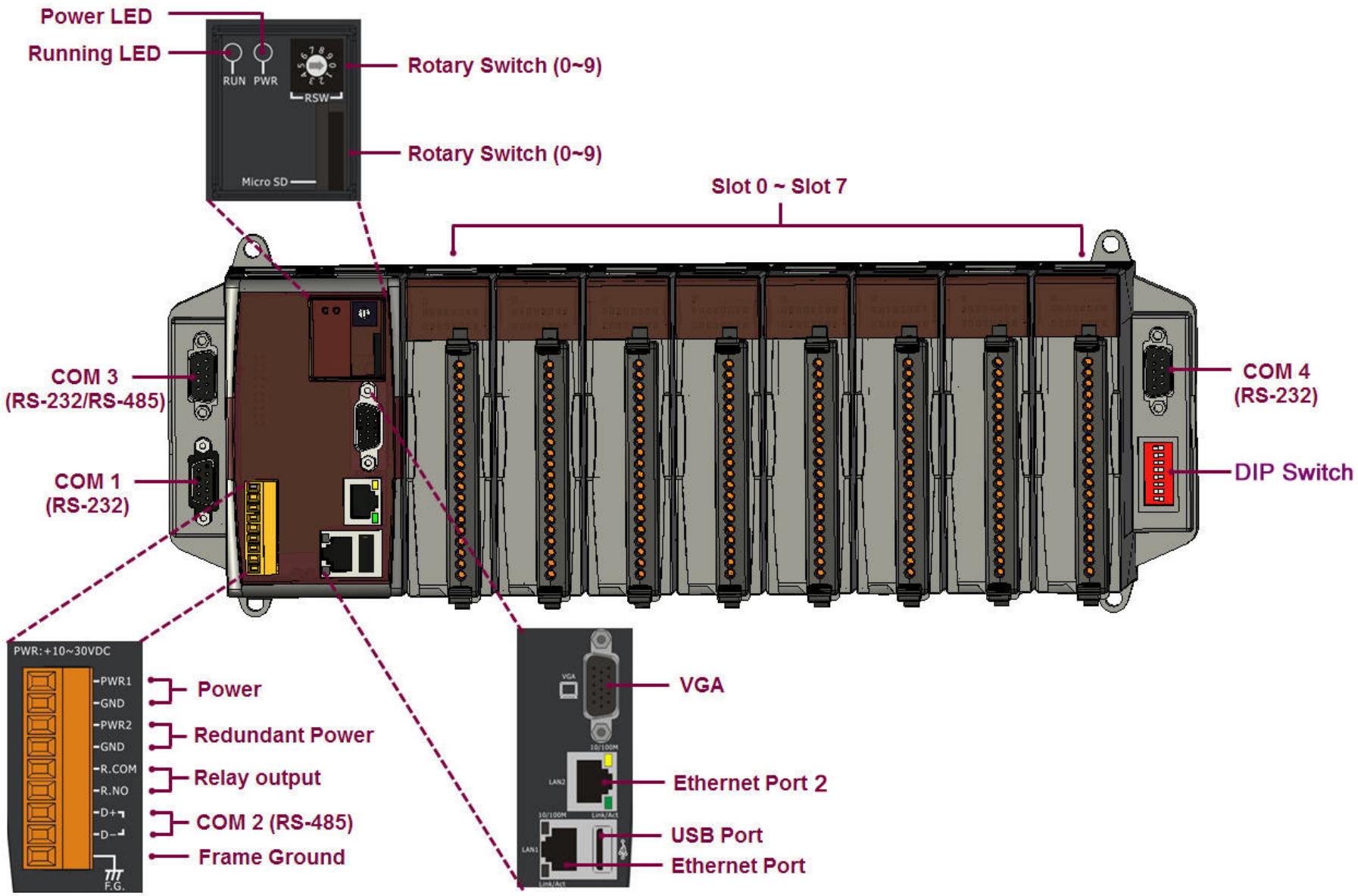
This section describes the names of parts of the WinPAC-8000 controller.

WP-8141:

WP-8441:

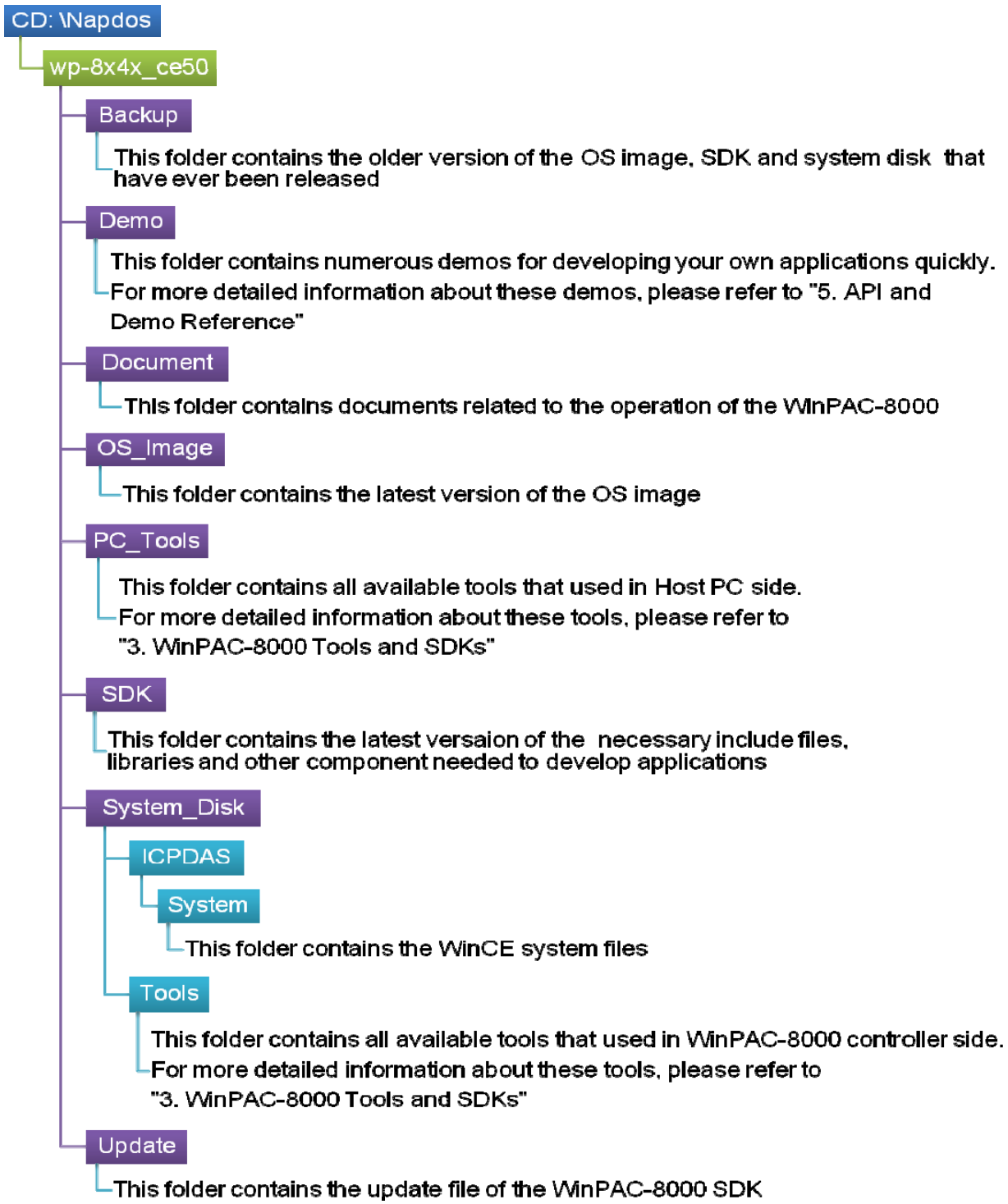

Power LED
Running LED
RUN PWR
RSW
Micro SD
Rotary Switch (0~9)
Rotary Switch (0~9)
Slot 0 ~ Slot 3
COM 3
(RS-232/RS-485)
COM 1
(RS-232)
COM 4
(RS-232)
DIP Switch
PWR: +10~30VDC
PWR1
GND
PWR2
GND
R.COM
R.NO
D+
D-
F.G.
Power
Redundant Power
Relay output
COM 2 (RS-485)
Frame Ground
VGA
VGA
Ethernet Port 2
LAN2
USB Port
LAN1
Ethernet Port

WP-8841:


Power LED
Running LED
RUN
PWR
RSW
Micro SD
Rotary Switch (0~9)
Rotary Switch (0~9)
Slot 0 ~ Slot 7
COM 3
(RS-232/RS-485)
COM 1
(RS-232)
COM 4
(RS-232)
DIP Switch
PWR: +10~30VDC
PWR1
GND
PWR2
GND
R.COM
R.NO
D+
D-
F.G.
Power
Redundant Power
Relay output
COM 2 (RS-485)
Frame Ground
VGA
Ethernet Port 2
USB Port
Ethernet Port

1.5. Companion CD

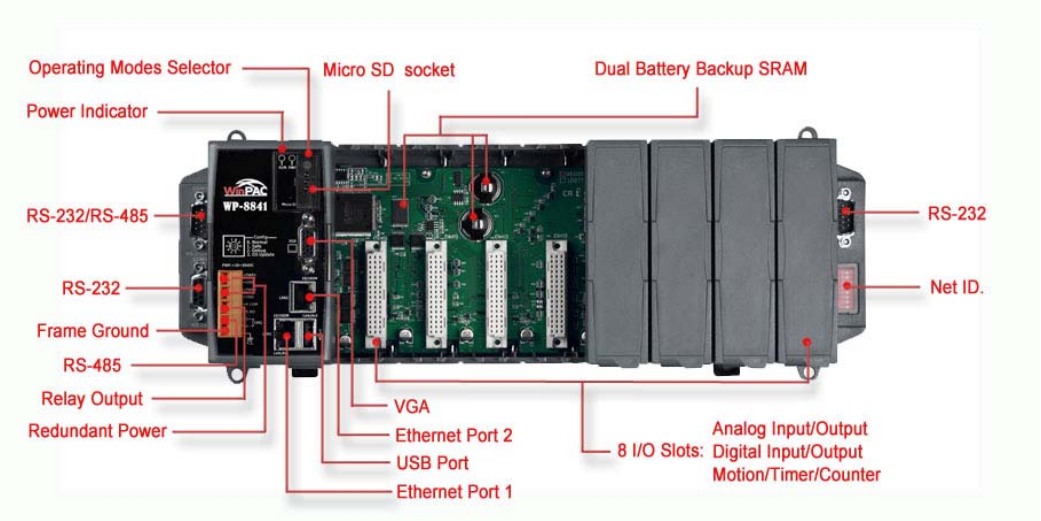
This section describes the content of the companion CD, which provides the software and documentation related to the WinPAC-8000 controller. The directory tree below will help you to quickly search the contents of the CD.



1.6. Comparing with WinCon 8x3x and 8x4x

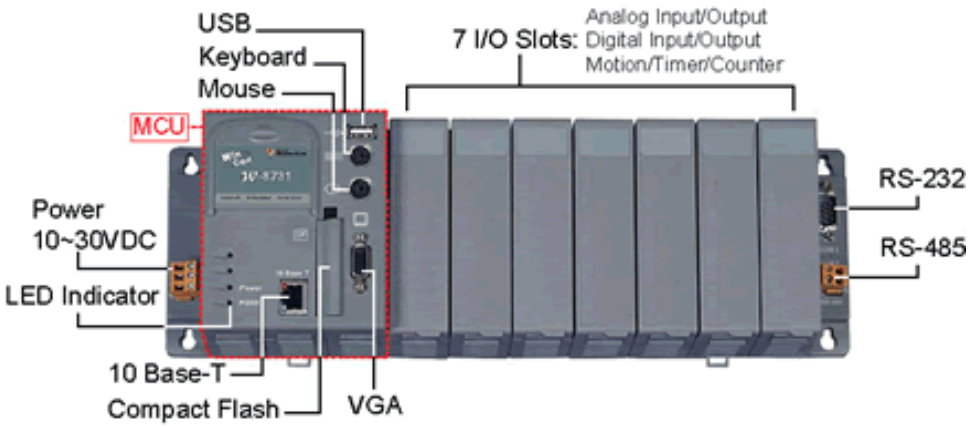
WP-8x4x:

WP-8x4x is the second generation of PAC. It is equipped with variant connectivities (VGA, USB, Ethernet, RS-232/485) and 1/4/8 slots for high performance parallel I/O modules (high profile I-8K series) and serial-type I/O modules (high profile I-87K I/O modules). Comparing to the first generation WinCon PAC, it not only improves the CPU performance (from 206 MHz to 520 MHz), but also add many reliablity features.



W-8x3x: (Phased Out Since 2008)

W-8x3x is a PAC equipped with PC interface and 3/7 slots for I/O modules. Its PC-like interface includes: PS/2(keyboard and mouse), VGA port , USB port and RS-232/RS-485 communication ports. The powerful backplane(3/7 slots) can adopts versatile ICPDAS's high performance parallel (I-8K I/O modules) and serial-type I/O modules(I-87K I/O modules)



W-8x4x:

WinCon-8x4x series embedded controllers are derived from existing WinCon-8x3x series. This evolution includes dual USB host ports to support more connectivity to USB I/O devices, dual Ethernet 10/100M ports to provide higher network performance, network security, and redundancy capability, and a rotate switch to configure your system and applications.

	WP-8x4x	W-8x3x	W-8x4x
OS	Windows CE 5.0	Windows CE 4.1	Windows CE 4.1
CPU	520MHz	206MHz	206MHz
SDRAM	128MB	64MB	64MB
Dual Battery Backup SRAM	512KB	-	-
Flash	48MB (32MB for OS image, 15MB for built-in Flash disk, 1MB for registry)	32MB (for OS image only)	32MB (for OS image only)
EEPROM	16KB	16KB	16KB

	WP-8x4x	W-8x3x	W-8x4x
Expansion Memory Slot	Micro SD slot (Default=1GB)	Compact Flash type II (Default=1GB)	Compact Flash type II (Default=1GB)
Interface	USB 1.1 x 1 VGA port x 1 (800x600)	USB1.1 x 1 PS/2 port x 2 (Keyboard and Mouse) VGA port x 1 (320x240~1024x768)	USB1.1 x 2 VGA port x 1 (320x240~1024x768)
Ethernet Port	RJ45 x 2, 10/100 BaseTX	RJ45 x1, 10BaseT	RJ45 x 2, 10/100 BaseTX
COM0	For I-87K module in I/O slot	-	-
COM1	RS-232	For I-87K module in I/O slot	For I-87K module in I/O slot
COM2	RS-485	RS-232	RS-232
COM3	RS-232/485	RS-485	RS-485
COM4	RS-232	-	-
I/O Expansion Slots	1/4/8	0/3/7	0/3/7
64-bit Hardware Serial Number	Yes	Yes	Yes
Push button	-	Yes (for Reset)	Yes (For Reset)
Programmable LED	1	-	-
Dual Watchdog Timer	Yes	Yes	Yes
Real Time Clock	Yes	Yes	Yes
DIP Switch (8 bit)	Yes	-	-
Rotary Switch	Yes (0~9)	-	Yes (0~F)

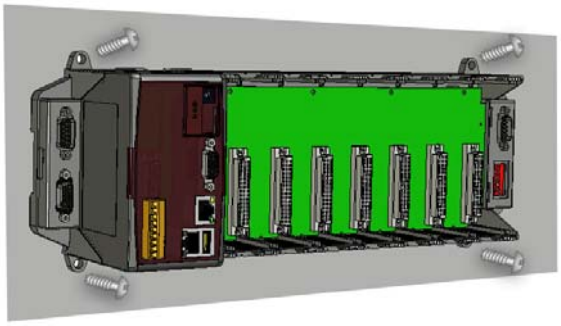
CHAPTER

2

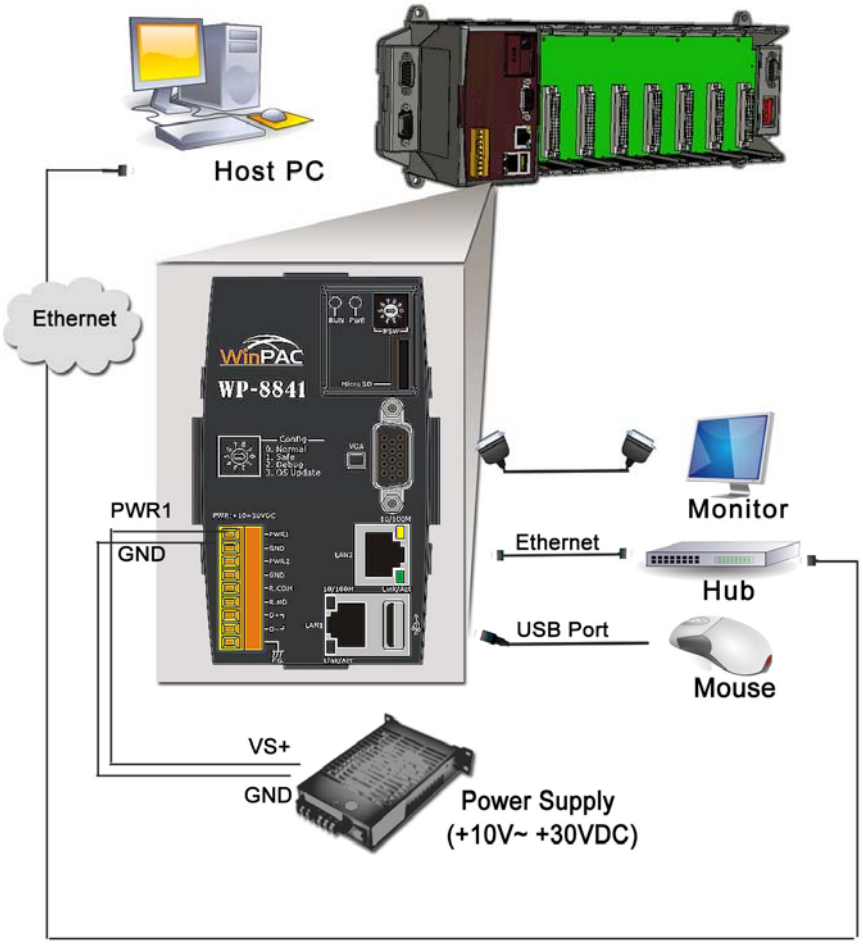
Quick Start

2.1. Installing the WinPAC-8000

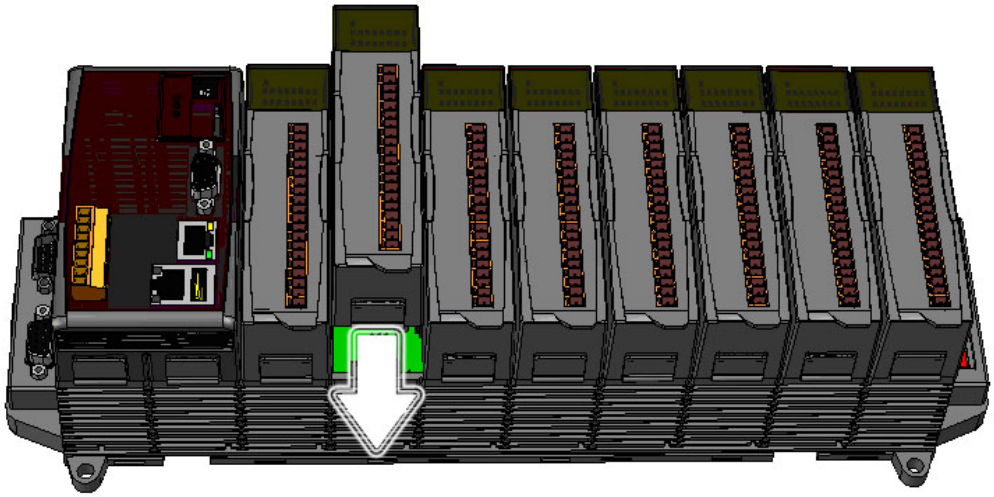
Step 1: Mounting the WinPAC-8000



Step 2: Powering the WinPAC-8000



2.2. Inserting the I/O module



For more information about expansion module that are compatible with the WinPAC-8000 controller, please refer to

http://www.icpdas.com/products/PAC/I-8000/8000_IO_modules.htm

By I-8K and I-87K series expansion modules, support is provided only in High Profile series.

I-8K series I/O modules (High Profile):

i-8K series I/O modules

The communication interface is parallel bus.
High profile: It can be plugged in WinPAC, LinPAC and iPAC-8000 controllers.
Low profile: It can be plugged in WinCon, LinCon and i-8000 controllers.

Type	High Profile	Low Profile	Description
Analog I/O Modules (Selection Guide)	▶ i-8017HW	i-8017H	AI modules have high sampling rate from 1K to 100Ksps (depends on software)
	NEW i-8017HS(Note 1)		
	▶ i-8024W	i-8024	
	NEW i-8037W	i-8037	
	▶ i-8040W	i-8040	
	NEW i-8040P		
	NEW i-8041W	i-8041	
	▶ i-8041AW		
	▶ i-8042W	i-8042	
	▶ i-8046W		
	▶ i-8048W	i-8048	
	▶ i-8050W	i-8050	

I-87K series I/O modules (High Profile):

i-87K series I/O modules

The communication interface is serial bus (RS-485) and the protocol is DCON. i-87K series modules are classed as
High profile: It can be plugged in WinCon, LinCon, i-8000, WinPAC, LinPAC and iPAC-8000 controllers and RU-87Pn, USB-87Pn, i-87K, RF-87K I/O expansion units.
Low profile: It can be plugged in WinCon, LinCon, i-8000 controllers and i-87K, RF-87K I/O expansion units.

Type	High Profile	Low Profile	Max. baudrate	Description
Analog I/O Modules (Selection Guide)	▶ i-87005W		115.2 Kbps	Slew rate of AO channels are programmable
	NEW i-87013W	i-87013		
	NEW i-87015			
	NEW i-87015P			
	▶ i-87016W			
	NEW i-87017W	i-87017		
	NEW i-87017W-A5			
	NEW i-87017R			
	NEW i-87017RC			
	NEW i-87018W	i-87018		
	NEW i-87018R			
	NEW i-87018Z			
	NEW i-87019R			
		i-87022		

Step 1: Read the relevant documentation

► The documentation for I-8K series modules is located at:

CD:\ Napdos\IO_Module\

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/io_module/

► The documentation for I-87K series modules is located at:

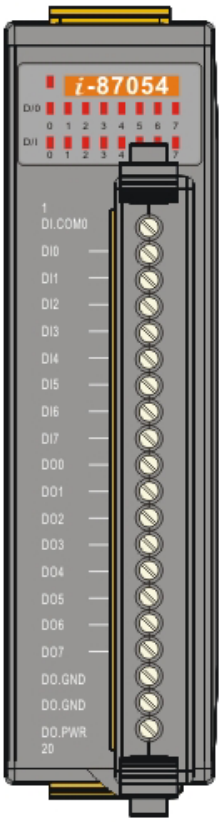
CD:\ Napdos\IO_Module\

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/io_module/

All documents includes the I/O module specifications, pin assignments and wiring connections.

For example, Pin Assignments and Wiring connections for the I-87054 module are as follows:

Pin Assignments



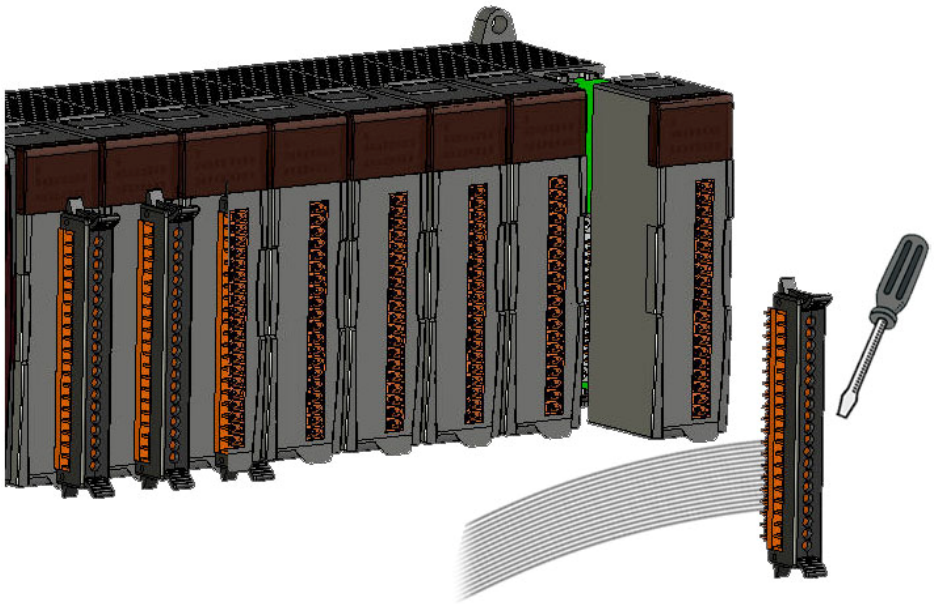
Terminal No.	Pin Assignment Name
01	DI.COM0
02	DI0
03	DI1
04	DI2
05	DI3
06	DI4
07	DI5
08	DI6
09	DI7
10	DO0
11	DO1
12	DO2
13	DO3
14	DO4
15	DO5
16	DO6
17	DO7
18	DO.GND
19	DO.GND
20	DO.PWR

Wire Connection

Input Type	ON State LED ON Readback as 1	OFF State LED OFF Readback as 0
Relay Contact	Relay ON Relay Close DI.COM DIx	Relay Off Relay Open DI.COM DIx
TTL/CMOS Logic	Voltage < 1V Logic Power Logic Level Low DI.COM DIx	Voltage > 3.5V Logic Power Logic Level High DI.COM DIx
NPN Output	Open Collector On ON DI.COM DIx	Open Collector Off OFF DI.COM DIx
PNP Output	Open Collector On ON DI.COM DIx	Open Collector Off OFF DI.COM DIx

Output Type	ON State LED ON Readback as 1	OFF State LED OFF Readback as 0
Drive Relay	Relay ON DO.PWR DOx DO.GND	Relay Off DO.PWR DOx DO.GND
Resistance Load	DO.PWR DOx DO.GND	DO.PWR DOx DO.GND

Step 2: Connect the wires



Step 3: Insert the I/O module

2.3. Operation Modes of the WinPAC-8000

There are four basic operation modes for running WinPAC-8000 that can be determined through a rotary switch. All the modes of operation will be explained in a later section

The table below lists the operating mode selection.

Rotary switch position	Modes of operation
0	Normal mode (Default)
1	Safe mode
2	Debug mode
3	OS update mode
4	Development mode
5	(Reserved)
6 ~ 9	(For user)

2.3.1. Normal mode (Default)

Normal mode is the default mode of operation and the one you will use most of the time. Use this mode for more tasks and configurations. Programs also are executed in this mode.

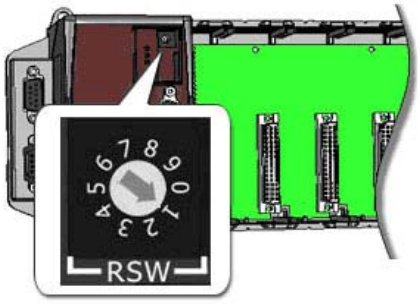
2.3.2. Safe mode

Safe mode is a trouble shooting. The mode loads the minimum required device drivers and system services to boot the WinPAC-8000.
If you have malicious software or a program cause the WinPAC-8000 cannot be boot or run the normal mode, you can boot in safe mode to solve the problem.

In normal mode, if the new settings are not saved when you change and save the settings using the WinPAC Utility, to solve this problem, perform the following steps:

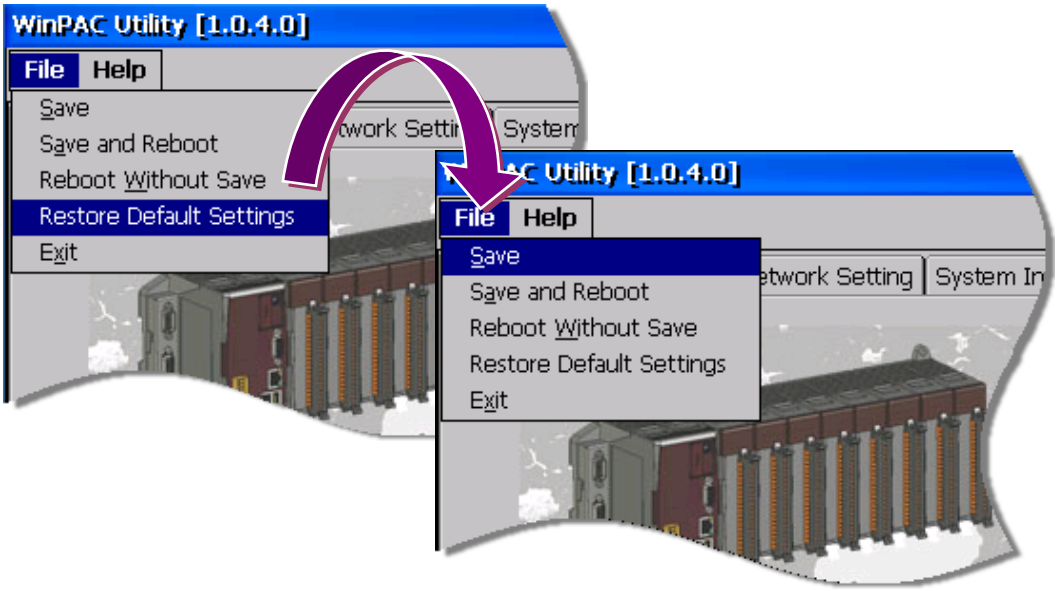
Step 1: Restart the WinPAC-8000 in safe mode

Turn the rotary switch to “1”, and then restart the WinPAC-8000.



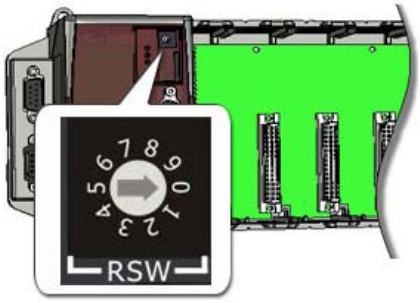
Step 2: Start the WinPAC Utility to restore the default settings

Start the WinPAC Utility, and then click the “Restore Default Settings” command and “Save” command from the “File” menu



Step 3: Restart the WinPAC-8000 in normal mode

Turn the rotary switch to “0”, and then restart the WinPAC-8000.



2.3.3. Debug mode

Debug mode is a special environment in which program debug functions can be used in addition to normal system functions.

Debug mode is unsupported.

2.3.4. OS Update mode

OS update mode is a way used to update OS. To update the WinPAC-8000 OS image, please refer to “2.6. Updating the WinPAC-8000 OS image”

2.3.5. User mode

Rotary switch position 6, 7, 8, 9 are reserved for user’s applications.
When WinPAC-8000 is boot with one of these rotary switch positions, it is boot at normal mode. User’s application can check the rotary switch position to run at different mode.

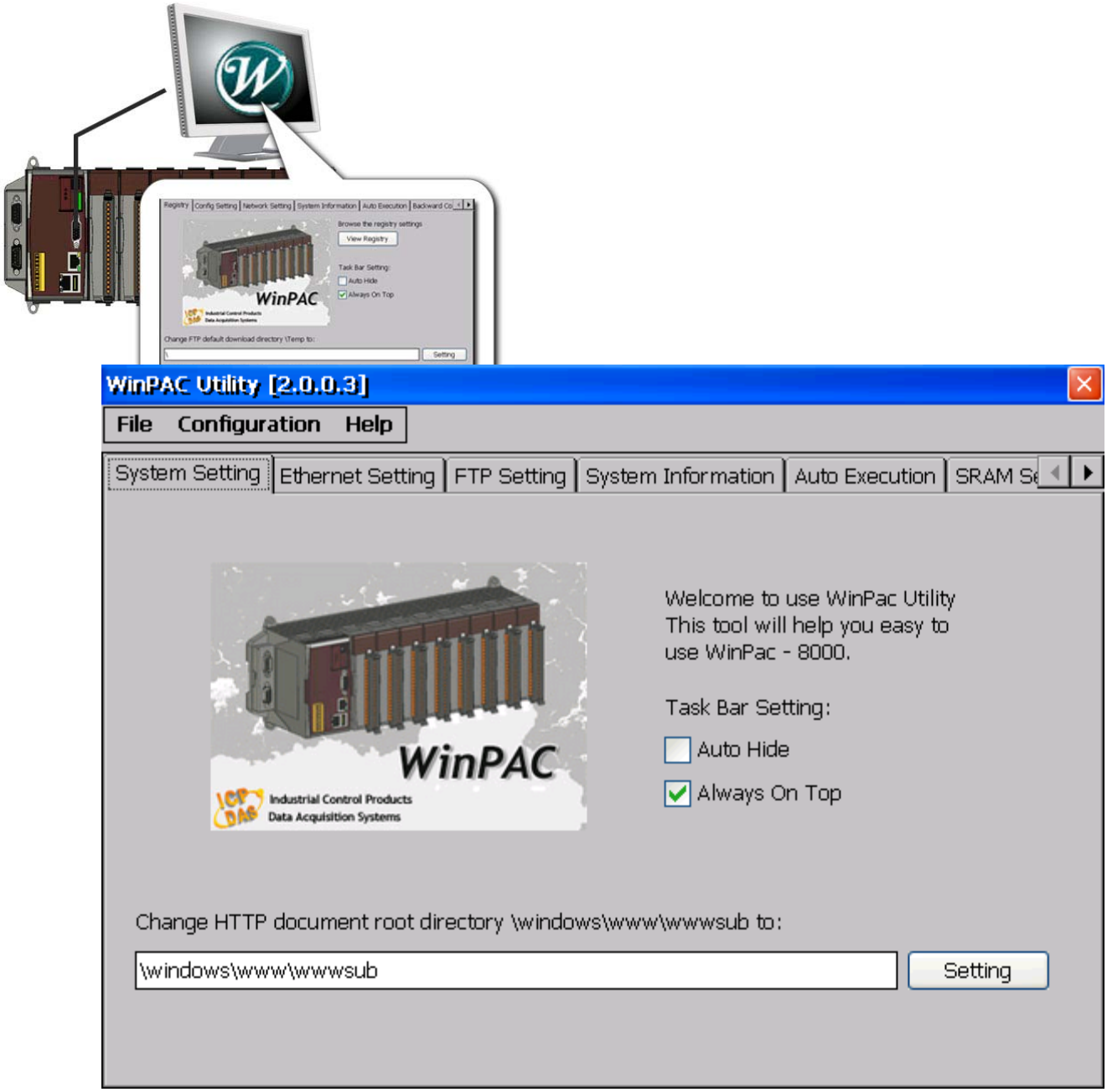
2.4. WinPAC Utility for Configuring the WinPAC-8000

The WinPAC Utility is a tool which is designed to quickly control and management the WinPAC 8000 controller.

Starting the WinPAC Utility to configure the WinPAC-8000

To start the WinPAC Utility, you can double-click the WinPAC Utility shortcut on the desktop.

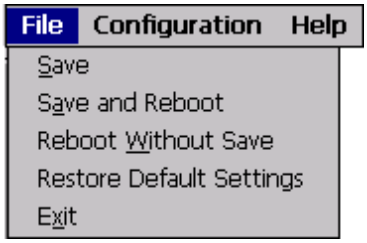
The WinPAC Utility window contains two basic components: menu bar and property tabs.



2.4.1. Using the WinPAC Utility menu bar

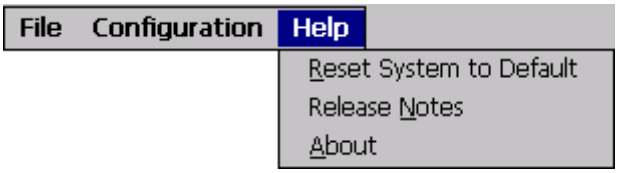
The WinPAC Utility includes the following function menu, all function menu will be explained in a later section:

- File menu

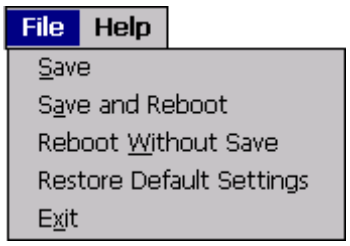


- Configuration menu

- Help menu



2.4.1.1. File menu



The menu commands	Use to
Save	This action is used to save the settings into flash, the new settings don't take effect until the WinPAC-8000 controller restart.
Save and Reboot	This action is used to save the settings into flash and restart the WinPAC-8000 controller, the new settings will take effect after the WinPAC-8000 controller restart.
Reboot Without Save	This action is used to restart the WinPAC-8000 controller without save the settings into flash.
Restore Default Settings	This action is used to restart the settings of WinPAC-8000 to its factory default values. The settings include config setting, network setting, auto execution, etc.
Exit	This action is used to exit the WinPAC Utility.

2.4.1.2. Configuration menu

The menu commands	Use to
Import Registry Key	This action is used to back up a subkey of registry by using a registration entries (.reg) file. **How to use:** Step 1: Select the “Import Registry Key”, then the “Open” dialog box will appear. Step 2: On the “Open” dialog box, select a specific .reg file to import. **Warning:** 1. The .reg file which should be saved by “Export Registry Key” 2. It will not save automatically after import a .reg file.
Export Registry Key	This action is used to make a back up of a registry subkey **How to use:** Step 1: Select the “Export Registry Key”, then the “Export Registry” box will appear Step 2: Select a specific root key. Step 3: Input a specific path of subkey. Step 4: Push the “OK” button, then the “Save As” dialog box will appear prompting you to select a location where you want to save this exported file. **Warning:** The export operation will export all the subkeys of the specific key which you input.

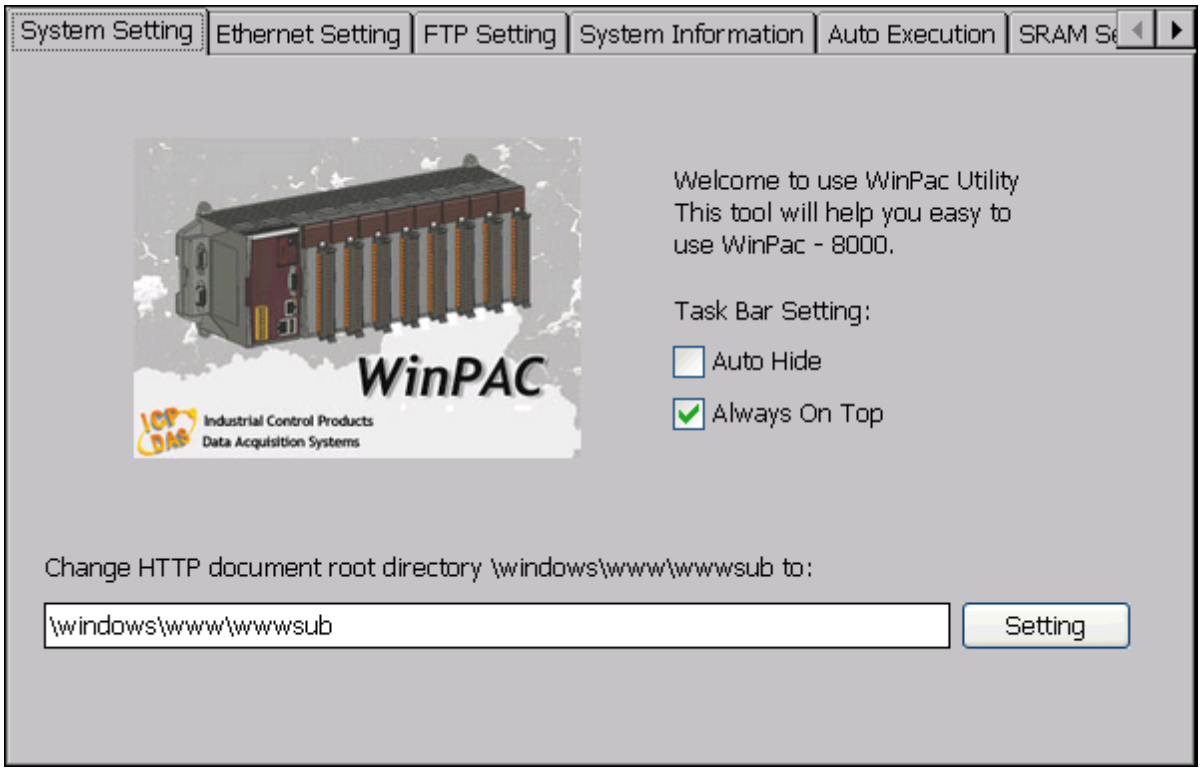
The menu commands	Use to
Store All Registry Setting	Stores all registry setting to flash from .das file which is saved by “Dump All Registry Setting”. **How to use:** Step 1: Select the “Store All Registry Setting” , then the “Open” dialog box will appear. Step 2: On the “Open” dialog box, select a specific .das file to store. **Warning:** 1. The .das file which should be saved by “Dump All Registry Setting” 2. It will save automatically after store .das file.
Dump All Registry Setting	Dump all registries setting to .das file. **How to use:** Select the “Dump All Registry Setting”, then the “Save As” dialog box will appear prompting you to select a location where you want to save this exported file.

2.4.1.3. Help menu

The Help menu commands	Use to
Reset System to Default	This action is used to reset the system interrupt status to default. The operation used in the situation when the interrupt crash. You can select this operation to reset the interrupt status without rebooting the device.
Release Notes	This action is used to check out what's new and the know issues.
About	This action is used to display a dialog box with information about WinPAC Utility, including the current version and copyright information.

2.4.2. Using the WinPAC Utility property tabs

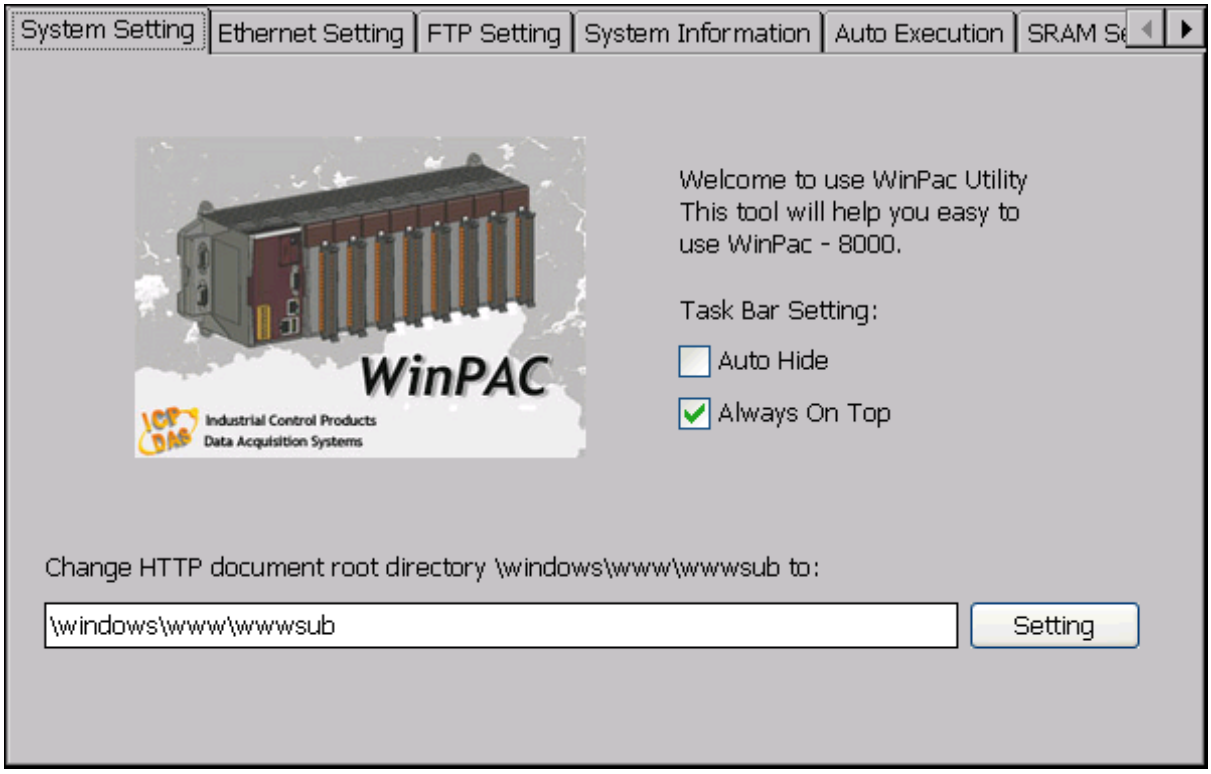
The WinPAC Utility includes the following property tabs, all property tabs will be explained in a later section:



- System Setting tab
- Ethernet Setting tab
- FTP Setting tab
- System Information tab
- Auto Execution tab
- SRAM Setting tab
- Backward Compatibility tab

2.4.2.1. System Setting tab

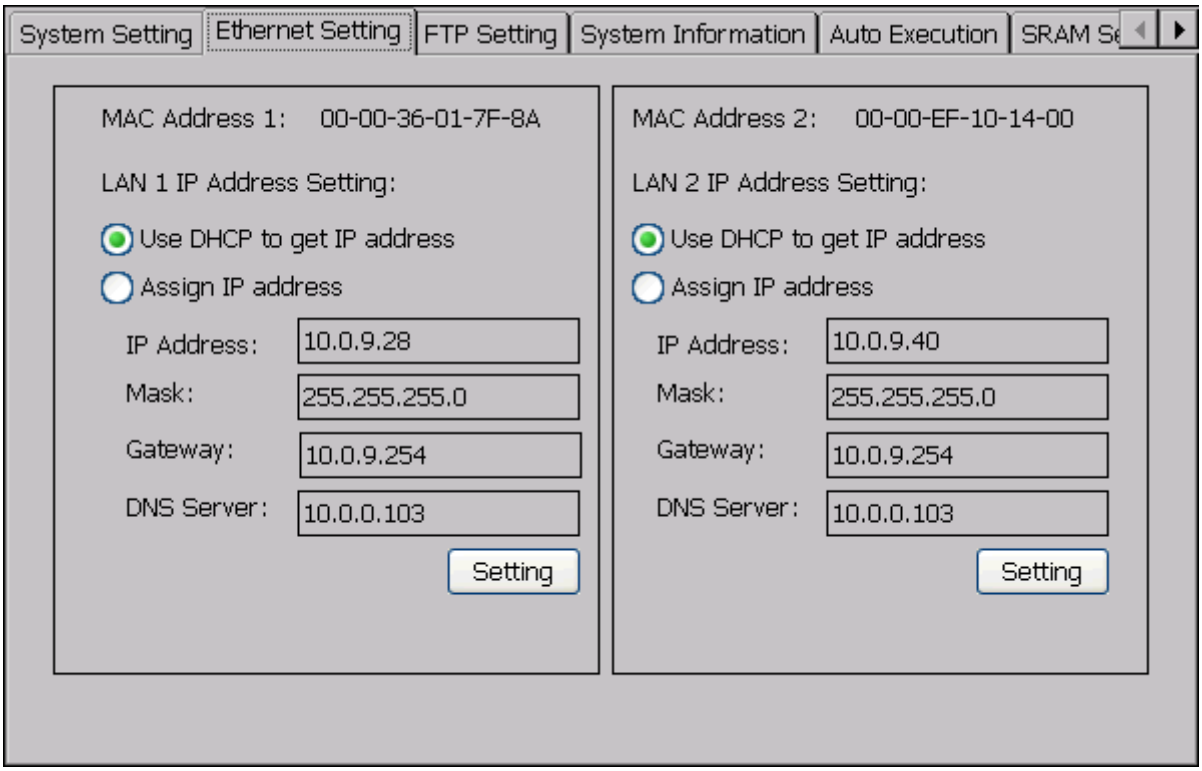
The System tab provides functions to configure the task bar HTTP directory path.



The tab use to	How to use
Configure the task bar settings	Hide: Select the “Auto Hide” check box Always on top: Select the “Always On Top” check box
Change the HTTP directory path	Enter a new path in the Change HTTP default download directory \windows\www\wwwsub to: field, and then press the “Setting” button

2.4.2.2. Ethernet Setting tab

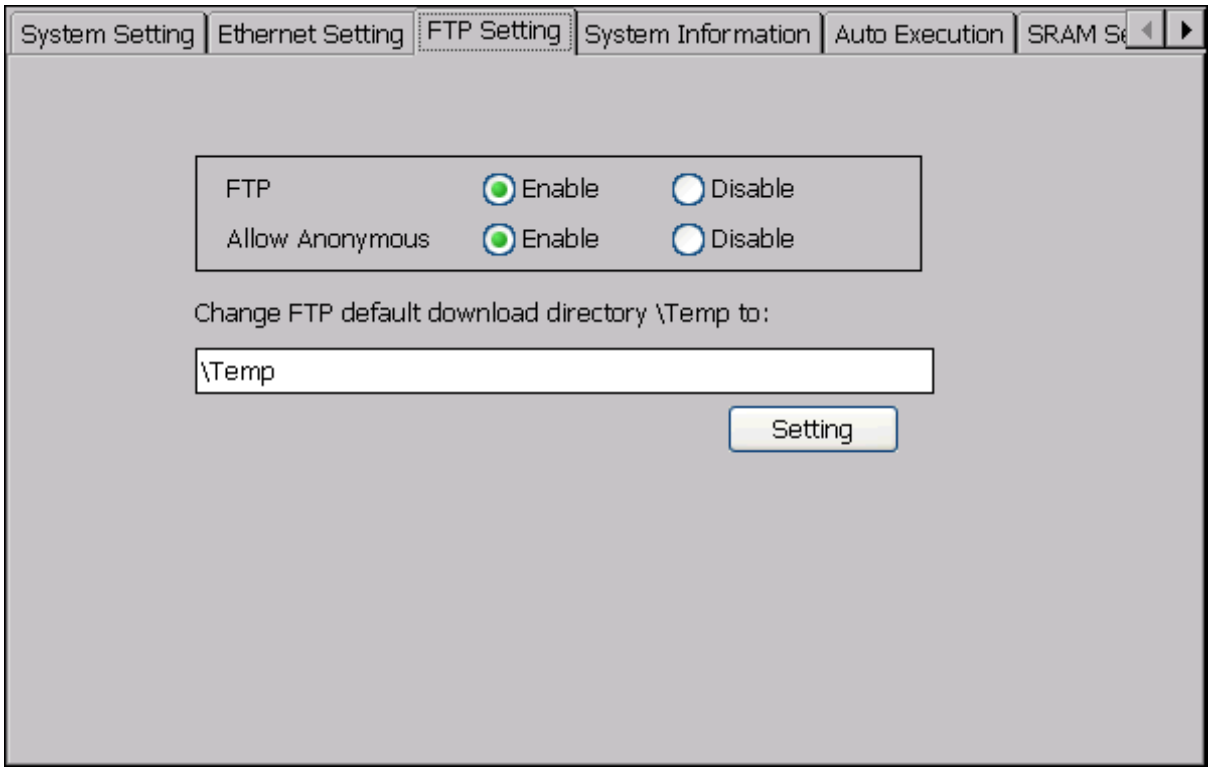
The Ethernet Setting tab provides functions to configure either DHCP (Roaming) or manually configured (Static) network settings and to monitor the MAC address. Generally, DHCP is the default settings, but if you don't have a DHCP server, you must configure the network settings by using manual configuration.



The tab use to	How to use
Monitor the MAC address	See the MAC address field that displays the physical address of Ethernet port
Configure the network settings	Using DHCP: Select the "Use DHCP to get IP address", and then press "Setting" button. Manually assign IP: Select the "Assign IP address" option, and then enter the "IP Address", "Mask", "Gateway" and "DNS Server" fields, and then press "Setting" button.

2.4.2.3. FTP Setting tab

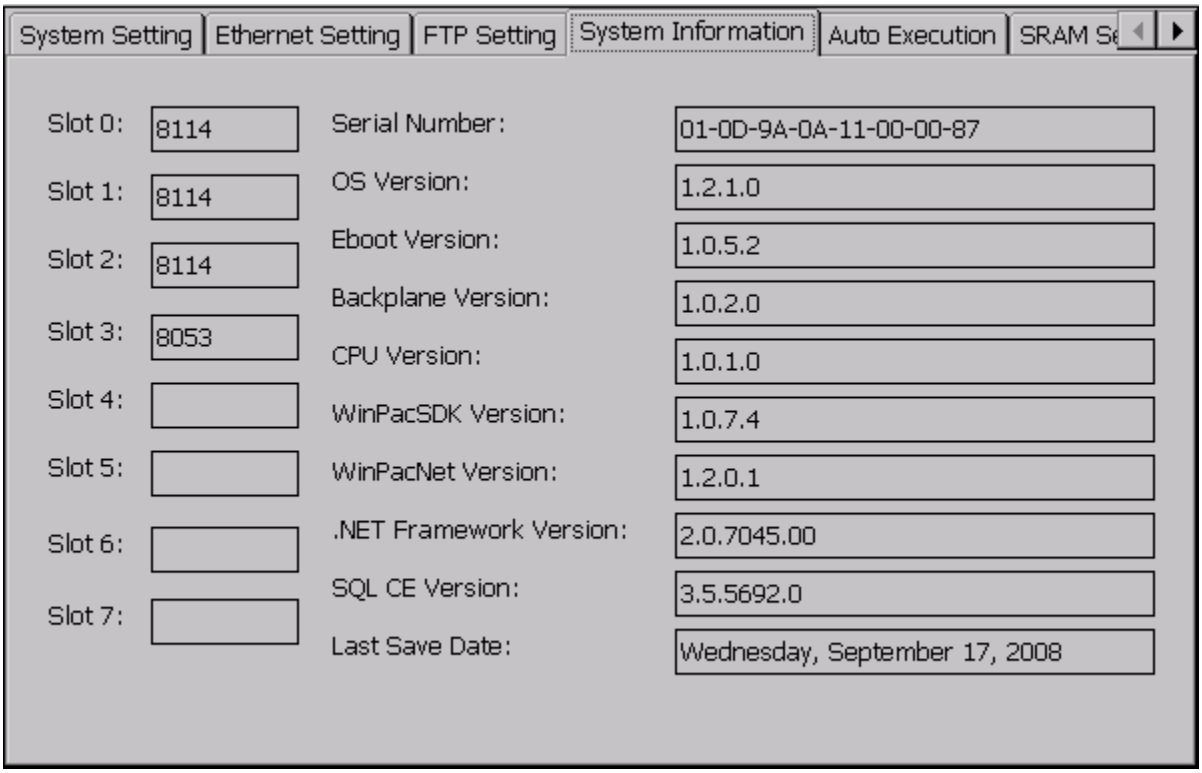
The FTP Setting tab provides functions to enable/disable the FTP access, enable/disable anonymous FTP access, and configure the FTP directory path.



The tab use to	How to use
Enable or Disable the FTP access	Enable: Select the “Enable” option Disable: Select the “Disable” option
Enable or Disable anonymous access	Enable: Select the “Enable” option Disable: Select the “Disable” option
Change the FTP directory path	Enter a new path in the Change FTP default download directory \temp to: field, and then press the “Setting” button

2.4.2.4. System Information tab

The System Information tab provides functions to monitor necessary system information of the WinPAC-8000. The system information is most important note of version control for upgrading system.



This tab provides the following information to monitor the system information of the WinPAC-8000.

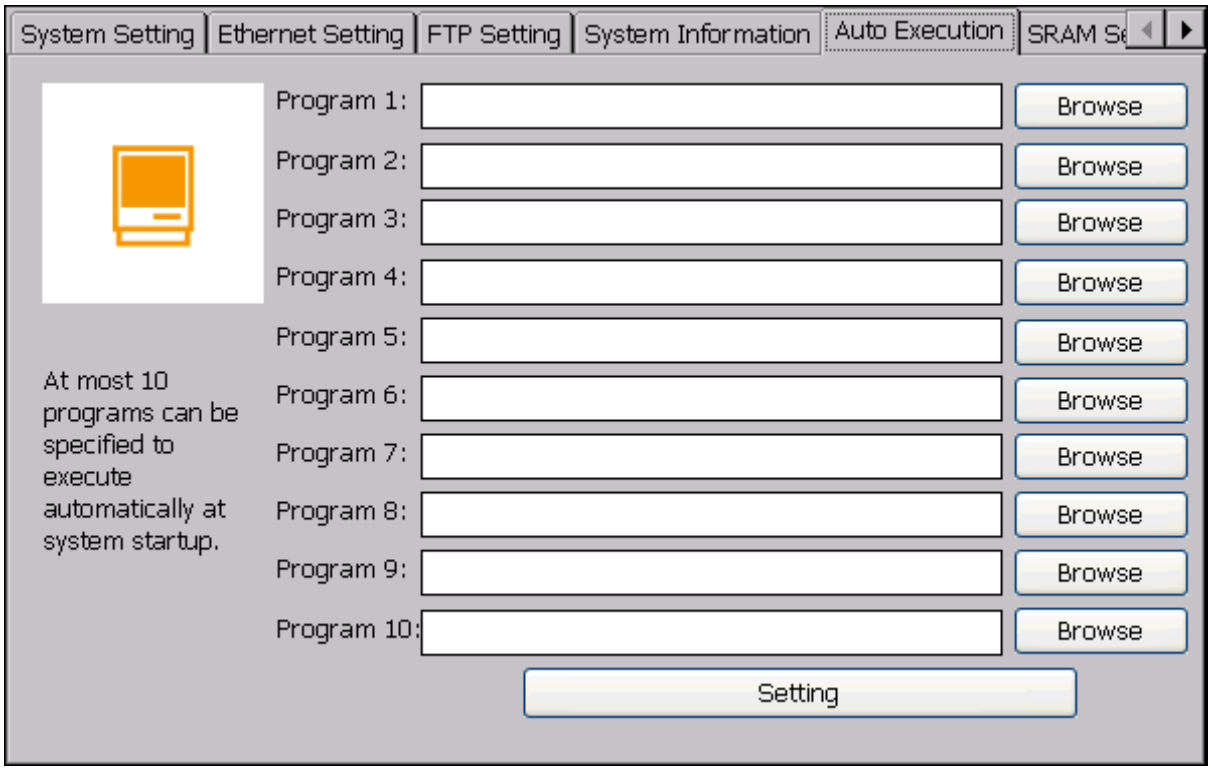
The tab use to	How to use
The local I/O slots	Display the module names plugged in the WinPAC-8000
Serial Number	Displays the serial number of the WinPAC-8000.
OS Version	Displays the current operating system version of the WinPAC-8000.
Eboot Version	Displays the current Eboot software version of the WinPAC-8000.
Backplane Version	Displays the current Backplane version of the WinPAC-8000

The tab use to	How to use
CPU Version	Displays the current CPU version of the WinPAC-8000.
WinPacSDK Version	Displays the current WinPACSDK_DLL version of the WinPAC-8000.
WinPacNet Version	Displays the current WinPACNet_DLL version of the WinPAC-8000.
.NET Framework Version	Displays the current .NET Framework version of the WinPAC-8000.
SQL CE Version	Displays the current SQL CE version of the WinPAC-8000.
Last Save Date	Displays the last saving time of the WinPAC Utility.

2.4.2.5. Auto Execution tab

The Auto Execute tab provides functions to configure programs running at WinPAC-8000 startup, it allows users to configure ten execute files at most.

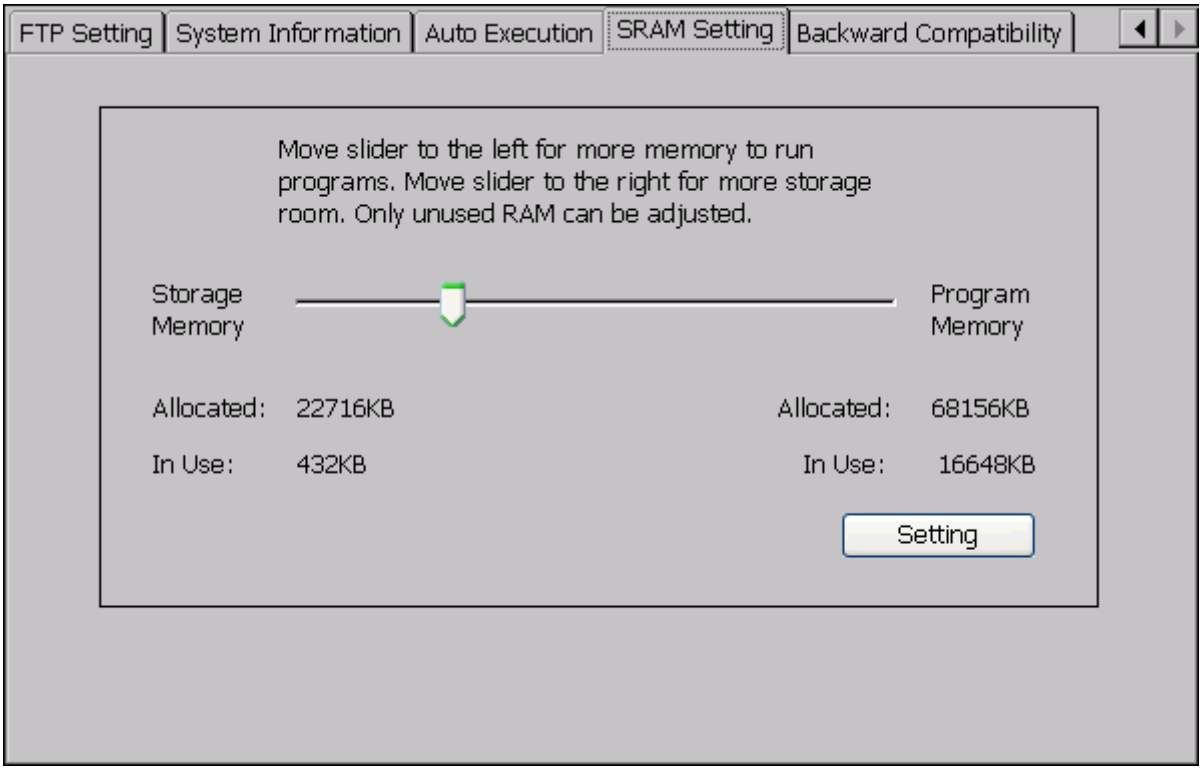
The allowed file types are .exe and .bat, and they are executed in order of program 1, program 2, ...



The Auto Execution tab use to	How to use
Configure programs running at startup	Press the “Browse” button to select the execute file which you want, and then press the “Setting” button

2.4.2.6. SRAM Setting tab

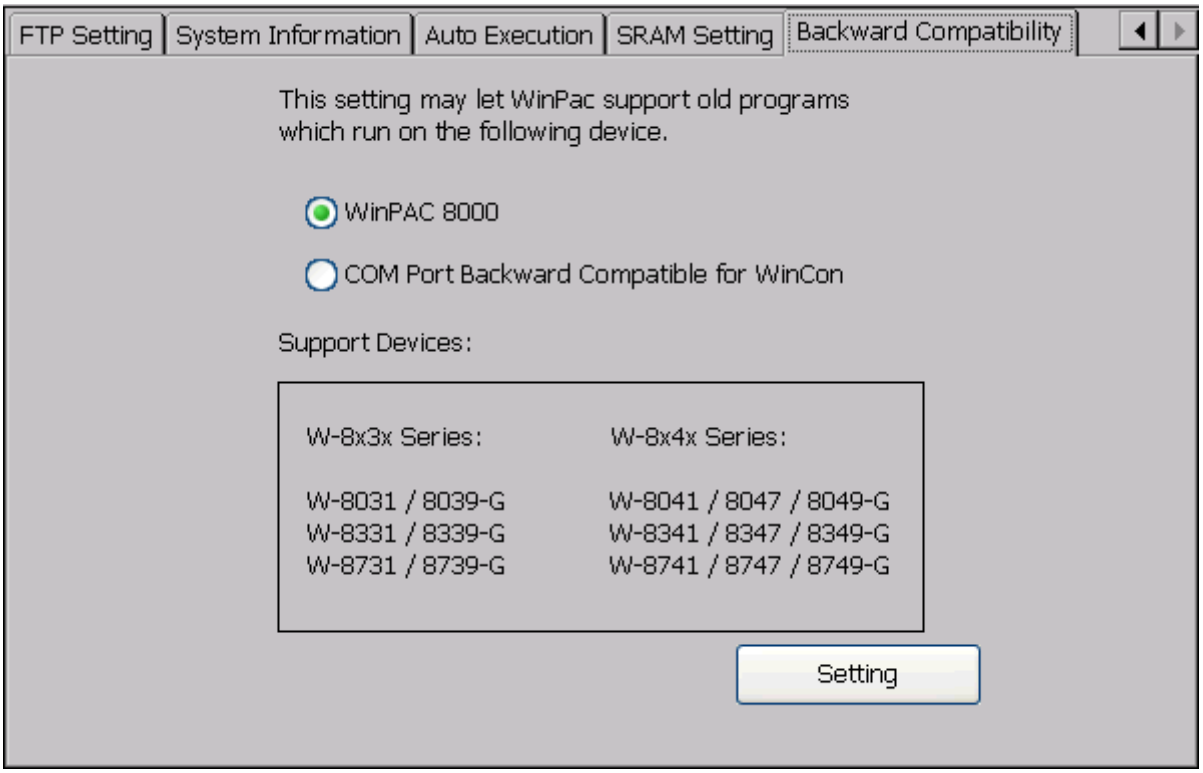
The SRAM Setting tab provides functions to adjust the unused RAM.



The Network Setting tab use to	How to use
Adjusting the unused RAM	Move the slider left to release more memory running programs or move the slider right to release more storage room, and then press the “Setting” button.

2.4.2.7. Backward Compatible tab

The Backward Compatible provides functions to keep old programs running on WinPAC-8000 without any modification.



The Backward Compatibility tab use to	How to use
Configure programs running at WinPAC-8000	Select the “COM Port Backward Compatible for WinCon”, and then press “Setting” button.

2.4.3. The factory default settings

The following table lists the factory default settings of the WinPAC Utility.

System Setting tab:

Function	Settings
Auto Hide	Uncheck
Always On Top	Check
HTTP default download directory	\window\www\wwwpub

Ethernet Setting tab:

Function	Settings
LAN1	DHCP
LAN2	DHCP

FTP Setting tab:

Function	Settings
FTP	Disable
Allow anonymous	Disable
FTP default download directory	\Temp

Auto Execution tab:

Function	Settings
All field	Empty

SRAM Setting tab:

Function	Settings
Memory settings	Quarter of the size of SRAM

Backward Compatibility tab:

Function	Settings
Option	WinPAC-8000

2.5. DCON Utility for configuring i-7K and I-87K series I/O modules

The DCON Utility is a tool which is designed to quickly control and management I-7K and I-87K series I/O modules via COM port and Ethernet port.

The following table summarizes the information about TCP port mapping and i-7K and I-87K series expansion modules of COM port on WinPAC.

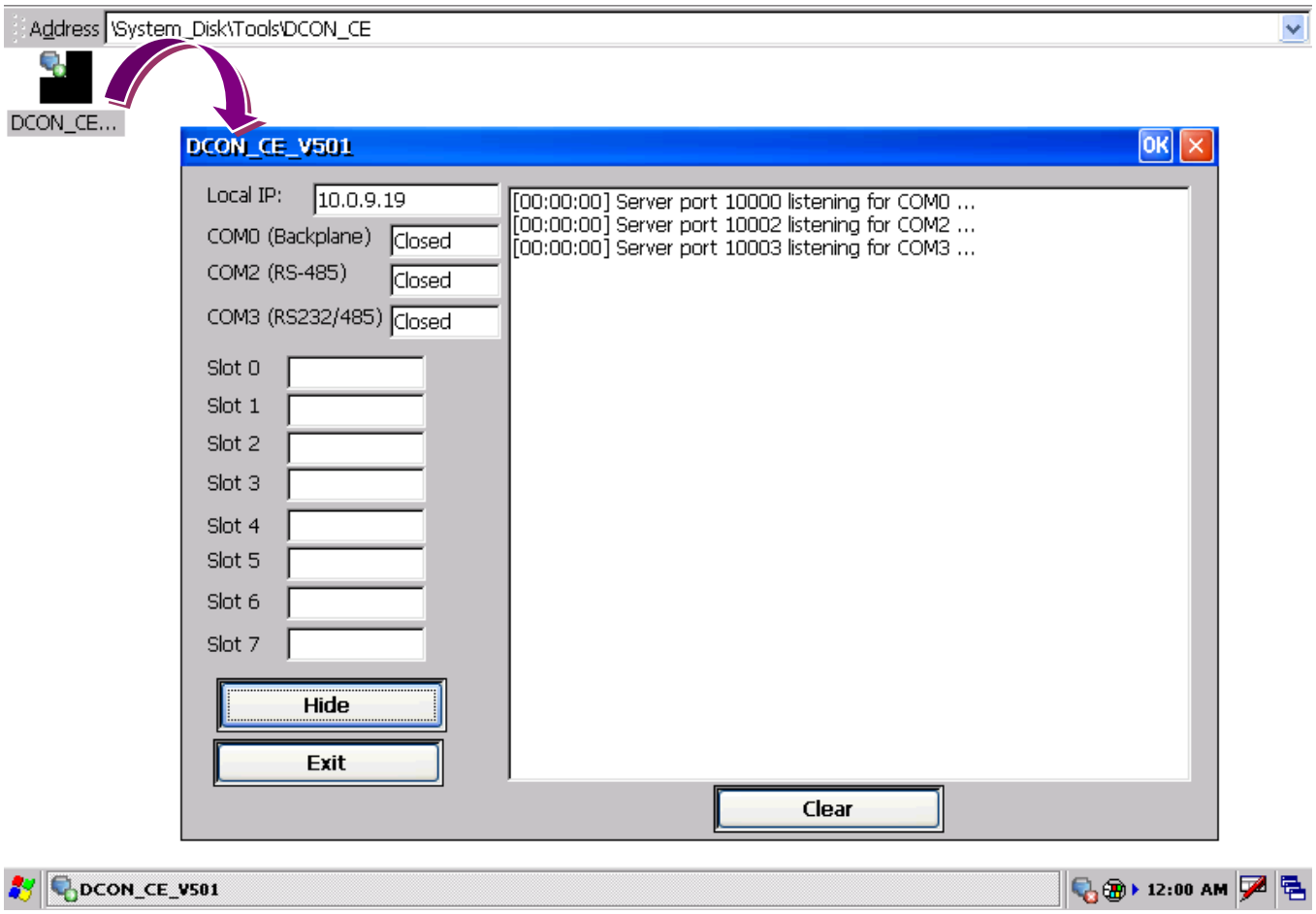
COM port of WinPAC	TCP Port	Support modules
COM 0 (Backplane)	10000	87K (High profile series)
COM 2 (RS-485)	10002	7K, 87K
COM 3 (RS-232/RS-485)	10003	7K, 87K

Starting the DCON Utility to configure I-87K series I/O modules

Step 1: On the WinPAC-8000, execute the DCON firmware

The DCON firmware can be obtained from:

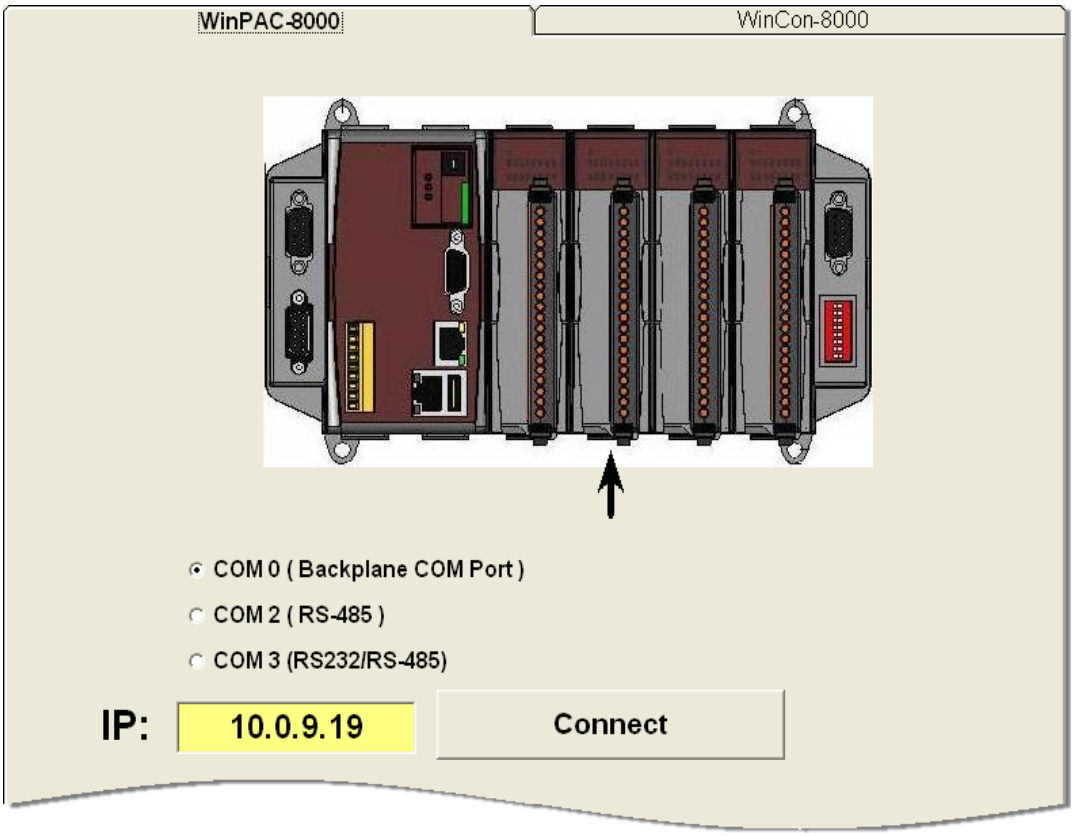
\System_Disk\tools\DCON_CE



Step 2: Double-Click the DCON Utility shortcut on the desktop of Host PC

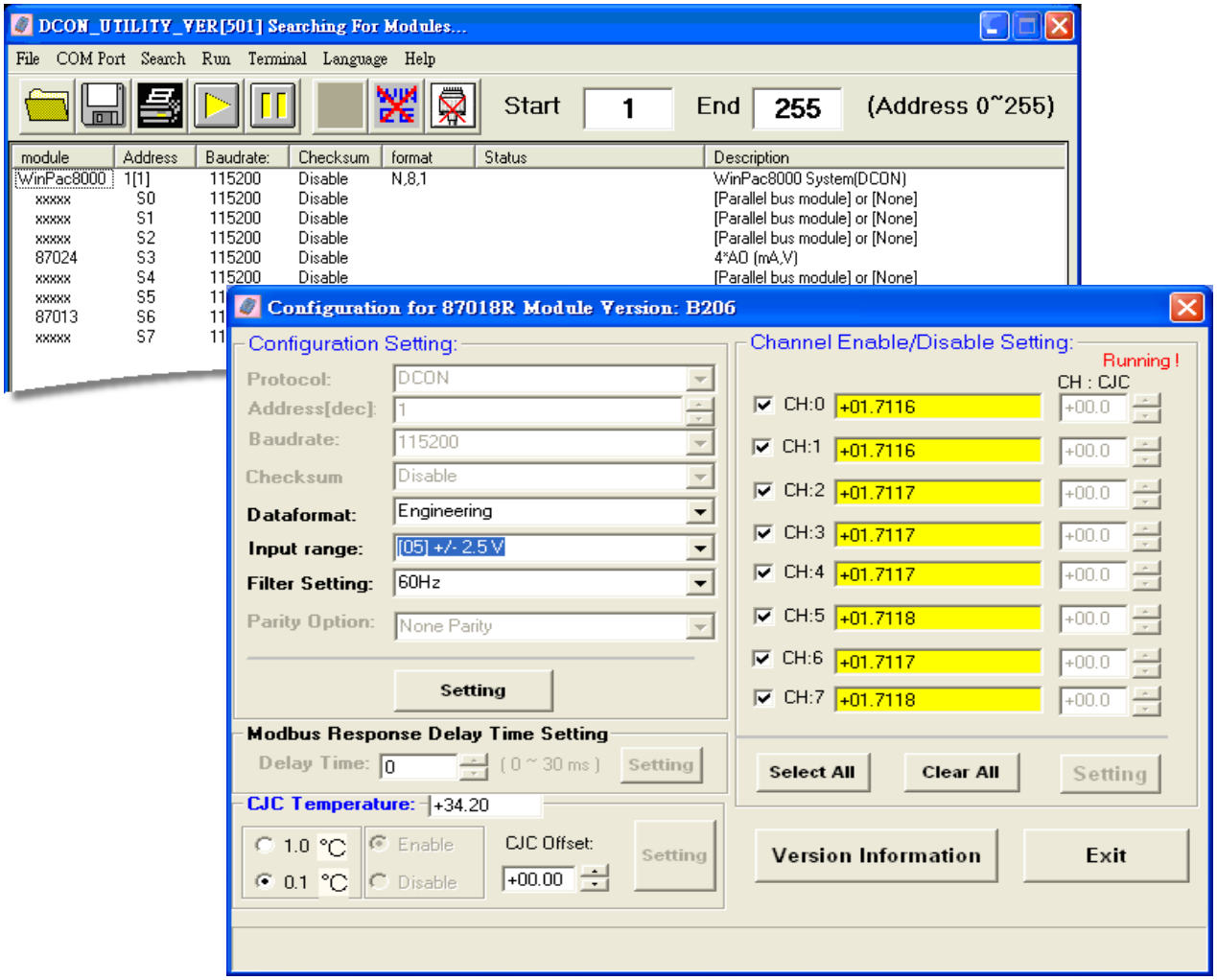
Step 3: Click on the WIN CE command button

Step 4: On the "WINCE device connection" dialog, choose a connection type and then type the IP address in the "IP" filed, then click the "Connect" button to search i-7K and I-87K series I/O modules

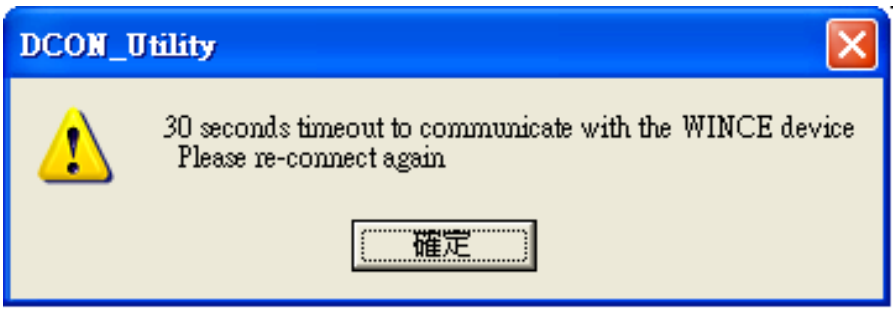


If DCON Utility cannot connect to WinPAC, the Ethernet connection between Host PC and WinPAC might be rejected by fire wall, please contact with MIS to open the Ethernet port.

Step 5: Click on the module name from the list to enter the configure form

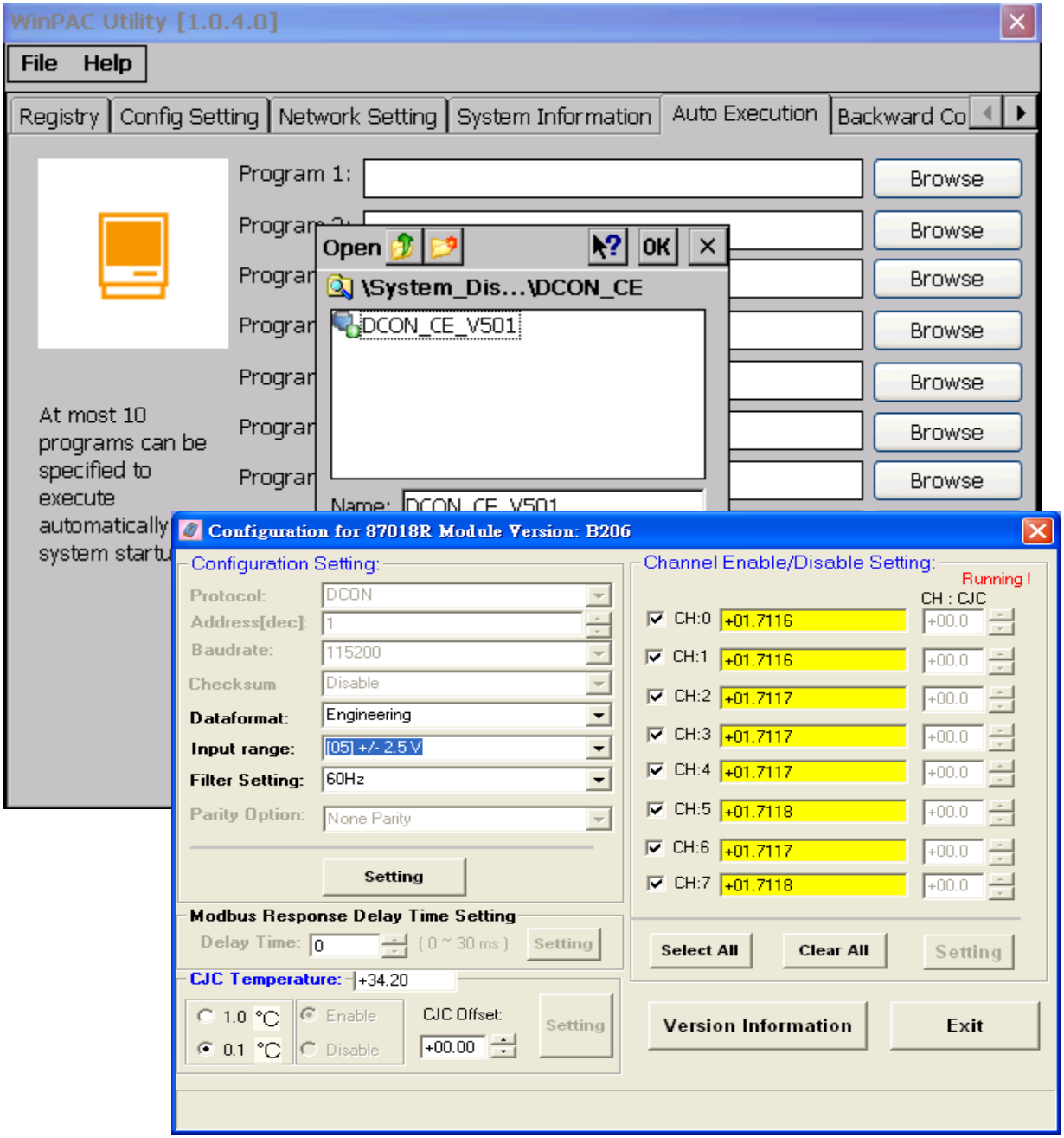


If there is no operation within 30 seconds, the connection will auto close to release the COM port occupied.



Advice

If you want to execute the DCON program automatically every time you start, you can easily take it by using WinPAC Utility.



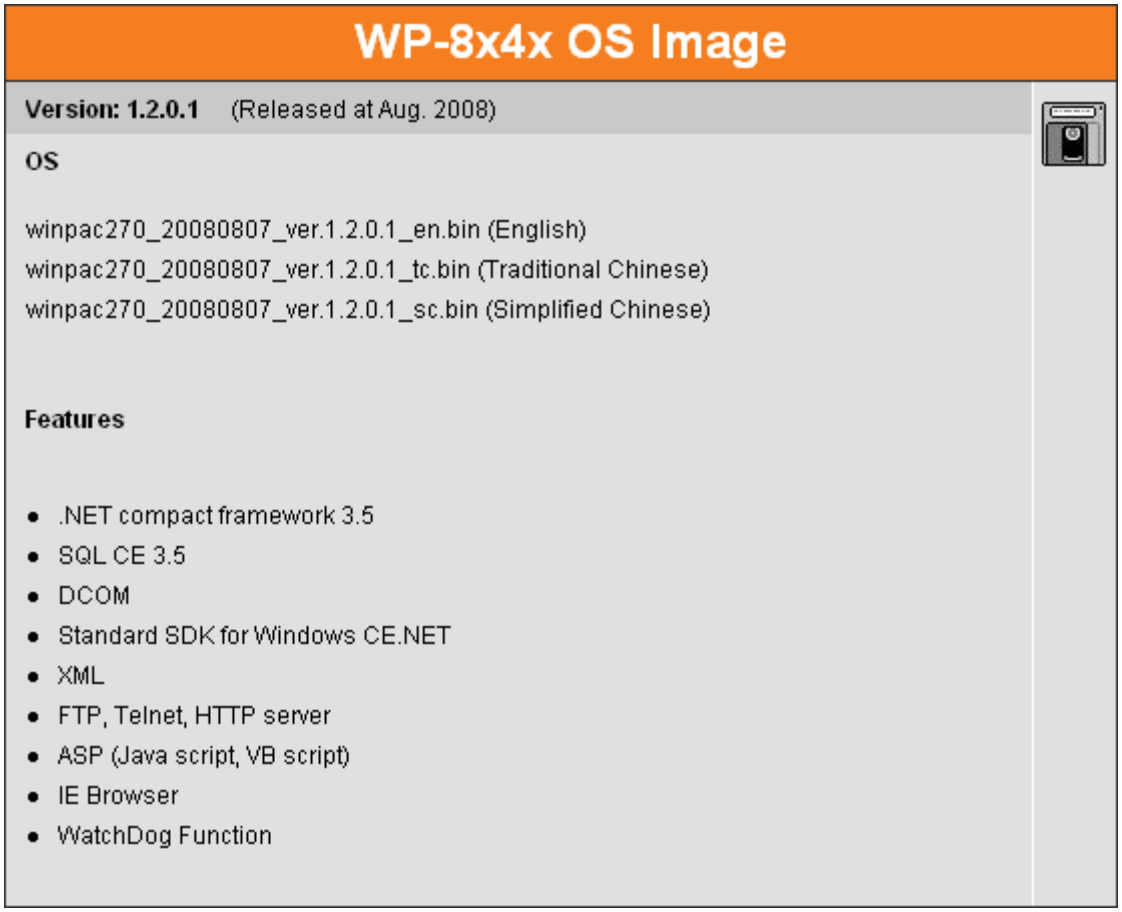
2.6. Updating the WinPAC-8000 OS image

ICP DAS will continue to add additional features to WinPAC-8000 OS in the future, so we advise you to periodically check the ICP DAS web site for the latest updates to WinPAC-8000 OS.

For more detailed information for updating the WinPAC OS image, please refer to **"Appendix E.1. How to upgrade for WinPAC-8000"**

The latest version of the WinPAC-8000 OS image file can be obtained from:

http://www.icpdas.com/products/PAC/winpac/download/winpac_8000/download_os_images.htm

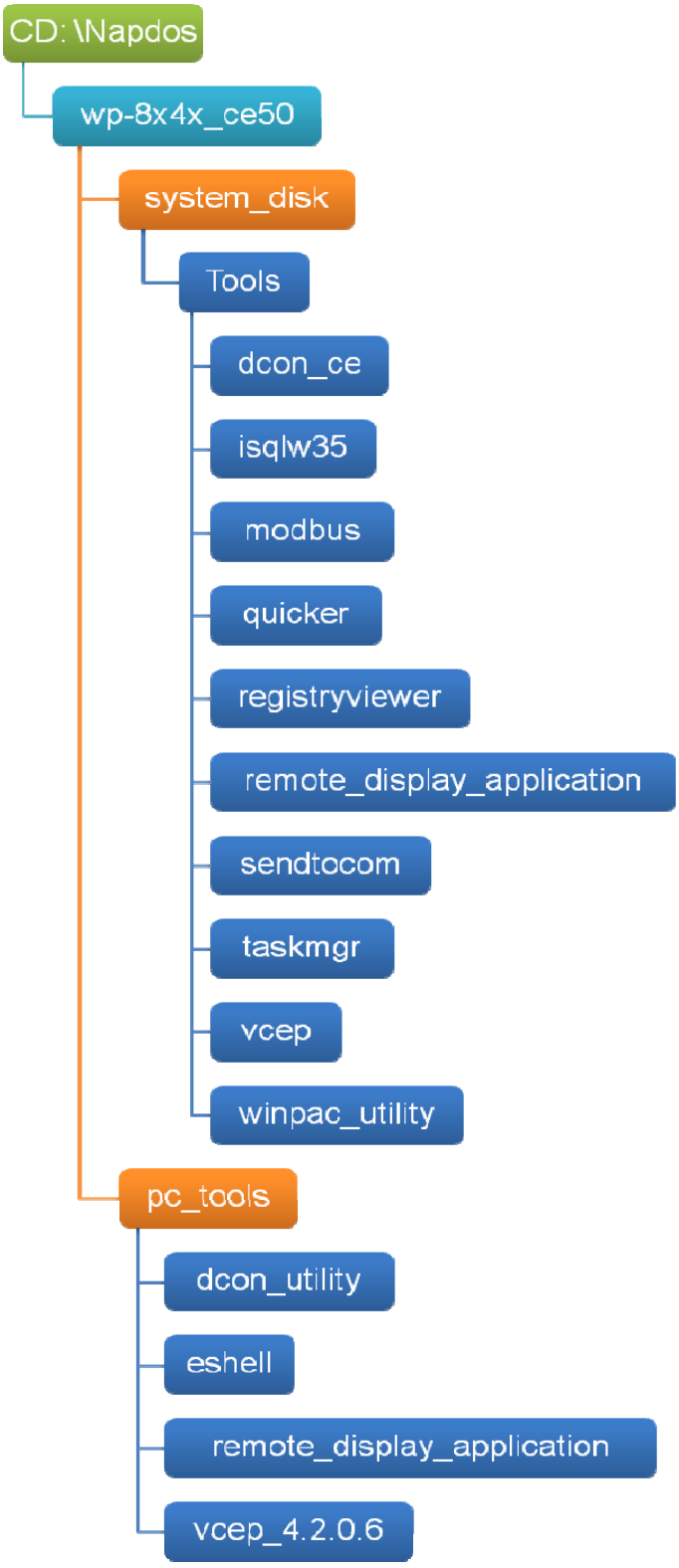


CHAPTER

3

WinPAC-8000 Tools

Following Tools has been installed on the WinPAC-8000.



- **DCON_CE**

  With Host PC running the DCON Utility, on the WinPAC-8000, the DCON_CE program allows user to view and monitor the status of the DCON Utility.

- **ISQLW35**

  The ISQLW35 implements SQL server compact 3.5 Query Analyzer.

- **Modbus**

  The Modbus provides various applications of Modbus protocol for configuring the WinPAC-8000.

- **Quicker**

  Quicker is an integrated omnibus software package, it allows user to quickly establish a DCS control system.

  For more information about the Quicker, please refer to “3.2. Quicker”.

- **RegistryViewer**

  The Registry Viewer allows user to view the registry value of Windows CE Operating System.

- **Remote display application**

  The remote display application allows user to view the display remotely of the WinPAC-8000 on a Host PC.

- **SendToCOM**

  The SendToCOM allows user to send/receive data to/from the expansion module via serial port.

- **TaskMgr**

  The TaskMgr provides details about programs and processes running on the WinPAC-8000.

- **VCEP**

  The VCEP allows user to manage the WinPAC-8000 remotely on a Host PC.

  For more information about the Quicker, please refer to “3.4. VCEP (Virtual CE Pro)”.

- **WinPAC Utility**

  The WinPAC Utility provides various useful functions such as configuring Ethernet settings, monitoring system settings and FTP services .etc for easy and quick management.

  For more information about the Quicker, please refer to “3.5. WinPAC Utility”.

3.1. DCON Utility (For PC side)

The DCON Utility is s toolkits kits that help user search the network, easily to Configure the I/O modules and test the I/O status via the serial port (RS-232/485) or ethernet port (using virtual com port). It support not only the DCON Protocol I/O modules but also the M Series I/O Modules (Modbus RTU M-7K,M-87K and will support Modbus ASCII M-87K) now.

For more detailed information on DCON Utility application, please refer to

http://www.icpdas.com/products/dcon/introduction.htm

3.2. Quicker

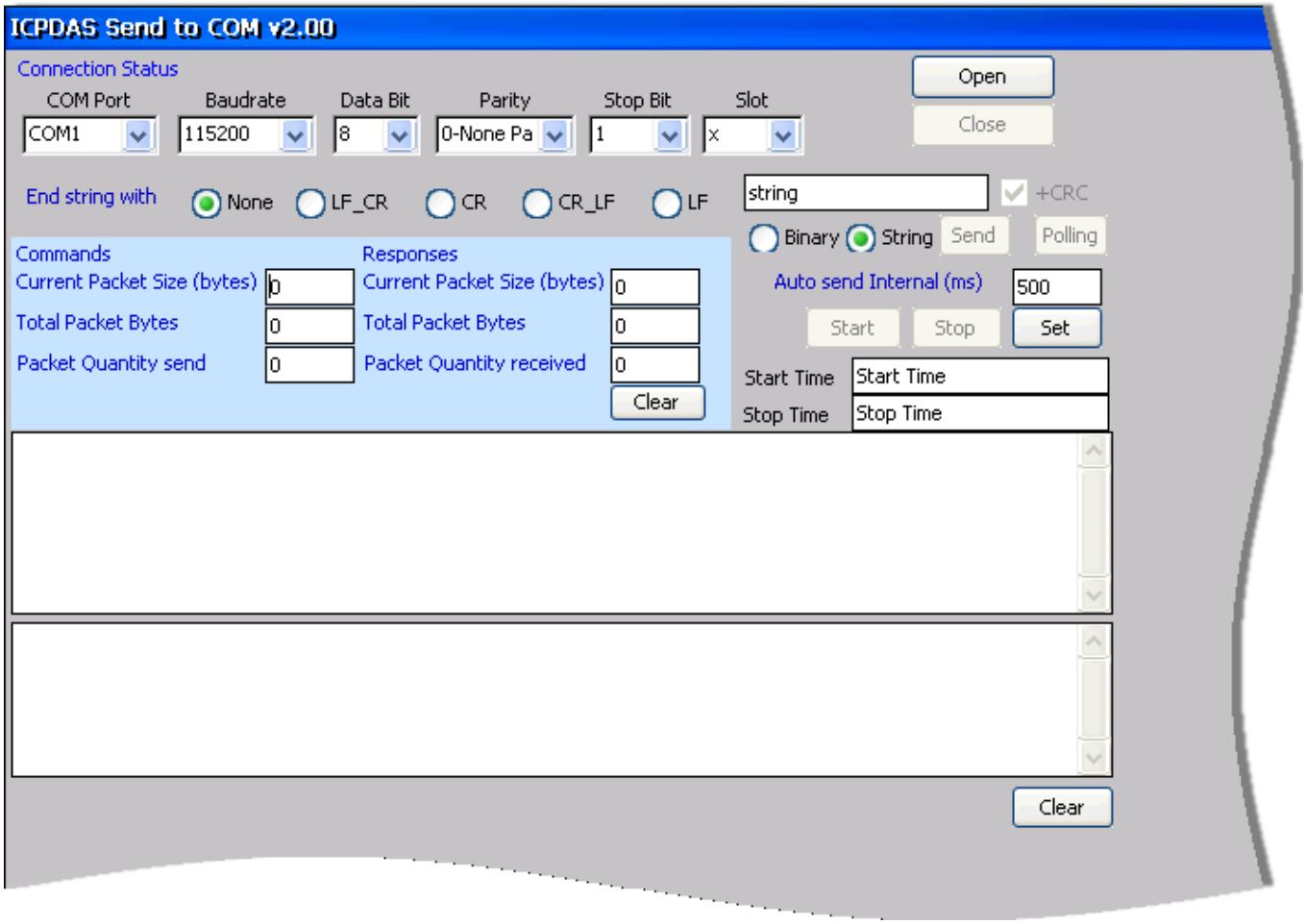
What is Quicker? Quicker is an integrated omnibus software package which combines OPC, Modbus TCP, Modbus RTU services, and Scankernel together. The particular design, “Rule Script”, lets user can quickly establish a DCS control system with logic control, multi-communication services. For UI design, Quicker uses an explorer-style user interface to display a hierarchical tree of modules and groups with their associated tags. A group can be defined as a subdirectory containing one or more tags. A module may have many subgroups of tags. All tags belong to their module when they are scanned to perform I/O. (The “OPC” stands for “OLE for Process Control” and the “DA” stands for “Data Access”.) For software use, Quicker creates a set-up procedure requiring at most three steps for different kinds of users. This kind of procedure simplifies the designing process for the programmer, and ensures the stability and efficiency of control system. Quicker can not only automatically map the physical I/O to a specific Modbus address, but also allows users to define their own variables into it. Therefore users can develop their own application program with eVC++, VB.NET, and VC#.NET programming language via the Modbus RTU and Modbus TCP protocol to share their specific data with Modbus client. Moreover, users can operate the Quicker and NAPOPC in coordination to create a fantastic solution integrating SCADA software with on-line data.

3.3. SendToCOM

The sendToCOM uses the serial port to communicate with expansion module. To use the sendToCOM, you can send data to expansion module through the serial port, and receive data from other device through the serial port.

For more information about these commands for communicating with expansion module, please refer to:

CD:\Napdos\io_module\ 87k_high_profile_modules.htm



3.4. VCEP (Virtual CE Pro)

ICPDAS VCEP is designed for managing your WinPAC-8000 anywhere. No matter where you are, ICPDAS VCEP provides a convenient environment on the Desktop PC and lets you control your WinCon-8000 remotely. ICPDAS VCEP is composed of two main components: The ‘Server’ which runs on WinCon-8000 and the ‘Client’ which runs on a Desktop PC. Once a connection is established between the client and server (initiated by the client), the client will periodically send requests for screen updates and send mouse/key click information to the server to simulate. Each video frame is inter-compressed against the previous frame and then intra-compressed with a modified LZW scheme to minimize the amount of data transmitted from server to client.

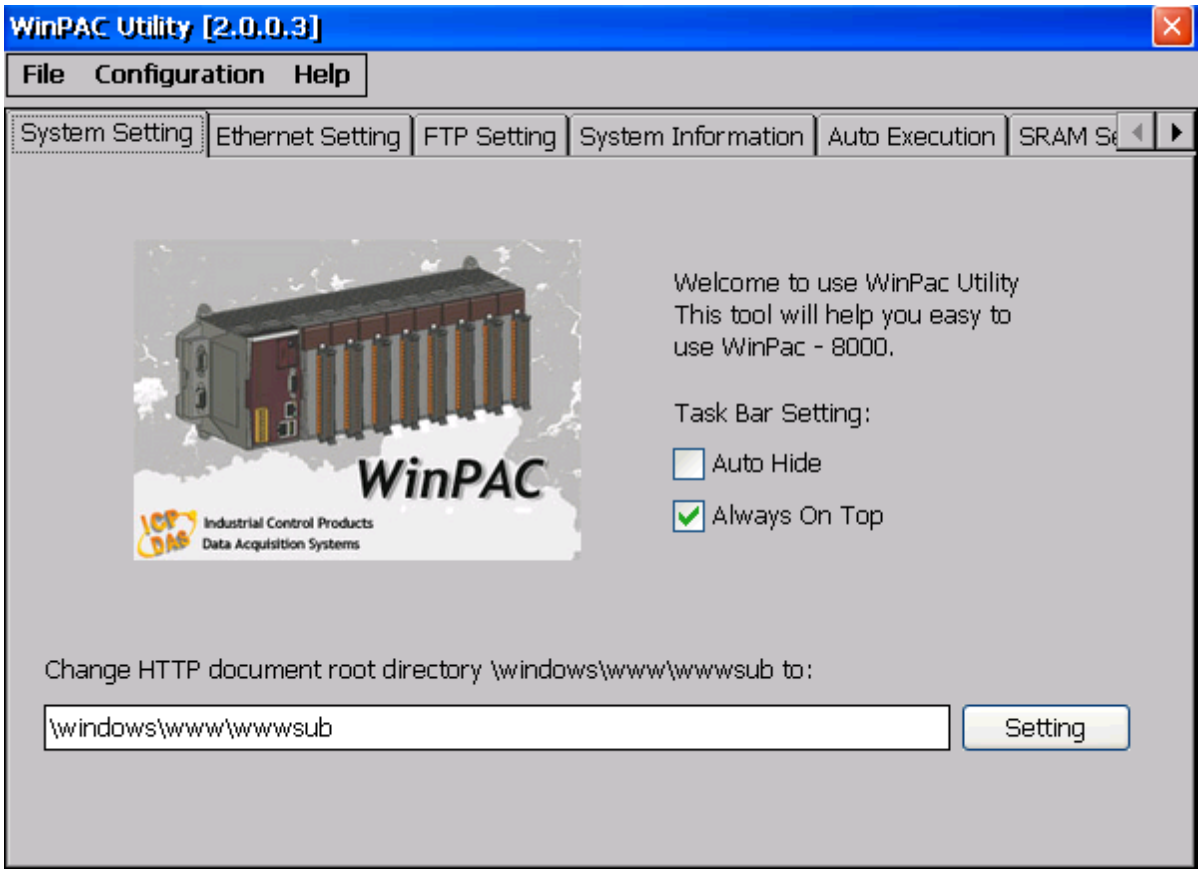
For more detailed information on VCEP application, please refer to

http://www.icpdas.com/products/PAC/wincon-8000/wincon_VirtualCE.htm

3.5. WinPAC Utility

The WinPAC Utility is a tool which is designed to quickly control and management the WinPAC-8000 controller.

For more detailed information on WinPAC Utility applications, please refer to **"2.4. WinPAC Utility for configuring the WinPAC-8000"**



CHAPTER 4

Your first program on the WinPAC-8000

Before writing your first program, ensure that you have the necessary development tools and the corresponding WinPAC SDKs are installed on your system.

4.1. Setting up the development tools

The following tools are available for application developers targeting Windows CE-based WinPAC-8000 controller. One of the following tools must be installed on the Host PC.

- Microsoft eMbedded Visual C++
- Visual Basic.net
- Visual C#

4.2. Installing and updating the WinPAC-8000 SDKs

The WinPAC SDKs are necessary for application developers targeting Windows CE-based WinPAC-8000 controller. The corresponding WinPAC SDKs must be installed on the Host PC.

To install the WinPAC SDKs, please perform the following steps:

Step 1: Insert the CD into your CD-ROM drive

Step 2: Run the "PAC270_SDK_YYYYMMDD.msi" located in

CD:\Napdos\wp-8x4x_ce50\SDK

Step 3: Follow the prompts until the WinPAC SDKs installation process is complete

4.3. API for WinPAC-8000

After installing the WinPAC SDKs, a number of functions can be installed on the Host PC. And this installation puts the header files and libraries into the following public places so they are easily changed by update the WinPAC SDKs.

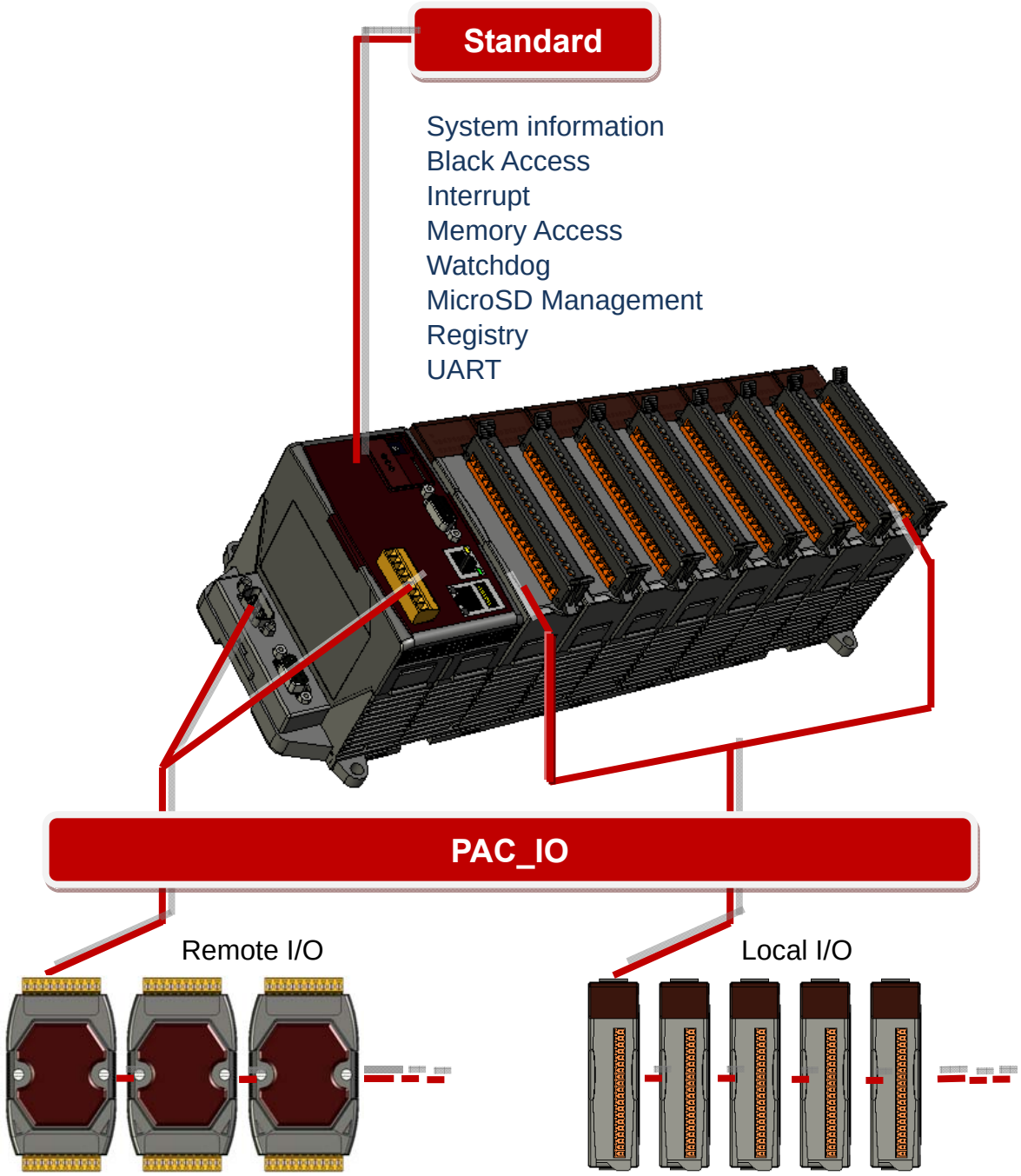
Header files:

C:\Program Files\Windows CE Tools\wce500\PAC270\Include

Libraries:

C:\Program Files\Windows CE Tools\wce500\PAC270\Lib

WinPacSDK Overview



- **System Information Reference**

  Provides reference information for the system status.

- **Backplane Access Reference**

  Provides reference information for the backplane access APIs, including Hot Plug and backplane information.

- **Interrupt Reference**

  Provides reference information for the Interrupt APIs

- **Memory Access Reference**

  Provides reference information for the memory R/W APIs, including EEPROM and SRAM.

- **Watchdog Reference**

  Provides reference information for the watchdog APIs, including hardware watchdog and OS watchdog.

- **Uart Reference**

  Provides reference information for the Uart APIs.

- **Registry Reference**

  Provides reference information for the registry.

- **MicroSD Management Reference**

  Provides reference information for the MicroSD Manager.

- **PAC_IO Reference**

  Provides reference information for IO APIs, including local and remote.

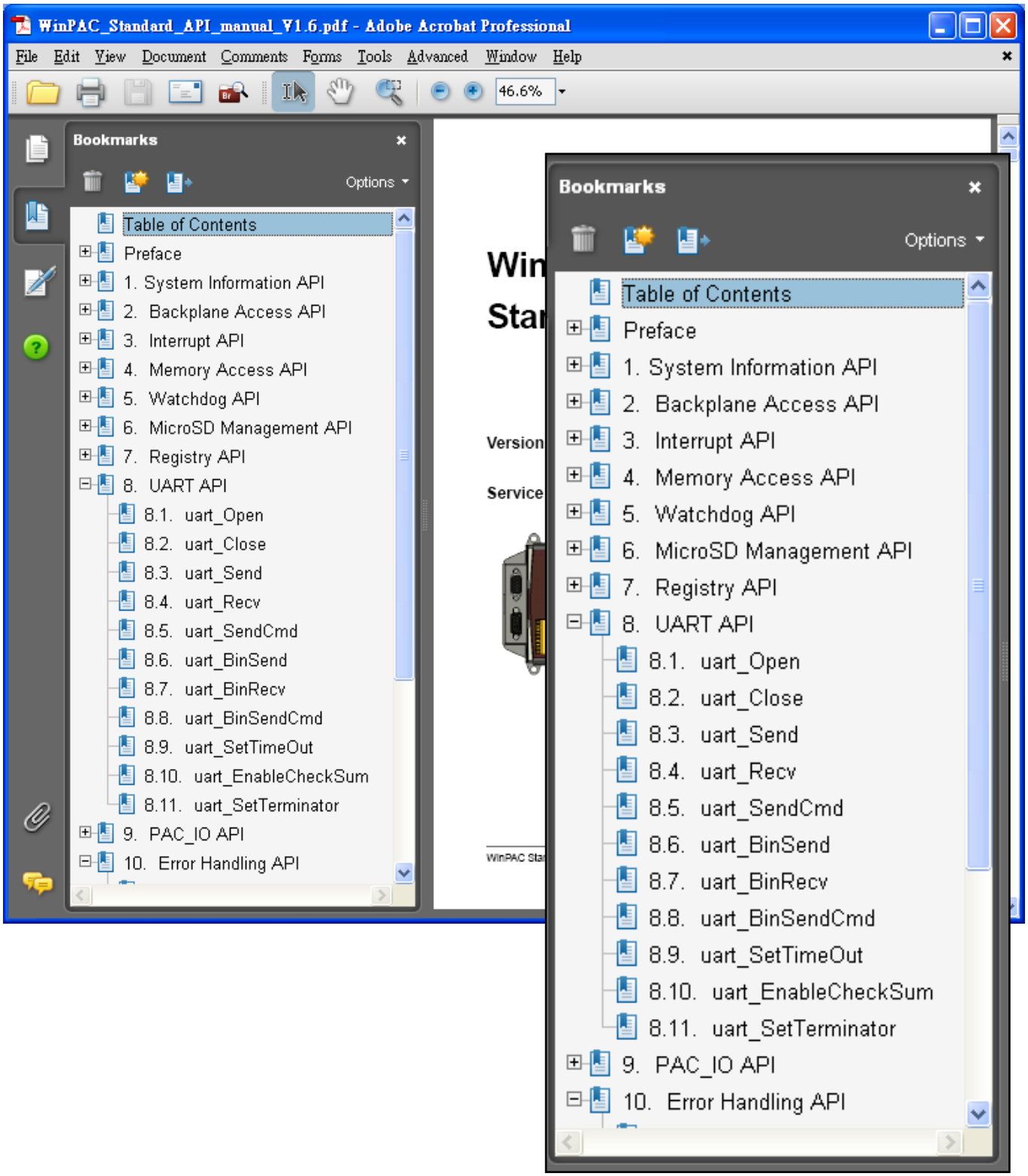
- **Error Handling API Reference**

  Provides reference information for error handling.

For full usage information regarding the description, prototype and the arguments of the functions, please refer to the “WinPAC Standard API Manual” located at:

CD:\Napdos\wp-8x4x_ce50\Document\SDK_Document\

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/document/sdk_document/winpac_standard_api_manual_v1.3.pdf



4.4. Your first program with eMbedded Visual C++

To create a demo program with eMbedded Visual C++ development tool includes the following main steps:

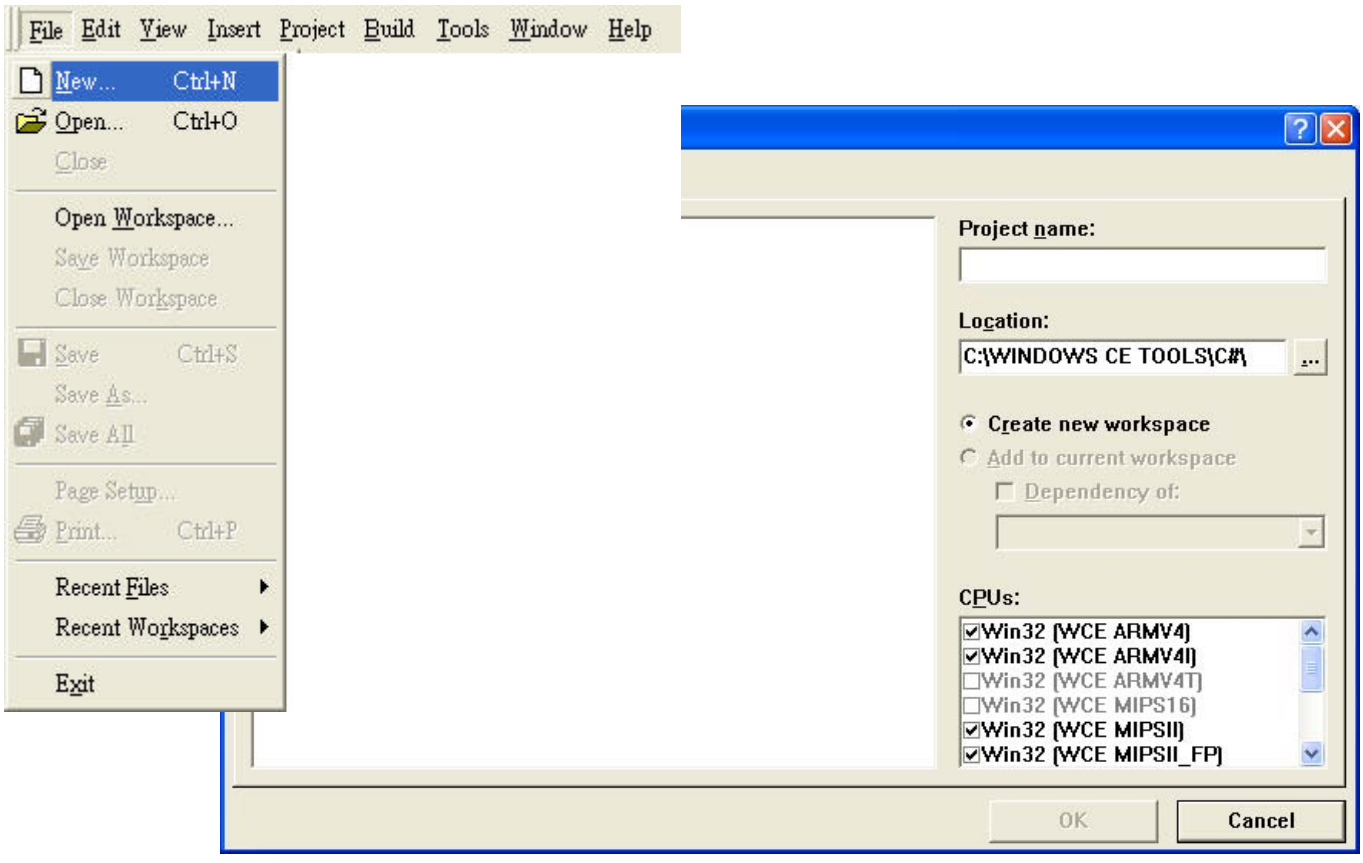
1. Create a new Forms-Based project
2. Configure compiler options
3. Design and Build an application program
4. Execute the application on the WinPAC-8000

All main steps will be described in the following subsection.

4.4.1. Create a new Forms-Based project

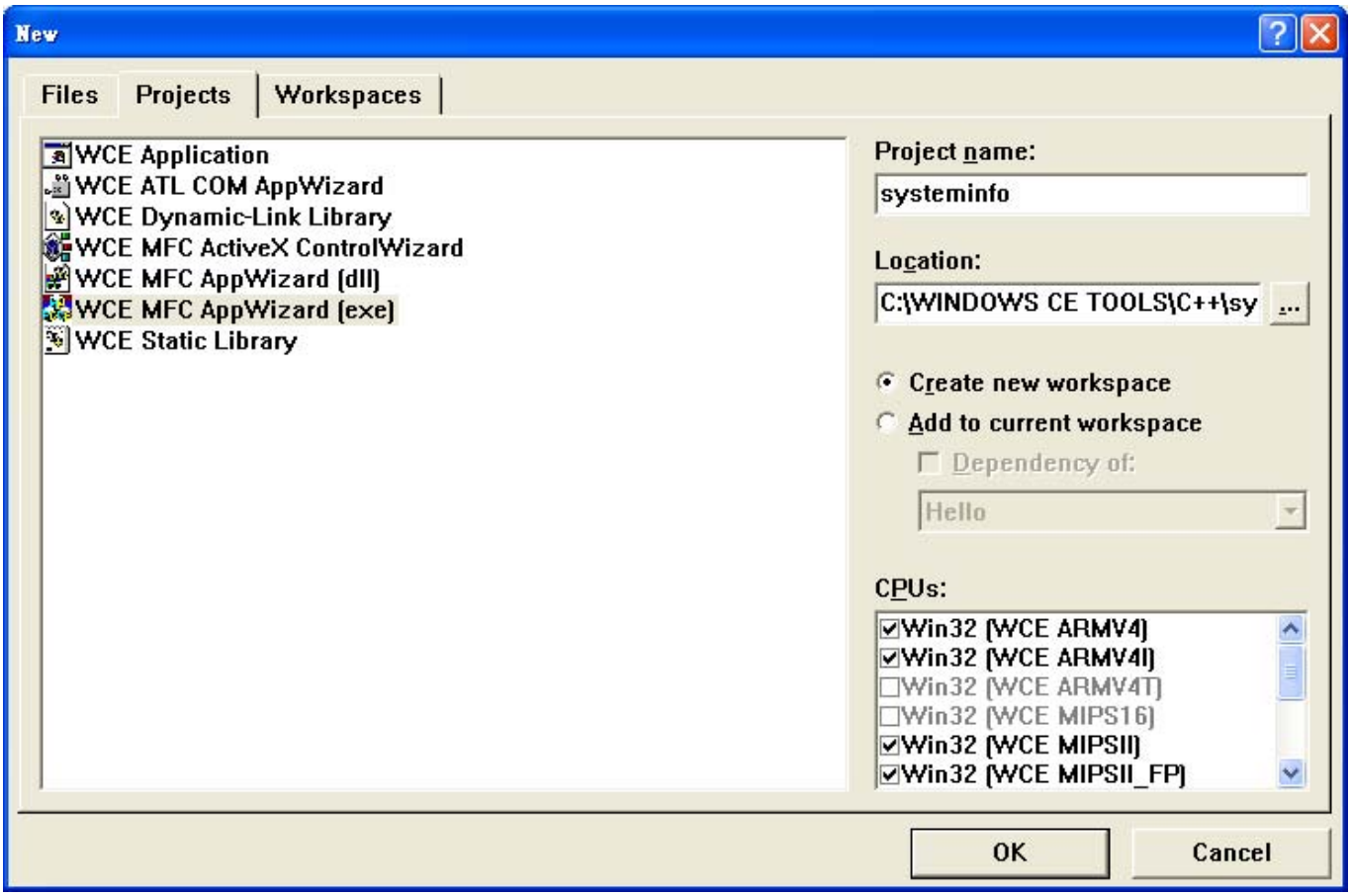
Step 1: Start the Microsoft Embedded Visual C++

Step 2: From the "File" menu, click the "New" command

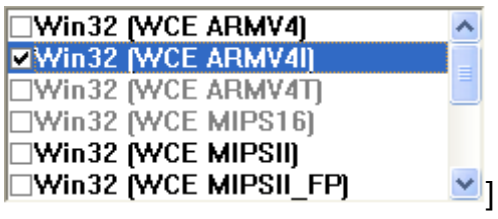


Step 3: In the “New” dialog, select the “Projects” tab and do the following in this order

- Select “WCE MFC AppWizard [exe]” from the list
- Type “systeminfo” in the “Project name” edit box
- Specify the directory in the “Location” field where you want to create the project
- In the “CPUs” list, select “Win32 [WCE ARMV4I]” and other options if necessary

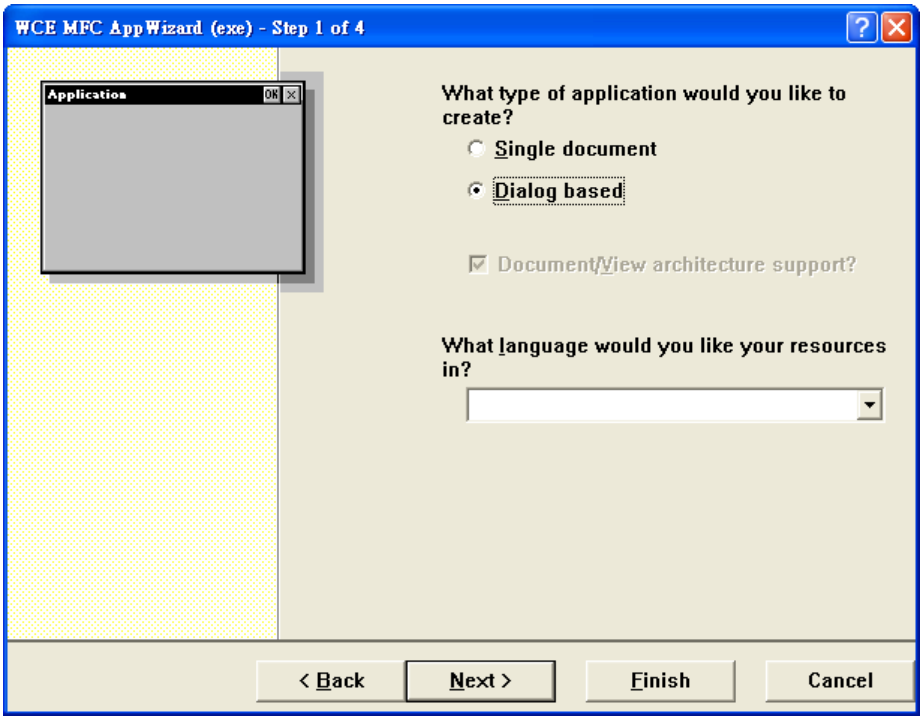


The selected CPU type must have “Win32 [WCE ARMV4I]”

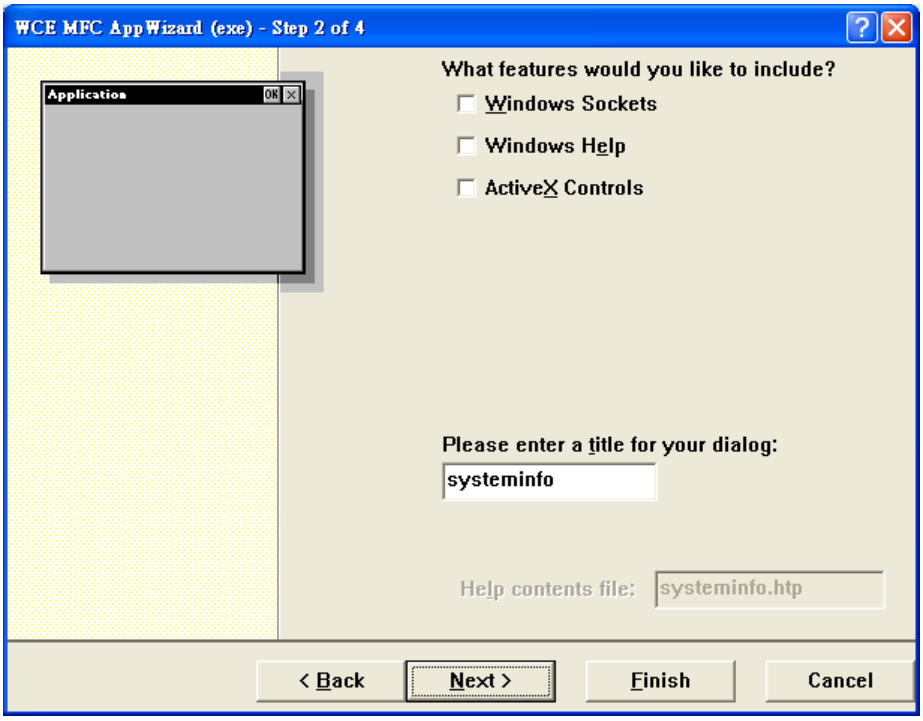


Step 4: Click the “OK” button to start the wizard

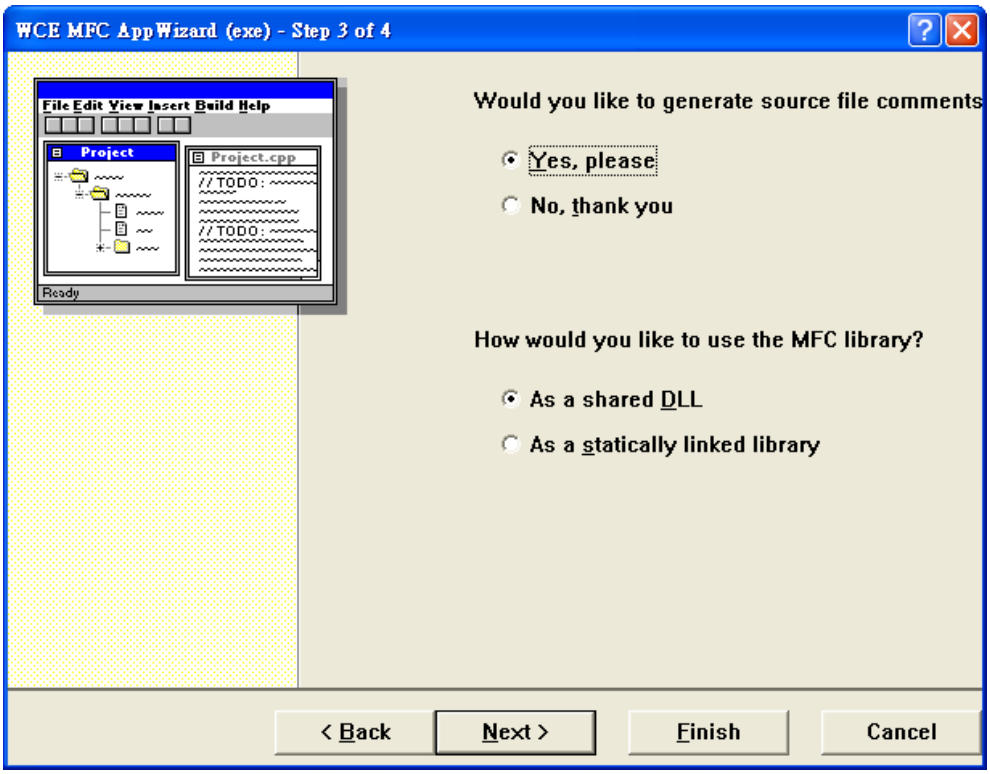
Step 5: On the first page of the wizard, select “Dialog based” option and then click the “Next” button to the next step



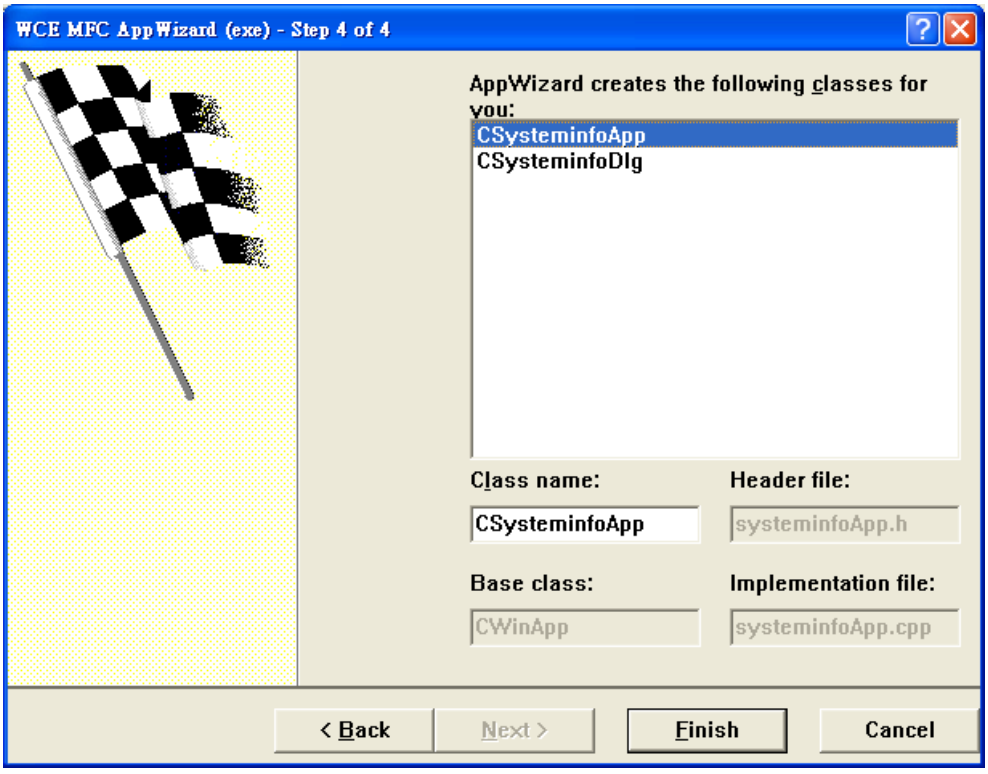
Step 6: On the next page of the wizard, leave all the options as they are, and then click the “Next” button to the next step



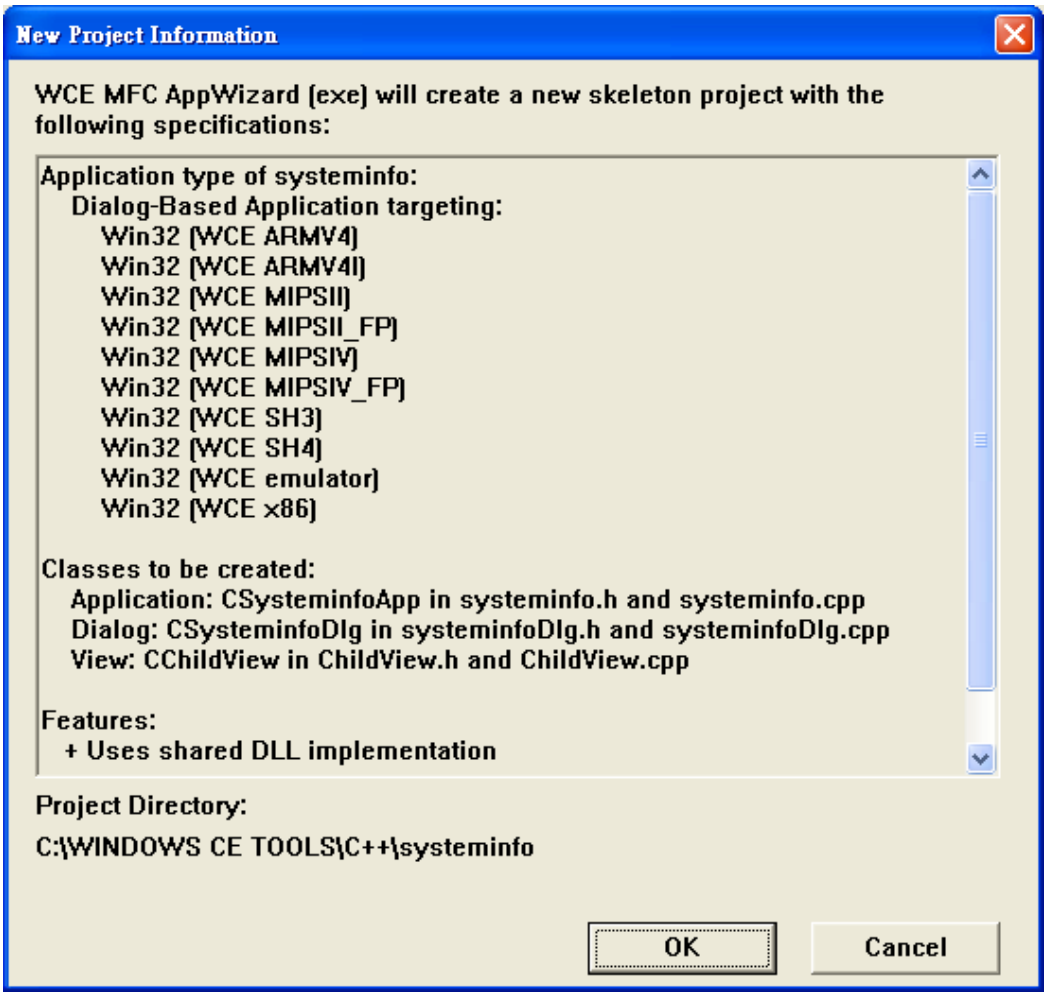
Step 7: On the next page of the wizard, leave all the options as they are, and then click the "Next" button to the next step



Step 8: On the next page of the wizard, leave all the options as they are, and then click the "Finish" button to the next step

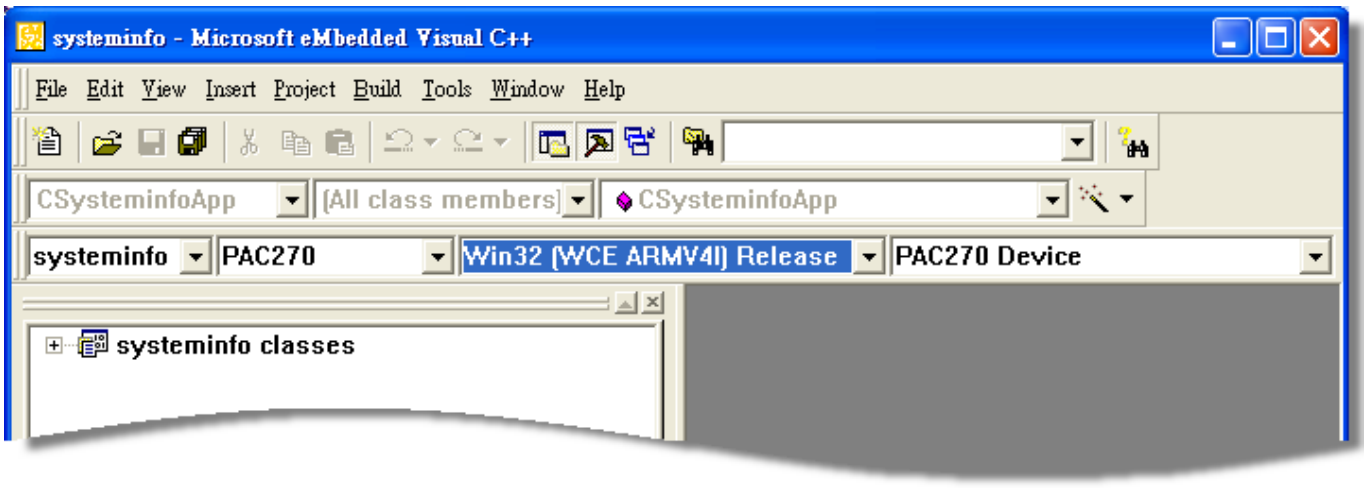


Step 9: The final summary appears, click the "OK" button to complete the wizard



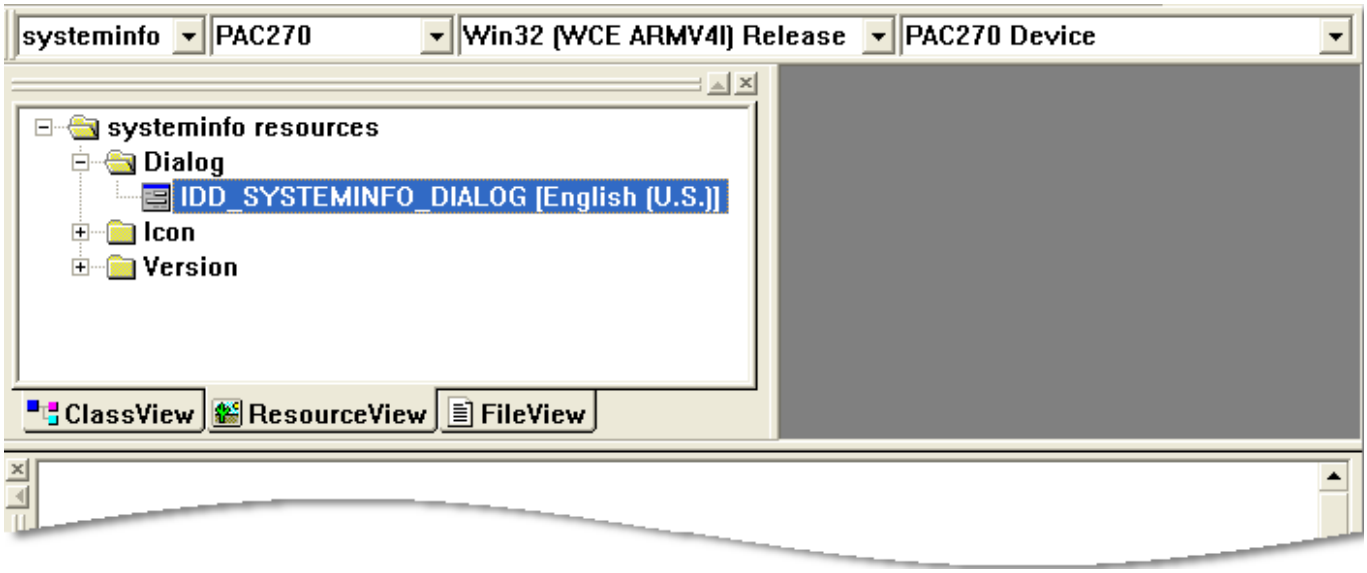
4.4.2. Configure compiler options

On the WCE configuration toolbar, select the "Win32 [WCE ARMV4] Release"

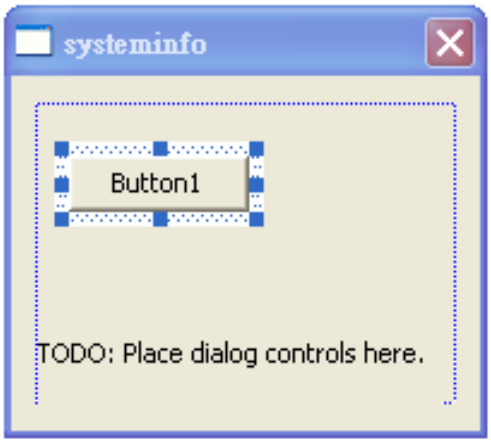


4.4.3. Design and Build an application program

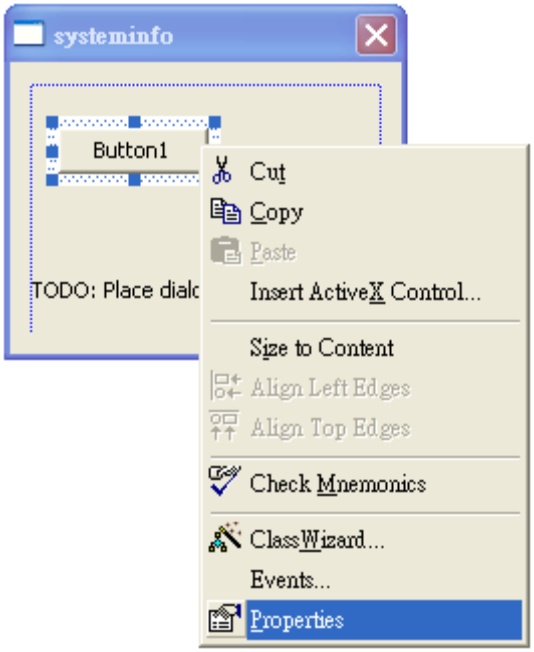
Step 1: On the "Workspace" window, select the "ResourceView" tab and expand the "dialog" folder, and then double-click the "IDD_DEMO_DIALOG" to open the dialog box



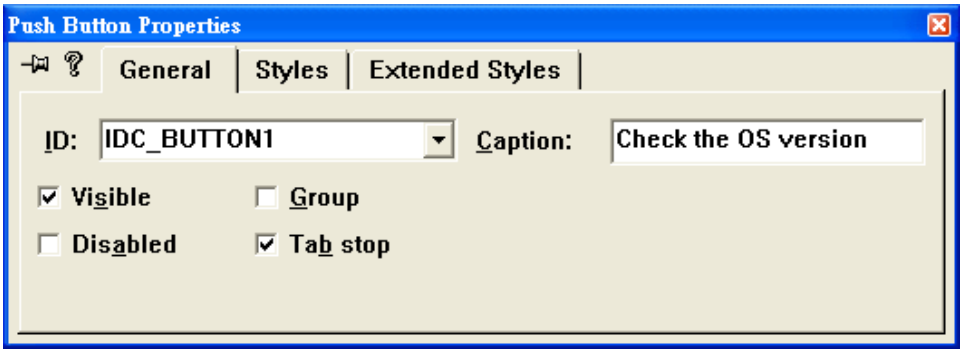
Step 2: Add the "  button" object in the "systeminfo" dialog box



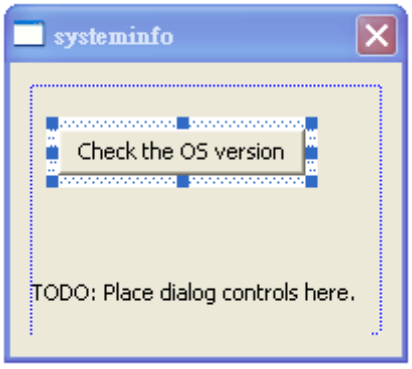
Step 3: In the "systeminfo" dialog box, right-click the button object and then click the "Properties" command



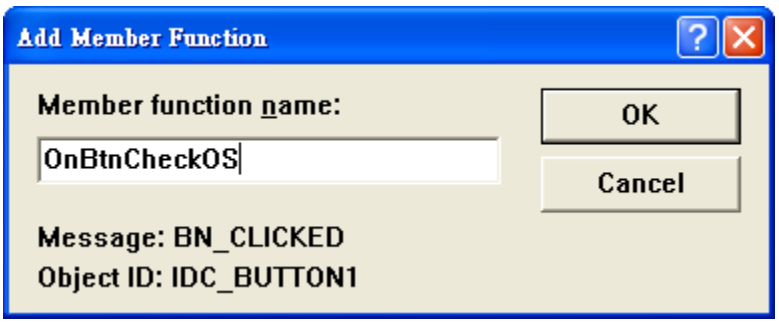
Step 4: Type the "Check the OS version" in the "Caption" edit box and then click the "  close" button



Step 5: In the "systeminfo" dialog box, double-Click the button object



Step 6: Type the "OnBtnCheckOS" in the "Member function name" edit box and then click the "OK" button



Step 7: Insert the following code into the Editor Window

```
char OS[32];
TCHAR buf[32];
pac_GetOSVersion(OS);
pac_AnsiToWideString(OS, buf);
MessageBox(buf,0,MB_OK);
```

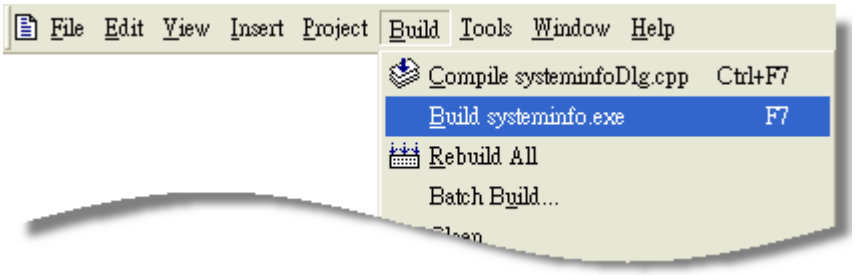
```
void CCheckosDlg::OnBtnCheckOS()
{
    // TODO: Add your control notification handler code here
    char OS[32];
    TCHAR buf[32];
    pac_GetOSVersion(OS);
    pac_AnsiToWideString(OS,buf);
    MessageBox(buf,0,MB_OK);
}
```

Step 8: Insert the "#include "WinpacSDK.h"" into the header area

```
// HelloDlg.cpp : implementation file
//

#include "stdafx.h"
#include "Hello.h"
#include "HelloDlg.h"

#include "WinPacSDK.h"
```

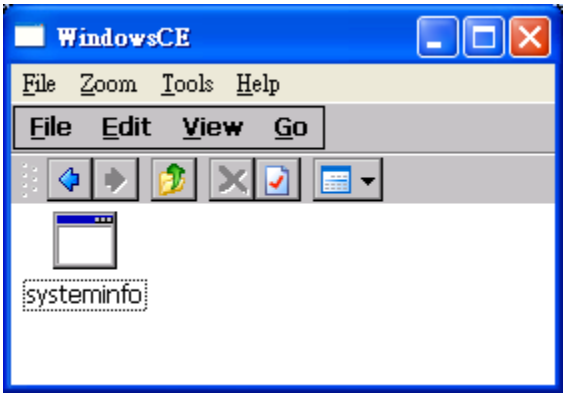
4.4.4. Execute the application program on the WinPAC-8000

Step 1: On the "Build" menu, click the "Build systeminfo.exe" command

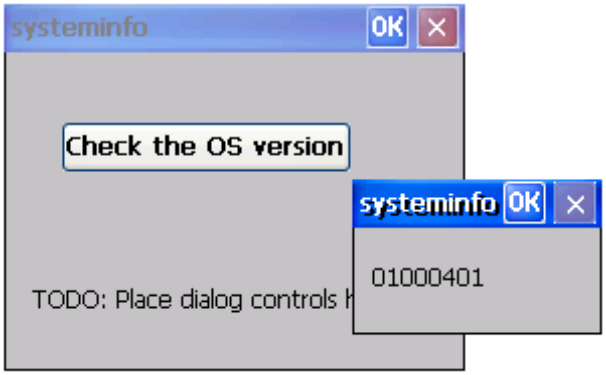


Step 2: Open the web browser and type the IP address to connect the FTP server of WinPAC-8000

Step 3: Upload the "systeminfo.exe" application to the WinPAC-8000 via the WinPAC FTP server



Step 4: On the WinPAC-8000, execute the uploaded file



4.5. Your first program with C#

To create a demo program with C# development tool includes the following main steps:

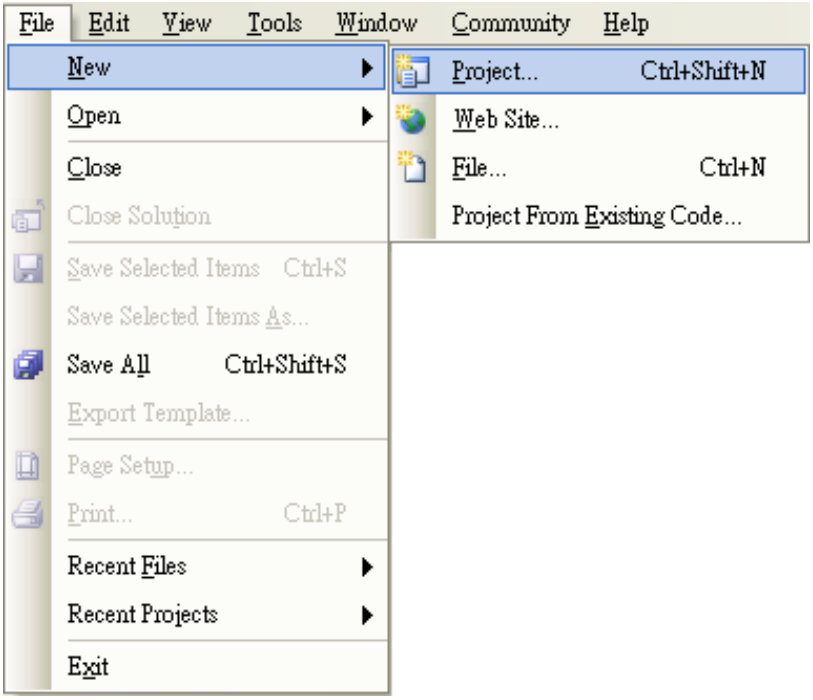
1. Create a new project
2. Add project reference for an application
3. Design and Build an application program
4. Execute the application on the WinPAC-8000

All main steps will be described in the following subsection.

4.5.1. Create a new project

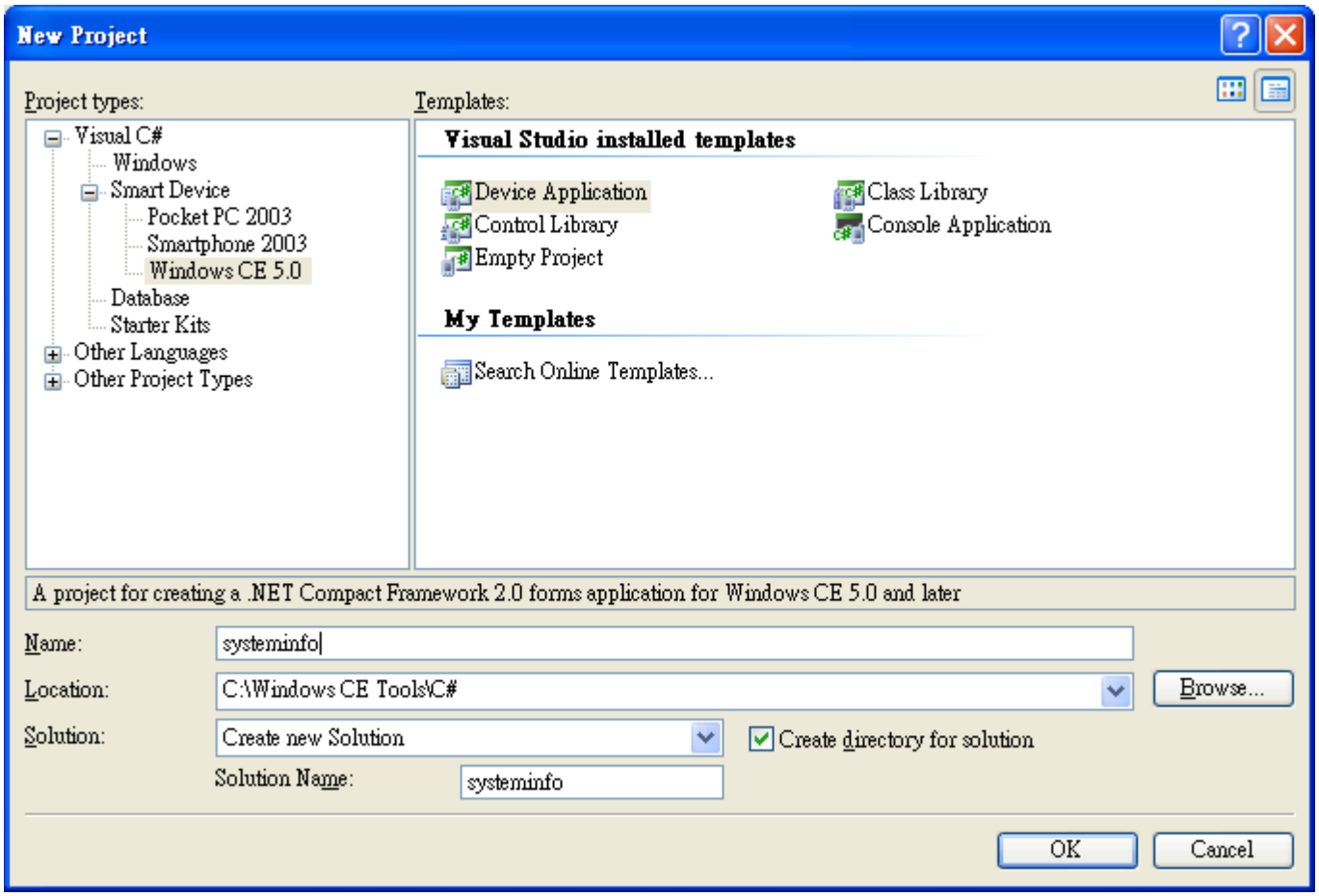
Step 1: Start Visual Studio 2005

Step 2: On the "File" menu, select the "New" command, and then click the "Project" command



Step 3: In the “New Project” dialog box do the following in this order

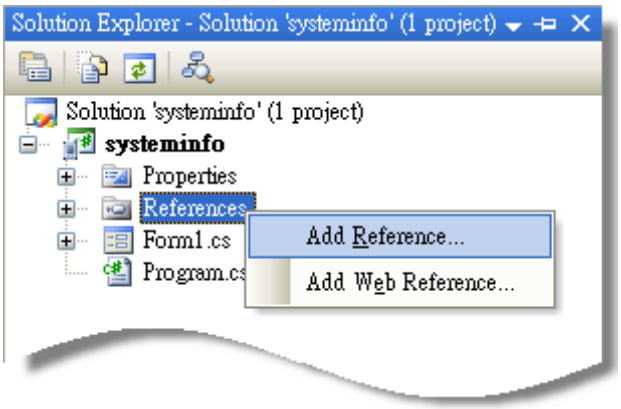
- In the “Project types” list, select the “Visual C#” and expand the “Smart Device” folder, and then select the “Windows CE 5.0”
- Select “Device Application” from the “Templates” list
- Type “systeminfo” in the “Name” edit box
- Specify the directory in the “Location” field where you want to create the project



Step 4: Click OK to start creating a “systeminfo” project

4.5.2. Add project reference for an application

Step 1: On the “Solution Explorer” window, right-click the “Reference” and then click the “Add Reference…” command

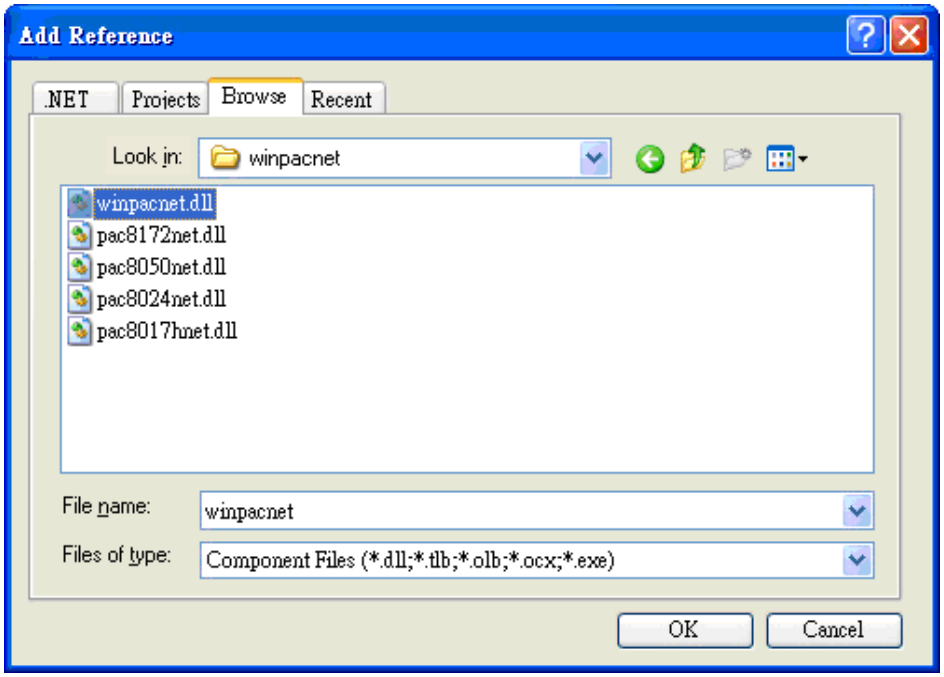


Step 2: In the “Add Reference” dialog box, select the “Browse” tab, and then specify the directory of the “WinPacNet.dll” file in the “File name” field

The “WinPacNet.dll” file can be obtained from:

CD:\Napdos\wp-8x4x_ce50\SDK\WinPacNet\

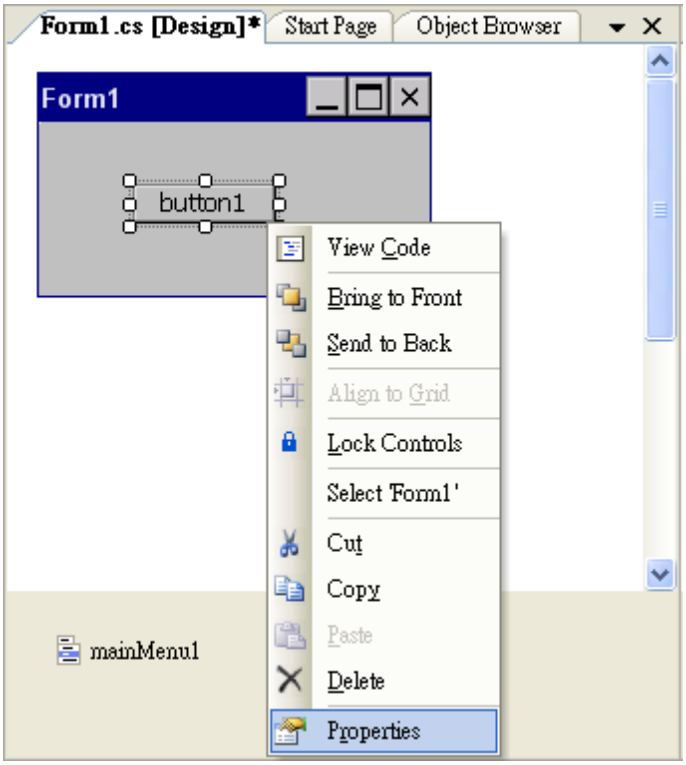
ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/sdk/WinPacNet/



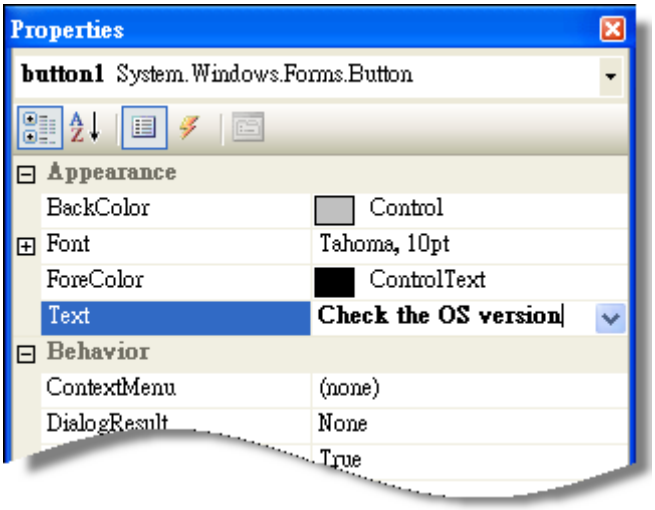
4.5.3. Design and Build an application program

Step 1: Add a “ Button button” object in the “From1” dialog box

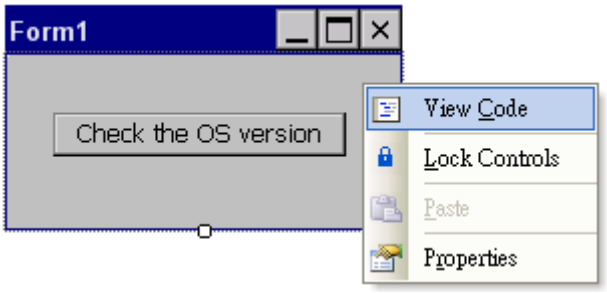
Step 2: Right-click the “button” object and click the “Properties” command



Step 3: On the “Properties” window, type “Check the OS version” in the “Text” edit box



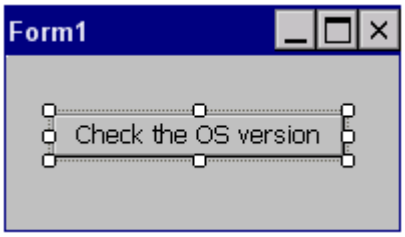
Step 4: Right-click the “Form1” dialog box and click the “View Code” command to open the editor window



Step 5: Insert the “using winpacnet;” into the header area after “using System.Windows.Forms;”

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using WinPacNet;

namespace test2
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
    }
}
```

Step 6: In the “Form1” dialog box, double-click the “button” object to open the editor window



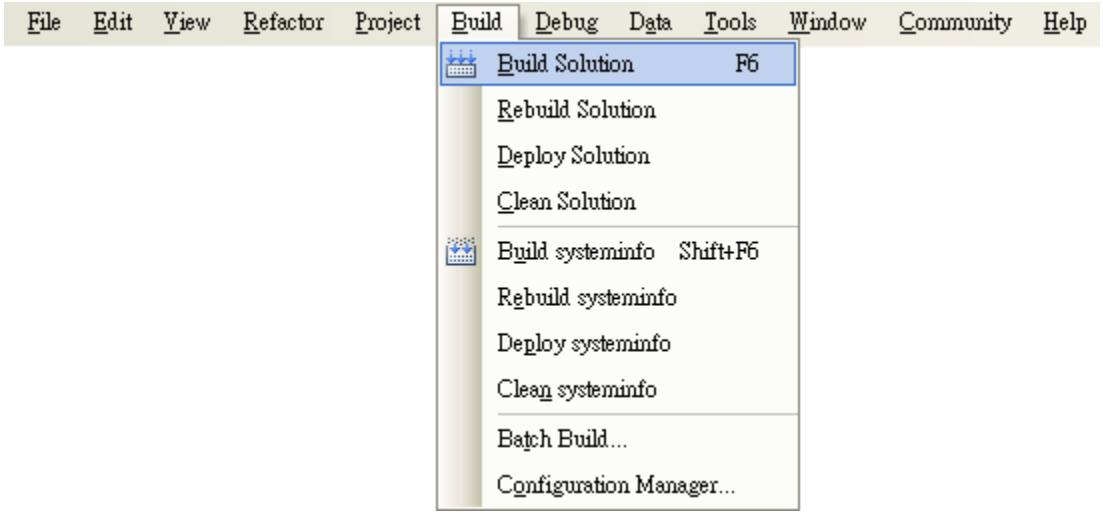
Step 7: Insert the following code in the Editor Window

MessageBox.Show(WinPAC.pac_GetOSVersion());

```
namespace systeminfo
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

        }

        private void button1_Click(object sender, EventArgs e)
        {
            MessageBox.Show(WinPAC.pac_GetOSVersion());
        }
    }
}
```

4.5.4. Execute the application on the WinPAC-8000

Step 1: On the "Build" menu, click the "Build Solution" command



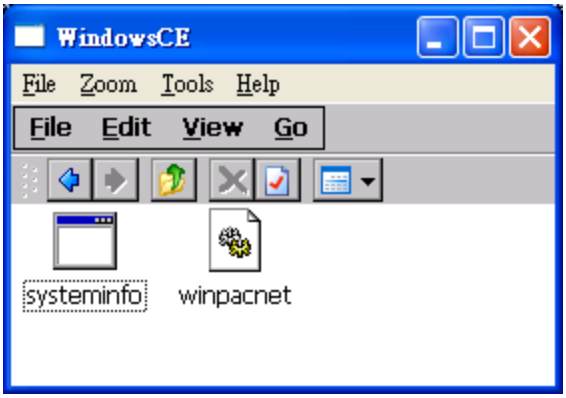
Step 2: Open the web browser and type the IP address to connect the FTP server of WinPAC-8000

Step 3: Upload the "systeminfo.exe" application and the corresponding "winpacnet.dll" file to the WinPAC-8000 via the WinPAC FTP server

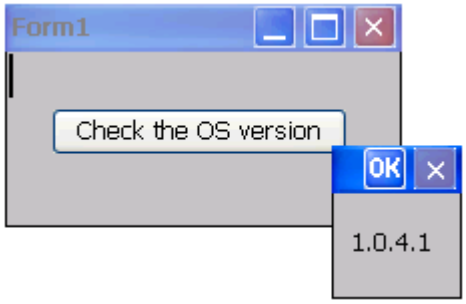
---

For applications programming in C# and VB.net with .net framework, when executing these application on the WinPAC-8000 controller, the corresponding "winpacnet.dll" file must be in the same directory as the .exe file

---



Step 4: On the WinPAC-8000, execute the uploaded file



4.6. Your first program with VB.net

To create a demo program with VB.net development tool includes the following main steps:

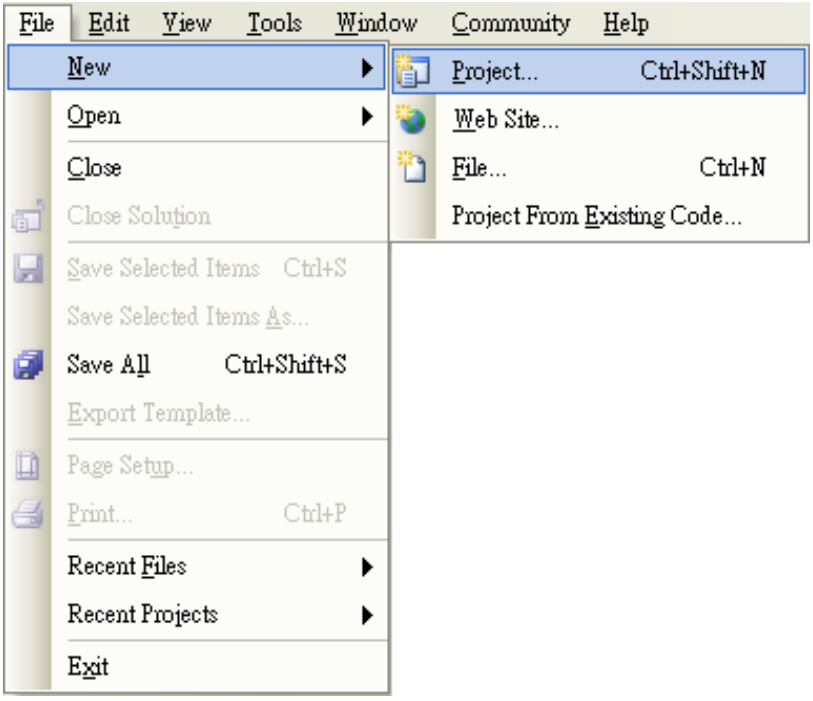
1. Create a new project
2. Add project reference for an application
3. Design and Build an application program
4. Execute the application on the WinPAC-8000

All main steps will be described in the following subsection.

4.6.1. Create a new project

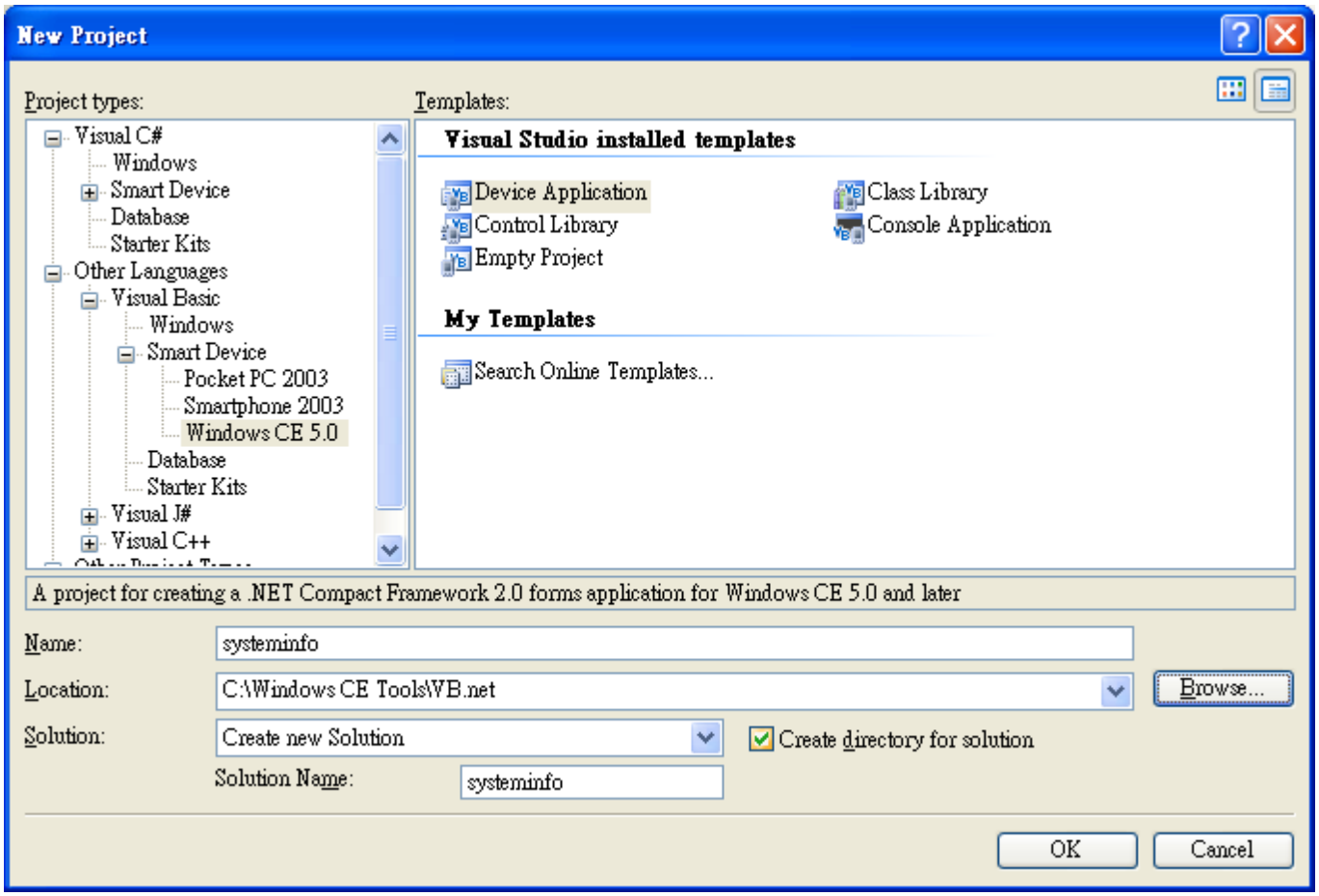
Step 1: Start Visual Studio 2005

Step 2: On the "File" menu, select the "New" command, and then click the "Project" command



Step 3: In the “New Project” dialog box do the following in this order

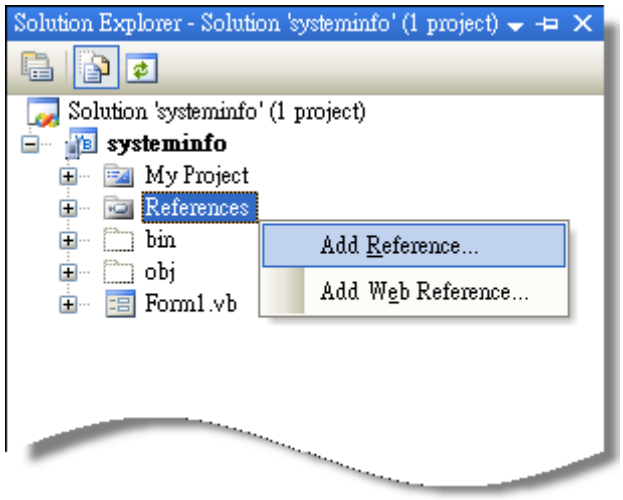
- In the “Project types” list, select the “Other Languages” and expand the “Visual Basic” and the “Smart Device” folder, and then select the “Windows CE 5.0”
- Select the “Device Application” option from the “Templates” list
- Type “systeminfo” in the “Name” entry box
- Specify the directory in the “Location” field where you want to create the project



Step 4: Click OK to start creating a “systeminfo” project

4.6.2. Add project reference for an application

Step 1: On the “Solution Explorer” window, right-click the “Reference” and then click the “Add Reference…” command

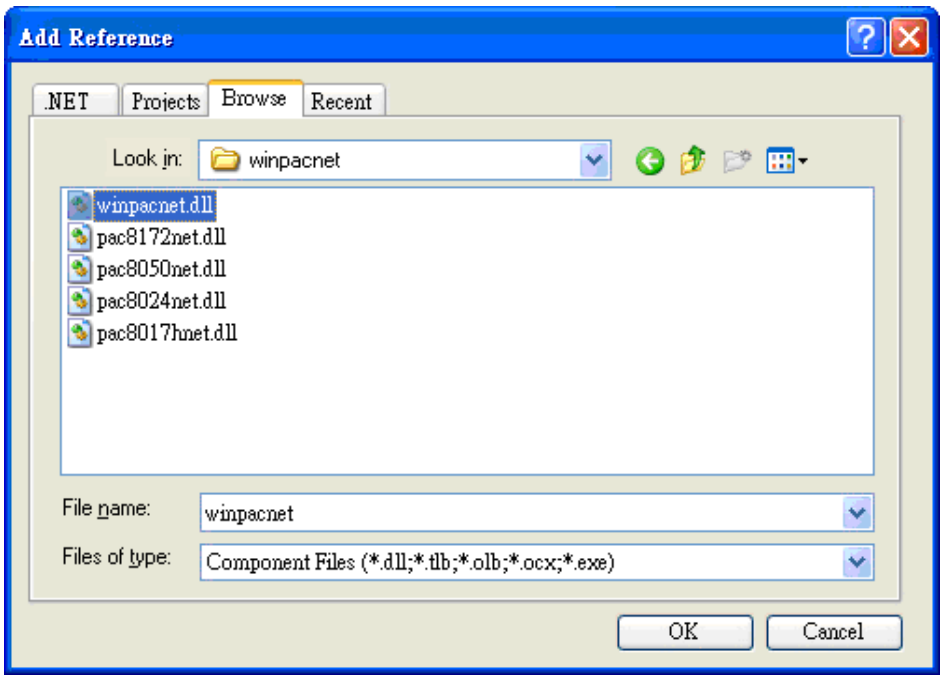


Step 2: In the “Add Reference” dialog box, select the “Browse” tab, and then specify the directory of the “WinPacNet.dll” file in the “File name” field

The “WinPacNet.dll” file can be obtained from:

CD:\Napdos\wp-8x4x_ce50\SDK\WinPacNet\

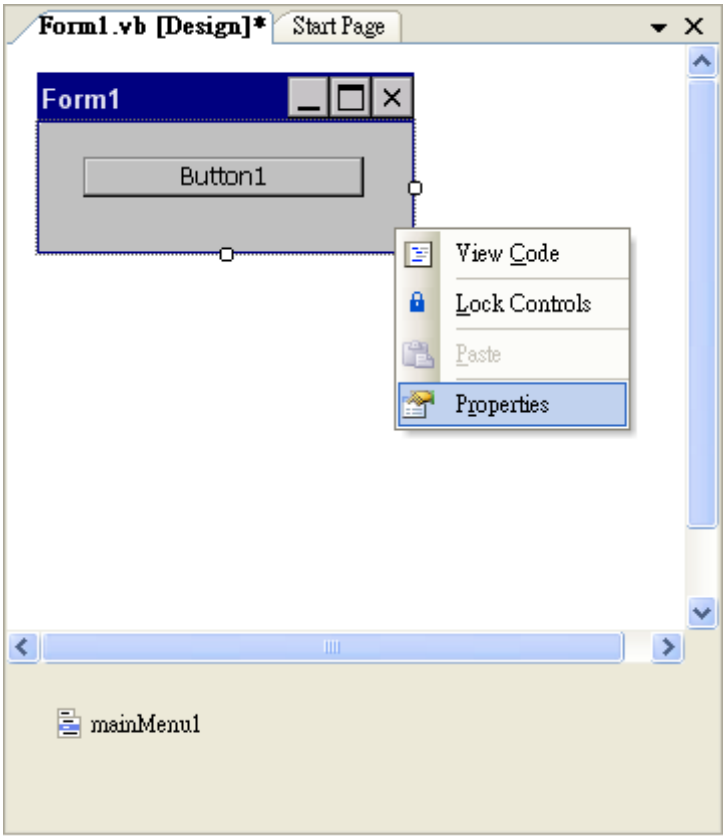
ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/sdk/winpacnet/



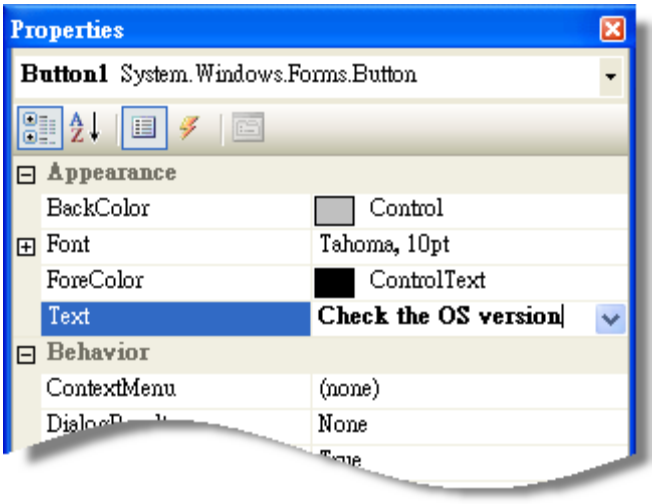
4.6.3. Design and Build an application program

Step 1: Add a “ Button button” object in the “From1” dialog box

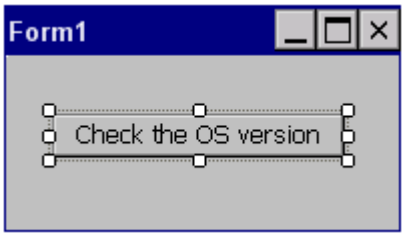
Step 2: Right-click the “button” object and click the “Properties” command



Step 3: On the “Properties” window, type “Check the OS version” in the “Text” edit box

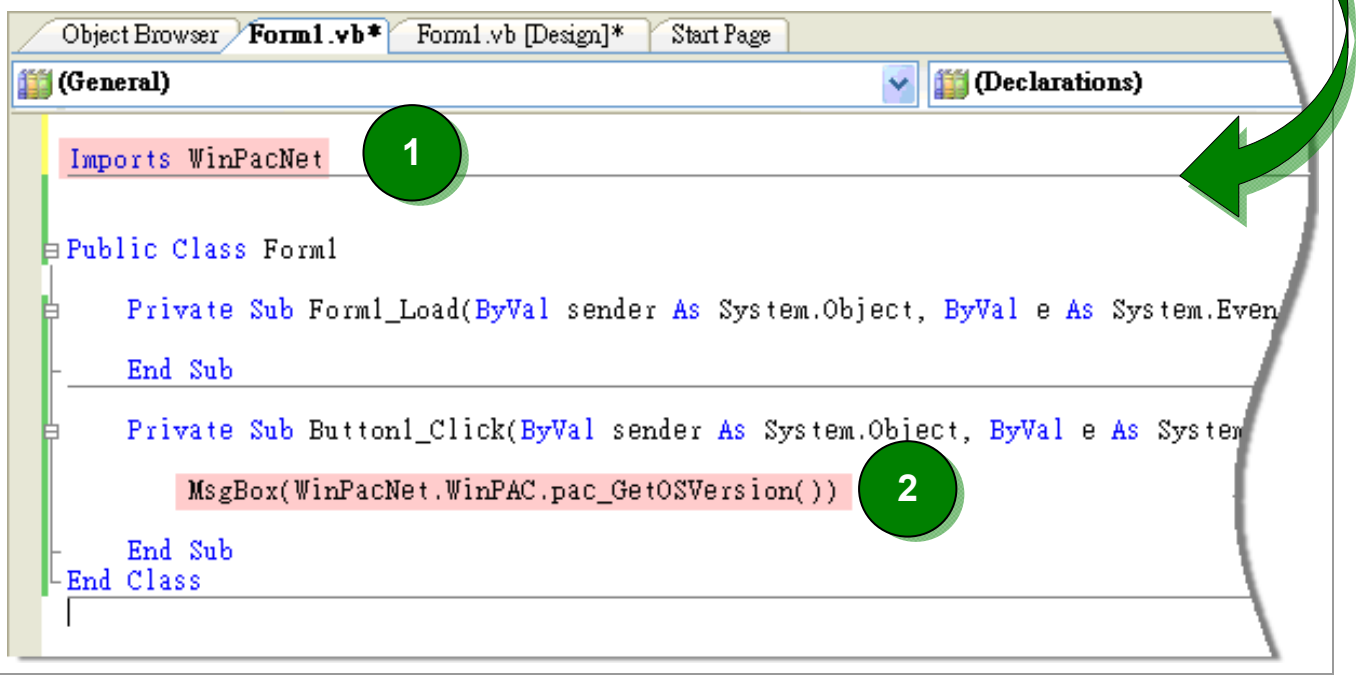


Step 4: In the "Form1" dialog, double-click the button object to open the editor window



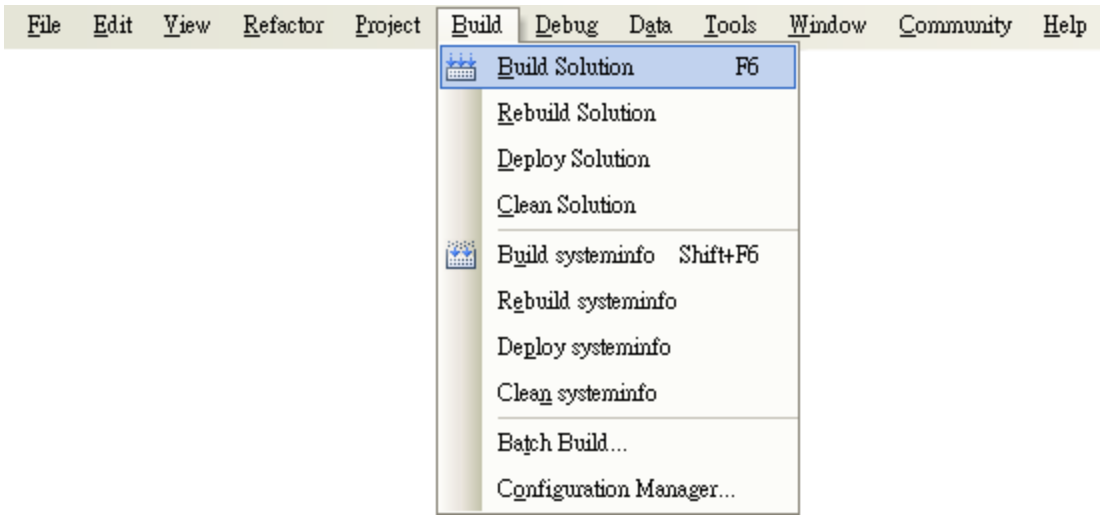
Step 5: Insert the following code in the Editor Window

1. Imports WinPacNet
2. MessageBox(WinPacNet.WinPAC.pac_GetOSVersion())



4.6.4. Execute the application on the WinPAC-8000

Step 1: On the "Build" menu, click the "Build Solution" command



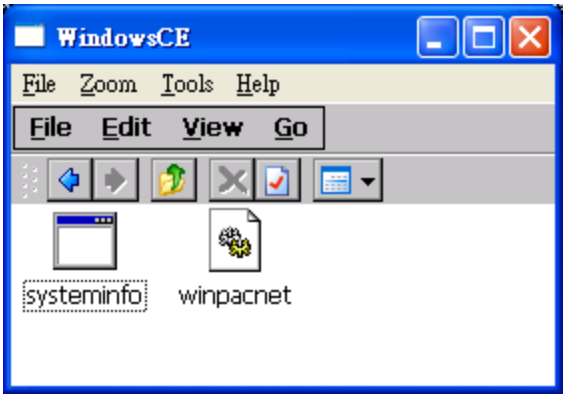
Step 2: Open the web browser and type the IP address to connect the FTP server of WinPAC-8000

Step 3: Upload the "systeminfo.exe" application and the corresponding "winpacnet.dll" file to the WinPAC-8000 via the WinPAC FTP server

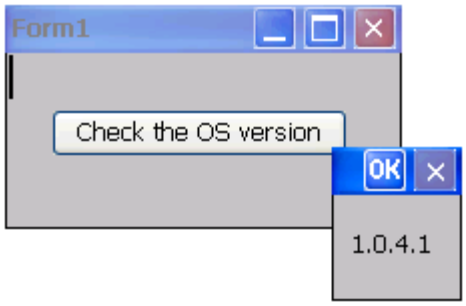
---

For applications programming in C# and VB.net with .net framework, when executing these application on the WinPAC-8000 controller, the corresponding "winpacnet.dll" file must be in the same directory as the .exe file

---



Step 4: On the WinPAC-8000, execute the uploaded file



CHAPTER

5

API and demo reference

There are several demo programs that have been designed for your controller. You can examine the demo source code, which includes numerous comments, to familiarize yourself with the WinPAC API, This will allow to develop your own applications quickly by modifying these demo programs. The following details the contents of the WinPAC-8000 demo programs.

Standard API:

eVC:

Folder	Demo	Explanation
system	systeminfo	Retrieves information about the OS version, CPU version, SDK version, etc.
backplane	backplaneinfo	Retrieves information about the DIP switch, backplane ID and slot count.
	Interrupt	Shows how to gain the interrupt advantage
memoryaccess	memory	Shows how to read/write date values from/to EEPROM
watchdog	watchdog	Displays how the watchdog operate
microsd	microsd	Shows how to enables/disables Micro SD
registry	registry	Shows how to read/write date values from/to registry
uart	diag	Shows how to read the name of local I/O modules via UART

These demo programs can be obtained from:

CD:\Napdos\wp-8x4x_ce50\Demo\WinPAC\eVC\Standard\

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/evc/standard/

dotnet (VB.net and C#) :

Folder	Demo	Explanation
system	systeminfo	Retrieves information about the OS version, CPU version, SDK version, etc.
backplane	backplaneinfo	Retrieves information about the DIP switch, backplane ID and slot count.
memoryaccess	memory	Shows how to read/write date values from/to EEPROM
	battery_backup_sram	Shows how to read or write to the battery backup
watchdog	watchdog	Displays how the watchdog operate
microsd	microsd_management	Shows how to enables/disables Micro SD
registry	registry	Shows how to read/write date values from/to registry
uart	diag	Shows how to read the name of local I/O modules via UART

For VB.net application, these demo programs can be obtained from:

CD:\Napdos\wp-8x4x_ce50\Demo\WinPAC\DotNET\VB.NET\Standard

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/dotnet/vb.net/standard/

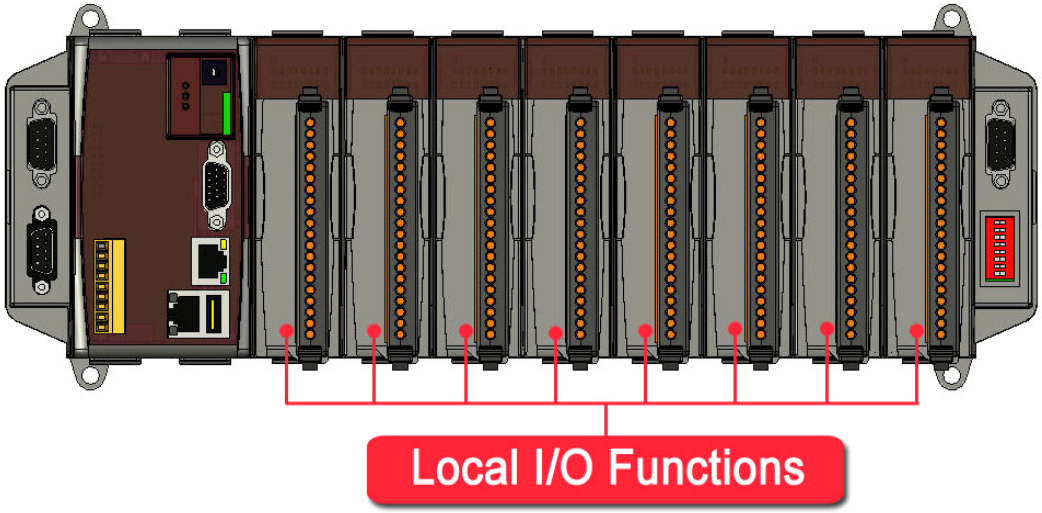
For C# application, these demo programs can be obtained from:

CD:\Napdos\wp-8x4x_ce50\Demo\WinPAC\DotNET\C#\Standard

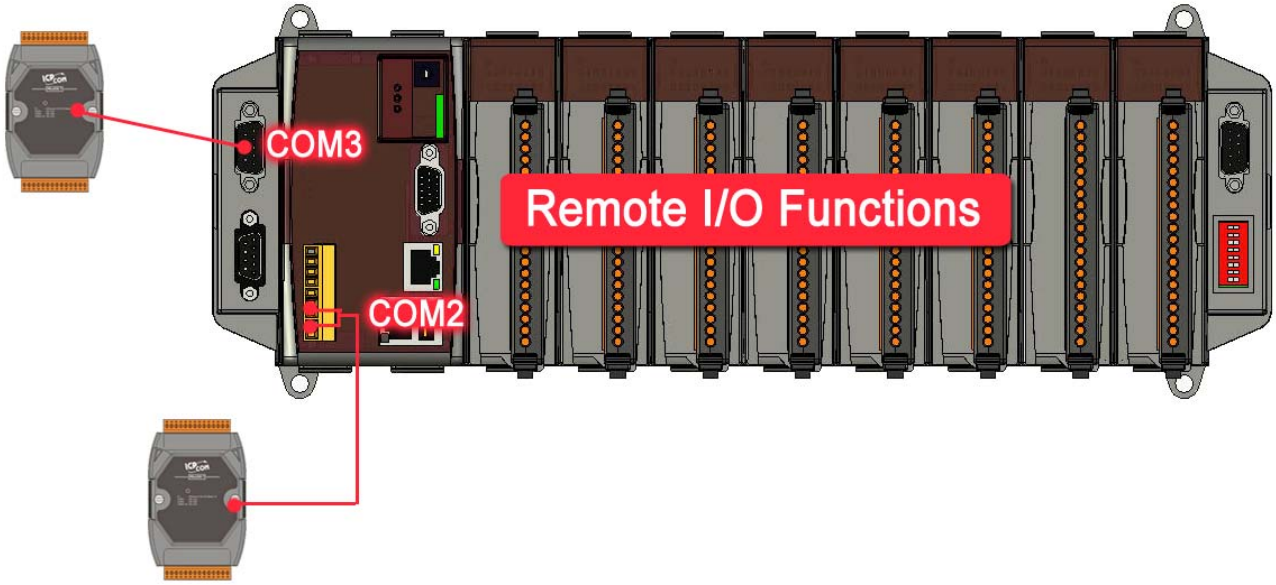
ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/dotnet/c#/standard/

PAC_IO:

Local (IO in slot):



Remote:



eVC:

Folder	Demo	Explanation
Local	find_io	Shows how to retrieve the module names and types which plugged in the WinPAC-8000.
	8k_di	Shows how to read the DI values of DI module. This demo program is used by 8K series DI modules.
	8k_do	Shows how to write the DO values to DO module. This demo program is used by 8K series DO modules.
	8k_dio	Shows how to read the DI and the DO values of the DIO module. This demo program is used by 8K series DIO modules.
	87k_basic	Shows how to send/receive a command/response application. This demo program is used by 87K series modules.
	87K_demo	Shows how use uart API and the IO modules located as slots. This demo program is used by 87K series modules.
	87k_ai	Shows how to read the AI values of AI module. This demo program is used by 87K series AI modules.
	87k_ao	Shows how to write the AO values to AO module. This demo program is used by 87K series AO modules.
	87k_di	Shows how to read the DI values of DI module. This demo program is used by 87K series DI modules.
	87k_do	Shows how to write the DO values to DO module. This demo program is used by 87K series DO modules.
	87k_dio	Shows how to read the DI and the DO values of the DIO module. This demo program is used by 87K series DIO modules.
Remote	7k87k_basic	Shows how to send/receive a command/response application. This demo program is used by 7K or 87K series

Folder	Demo	Explanation
		AI modules which connected through a COM port.
	7k87k_ai	Shows how to read the AI values of AI module. This demo program is used by 7K or 87K series AI modules which connected through a COM port.
	7k87k_ao	Shows how to write the AO values to AO module. This demo program is used by 7K or 87K series AI modules which connected through a COM port.
	7k87k_di	Shows how to read the DI values of DI module. This demo program is used by 7K or 87K series AI modules which connected through a COM port.
	7k87k_do	Shows how to write the DO values to DO module. This demo program is used by 7K or 87K series AI modules which connected through a COM port.
	7k87k_dio	Shows how to read the DI and the DO values of the DIO module. This demo program is used by 7K or 87K series AI modules which connected through a COM port.

These demo programs can be obtained from:

CD:\Napdos\wp-8x4x_ce50\Demo\WinPAC\eVC\PAC_IO\

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/evc/pac_io/

dotnet:

Folder	Demo	Explanation
Local	find_io	Shows how to retrieve the module names and types which plugged in the WinPAC-8000.
	8k_di	Shows how to read the DI values of DI module. This demo program is used by 8K series DI modules.
	8k_do	Shows how to write the DO values to DO module. This demo program is used by 8K series DO modules.
	8k_dio	Shows how to read the DI and the DO values of the DIO module. This demo program is used by 8K series DIO modules.
	87k_basic	Shows how to send/receive a command/response application. This demo program is used by 87K series modules.
	87K_demo	Shows how use uart API and the IO modules located as slots. This demo program is used by 87K series modules.
	87k_ai	Shows how to read the AI values of AI module. This demo program is used by 87K series AI modules.
	87k_ao	Shows how to write the AO values to AO module. This demo program is used by 87K series AO modules.
	87k_di	Shows how to read the DI values of DI module. This demo program is used by 87K series DI modules.
	87k_do	Shows how to write the DO values to DO module. This demo program is used by 87K series DO modules.
	87k_dio	Shows how to read the DI and the DO values of the DIO module. This demo program is used by 87K series DIO modules.
Remote	7k87k_basic	Shows how to send/receive a command/response application. This demo program is used by 7K or 87K series

Folder	Demo	Explanation
		AI modules which connected through a COM port.
	7k87k_ai	Shows how to read the AI values of AI module. This demo program is used by 7K or 87K series AI modules which connected through a COM port.
	7k87k_ao	Shows how to write the AO values to AO module. This demo program is used by 7K or 87K series AI modules which connected through a COM port.
	7k87k_di	Shows how to read the DI values of DI module. This demo program is used by 7K or 87K series AI modules which connected through a COM port.
	7k87k_do	Shows how to write the DO values to DO module. This demo program is used by 7K or 87K series AI modules which connected through a COM port.
	7k87k_dio	Shows how to read the DI and the DO values of the DIO module. This demo program is used by 7K or 87K series AI modules which connected through a COM port.

For VB.net application, these demo programs can be obtained from:

CD:\Napdos\wp-8x4x_ce50\Demo\WinPAC\DotNET\VB.NET\PAC_IO\

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/dotnet/vb.net/pac_io/

For C# application, these demo programs can be obtained from:

CD:\Napdos\wp-8x4x_ce50\Demo\WinPAC\DotNET\C#\PAC_IO

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/dotnet/c#/pac_io/

5.1. Principles with Applications

There are a number of principles are introduced for using WinPAC API, which are fundamental have to know if you use demos or develop applications with WinPAC API.

5.1.1. Component synchronism between Host PC and WinPAC-8000

ICP DAS will continue to improve WinPAC SDK to make you develop applications easily and quickly. Each updated WinPAC SDK may contain header files, libraries, document …etc.

There are two methods to update components:

1. Auto update

Step 1: On the host PC side, run the “wp8000_update_for_desktop(wp20080917).msi” file

Which is located at:

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/update/wpyyyymmdd/wp8000_update_for_desktop(wpyyyymmdd)/

Step 2: On the WinPAC-8000 controller side, run the “wp8000_update(wpmmddyyyy).CAB” file

Which is located at:

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/update/wpyyyymmdd/wp8000_update_for_device(wpyyyymmdd)/

2. Manually update components

If you run the “PAC270_SDK_YYYYMMDD.msi” to install the component may take several minutes, if you need to reduce the installation time, you can choose manually updating the components.

The WinPAC SDK installation files are divided into the following parts:

1. WinPAC SDK updates for eMbedded Visual C++
2. WinPAC SDK updates for dotnet

WinPAC SDK updates for eMbedded Visual C++

Step 1: Get the latest version of the eMbedded Visual C++ components

The latest version of the eMbedded Visual C++ components can be obtained from:

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/sdk/winpacsdk/

Step 2: Copy the latest version of header files and libraries to Host PC

The header files are located at:

C:\Program Files\Windows CE Tools\wce500\PAC270\Include

The libraries are located at:

C:\Program Files\Windows CE Tools\wce500\PAC270\Lib

Step 3: Copy the latest version of DLL files to WinPAC-8000

The DLL files are located at:

\System_Disk\ICPDAS\System

WinPAC SDK updates for dotnet

Step 1: Get the latest version of the dotnet components

The latest version of the eMbedded Visual C++ components can be obtained from:

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/sdk/winpacnet/

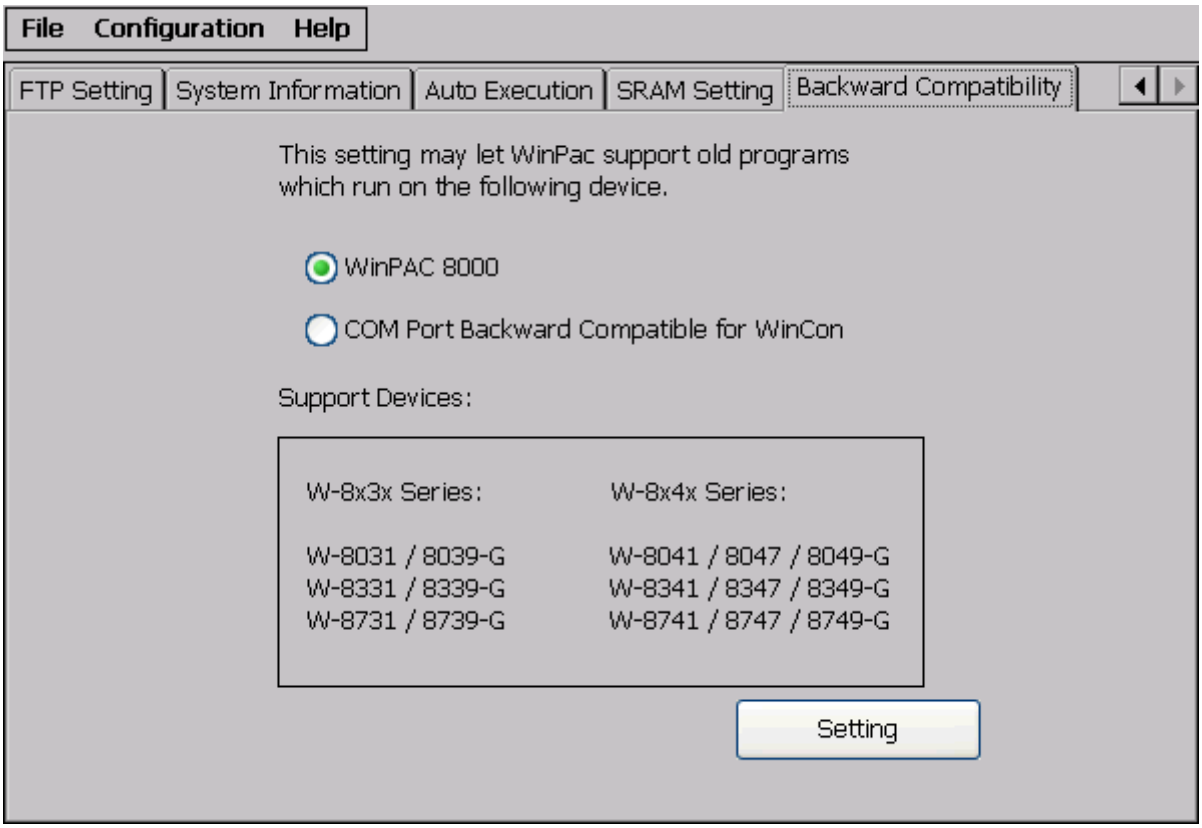
Step 2: Copy the latest version of DLL to Host PC and WinPAC-8000

The DLL files on Host PC are located at anywhere only the solution can reference it.

The DLL files on WinPAC-8000 are located at the same directory as the .exe file.

5.1.2. Backward compatible

The WinPAC-8000 is backwards compatible with WinCon-8000. Therefore, the old WinCon programs can run on WinPAC-8000 without any modification, you just have to configure the "COM Port Backward Compatible" function on WinPAC Utility and copy the applications (.exe file) from WinCon-8000 to WinPAC-8000.

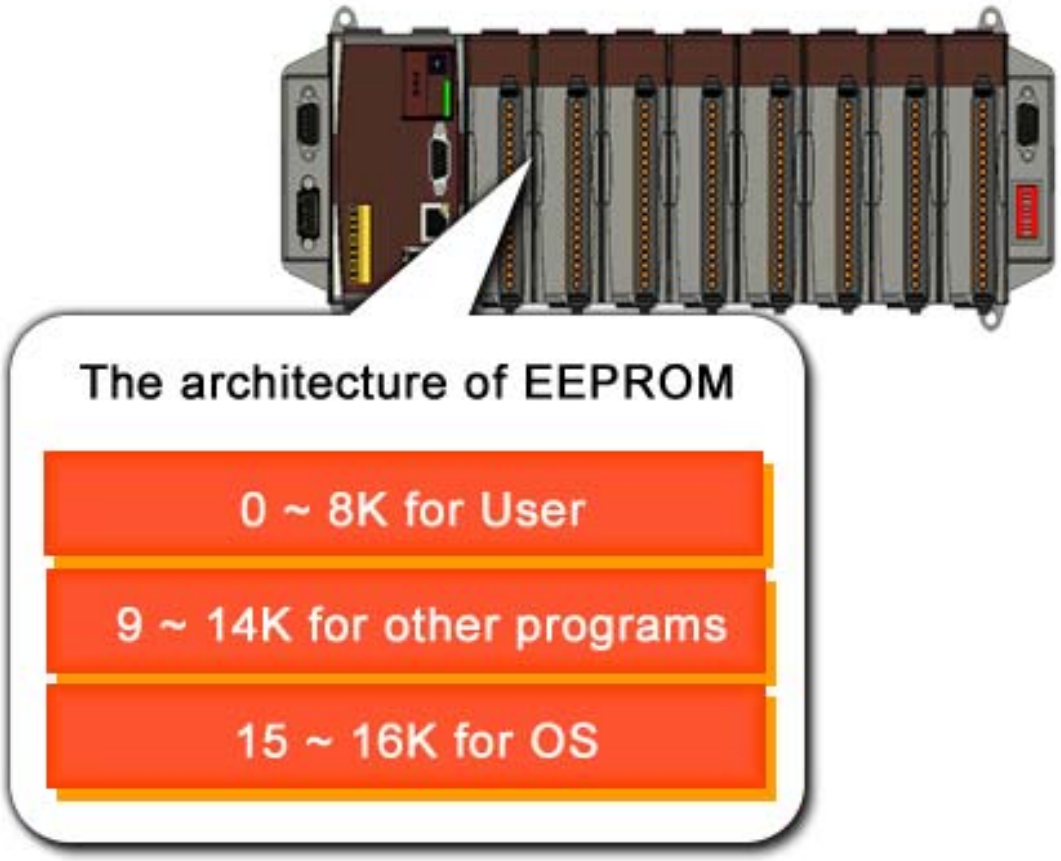


After performing above-mentioned procedures, if the program can't run on WinPAC-8000, there may be some project configure with old WinCon settings.

For more detailed information for adjusting the old WinPAC settings, please refer to: **"Appendix E. How to recompile WinCon programs"**

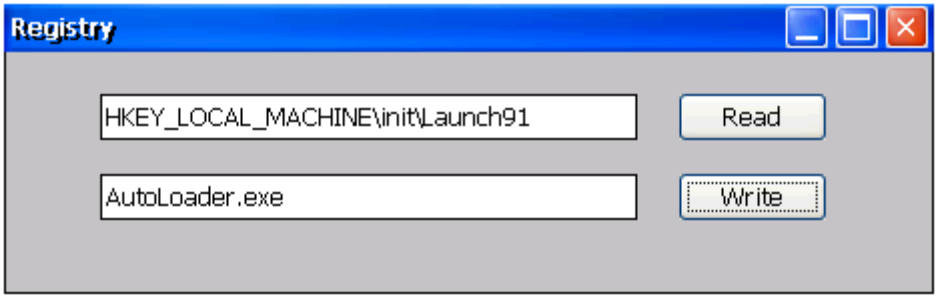
5.1.3. EEPROM allocation

The WinPAC-8000 controller has a 16KB of EEPROM that is allocated for user, program and OS as shown below:



5.1.4. Registry Mechanism

The API provides functions and demos to set registry, the registry is a database used to store important settings and configuration information for WinPAC-8000. The registry contains information and settings for all the hardware, software, users, and preferences of the WinPAC 8000 OS. Therefore, be careful for editing the registry functions and demos.



5.2. Demo programs with eMbedded Visual C++

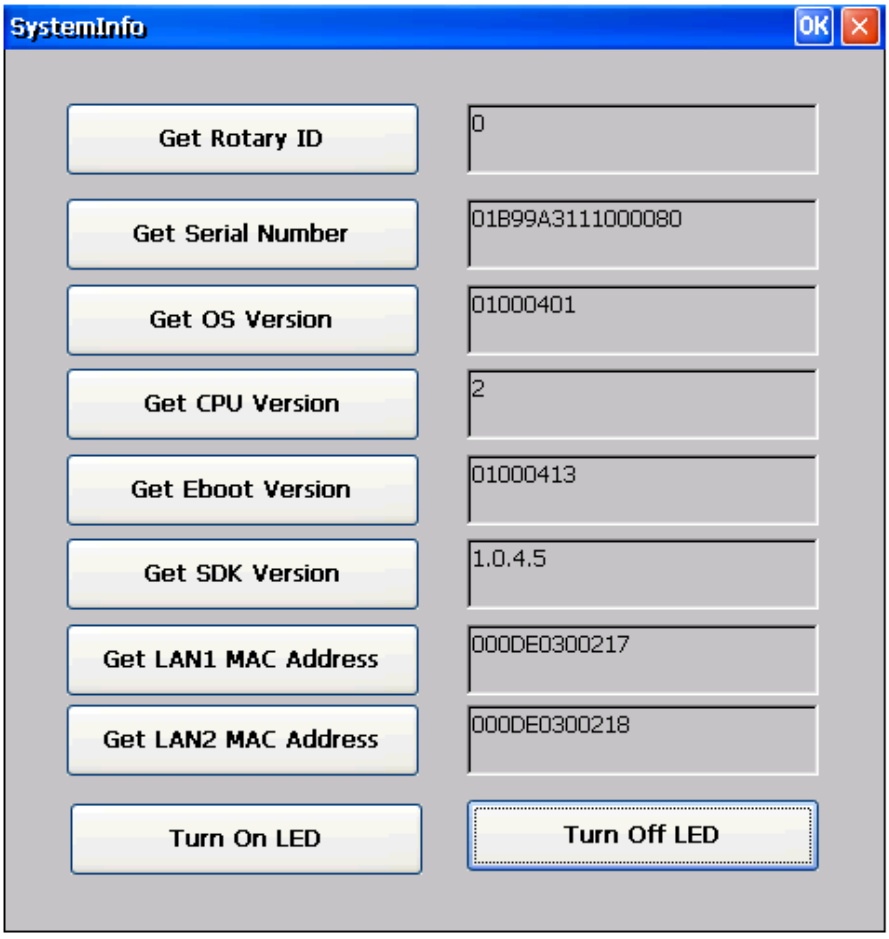
Examples are good way to understand the WinPAC SDK. This section includes the examples which extract from WinPAC demo and cover most of the common usages of each WinPAC API for using eMbedded Visual C++.

5.2.1. Demo programs for system information

System operations include basic operation, such as reboot and changing slot and Version display, including OS, Eboot, SDK, Serial Number, and Mac address.

How to use system information functions

Here is an example which extract from “systeminfo” of WinPAC demos



Download the complete demo:

CD:\Napdos\wp-8x4x_ce50\Demo\WinPAC\eVC\Standard\System\Systeminfo\

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/evc/standard/system/systeminfo/

The example illustrates how to:

1. **Read the rotary ID**

   pac_GetRotaryID();

2. **Read the Serial Number**

   pac_GetSerialNumber(LPSTR SerialNumber);

3. **Read the OS version**

   pac_GetOSVersion(LPSTR os_version);

4. **Read the CPU version**

   pac_GetCPUVersion(LPSTR cpu_version);

4. **Read the Eboot version**

   pac_GetEbootVersion(LPSTR eboot_version);

5. **Read the SDK version**

   pac_GetSDKVersion(LPSTR sdk_version);

6. **Read the MAC address Version**

   pac_GetMacAddress(BYTE LAN, LPSTR MacAddr);

7. **Turn on/off LED**

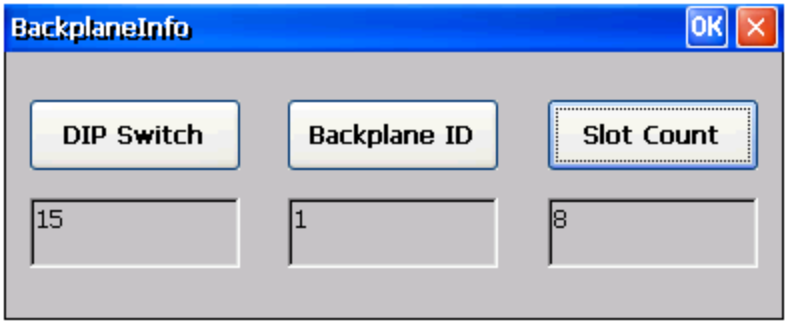
   pac_EnableLED(BOOL bFlag);

5.2.2. Demo programs for backplane access

Backplane operations include hot plug, interrupt and backplane information, such as NET ID and backplane version.

How to use backplane access functions

Here is an example which extract from “backplaneinfo” of WinPAC demos



Download the complete demo:

CD:\Napdos\wp-8x4x_ce50\Demo\WinPAC\eVC\Standard\Backplane\BackplaneInfo\

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/evc/standard/backplane/backplaneinfo/

The example illustrates how to:

1. **Read the DIP Switch**

   pac_GetDIPSwitch();

2. **Read the backplane ID**

   pac_GetBackplaneID(LPSTR backplane_version);

3. **Read the slot count**

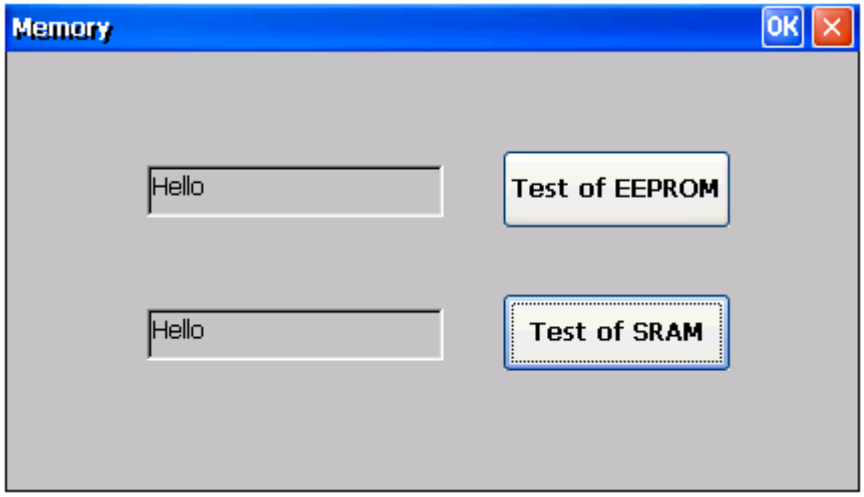
   pac_GetSlotCount();

5.2.3. Demo programs for memory access

Memory operations include basic management operations, such as reading from and writing to the EEPROM or SRAM.

How to use memory access functions

Here is an example which extract from “memory” of WinPAC demos



Download the complete demo:

CD:\Napdos\wp-8x4x_ce50\Demo\WinPAC\eVC\Standard\MemoryAccess\Memory\

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/evc/standard/memory access/memory/

The example illustrates how to:

1. **Enable the EEPROM**

   pac_EnableEEPROM(bool);

2. **Write the EEPROM**

   pac_WriteMemory(DWORD address, LPBYTE lpBuffer, DWORD dwLength, int mem_type);

3. **Read the EEPROM**

   pac_ReadMemory(DWORD address, LPBYTE lpBuffer, DWORD dwLength, int mem_type);

5.2.4. Demo programs for watchdog

Watchdog operations include basic management operations, such as turning on and refreshing.

How to use watchdog functions

Here is an example which extract from “watchdog” of WinPAC demos

Download the complete demo:

CD:\Napdos\wp-8x4x_ce50\Demo\WinPAC\eVC\Standard\WatchDog\

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/evc/standard/watchdog/

The example illustrates how to:

1. **Enable the watchdog**

   pac_EnableWatchDog(int wdt, DWORD value);

2. **Refresh the watchdog**

   pac_RefreshWatchDog(int wdt);

3. Disable the watchdog

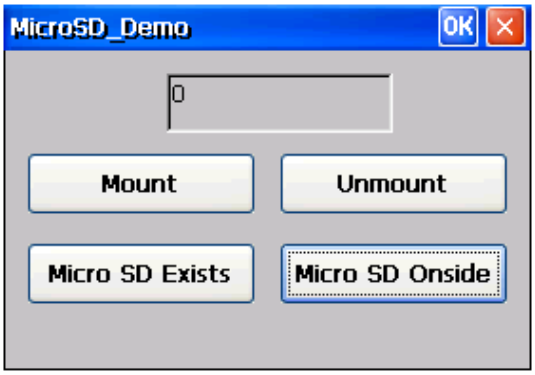
pac_DisableWatchDog(int wdt);

5.2.5. Demo programs for MicroSD management

MicroSD operations include basic management operations, such as mounting and unmounting.

How to use MicroSD management functions

Here is an example which extract from “microsd_management” of WinPAC demos



Download the complete demo:

CD:\Napdos\wp-8x4x_ce50\demo\WinPAC\eVC\Standard\MicroSD\MicroSD\

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/evc/standard/microsd/microsd/

The example illustrates how to:

1. **Check the Micro SD whether has been standby ready or not**

   pac_SDExists();

2. **Mount the Micro SD**

   pac_SDMount(LPTSTR szPartitionName);

3. **Dismount the Micro SD**

   pac_SDUnmount();

4. **Check the Micro SD whether on-side or not**

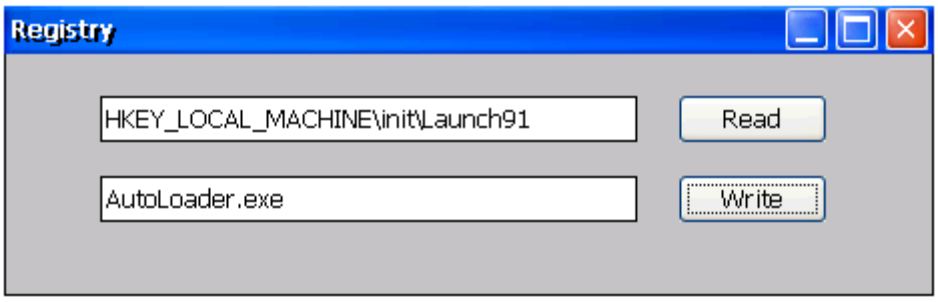
   pac_SDOnside();

5.2.6. Demo programs for registry

Registry operations include basic management operations, such as reading from and writing to the registry.

How to use registry functions

Here is an example which extract from “Registry” of WinPAC demos



Download the complete demo:

CD:\Napdos\wp-8x4x_ce50\demo\WinPAC\eVC\Standard\Registry\Registry\

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/evc/standard/registry/registry/

The incorrect use of registry demo can cause serious problems that may require you to run safe mode to reload them.

The example illustrates how to:

1. Assign the specified registry key date which type is string

pac_RegSetString(LPCTSTR KeyName, LPCTSTR assignStr, DWORD dwLength);

2. Read the value of the specified registry key

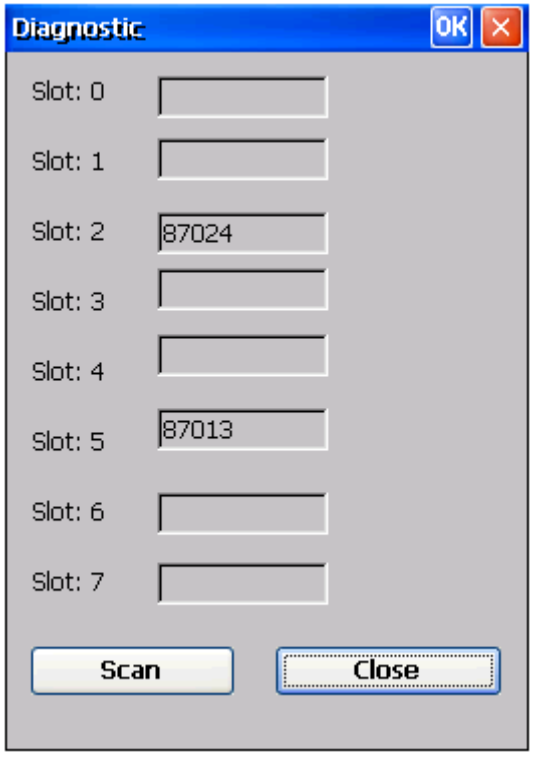
pac_RegGetDWORD(LPCTSTR KeyName, LPTSTR lpData, DWORD dwLength);

5.2.7. Demo programs for UART

Uart operations include basic management operations, such as opening, sending, receiving, and closing.

How to use UART functions

Here is an example which extract from “diag” of WinPAC demos



Download the complete demo:

CD:\Napdos\wp-8x4x_ce50\demo\WinPAC\eVC\Standard\UART\Diag\

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/evc/standard/uart/diag/

The example illustrates how to:

1. **Open the COM port and specifies the baud rate, parity bits and stop bits**

   uart_Open(LPCSTR ConnectionString);

2. Close the COM port which have been opened

uart_Close(HANDLE hPort);

3. Send commands through the COM port which have been opened

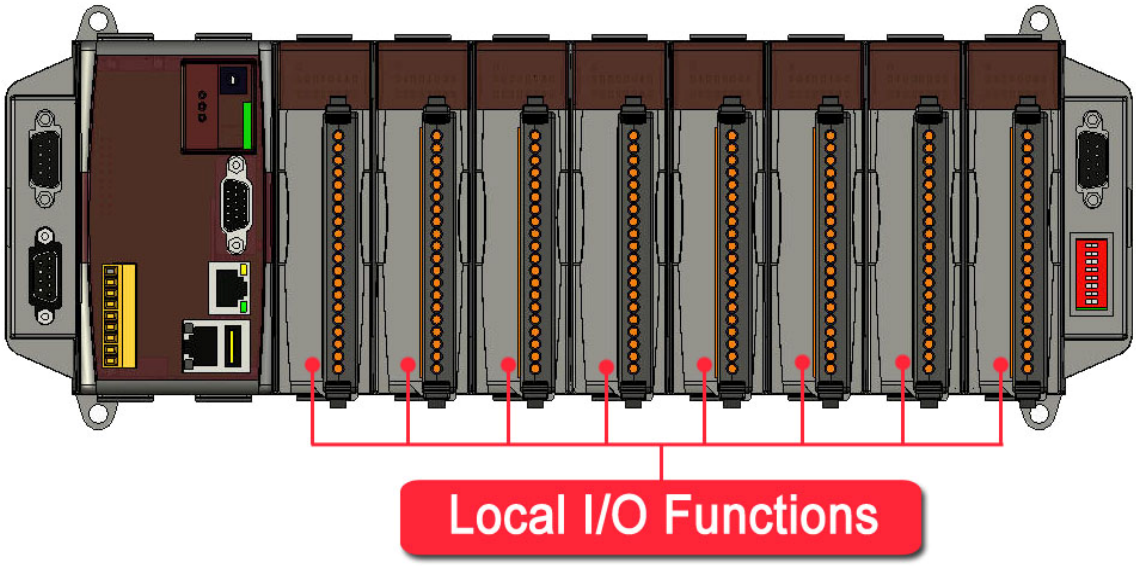
uart_SendCmd(HANDLE hPort, LPCSTR cmd, LPSTR szResult);

5.2.8. Demo programs for PAC_IO

PAC_IO API supports to operate IO modules not only in slot but in remote mode.

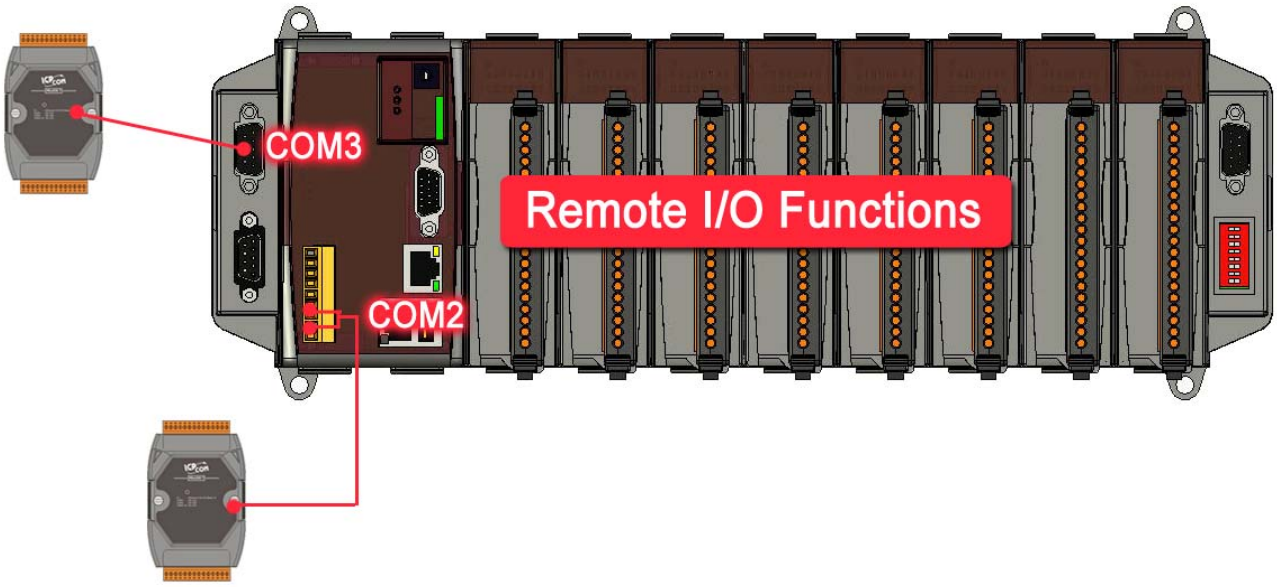
In the local mode, the slot range is from 0 to 7.

Local (IO in slot):



In the remote mode, the slot range is from 0 to 255.

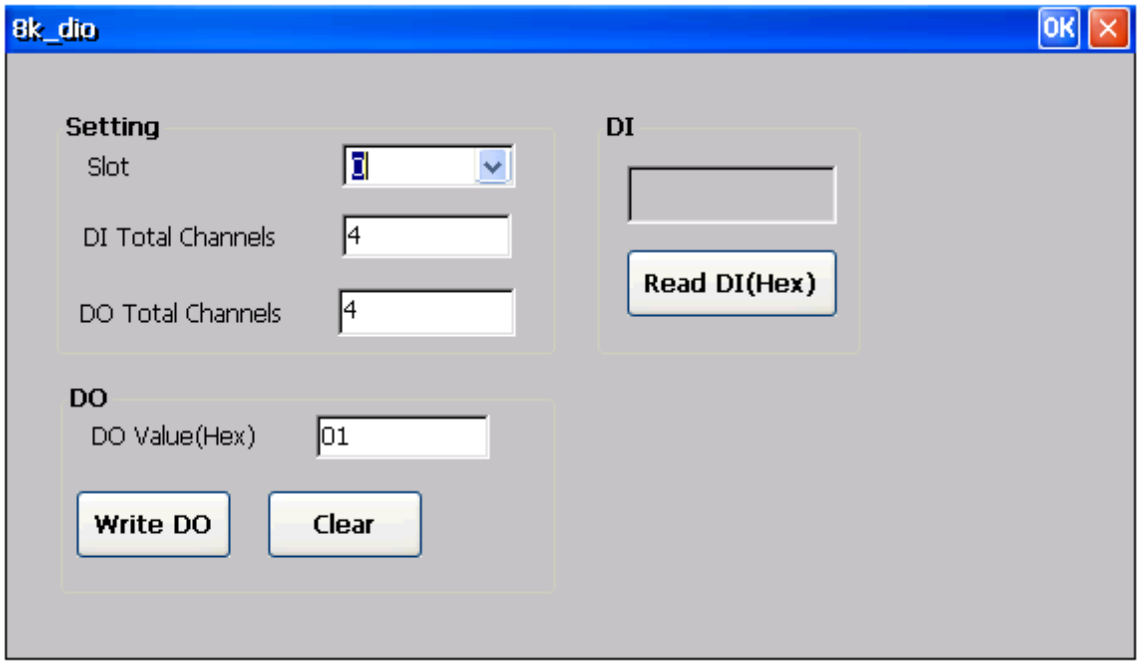
Remote:



How to use Local I/O functions

Here are some examples

Example 1: The example extract from “8k_dio” of WinPAC demos



Download the complete demo:

CD:\Napdos\wp-8x4x_ce50\demo\WinPAC\eVC\PAC_IO\Local\8k_dio\

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/evc/pac_io/local/8k_dio/

The example illustrates how to:

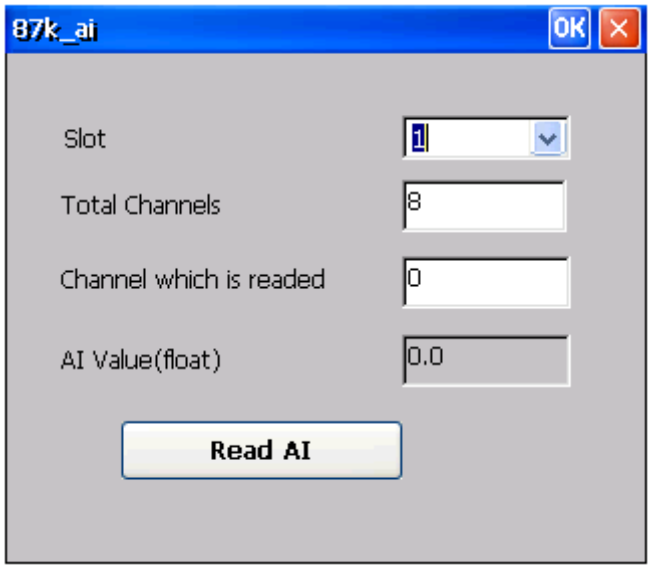
1. **Read the DI value of the DI module**

   pac_ReadDI(HANDLE hPort, int slot, int iDI_TotalCh, DWORD *lDI_Value);

2. **Write the DO value to the DO modules**

   pac_WriteDO(HANDLE hPort, int slot, int iDO_TotalCh, DWORD lDO_Value);

Example 2: The example extract from “87k_ai” of WinPAC demos



Download the complete demo:

CD:\Napdos\wp-8x4x_ce50\demo\WinPAC\eVC\PAC_IO\Local\87k_ai\

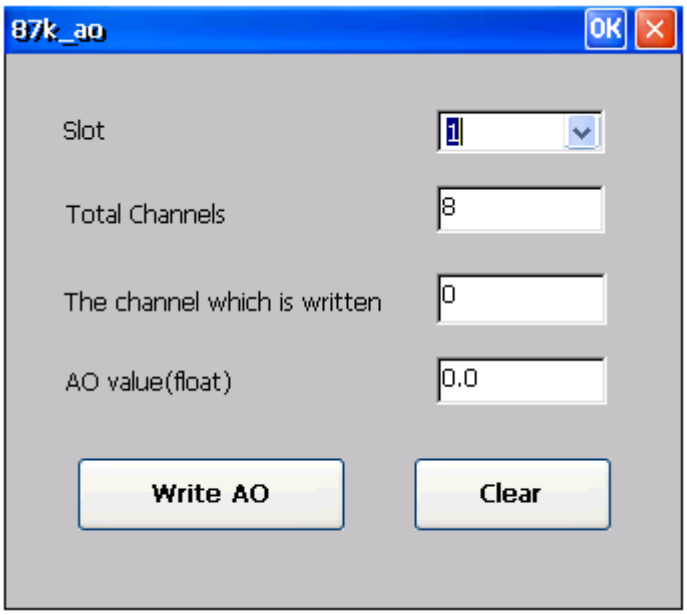
ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/evc/pac_io/local/87k_ai/

The example illustrates how to:

Read the engineering-mode AI value of the AI module

pac_ReadAI(hPort, iSlot,iChannel,iAI_TotalCh, &fValue);

Example 3: The example extract from “87k_ao” of WinPAC demos



Download the complete demo:

CD:\Napdos\wp-8x4x_ce50\demo\WinPAC\eVC\PAC_IO\Local\87k_ao\

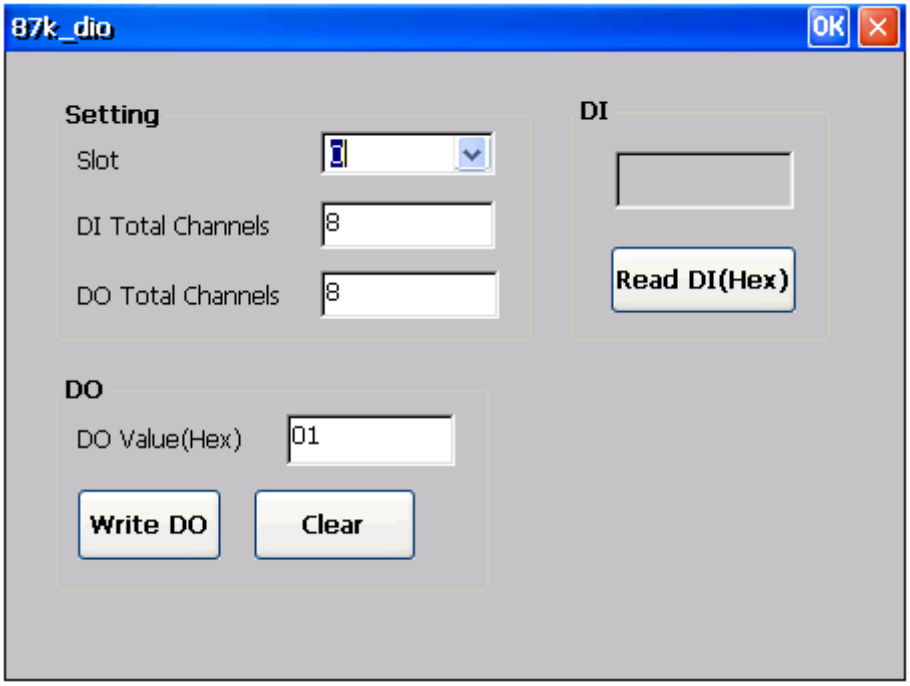
ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/evc/pac_io/local/87k_ao/

The example illustrates how to:

Write the AO value to the AO modules

pac_WriteAO(hPort, iSlot,iChannel,iAO_TotalCh,fValue);

Example 4: The example extract from “87k_dio” of WinPAC demos



Download the complete demo:

CD:\Napdos\wp-8x4x_ce50\demo\WinPAC\eVC\PAC_IO\Local\87k_dio\

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/evc/pac_io/local/87k_dio/

The example illustrates how to:

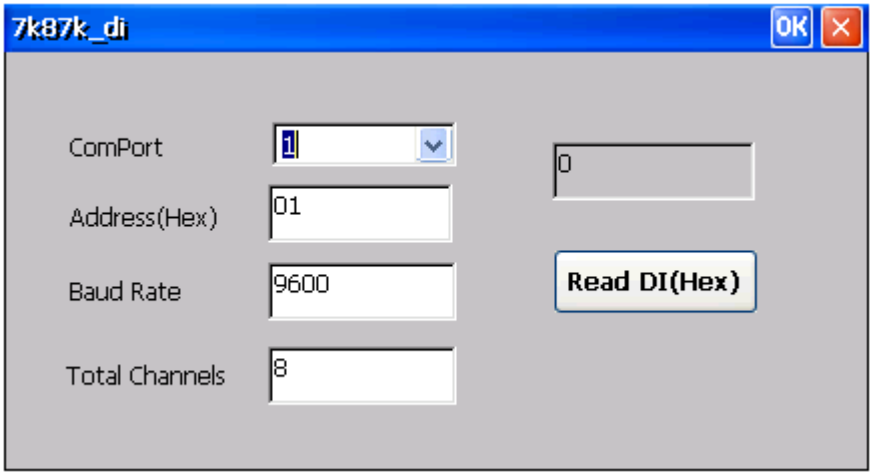
Read the DI and the DO values of the DIO module

pac_ReadDIO(HANDLE hPort, int slot, int iDI_TotalCh, int iDO_TotalCh, DWORD* lDI_Value, DWORD* lDO_Value);

How to use remote I/O functions

Here are some examples

Example 1: The example extract from “7k87k_di” of WinPAC demos



Download the complete demo:

CD:\Napdos\wp-8x4x_ce50\demo\WinPAC\eVC\PAC_IO\Remote\7K87k_di\

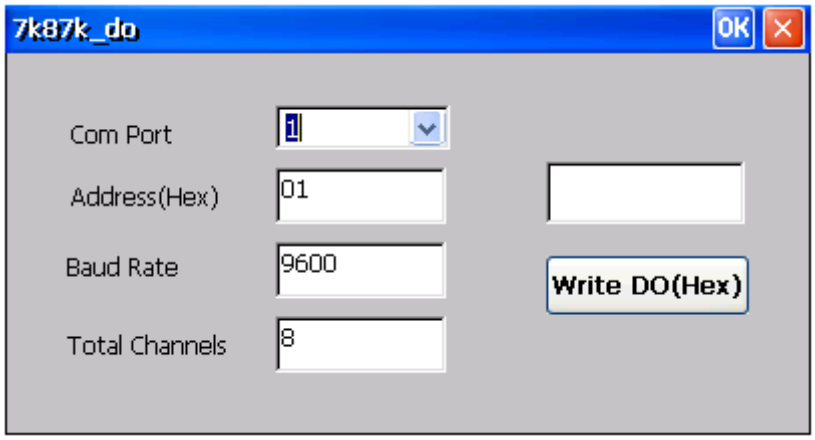
ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/evc/pac_io/remote/7k87k_di/

The example illustrates how to:

Read the DI value of the DI module

pac_ReadDI(HANDLE hPort, int PAC_REMOTE_IO (iAddr) , int iDI_TotalCh, DWORD *lDI_Value);

Example 2: The example extract from “7k87k_do” of WinPAC demos



Download the complete demo:

CD:\Napdos\wp-8x4x_ce50\demo\WinPAC\eVC\PAC_IO\Remote\7k87k_do\

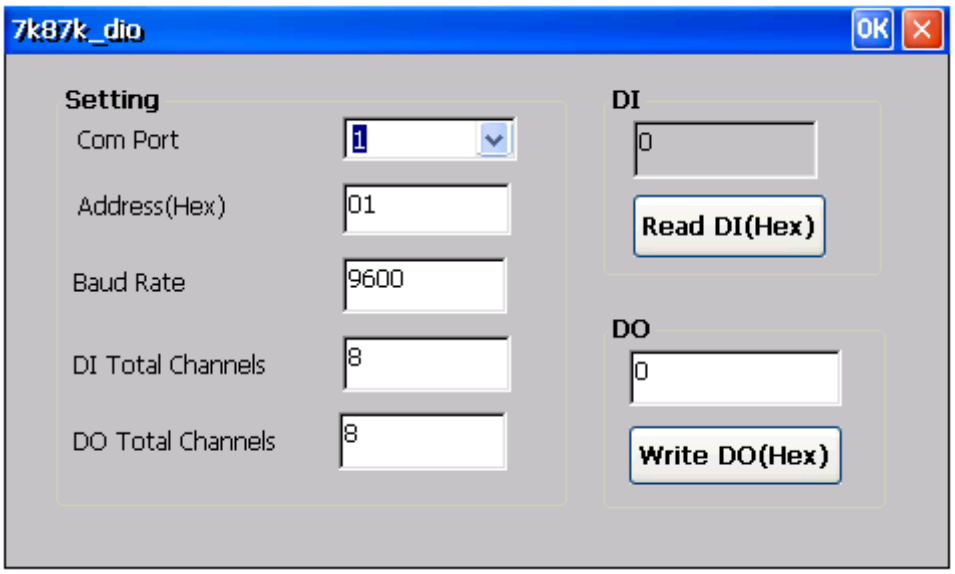
ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/evc/pac_io/remote/7k87k_do/

The example illustrates how to:

Write the DO value to the DO modules

pac_WriteDO(HANDLE hPort, int PAC_REMOTE_IO (iAddr) , int iDO_TotalCh, DWORD lDO_Value);

Example 3: The example extract from “7k87k_dio” of WinPAC demos



Download the complete demo:

CD:\Napdos\wp-8x4x_ce50\demo\WinPAC\eVC\PAC_IO\Remote\7k87k_dio\

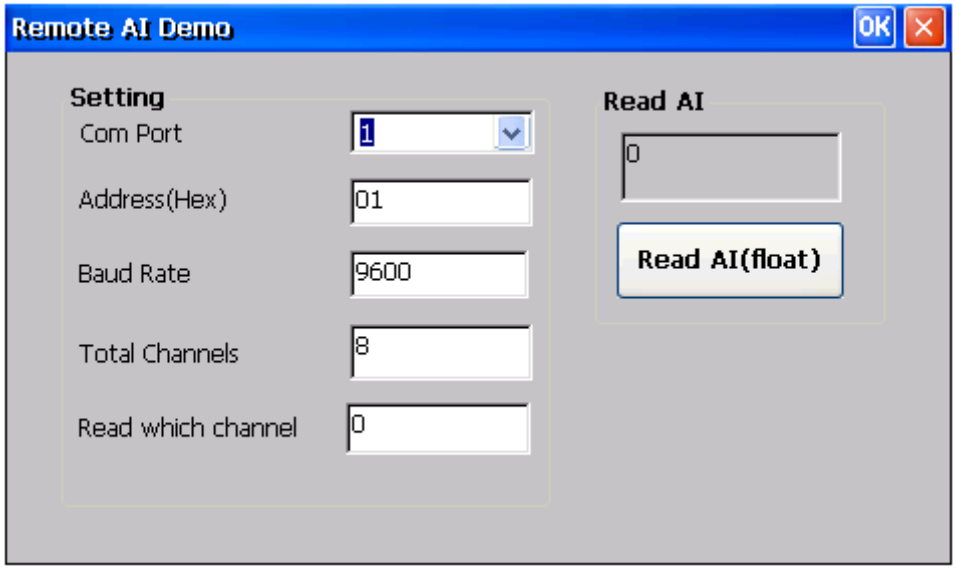
ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/evc/pac_io/remote/7k87k_dio/

The example illustrates how to:

Read the DI and the DO values of the DIO module

pac_ReadDIO(HANDLE hPort, int PAC_REMOTE_IO (iAddr) , int iDO_TotalCh, DWORD lDO_Value);

Example 4: The example extract from “7k87k_ai” of WinPAC demos



Download the complete demo:

CD:\Napdos\wp-8x4x_ce50\demo\WinPAC\eVC\PAC_IO\Remote\7k87k_ai\

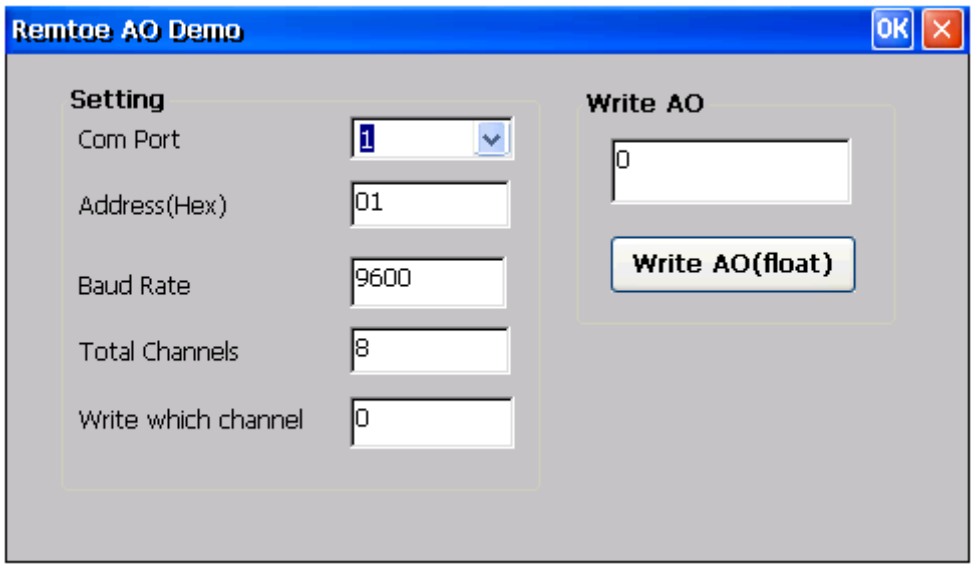
ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/evc/pac_io/remote/7k87k_ai/

The example illustrates how to:

Read the engineering-mode AI value of the AI module

pac_ReadAI(hPort, PAC_REMOTE_IO (iAddr) ,iChannel,iAI_TotalCh, &fValue);

Example 5: The example extract from “7k87k_ao” of WinPAC demos



Download the complete demo:

CD:\Napdos\wp-8x4x_ce50\demo\WinPAC\eVC\PAC_IO\Remote\7k87k_ao\

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/evc/pac_io/remote/7k87k_ao/

The example illustrates how to:

Write the AO value to the AO modules

pac_WriteAO(hPort, PAC_REMOTE_IO (iAddr) , iChannel,iAO_TotalCh,fValue);

5.2.9. Demo programs for error handling

Error handling operations enable you to receive and display error information for your application.

How to use error handling

Here is an example which extract from “Error” of WinPAC demos

Download the complete demo:

CD:\Napdos\wp-8x4x_ce50\demo\WinPAC\eVC\Other\Error\

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/evc/other/error/

The example illustrates how to:

1. **Return the last-error code value**

   pac_GetLastError();

2. **Retrieve an error message string**

   pac_GetErrorMessage(DWORD dwMessageID, LPTSTR lpBuffer);

5.2.10. Demo programs for interrupt

The I/O modules that support interrupt function can be used for counting, timing, detecting external events, and sending and receiving data using the parallel bus.

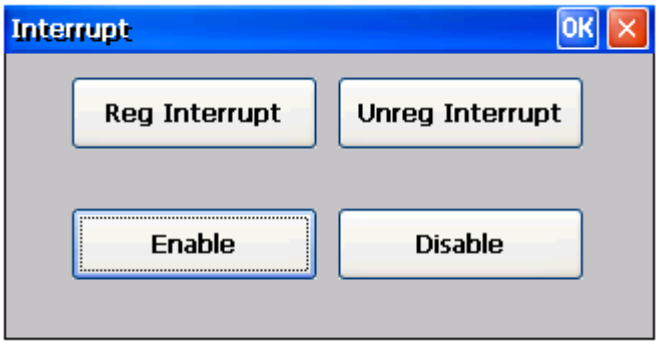
If want to use the functions about interrupt, you should use the IO modules which support these operations.

For more information please reference website: http://www.icpdas.com

---

How to use interrupt function

Here is an example which extract from “interrupt” of WinPAC demos



Download the complete demo:

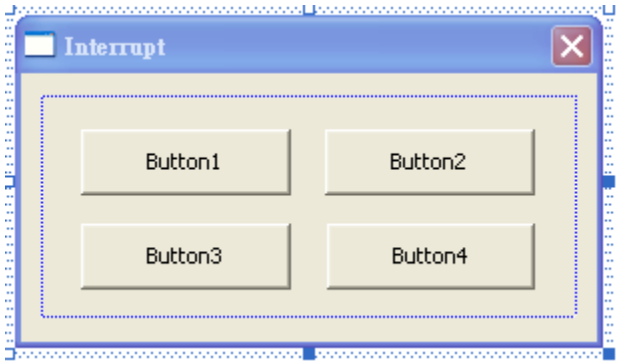
CD:\Napdos\wp-8x4x_ce50\demo\WinPAC\eVC\Standard\Backplane\Interrupt\

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/evc/standard/backplane/interrupt/

Steps to write interrupt application:

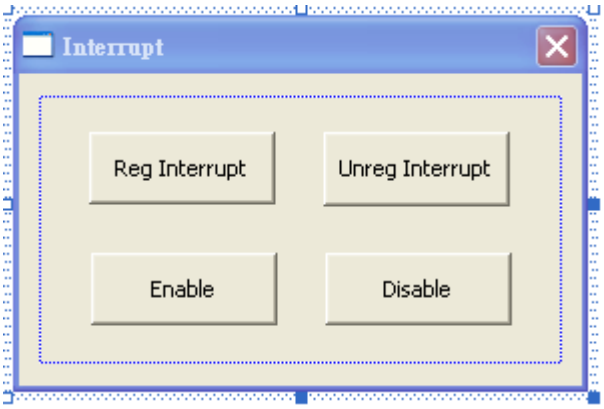
Step 1: Create a new project named “Interrupt”, whose type is “WCE MFC AppWizard (exe)” and dialog base.

Step 2: Design the “Interrupt” dialog box as follows:

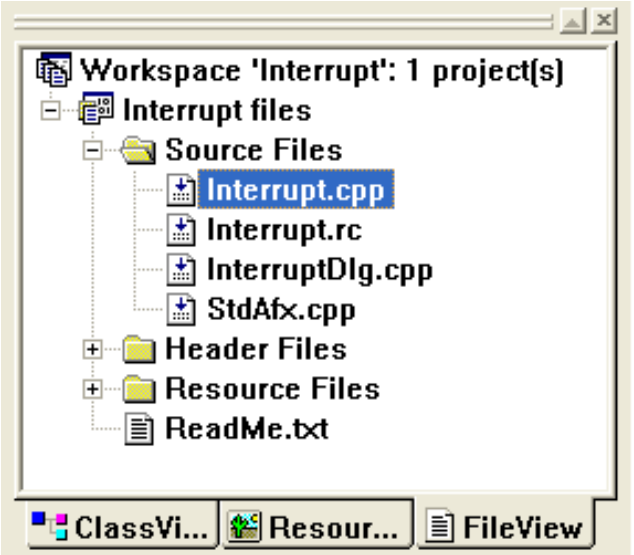


Step 3: Right-Click on each button object and select the “Properties” command to configure the caption as follows:

Object	Caption
button1	Reg Interrupt
button2	Unreg Interrupt
button3	Enable
button4	Disable



Step 4: On the “Workspace” window, select the “FileView” tab and expand the “Source Files” folder, and then double-click the “Interrupt.cpp” to open the editor window.



Step 5: Add the following code in the header area.

#include the WinPacSDK.h

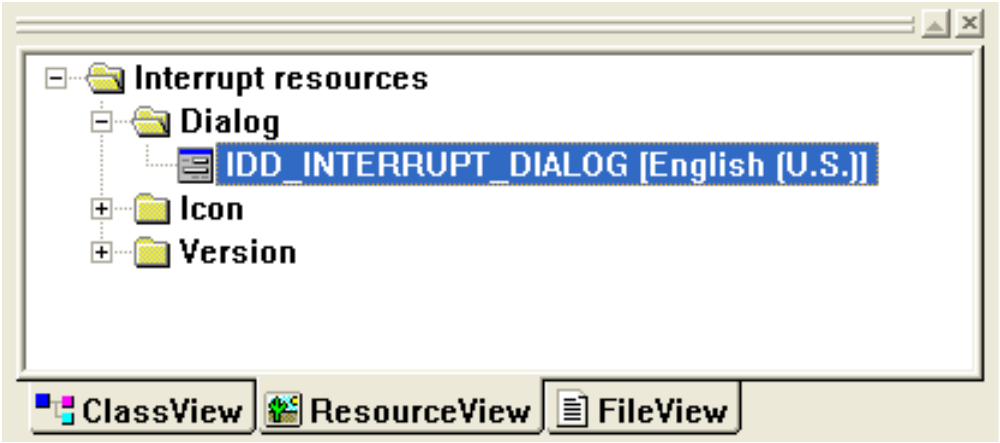
```
// InterruptDlg.cpp : implementation file
//

#include "stdafx.h"
#include "Interrupt.h"
#include "InterruptDlg.h"

#include "WinPacSDK.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif
```

Step 6: Add the following code in the edit area below.

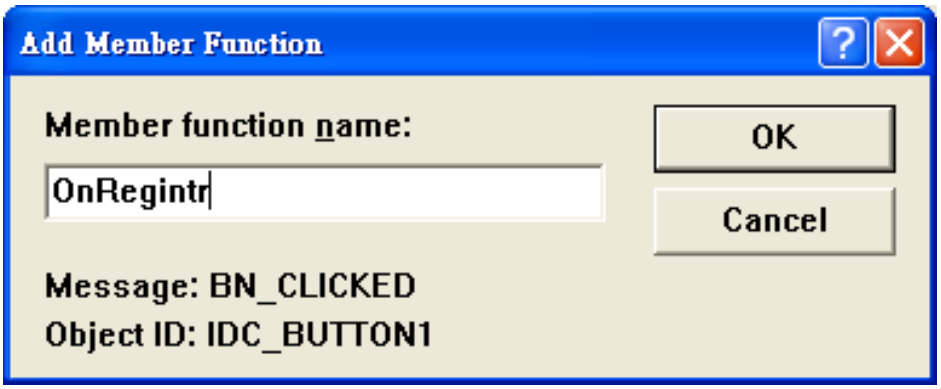
```
Int CALLBACK demo_callback_proc7()
{
    static int c = 0;
    PAC_OUTP ( 0x1000 * ( 1+7 ) + 0xAAA, c + + %0x7F );
    PAC_OUTP ( 0x1000 * ( 1+7 ) + 0x4, c + + %0x7F );
    return0;
}
```

```
    // TODO: Add extra initialization here

    return TRUE;  // return TRUE  unless you set
}

int CALLBACK demo_callback_proc7()
{
    static int c = 0;
    PAC_OUTP(0x1000*(1+7)+0xAAA, c++%0x7F);
    PAC_OUTP(0X1000*(1+7)+0x4, c++%0x7F);
    return 0;
}
```

Step 7: On the “Workspace” window, select the “ResourceView” tab and expand the “Dialog” folder, and then double-click the “IDD_INTERRUPT_DIALOG [English [U.S.]]” to open the Interrupt dialog.



Step 8: Double-click the “Reg Interrupt” object to configure the Member function name “OnRegintr”

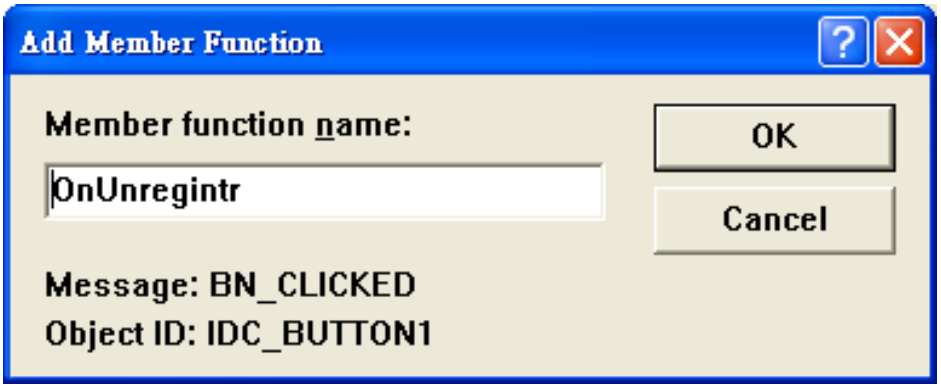


Step 9: Add the following code in the “OnUnregintr” function edit area

pac_RegisterSlotInterrupt(7, demo_callback_proc7);

```
void CInterruptDlg::OnRegintr()
{
    // TODO: Add your control notification handler code he
    pac_RegisterSlotInterrupt(7, demo_callback_proc7);
}
```

Step 10: Double-click the “Unreg Interrupt” object to configure the Member function name “OnUnregintr”

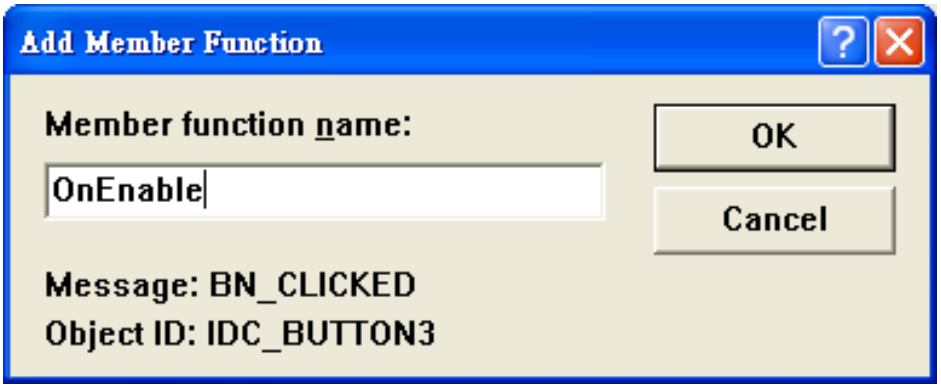


Step 11: Add the following code in the “OnUnregintr” function edit area

pac_UnregisterSlotInterrupt(7);

```
void CInterruptDlg::OnUnregintr()
{
    // TODO: Add your control notific
    pac_UnregisterSlotInterrupt(7);
}
```

Step 12: Double-click the “Enable” object to configure the Member function name “OnEnable”



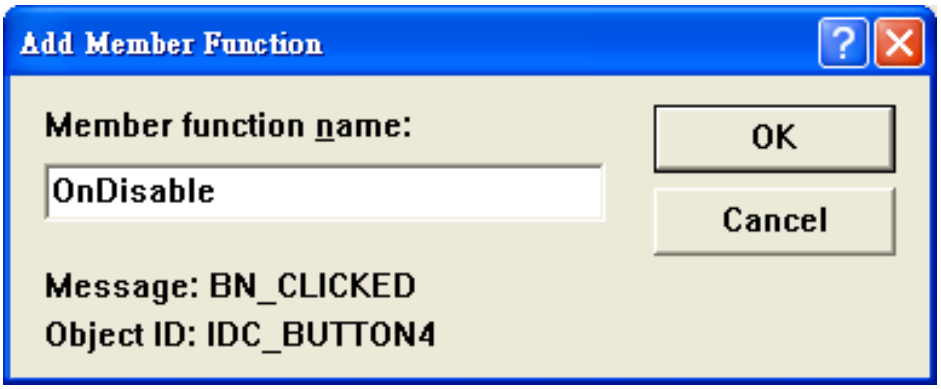
Step 13: Add the following code in the “OnEnable” function edit area

pac_EnableSlotInterrupt(7, true);

PAC_OUTP (0x 1000* (1 + 7) + 0x4, 6);

```
void CInterruptDlg::OnEnable()
{
    // TODO: Add your control notificatio
   pac_EnableSlotInterrupt(7, true);
   PAC_OUTP(0x1000*(1+7)+0x4, 6);

}
```

Step 14: Double-click the “Disable” object to configure the Member function name “OnDisable”



Step 15: Add the following code in the “OnDisable” function edit area

pac_EnableSlotInterrupt (7, false);

```
void CInterruptDlg::OnDisable()
{
    // TODO: Add your control notificatio
    pac_EnableSlotInterrupt(7, false);
}
```

5.3. Demo programs with C#

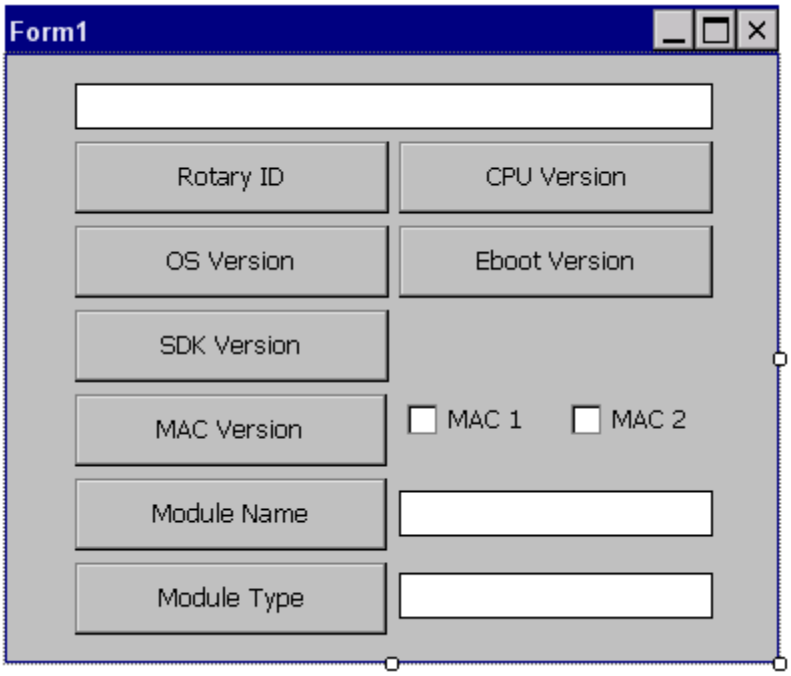
Examples are good way to understand the WinPAC SDK. This section includes the examples which extract from WinPAC demo and cover most of the common usages of each WinPAC API for using C#.

5.3.1. Demo programs for system information

System operations include basic operation, such as reboot and changing slot and Version display, including OS, Eboot, SDK, Serial Number, and Mac address.

How to use system information functions

Here is an example which extract from “systeminfo” of WinPAC demos



Download the complete demo:

CD:\Napdos\wp-8x4x_ce50\demo\WinPAC\DotNET\C#\Standard\System\Systeminfo\

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/dotnet/c#/standard/system/systeminfo/

The example illustrates how to:

1. **Read the CPU version**

   WinPAC..pac_GetCPUVersion();

2. **Read the rotary ID**

   WinPAC.pac_GetRotaryID();

3. **Read the OS version**

   WinPAC..pac_GetOSVersion();

4. **Read the Eboot version**

   WinPAC..pac_GetEbootVersion();

5. **Read the SDK version**

   WinPAC..pac_GetSDKVersion();

6. **Read the MAC address Version**

   WinPAC..pac_GetMacAddress(byte LAN);

7. **Read the module type**

   WinPAC.pac_GetModuleName();

8. **Read the module name**

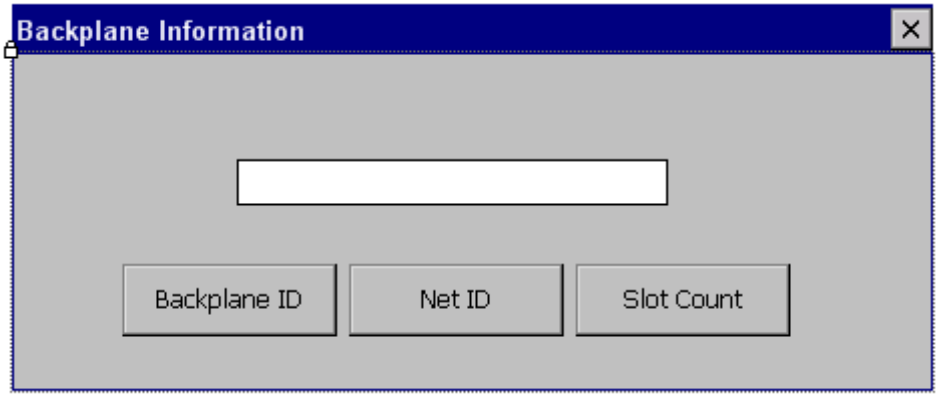
   WinPAC.pac_GetModuleType(byte slot, string strName);

5.3.2. Demo programs for backplane access

Backplane operations include hot plug, interrupt and backplane information, such as NET ID and backplane version.

How to use backplane access functions

Here is an example which extract from “backplaneinfo” of WinPAC demos



Download the complete demo:

CD:\Napdos\wp-8x4x_ce50\demo\WinPAC\DotNET\C#\Standard\Backplane\Backplaneinfo\

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/dotnet/c#/standard/backplane/backplaneinfo/

The example illustrates how to:

1. **Read the backplane ID**

   WinPAC.pac_GetBackplaneID();

2. **Read the DIP Switch**

   WinPAC.pac_GetDIPSwitch();

3. **Read the slot count**

   WinPAC.pac_GetOSVersion();

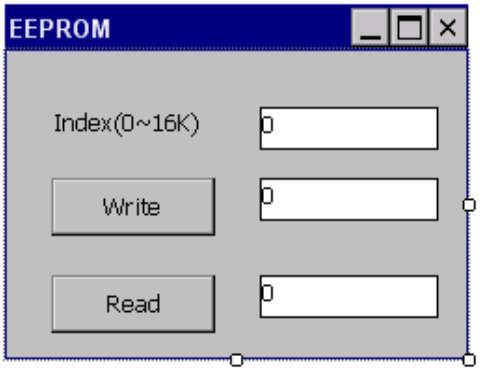
5.3.3. Demo programs for memory access

Memory operations include basic management operations, such as reading from and writing to the EEPROM or SRAM.

How to use memory access functions

Here are some examples

Example 1: The example extract from “eeprom” of WinPAC demos



Download the complete demo:

CD:\Napdos\wp-8x4x_ce50\demo\WinPAC\DotNET\C#\Standard\MemoryAccess\EEPROM\

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/dotnet/c#/standard/memoryaccess/eeprom/

The example illustrates how to:

1. Enable the EEPROM

WinPAC.pac_EnableEEPROM(bool);

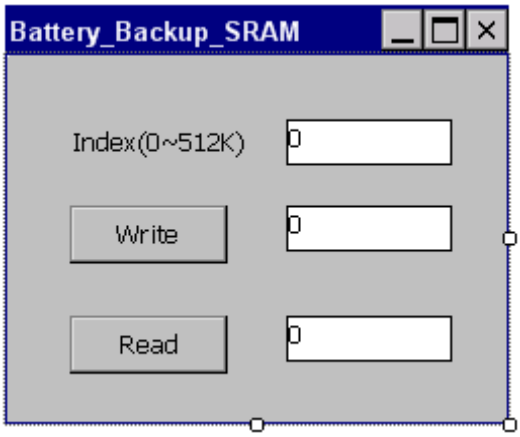
2. Write the EEPROM

WinPAC.pac_WriteMemory(unit address, byte [] lpBuffer, unit dwLength, int mem_type);

3. Read the EEPROM

WinPAC.pac_ReadMemory(unit address, byte [] lpBuffer, unit dwLength, int mem_type);

Example 2: The example extract from “battery_backup_sram” of WinPAC demos



Download the complete demo:

CD:\Napdos\wp-8x4x_ce50\demo\WinPAC\DotNET\C#\Standard\MemoryAccess\Battery_Backup_SRAM\

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/dotnet/c#/standard/memoryaccess/battery_backup_sram/

The example illustrates how to:

1. Write the EEPROM

WinPAC.pac_WriteMemory(unit address, string lpBuffer, unit dwLength, int mem_type);

2. Read the EEPROM

WinPAC.pac_ReadMemory(unit address, string lpBuffer, unit dwLength, int mem_type);

5.3.4. Demo programs for watchdog

Watchdog operations include basic management operations, such as turning on and refreshing.

How to use watchdog functions

Here is an example which extract from “watchdog” of WinPAC demos

Download the complete demo:

CD:\Napdos\wp-8x4x_ce50\demo\WinPAC\DotNET\C#\Standard\WatchDog\

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/dotnet/c#/standard/watchdog/

The example illustrates how to:

1. Enable the watchdog

WinPAC.pac_EnableWatchDog(int wdt, unit value);

2. Refresh the watchdog

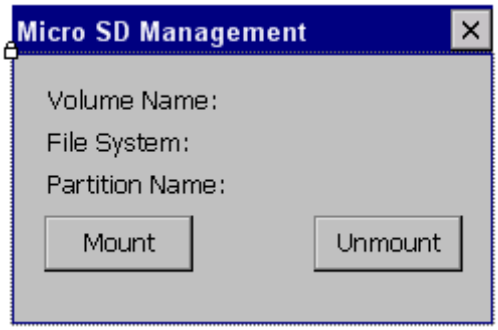
WinPAC.pac_RefreshWatchDog(int wdt);

5.3.5. Demo programs for MicroSD management

MicroSD operations include basic management operations, such as mounting and unmounting.

How to use MicroSD management functions

Here is an example which extract from “microsd_management” of WinPAC demos



Download the complete demo:

CD:\Napdos\wp-8x4x_ce50\demo\WinPAC\DotNET\C#\Standard\MicroSD\MicroSD_Management\

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/dotnet/c#/standard/microsd/microsd_management/

The example illustrates how to:

1. Check the Micro SD whether has been standby ready or not

WinPAC.pac_SDExists();

2. Mount the Micro SD

WinPAC.pac_SDMount(string szPartitionName);

3. Dismount the Micro SD

WinPAC.pac_SDUnmount();

4. Check the Micro SD whether on-side or not

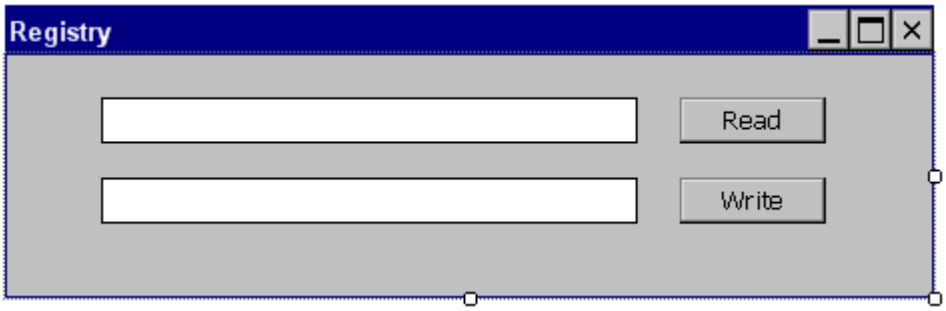
WinPAC.pac_SDOnside();

5.3.6. Demo programs for registry

Registry operations include basic management operations, such as reading from and writing to the registry.

How to use registry functions

Here is an example which extract from “Registry” of WinPAC demos



Download the complete demo:

CD:\Napdos\wp-8x4x_ce50\demo\WinPAC\DotNET\C#\Standard\Registry\Registry\

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/dotnet/c#/standard/registry/registry/

The incorrect use of registry demo can cause serious problems that may require you to run safe mode to reload them.

The example illustrates how to:

1. Assign the specified registry key date which type is string

WinPAC.pac_RegSetString(byte[ ] KeyName, string assignStr, unit dwLength);

2. Read the value of the specified registry key

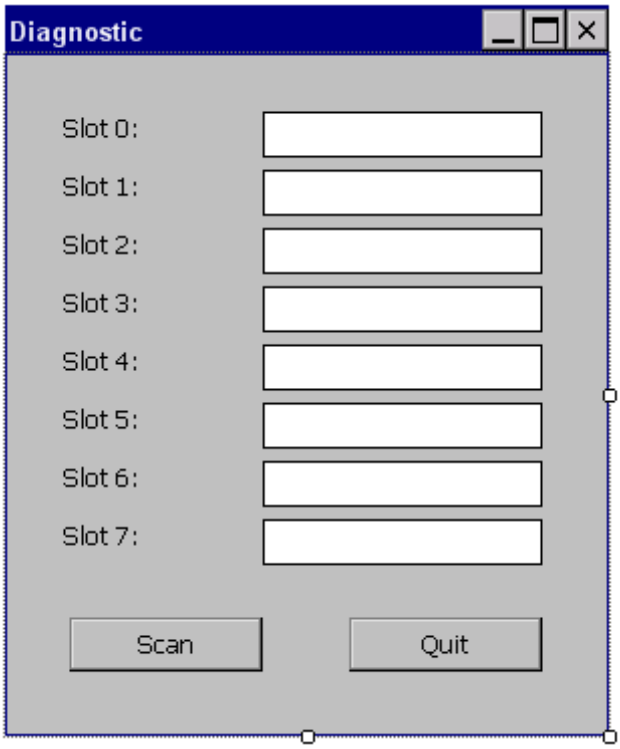
WinPAC.pac_RegGetDWORD(string KeyName, string lpData, unit dwLength);

5.3.7. Demo programs for UART

Uart operations include basic management operations, such as opening, sending, receiving, and closing.

How to use UART functions

Here is an example which extract from “diag” of WinPAC demos



Download the complete demo:

CD:\Napdos\wp-8x4x_ce50\demo\WinPAC\DotNET\C#\Standard\UART\Diag\

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/dotnet/c#/standard/uart/diag/

The example illustrates how to:

1. Open the COM port and specifies the baud rate, parity bits and stop bits

WinPAC.uart_Open("COM1:,9600,N8,1");

2. Close the COM port which have been opened

```
WinPAC.uart_Close(hOpen);
```

3. Send commands through the COM port which have been opened

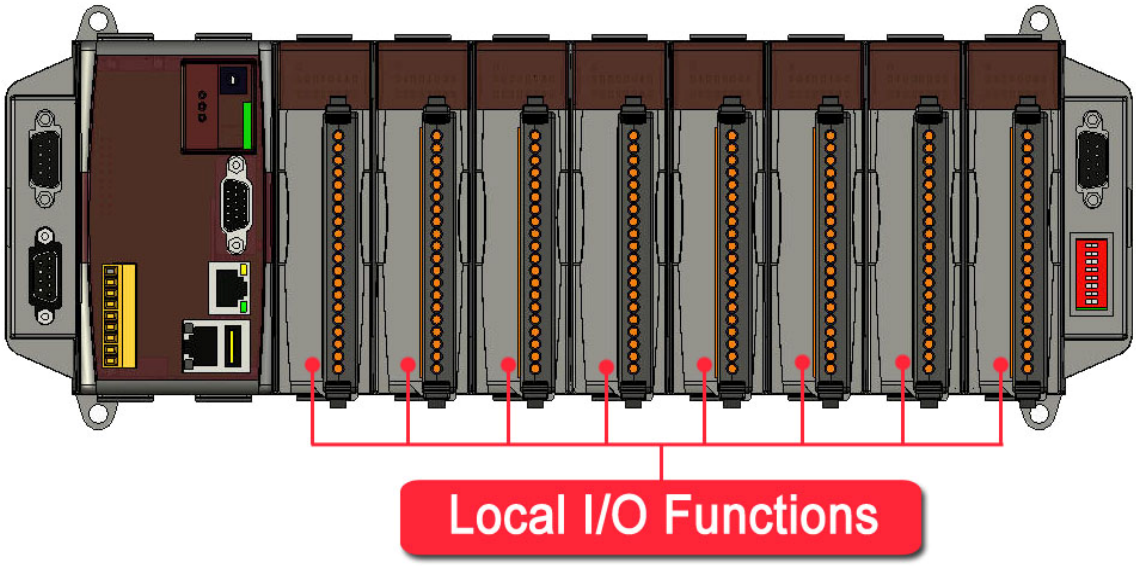
```
WinPAC.uart_SendCmd(hOpen, WinPAC.Misc.AnsiString("$00M"), buf);
```

5.3.8. Demo programs for PAC_IO

PAC_IO API supports to operate IO modules not only in slot but in remote mode.

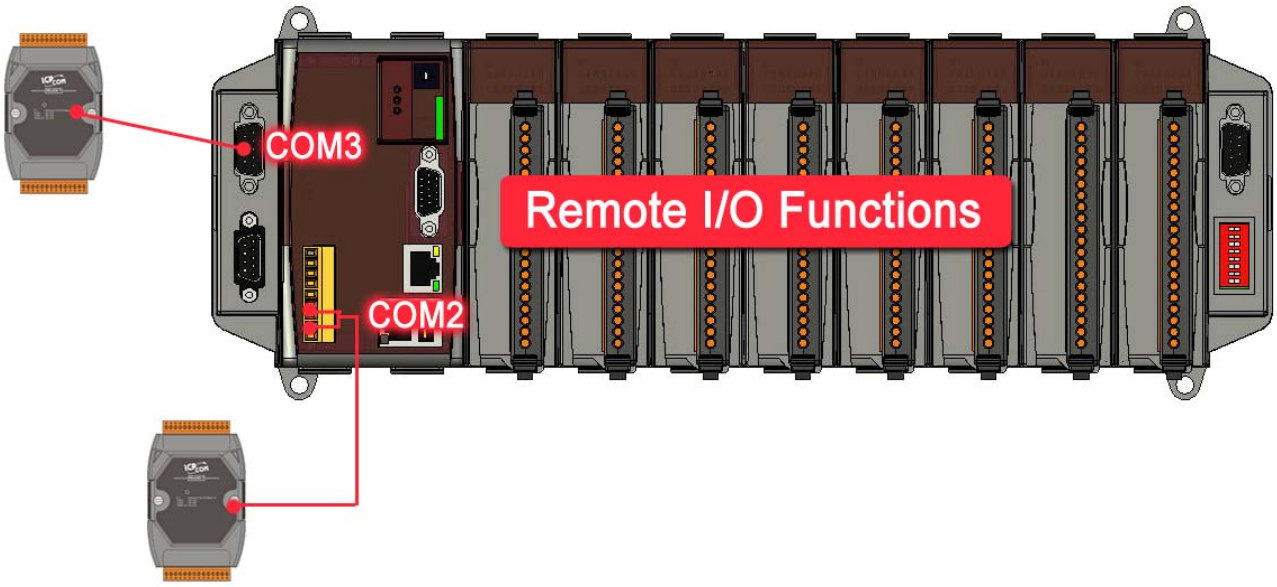
In the local mode, the slot range is from 0 to 7.

Local (IO in slot):



In the remote mode, the slot range is from 0 to 255.

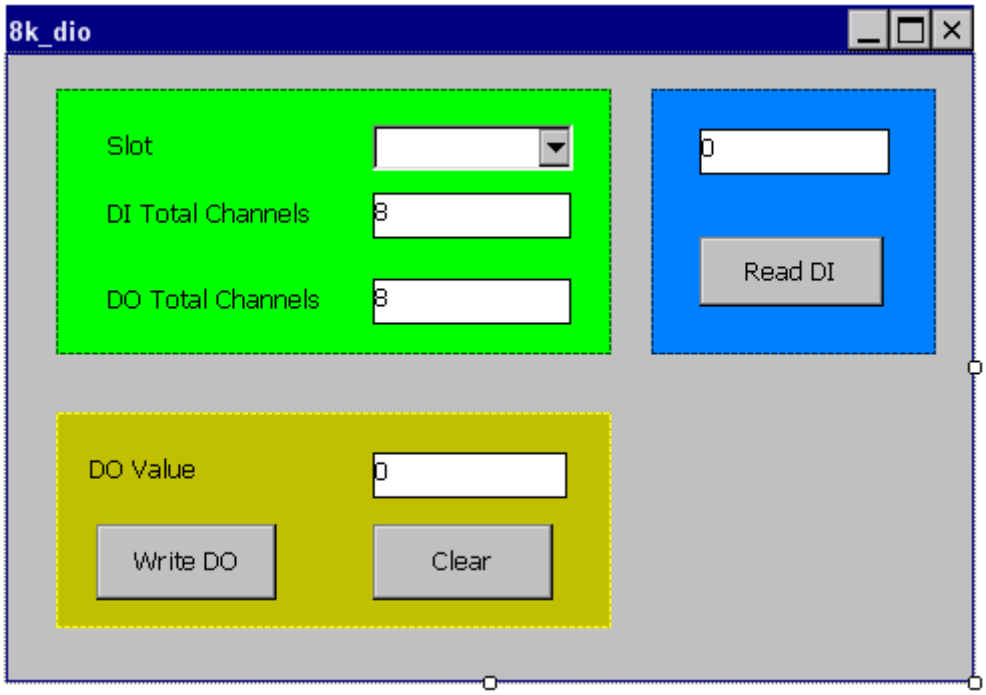
Remote:



How to use Local I/O functions

Here are some examples

Example 1: The example extract from “8k_dio” of WinPAC demos



Download the complete demo:

CD:\Napdos\wp-8x4x_ce50\demo\WinPAC\DotNET\C#\PAC_IO\Local\8k_dio\

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/dotnet/c#/pac_io/local/8k_dio/

The example illustrates how to:

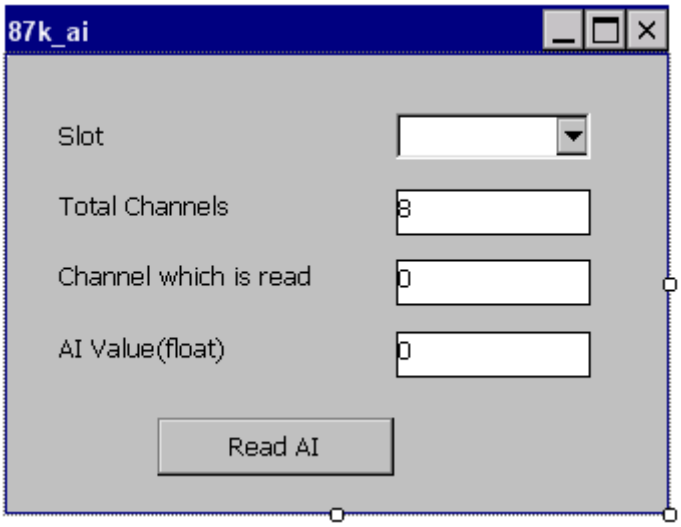
1. **Read the DI value of the DI module**

   WinPAC.PAC_IO.pac_ReadDI(IntPtr hPort, int slot, int iDI_TotalCh, ref unit lDI_Value);

2. **Write the DO value to the DO modules**

   WinPAC.PAC_IO.pac_WriteDO(IntPtr hPort, int slot, int iDO_TotalCh, unit lDO_Value);

Example 2: The example extract from “87k_ai” of WinPAC demos



Download the complete demo:

CD:\Napdos\wp-8x4x_ce50\demo\WinPAC\DotNET\C#\PAC_IO\Local\87k_ai\

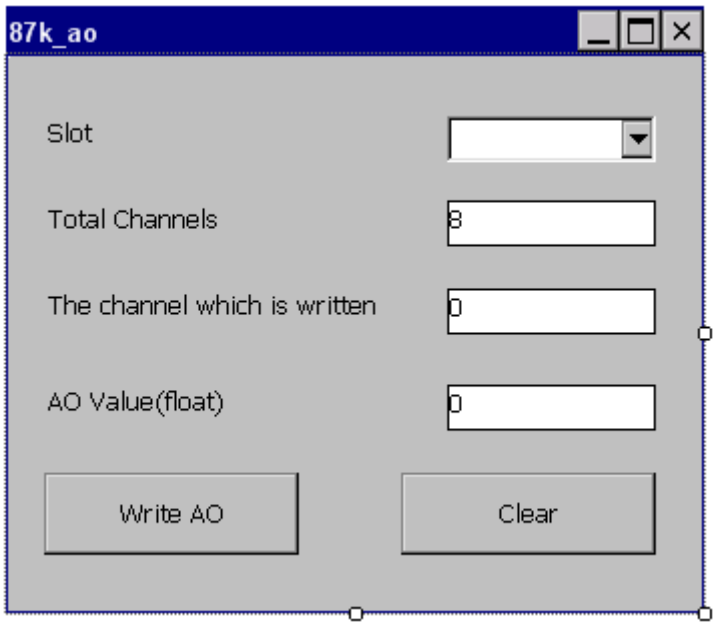
ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/dotnet/c#/pac_io/local/87k_ai/

The example illustrates how to:

Read the engineering-mode AI value of the AI module

WinPAC.PAC_IO.pac_ReadAI(hPort, iSlot,iChannel,iAI_TotalCh, ref fValue);

Example 3: The example extract from “87k_ao” of WinPAC demos



Download the complete demo:

CD:\Napdos\wp-8x4x_ce50\demo\WinPAC\DotNET\C#\PAC_IO\Local\87k_ao\

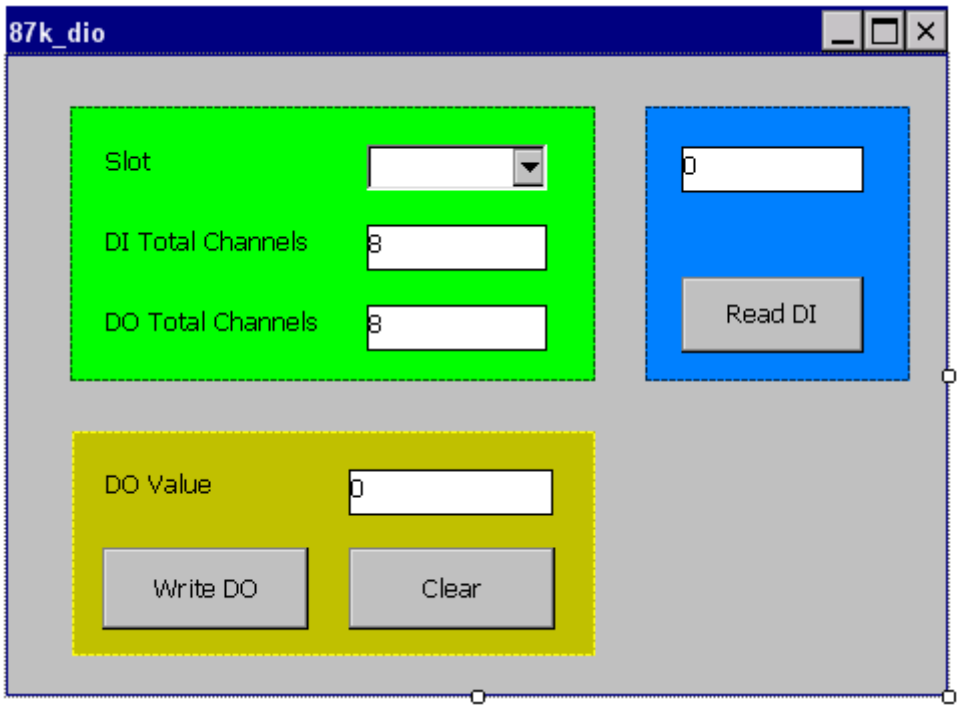
ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/dotnet/c#/pac_io/local/87k_ao/

The example illustrates how to:

Write the AO value to the AO modules

WinPAC.PAC_IO.pac_WriteAO(hPort, iSlot,iChannel,iAO_TotalCh,fValue);

Example 4: The example extract from “87k_dio” of WinPAC demos



Download the complete demo:

CD:\Napdos\wp-8x4x_ce50\demo\WinPAC\DotNET\C#\PAC_IO\Local\87k_dio\

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/dotnet/c#/pac_io/local/87k_dio/

The example illustrates how to:

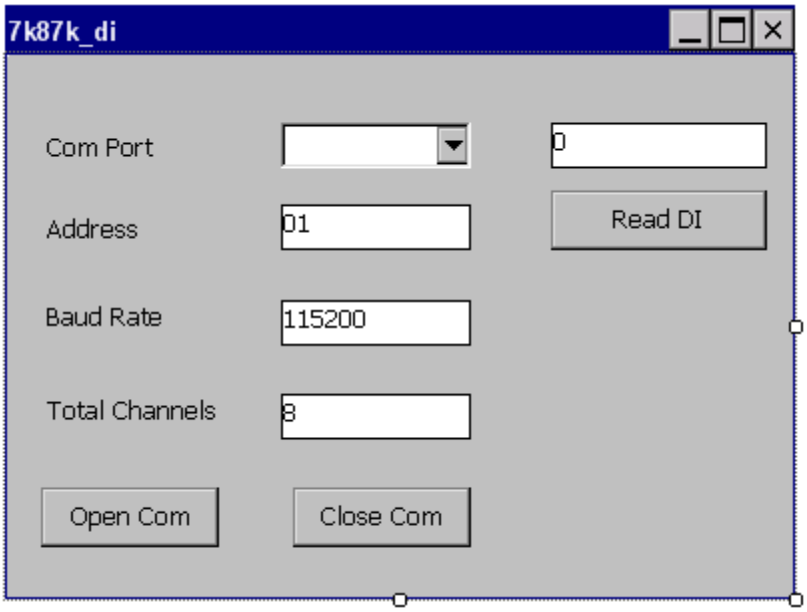
Read the DI and the DO values of the DIO module

WinPAC. pac_ReadDIO(IntPtr hPort, int slot, int iDI_TotalCh, int iDO_TotalCh, ref unit lDI_Value, ref unit lDO_Value);

How to use remote I/O functions

Here are some examples

Example 1: The example extract from “7k87k_di” of WinPAC demos



Download the complete demo:

CD:\Napdos\wp-8x4x_ce50\demo\WinPAC\DotNET\C#\PAC_IO\Remote\7K87k_di\

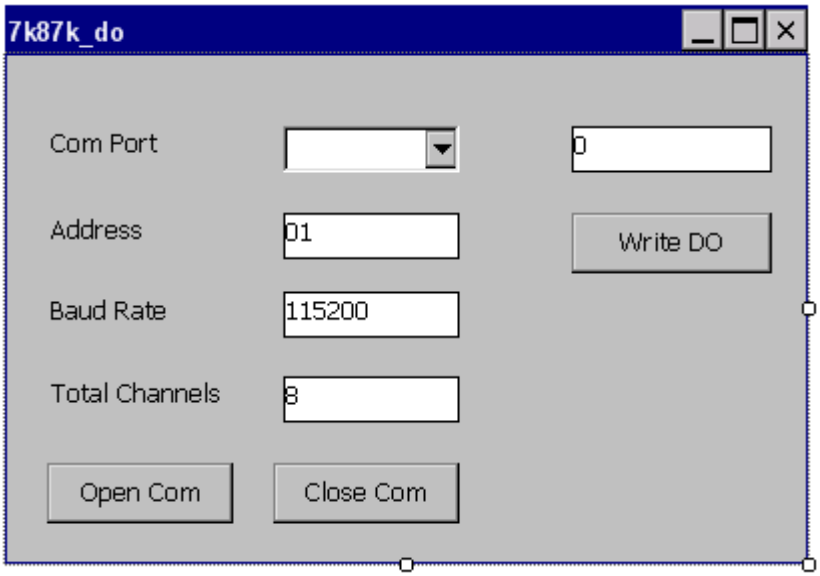
ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/dotnet/c#/pac_io/remote/7k87k_di/

The example illustrates how to:

Read the DI value of the DI module

WinPAC.PAC_IO.pac_ReadDI(IntPtr hPort, int PAC_REMOTE_IO (iAddr) , int iDI_TotalCh, ref unit lDI_Value);

Example 2: The example extract from “7k87k_do” of WinPAC demos



Download the complete demo:

CD:\Napdos\wp-8x4x_ce50\demo\WinPAC\DotNET\C#\PAC_IO\Remote\7k87k_do\

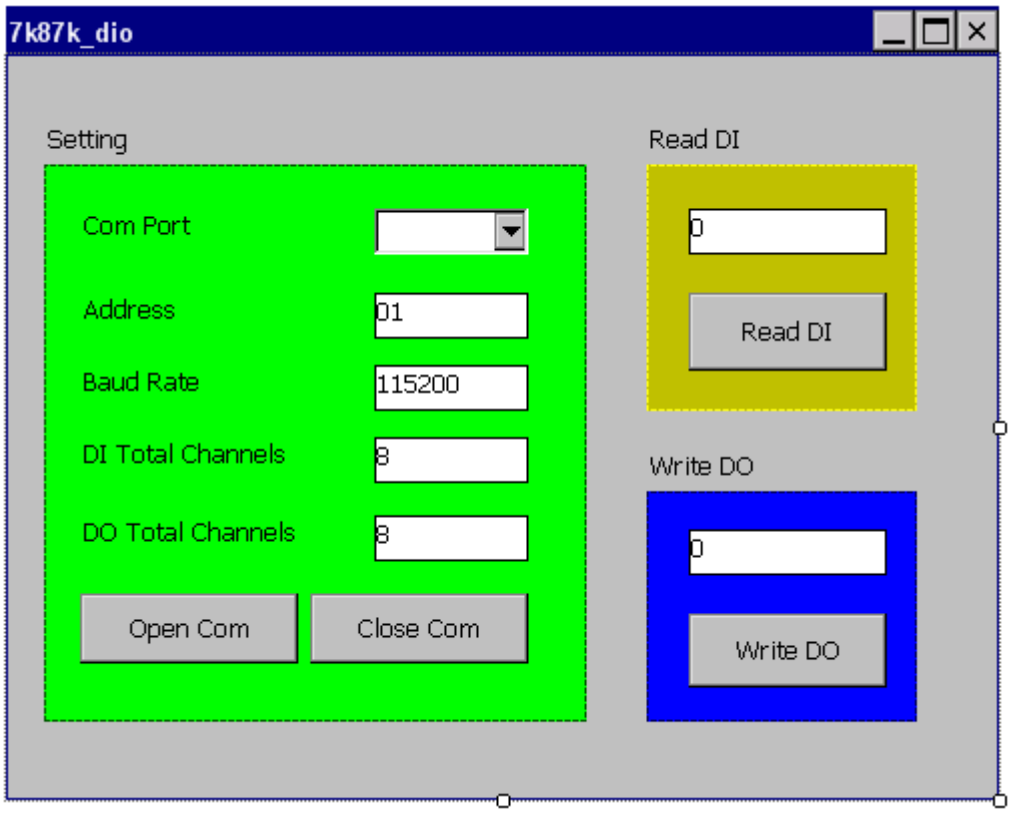
ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/dotnet/c#/pac_io/remote/7k87k_do/

The example illustrates how to:

Write the DO value to the DO modules

WinPAC.PAC_IO.pac_WriteDO(IntPtr hPort, int PAC_REMOTE_IO (iAddr) , int iDO_TotalCh, unit lDO_Value);

Example 3: The example extract from “7k87k_dio” of WinPAC demos



Download the complete demo:

CD:\Napdos\wp-8x4x_ce50\demo\WinPAC\DotNET\C#\PAC_IO\Remote\7k87k_dio\

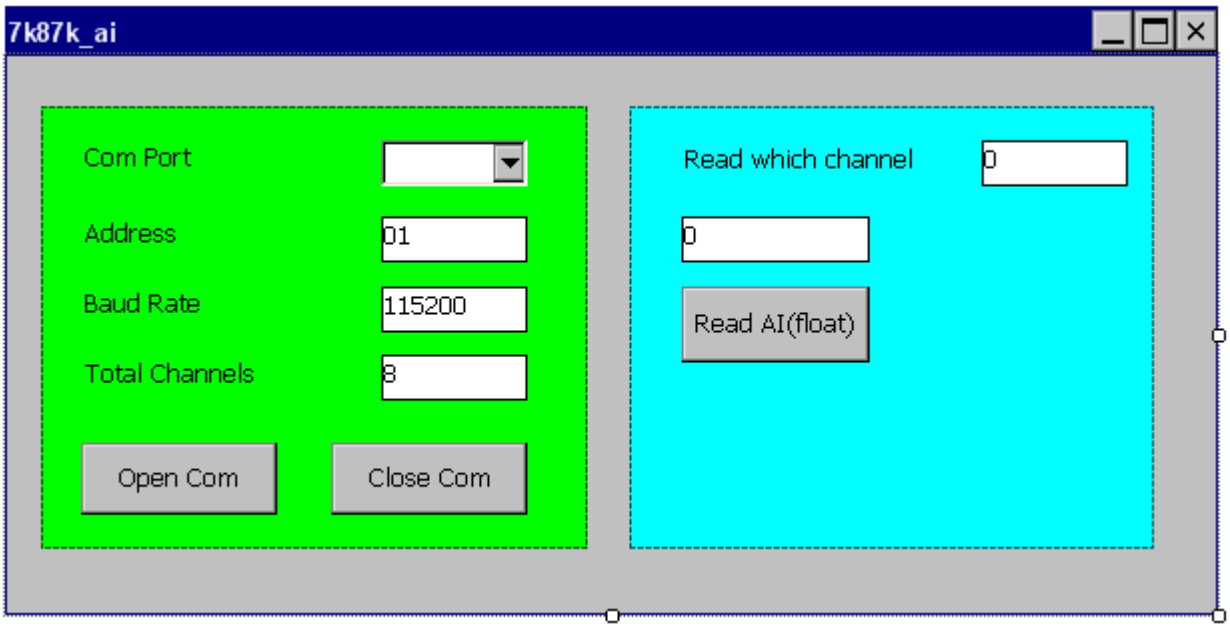
ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/dotnet/c#/pac_io/remote/7k87k_dio/

The example illustrates how to:

Read the DI and the DO values of the DIO module

WinPAC.PAC_IO.pac_ReadDIO(IntPtr hPort, int PAC_REMOTE_IO (iAddr) , int iDO_TotalCh, unit lDO_Value);

Example 4: The example extract from “7k87k_ai” of WinPAC demos



Download the complete demo:

CD:\Napdos\wp-8x4x_ce50\demo\WinPAC\DotNET\C#\PAC_IO\Remote\7k87k_ai\

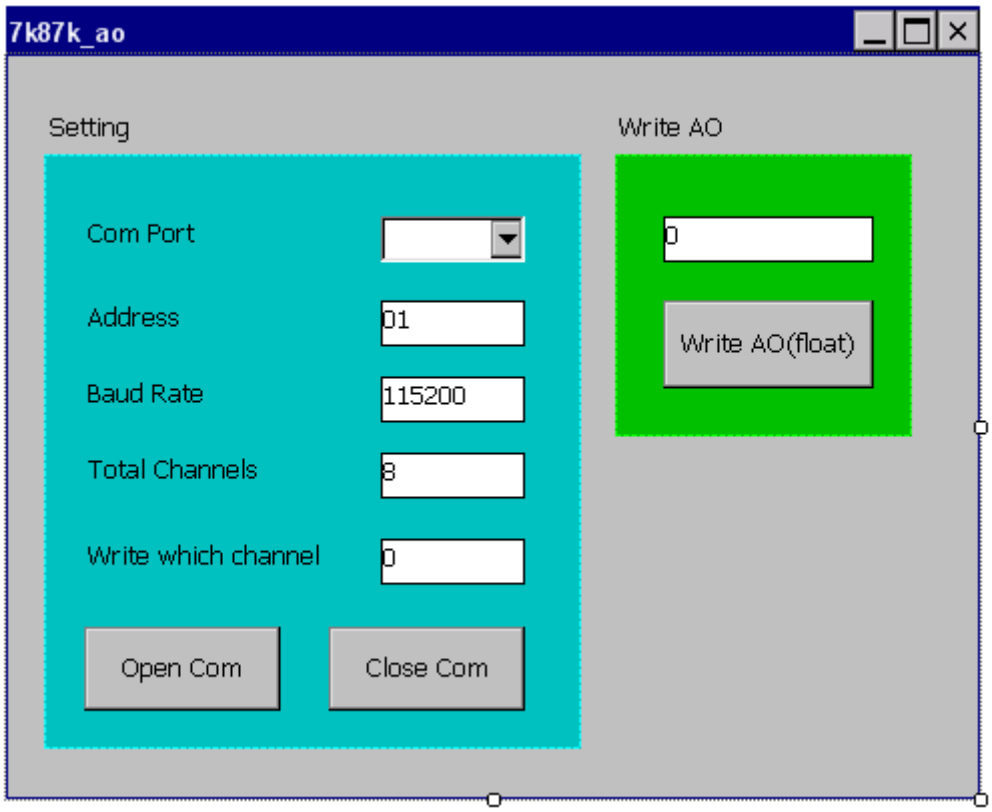
ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/dotnet/c#/pac_io/remote/7k87k_ai/

The example illustrates how to:

Read the engineering-mode AI value of the AI module

WinPAC.PAC_IO.pac_ReadAI(hPort, PAC_REMOTE_IO (iAddr) ,iChannel,iAI_TotalCh, ref fValue);

Example 5: The example extract from “7k87k_ao” of WinPAC demos



Download the complete demo:

CD:\Napdos\wp-8x4x_ce50\demo\WinPAC\DotNET\C#\PAC_IO\Remote\7k87k_ao\

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/dotnet/c#/pac_io/remote/7k87k_ao/

The example illustrates how to:

Write the AO value to the AO modules

WinPAC.PAC_IO.pac_WriteAO(hPort, PAC_REMOTE_IO (iAddr) ,

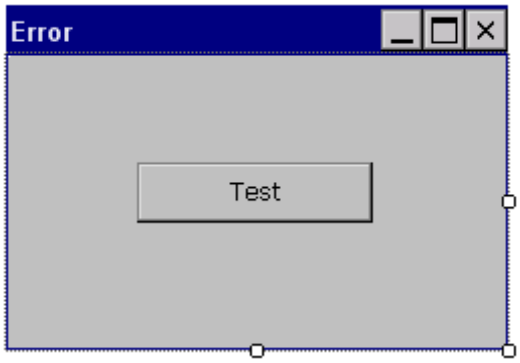
iChannel,iAO_TotalCh,fValue);

5.3.9. Demo programs for error handling

Error handling operations enable you to receive and display error information for your application.

How to use error handling

Here is an example which extract from “Error” of WinPAC demos



Download the complete demo:

CD:\Napdos\wp-8x4x_ce50\demo\WinPAC\DotNET\C#\Other\Error\

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/dotnet/C#/other/error/

The example illustrates how to:

1. **Return the last-error code value**

   pac_GetLastError();

2. **Retrieve an error message string**

   pac_GetErrorMessage(unit dwMessageID, string lpBuffer);

APPENDIX

A

Frame Ground

Electronic circuits are constantly vulnerable to Electro-Static Discharge (ESD), which become worse in a continental climate area. Some I-7000 ,M-7000 and I-8000 series modules feature a new design for the frame ground, which provides a path for bypassing ESD, allowing enhanced static protection (ESD) capability and ensures that the module is more reliable.

The following options will provide a better protection for the module:

The I-8000 controller has a metallic board attached to the back of the plastic basket as shown in the Figure 2-1 below. When mounted to the DIN rail, connect the DIN rail to the earth ground because the DIN rail is in contact with the upper frame ground as shown in the Figure 2-2 below.

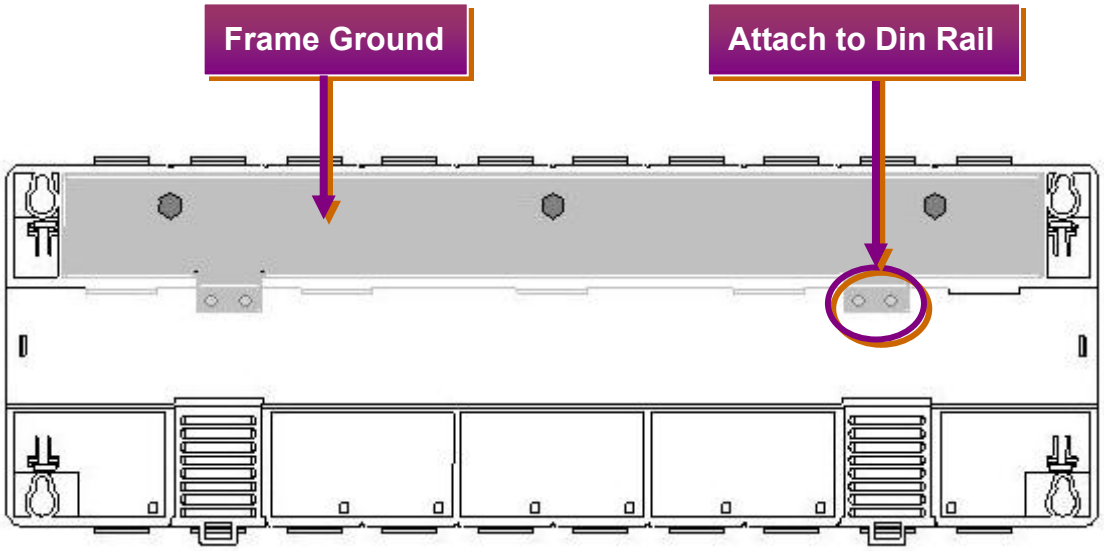


Figure B-1

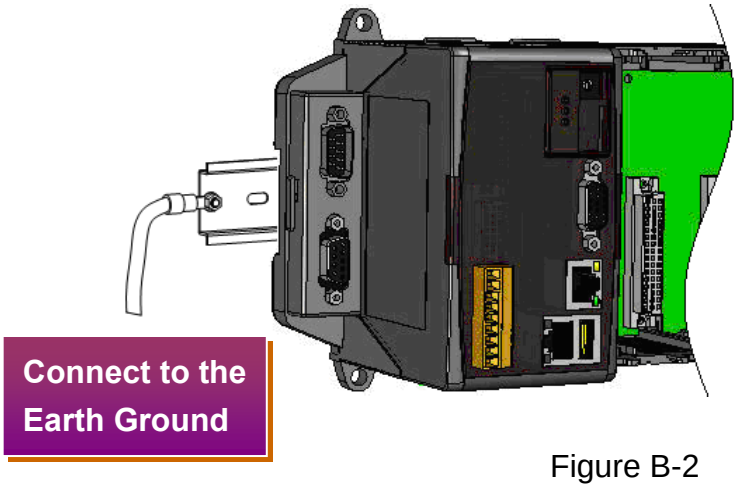


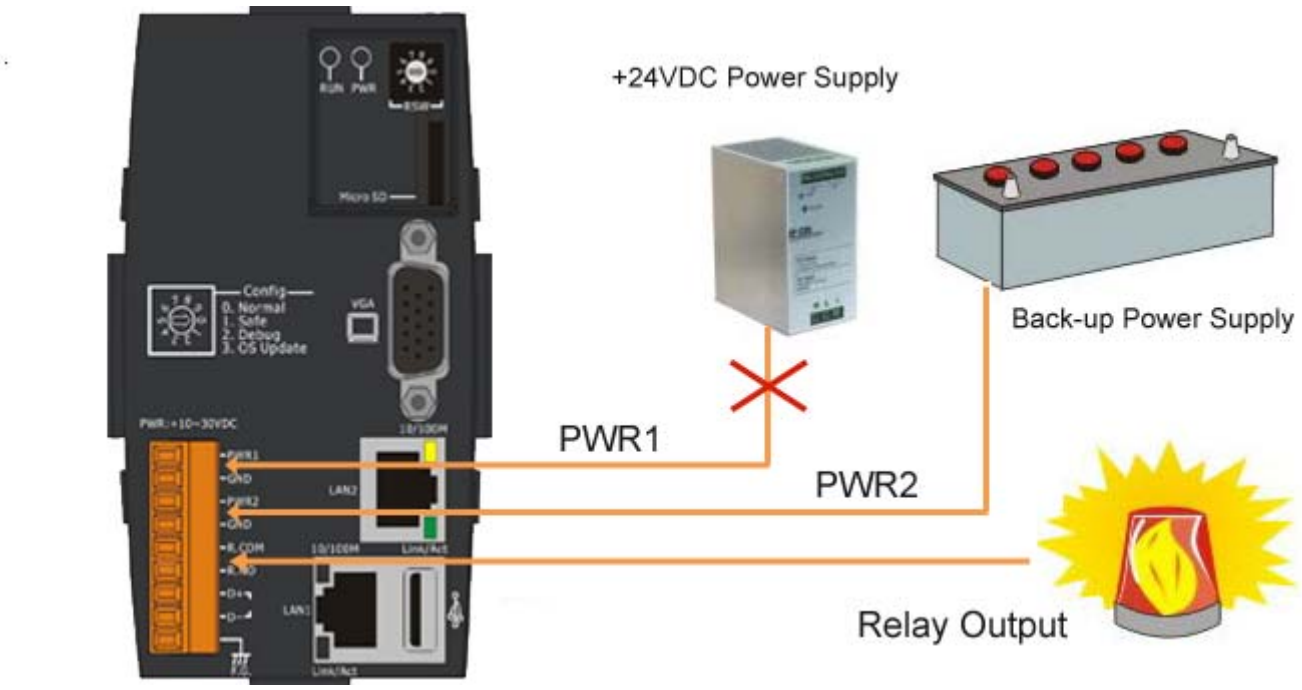
Figure B-2

APPENDIX B

Redundant Power

The WinPAC-8000 provides two power inputs that can be connected simultaneously to live DC power sources. If one of the power inputs fails, the other live source acts as a backup to automatically support the the WinPAC-8000's power needs.

The WinPAC-8000 provides relay contact outputs to warn technicians on the shop floor when the power fails.



APPENDIX

C

I-8K and I-87K modules

There are 1/4/8 slot options to expand local I/O. And the I/O modules can be parallel bus type (high profile I-8K series) and serial bus type (high profile I-87K series). The difference between them is

Item	I-8K Series	I-87K Series
Microprocessor	No	Yes (8051)
Communication interface	Parallel bus	Serial bus
Communication speed	Fast	Slow
DI latched function	No	Yes
Counter input (for digital input module)	No	Yes (100 Hz)
Power on value	No	Yes
Safe value	No	Yes
Programmable slew-rate for AO module	No	Yes

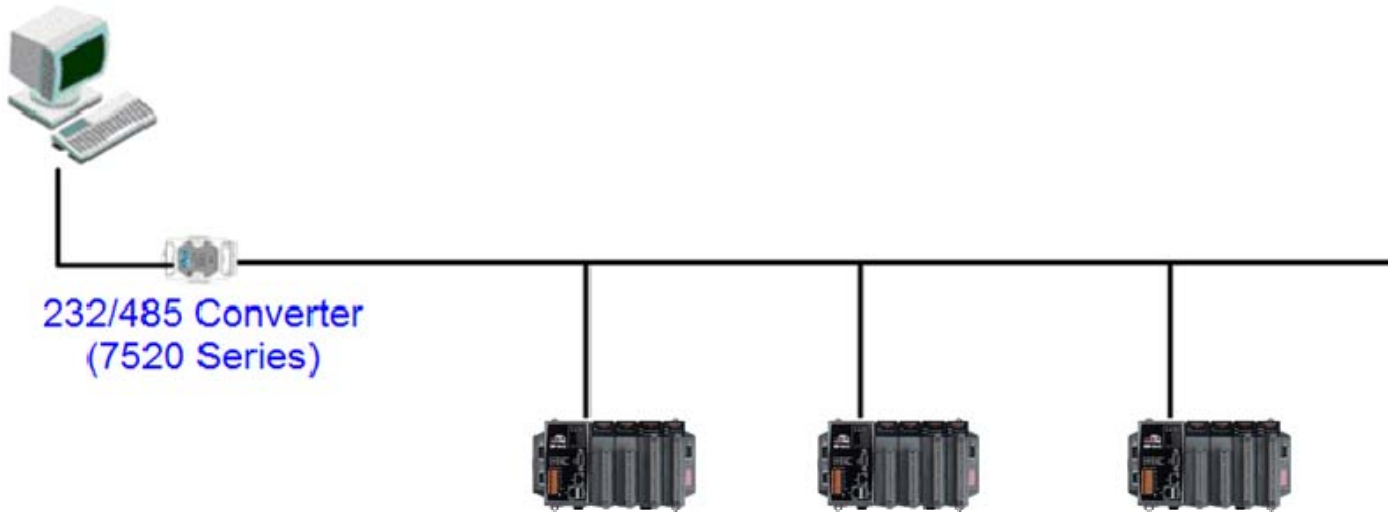
APPENDIX D

Application of RS-485 Network

The RS-485 length can be up to 4000 ft or 1.2 km over a single set of twisted–pair cables, if the RS-485 network is over 4000 ft or 1.2Km, the RS-485 repeater must be added to extend the RS-485 network.

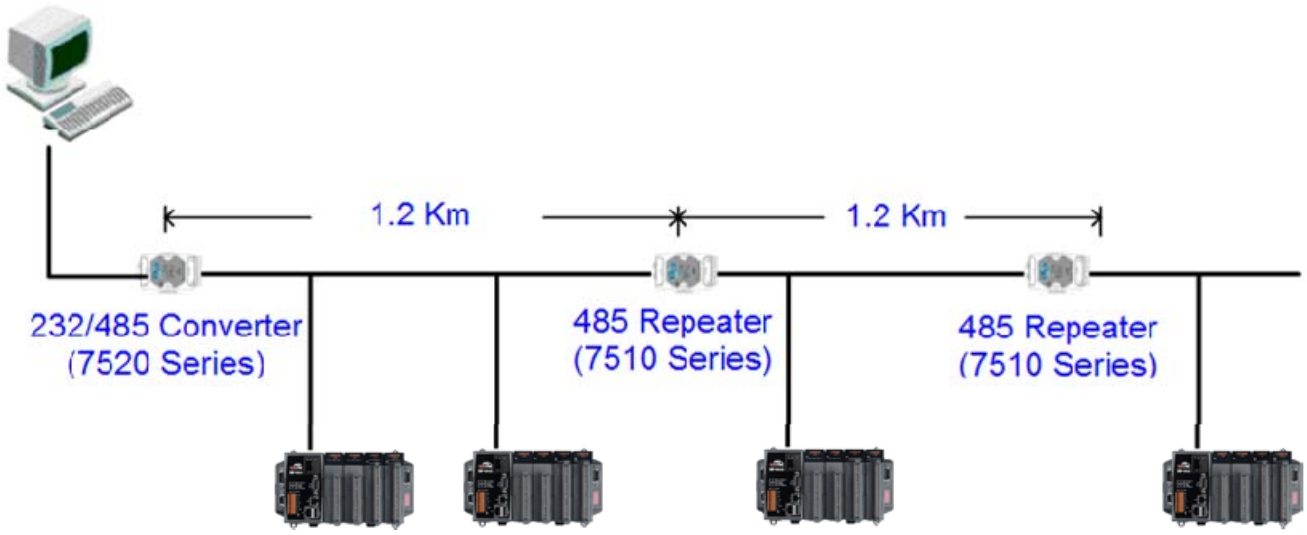
D.1. Basic RS-485 Network

The basic component of the RS-485 network consist of a Master Controller (or using a PC as a host controller), and some RS-485 devices.



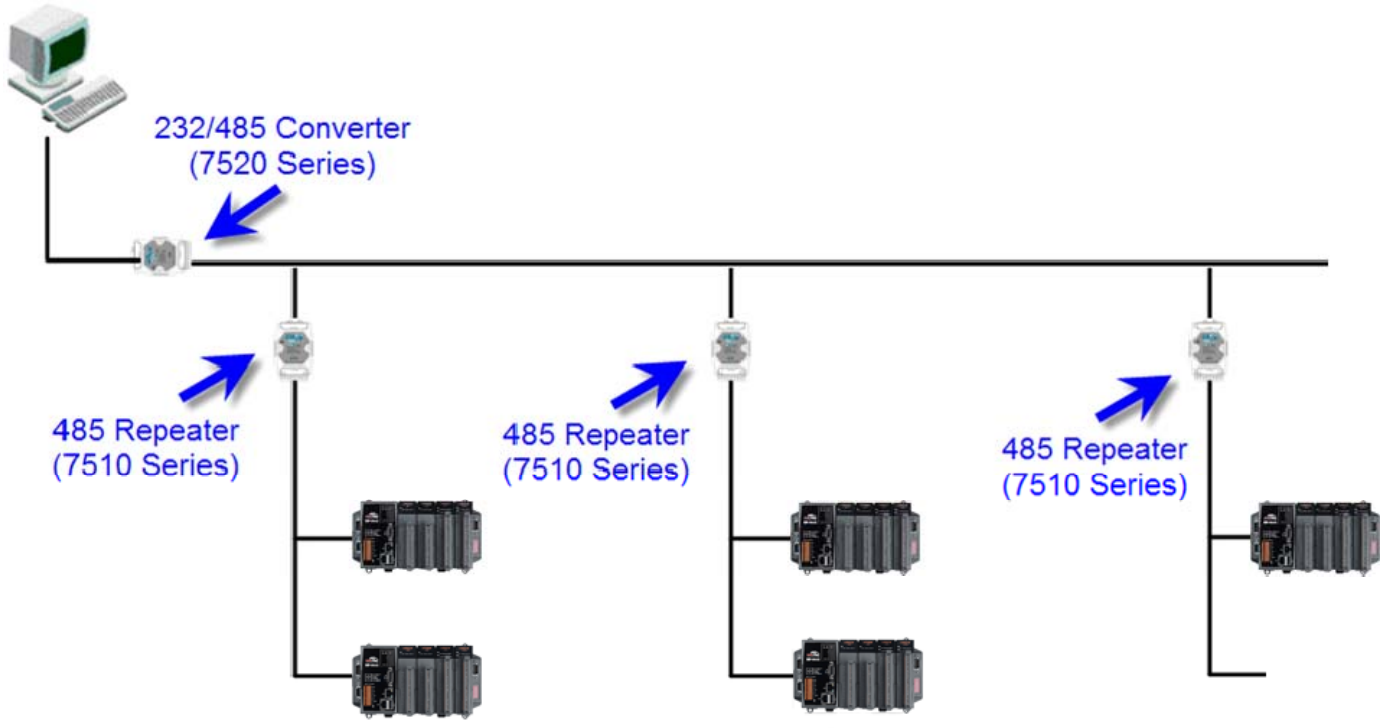
D.2. Daisy Chain RS-485 Network

All RS-485 devices are wired directly to the main network, If the network is up to 1.2 Km, it will need a repeater (7510 series) to extend the network length.

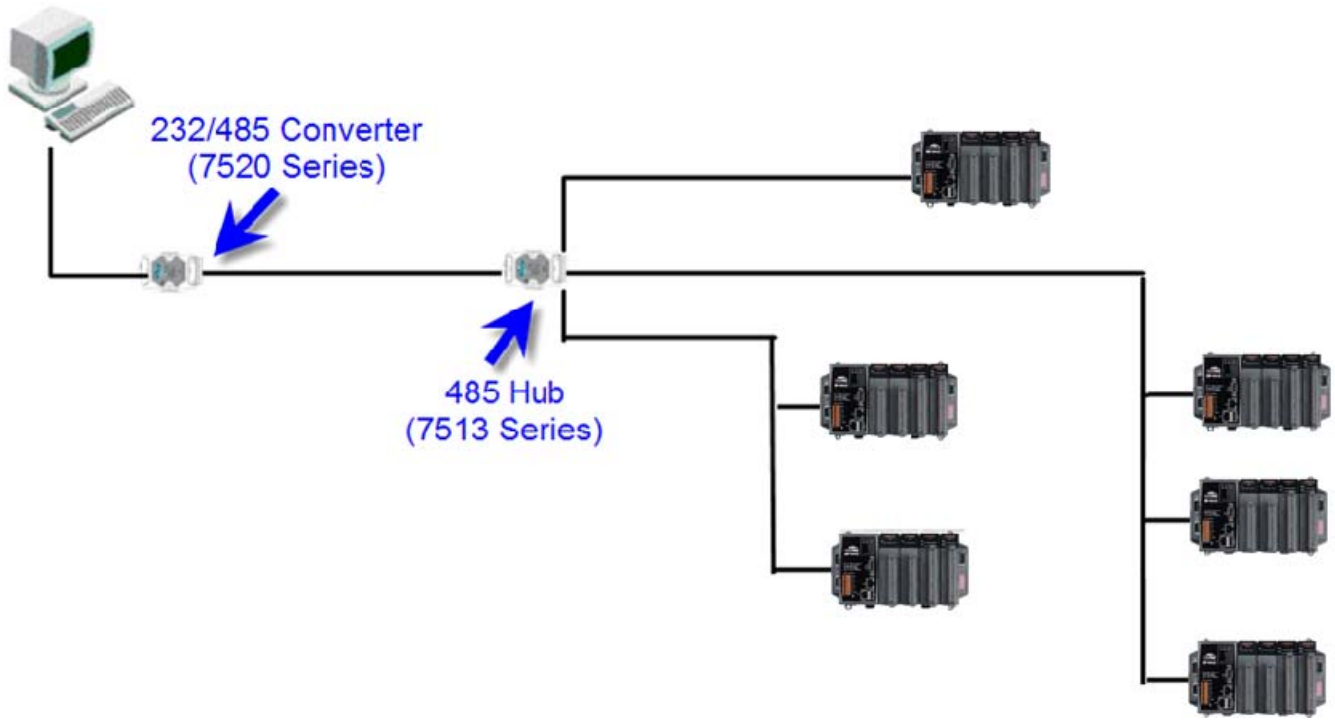


D.3. Star Type RS-485 Network

There are branches along the main network. In this case, it is better to have a repeater to isolate or filter the noise that is made by devices.

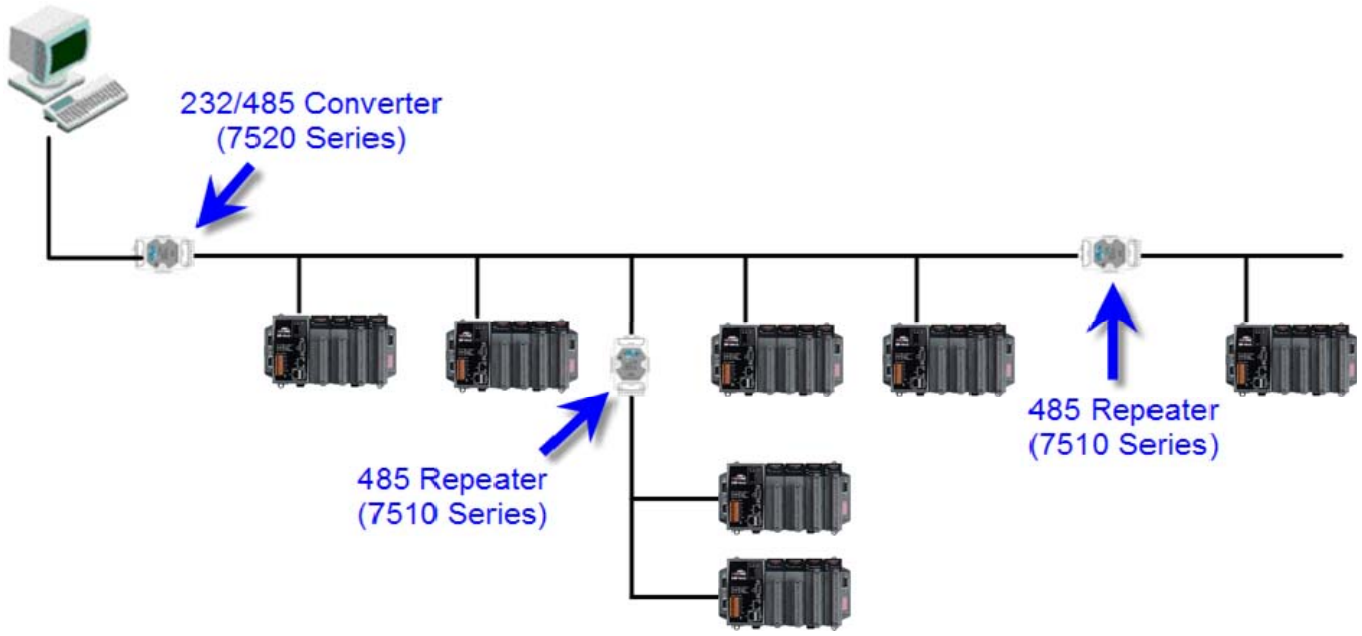


There is a better choice to use 7513 as a RS-485 hub on start type network.



D.4. Random RS-485 Network

There are branches along the main wire. In this case, it is better to have a repeater to isolate or filter the noise that is made by devices.



D.5. Master and Slave configuration

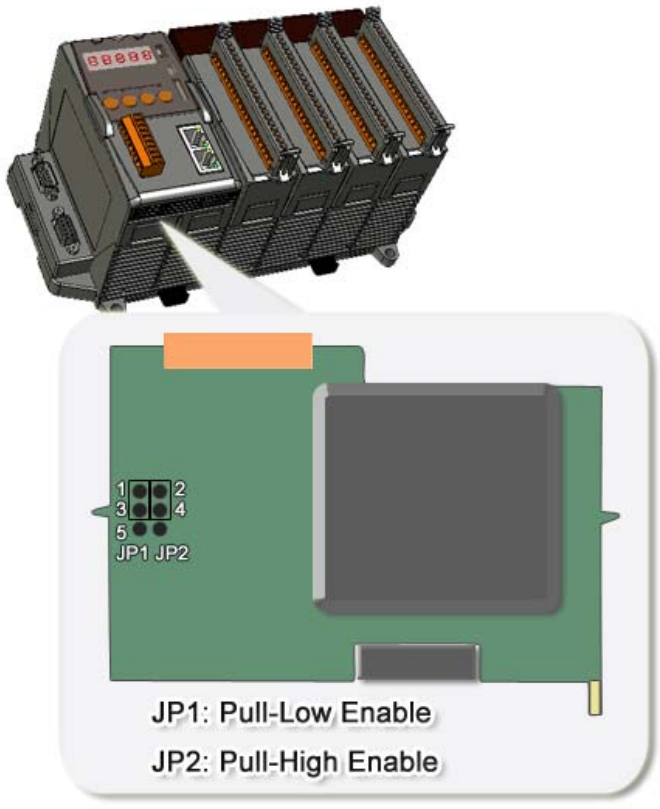
There must be having one master to have a pull-high/pull-low resistor in the same network.

In a master/slave applications, ”Master” is the default configuration of WinPAC-8000.

D.5.1. WinPAC-8000 as a slave

For most of application, when using one 7520 series as RS-232/485 converter, its pull-high/pull-low resistors are set to enabled. Then the WinPAC-8000 and all the other devices on this network must be slave mode (the pull-high/pull-low resistors must be disabled).

Please refer to the figure D-1 to for the jumpers’ setting of the pull-high/pull-low resistors which are located at the power board of WinPAC-8000.



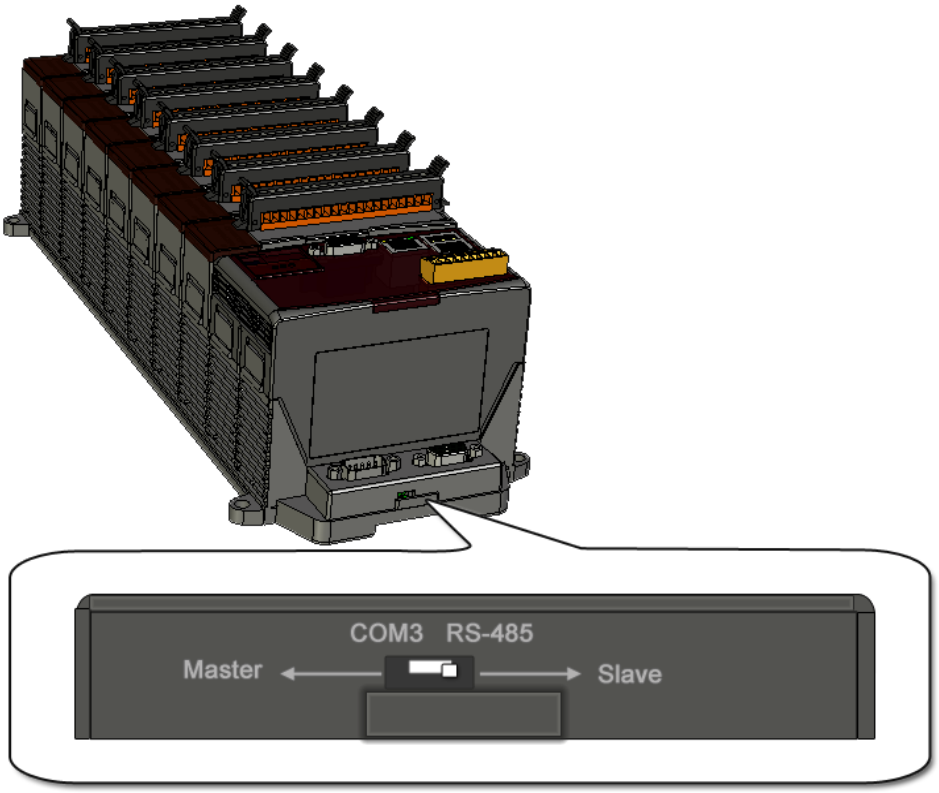
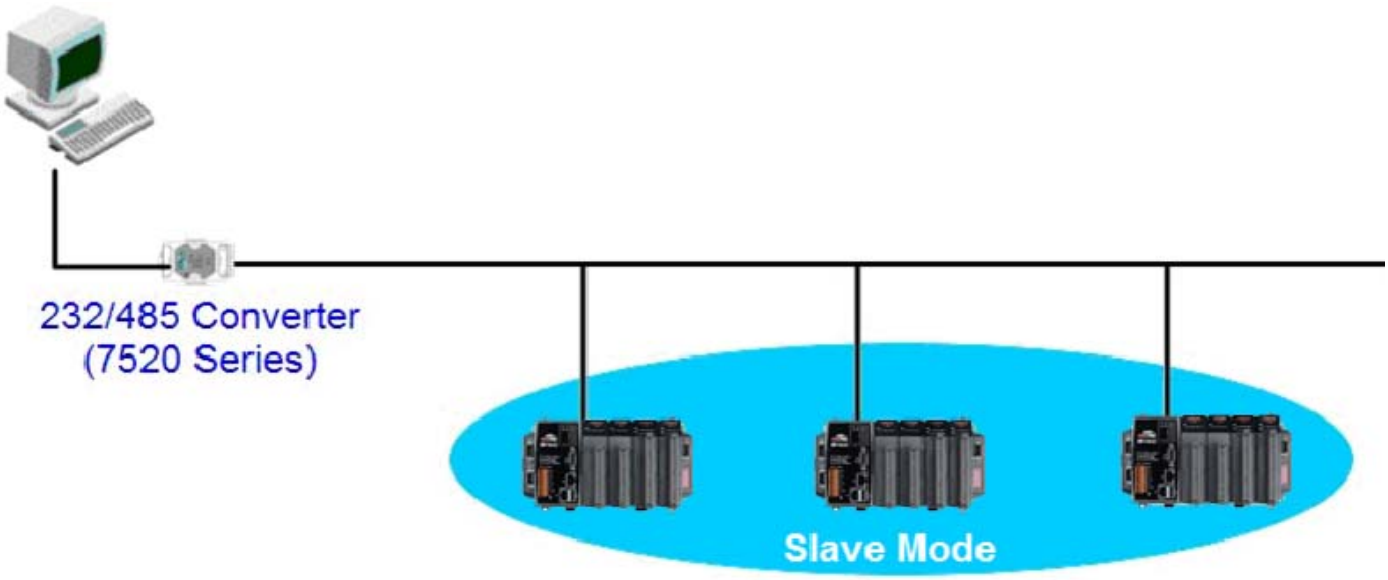
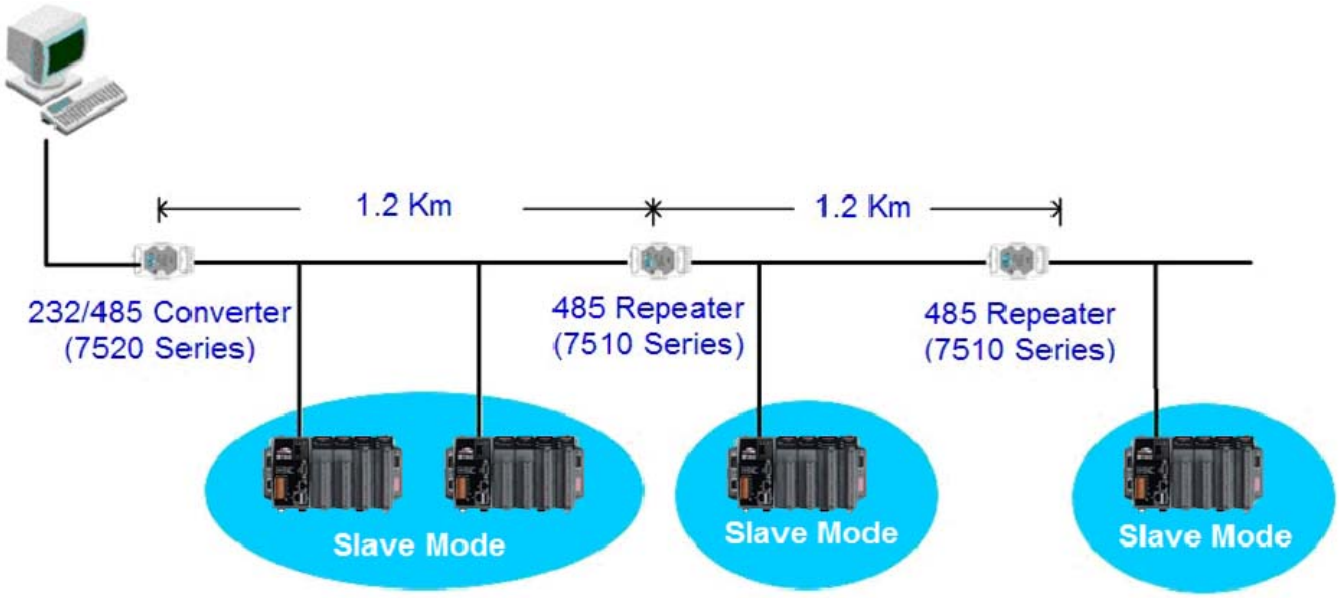


Figure D-1



If there are repeaters on the RS-485 network, there will be pull-high/pull-low resistors on both sides of the repeaters (i-7510)


1.2 Km
1.2 Km
232/485 Converter
(7520 Series)
485 Repeater
(7510 Series)
485 Repeater
(7510 Series)
Slave Mode
Slave Mode
Slave Mode

D.5.2. WinPAC-8000 as a Master (Default)

When one of WinPAC-8000 is set to master, then all the other devices on the same network must be slave mode. then the master one's (WinPAC-8000) pull-high/pull-low resistors have to adjusted to enabled.

Please refer to the Figure D-2 to for the jumpers' setting of the pull-high/pull-low resistors which are located at the power board oWinPAC-8000.

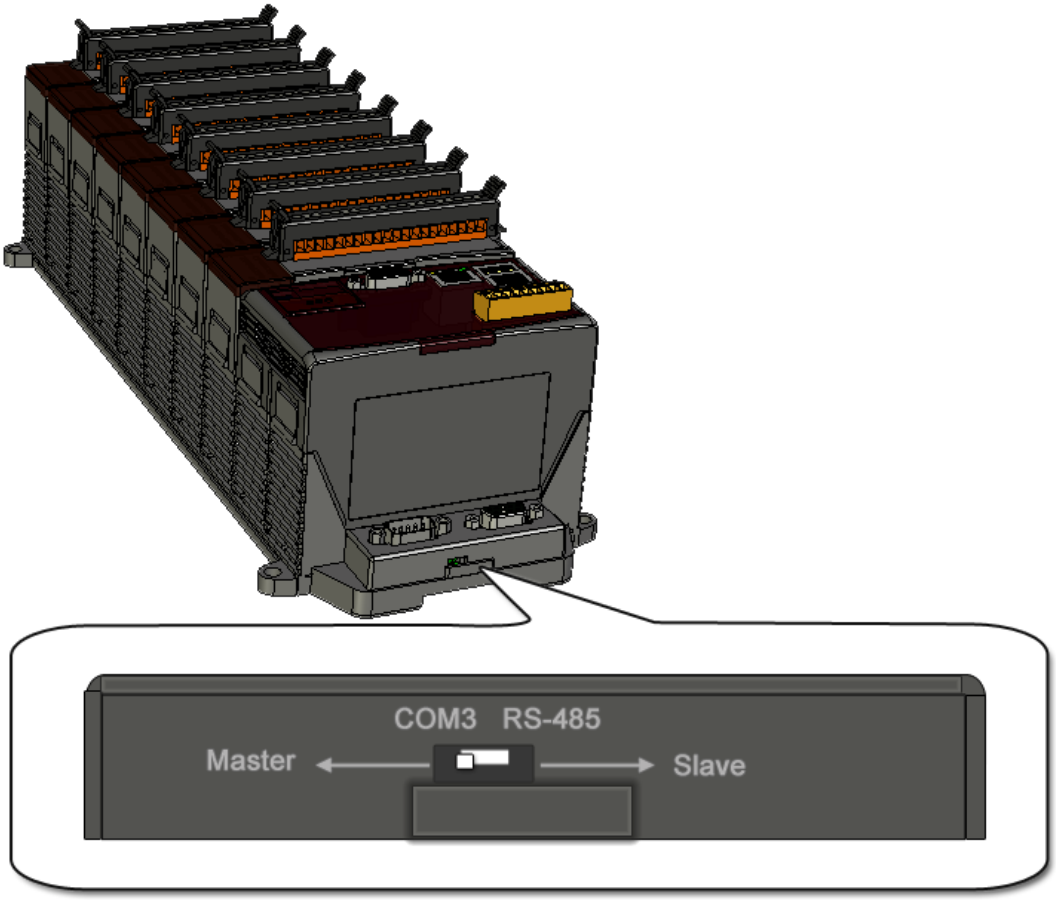
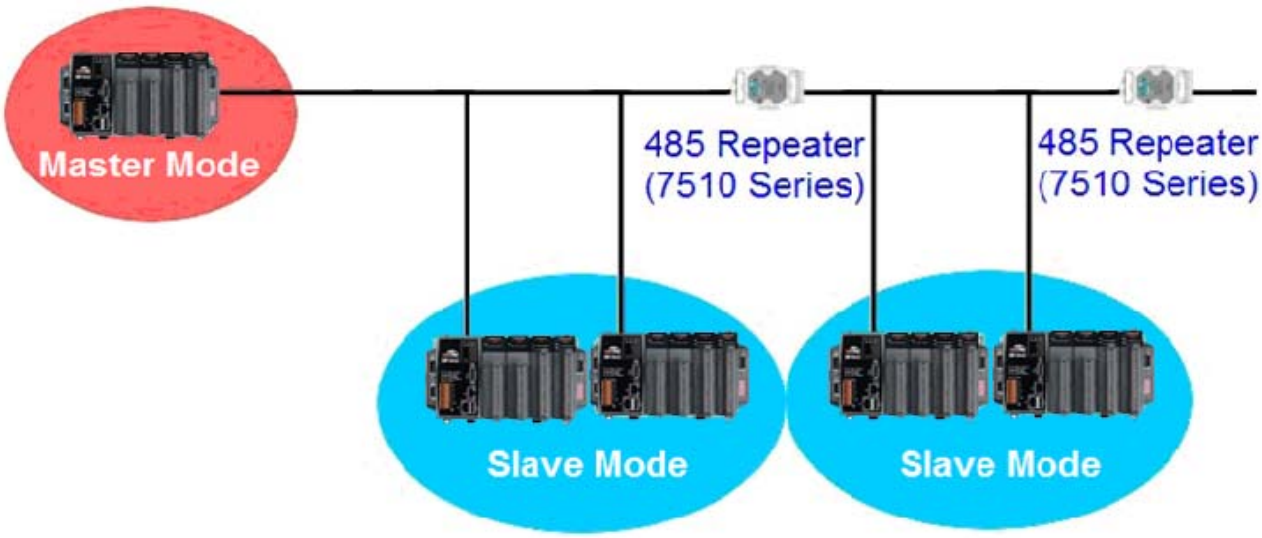


Figure D-2


Master Mode
485 Repeater
(7510 Series)
485 Repeater
(7510 Series)
Slave Mode
Slave Mode

APPENDIX

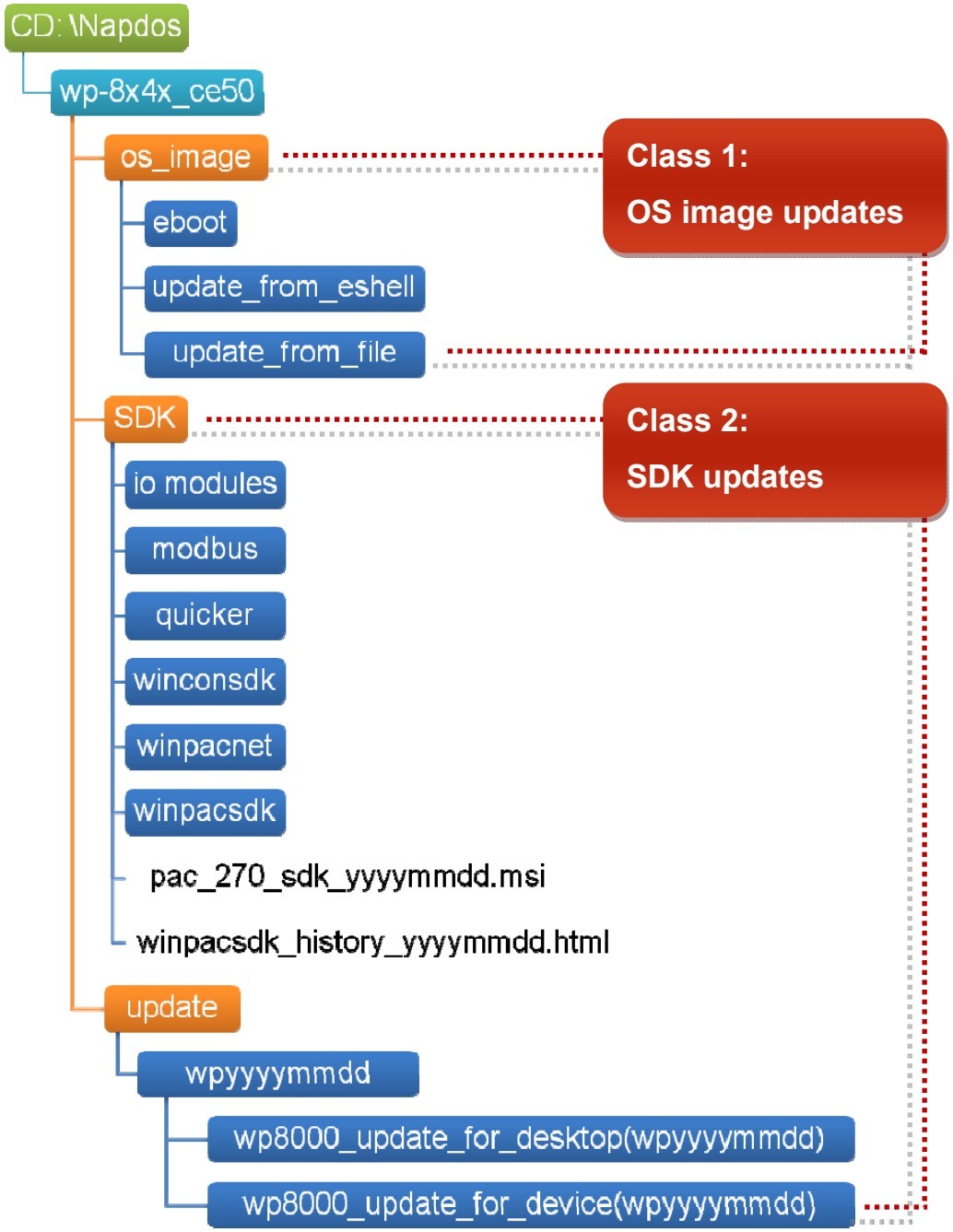
E

Tips - How to

E.1. How to upgrade for WinPAC-8000

ICP DAS will continue to add additional features to WinPAC-8000 SDK and OS in the future, so we advise you to periodically check the ICP DAS web site for the latest updates to WinPAC-8000 SDK and OS.

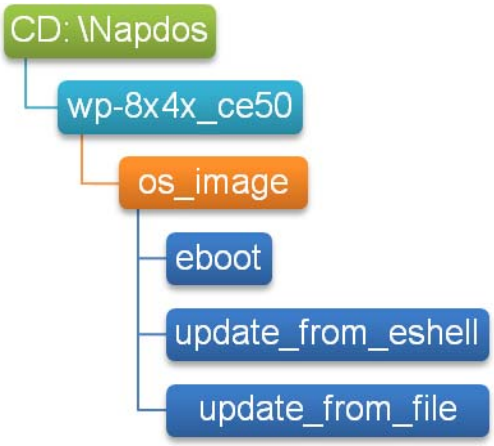
WinPAC-8000 upgrades that can be divided into the following two main classes:



Class 1. OS image updates

The update files of OS image are located in:

CD:\Napdos\wp-8x4x_ce50\os_image

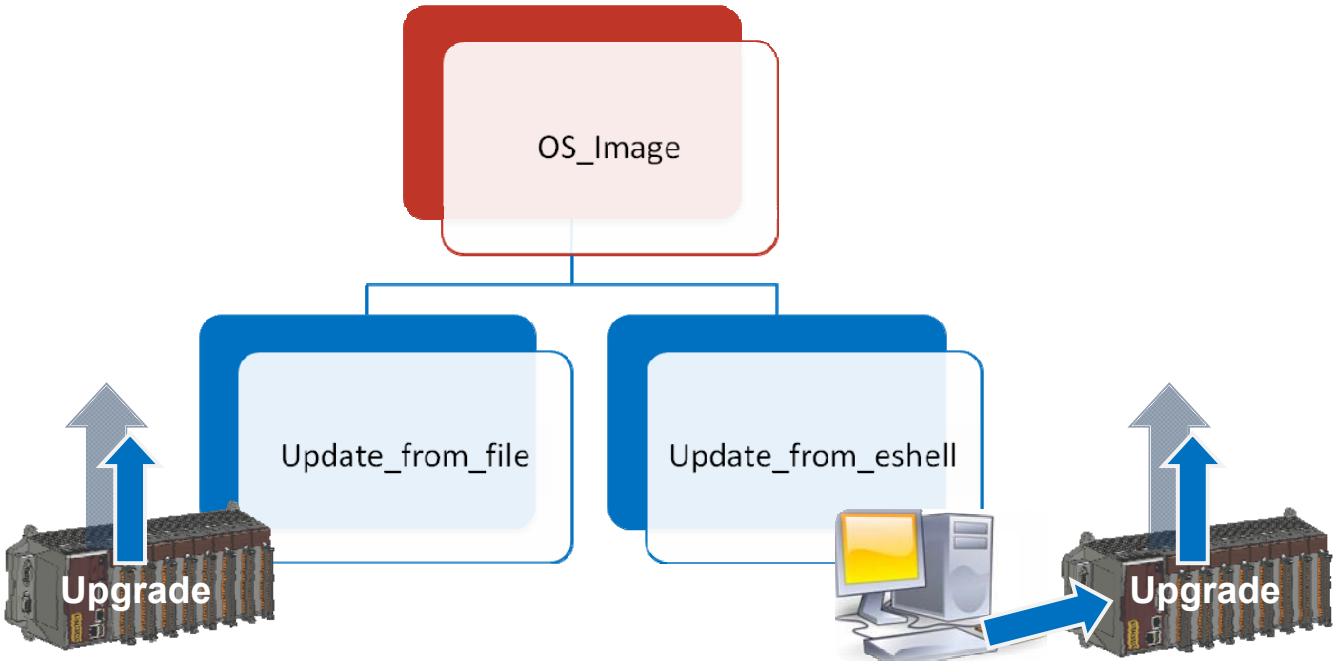


There are two different ways of WinPAC-8000 OS image update:

i. Update from file

(We recommend that you use this method for quick and easy to update the WinPAC-8000 OS image)

ii. Update from eshell

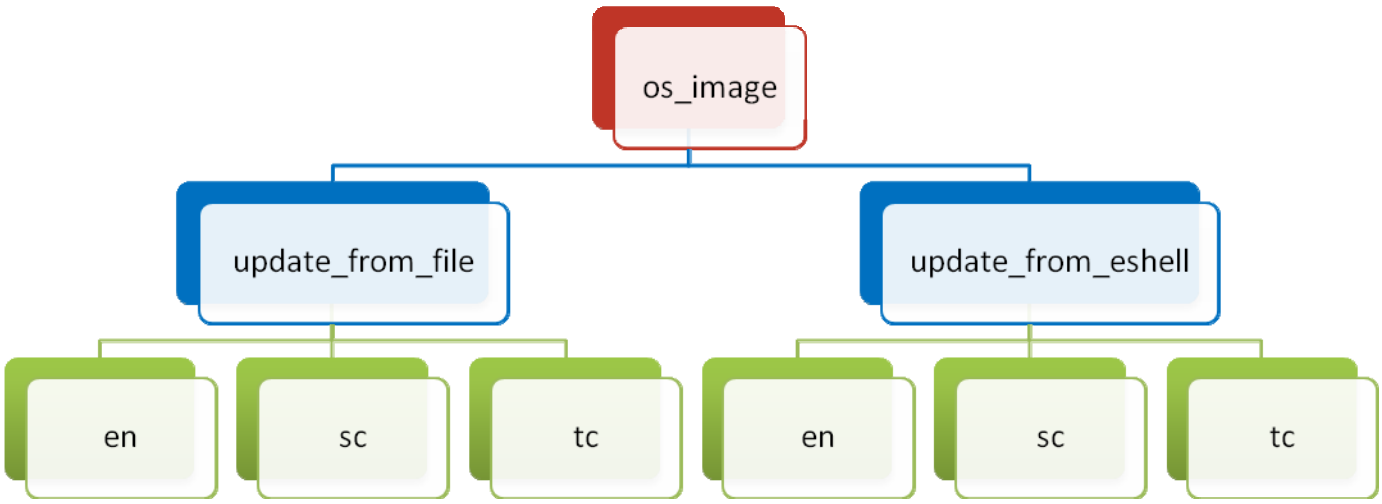


The WinPAC-8000 OS supports multi-language:

i. en- English

ii. sc- Simplified Chinese

iii. tc- Traditional Chinese



The following section provides procedures related to the update of WinPAC-8000 OS:

E.1.1. How to update the Boot Loader and the WinPAC-8000 OS image from files

E.1.2. How to update the WinPAC-8000 OS image from eshell

Eshell can be used to update the WinPAC-8000 OS image, but cannot be used to directly change OS language. The following section provides procedures related to change the OS language using eshell tool:

E.1.2.1. How to change the OS language

Class 2. SDK updates

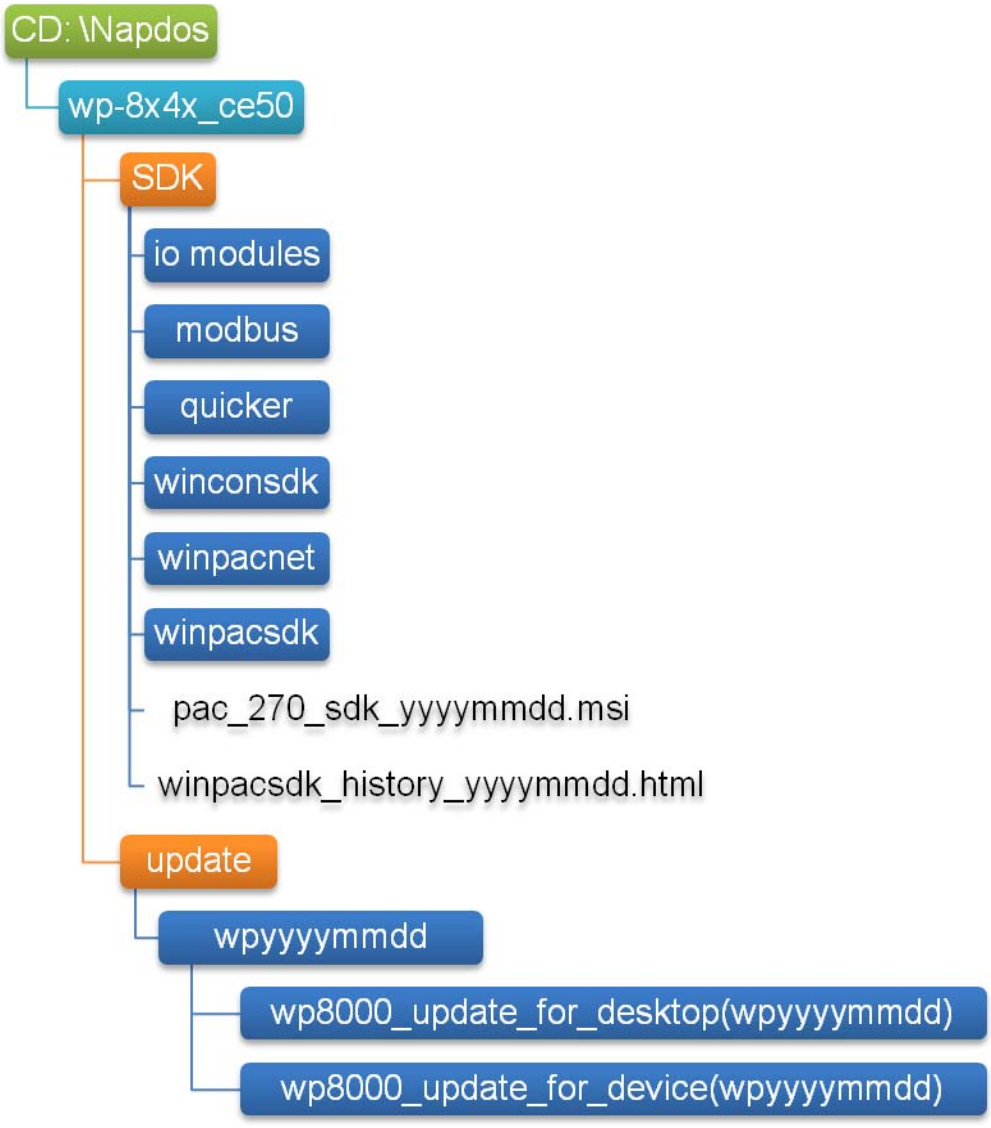
The WinPAC-8000 SDK is divided into the following two folders which contains various components related to the WinPAC-8000 SDK:

i. SDK

This folder contains the latest version of the WinPAC-8000 components related to the WinPAC-8000 SDKs or APIs.

ii. Update

This folder contains all-in-one update package that is easy to use and offers a step-by-step wizard to update the latest version of the WinPAC-8000 SDK.



The following section provides information on SDK updates:

E.1.3. How to install the WinPAC-8000 SDK update package

E.1.4. How to manually update the WinPAC-8000 SDK

E.1.1 How to update the Boot Loader and the WinPAC-8000 OS image from files

There are two different ways of WinPAC-8000 OS image update:

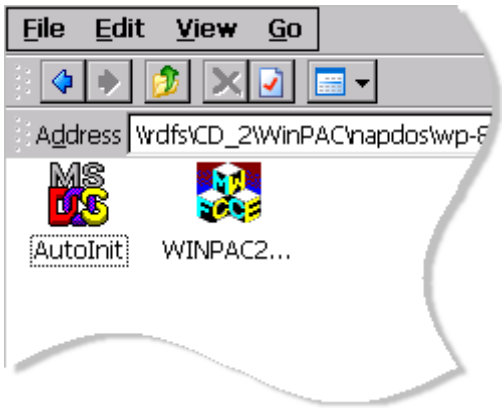
i. Updates from files (Please refer to this section)

We recommend that you use this method for quick and easy to update the WinPAC-8000 OS image

ii. Updates from eshell tool (Please refer to section “E.1.2. How to update the WinPAC-8000 OS image from eshell”)

Step 1: Get the latest version of the execute file and the corresponding “autoinit.bat” file and run it on the WinPAC-8000 controller side

Each folder contains an execute file and a corresponding “autoinit.bat” file.



Download these file to the WinPAC-8000, you can:

i. On the WinPAC-8000 controller side, upload files from ICP DAS FTP via an Ethernet connection.

ii. On the Host PC, upload files to the WinPAC-8000 via FTP

iii. On the Host PC, copy the file to Micro SD, USB Disk or SRAM.

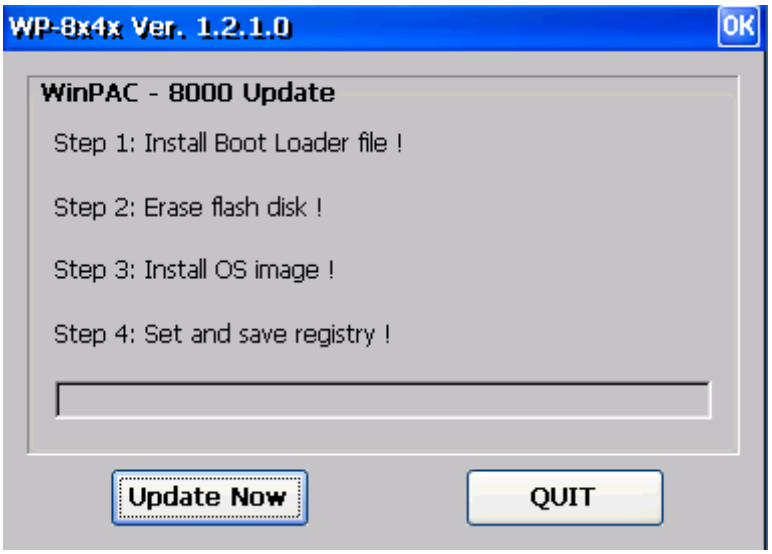
The latest version of the WinPAC-8000 OS image file can be obtained from:

CD:\Napdos\wp-8x4x_ce50\OS_image\update_from_file\

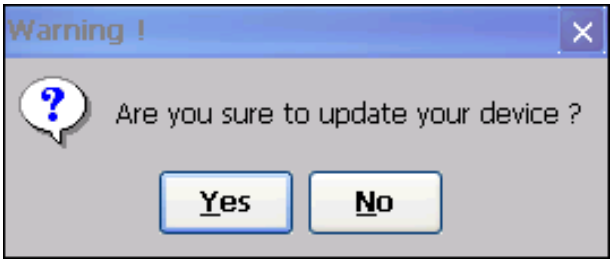
http://www.icpdas.com/products/PAC/winpac/download/winpac_8000/download_os_images.htm

Step 2: On the "Warning !" dialog, click the "OK" button

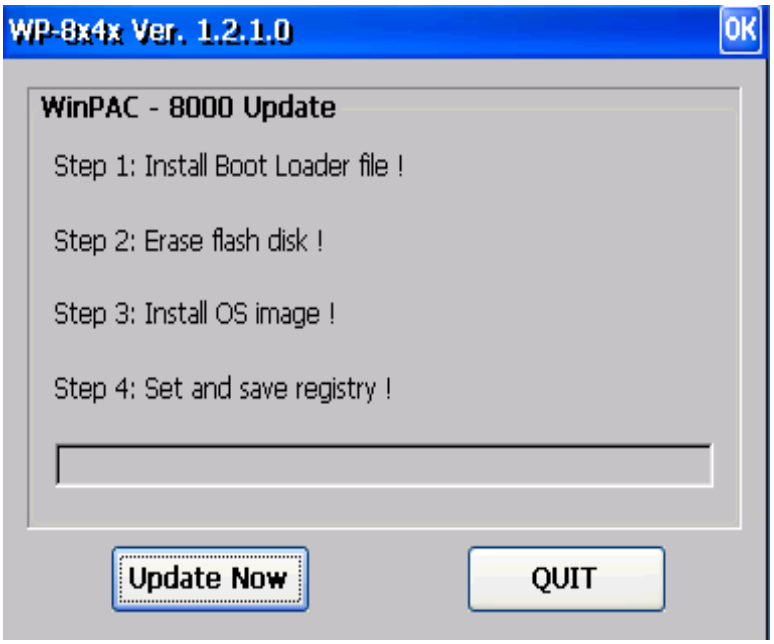
Step 3: On the main dialog, click the "Update Now" button



Step 4: On the “Warning !” dialog, click the “Yes” button



Step 5: On the main dialog, click the “Update Now” button to start installation



Please never turn the WinPAC-8000 off during OS load. Besides We recommend you turn off all other application before updating.

The installation will perform the following tasks:

i. Install Boot Loader file

Important warning

Be careful, if the boot loader broken off in this step and cannot restart in safe mode, you have to send it back to us.

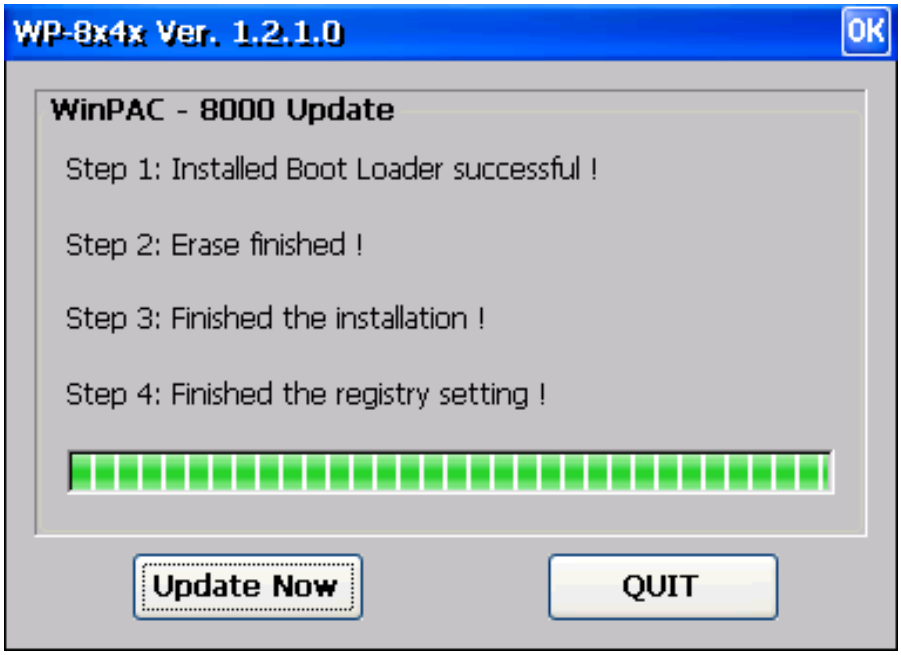
---

ii. Erase flash disk

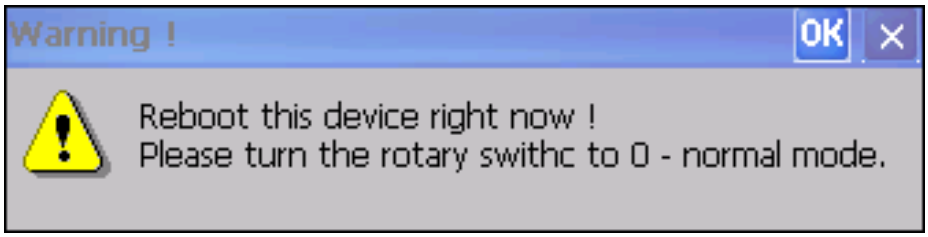
iii. Install OS image

iv. Set registry settings to default

This step will reset the registry settings to default, all of your before settings will lost.



Step 6: After completing the above-mentioned tasks, the “Warning !” dialog will appear as follow, after clicking “OK” button to finish updating OS image, be sure the WinPAC-8000 at normal mode.



E.1.2. How to update the WinPAC-8000 OS image from eshell

There are two different ways of WinPAC-8000 OS image update:

i. Updates from files (Please refer to section “E.1.1. How to update the Boot Loader and the WinPAC-8000 OS image from files”)

We recommend that you use this method for quick and easy to update the WinPAC-8000 OS image

ii. Updates from eshell tool (Please refer to this section)

By default, the OS update from Host PC to WinPAC via LAN 1. Therefore, to update the OS image, make sure LAN 1 is connected.

Step 1: Get the latest version of the WinPAC-8000 OS image file

The latest version of the WinPAC-8000 OS image file can be obtained from:

CD:\Napdos\wp-8x4x_ce50\OS_image\

http://www.icpdas.com/products/PAC/winpac/download/winpac_8000/download_os_images.htm

Step 2: Run the ESHELL software on the Host PC

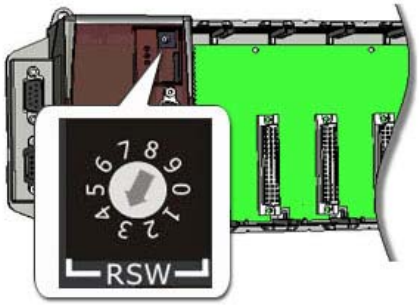
ESHELL you can be obtained at:

CD:\Napdos\wp-8x4x_ce50\PC_Tools\ESHELL

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/pc_tools/eshell/

Step 3: Reboot the WinPAC-8000 at update OS mode

Turn the rotary switch to “3”, and then reboot the WinPAC-8000.

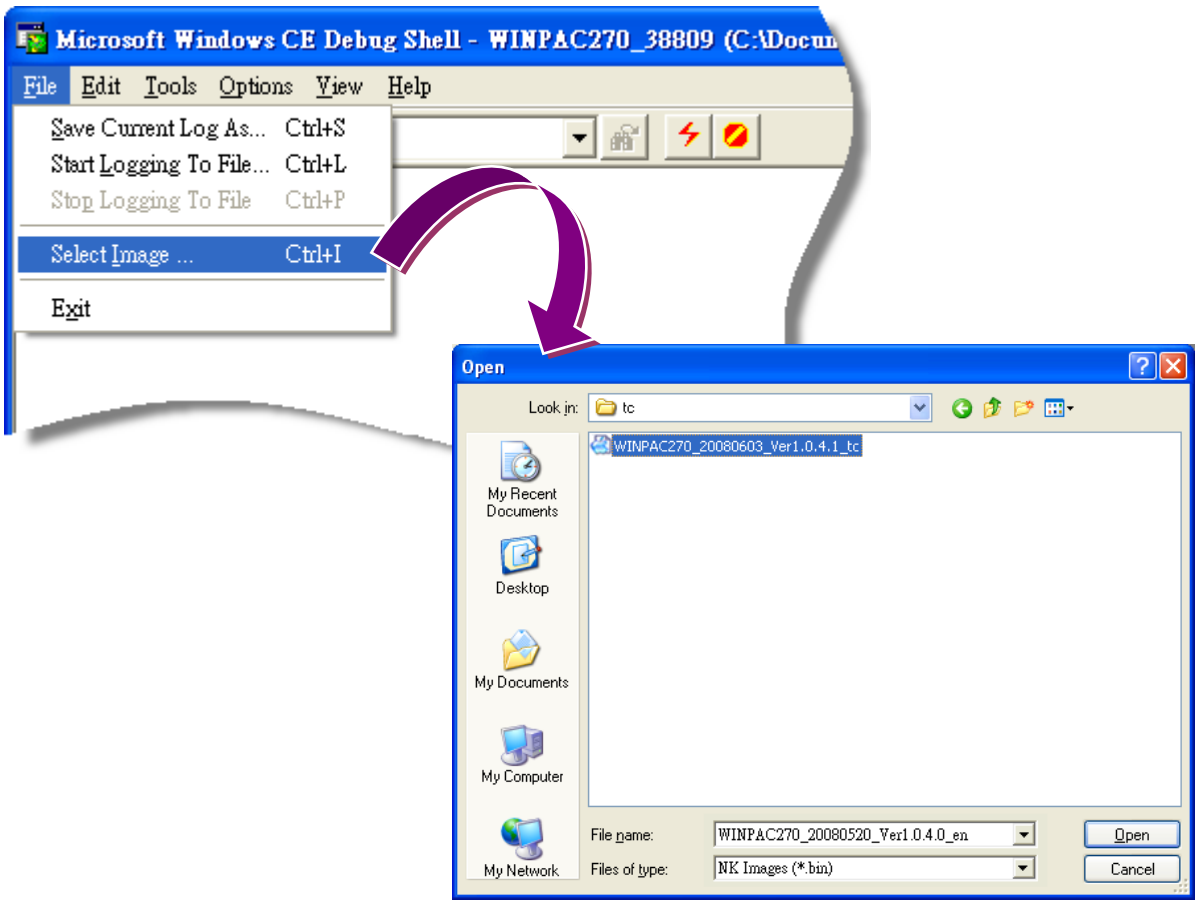


Step 4: Select the device

Select the device which you want to update the OS image to.

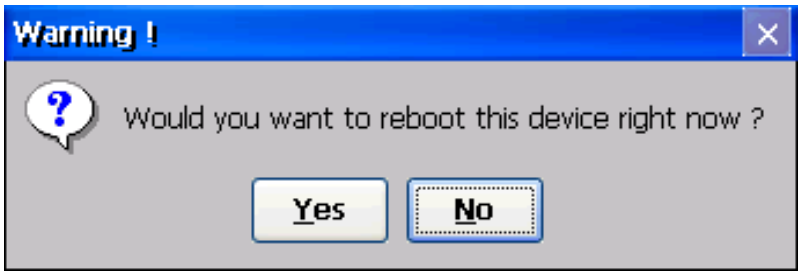
Step 5: Select the latest version of the WinPAC-8000 OS image

Select “Select Image…” from File menu to select the latest version of the WinPAC-8000 OS image.



After opening the os image file, your Host PC may display a black screen, please wait while the OS loads.

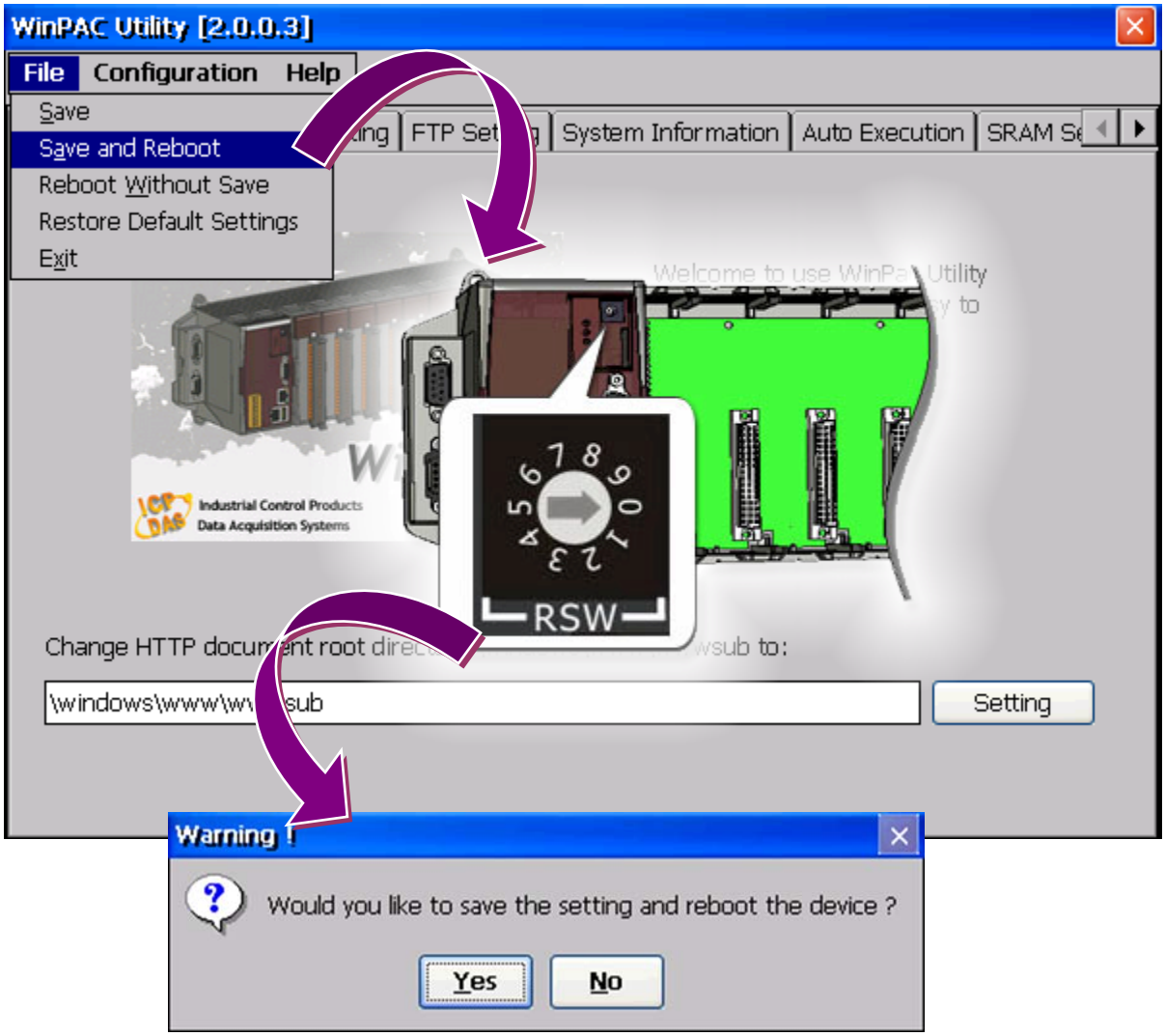
Step 6: On the “Warning !” dialog, click the “No” buttion



Step 7: Run the WinPAC Utility to save the settings and reboot the WinPAC-8000

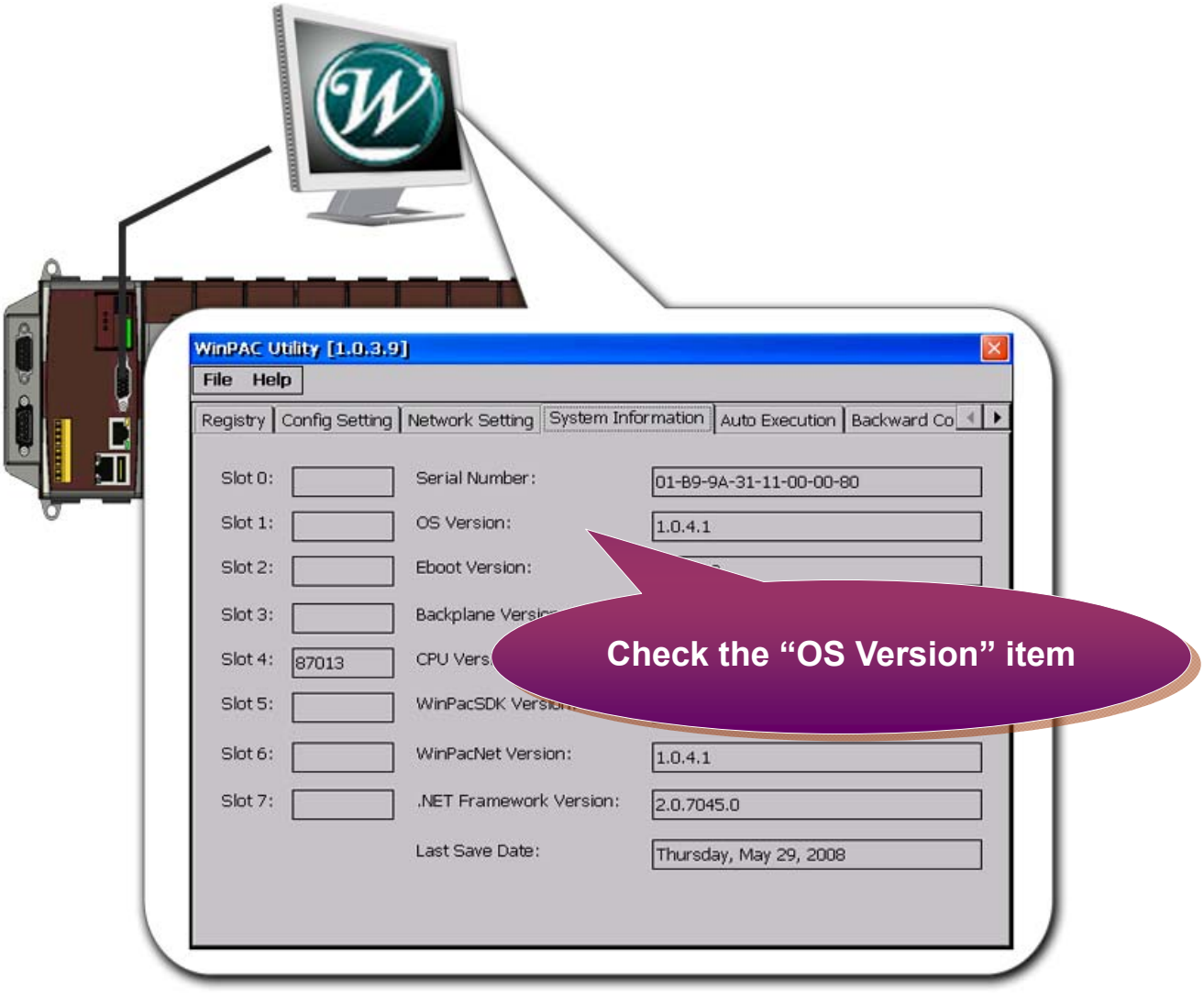
Run the WinPAC Utility, and then select the “Save and Reboot” command from the “File” menu to save the settings and reboot the WinPAC-8000.

After clicking the “Save and Reboot” command, the “Warning !” dialog will appear to confirm the reboot, make sure the WinPAC-8000 at normal mode before confirmation.



Step 8: Run the WinPAC Utility to check the OS version

Run the WinPAC Utility, and then select the “System Information” tab to check the OS version.



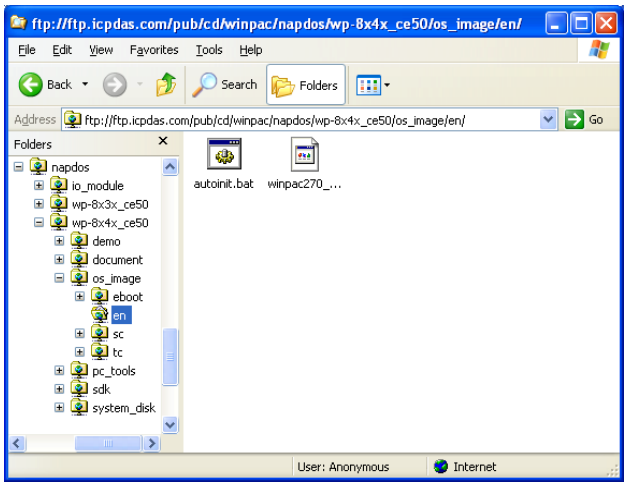
E.1.2.1. How to change the OS language

The WinPAC-8000 OS support multi language that includes English, Traditional Chinese and Simplified Chinese.

By default, the OS language change from Host PC to WinPAC via LAN 1. Therefore, to update the OS image, make sure LAN 1 is connected.

Step 1: Get the latest version of the WinPAC-8000 OS image file and the corresponding "autoinit.bat"file

Each language folder contains an OS image and a corresponding "autoinit.bat" file.



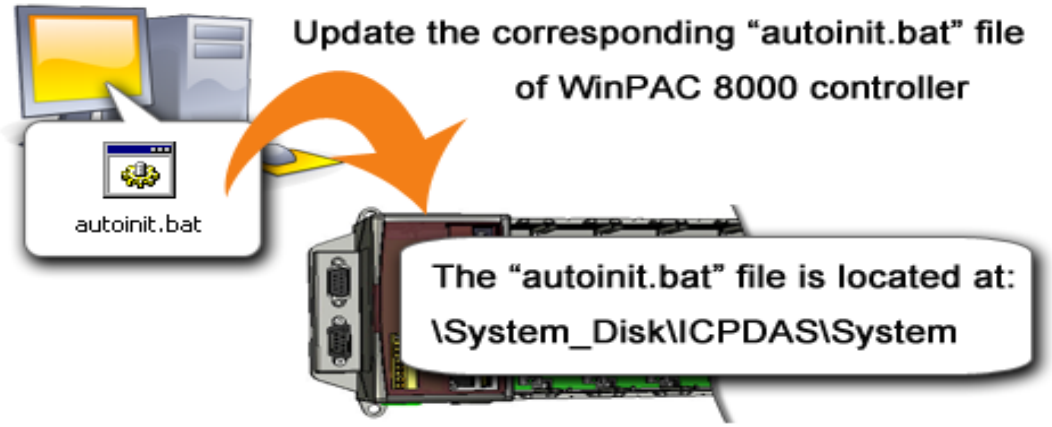
The latest version of the WinPAC-8000 OS image file and the corresponding "autoinit.bat" file can be obtained from:

CD:\Napdos\wp-8x4x_ce50\OS_image\

http://www.icpdas.com/products/PAC/winpac/download/winpac_8000/download_os_images.htm

Step 2: Update the corresponding “autoinit.bat”file

To update the corresponding “autoinit.bat” file of the WinPAC-8000 controller which is located at: \System_Disk\ICPDAS\System



Step 3: Run the ESHELL software on the Host PC

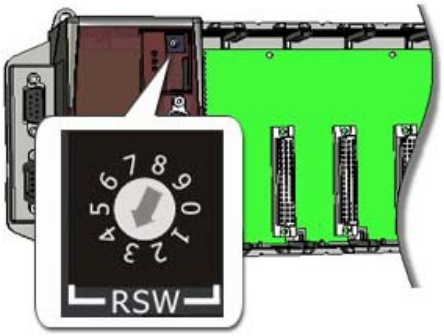
ESHELL you can be obtained at:

CD:\Napdos\wp-8x4x_ce50\PC_Tools\ESHELL

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/pc_tools/eshell/

Step 4: Reboot the WinPAC-8000 at update OS mode

Turn the rotary switch to “3”, and then reboot the WinPAC-8000.

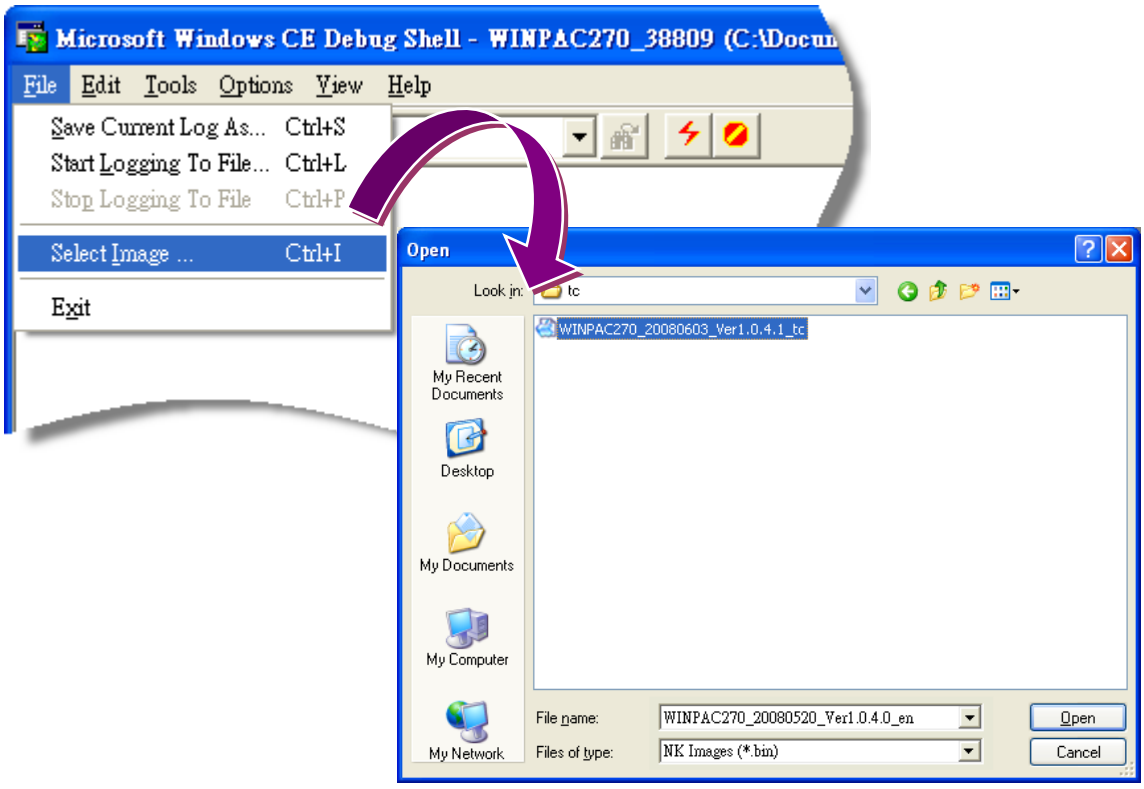


Step 5: Select the device

Select the device which you want to update the OS image to.

Step 6: Select the latest version of the WinPAC-8000 OS image (Host PC side)

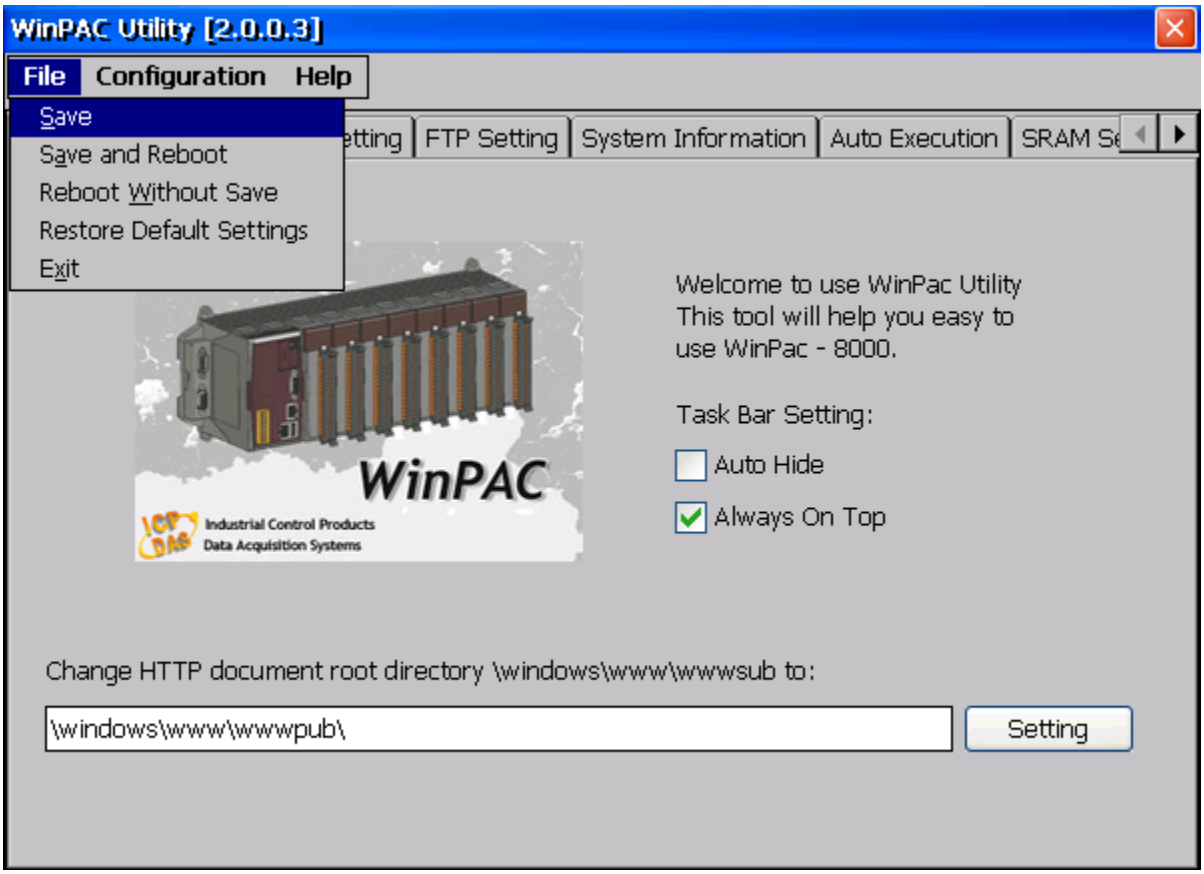
Select “Select Image…” from File menu to select the latest version of the WinPAC-8000 OS image.



After opening the os image file, your Host PC may display a black screen, please wait while the OS loads.

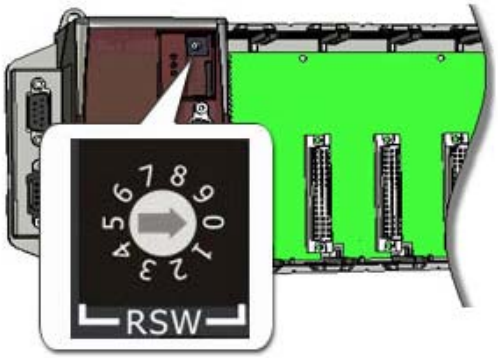
Step 7: Run the WinPAC Utility to save the configuration

Run the WinPAC Utility, and then select the “Save” command from the “File” menu to save the configuration.



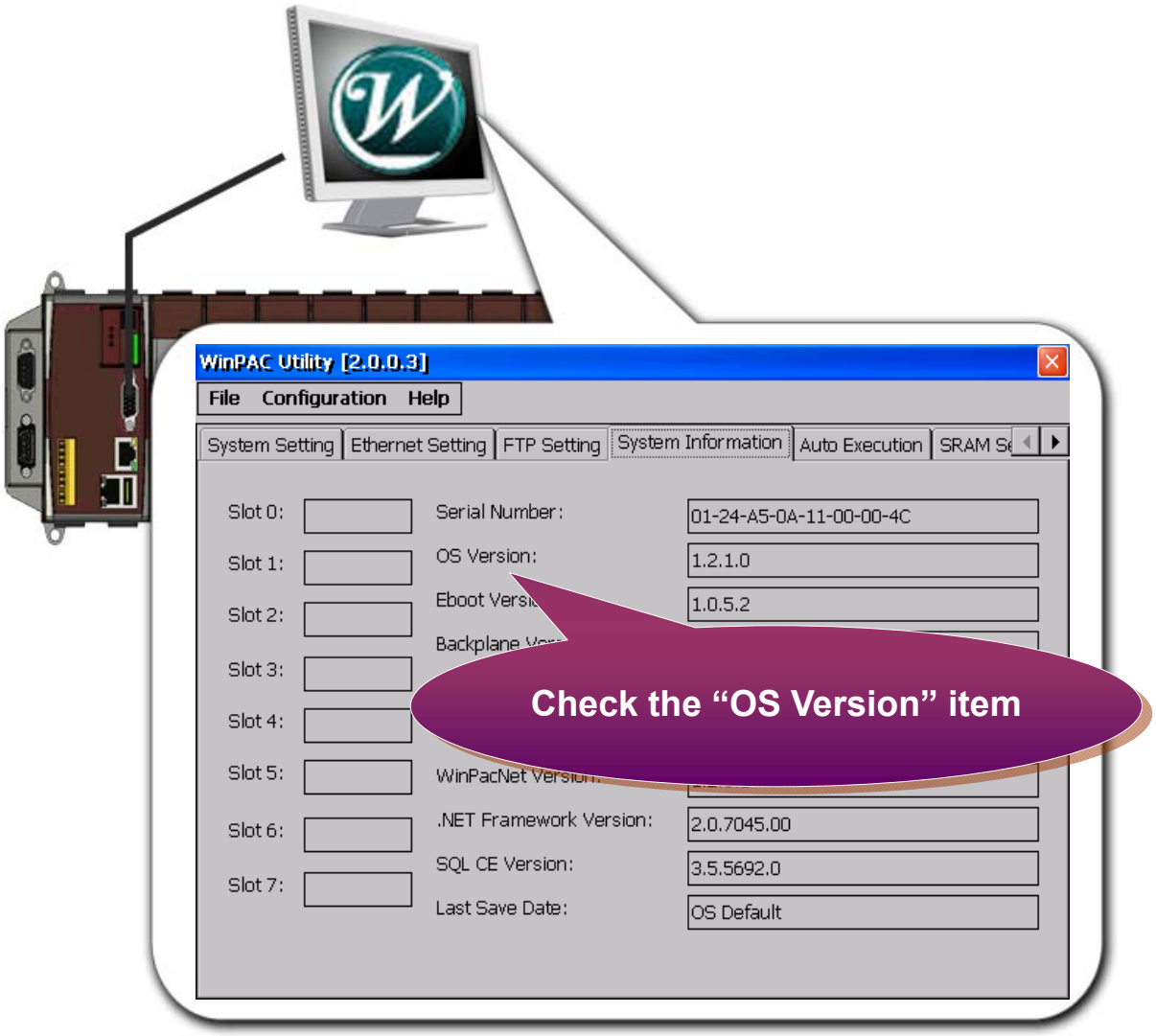
Step 8: Reboot the WinPAC-8000 in normal mode

Turn the rotary switch to “0”, and then reboot the WinPAC-8000.



Step 9: Run the WinPAC Utility to check the current OS version

Run the WinPAC Utility, and then select the “System Information” tab to check the current OS version.

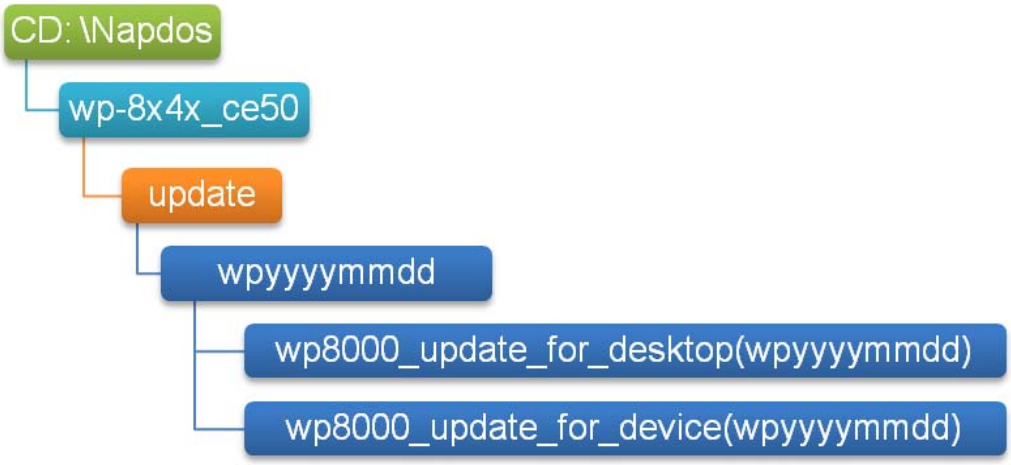


E.1.3. How to install the WinPAC-8000 SDK update package

The update packages are located in:

CD:\Napdos\wp-8x4x_ce50\update

By eVC and donet development tools, the update package are used only in eVC.



Step 1: On the host PC side, run the "PAC270_SDK_YYYYMMDD.msi" file which is located at:

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/Update/wpyyyymmdd/wp8000_update_for_desktop(wpyyyymmdd)

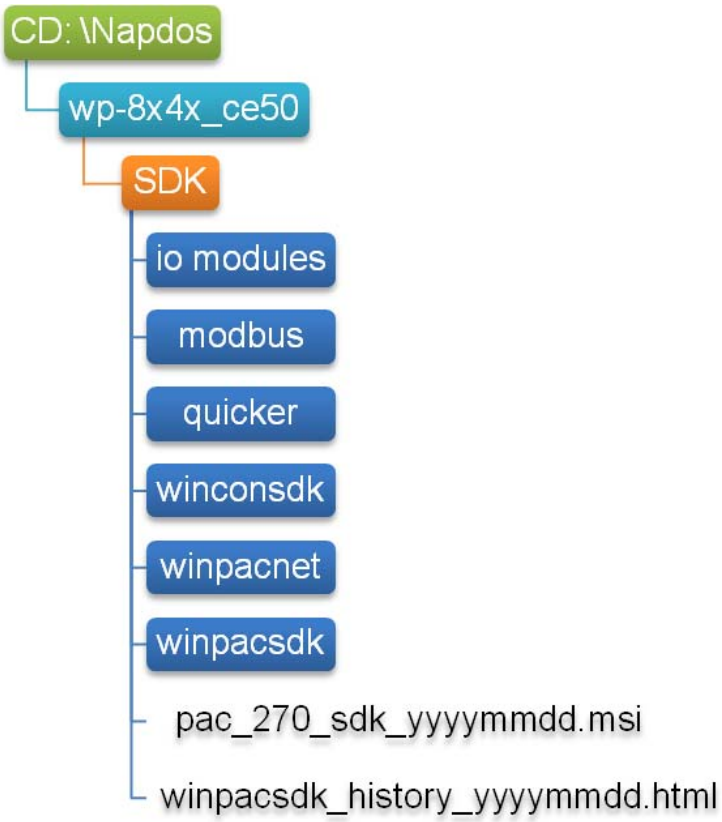
Step 2: On the WinPAC-8000 controller side, via Ethernet connection, download and run the "wp8000_update(wpmmddyyyy).CAB" file Which is located at:

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/Update/wpyyyymmdd/wp8000_update_for_device(wpyyyymmdd)

E.1.4. How to manually update the WinPAC-8000 SDK

The WinPAC-8000 SDK are located in:

CD:\Napdos\wp-8x4x_ce50\SDK



By eVC and donet development tools, the WinPAC SDK installation are divided into the following two parts:

i. WinPAC SDK updates for eMbedded Visual C++

ii. WinPAC SDK updates for dotnet

WinPAC SDK updates for eMbedded Visual C++

To determine the SDK version that is compatibly running on the WinPAC-8000, you can read the “Release Note” which is located under each SDK folder, these files provides important updated information for what we fixed and added.

Step 1: Get the latest version of the eMbedded Visual C++ components

The latest version of the eMbedded Visual C++ components can be obtained from:

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/sdk/winpacsdk/

Step 2: Copy the latest version of header files and libraries to Host PC

The header files are located at:

C:\Program Files\Windows CE Tools\wce500\PAC270\Include

The libraries are located at:

C:\Program Files\Windows CE Tools\wce500\PAC270\Lib

Step 3: Copy the latest version of DLL files to WinPAC-8000

The DLL files are located at:

\System_Disk\ICPDAS\System

WinPAC SDK updates for dotnet

To determine the SDK version that is compatibly running on the WinPAC-8000, you can read the “Release Note” which is located under each SDK folder, these files provides important updated information for what we fixed and added.

Step 1: Get the latest version of the dotnet components

The latest version of the eMbedded Visual C++ components can be obtained from:

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/sdk/winpacnet/

Step 2: Copy the latest version of DLL to Host PC and WinPAC-8000

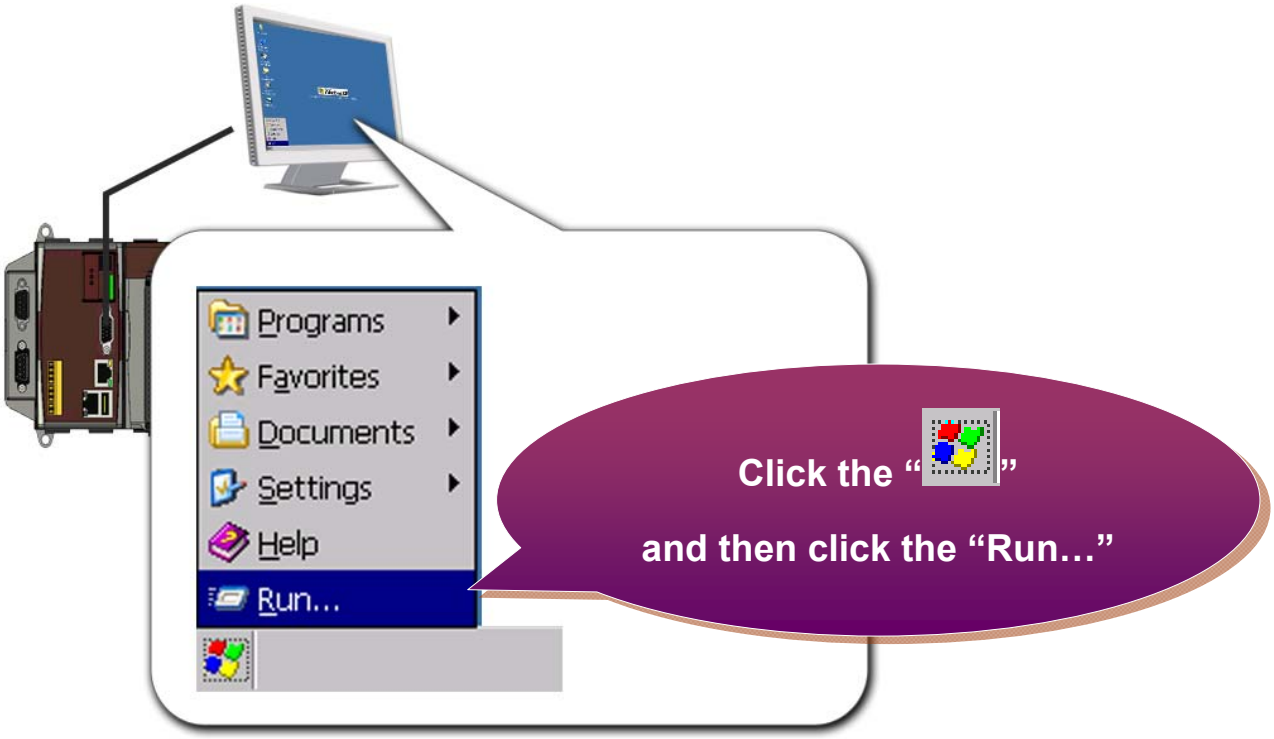
The DLL files on Host PC are located at anywhere only the solution can reference it.

The DLL files on WinPAC-8000 are located at the same directory as the .exe file.

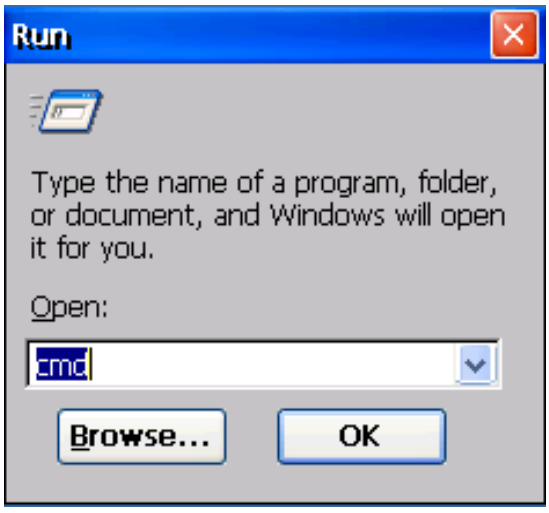
E.2. How to establish a new telnet and FTP account

To establish a new telnet and FTP account, please perform the following steps:

Step 1: On the WinPAC-8000 controller side, select the "Start" menu, and then click the "Run…" command

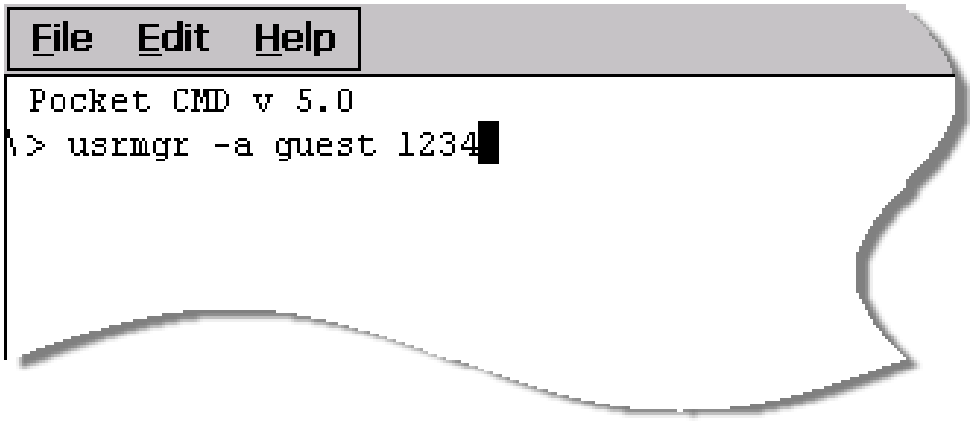


Step 2: On the "Run" dialog, type "cmd" in the field and then click the "OK" button

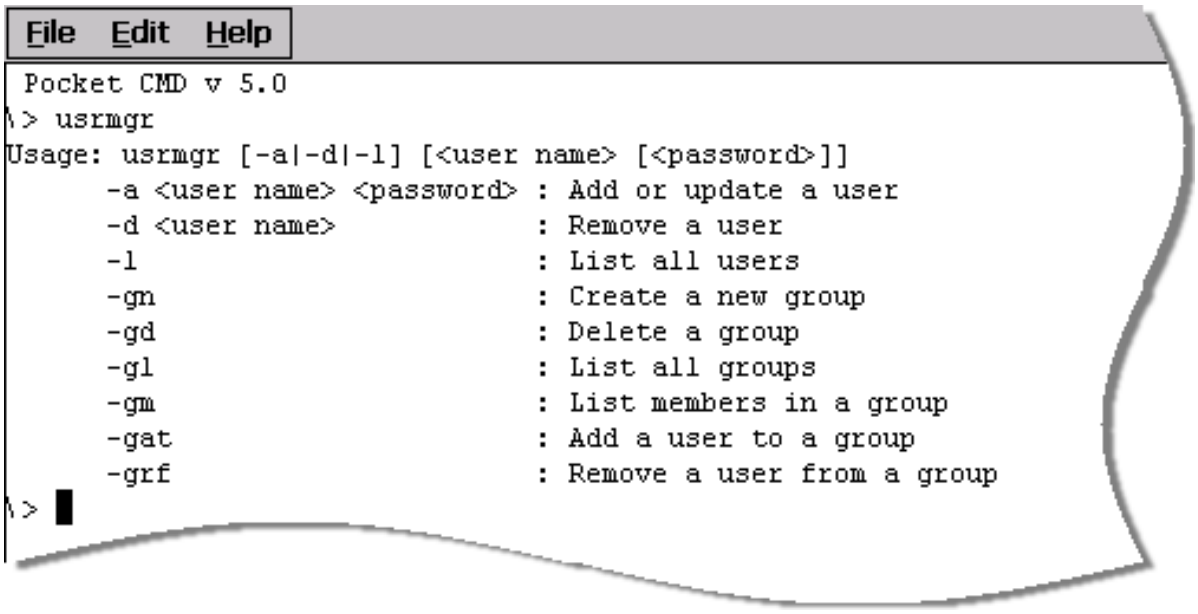


Step 3: Establish a new account

[Syntax] usrmgr –a <username> <password>



For more information about managing accounts, you just type “usrmgr” command to get a complete list of supported commands..



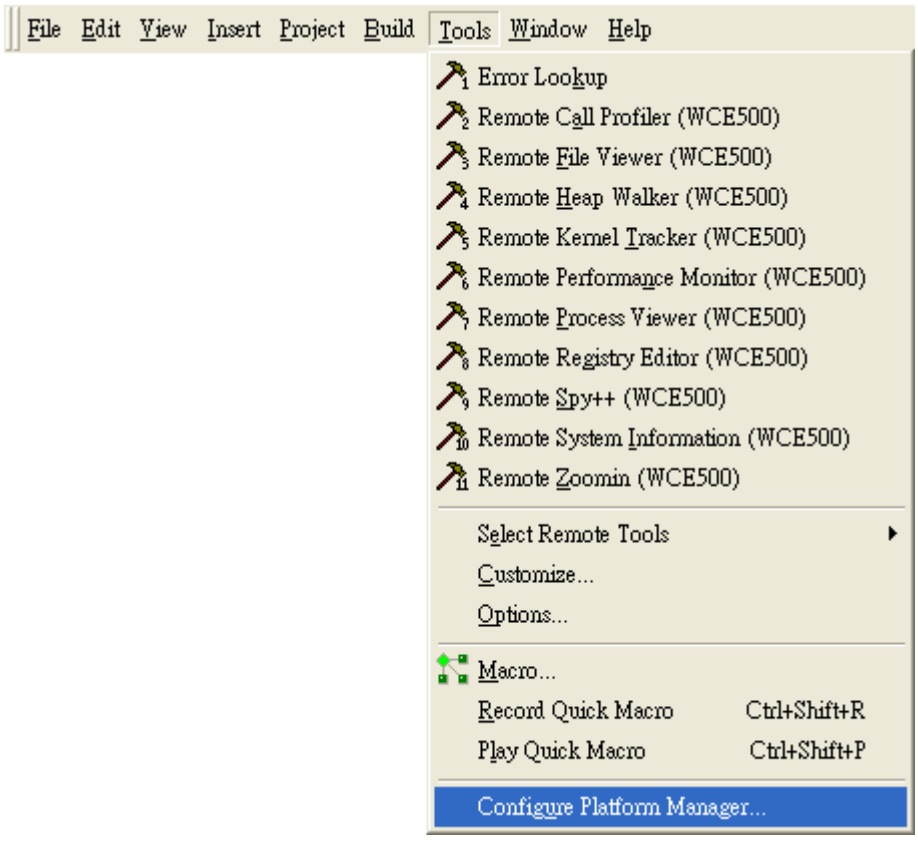
The setting may not save. That is, after rebooted, the setting will disappear and you should set the account again.

E.3. How to online debug WinPAC

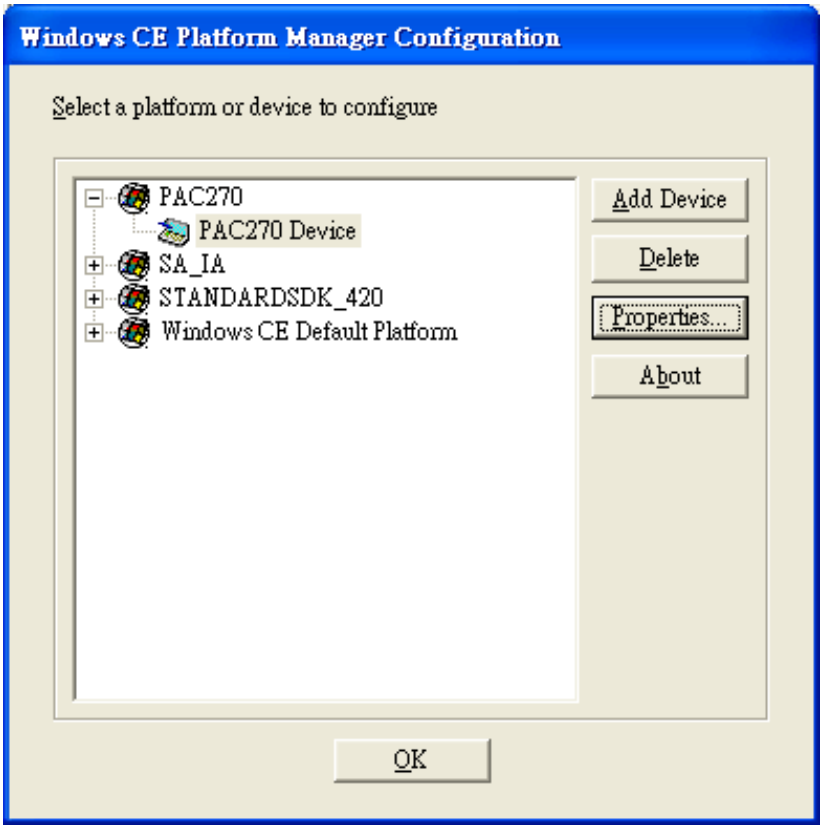
Debugging is a process that you use to find and resolve errors, or bugs, in a program.

E.3.1. Debug WinPAC programs in EVC++

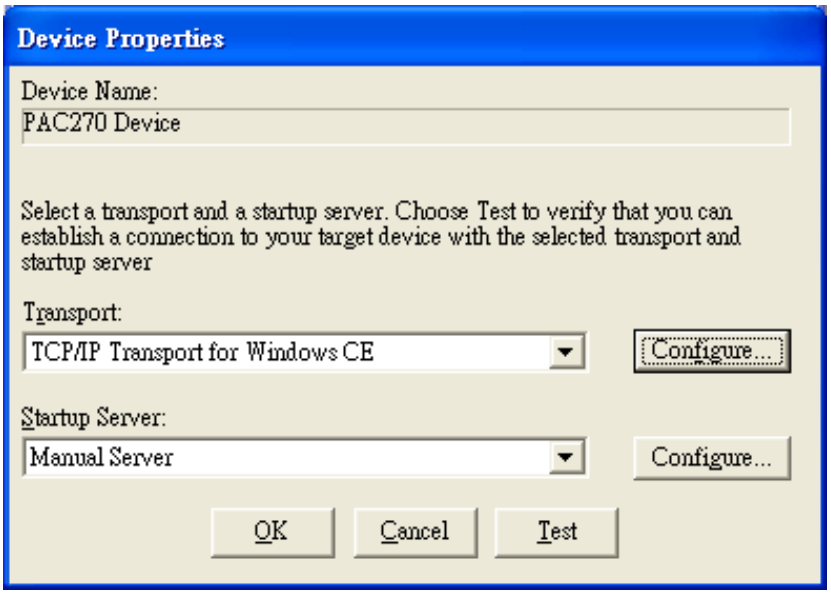
Step 1: On the “Tools” menu, click “Configure Platform Manager…” command



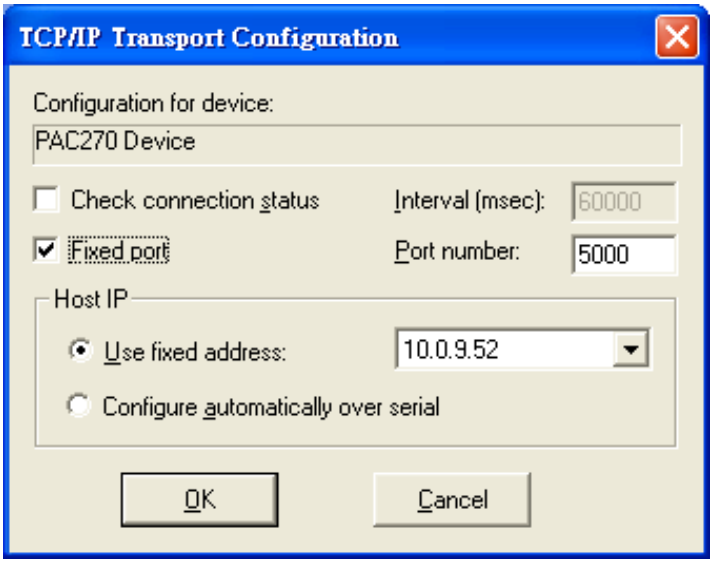
Step 2: On the "Windows CE Platform Manager Configuration" dialog, click the "Properties…" button



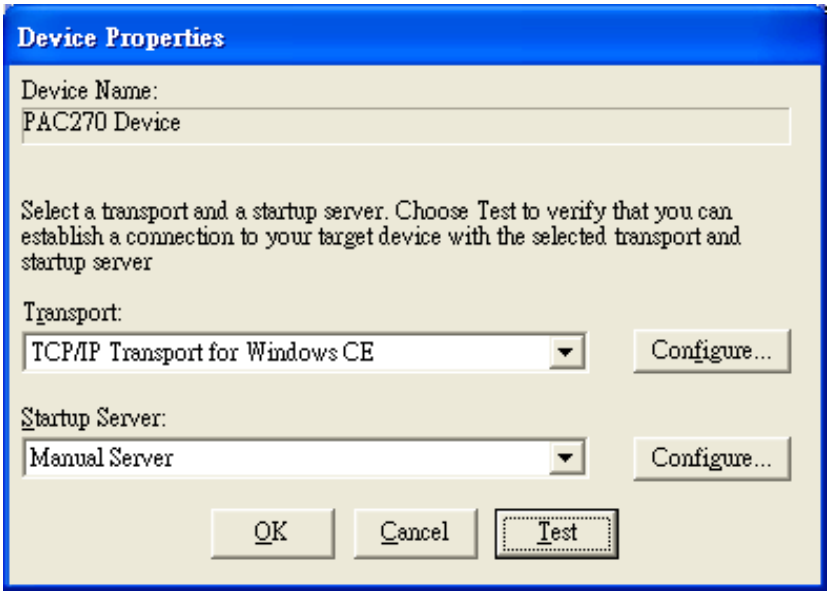
Step 3: On the "Device Properties" dialog, click the "Configure…" button



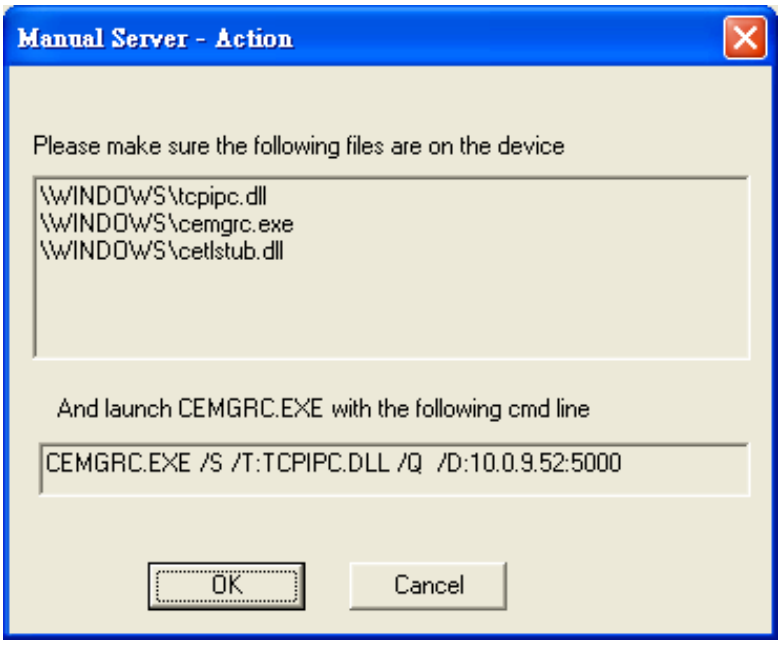
Step 4: On the "TCP/IP Transport Configuration" dialog, select the "Fixed port" check box, and then click the "OK" button



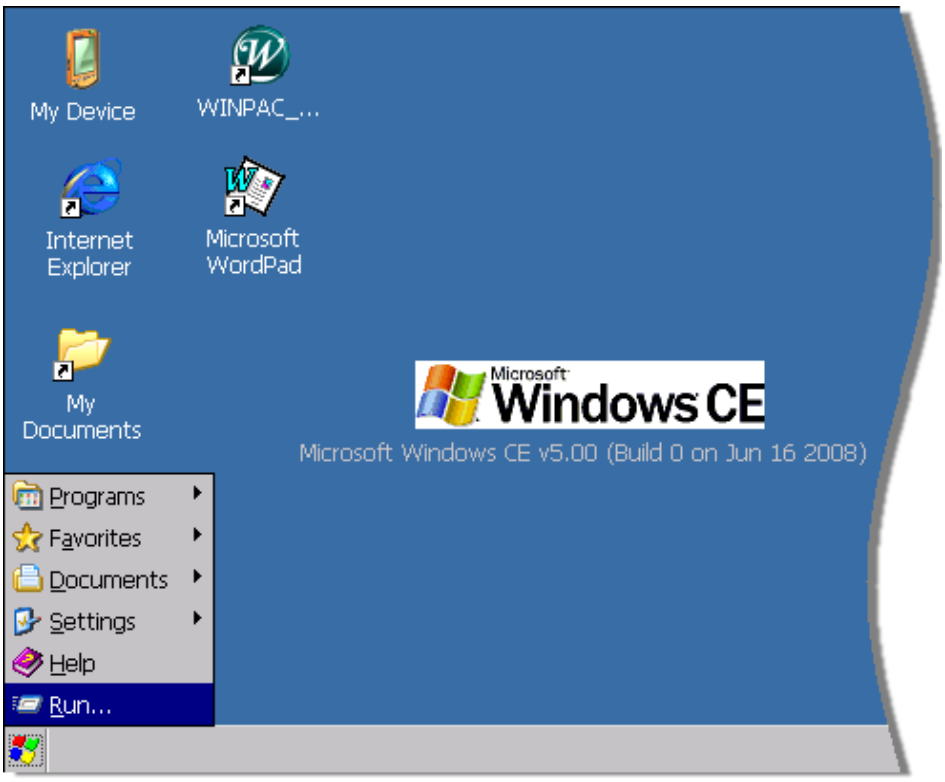
Step 5: On the "Windows CE Platform Manager Configuration" dialog, click the "Test" button



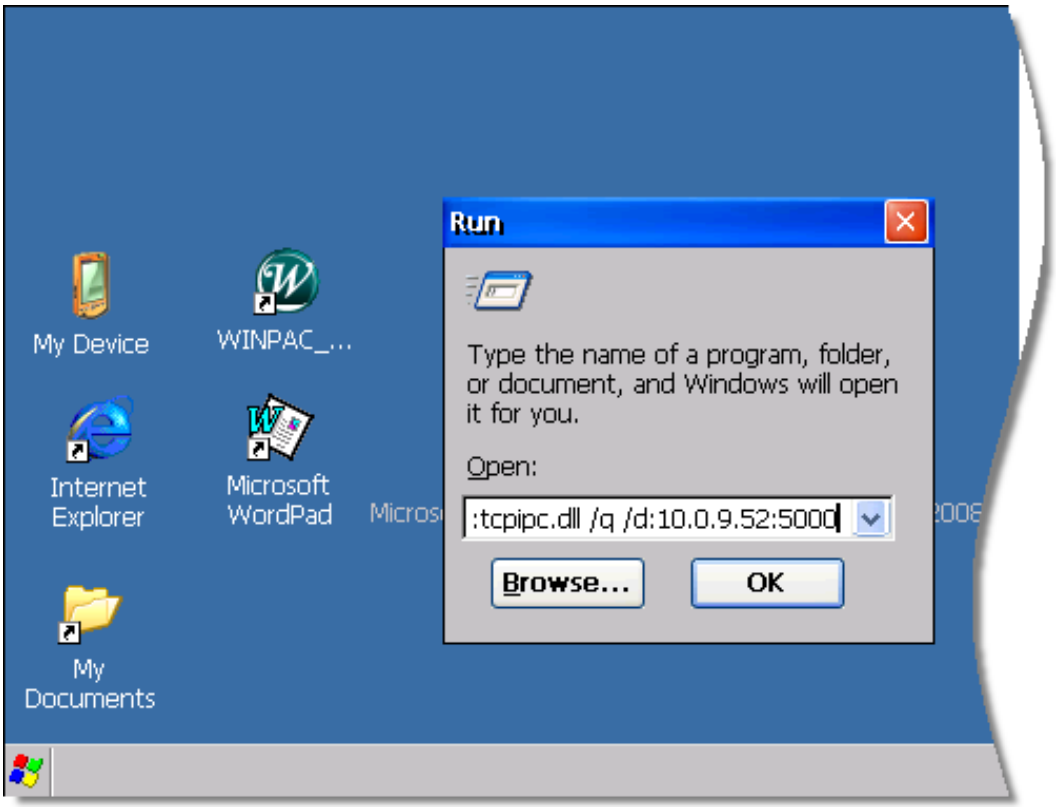
Step 6: The “Manual Server - Action” dialog will appear displaying a command line, before click the “OK” button to close dialog, turn to the WinPAC-8000 controller side to do the next two-steps



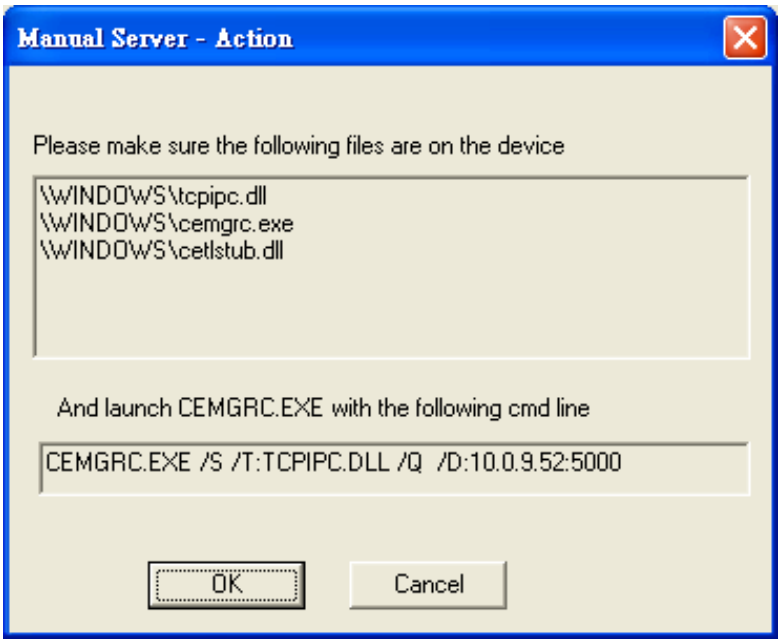
Step 7: On the WinPAC-8000 controller side, select the “ Start” menu, and then click the “Run…” command



Step 8: On the "Run" dialog, type the command which displays in step 5 and then click the "OK" button

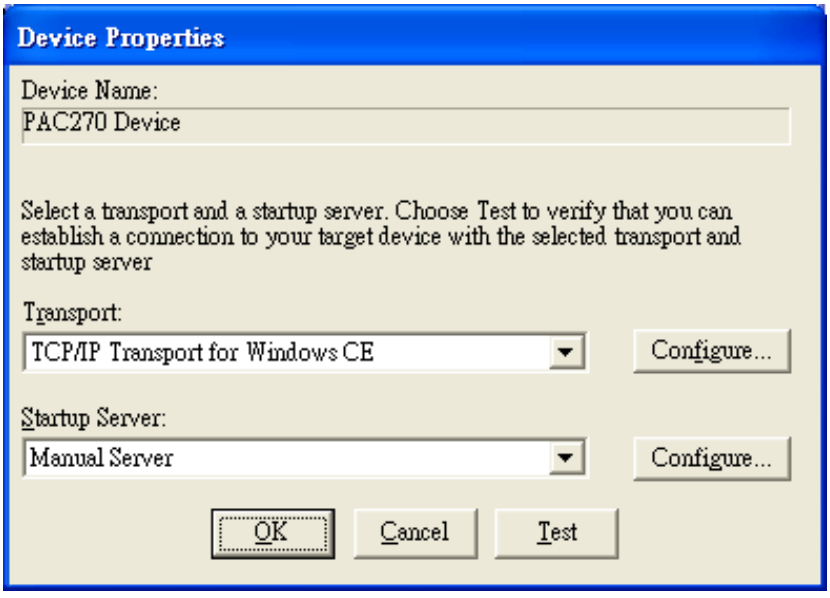


Step 9: Return to the Host PC side, on the "Manual Server – Action" dialog, click the "OK" button

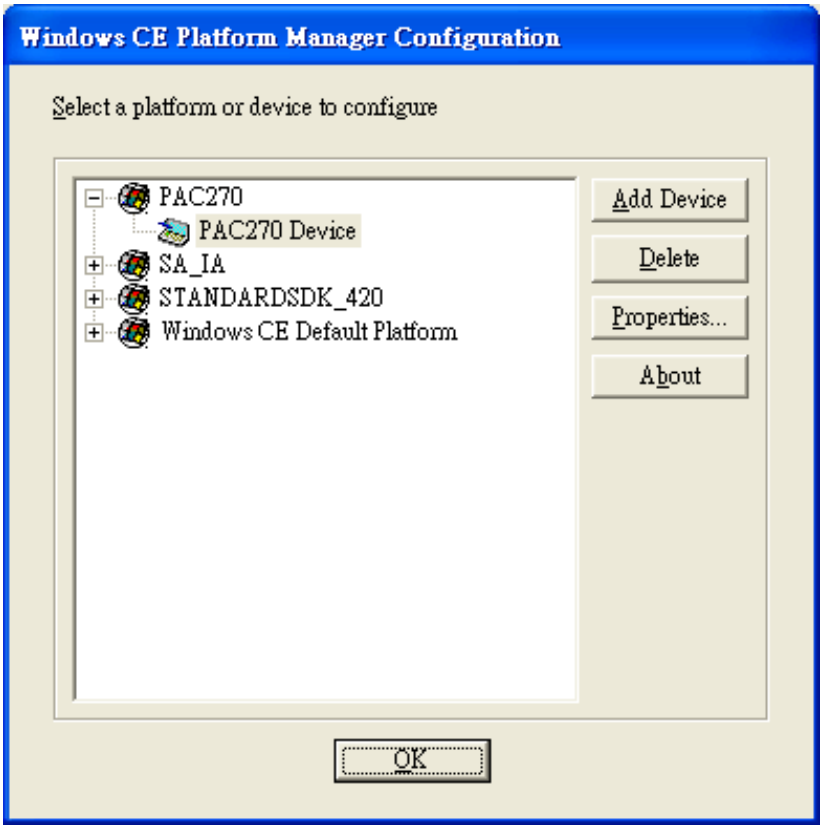


Step 10: On the “Testing Device Connection” dialog, click the “OK” button

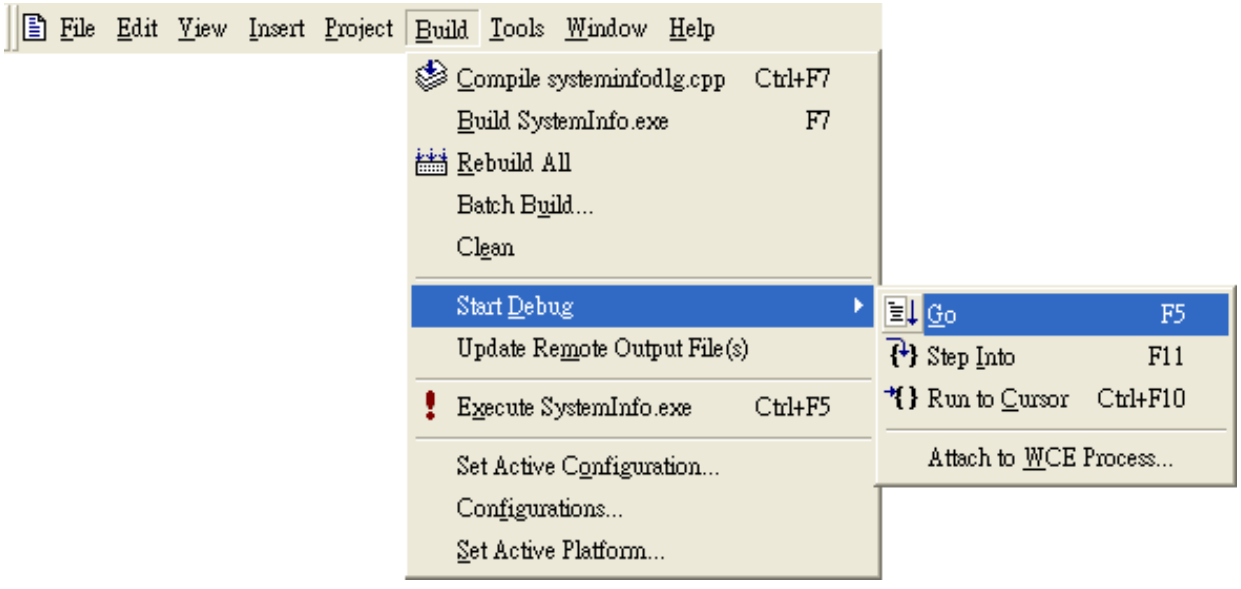
Step 11: On the “Device Properties” dialog, click the “OK” button



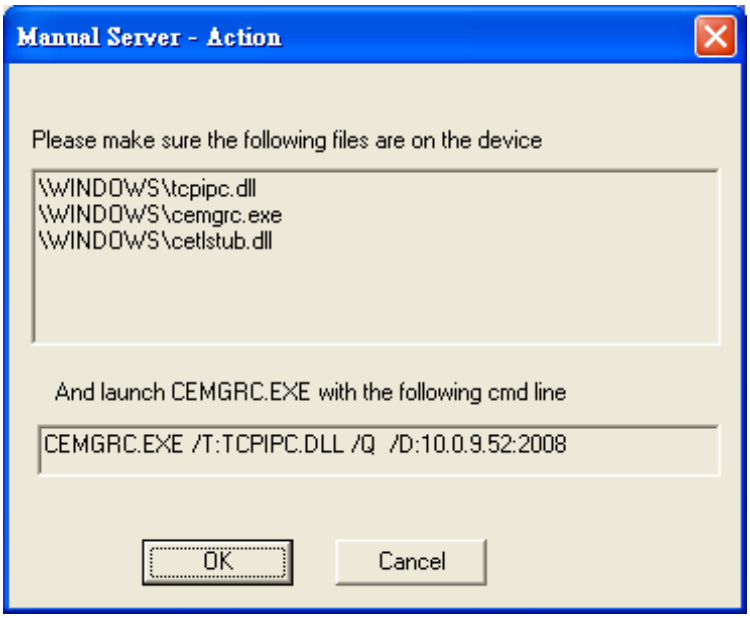
Step 12: On the “Windows CE platform or device to configure” dialog, click the “OK” button



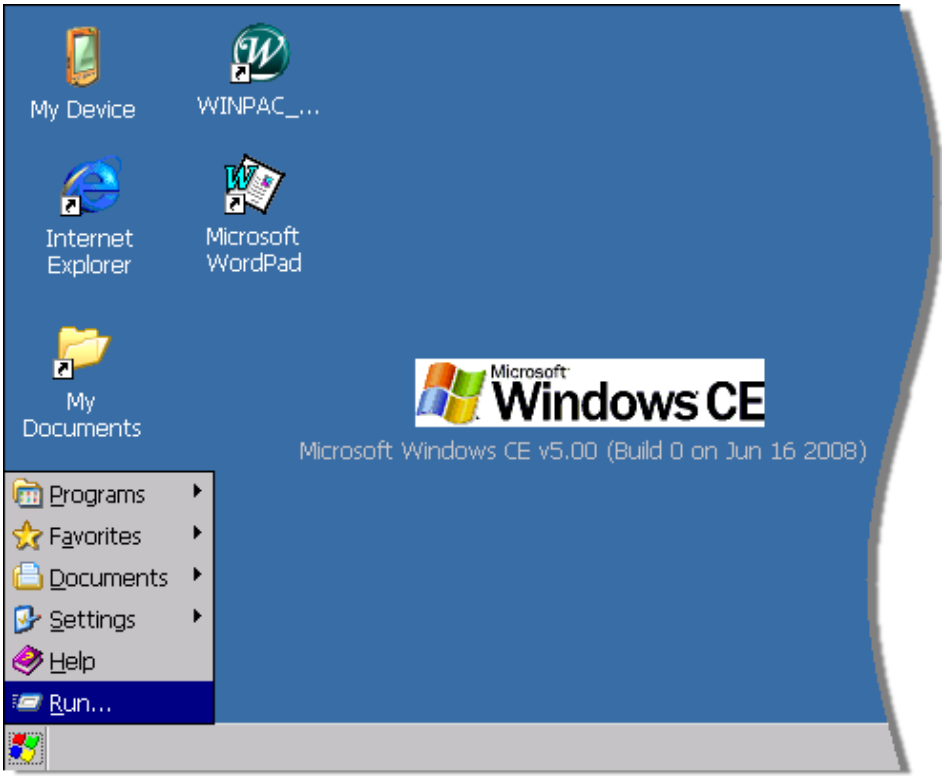
Step 13: On the “Build” menu, select the “Start Debug” command and then click the “Go” command



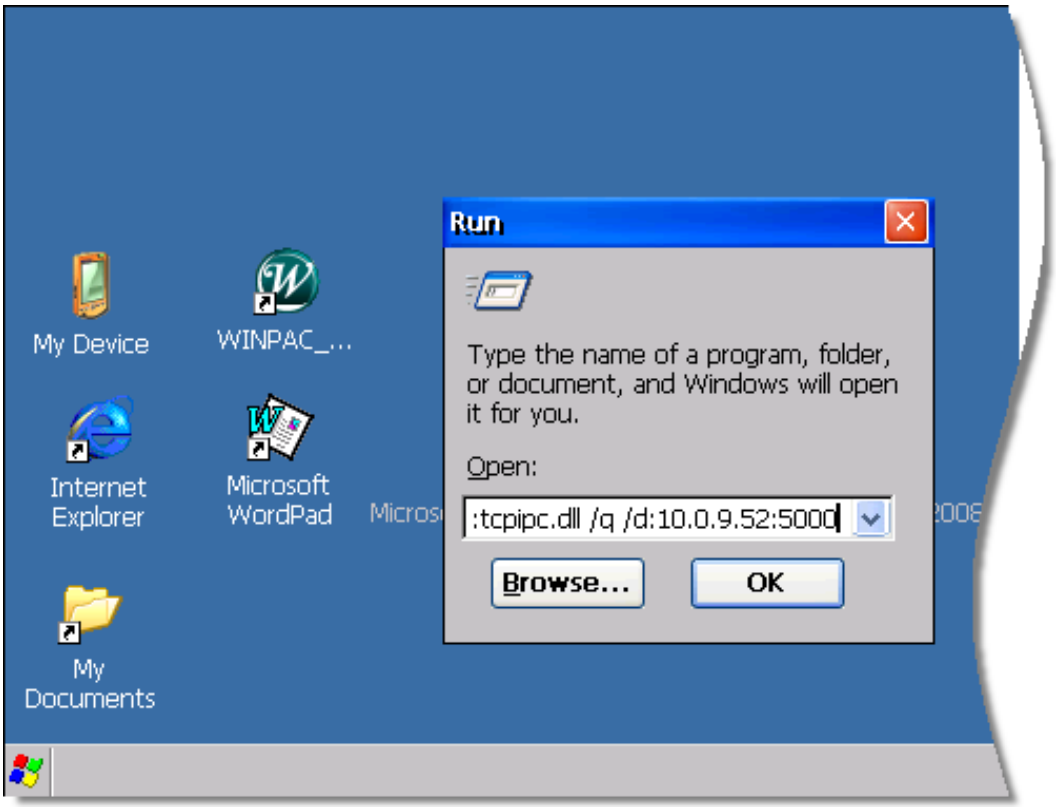
Step 14: The “Manual Server - Action” dialog will appear displaying a command line, before click the “OK” button to close dialog, turn to the WinPAC-8000 controller side to do the next two-steps



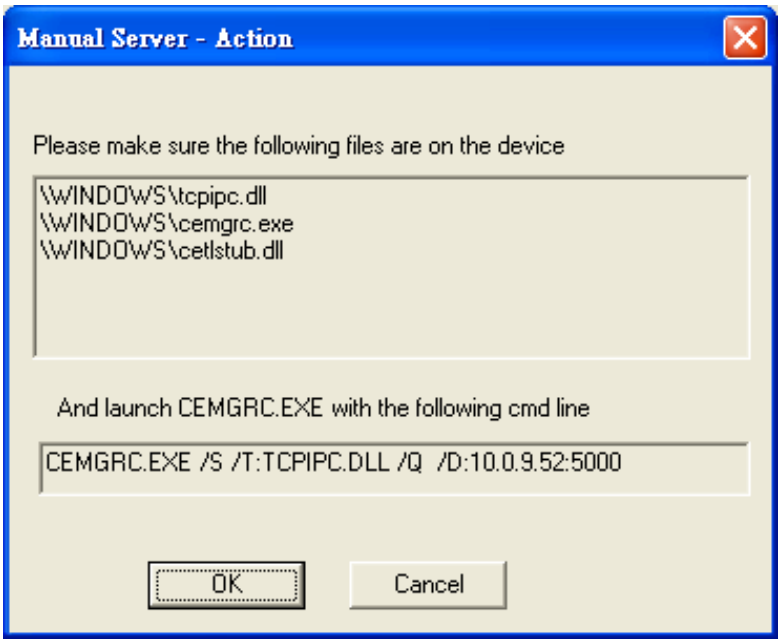
Step 15: On the WinPAC-8000 controller side, select the “ Start” menu, and then click the “Run…” command



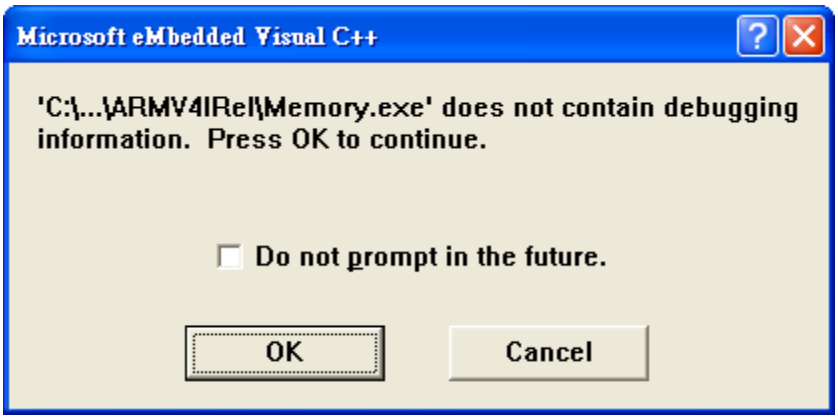
Step 16: On the “Run” dialog, type the command which displays in step 5 and then click the “OK” button



Step 17: Return to the Host PC side, on the “Manual Server – Action” dialog, click the “OK” button

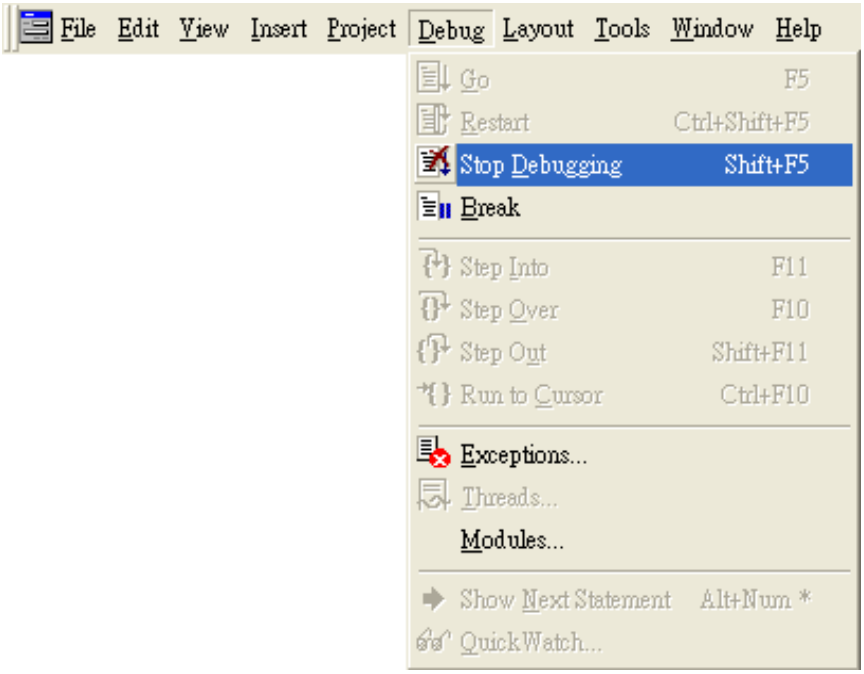


Step 18: On the “Manual Server - Action” dialog, click the “OK” button



Step 19: Connection established. Then you can debug on line.

If you want to quit the debugger and return to editing, you can click the “Stop Debugging “ button from “Debug” menu



E.3.2. Debug WinPAC programs in Visual Studio 2005/2008

Step 1: Copy the following files to WinPAC-8000 :\System_Disk\ICPDAS\System folder

- Clientshutdown.exe
- ConmanClient2.exe
- CMaccept.exe
- eDbgTL.dll
- TcpConnectionA.dll

Before copying these files to WinPAC-8000, make sure these files and the "ConMan2.dll" file are the same version.

The "ConMan2.dll" file is located at:

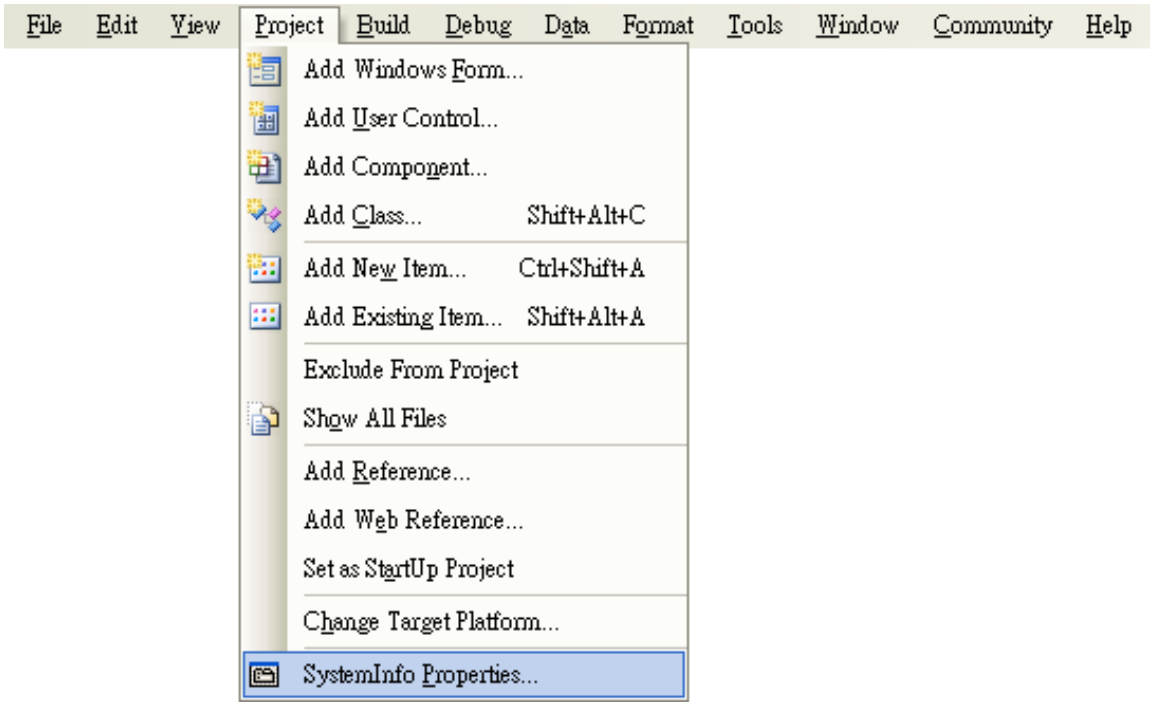
C:\Program Files\Common Files\Microsoft Shared\CoreCon\1.0\Bin

By default, these files are located at:

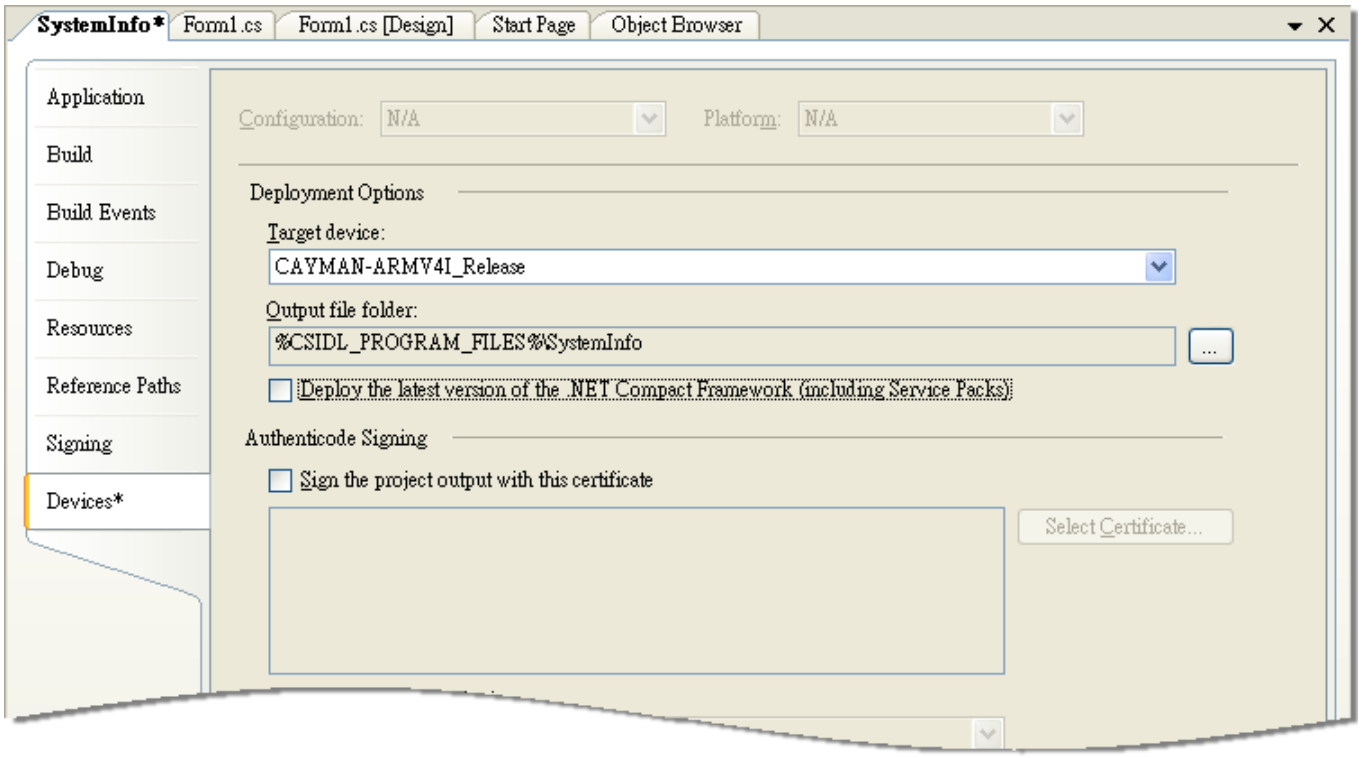
C:\Program Files\Common Files\Microsoft Shared\CoreCon\1.0\Target\wce400\armv4i

---

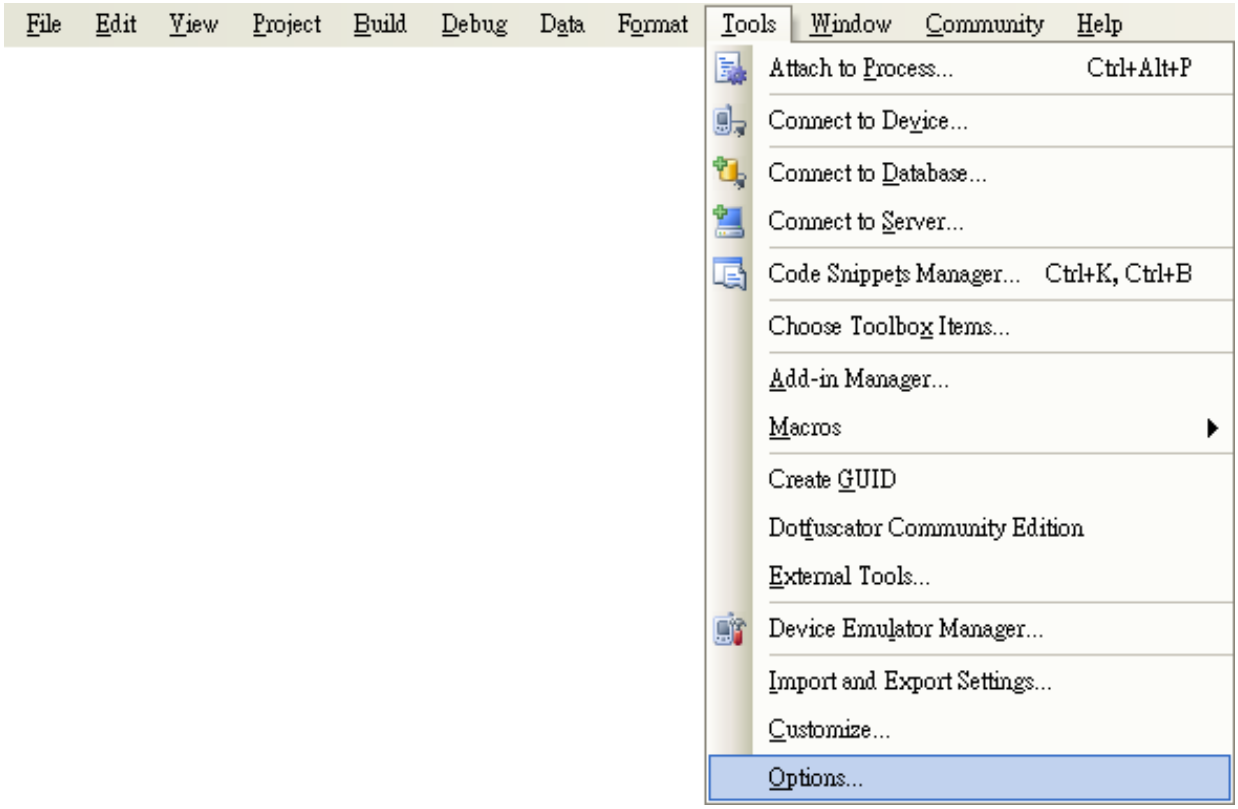
Step 2: On the "Project" menu, click "[Project Name] Properties…" command



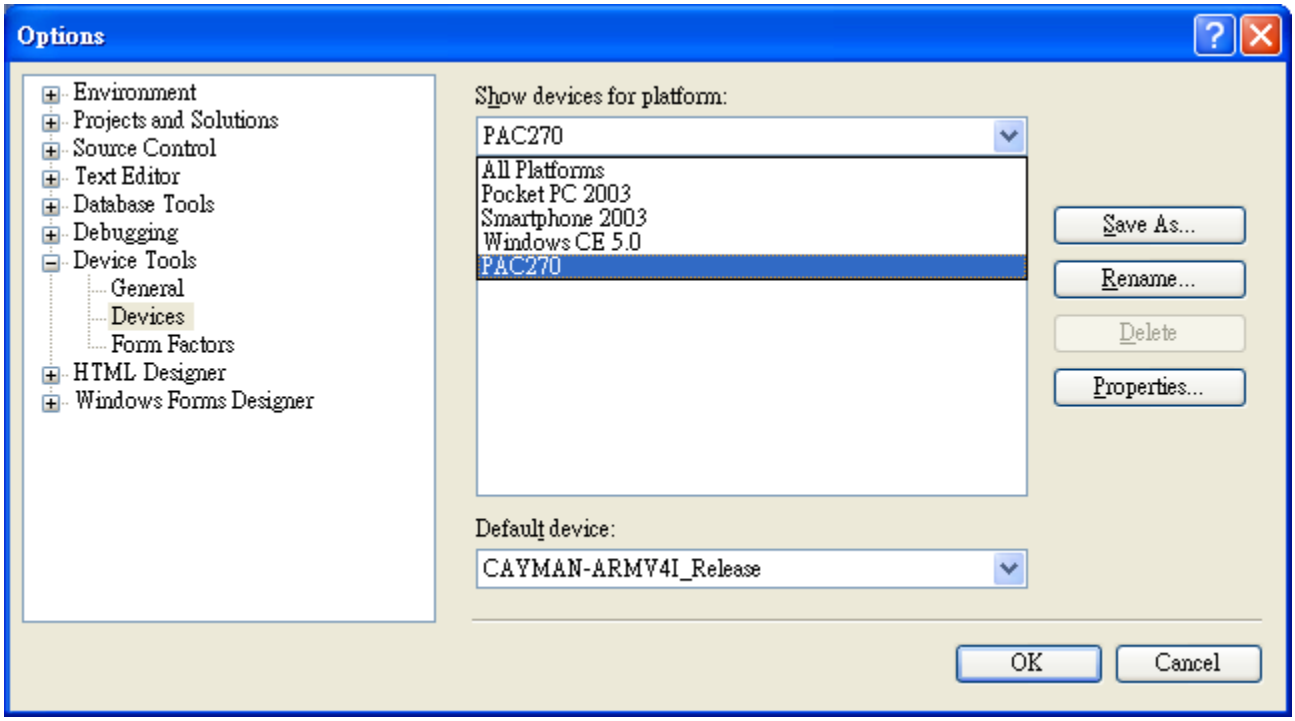
Step 3: On the "SystemInfo*" tab, unselect "Deploy the latest version of the .NET compact Framework (includeing Service Packs)" check box



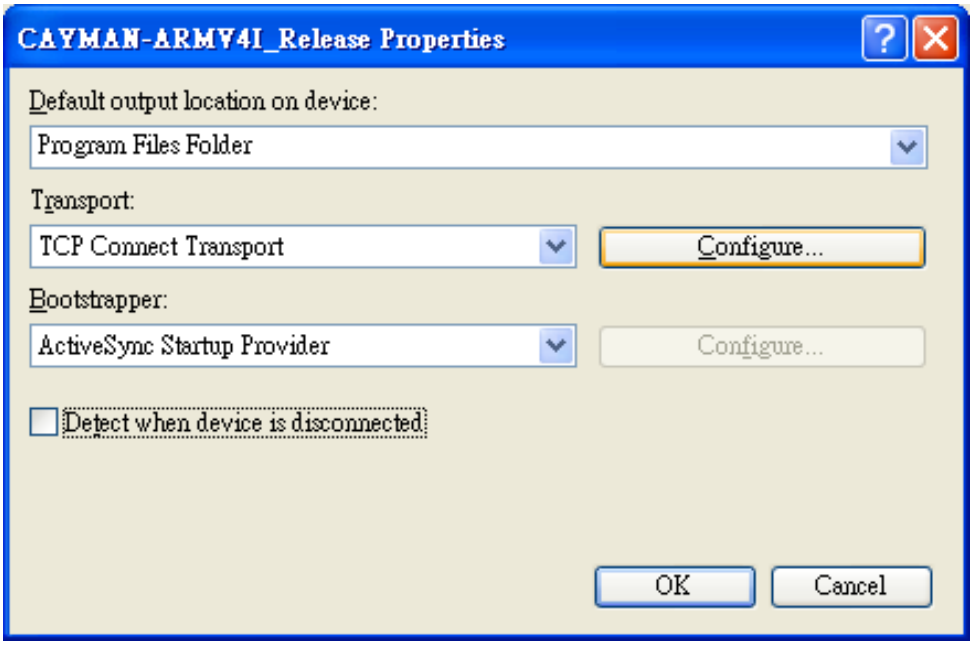
Step 4: On the "Tools" menu, click "Options…" command



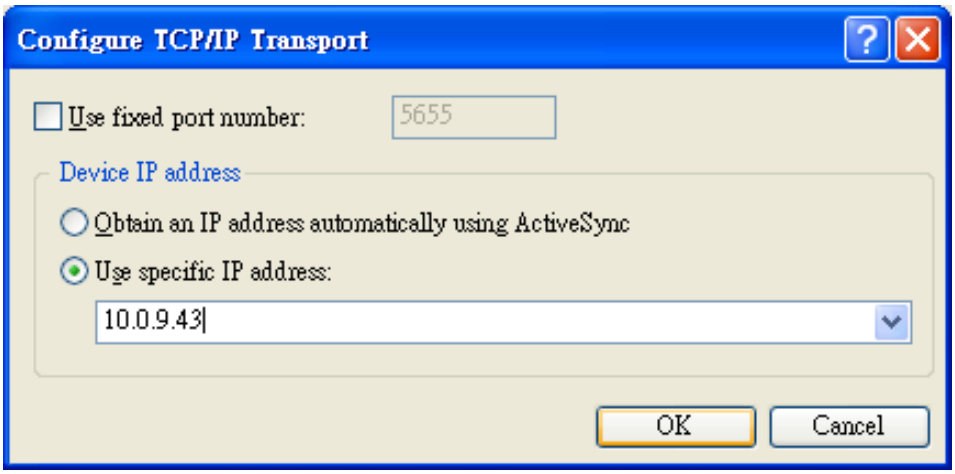
Step 5: On the “Options” dialog, select “PAC 270” from the “Show devices platform” list, and then click the “Properties…” button



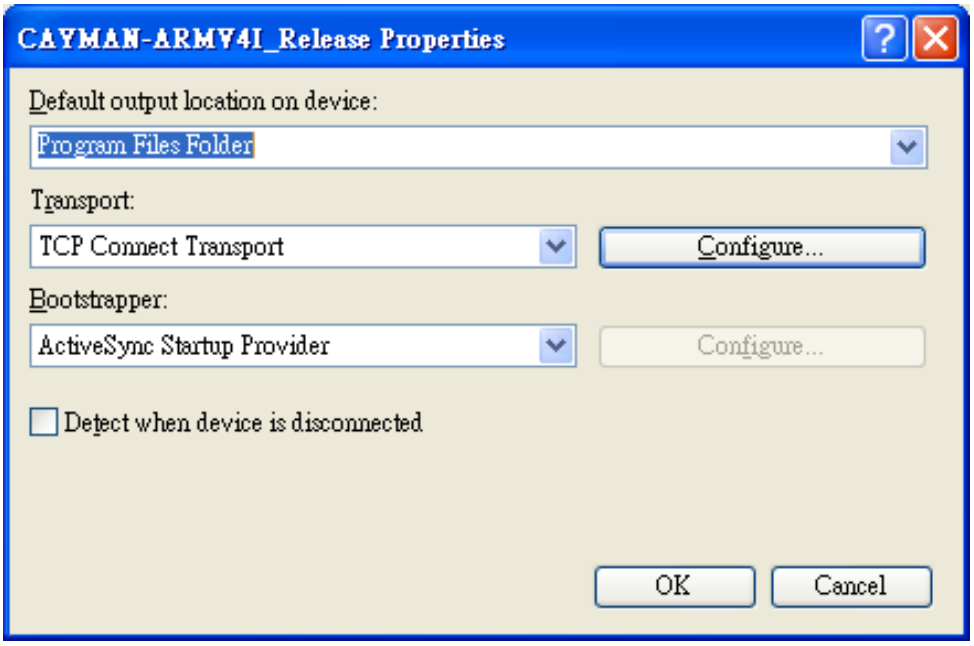
Step 6: On the “CAYMAN-ARMV4I_Release Properties” dialog, click the “Configure…” button



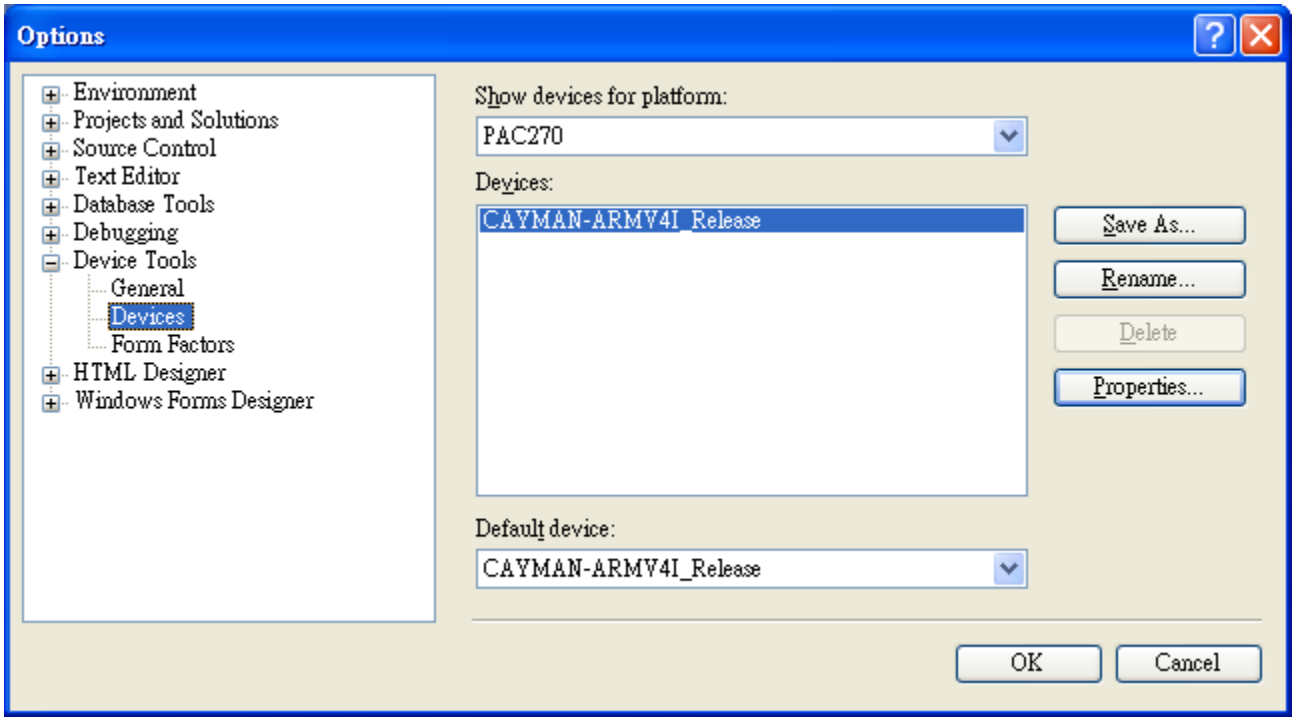
Step 7: On the “Configure TCP/IP Transport” dialog, select the “Use specific IP address” option and type the IP address of WinPAC-8000, and then click the “OK” button



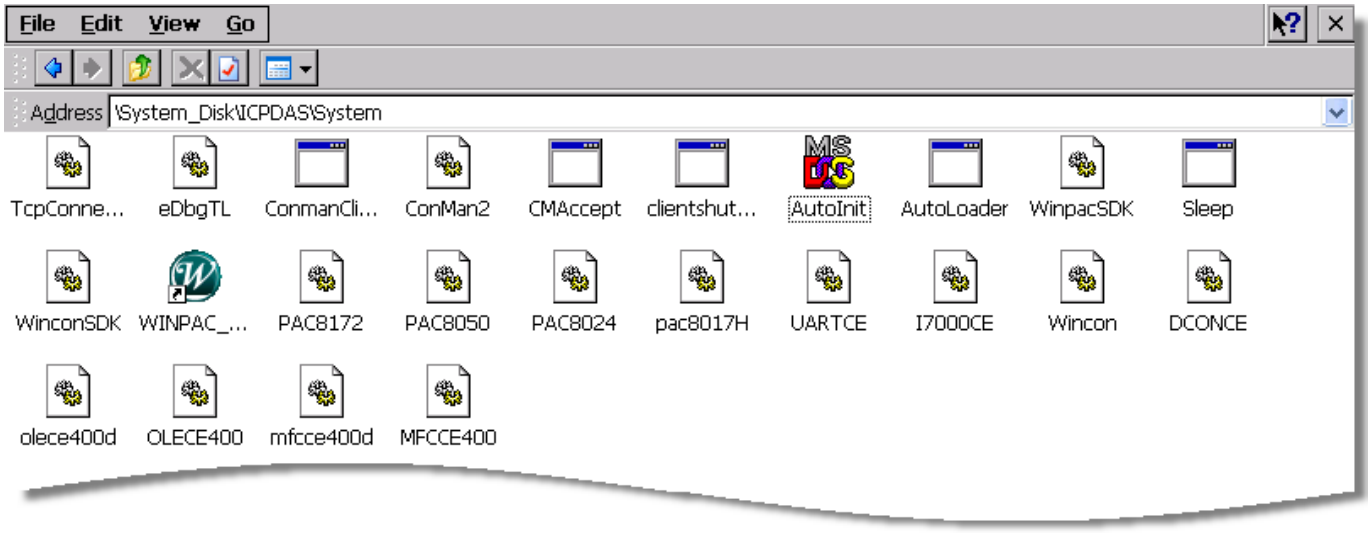
Step 8: On the “CAYMAN-ARMV4I_Release Properties” dialog, click the “OK” button



Step 9: On the “Options” dialog, click the “OK” button

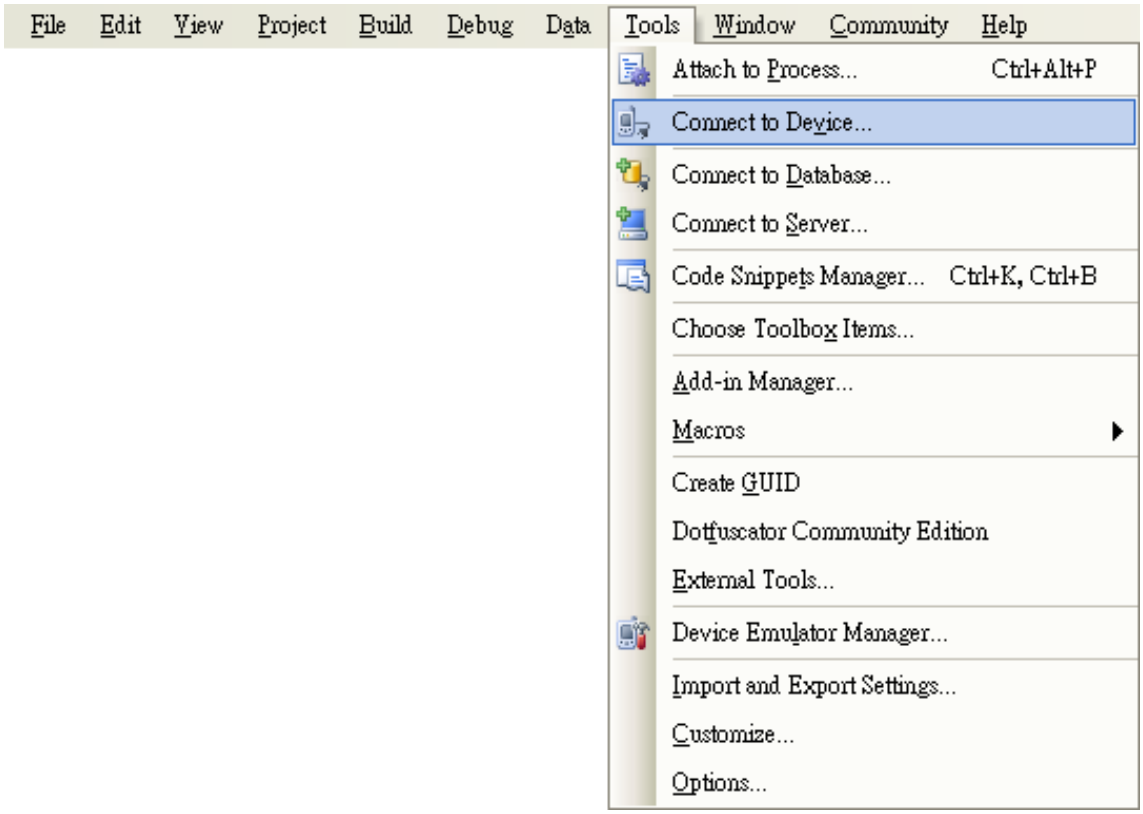


Step 10: On the WinPAC-8000 controller side, run the “CommanClient2” and the “CMAccept.exe” applications which is located at: \System_Disk\ICPDAS\System

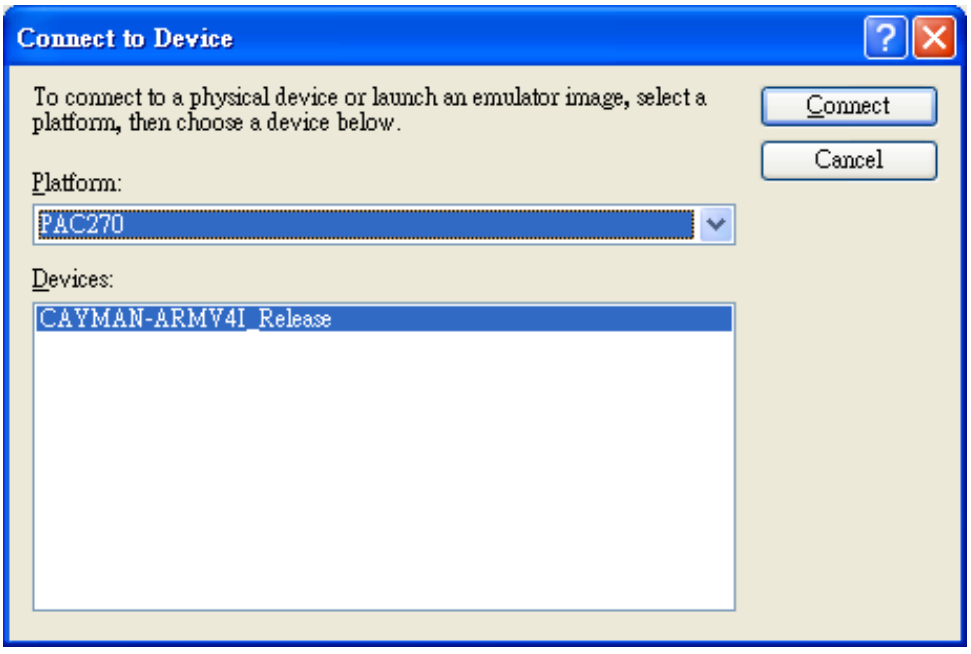


The step 11 and step 12 must be completed within three minutes.

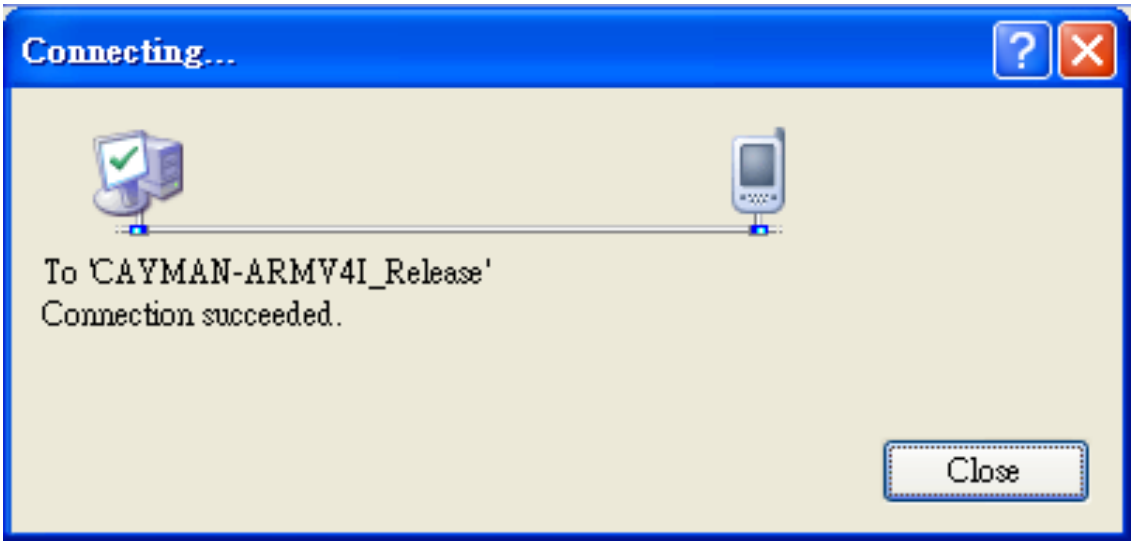
Step 11: On the “Tools” menu, click “Connect to Device…” command



Step 12: On the “Connect to Device” dialog, select “PAC 270” from “Platform” list and then click the “Connect” button



Step 13: On the “Tools” menu, click “Connect to Device…” command



Step 14: Connection established. Then you can debug on line.

E.4. How to recompile WinCon programs

To recompile Wincon programs to run on WinPAC, certain components of the programs require adjustments that divides into two parts:

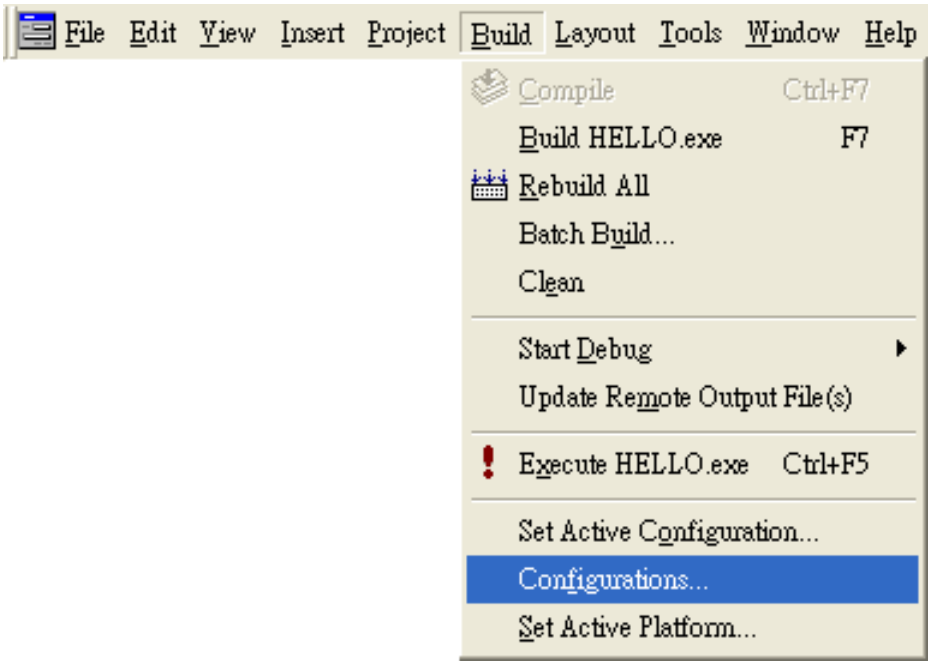
1. Compiler old programs which ran on Wincon 8x3x and 8x4x
2. Modify .vcp file to upgrade the old WinCon project

In general, you only need to do part 1, after this, if the program still can't be compiled to an application, the part 2 just need to do.

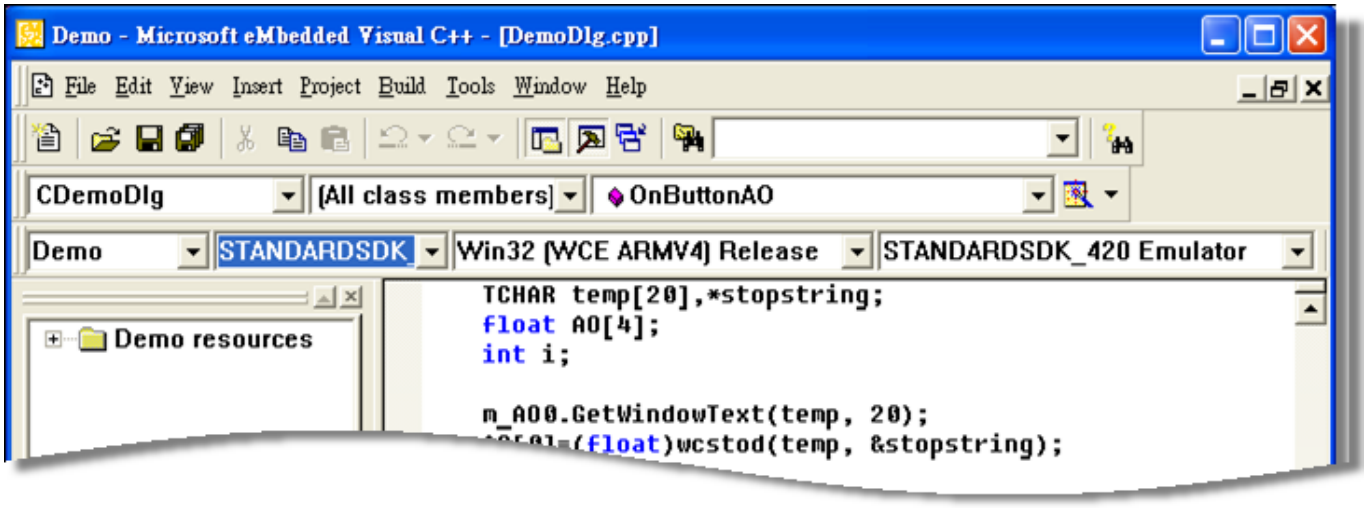
E.4.1. Compiler old programs which ran on Wincon 8x3x and 8x4x

Step 1: Open project which programmed in WinCon using eMbedded Visual C++

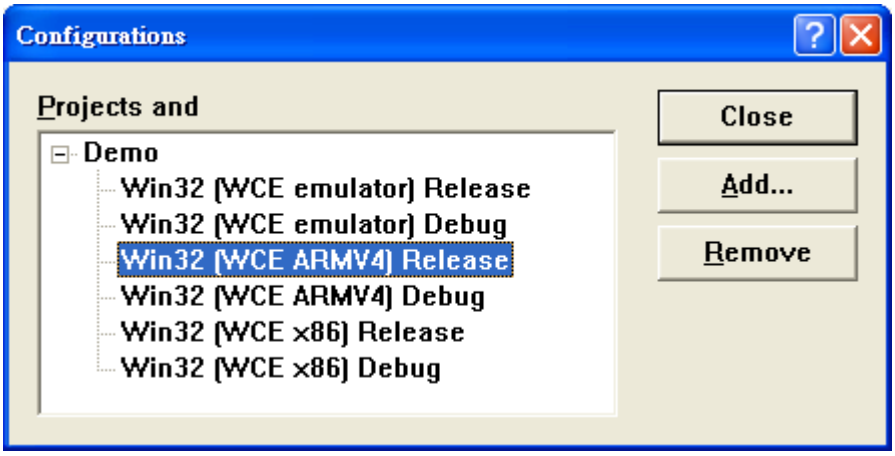
Step 2: On the "Build" menu, click "Configurations" command



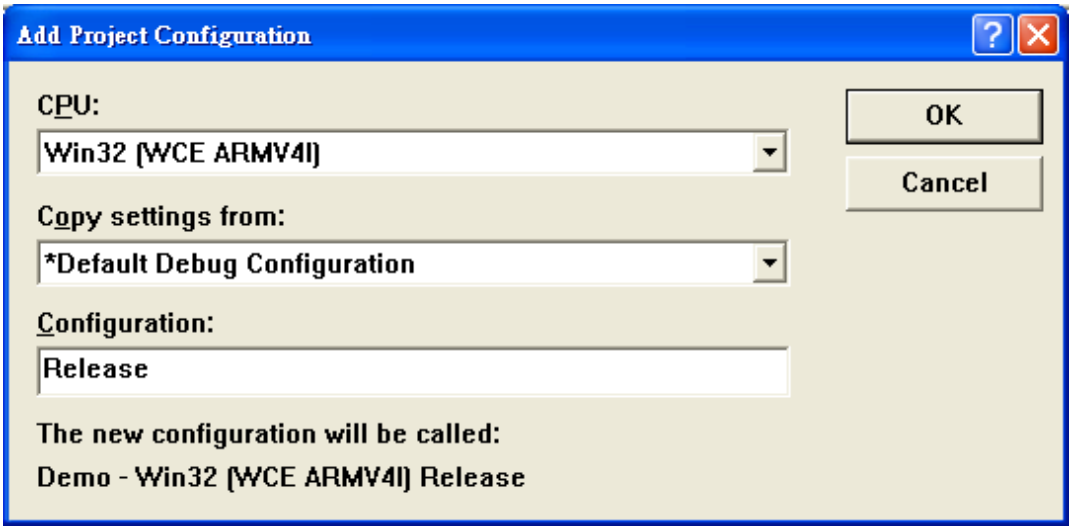
Step 3: Make sure the CPU type is “STANDARDSDK”



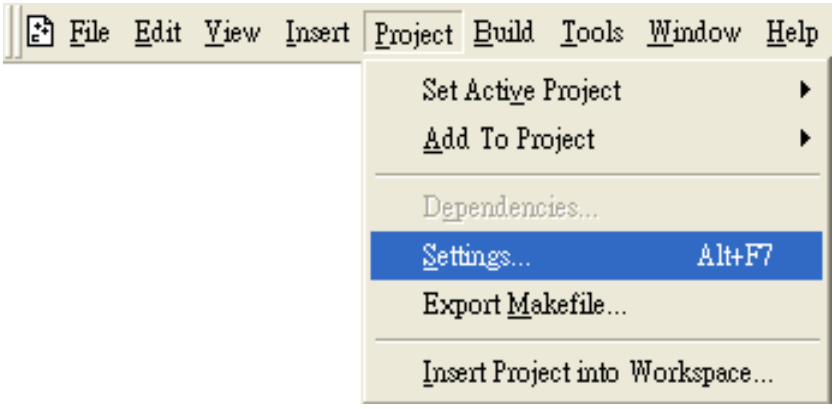
Step 4: On the “Configurations” dialog, click the “Add…” button



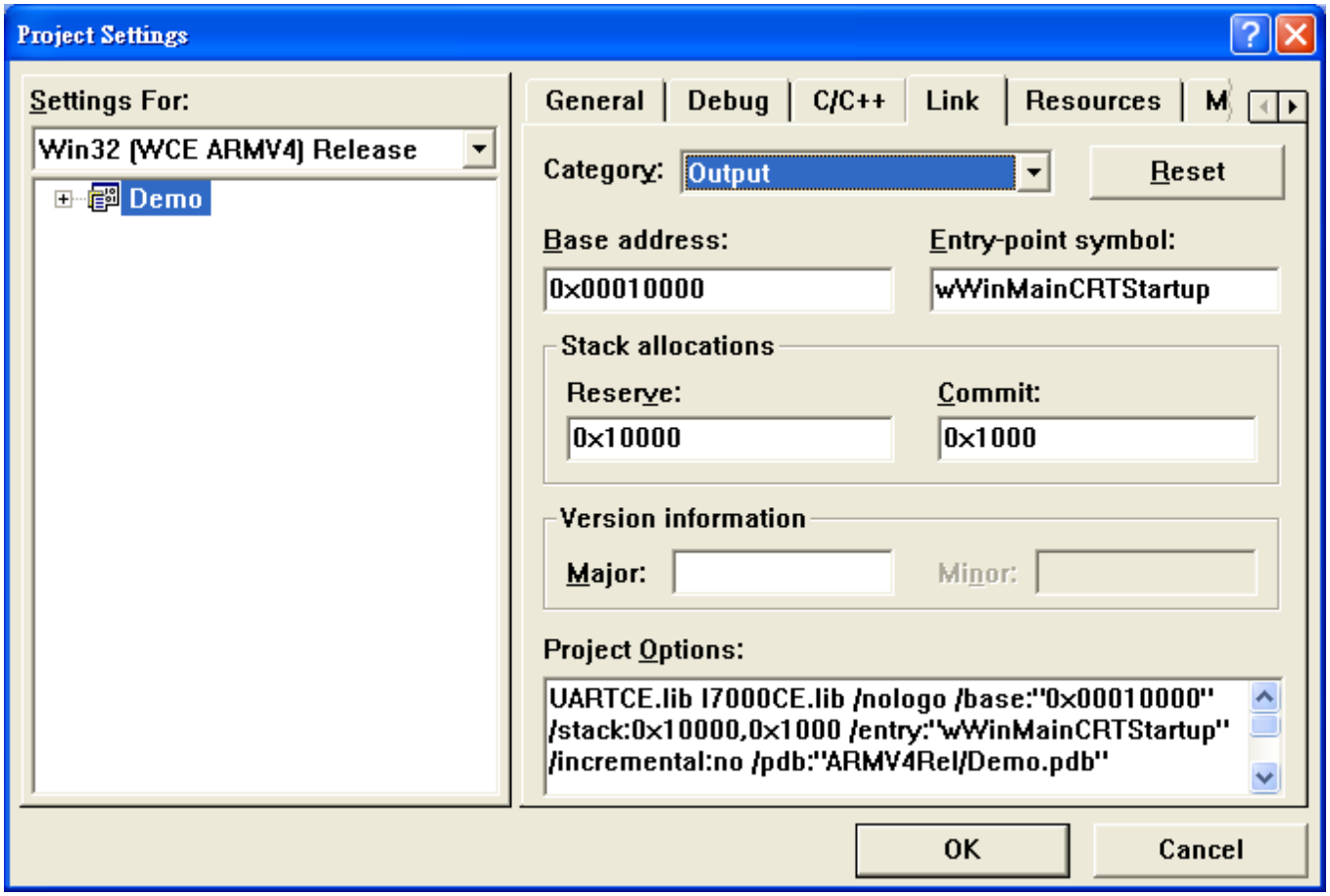
Step 5: On the “Add Project Configuration” dialog, choose one of the CPU type and then click the “OK” button.



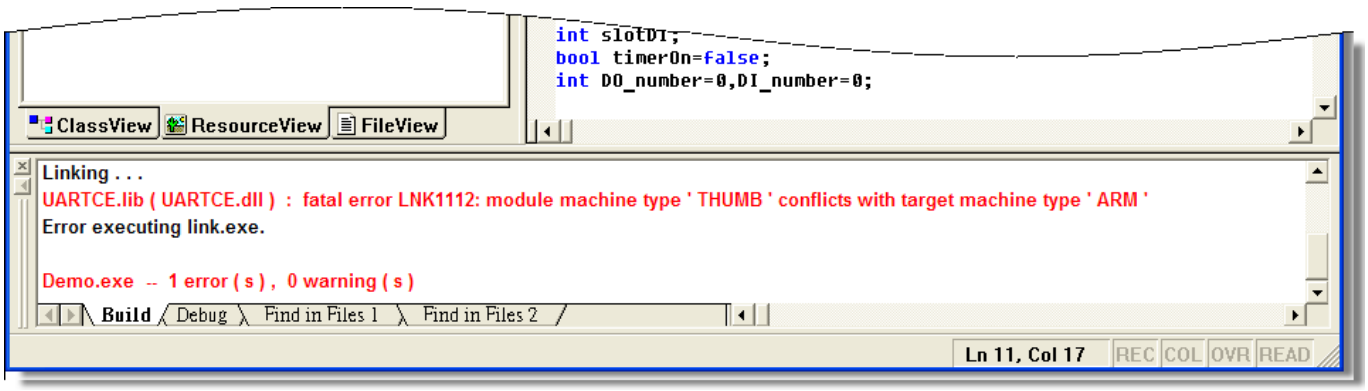
Step 6: On the “Project” menu, click “Settings…” command



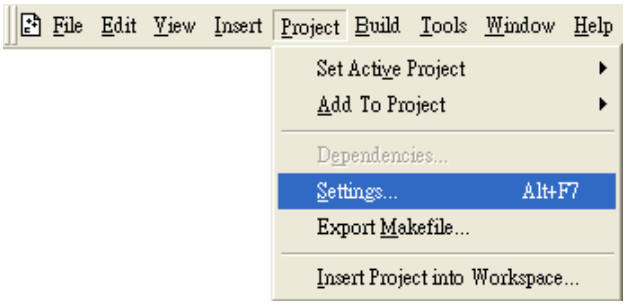
Step 7: On the “Project Settings” dialog, select the “Link” tab and change the value of the “Entry-point symbol” field, “WinMainCRTStarup” to “wWinMainCRTStartup”,



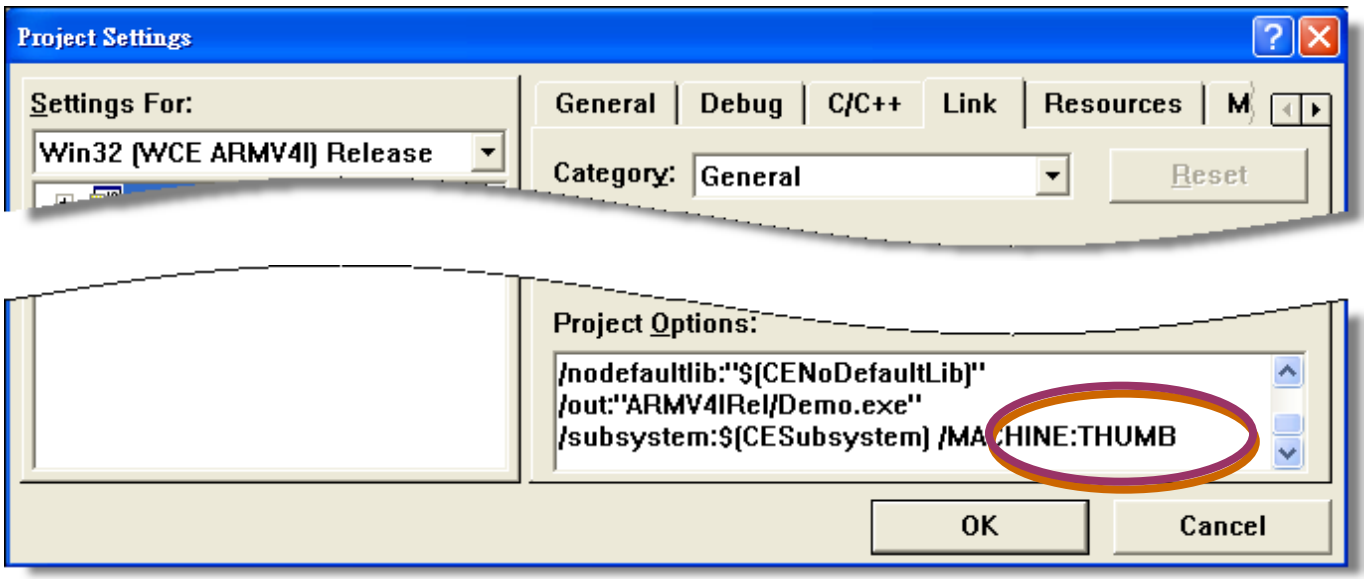
Step 8: After performing above-mentioned steps, build the project, your project should build success. If not, it will show error message as follow. Please continue with the following steps



Step 9: On the "Project" menu, click "Settings…" command



Step 10: On the "Project Settings" dialog, select the "Link" tab and change the value of the "Project Options" field, "ARM" to "THUMB", and then built the project



E.4.2. Modify .vcp file to upgrade the old WinCon project

Step 1: Open a text editor to modify the .vcp file

Step 2: In the .vcp file, replace “0xa301” with “0xa501”

Step 3: In the .vcp file, replace “ARMV4” with “ARMV4I”

Step 4: In the .vcp file, replace “MACHINE:ARM” with “MACHINE:THUMB”

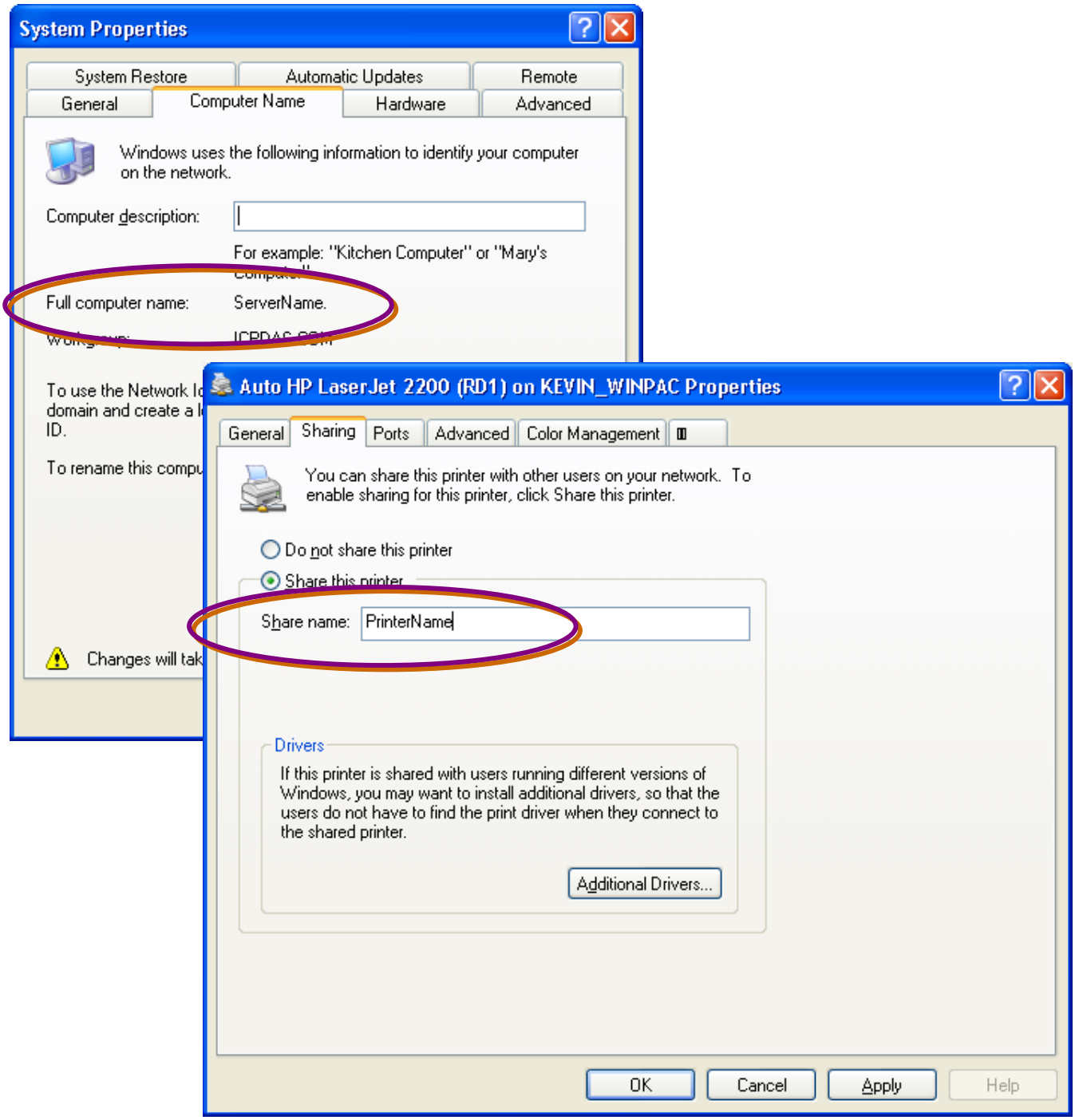
Step 5: Save the .vcp file just edited

Step 6: Open the old WinCon project and recompile it

E.5 How to use network printer

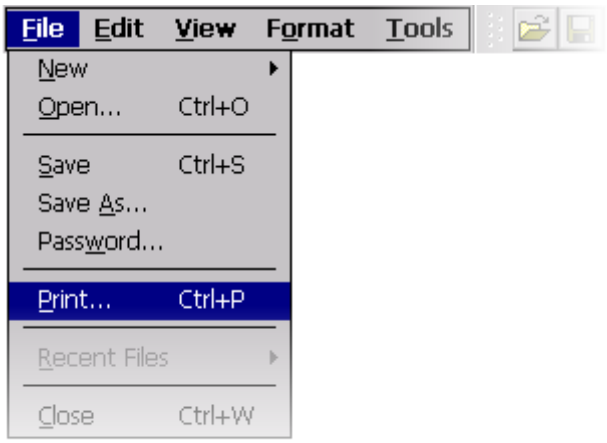
To use a shared network printer, please perform the following steps:

Step 1: On the Host PC side, check the name of the Host PC and the shared printer



Step 2: On the WinPAC-8000 controller side, open a WordPad format file

Step 3: Select "Print" from File menu

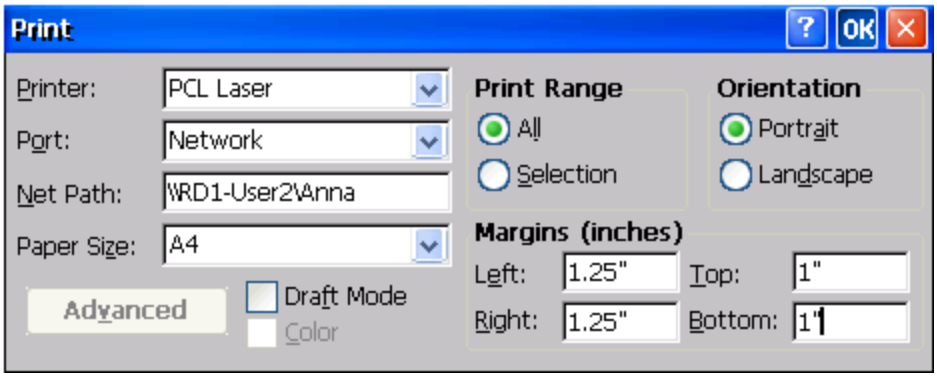


Step 4: Set up the printer

1. PCL Laser
2. Network
3. \\ServerName\PrinterName

   The "ServerName" is your PC's name or IP.

   The "PrinterName" is your printer's shared name of your PC.
4. Select the paper size



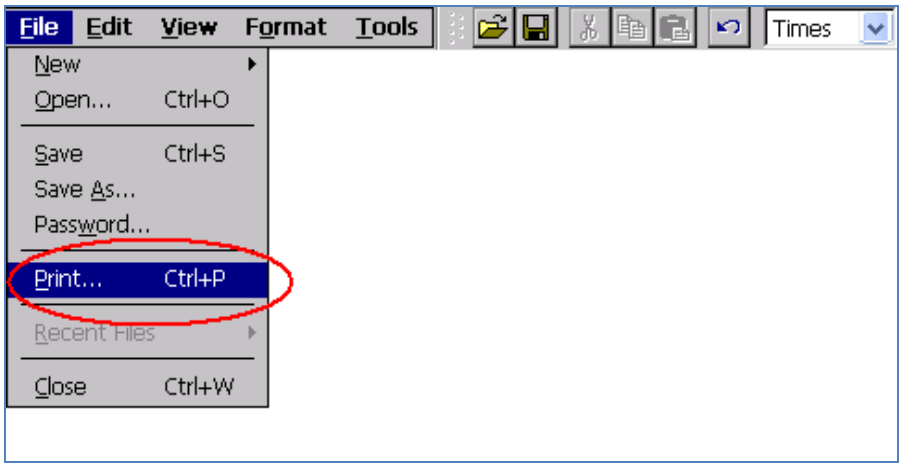
WinPAC-8000 only support HP Laser Jet Printer

E.6 How to use printer via USB

To use a shared network printer via USB, please perform the following steps:

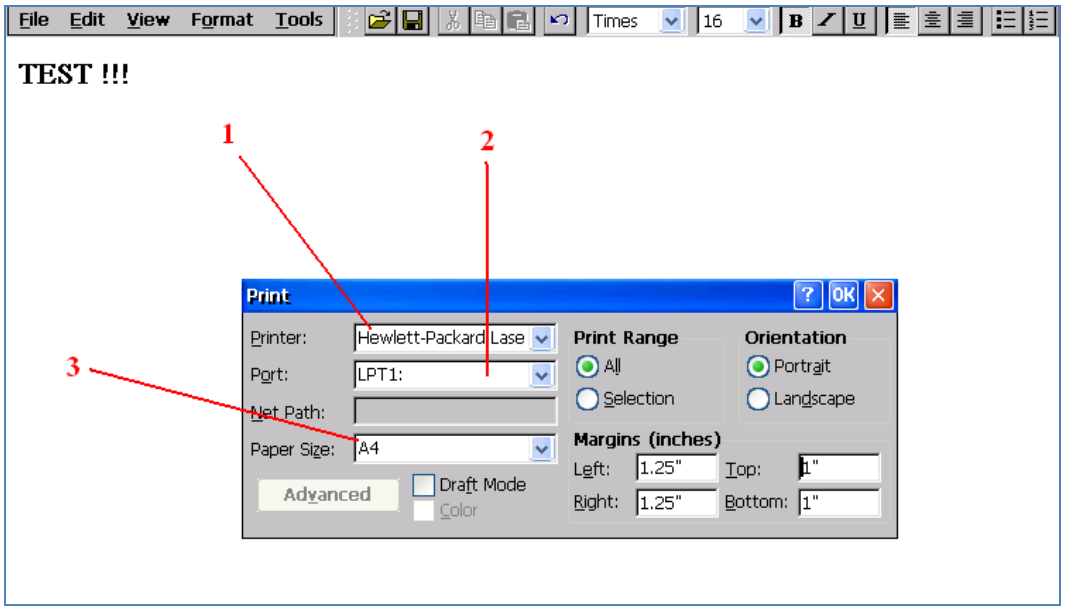
Step 1: On the WinPAC-8000 controller side, open a WordPad format file

Step 2: Select "Print" from File menu



Step 3: Set up the printer

1. Hewlett-Packard LaserJet
2. LPT1:
3. Select the paper size

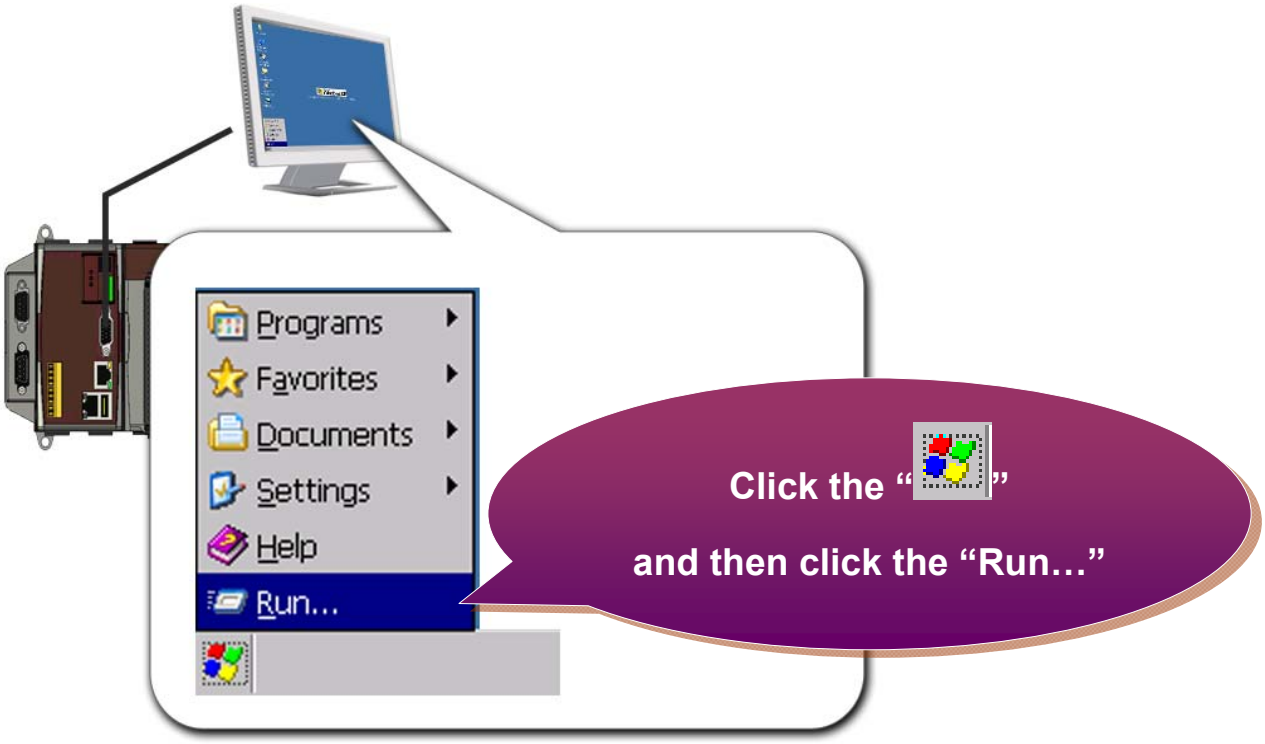


WinPAC-8000 only support HP Laser Jet Printer

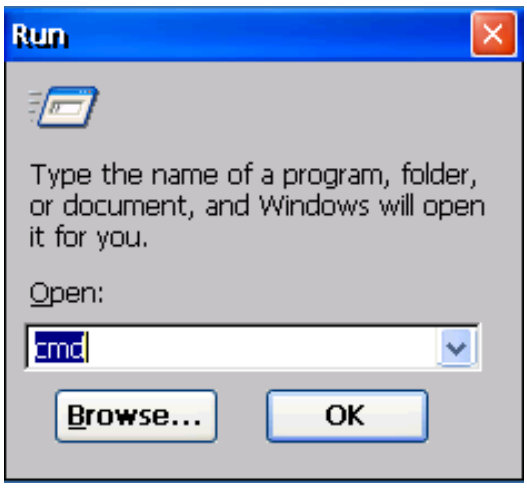
E.7 How to use services tool

The services tool can help you turn on, turn off and monitor the WinCE services.

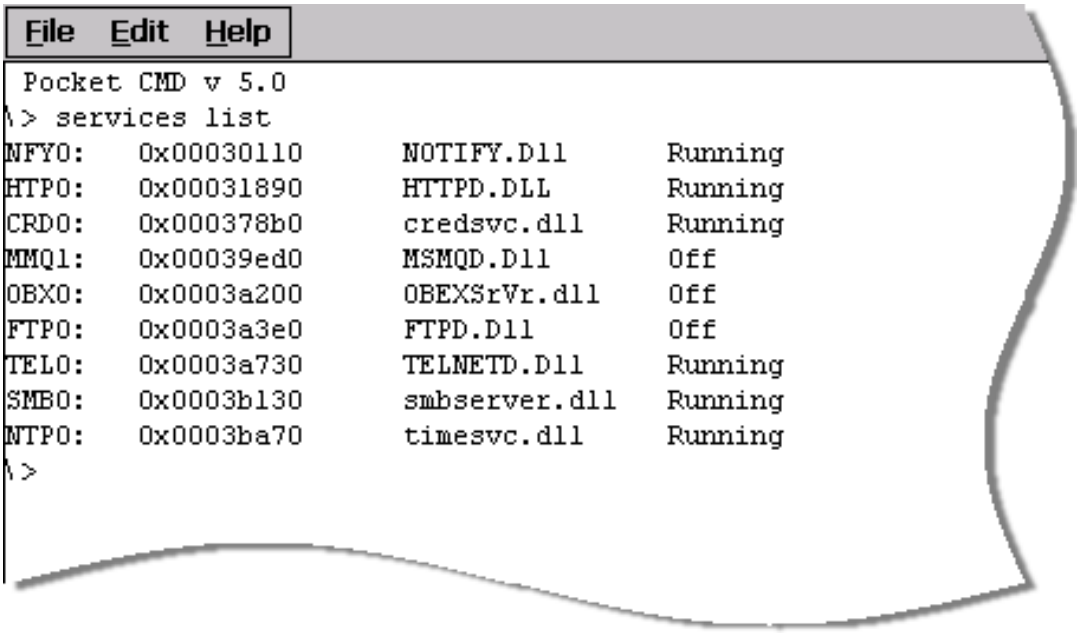
Step 1: On the WinPAC-8000 controller side, select the "Start" menu, and then click the "Run…" command



Step 2: On the "Run" dialog, type "cmd" in the field and then click the "OK" button



Step 3: On the command prompt, type “services list” to display all services

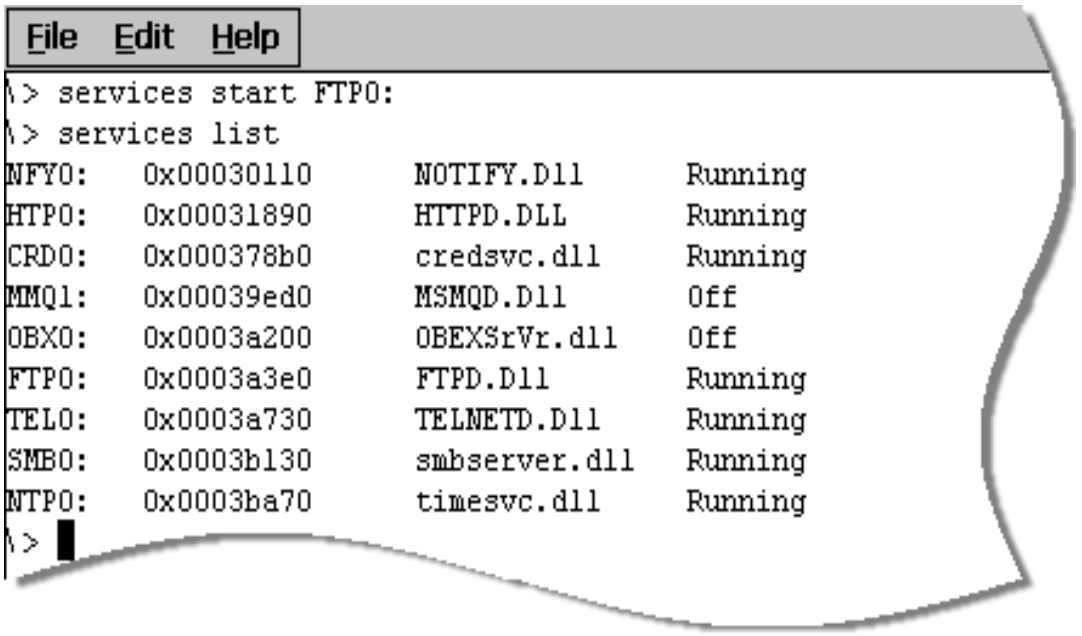


Step 4: Type the commands to configure service

[Syntax] services start <services name>

For example, turn on the “FTP” service:

services start FTP0:



For more information about using services tool, you just type “services help” command to get a complete list of supported commands..

File Edit Help

```
 Pocket CMD v 5.0
\> services help
Commands:
        help - print this text
        list - lists loaded services
        load <service name> - activates a service that is inactive
        stop <service instance> stops/pauses a service (does not unload)
        start <service instance> - starts/resumes a service
        refresh <service instance> - causes service to refresh its config
        unload <service instance> - causes service to be unloaded and removed
        register <service name> - service will be automatically loaded on next r
eboot
        unregister <service name> - service will not be automatically loaded on
next reboot
        command <service name> [arg1 arg2 ...] - send service-specific command t
o service
        help <service name> - get information on what service-specific commands
are supported

        <service name> - service's name in the registry (i.e. HTTPD)
        <service instance> - particular instantiation (i.e. HTP0:)
Flags:
        -f <file name>
        -s silent
        -d output to debugger
\>
```