W-8x8x Getting Started

(Version 1.3)

Hardware & Software & Application
Using I-7000/I-8000 Series Modules

Warranty

All products manufactured by ICPDAS Inc. are warranted against defective materials for a period of one year from the date of delivery to the original purchaser.

Warning

ICPDAS Inc. assumes no liability for damages consequent to the use of this product. ICPDAS Inc. reserves the right to change this manual at any time without notice. The information furnished by ICPDAS Inc. is believed to be accurate and reliable. However, no responsibility is assumed by ICPDAS Inc. for its use, or for any infringements of patents or other rights of third parties resulting from its use.

Copyright

Trademark

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

License

The user can use, modify and backup this software on a single machine. The user may not reproduce, transfer or distribute this software, or any copy, in whole or in part.

Contents

1 Introduction and hardware setup

Welcome to the W-8x8x user's Quick Start manual. ICPDAS provides this manual for new user how to install the hardware and software quick start.

1.1 Introduction

The main hardware of the W-8x8x embedded controller are depicted in figure 1.1, which include **VGA, USB, compact flash, series, Ethernet and an I/O slot standard interface** . ICPDAS also provides another remote I/O modules. You can communicate the remote I/O modules by pass through the COM3 (RS-485 network). On the standard Ethernet port, you can use the intranet and internet advantages such as www , ftp, SQL database application and so on.

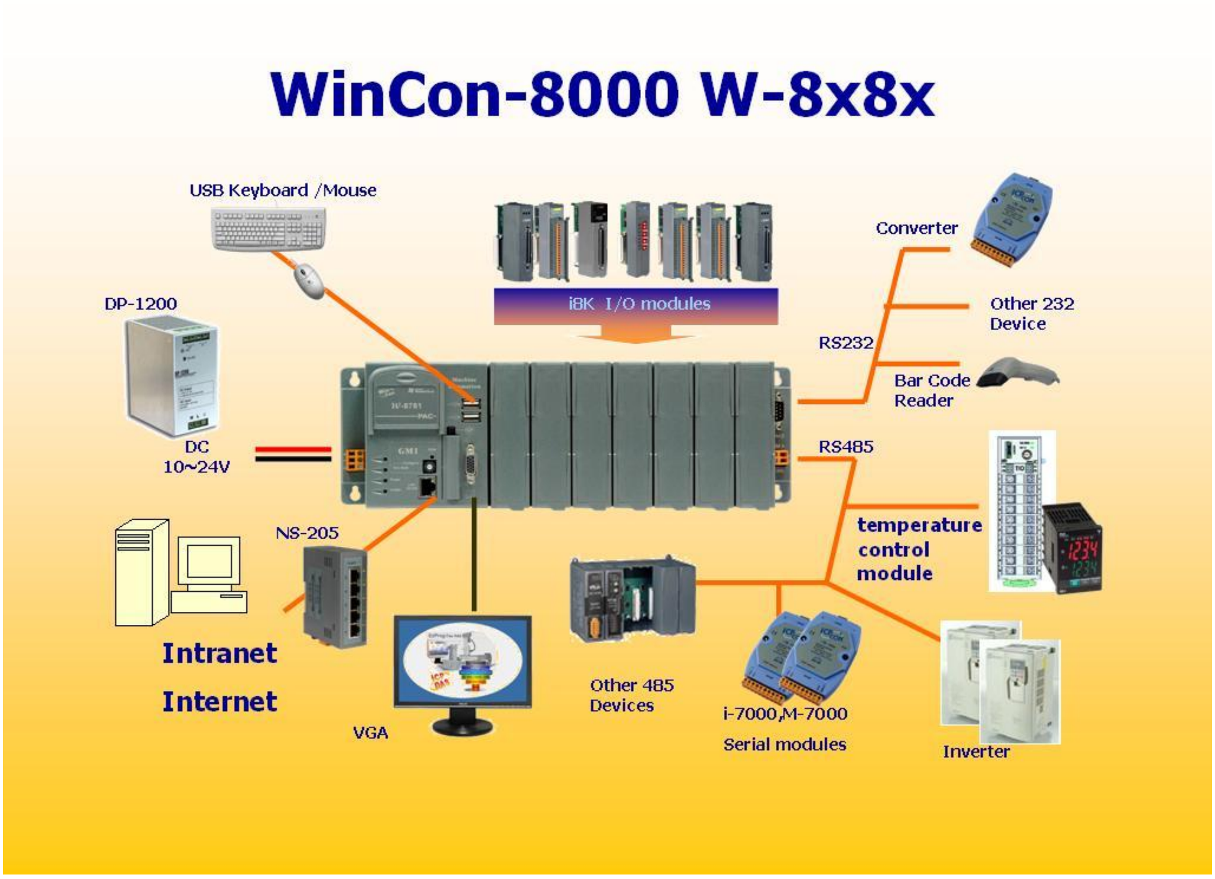


Fig. 1-1

For The software develop, ICPDAS provides two DLL files, namely the WinconSDK_DLL and the Wincon_DLL, for the I-8000 series modules which are used in the W-8x8x Embedded Controller. The **WinconSDK_DLL** has all the essential DLL functions designed for the I-8000 series modules for Microsoft WinCE.Net platform. It can be applied on embedded Visual C++, and even on the newer platforms. Users can easily develop WinCE.NET applications on W-8x8x by using this toolkit. The various functions in **WinconSDK_DLL** are divided into the following sub-group functions for easy use in different applications. For more detail please refer to the **WinconSDK.lib for embedded Visual C++**

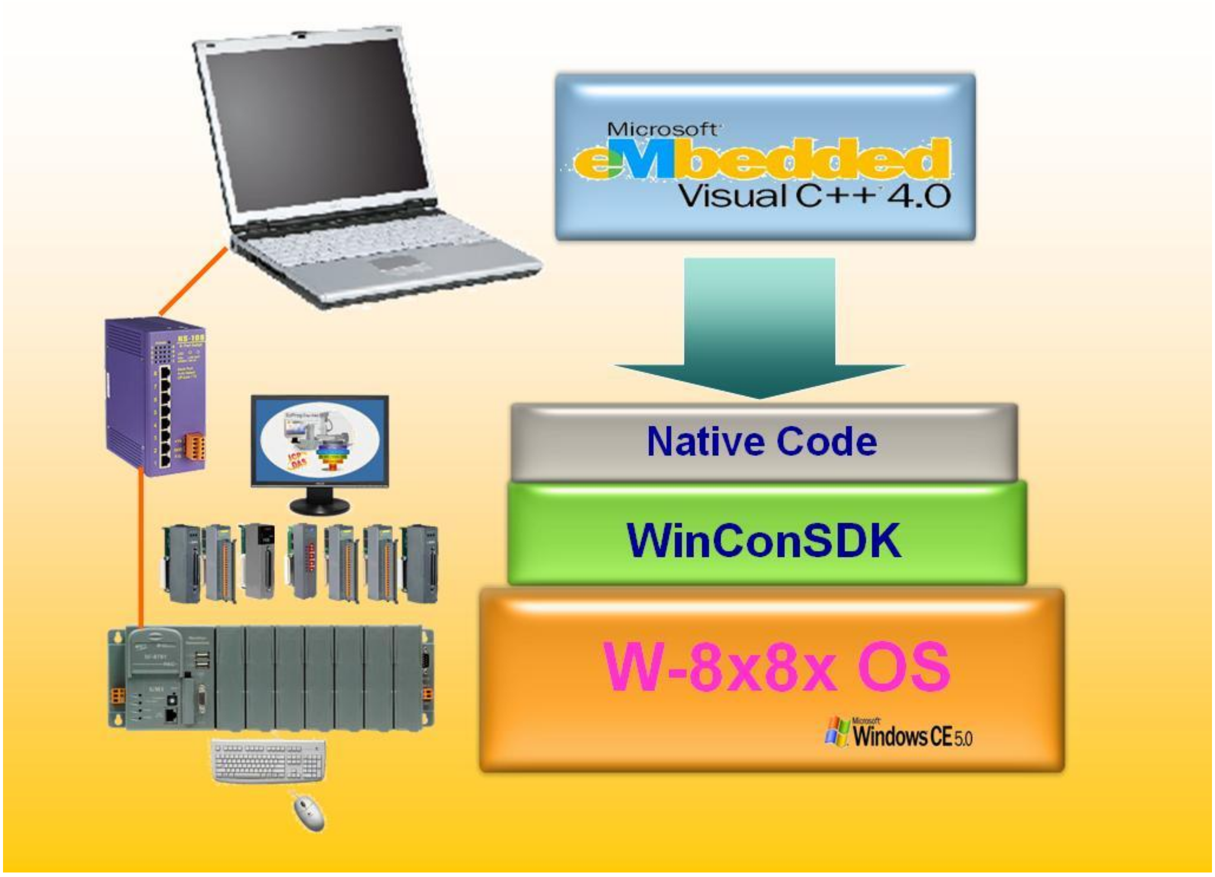


Fig. 1-2

1.2 Hardware Setup(for W-8X8X)

There is the W-8781 hardware setup sample (Fig. 1-3). In the right side connect the standard computer interface like 15 pin VGA port monitor (can support the touch screen as **HMI** device),USB, and Ethernet. In the middle and left area, you can plug in the **ICPDAS's I-8K** series I/O modules. The **W-87XX** can plug in 7(max.) local I/O modules and **W-8381** for 3(max.) local I/O modules. On the under lift side, The **COM2** is a standard 5 wire **RS232** com port. You can connect the standard RS232 device (modem …). The **COM3** is standard **RS485** port. You can connect to the RS485 devices like **ICPDAS's** products **I-7K** series I/O modules and **I-87K**X expansion unit.

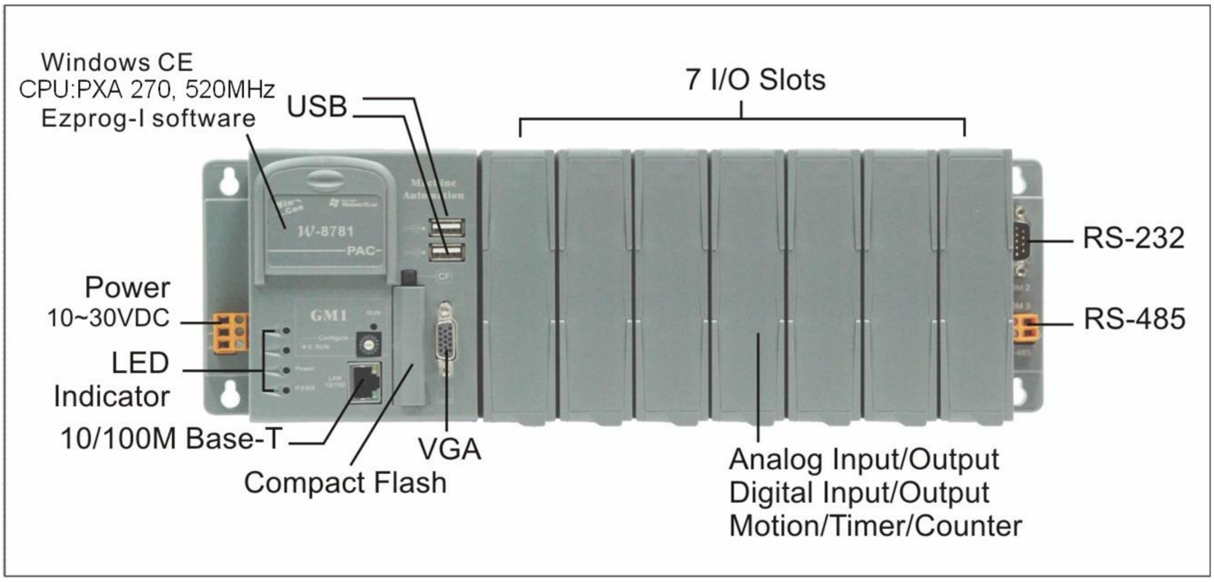


Fig. 1-3

By the Ethernet ,you can communicate to the Ethernet devices such as I-7188EX(palm size Embedded Controller), SQL Server , mail Server, MMS Server, iPush Server and so on. (Fig. 1-4)

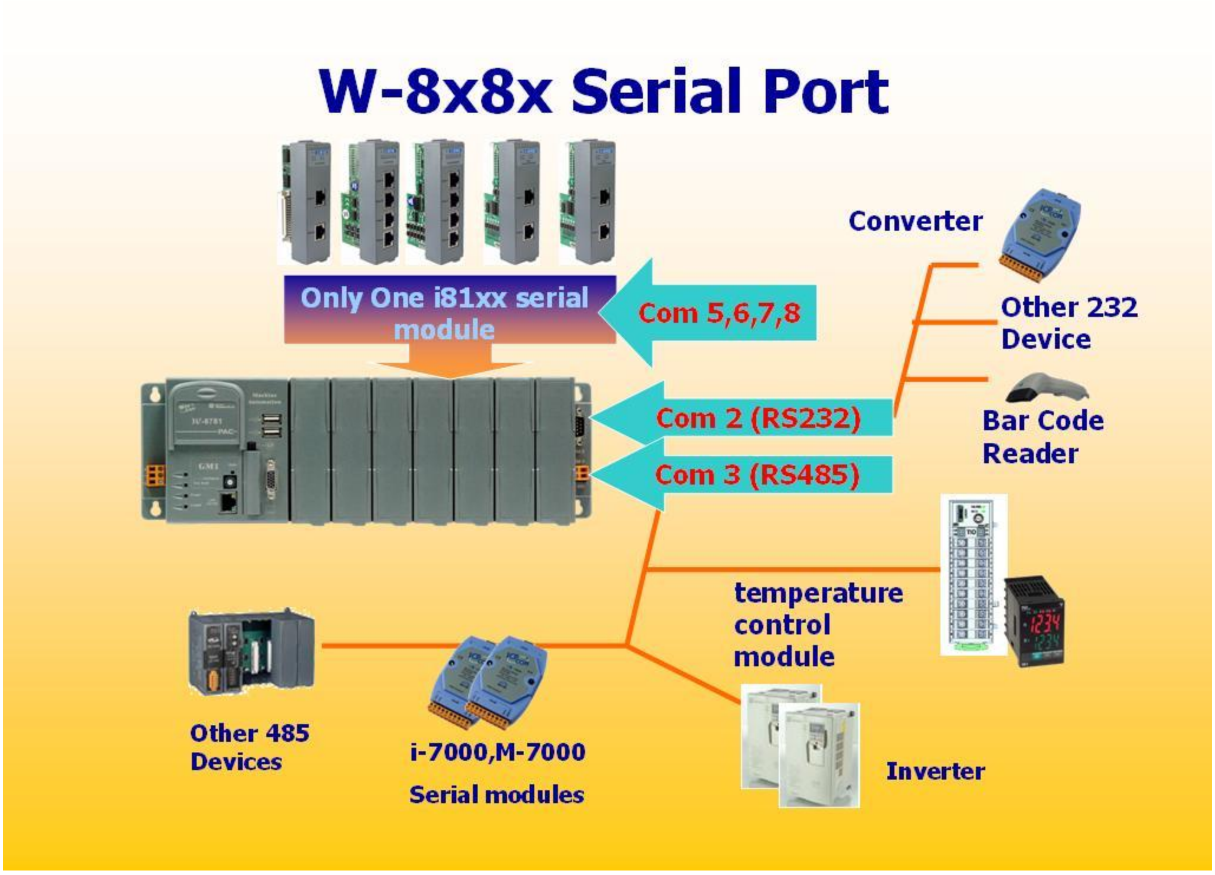


Fig. 1-4

If the devices connect was completed, please connect the DC 10V~30V input to the under right side and frame ground to the earth ground finally(if you have earth ground). Please attention that Wincon’s power consumption is low(typical 0.15A at 24VDC) , But On the **Power On** moment (in 200ms), the Wincon need more current to start up (typical 1.2A at 24VDC).

1.3 Specifications

Main Control unit

- xScale PXA-270 CPU, 520MHz
- SDRAM : 128M bytes
- Flash : 32M bytes
- FRA M : 128K bytes
- 64-bit hardware unique serial number
- Built-in Watchdog Timer
- Real Time Clock
- 1 VGA port :320x240x16 to 1024x768x16 Default is 800x600x16
- 1 Compact Flash slot : CF memory card
- Reset button(reboot after holding on 3 secs)
- Power LEDs
- Multiple Booting Selection :

W-8X8X

- USB 1.1 Host x 2
- 1 x RJ45, 10/100BaseT

Cabinet

- COM0: Internal use
- COM1: Reserved
- COM2: 9-pin D-Sub (RS-232)
- COM3: 2-wire Terminal Block (RS-485)
- FRnet(option)
- I/O Expansion Slot :
  3 - slot for W-8**38**1
  7 - slot for W-8**78**1

- Power Supply : 20W, Unregulated + 10Vdc +30Vdc

- Environment :
  Operating Temp. : -25℃ to + **75°C (Depends on I/O modules)**
  Storage Temp. : -30℃ to +85℃
- Humidity : 5~95%

- Dimensions :
  230.25x110x93.8 (3 slot)
  354.26x110x93.8(7 slot)

- I/O module(optional)
  I-8000 series modules, which include DI,DO,AO,AI…
  I-7000 series modules, which include DI,DO,AO,AI…

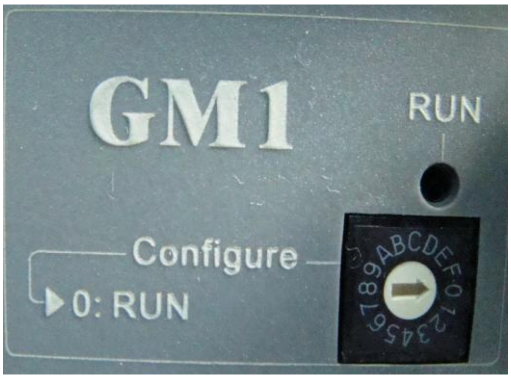
For more information please refer to relative catalog or http://www.icpdas.com

1.4 System Recover

1.4.1 System Recover form CF card

To prevent the disappear of data in Flash in W8x8x for accident or incorrect operation(exp: the unstable power) from operating not properly ,the W8x8x has the function of System Recover .User can recover the Flash's data that include the WinCE's Kernel and the WinCE's Registry before entering the OS of WinCE by System Recover.

Because the System Recover of W8x8x is in the Bootloader stage ,it's needless to enter the WinCE's OS or to operate by Keyboard and Mouse. The System Recover is writing the file W8x8xBackup.bak at the root path of CF card back to the flash in W8x8x. The thing has to attention the file's name isn't include date and time ,and the file must be made by Wincon Utility's back up system in W8x8x. Besides the name's date and time of the file, it will maybe lead the W8x8x to operate not properly if the User modified the content of the file that is for recover.



The steps of recover:

Step 1: Power off the W-8x8x

Step 2: Put the CF card in W-8x8x that has W8x8x_Backup.bak, and something have to attention:

→ the file's name must be "W8x8x_Backup.bak"

→ W8x8x_Backup.bak must be at the root of CF card

→ W8x8x_Backup.bak is made by WinCon Utility's backup function and user didn't change it's content, only have to move the date and time in file's name.

Step 3: Turn the Rotary Switch to "9"

Step 4: Hold on the Reset Button on the panel of W8x8x,and power on until the led stop blinking.

Step 5: To Observe the led until it has been snuffed out ,after rebooting the system should be recovered.

1.4.2 Recover Registry

The registry is the major system setting in WinCE, it's permitted that user restore the registry to the factory settings, if the user set error or forgot some settings(exp:IP,) to cause the system abnormal . The steps as follow:

Step 1: Power off the W-8x8x

Step 2: Turn the Rotary Switch to ”B”

Step 3: Hold on the Reset Button on the panel of W8x8x,and power on until the led stop blinking.

Step 4: To Observe the led until it has been snuffed out ,after rebooting the system should be recovered.

1.4.3 The Selections of Boot

The W-8x8x can be boot by:

1. the CE kernel in ***Flash***
2. the CE Kernel image(NK.bin) in ***CF card***
3. the CE Kernel image(NK.bin) in ***USB drive***

If user want to boot with CF card in the W8x8x or USB disk ,it's necessary to put the CE kernel image file “NK.bin” on the root path of CF card or USB disk.

Boot with the Kernel Image in ***Flash*** :

Step 1: Power off the W-8x8x

Step 2: Turn the Rotary Switch to ”0”

Step 3: Power on and waiting for the boot finish .

Boot with the Kernel Image(NK.bin) in ***CF card***:

Step 1: Power off the W-8x8x

Step 2: Turn the Rotary Switch to ”2”

Step 3: Power on and waiting for the boot finish (about 45 sec).

Boot with the Kernel Image(NK.bin) in ***USB disk***:

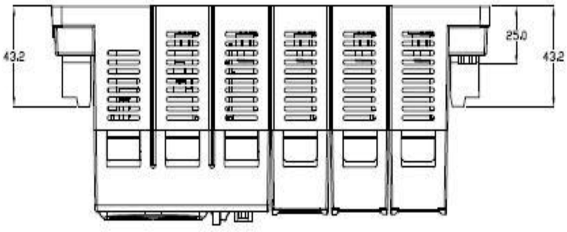
Step 1: Power off the W-8x8x

Step 2: Turn the Rotary Switch to ”4”

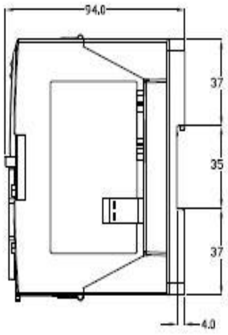
Step 3: Power on and waiting for the boot finish (about 120 sec).

1.5 Dimensions

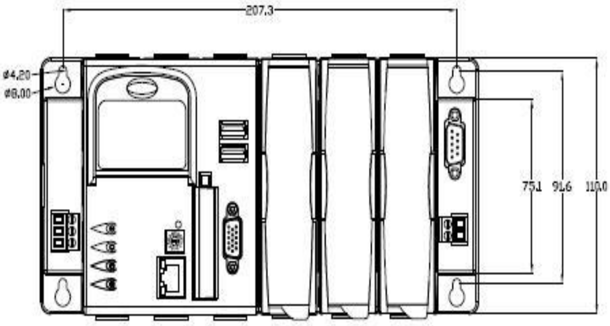
3 slots :



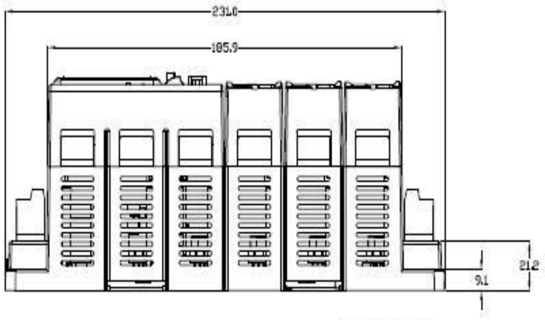
Top View



Right Side View

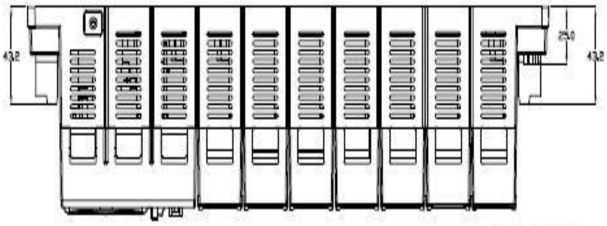


Front View



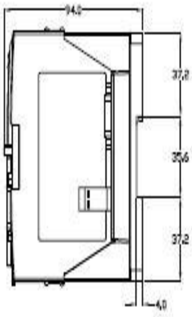
Bottom View

7 slots :

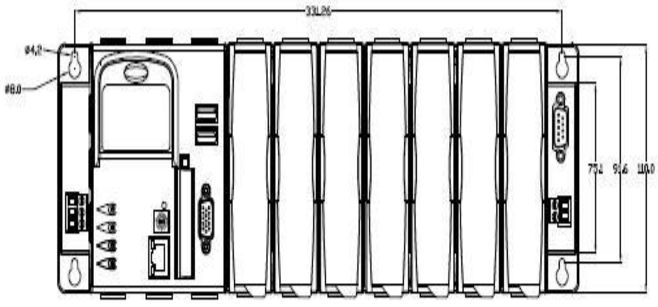


Top View

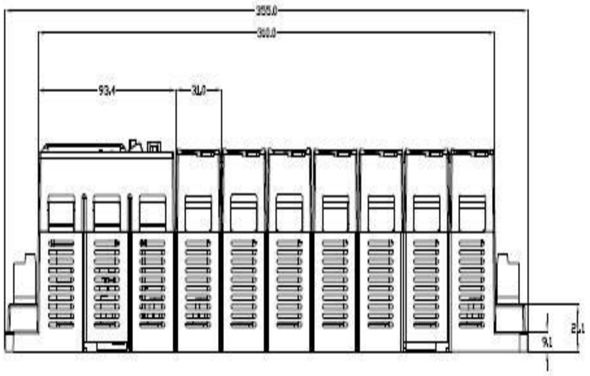
Unit: mm



Right Side View



Front View



Bottom View

1.6 What's News

Added 1.3 Specifications

Added 1.4 System recover

Added 1.5 Dimensions

Added 2.2.2 Touch Panel driver in General setting

Added 2.2.6 Enable one of communication modules in W8x8x in New Card Wizard

1.7 Package List

- One set of W-8x8x hardware
- One Compact Flash card for storing system files.
- One software utility CD

2. W-8x8x System Settings

In this section, we will explore how to set the Windows CE System and the “WinCon Utility” for the W-8x8x embedded controller. You can change configurations, such as the system time or network setting of the W-8x8x through the Windows CE control panel. WinCon Utility allows you to view W-8x8x’s information or save the current system configuration into Windows CE OS image.

2.1 Windows CE Settings

Setting Up the System Time

You can setup a new date or time in the Windows CE system by using the following steps:

1. Choose **Start → Settings → Control panel** to open the Control panel dialog.

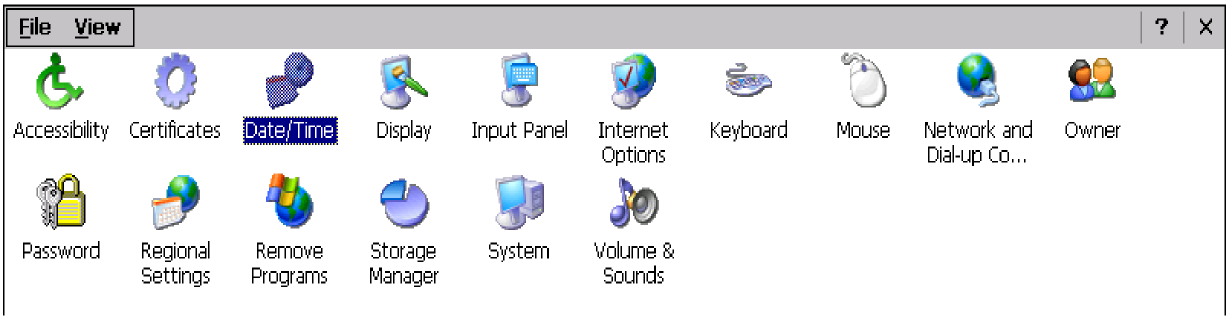
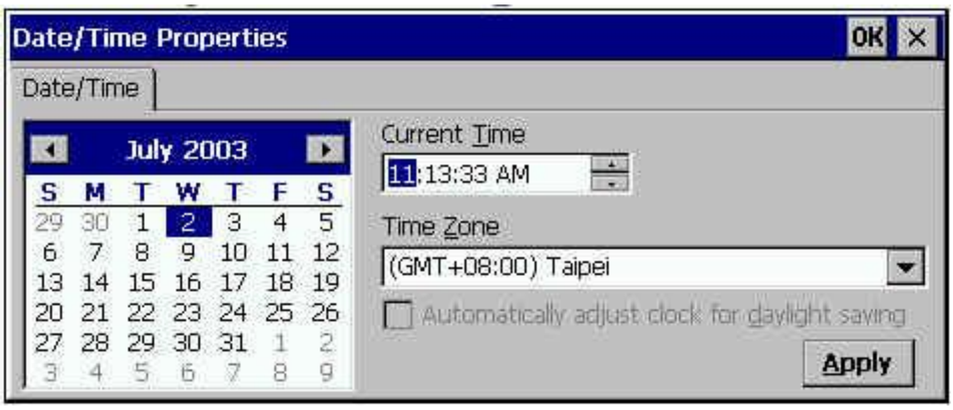


Fig. 2-1

2. Double click the Date/Time icon on the Control panel dialog.



July 2003

S	M	T	W	T	F	S
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

Current Time: 11:13:33 AM

Time Zone: (GMT+08:00) Taipei

□ Automatically adjust clock for daylight saving

Apply

Fig. 2-2

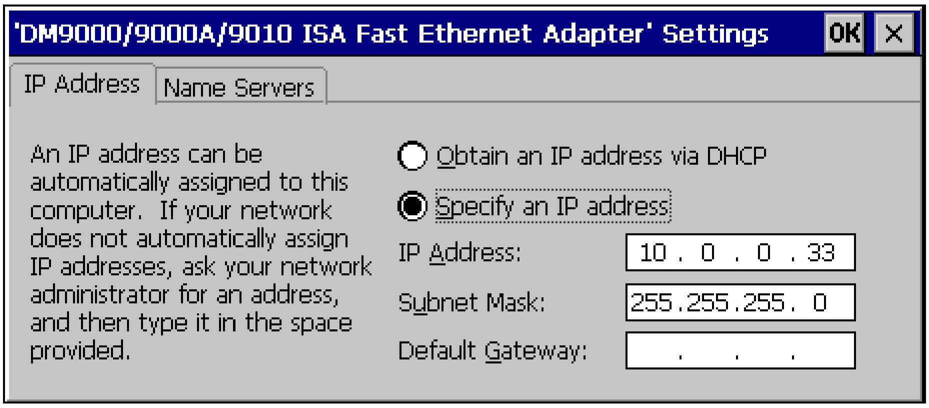
3. When the Date/Time Properties dialog displays, set the date or current Time and click the Apply button to set your system date and time.

> ✍ **Note:** If you have changed any value of the date and time. You must save the registry by means of WinCon Utility tools. For more information about WinCon Utility tools, please refer to the WinCon Utility section.

Setup the network

Generally, most users don't need to setup the network because DHCP is the default setting. However, if your network system does not contain a DHCP server, you need to configure the network setting by using the manual method. The following steps demonstrate the procedure for how to configure the network system.

1. Choose **Start → Settings → Network and Dial-up Connections** on the Windows CE desktop to open this dialog.
2. Double click the ETH11 icon to open the “DM9000/9000A/9010 ISA Fast Ethernet Adapter Settings” dialog.



An IP address can be automatically assigned to this computer. If your network does not automatically assign IP addresses, ask your network administrator for an address, and then type it in the space provided.

○ Obtain an IP address via DHCP

● Specify an IP address

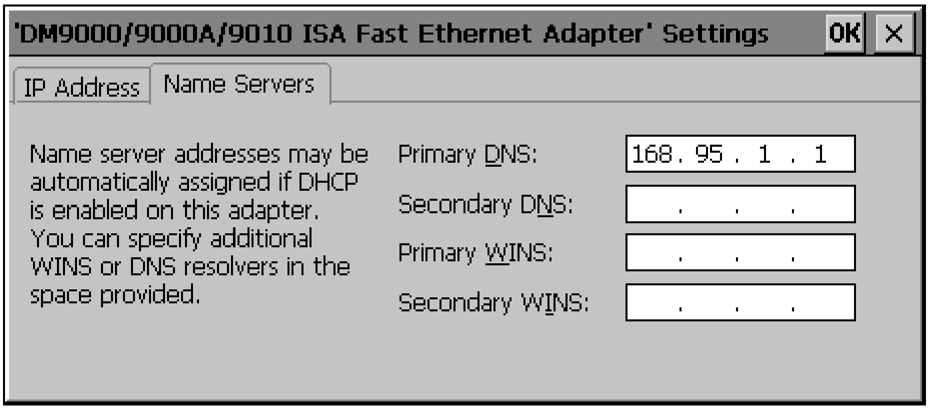
IP Address: 10 . 0 . 0 . 33

Subnet Mask: 255.255.255. 0

Default Gateway: . . .

Fig. 2-3

3. When the “DM9000/9000A/9010 ISA Fast Ethernet Adapter Settings” dialog displays (see figure), click (enable) the “Specify an IP address” radio button in the IP Address tab and type in the IP Address, Subnet Mask, and Default Gateway into the respective fields.
4. Choose the “Name Servers” tab and also type in the Primary DNS, Secondary DNS, Primary WINS, and Secondary WINS into the respective fields, as shown in the figure below.



Name server addresses may be automatically assigned if DHCP is enabled on this adapter. You can specify additional WINS or DNS resolvers in the space provided.

Primary DNS: 168 . 95 . 1 . 1

Secondary DNS: . . .

Primary WINS: . . .

Secondary WINS: . . .

Fig. 2-4

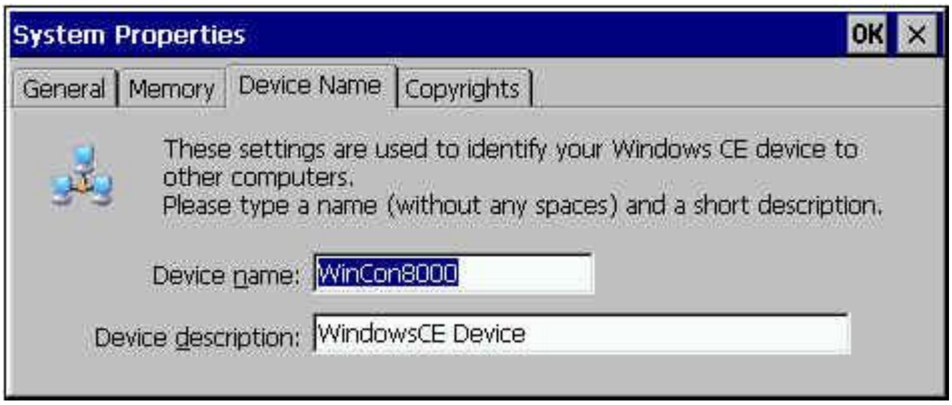
5. Click OK.

✍ **Note**: If you have changed any value of network configuration, you must save the registry by means of WinCon Utility tools. For more information about the WinCon Utility tool, please refer to the WinCon Utility section.

Setting up the Device Name

You can configure W-8x8x to have the device name of your choice. To change the device name please refer to the following steps:

1. Choose **Start → Settings → Control panel** to open the Control panel dialog.
2. Double click the System icon on the Control panel dialog to open the System Properties.
3. When the System Properties dialog is displayed (see figure), select the Device Name tab in the dialog window.



These settings are used to identify your Windows CE device to other computers.
Please type a name (without any spaces) and a short description.

Device name: WinCon8000

Device description: WindowsCE Device

Fig. 2-5

4. Type your preferred Device Name in the Device Name box, and click OK.

✍ **Note**: If you have changed any information of the Device Name, you must save the registry by means of WinCon Utility tools. For more information about the WinCon Utility tool, please refer to the WinCon Utility section.

Here, we only provide some demonstrations for configuring your settings. The configuration steps and operation methods are the same as with the windows system. However, you need to keep in mind **"if you have changed any setting on W-8x8x embedded controller, you would need to use the WinCon utility to save the current setting into non-volatile internal memory"**. Otherwise, when you restart the system, the setting will not change.

2.2 WinCon Utility

The WinCon Utility provides many tools to save/view the system information registry and to setup the HTTP/FTP path and update non-volatile internal memory within the W-8x8x embedded controller. This handy utility (WinCon Utility 1.exe located in the Compact Flash/ICPDAS/Tools directory) should be located in the computer's Program group. Therefore, you can launch it on the computer through **Start → Programs → WinCon Utility** menu. The WinCon Utility provides many functions within the following five tabs:

- System Configuration Tab
- General Setting Tab
- Auto-execute Tab
- Version Update Tab
- Server Setting Tab
- New Card Wizard Tab

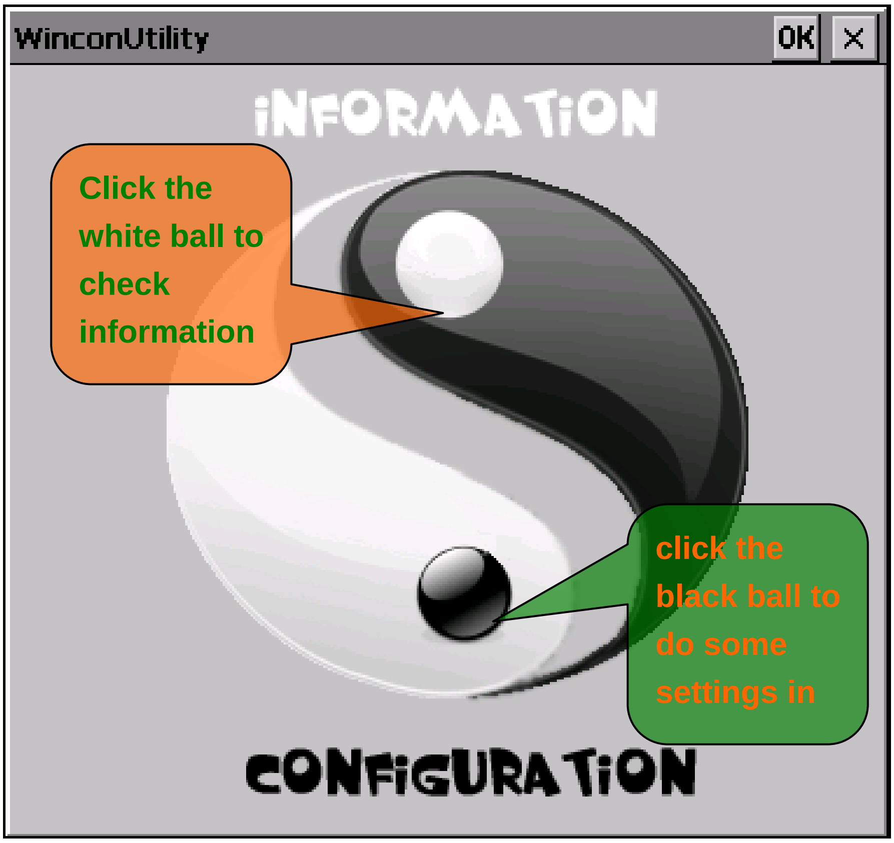
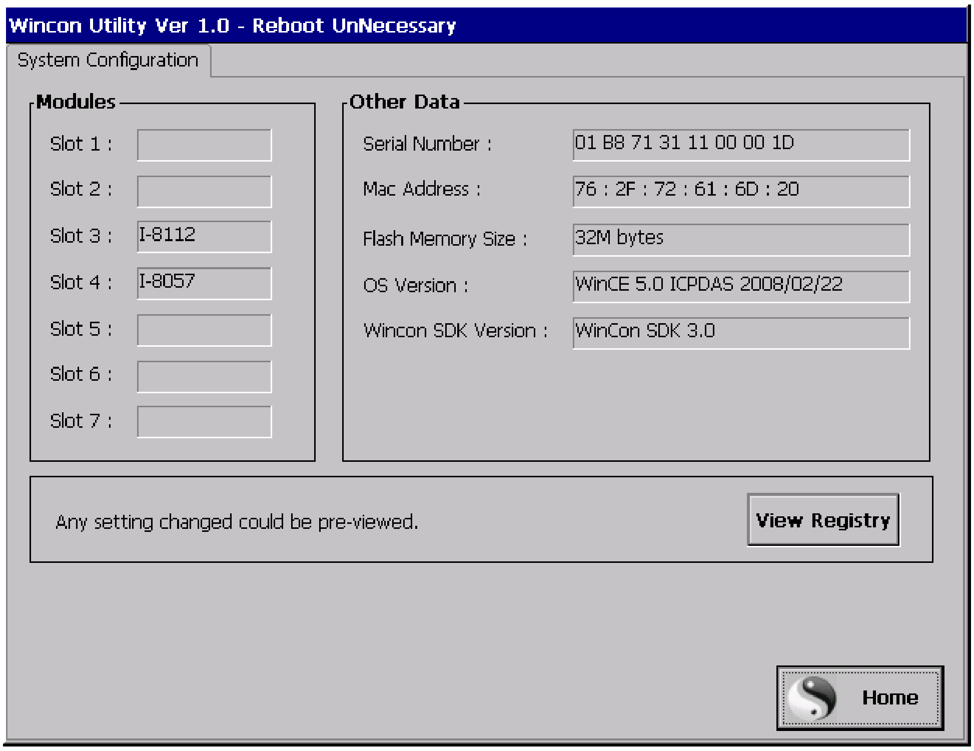

WinconUtility
OK
×
iNFORMATiON
Click the white ball to check information
click the black ball to do some settings in
CONFiGURATiON

Fig. 2-6

2.2.1 System Configuration Tab

Click the location of white ball will enter ***System Configuration Tab***. The System Configuration tab allows you to view the information in the Wincon-8000 embedded controller system.



Wincon Utility Ver 1.0 - Reboot UnNecessary

System Configuration

Modules

Slot 1 :

Slot 2 :

Slot 3 : I-8112

Slot 4 : I-8057

Slot 5 :

Slot 6 :

Slot 7 :

Other Data

Serial Number : 01 B8 71 31 11 00 00 1D

Mac Address : 76 : 2F : 72 : 61 : 6D : 20

Flash Memory Size : 32M bytes

OS Version : WinCE 5.0 ICPDAS 2008/02/22

Wincon SDK Version : WinCon SDK 3.0

Any setting changed could be pre-viewed.

Fig. 2-7

The System Configuration tab includes the following folders:

- **View Registry button:** Any settings are changed in the WinCon embedded controller can be pre-viewed by using this function. It is just like the regedit function in the windows system that you are very familiar with (shown in below figure).

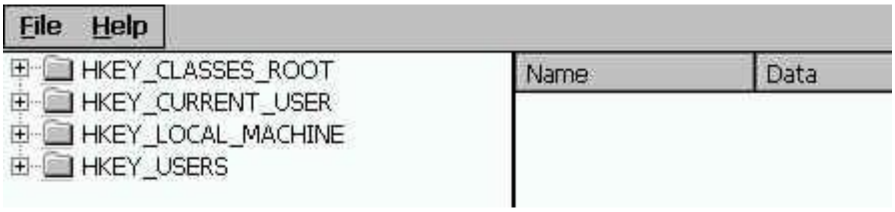


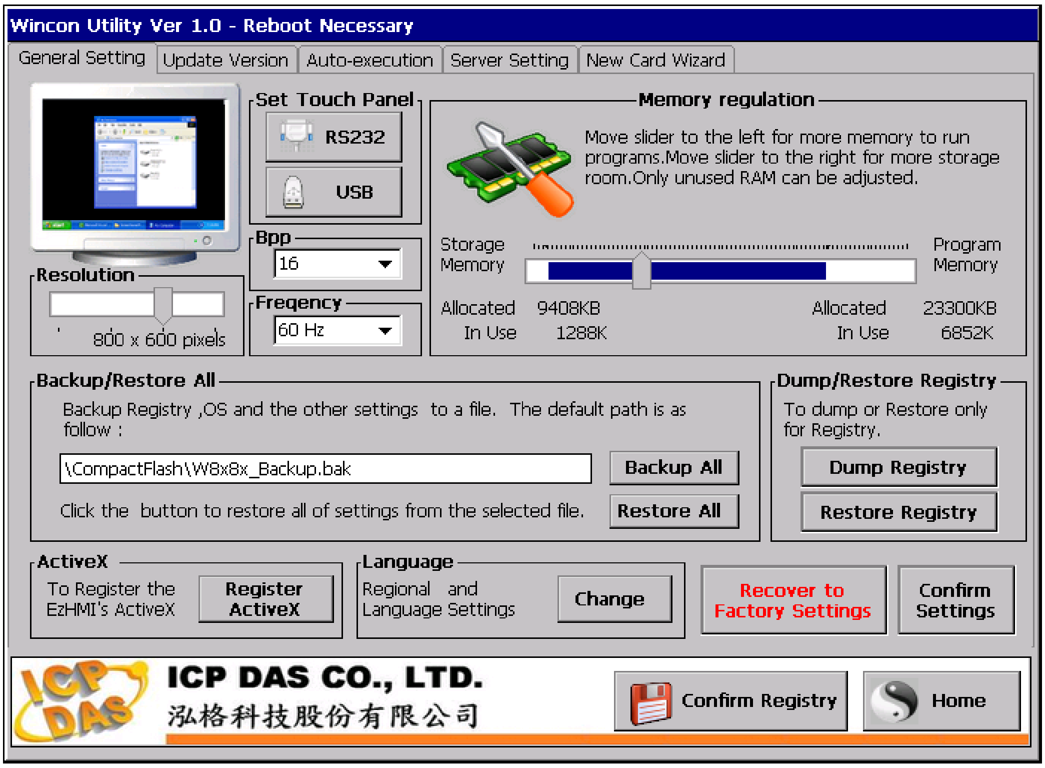
Fig. 2-8

- **Slot 1~7 box:** The Slot1~7 fields display the module names plugged in the W-8x8x.

- **Serial Number box:** This field displays the serial number of the W-8x8x.
- **MAC address:** The field displays the physical address of Ethernet port.
- **Flash Memory Size box:** This field displays the Flash memory size of the W-8x8x.
- **OS Version box:** This field displays the current operating system.
- **WinCon SDK Version box:** This field displays the current WinconSDK.DLL version.

2.2.2 General Setting Tab

It is very important to save the registry when you change any system information. Then you need to click the "***Confirm Registry***" button to renew the system configuration. If you do not save the current configuration into the registry, you will lose your information settings when you reboot the Wincon-8x8x.



Resolution: 800 x 600 pixels

Bpp: 16

Freqency: 60 Hz

Move slider to the left for more memory to run programs.Move slider to the right for more storage room.Only unused RAM can be adjusted.

Storage Memory: Allocated 9408KB, In Use 1288K

Program Memory: Allocated 23300KB, In Use 6852K

Backup Registry ,OS and the other settings to a file. The default path is as follow :

\CompactFlash\W8x8x_Backup.bak

Click the button to restore all of settings from the selected file.

To dump or Restore only for Registry.

To Register the EzHMI's ActiveX

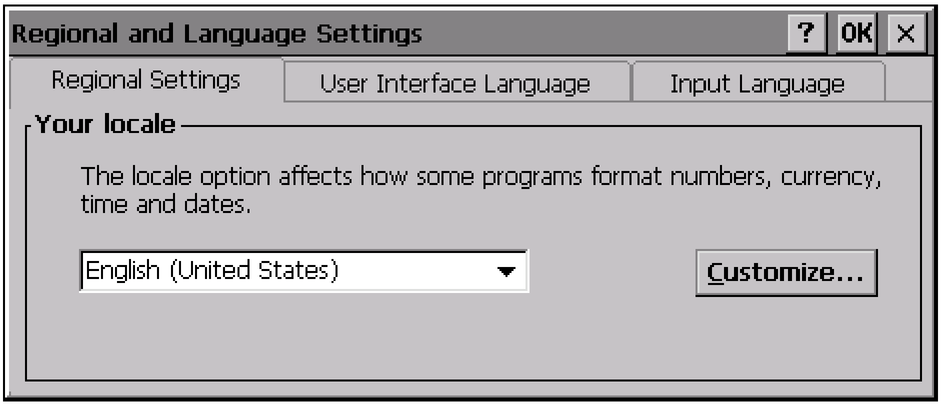
Regional and Language Settings

Fig. 2-9

The General Setting tab includes the following folders:

- **Change the VGA resolution box:** You can setting the VGA Resolution to 320x240,640x480,800x600 or 1024x768, and 2,4,8,16 bits color (Bpp),the monitor reflash Frequency for normal TFT LCD setting is 60 Hz.

- **Interface type of Touch Panel:** If You want to use W-8x8x with touch panel, You have to select which interface the touch panel has. And now, the driver of supposed touch panel only for APLEX.
- **Memory regulation:** You can move the slider to the left for more memory to run programs or move the slider to the right for more storage room. Only unused RAM can be adjusted.
- **Backup/Restore All:** This function provides to backup Registry、OS ,and the other settings to a file. You can set the file's path on the path edit. User can also restore all of settings from a file by Restore All button.
- **Dump/Restore Registry:** To dump and restore only for registry.
- **ActiveX:** If the user will use the EzProg-I ,you can register the ActiveX object by the button.
- **Recover to Factory Setting button:** It will take several seconds to clear your registry settings back to **Factory Setting** and Wright to **non-volatile internal memory**.
- **Language :** You can change regional and interface language setting here.



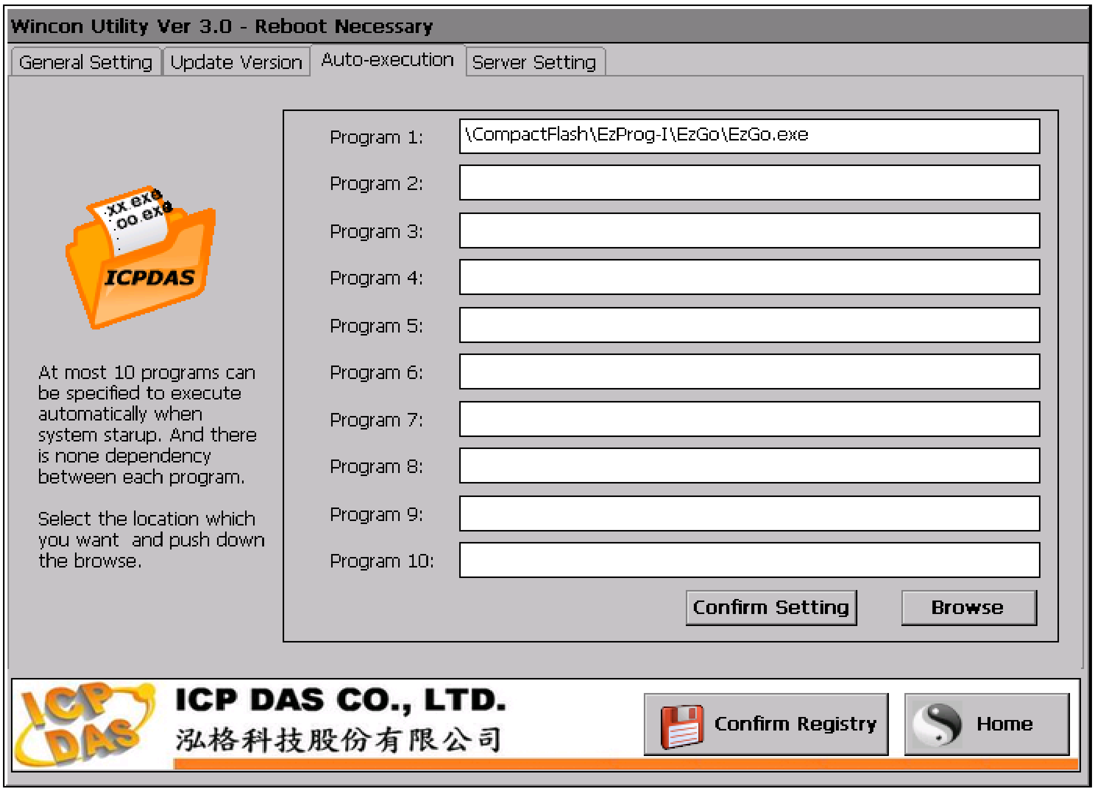
Your locale

The locale option affects how some programs format numbers, currency, time and dates.

English (United States)

2.2.3 Auto-execute Tab

The Auto-execute tab, provides ten execute files, which can be run after the WinCE system has been launched on the W-8x8x system. You can set ten execute files through the Browse button on the tab for WinCon Utility, as shown in the below figure. Note that they are executed in order of program 1, program 2, ...



Wincon Utility Ver 3.0 - Reboot Necessary

Program 1: \CompactFlash\EzProg-I\EzGo\EzGo.exe

At most 10 programs can be specified to execute automatically when system starup. And there is none dependency between each program.

Select the location which you want and push down the browse.

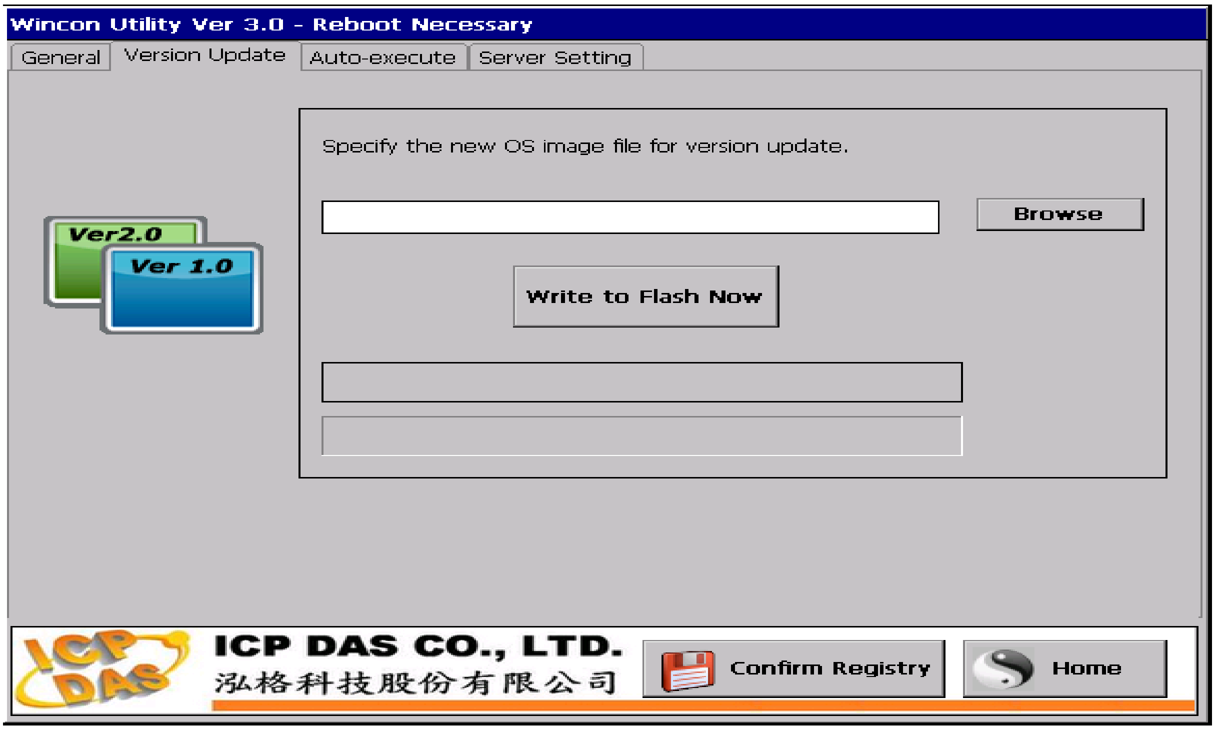
Fig. 2-10

The tab includes the following folders:

- **Program 1~10 boxes:** These files allow one to configure the auto-execute files for W-8x8x for when it is started up. You can choose the execute file and file directory path by means of the Browse button.
- **Confirm Setting button:** If you have changed the settings for the Program 1 ~ 10 field contents, you must then click the Confirm Setting button before closing the WinCon Utility window.

2.2.4 Version Update Tab

The Version Update tab provides the function to be able to update newer versions of the operating system. Users can download the OS image file from the web site: http://www.icpdas.com. You can choose the new OS image file name and directory path with the Browse button. Click the "***Write to flash now***" button to update the current OS version. It will take ten or more minutes to update your OS to Flash memory, and then reboot your system. Before Update OS Version, back up the registry if You hope to have the same settings.

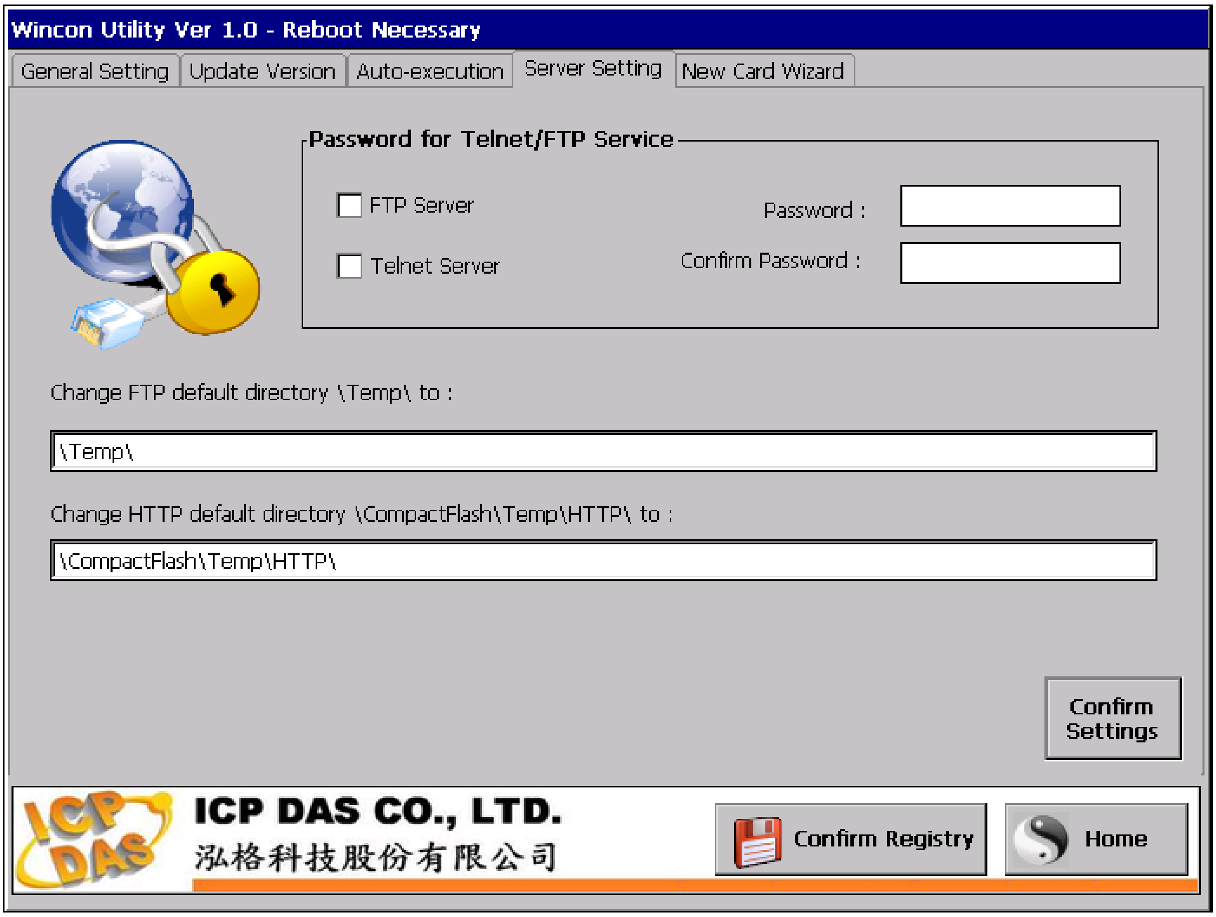


Specify the new OS image file for version update.

Fig. 2-11

2.2.5 Server Setting Tab

This tab provides functions to set the password for Telnet/FTP service and to setup the HTTP/FTP directory path.



Password for Telnet/FTP Service

□ FTP Server　Password :

□ Telnet Server　Confirm Password :

Change FTP default directory \Temp\ to :

\Temp\

Change HTTP default directory \CompactFlash\Temp\HTTP\ to :

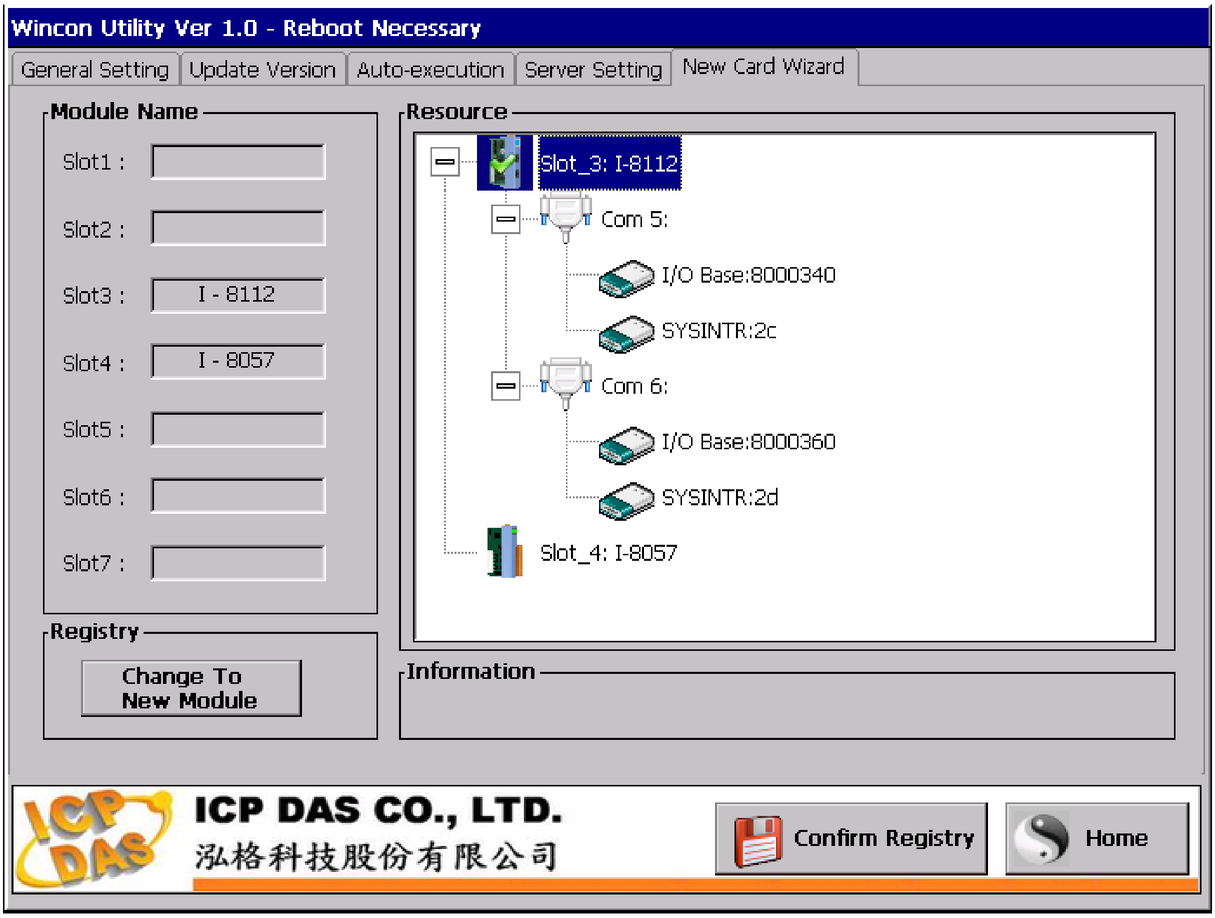
\CompactFlash\Temp\HTTP\

Fig. 2-12

- Default FTP/Telnet account: **root**
- **Password for Telnet/FTP Service:** You can set password protection for FTP or Telnet server here.
- **Change FTP default directory to box:** Enter a FTP default directory path and click confirm settings button to setup the defined path to the ftp server.
- **Change HTTP default directory to box:** Enter a HTTP default directory path and click on the confirm settings button to setup the defined path for the web server.

2.2.6 New Card Wizard Tab

This page shows not only which module in W-8x8x but also which communication module registered. The primary purpose of this tab is to register the communication module that user want to use. But W-8x8x only support one communication module now. If user put on more than one communication module, the registry button will be disable.



Module Name

Slot1 :

Slot2 :

Slot3 : I - 8112

Slot4 : I - 8057

Slot5 :

Slot6 :

Slot7 :

Resource

```
Slot_3: I-8112
    Com 5:
        I/O Base:8000340
        SYSINTR:2c
    Com 6:
        I/O Base:8000360
        SYSINTR:2d
Slot_4: I-8057
```

Registry

Information

Fig. 2-13

- **Module Name :** User can see all kinds of modules on W-8x8x.
- **Resource :**User can see what the communication module in use is, and what the com port number is.
- **Registry :** If the W-8x8x only has one communication module, this button will be enable. And this button will help user to register communication module.
- **Information :** If the W-8x8x has more than one communication module, here will appear the alarm message.

3. Architecture under WinCE.NET

The WinconSDK.DLL is the dynamic link library (DLL) files that are designed for applications running on the W-8x8x main controller unit and its modules with Windows CE.NET. The user can apply them to develop their own applications with many development tools, such as embedded Visual C++. In order to assist you, there are many demo programs provided for the user when using EVC++. Based on the demo programs, users can easily understand how to use the functions and develop their own applications within a short period of time.

The relationships among the WinconSDK.DLL and user's application are depicted as follows:

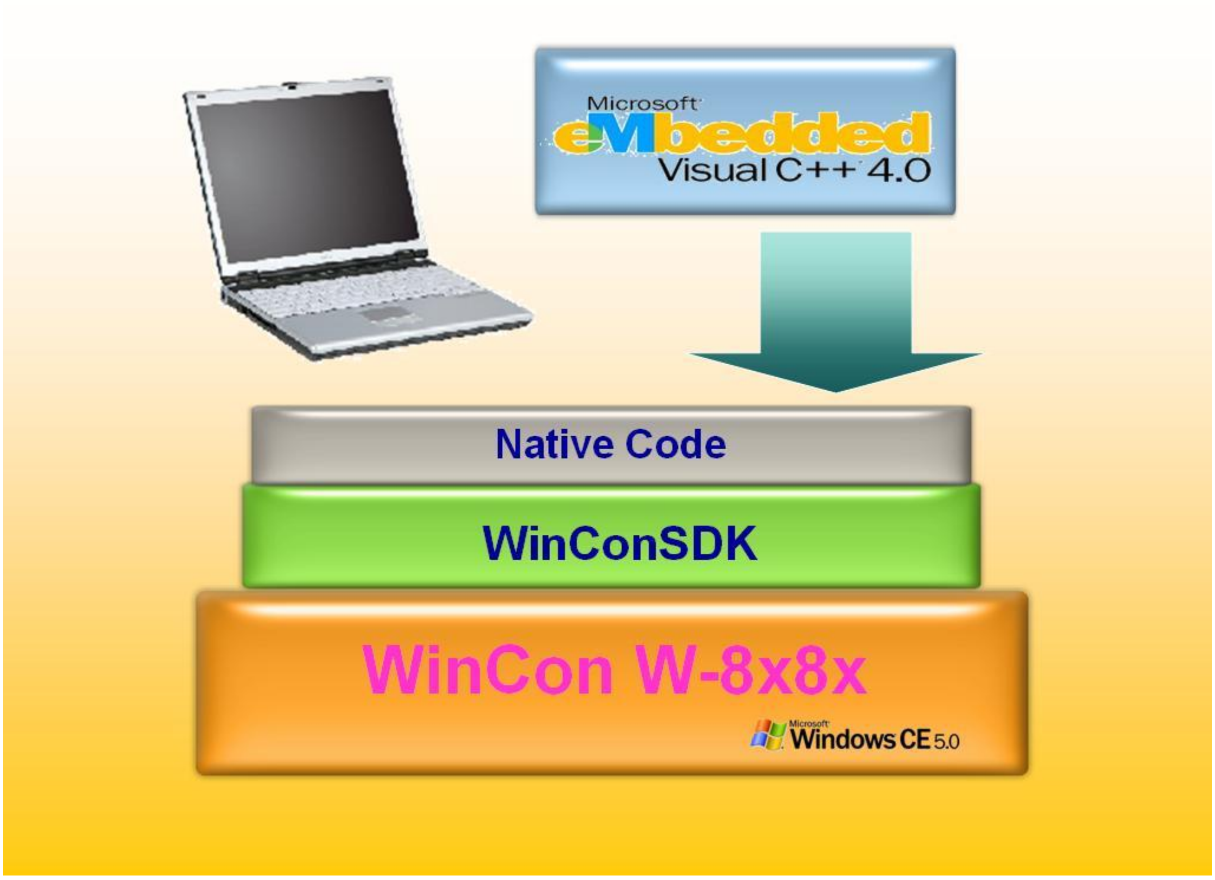


Fig. 3-1

Functions for the W-8x8x Main Controller Unit;

1. Get Module ID functions;
2. Analog Input/Output functions;
3. Digital Input/Output functions;
4. Motion card application functions;
5. Encoder card application functions;
6. Support for eMbedded Visual C^{++} Language;

The DLL functions in **Wincon_SDK.DLL** are designed for WinCE.NET and can be utilized by eVC++. The main characteristics of the **Wincon_SDK.DLL** that have been developed the application in the eVC++ developer environments, are very simple to use just like the applications in ActiveX(OCX) controls. Users can easily find the method they need for their application from the descriptions in the following section and in the demo programs developed by ICP DAS.

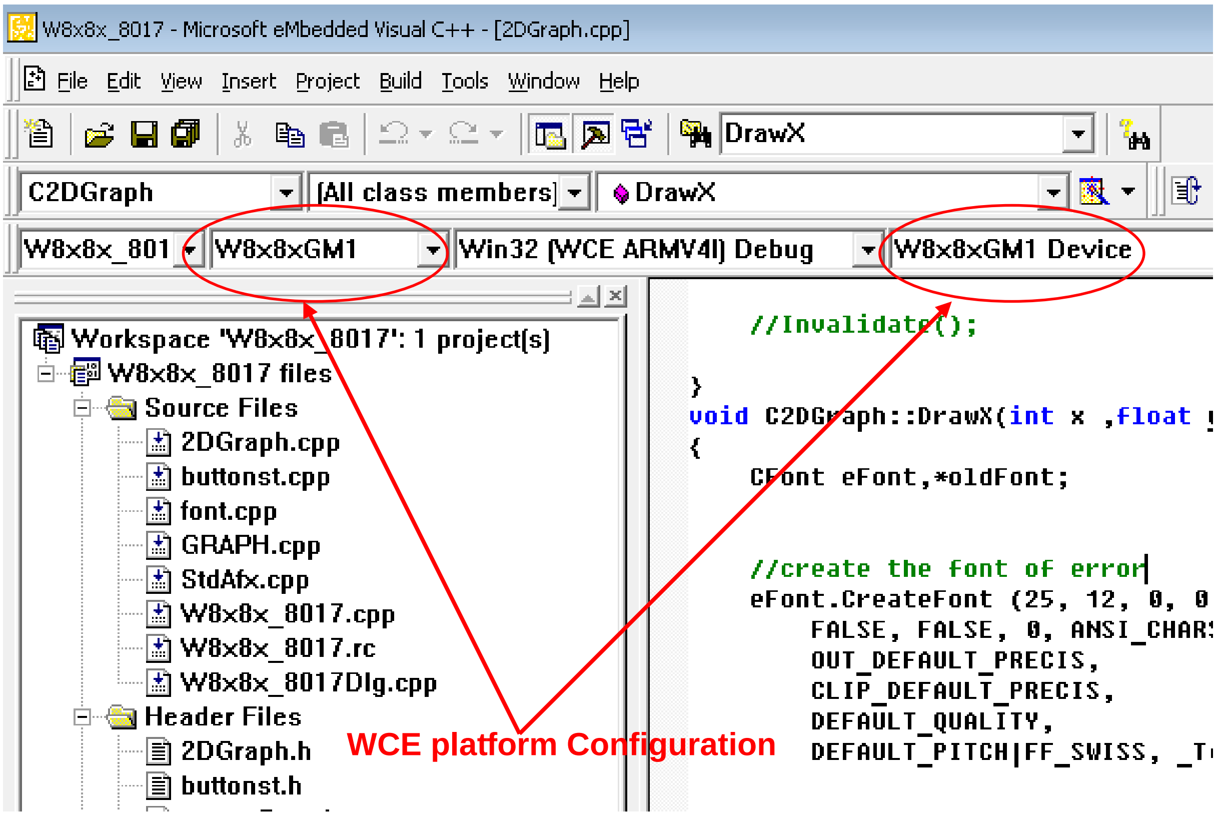
4. Installation W-8x8x SDK for EVC on PC

“W-8x8x SDK for EVC ” consists of the following major items.

- WinCon SDK DLL & library files for EVC
- WinCon SDK include files for EVC

Step-by-Step Installations of WinconSDK

1. Please insert the CD-ROM into your CD-ROM driver.
2. Then run “W8X8XGM1_YYYYMMDD.msi” in the folder of CD-ROM: \SDK\.
3. After successfully the software will be installed:
   - WinCon SDK library & DLL files are located at **C:\Program Files\Windows CE Tools\wce500\W8x8xGM1\Lib\ARMV4**
   - WinCon SDK include files for EVC are located at **C:\Program Files\Windows CE Tools\wce500\W8x8xGM1\Include\Armv4**
4. **W8x8xGM1** is the hardware platform name for W-8X8X .



DrawX

C2DGraph | [All class members] | DrawX

W8x8x_801 | W8x8xGM1 | Win32 (WCE ARMV4I) Debug | W8x8xGM1 Device

```
    //Invalidate();

}
void C2DGraph::DrawX(int x ,float
{
    CFont eFont,*oldFont;

    //create the font of error
    eFont.CreateFont (25, 12, 0, 0
        FALSE, FALSE, 0, ANSI_CHAR
        OUT_DEFAULT_PRECIS,
        CLIP_DEFAULT_PRECIS,
        DEFAULT_QUALITY,
        DEFAULT_PITCH|FF_SWISS, _T
```

WCE platform Configuration

5. Start The First Application Program

5.1 Applications in eMbedded Visual C^{++} (MFC)

When users want to create a new program, they have to use these files from the EVC^{++} environment toolkit, as shown in the following:

1. \WINCON\INC\WinconSDK.h → include all .h files
2. \WINCON\INC\WinCon.h → functions for D/I, D/O
3. \WINCON\INC\i8017h.h → functions for I-8017H Module
4. \WINCON\INC\i8024.h → functions for I-8024 Module
5. \WINCON\INC\i8092.h → functions for I-8092 Module
6. \WINCON\INC\i8094.h → functions for I-8094 Module
7. \WINCON\LIB\WinconSDK.lib → import library of WinconSDK.DLL

The application procedures for how to create a program are listed below. More detailed information for each step will be described in the following section.

1. **Create a Forms-Based Application;**
2. **Specify the Locations for Demo Files;**
3. **Configure Compiler Options;**
4. **Configure Link Options**
5. **Design an application program;**
6. **Build the application program;**
7. **Execute the application program on the W-8x8x platform;**
8. **Example list for the reference of user program design.**

5.1.1 Create a Forms-Based Application

A form is the dialog box with control items that lets users access and possibly change data. You may want to develop an application in which the user chooses from a selection of forms. Commonly, a forms-based application allows the user to choose "**New**" on the "**File**" menu to access forms. A dialog-based application, which does provide menus, is also considered a forms-based application.

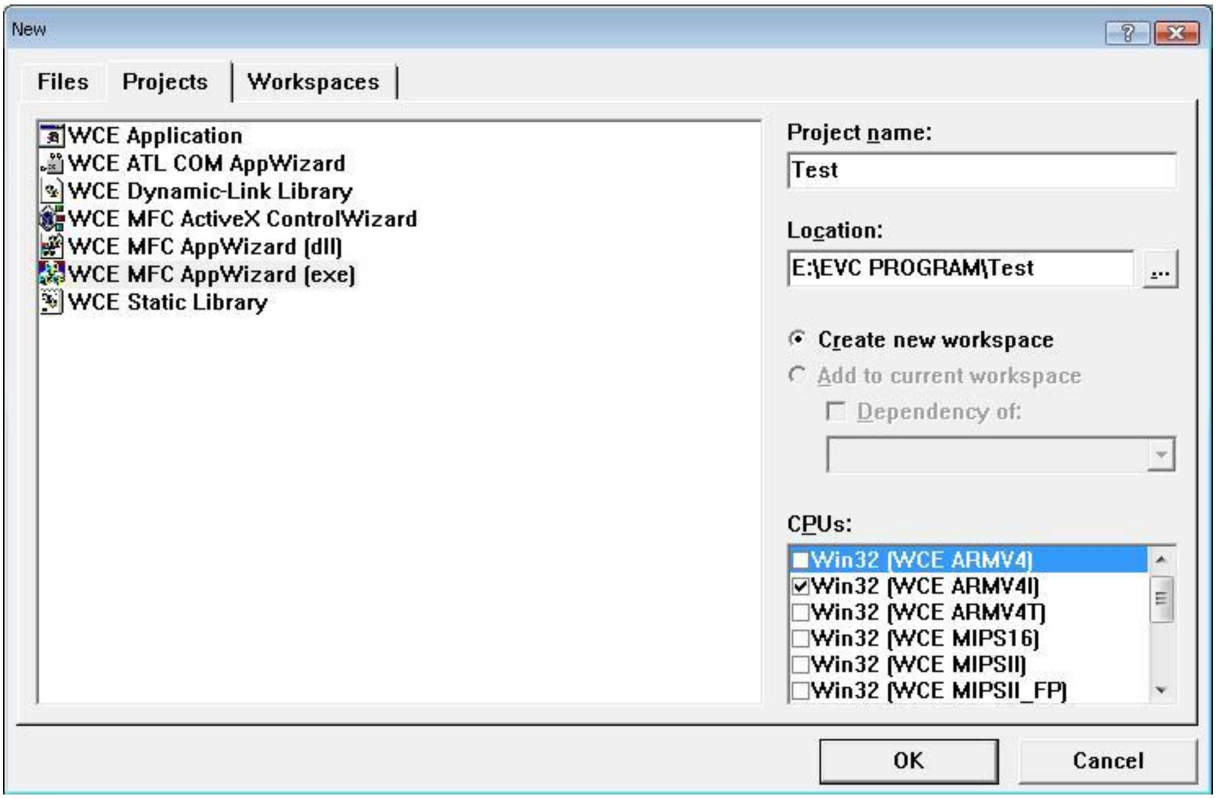
Users can apply the **WCE MFC AppWizard (exe)** to create a forms-based application. When creating a new form, they have the choice to create a single document interface or a dialog-based application.

- A single document interface (SDI), forms-based application allows only one instance of a particular form to run at a time. However, it is possible to run different forms at the same time from an SDI, forms-based application by selecting a new form from the “New” command in the File menu.
- A dialog-based application is a form based application by definition. However, if you choose this option, you will not be able to use the “New Form” command on the Insert menu to add additional forms to the application. A dialog-based application does not use the document/view architecture. So, when you need to add forms, you have to manage the creation and access methods for them.

The last step of the wizard lets you view and change settings for the classes created by the wizard. For a single or multiple document interface application, you should set “CformView” as the base class for the view class of your application. If your application is database supported, you can also select any class that is derived from “CformView”. A form is any window derived from “CformView” or from any class that comes from “CformView”.

To create a demo for forms-based application

1. Firstly, users need to open the EVC^{++} software. Then, in the “**File**” please click the “**New**” to open a new program.
2. In the “**Projects**“ tab, select “**WCE MFC AppWizard (exe)**” and set the **Project name: “Demo”** and the “**Location” ,**then select the location that you want . Then, select “**Win 32[WCE ARMV4I]**“ in the **CPUs** list box and other options if necessary. Finally click “**OK**” to start the wizard process, as shown in the following figure.



- WCE Application
- WCE ATL COM AppWizard
- WCE Dynamic-Link Library
- WCE MFC ActiveX ControlWizard
- WCE MFC AppWizard (dll)
- WCE MFC AppWizard (exe)
- WCE Static Library

Project name: Test

Location: E:\EVC PROGRAM\Test

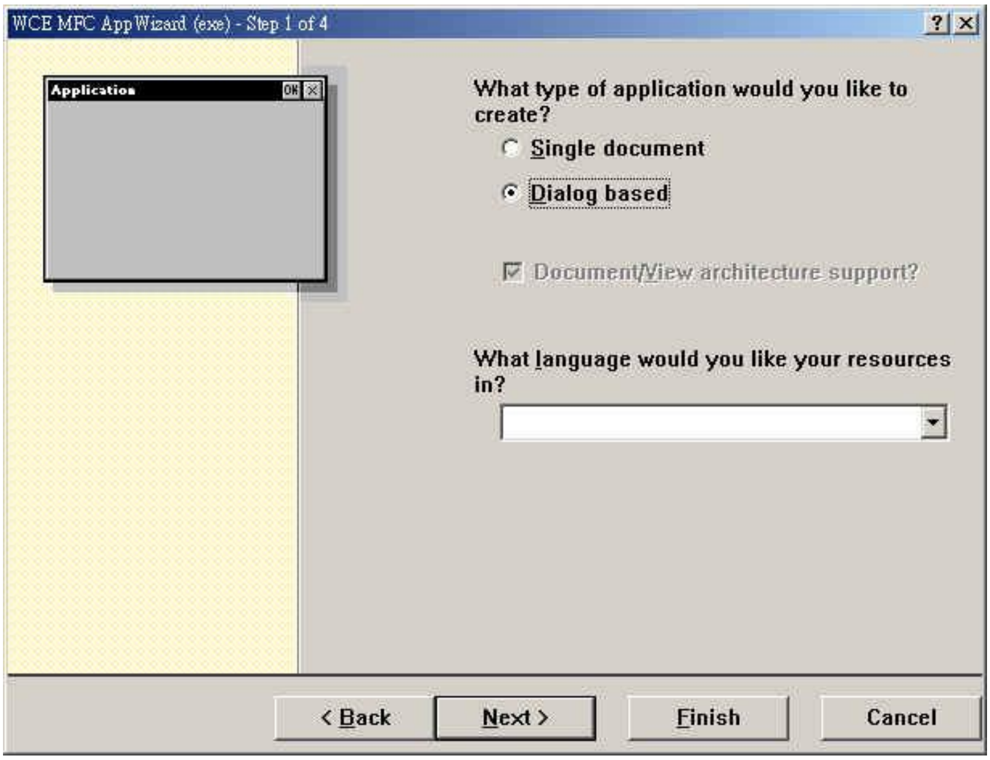
● Create new workspace
○ Add to current workspace
□ Dependency of:

CPUs:
□ Win32 (WCE ARMV4)
☑ Win32 (WCE ARMV4I)
□ Win32 (WCE ARMV4T)
□ Win32 (WCE MIPS16)
□ Win32 (WCE MIPSII)
□ Win32 (WCE MIPSII_FP)

Fig. 5-1-1

3. The configuration of WCE MFC AppWizard.

(3a) WCE MFC AppWizard: Step 1 of 4: Set “**Dialog based**” as shown in the following figure. Click “**Next**” to go to the following step.



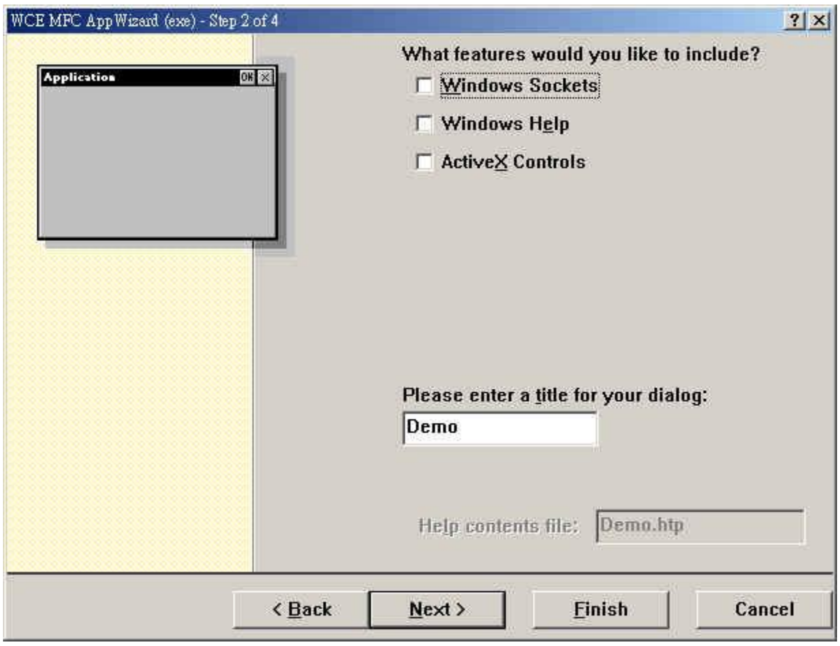
What type of application would you like to create?
○ Single document
● Dialog based

☑ Document/View architecture support?

What language would you like your resources in?

Fig. 5-1-2

(3b) WCE MFC AppWizard-Step 2 of 4: Please enter a title for your dialog in the defined area and click “Next” to go to the following step (see figure 5-1-3).



WCE MFC AppWizard (exe) - Step 2 of 4

What features would you like to include?

☐ Windows Sockets

☐ Windows Help

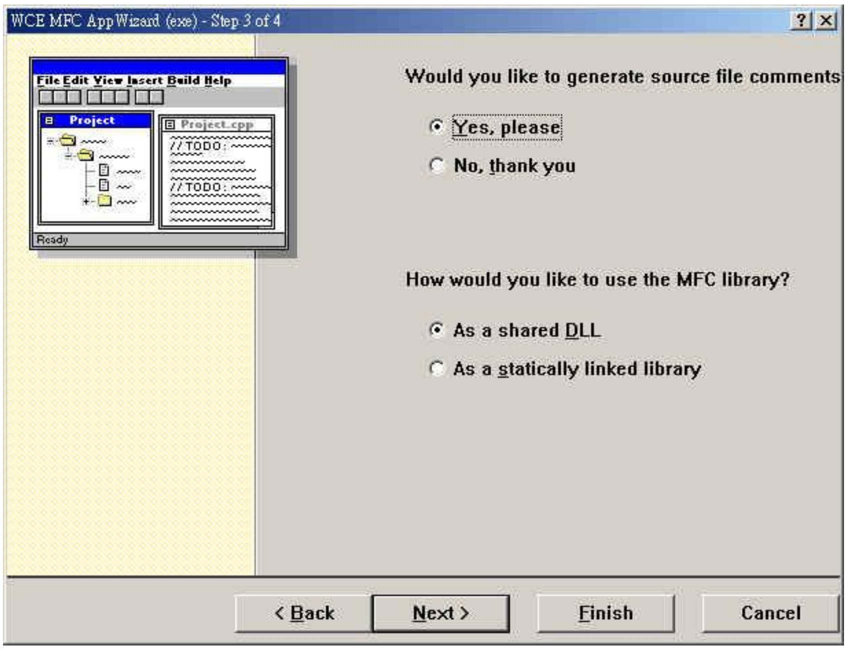
☐ ActiveX Controls

Please enter a title for your dialog: Demo

Help contents file: Demo.htp

Fig. 5-1-3

(3c) WCE MFC AppWizard-Step 3 of 4: In this step, please choose “Yes, please” to generate source file comments. And set the MFC library as a shared DLL or statically linked library. Then Click “Next” to go to the next step.



WCE MFC AppWizard (exe) - Step 3 of 4

Would you like to generate source file comments

● Yes, please

○ No, thank you

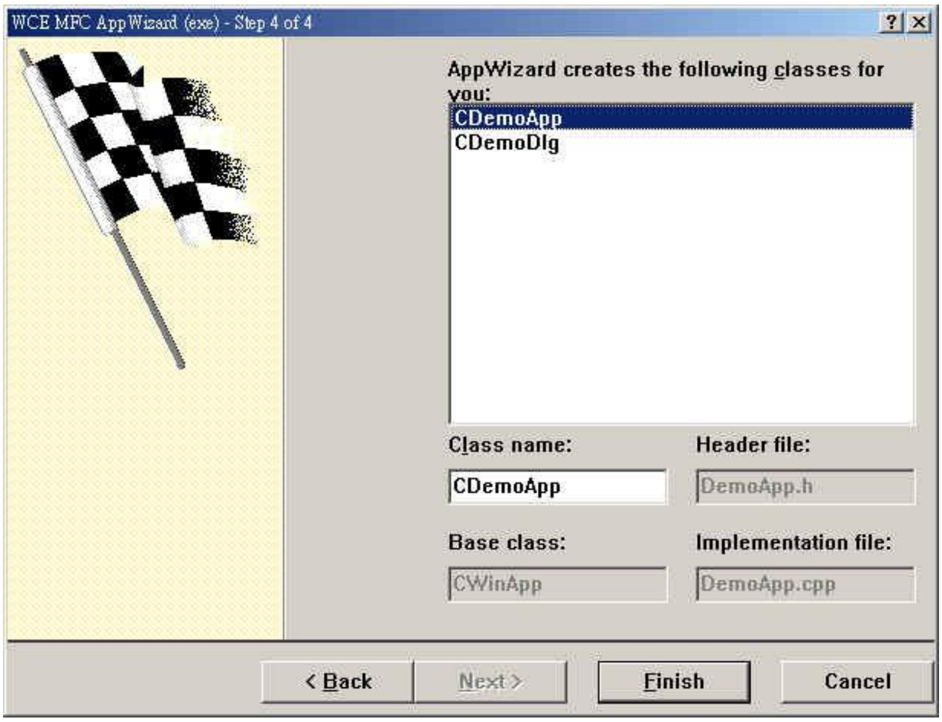
How would you like to use the MFC library?

● As a shared DLL

○ As a statically linked library

Fig. 5-1-4

(3d) WCE MFC AppWizard-Step 4 of 4: Click the “Finish” button to start the AppWizard to create the classes, which are shown in the figure below. (see figure 5-1-5).



WCE MFC AppWizard (exe) - Step 4 of 4

AppWizard creates the following classes for you:

CDemoApp
CDemoDlg

Class name: CDemoApp

Header file: DemoApp.h

Base class: CWinApp

Implementation file: DemoApp.cpp

Fig. 5-1-5

5.1.2 Specify Locations for Demo Files

The **Directories** settings in **Options** for eMbedded Visual C^{++} specify search paths for files in your projects. These include lists of paths for the following types of files:

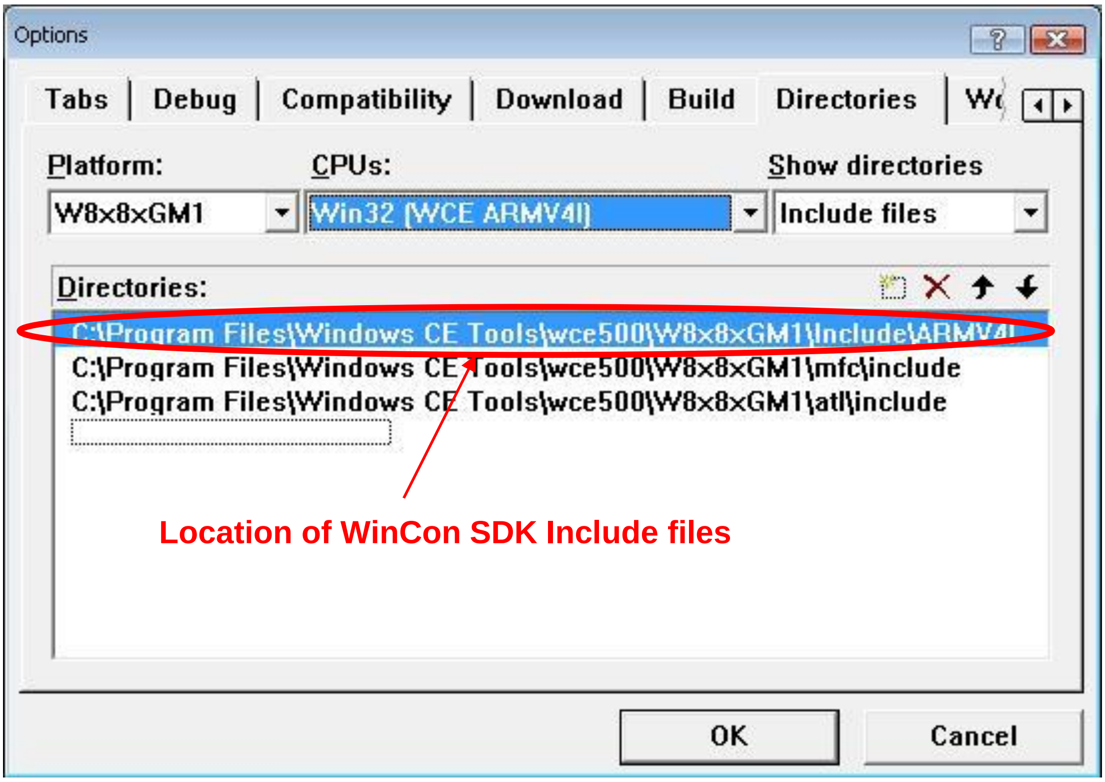
File type	Path contents
Executable files	Specify locations for the build utilities, such as the compilers (CL, CLARM, CLMIPS, and so on), NMAKE, LINK, and BSCMAKE.
Include files	Specifies where the compiler should look for “include” files (for example #include <stdio.h>).
Library files	Specify where the linker should look for libraries to resolve external references.
Source files	Specify where the debugger should look for default source files, such as the Microsoft Foundation Class Library and the Microsoft run-time library.

Path information is stored in directory settings and it allows you to configure them. You can change both the paths listed in these settings and the order in which eMbedded Visual C^{++} searches for them.

The path specifications have separated the setting for the different devices and the different platforms (such as WCE emulator) supported by your installation of eMbedded Visual C^{++}.

To specify default folders for demo project files

1. Open the "**Options**" dialog in the "**Tools**" menu.
2. In the "**Directories**" tab, select the "**W8x8xGM1**"(for W-8X8X) on the "**Platform**" item, "**Win32 [WCE ARMV4I]**" in the "**CPUs**" item and also the "**Show directories**" in the "**include files**" or "**Library files**" item.
3. Since all the WinCon specific include and library files have been allocated in default directories, users don't need to modify the contents of default directories of include and library files .
4. Fig 5-1-6 display the default location of include files.



Platform: W8x8xGM1

CPUs: Win32 (WCE ARMV4I)

Show directories: Include files

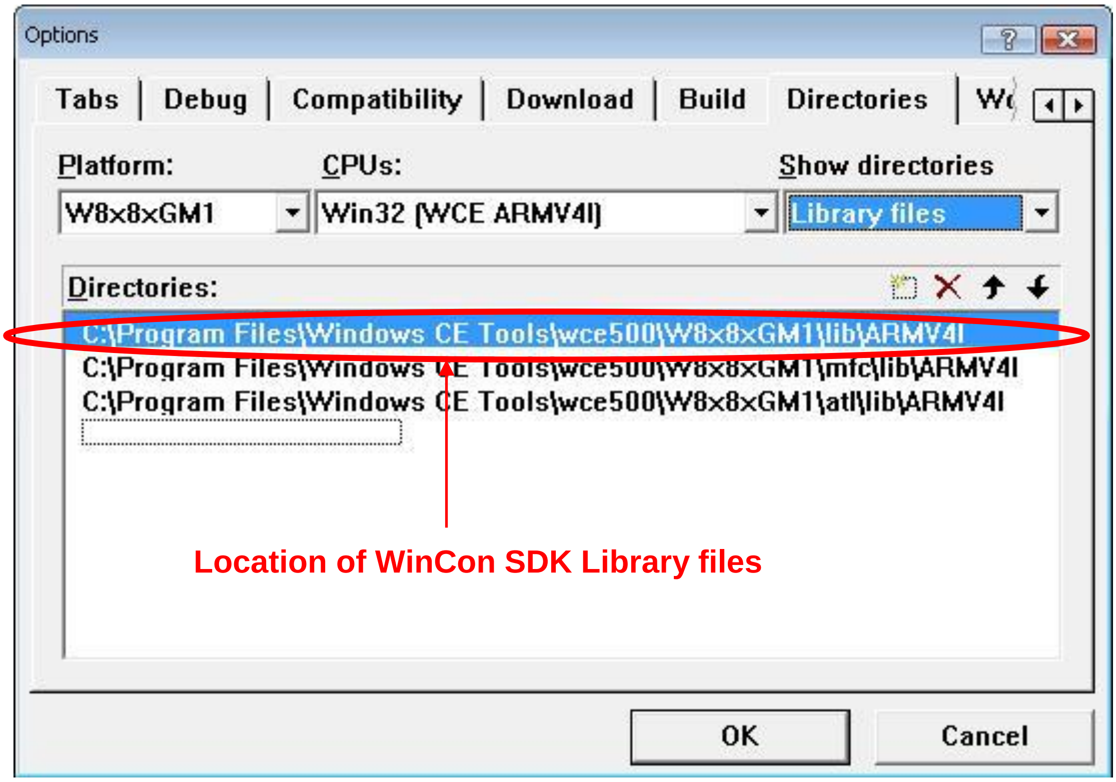
Directories:

C:\Program Files\Windows CE Tools\wce500\W8x8xGM1\Include\ARMV4I
C:\Program Files\Windows CE Tools\wce500\W8x8xGM1\mfc\include
C:\Program Files\Windows CE Tools\wce500\W8x8xGM1\atl\include

Location of WinCon SDK Include files

Fig. 5-1-6

5. Fig 5-1-7 displays the default location of Library files.



Platform: W8x8xGM1

CPUs: Win32 (WCE ARMV4I)

Show directories: Library files

Directories:

C:\Program Files\Windows CE Tools\wce500\W8x8xGM1\lib\ARMV4I
C:\Program Files\Windows CE Tools\wce500\W8x8xGM1\mfc\lib\ARMV4I
C:\Program Files\Windows CE Tools\wce500\W8x8xGM1\atl\lib\ARMV4I

Location of WinCon SDK Library files

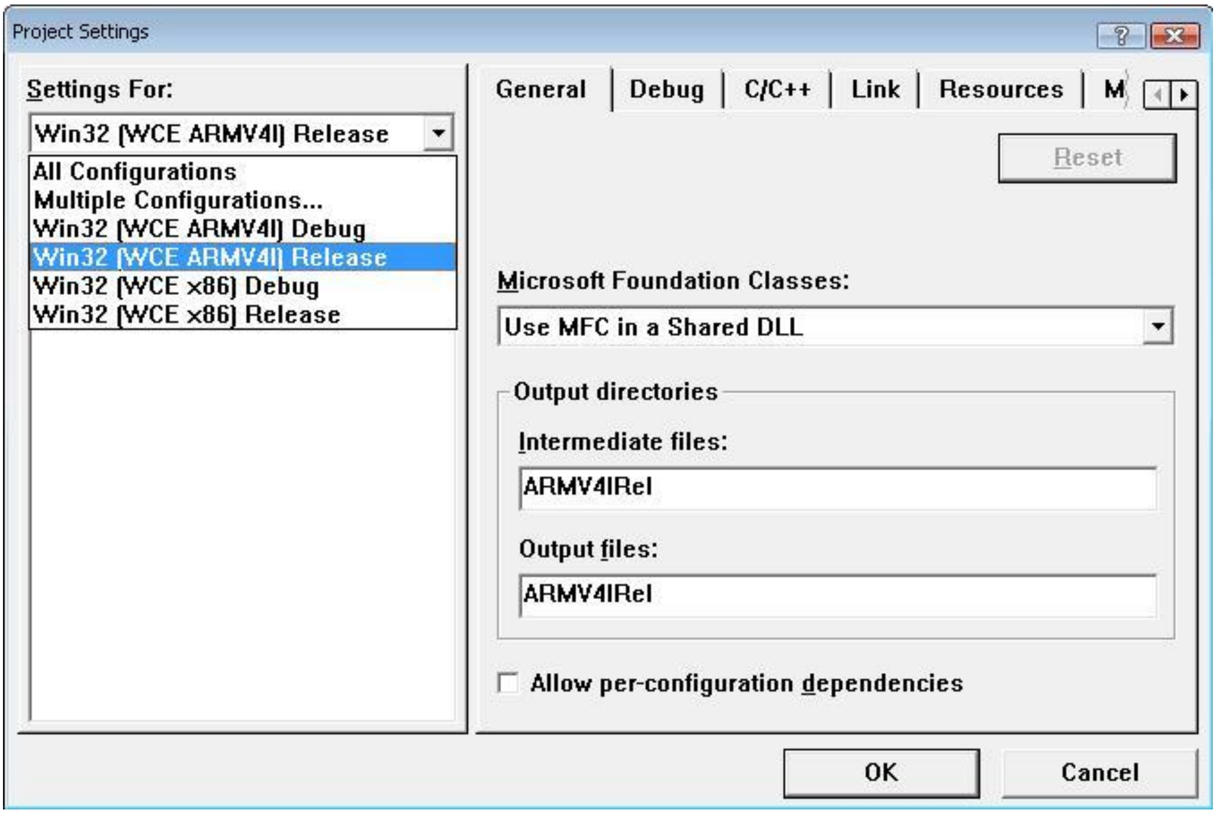
Fig. 5-1-7

✍ **Note**: If you want to remove a folder, select it and then press DELETE.

5.1.3 Configure Compiler Options

To set the compiler options in the development environment, please follow these steps:

1. In the “**Project**“ menu, choose “**Settings**” to open the “**Project Setting**” dialog.
2. Select the configuration “**Win32 [WCE ARMV4I] Release**” in “**Settings for**” item as shown in following figure.



Settings For: Win32 (WCE ARMV4I) Release

All Configurations
Multiple Configurations...
Win32 (WCE ARMV4I) Debug
Win32 (WCE ARMV4I) Release
Win32 (WCE x86) Debug
Win32 (WCE x86) Release

Microsoft Foundation Classes: Use MFC in a Shared DLL

Output directories

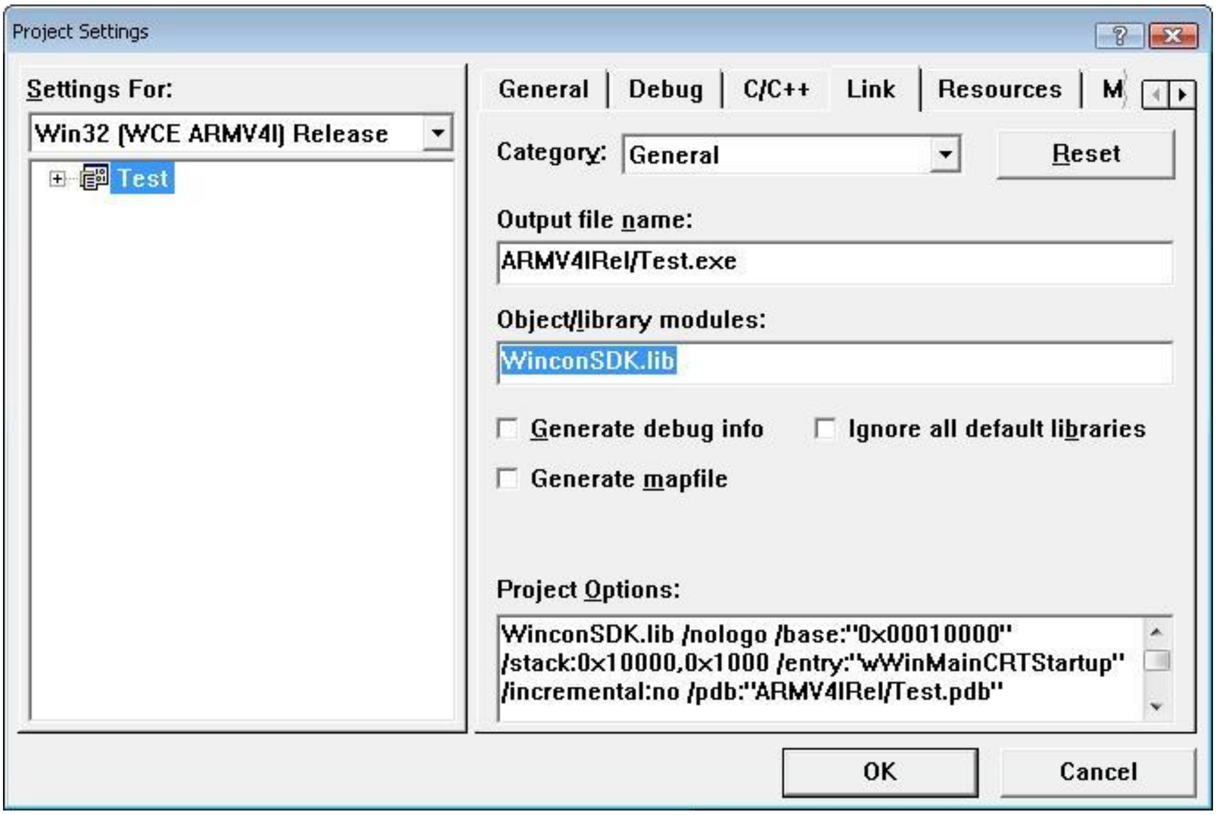
Intermediate files: ARMV4IRel

Output files: ARMV4IRel

□ Allow per-configuration dependencies

Fig. 5-1-8

3. In the Project Settings dialog, select the "**Link**" tab and set the library file name as "**WinconSDK.lib**" in "**Object/library modules**" item.



Settings For: Win32 (WCE ARMV4I) Release

Test

Category: General

Output file name: ARMV4IRel/Test.exe

Object/library modules: WinconSDK.lib

□ Generate debug info □ Ignore all default libraries

□ Generate mapfile

Project Options:

```
WinconSDK.lib /nologo /base:"0x00010000"
/stack:0x10000,0x1000 /entry:"wWinMainCRTStartup"
/incremental:no /pdb:"ARMV4IRel/Test.pdb"
```

Fig. 5-1-9

5.1.4 Configure link options

After finishing program, You have to download the execute file to W-8x8x by CF card ,FTP ,etc….Now, You have another choice by online download. You only have to link the PC with W-8x8x by telnet, after linking the execute file will exit at the path "\My Device\" in W-8x8x. You must do something for the link.

STEP 1: select Configure Platform Manager in Tools

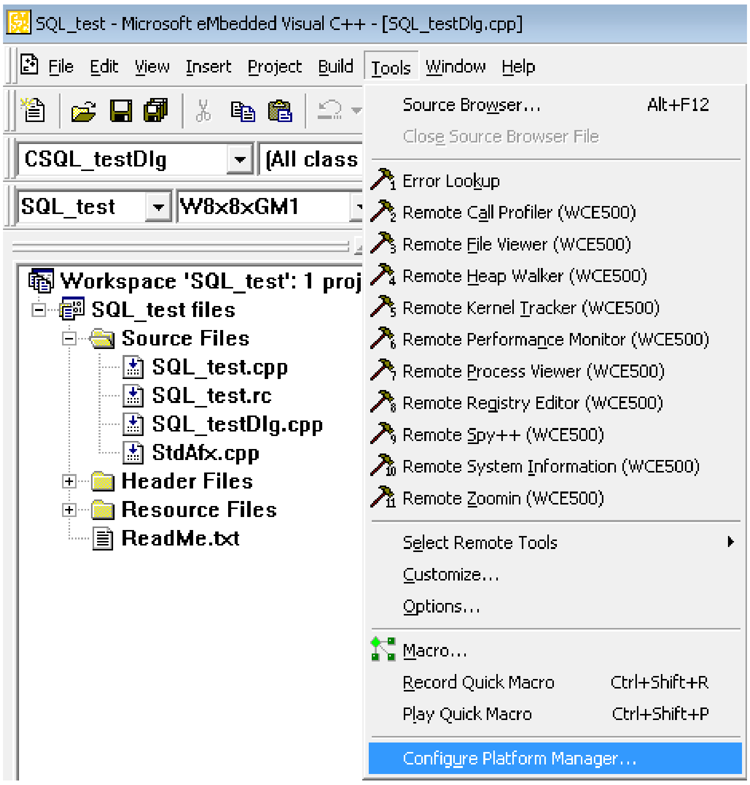

SQL_test - Microsoft eMbedded Visual C++ - [SQL_testDlg.cpp]
File Edit View Insert Project Build Tools Window Help
CSQL_testDlg
[All class
SQL_test
W8x8xGM1
Workspace 'SQL_test': 1 proj
SQL_test files
Source Files
SQL_test.cpp
SQL_test.rc
SQL_testDlg.cpp
StdAfx.cpp
Header Files
Resource Files
ReadMe.txt
Source Browser... Alt+F12
Close Source Browser File
Error Lookup
Remote Call Profiler (WCE500)
Remote File Viewer (WCE500)
Remote Heap Walker (WCE500)
Remote Kernel Tracker (WCE500)
Remote Performance Monitor (WCE500)
Remote Process Viewer (WCE500)
Remote Registry Editor (WCE500)
Remote Spy++ (WCE500)
Remote System Information (WCE500)
Remote Zoomin (WCE500)
Select Remote Tools
Customize...
Options...
Macro...
Record Quick Macro Ctrl+Shift+R
Play Quick Macro Ctrl+Shift+P
Configure Platform Manager...

Fig. 5-1-10

STEP 2: select the W-8x8xGM1 Device and Click the Properties button

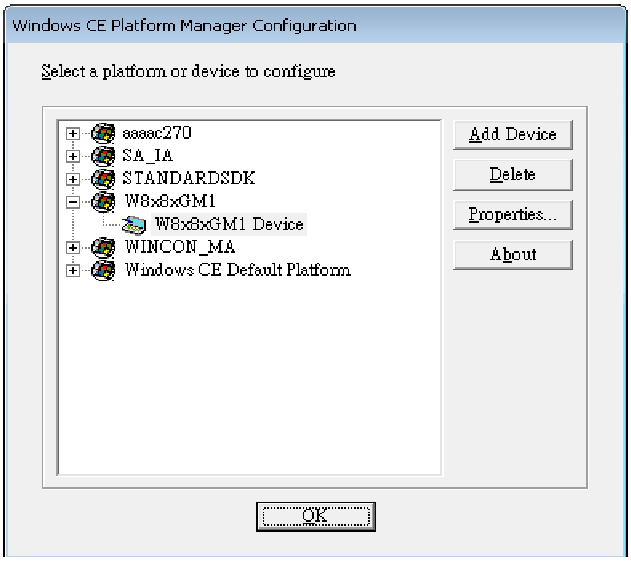
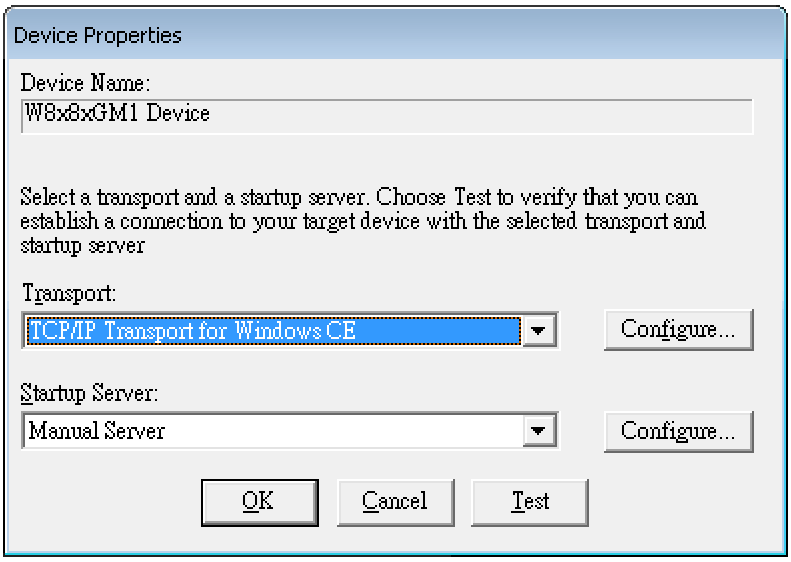

Windows CE Platform Manager Configuration
Select a platform or device to configure
aaaac270
SA_IA
STANDARDSDK
W8x8xGM1
W8x8xGM1 Device
WINCON_MA
Windows CE Default Platform
Add Device
Delete
Properties...
About
OK

Fig. 5-1-11



Device Properties

Device Name: W8x8xGM1 Device

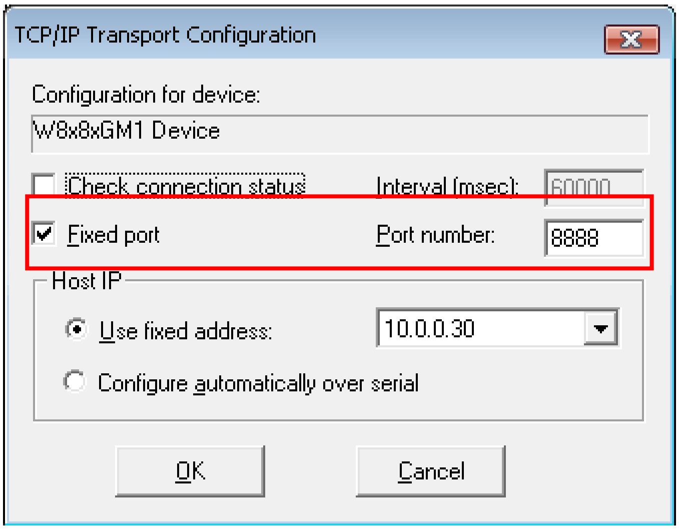
Select a transport and a startup server. Choose Test to verify that you can establish a connection to your target device with the selected transport and startup server

Transport: TCP/IP Transport for Windows CE

Startup Server: Manual Server

Fig. 5-1-12

Push down the Configure button of Transport appears follow page. You can choose Fixed port and enter the Port number to avoid the command line different every time.



TCP/IP Transport Configuration

Configuration for device: W8x8xGM1 Device

□ Check connection status Interval (msec): 60000

☑ Fixed port Port number: 8888

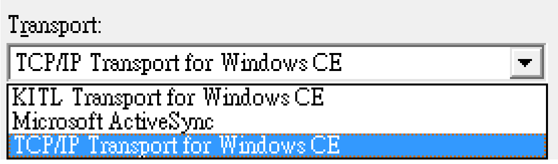
Host IP

● Use fixed address: 10.0.0.30

○ Configure automatically over serial

Fig. 5-1-12-1

STEP 3:select TCP/IP Transport for Windows CE in Transport selections

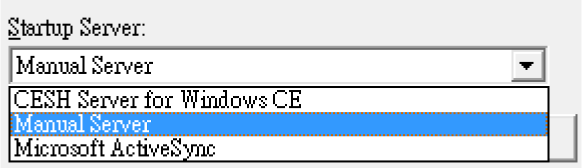


Transport: TCP/IP Transport for Windows CE

KITL Transport for Windows CE
Microsoft ActiveSync
TCP/IP Transport for Windows CE

Fig. 5-1-13

STEP 4:select Manual Server in Startup Server selections



Startup Server: Manual Server

Fig. 5-1-14

STEP 5:select GO in Start Debug of Build or push down the button “F5”

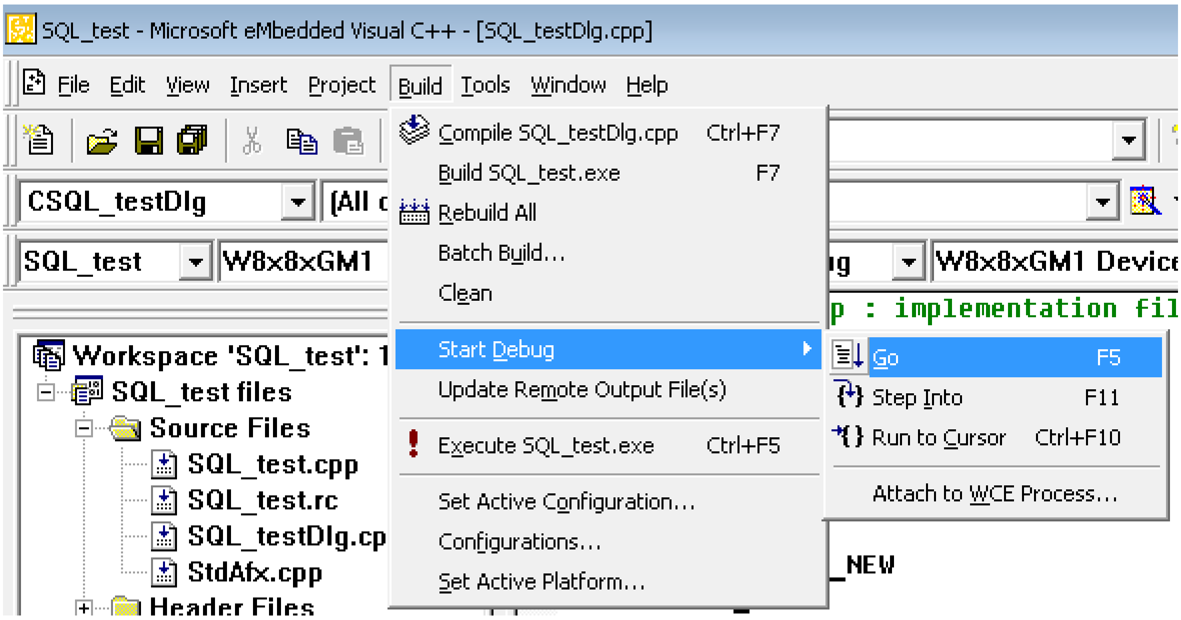
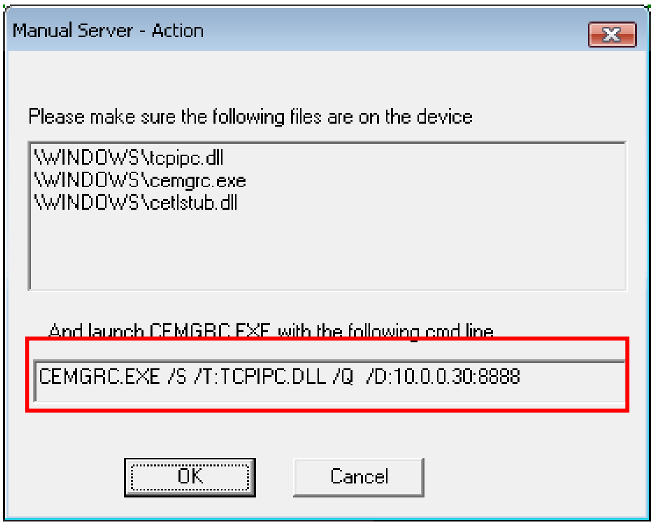


Fig. 5-1-15

STEP 6:remember the command line but don’t push down “OK” button until step8 finish


Manual Server - Action

Please make sure the following files are on the device

```
\WINDOWS\tcpipc.dll
\WINDOWS\cemgrc.exe
\WINDOWS\cetlstub.dll
```

And launch CEMGRC.EXE with the following cmd line

```
CEMGRC.EXE /S /T:TCPIPC.DLL /Q  /D:10.0.0.30:8888
```

Fig. 5-1-16

SETP 7:select “Run” in start menu of W-8x8x

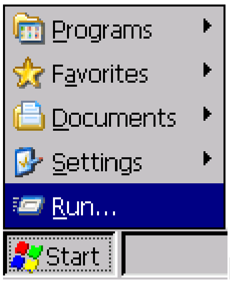
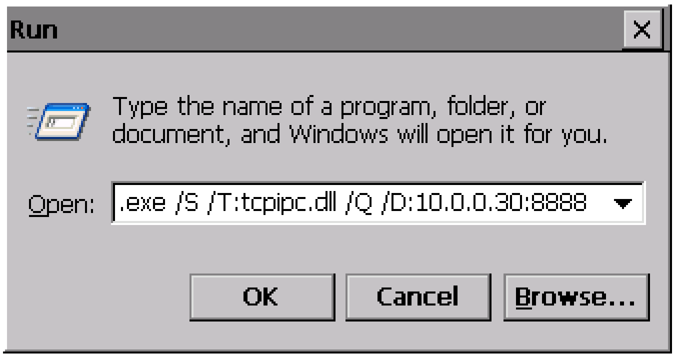


Fig. 5-1-17

STEP 8:key in the command line on the open path and push down "OK" button then back to step6 push down the "OK" button


Run
Type the name of a program, folder, or document, and Windows will open it for you.
Open:
OK
Cancel
Browse...

```
.exe /S /T:tcpipc.dll /Q /D:10.0.0.30:8888
```

Fig. 5-1-18

Finish about 8 steps, You also finish download the execute file to W-8x8x. But You have to pay attention to two things. First, if You didn't move the downloaded file to CF card it will disappear. Another is that if Your project is debug mode the execute file can not run after You move it to CF card and reboot. Only the execute file built on release mode can run after reboot.

5.1.5 Design an Application Program

After you finish the initial creation of a project, you can build user interfaces. These interfaces can involve the first design and creation of dialog boxes, menus, toolbars, accelerators, and other visual and interactive elements. Then you can hook them up to code. Due to the difference in size and shapes of devices, the user interface elements must be tailored to the design requirements of each targeted device. In the following we will explore some methods on how to create control items in the user's program.

To create an Edit Box:

1. In the "**Workspace**"window, select the "**ResourceView**"tab. Click on the "**Dialog**" folder to expand it and double-click the "**IDD_DEMO_DIALOG**" in order to open the dialog form as shown in below figure.

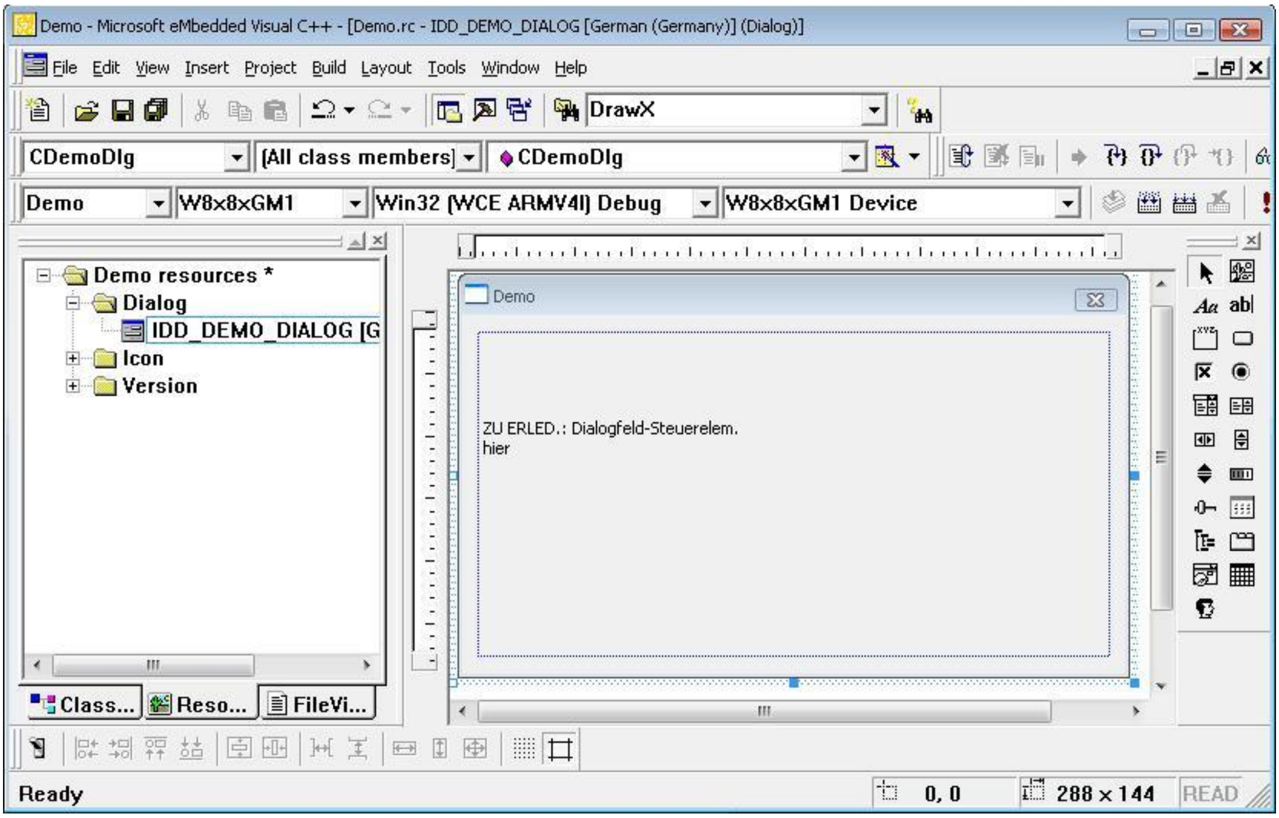
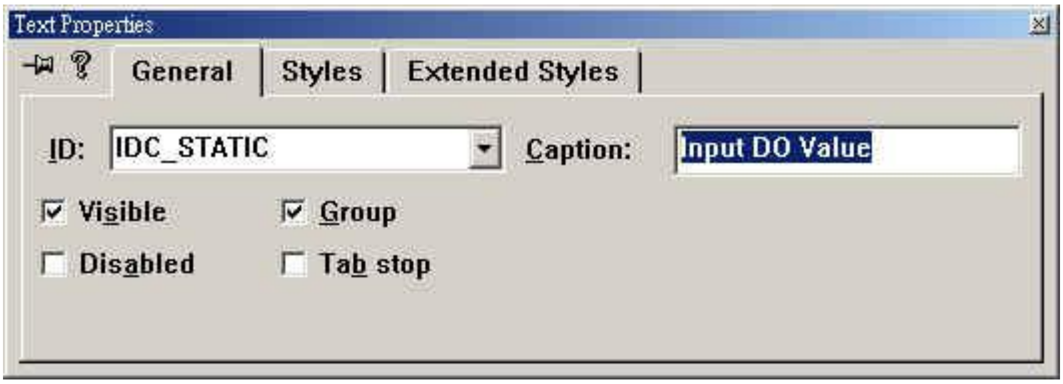


Fig. 5-1-19

2. By clicking the icon Aa to create a static text object.

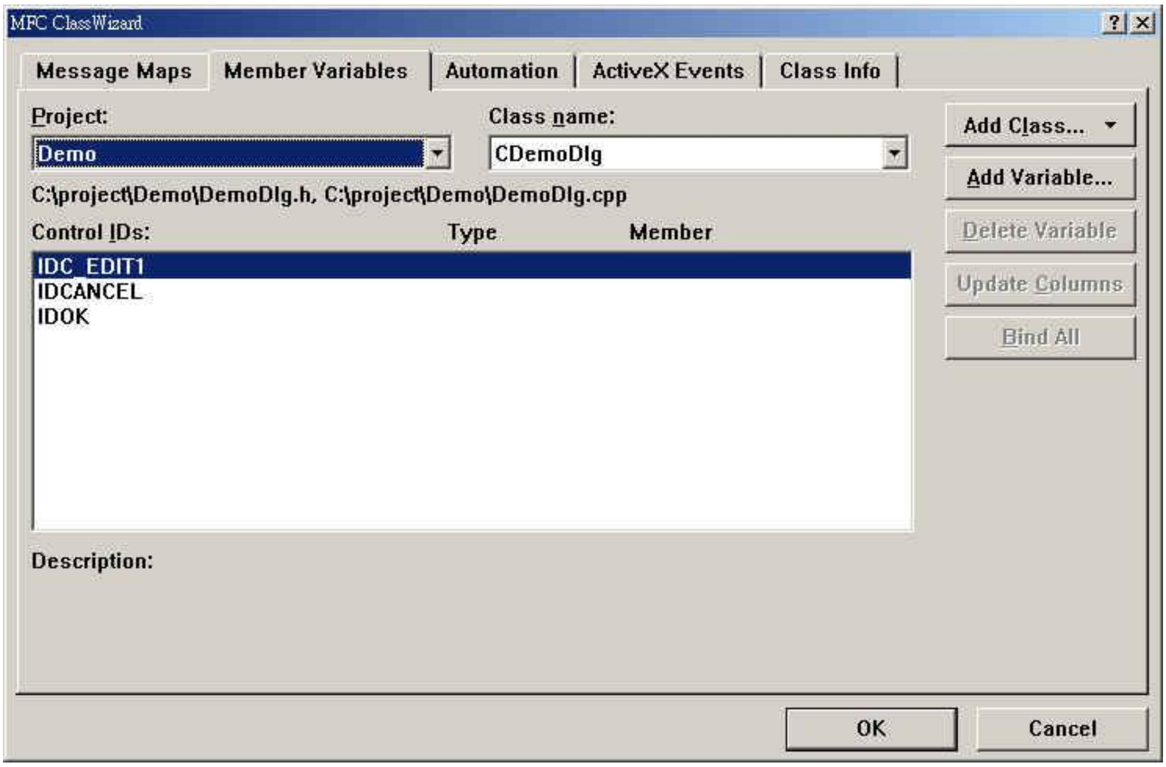
3. Right-click on the Static text object and select "**Properties**".
4. When the Text Properties dialog is shown, set "**Input DO Value**" in the "**Caption**" field.



ID: IDC_STATIC
Caption: Input DO Value
☑ Visible ☑ Group
□ Disabled □ Tab stop

Fig. 5-1-20

5. Click the icon ab| to create an edit box object.
6. Right-click on the edit box object and select "**Class Wizard**".
7. When the MFC Class-Wizard dialog is shown, select the "**Member Variables**" tab.



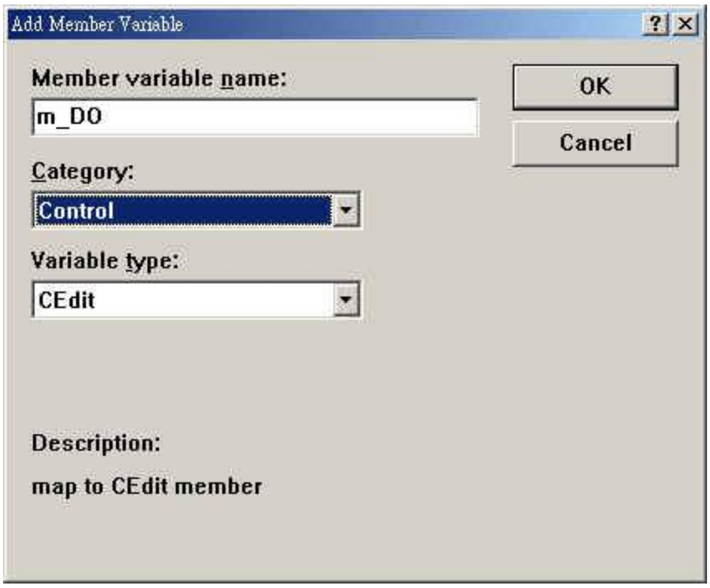
Project: Demo
Class name: CDemoDlg
C:\project\Demo\DemoDlg.h, C:\project\Demo\DemoDlg.cpp

Control IDs:	Type	Member
IDC_EDIT1		
IDCANCEL		
IDOK		

Description:

Fig. 5-1-21

8. Click the "**Add Variable**" button to add a new member variable.



Add Member Variable

Member variable name: m_DO

Category: Control

Variable type: CEdit

Description:

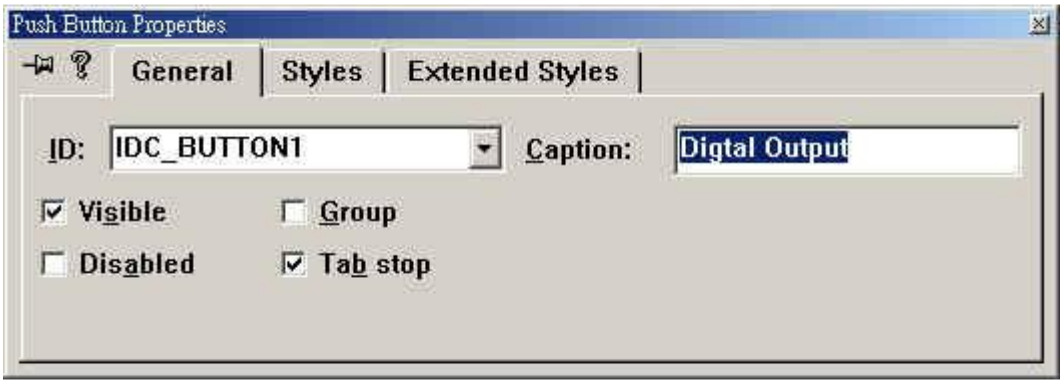
map to CEdit member

Fig. 5-1-22

9. Set “**m_DO**” in the “**Member variable name**“ field, and configure the “**Category:**“ dropdown-box as “**Control**”, and “**Variable type:**“ as “**CEdit**”.
10. Once you have completed the above list of steps, click the “**OK**“ button to save the Member variable and close the dialog.

To create a Button:

1. Create a new button object by clicking on the icon ▭ in the “**Con..**” window.
2. Right-click on the Button object and select “**Properties**”.
3. When the **Push Button Properties** dialog is shown, type “**Digital Output**” into the **Caption** field (see figure), and close the dialog.



Push Button Properties

ID: IDC_BUTTON1

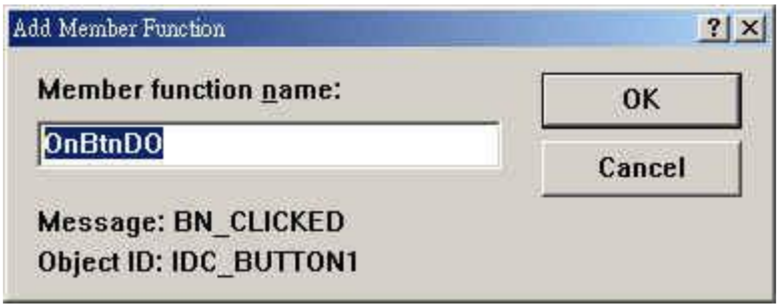
Caption: Digtal Output

☑ Visible ☐ Group

☐ Disabled ☑ Tab stop

Fig. 5-1-23

4. Double-click on the Push Button object so that you can edit the button’s event code. Then type “OnBtnDO” into the “**Member function name:”** field (see figure 5-1-15), and click the OK button to display the Editor window.

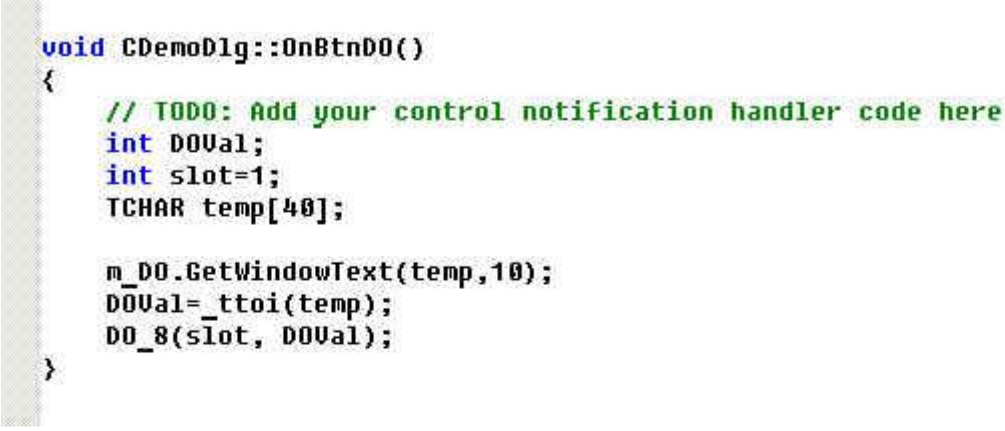


Member function name: OnBtnDO

Message: BN_CLICKED
Object ID: IDC_BUTTON1

Fig. 5-1-24

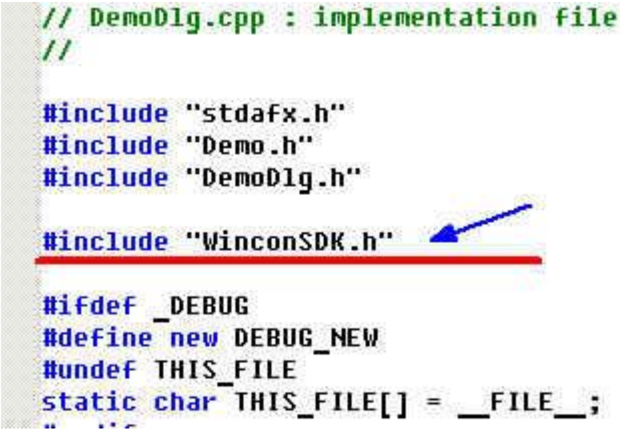
5. In the Editor window, type the code in as follows:



```
void CDemoDlg::OnBtnDO()
{
	// TODO: Add your control notification handler code here
	int DOVal;
	int slot=1;
	TCHAR temp[40];

	m_DO.GetWindowText(temp,10);
	DOVal=_ttoi(temp);
	DO_8(slot, DOVal);
}
```

Fig. 5-1-25

6. Press CTRL+Home to move your cursor to the top, and then insert the "**#include** **"WinconSDK.h"** " within the header area.



```
// DemoDlg.cpp : implementation file
//

#include "stdafx.h"
#include "Demo.h"
#include "DemoDlg.h"

#include "WinconSDK.h"

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

Fig. 5-1-26

5.1.6 Build the Application Program

Microsoft eMbedded Visual C++ provides two ways for building an application program. The first and most common method is to build the application program within the eMbedded Visual C^{++} development environment. The second method is to build the application program by using command-line tools within the MS-DOS prompt environment. The procedure for building an application involves a number of items namely the preprocessor, the compiler, and the linker. Their main functions are described as below:

- The preprocessor prepares source files for the compiler by translating macros, operators, and directives.
- The compiler creates an object file containing machine code, linker directives, sections, external references, and function/data names.
- The linker combines the code from the object files created by the compiler and from the statically-linked libraries, plus it resolves the name references, and also creates an executable file.

The Building Process

The following diagram shows the components in the building process using eMbedded Visual C++ and starting with the editor in which you create your source code.

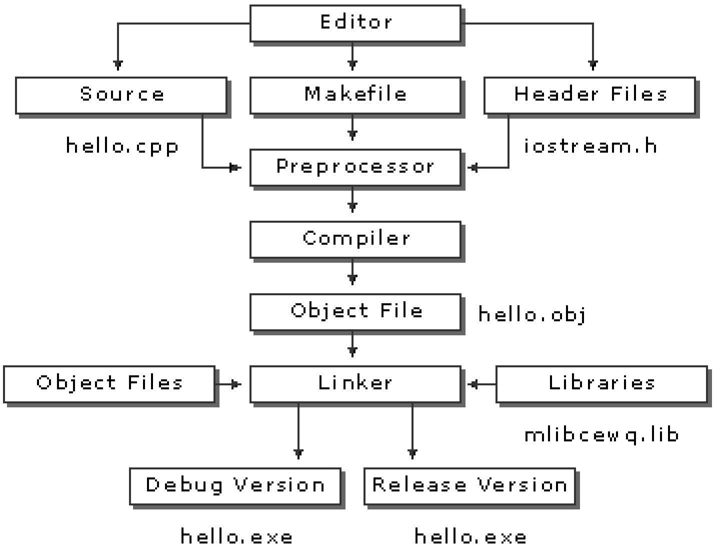
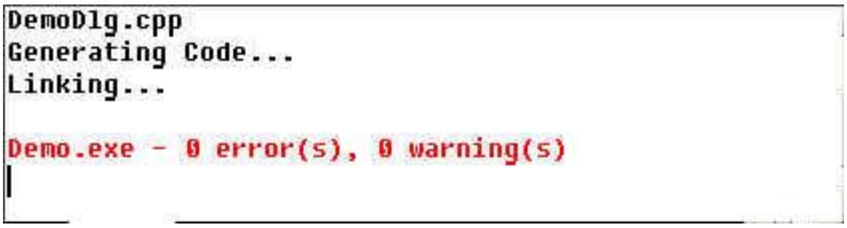


Fig. 5-1-27

If you build your program outside the IDE, you may use a makefile to invoke the command-line tools. Microsoft eMbedded Visual C^{++} provides the NMAKE utility for processing makefiles. If you build your program within the IDE, the eMbedded Visual C++ project system will use the project (.vcp) file to store the make information. The .vcp file is not compatible with NMAKE. However, if your program uses a makefile rather than a .vcp file, you can still build it in the development environment as an external project.

To Build the Demo Application

1. On the "**Build**" menu, choose "**Build Demo**".
2. If you have finished the above steps, you would obtain the following message in the Output window, which means the execution application has been built. Otherwise, you will get error messages. You would then need to debug the source code and rebuild the application.



```
DemoDlg.cpp
Generating Code...
Linking...

Demo.exe - 0 error(s), 0 warning(s)
```

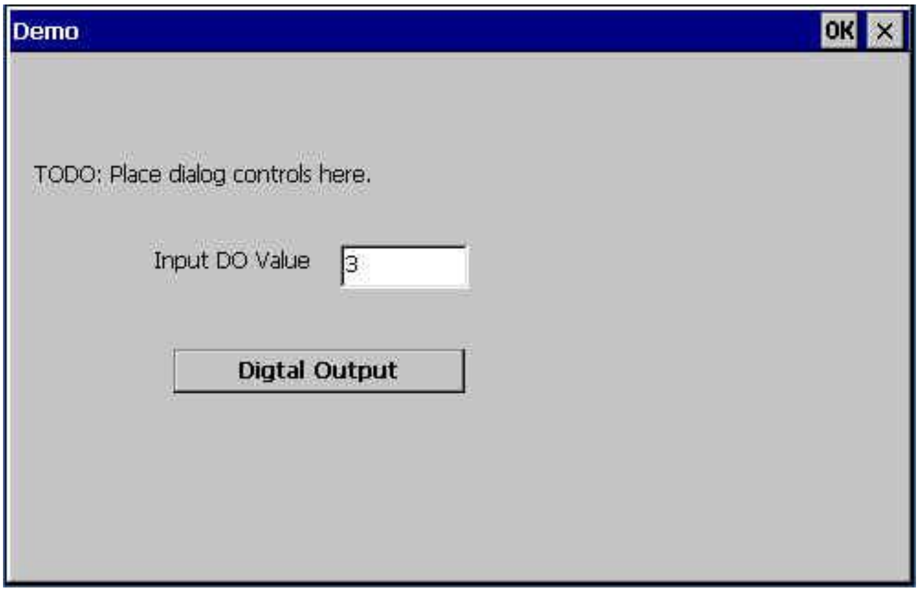
Fig. 5-1-28

3. If there no error messages are produced, you will find the Demo.exe file in C:\Project\Demo\ARMV4Rel\.

5.1.7 Execute the Application Program on the W-8x8x Platform

When you have finished the building process for the Demo.exe, you could apply the developed execute file (Demo.exe) into the W-8x8x controller unit. For example, you can use a portable memory stick (Like Pen Driver) to store the file and to copy it to W-8x8x. In a following section, we will develop a method for how to download the application to the main controller unit through the Ethernet (TCP/IP) or RS-232 series protocol. The main idea for running the user program within the W-8x8x target control device is to make the process simple just like in the following steps.

1. Copy the Demo.exe from C:\Project\Demo\ARMV4Rel\ subfolder to the W-8x8x.
2. Double-click on the Demo.exe file to run it on the W-8x8x, as shown in following figure.



TODO: Place dialog controls here.

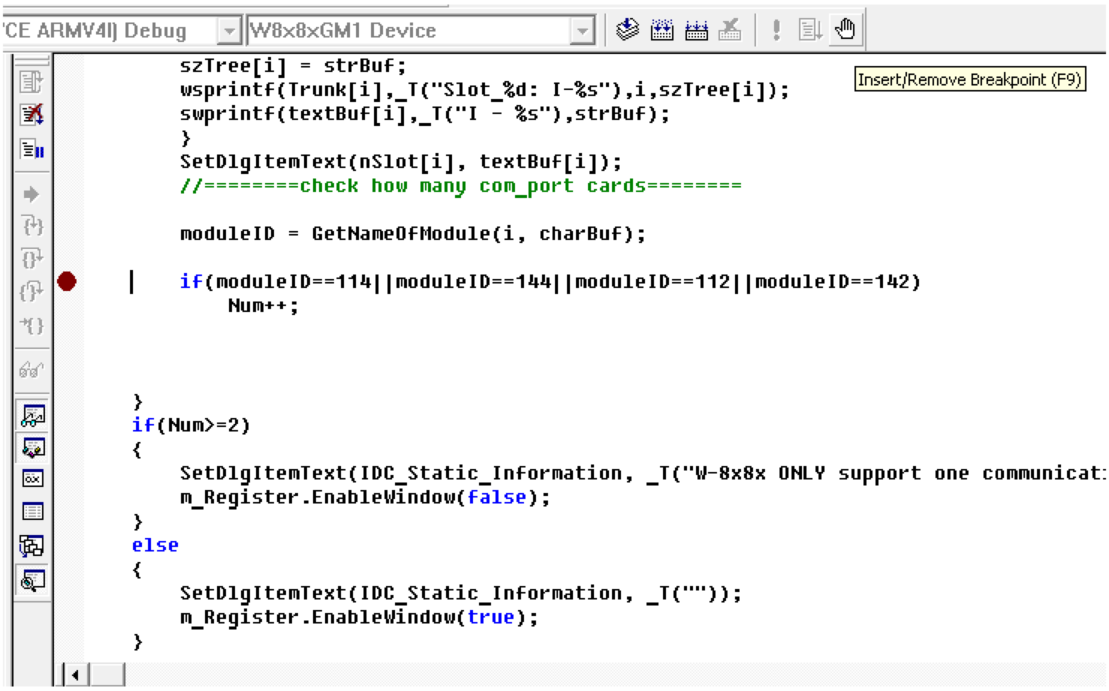
Input DO Value 3

Fig. 5-1-29

3. Type “3” into the “**Input DO Value**” field (above figure), and click the “**Digital Output**” button. Then, you can turn on the channel 0 and 1 of the Digital output Module plugged into the slot 1 of W-8x8x.

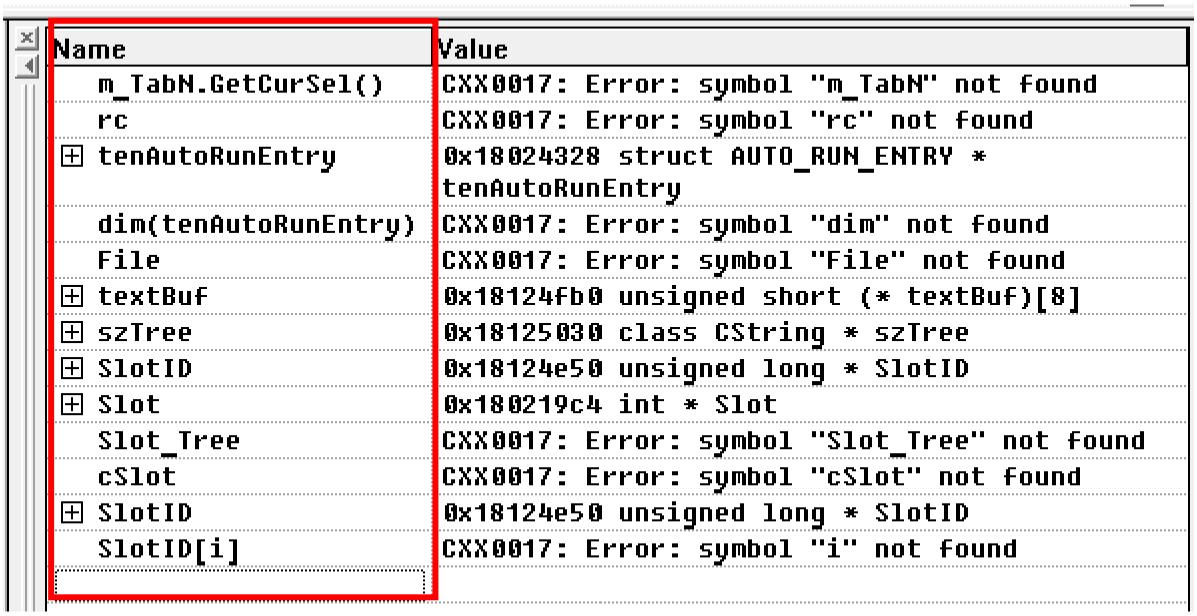
5.1.8 Online debug

When You want to debug online, push down “F9” or Click at any location that You want to debug. Then a red point will appear at that location, it means this program will stop there.



```
            szTree[i] = strBuf;
            wsprintf(Trunk[i],_T("Slot_%d: I-%s"),i,szTree[i]);
            swprintf(textBuf[i],_T("I - %s"),strBuf);
            }
            SetDlgItemText(nSlot[i], textBuf[i]);
            //========check how many com_port cards========

            moduleID = GetNameOfModule(i, charBuf);

            if(moduleID==114||moduleID==144||moduleID==112||moduleID==142)
                Num++;



        }
        if(Num>=2)
        {
            SetDlgItemText(IDC_Static_Information, _T("W-8x8x ONLY support one communicat
            m_Register.EnableWindow(false);
        }
        else
        {
            SetDlgItemText(IDC_Static_Information, _T(""));
            m_Register.EnableWindow(true);
        }
```

Besides, if You have many variables want to check, You can add these variables in Watch table.



Name	Value
m_TabN.GetCurSel()	CXX0017: Error: symbol "m_TabN" not found
rc	CXX0017: Error: symbol "rc" not found
⊞ tenAutoRunEntry	0x18024328 struct AUTO_RUN_ENTRY * tenAutoRunEntry
dim(tenAutoRunEntry)	CXX0017: Error: symbol "dim" not found
File	CXX0017: Error: symbol "File" not found
⊞ textBuf	0x18124fb0 unsigned short (* textBuf)[8]
⊞ szTree	0x18125030 class CString * szTree
⊞ SlotID	0x18124e50 unsigned long * SlotID
⊞ Slot	0x180219c4 int * Slot
Slot_Tree	CXX0017: Error: symbol "Slot_Tree" not found
cSlot	CXX0017: Error: symbol "cSlot" not found
⊞ SlotID	0x18124e50 unsigned long * SlotID
SlotID[i]	CXX0017: Error: symbol "i" not found