

Windows® CE.NET Based Embedded System

User's Manual

Copyright Notice

©Copyright 2002 by ICP Electronics Inc. All Rights Reserved.
Manual 1st edition April 01, 2002.

The information in this document is subject to change without prior notice in order to improve reliability, design and function and does not represent a commitment on the part of the manufacturer.

In no event will the manufacturer be liable for direct, indirect, special, incidental, or consequential damages arising out of use or inability to use the product or documentation, even if advised of the possibility of such damages.

This document contains proprietary information protected by copyright. All rights are reserved. No part of this manual may be reproduced by any mechanical, electronics, or other means in any form without prior written permission of the manufacturer.

ICP Electronics Inc.

www.iei.com.tw

Table of Content

Introduction	3
Installation and Setting	4
Tutorial – Using Windows CE.NET	6
Shut down the system	6
DOS Command Prompt (CMD.EXE).....	6
Task Manager	6
IEI Auto Launch Application Solution.....	6
Touch Screen.....	7
Calibration Utility.....	7
Keyboard	7
Mouse	7
IDE Hard Disk/Flash Disk/Compact Flash Disk/CD-ROM/DVD-ROM	8
DiskOnChip.....	8
Serial Port.....	8
Parallel Port	9
Ethernet	9
Audio.....	10
Digital I/O	10
Watchdog timer.....	10
USB	11
IrDA (SIR)	11
Communication Programs	11
Browser.....	12
Application Programming.....	13
Cross Platform Development.....	13
Development Tools	13
Remote Connection between Host and Target	15
ActiveSync Serial Port Connection	15
Ethernet Connection.....	20
Visual C++ Programming	21
Win32 Programming.....	21
MFC Programming	21
Utilities	22

Introduction

Windows CE .NET is the families of the Microsoft Embedded system and is designed from the ground up for the embedded marketplace, Windows CE .NET delivers a robust real-time operating system for rapidly building the next generation of smart mobile and small footprint devices. With a complete operating system feature set and end-to-end development environment, Windows CE .NET contains everything you need to create a custom Windows CE-based device that requires rich networking, hard real-time, and a small footprint, as well as rich multimedia and Web browsing capabilities.

IEI Windows CE.NET based Embedded System

With the Windows CE.NET ready OS Image and Windows CE runtime license. You don't have to waste time on developing device drivers of board or using the Platform Builder to build the Windows CE Image by yourself. All you have to do is to develop your Windows CE.NET applications and ship the solution to your customer on the fly!

IEI Enhancement for Windows CE

IEI continues the efforts to develop Windows CE.NET drivers, tools and components to enhance Microsoft Windows CE.NET package. The key enhancements are as followings.

- Auto Launch Application Function
- Digital I/O Function
- Watchdog Timer

Installation and Setting

Windows CE.NET is an embedded operating system and the Windows CE.NET image comes with the special hardware and settings for on-board devices. So you may not change the related BIOS, IRQ, jumper, I/O address, DMA settings for the on-board devices. If you really need to change the settings, please remember the original settings before doing any changes.

DRAM Installation

16MB DRAM is usually too small to run the full configuration of Windows CE.NET image, you may need larger size of DRAM. We recommend at least 32MB DRAM for most applications. This evaluation kit installs 128MB DRAM by default.

Utility & Applications

Some utility programs are available on CD, like Autolaunch.exe, Regflush.exe etc...You can copy these files to local storage (\Hard Disk or \DiskOnChip) in Windows CE.NET and run them.

System Files

The embedded system you get should have Windows CE.NET software properly installed and the followings are the system files in Flash disk or DOC. Please don't delete these system files or the system may not boot or work normally.

Boot Loader File: loadcepc.exe

Boot loader will load the image file to memory to boot. Please don't delete this file or change the default setting.

Image File (nk.bin)

This is the Windows CE.NET image file. Please don't delete it.

Note: Microsoft Windows CE.NET and IEI software are protected by copyright laws. Please don't make any illegal copy.

Tutorial – Using Windows CE.NET

Shut down the system

Unlike other Windows OS, there is a shut down button. Because the Windows CE.NET default shell is designed for Pocket PC, the Suspend button of Windows CE.NET does not work for x86 system. To shut down the x86 Windows CE.NET system, you have to close all applications. Then reset the system or turn off power supply.

DOS Command Prompt (CMD.EXE)

The DOS command prompt provides similar function as the DOS prompt. You can copy, delete, list and execute files. If you need more information, just type **Help** for a list of commands.

You can run DOS command by select **Start > Programs > Command Prompt** or just run **cmd.exe**.

Task Manager

Press **Alt+Tab** key to run Task Manager.

Function button description:

End Task: You can stop the selected task.

Switch To: You can use arrow keys to select the task and switch to this task.

Cancel: Exit Task Manager.

IEI Auto Launch Application Solution

The default way to launch applications during Windows CE booting is to put your applications in a special Registry key and rebuild the Windows CE Image. This is too complicated and not flexible for most application developers. To solve this problem, IEI has developed the "Auto Launch" solution. With this solution, you can let the system run your applications without rebuilding the Windows CE Image.

Touch Screen

If the target device does not have the Touch Screen, please skip this session. Touch Screen is available as an option on most Panel PCs. The default pre-configured Windows CE.NET Image does not include Touch Screen driver and the serial port COM1/COM2 are available for the other device. To add a Touch Screen to the Windows CE.NET platform, the Windows CE Image needs to be rebuilt with the Touch Screen driver. The Touch Screen calibration utility is located in **My computer > Windows** of Windows CE.NET.

Calibration Utility

You need to calibrate after first time booting to Windows CE.NET. Here is the procedure:

1. Run the calibrate.exe for the Touch Screen.
2. Just follow the on-screen instruction to complete the calibration.
3. Then the Touch Screen function will work.

Keyboard

Most applications in embedded system don't need a keyboard. However, you may need to configure your embedded system by connecting a standard keyboard. After that, you could power off the system, then remove the keyboard and reboot.

Mouse

PS/2 Mouse is the default pointing device of Windows CE.NET. For some systems that do not support a PS/2 Mouse, it just supports Serial Mouse as the pointing device. Like the keyboard, the system doesn't need a mouse for most embedded applications.

IDE Hard Disk/Flash Disk/Compact Flash Disk/CD-ROM/DVD-ROM

If the target device does not support IDE device, please skip this session. The IDE driver built in for the pre-configured Windows CE.NET Image supports the IDE Interface device, like IDE HDD, Flash disk, CompactFlash™ Card, CD-ROM and DVD-ROM.

Directory Name: There is no drive letter (A:, C: or D:) in Windows CE.NET. The directory name for the IDE device is:
HDD: \Hard Disk.
CD-ROM: \CDROM Drive.

DiskOnChip

The M-Systems TrueFFS DiskOnChip driver is built into the pre-configured Windows CE.NET Image if the board has onboard DiskOnChip socket. Please skip this session if the target device does not have DiskOnChip socket.

Directory: There is no drive letter (A:, C: and D:) in Windows CE.NET. The directory name for DiskOnChip is \DiskOnChip.

Serial Port

Serial Port Testing Procedure:

1. Prepare another Windows 95/98/NT Client PC.
2. Prepare a Null Modem cable to connect Client PC and Target Windows CE.NET Device.
3. Start the HyperTerminal.exe (Hyperterm.exe) on Windows 95/98/NT PC, set the baud rate as 19200.
4. On the Windows CE.NET Target device
Run the program **pegterm.exe**
Create a New Session.
Input any number in the Telephone Number field.
Select **Force Local**
Click **Configure...**
Select **Manual Dial** in Port Settings.

Set the baud rate as same as 95/98/NT PC.

Type some characters on both keyboards to verify the connection function work or not.

Parallel Port

Windows CE.NET only supports PCL printer driver. Most HP printers should support PCL function. If your printer does not support PCL function, then you have to develop the Windows CE.NET printer driver.

To test the printer port, you may:

Connect a PCL compatible printer and use the IE to print the web page.

Ethernet

Network Setting

If the target device you have does not have network chip, please skip this session. In Windows CE.NET you need to change the Device Name in the Control Panel before using the network.

Please do the procedure as followings:

1. Go to **Control Panel > Communications** Properties.
2. Change the default name "**WinCE**" to any other name.
Ps: You have to select a unique name.
4. You may change the Network and TCP/IP settings. (Go to **Control Panel > Network**) The default IP setting is DHCP. You may change to static IP.
5. Click "suspend" in Start Menu to save the Registry setting.
6. Reboot the system.
7. You can use **net** command and **UNC** to access Windows 95/98/NT/2000 share directories and files. However, Windows CE.NET does not support Windows 95/98/NT/2000 to access files on Windows CE.NET device. You may develop an application like FTP to help out with this solution.

Audio

If the target device does not support audio function, please skip this session.

You may go to **Control Panel > Volume & Sounds** to adjust the volume or change the .WAV file for sounds of system events.

Digital I/O

If the target device does not support Digital I/O function, please skip this session.

`_inp()`, `_inpw()`, `_outp()` and `_outpw()` functions can be used in your Windows CE.NET application to access the I/O address directly.

IEI I/O Port Test Utility (DIOTST.exe)

You may use the utility to read and write data of I/O ports and test the digital I/O functions.

Watchdog timer

If the target device does not support watchdog timer function, please skip this session.

`_inp()`, `_inpw()`, `_outp()` and `_outpw()` function calls can be used in your Windows CE.NET application to access the I/O port, and you can enable/disable the watchdog timer. Different boards may have different I/O to enable/disable watchdog timer. For more information, please refer to hardware manual.

IEI Watchdog Timer Test Utility (WDTTST.exe)

You may use the utility to test the on board watchdog timer function.

USB

If the target device does not support USB, please skip this session.

To enable USB function, please follow the procedure:

1. Enable USB function in BIOS setting.
2. Power off the Target device.
3. Power on the Target device.
4. Plug in USB device like mouse.

Note:

To enable USB device, you need the Windows CE.NET driver for the USB device.

IrDA (SIR)

If the target device does not support IrDA function, please skip this session.

Please do the following procedure to enable IrDA:

1. Enable IrDA in BIOS setting.
2. Connect an IrDA Adapter to the Target device. Example: IEI IR210L Infrared data communication adaptor.

Communication Programs

Click **Start > Settings > Network and Dial-up Connections** .

Remote Networking (connmc.exe): Check Topics about "ActiveSync Serial Port Connection", "Ethernet Connection"

Browser

The browser is from Windows CE.NET Platform Builder. Microsoft provides IESample as a sample. Developers can refer to the sample code to develop the browser.

For more information, see:

Programmer's Guide to Internet Explorer for Microsoft Windows CE.NET.

<http://www.microsoft.com/windows/embedded/ce.net>

Application Programming

Cross Platform Development

For Windows 98/NT/2000 application programming, the development tool like Visual C++ and application can be on the same PC. So developer can develop, debug and test application on the same PC. However, this may be impossible for most embedded system development because some embedded systems are headless (no display, keyboard or mouse) or the CPU is not powerful enough to install the development tool. Cross platform development is the solution. The cross platform development includes a powerful development host, target embedded system and the cable connection between two system.

Windows CE Application Cross Platform Development Environment

Host: PC with development tools

Target: Your Windows CE Embedded System

Connection: Ethernet or serial port connection

Development Tools

The WinCE.NET support the managed (.NET-enabled) and unmanaged (native) code.

Managed Code: The .NET Compact Framework makes available to programmers a rich framework include UI classes, data access, XML support, automatic memory management, and garbage collection. Managed code is executed by one of the primary components of the .NET Framework, the Common Language Runtime (CLR). The CLR is a high performance execution engine that provides run-time services such as memory management, cross-language inheritance, code access security, and automatic lifetime control of objects. Visual Studio .NET provides the development environment for creating

managed code:

- **Visual Studio .NET**
- **Visual Basic .NET**
- **Visual C++ .NET**
- **Visual C# .NET**

Unmanaged code: Often referred to as "native code", unmanaged code executes directly against Windows CE.NET, providing backward compatibility and performance advantages. For creating applications or components using unmanaged code, developers can use:

- **eMbedded Visual C++ 4.0**

In this manual, we only mention the unmanaged code programming, if you want to program using the .NET framework please refer to the Microsoft .NET documents.

IEI Platform SDK

To develop Windows CE application for IEI platform, you need to install the platform SDK. There is a SDK in the CD, which is for using Embedded Visual C++ 4.0 to develop your application.

Development Host System Requirement

1. PC with Pentium processor, Pentium 150 MHz or higher.
2. Windows NT Workstation 4.0 with Service Pack 5 or later, or Microsoft Windows 2000.

Development Host Installation

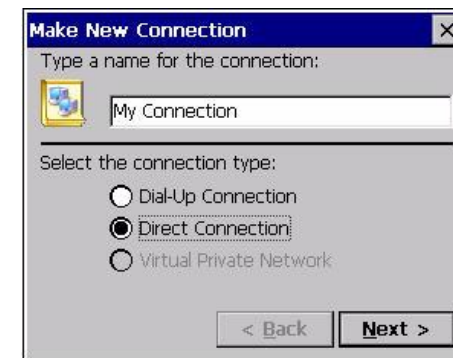
1. Install Windows NT 4.0 + Service Pack 5 or later, or Windows 2000.
2. Install Microsoft eMbedded Visual C++ 4.0 .
3. Install IEI Platform SDK.

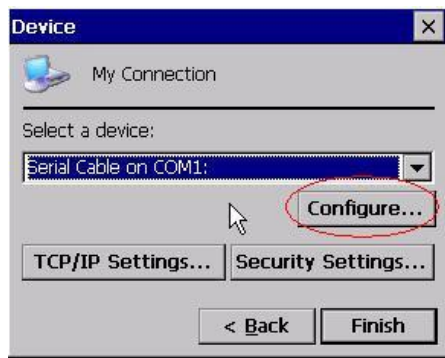
Remote Connection between Host and Target

You may select Ethernet or serial port connection. However, you can only select one at a time for a target device. If the platform has Ethernet, please choose Ethernet connection, because Ethernet connection is much faster. If the platform doesn't have Ethernet function, please select serial port connection.

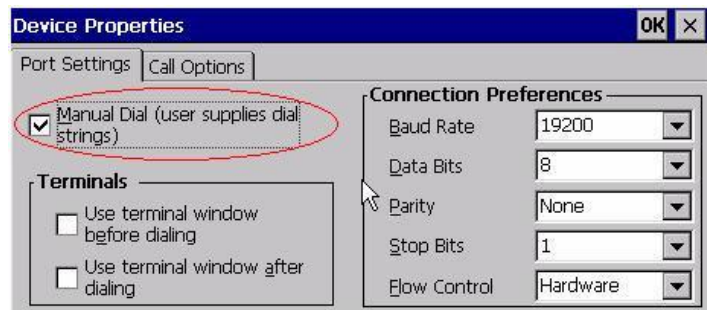
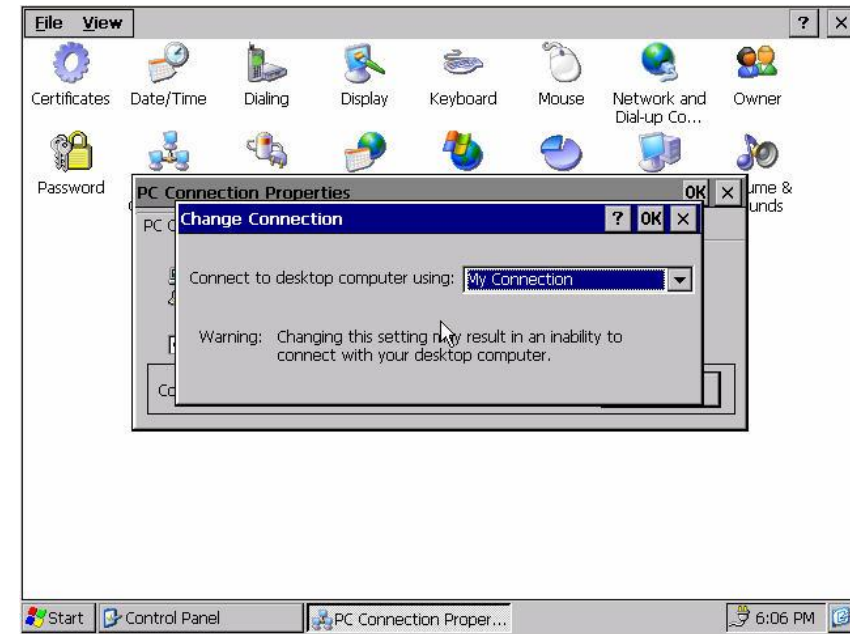
ActiveSync Serial Port Connection

1. Prepare a Null Modem serial cable. Please make sure the cable pin-out is exactly the same as in the Appendix. Use this cable to connect the development host PC and Windows CE device.
2. On the Windows CE.NET Target device: Run **connmc.exe** Click the "**Make New Connection**". Enter any name for the connection (e.g. "My Connection", then click the "**Direct Connection**" option. Then click "**Next**". Select "**Serial Cable on COM1**". Click on the "**Configure...**" button. Set the Connection Preferences as follows:





Baud Rate: 19200 (or other speed)
 Data Bits: 8
 Parity: None
 Stop Bits: 1
 Flow Control: Hardware



Note: Make sure you have enable "Manual Dial" in Port Settings and your host PC has the same serial port settings (Baud Rate and others) as the above.

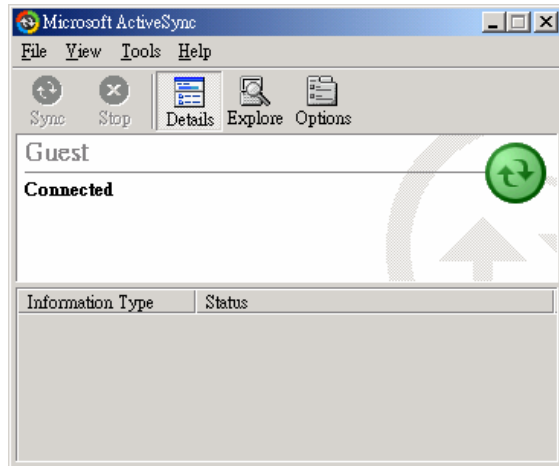
Then close the "**Direct Connection**" window. Open control panel. Double click the "**Communication**" icon. Click the "**PC Connection**" Tab. Select the "**Change...**" button. Select the name that you create before (e.g. "My Connection"). Then close the "**Communications Properties**" window. Close the "**Control Panel**" window.

3. Download ActiveSync 3.5 from Microsoft web site to your PC.
4. Install ActiveSync 3.5 on your development host PC.
5. Run ActiveSync 3.5. When the "**Get Connected**" screen is shown on the host system. On the Windows CE.NET system, click on the "Start" then click on "Run" and type "repllog" (but don't hit "OK").



- On the Host system "**Get Connected**" screen, click "**Next**". Then click "**OK**" on the Windows CE.NET device "**Run**" screen (a "**connecting to...**" box will appear).
- When a serial port connection has been established between the host system and the Windows CE.NET device, the "ActiveSync" icon on the Host PC task bar will turn green, and a "**New Partnership**" screen will appear.
- Choose the "**No**" option that states, "I don't want to synchronize information".

Note:



Configure Platform for ActiveSync Serial Connection

- Run eMbedded Visual C++ 4.0.
- Run Tools > "**Configure Platform Manager...**" .
- Select the right platform/device (e.g. Wafer 582x) to configure.
- Click "**Properties ...**" .
- Select "**Microsoft ActiveSync**".
- Click "**Advanced**" button.
- Select "**Microsoft ActiveSync**" and press "**OK**" button.
- Press "**Test**" button to test the connection.

Note: You have to change the "device name" in Windows CE control panel for every new ActiveSync connection, or there will be a duplicate name warning on Windows CE device.

- Once connected, you can use remote tools (e.g. Remote Registry Editor) on the eMbedded Visual C++.

Ethernet Connection

1. Connect development host and Windows CE device to the network. (Make sure they both get the IP addresses)
2. Run eMbedded Visual C++ 4.0.
3. Run Tools > "**Configure Platform Manager...**".
4. Select the right platform/device (e.g. Wafer 582x) to configure.
5. Click "**Properties ...**".
6. Select "**TCP/IP Transport for Windows CE**" in "**Transport**".
7. Click "**Configure**".
8. Select "**Fixed Port**" and press "**OK**" button.
9. Select "**Manual Server**" in "**Startup Server**".
10. Press "**Test**" button to test the connection.
11. It shows command line parameters of **CEMGR.CEXE**. Write down the long command line on paper. Press "**OK**".
12. Enter the long command line on Windows CE device. You may need to enter this long string very frequently for the future debugging. The best way is to write a .bat file and reuse it.
13. Once connected, you may use the remote tools like "Remote Registry Editor".

Visual C++ Programming

Win32 Programming

This sample shows you how to develop a simple Win32 Hello World application.

Start eMbedded Visual C++ 4.0. Select **New** from the **File** menu, click on **Projects** and select **WCE Application**. Then enter the project name **HelloW32**, in the directory of your choice. Also select the proper platform (e.g. **Win32 (WCE x86)** and **Win32 (WCE x86em)**). Click **OK**.

The **WCE Application - Step 1 of 1** window is shown. Select **A typical "Hello World" application** and click **Finish**. This generates a Win32 application that displays... you know what by now. You can see in the left panel (called the Workspace) some files being listed.

Select **Build HelloW32.exe** from the **Build** menu. The entire project is built in a few seconds and the executable (HelloW32.exe) is copied to the target Windows CE device through serial ActiveSync or Ethernet connection you set before.

MFC Programming

This sample shows you how to develop a simple MFC application.

Start eMbedded Visual C++ 4.0. Select **New** from the **File** menu, click on **Projects** and select **WCE MFC AppWizard (exe)**. Then enter the project name **HelloMFC**, in the directory of your choice. Also select the proper platform (e.g. **Win32 (WCE x86)** and **Win32 (WCE x86em)**). Click **OK**.

You may use default settings for the reset of Wizard steps

and click **Finish**.

Select **Build HelloMFC.exe** from the **Build** menu. The entire project is built in a few seconds and the executable (HelloMFC.exe) is copied to the target Windows CE device through serial ActiveSync or Ethernet connection you set before.

Utilities

Autolaunch.EXE

This utility allows you to execute application that you want to run first after booting to Windows CE.NET.

Copy this file to DOC or HDD through neighborhood or ActiveSync, then double click autolaunch.exe.

Just click browse to select the application you want to run first after booting to Windows CE.NET.

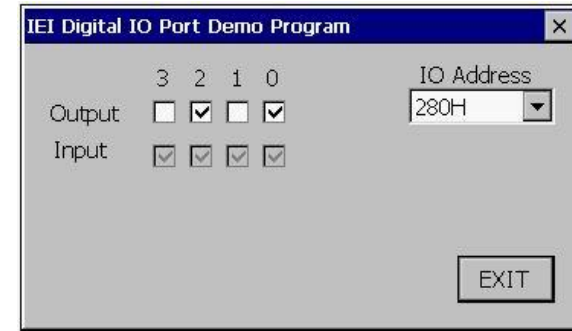
Next time when booting to Windows CE.NET, the application will run automatically.

Regflush.EXE

This utility helps you to save settings in registry. Even after power reset, these setting will be kept in your DOC or HDD device. For example, you can set IP address as fixed IP then double click regflush.exe. After reset power and booting to Windows CE.NET, the IP address setting will be the same as before.

DIOTST.EXE

Digital I/O test utility, it helps you to test digital input and output function on your board.



WDTTST.EXE

Watchdog Timer function test utility, after setting time and selecting proper IO address (please refer to your user manual), press Start Timer button to trigger watchdog timer and start count down. Press Rest Timer button to restart timer or the system will issue a reset to perform a cold boot.

