

Windows® CE 5.0 Starter Kit (Standard Edition)

HiCO.ARM9_1ST-CE500

Windows® CE 5.0 Starter Kit (Professional Edition)

HiCO.ARM9_1STPro-CE500

User Manual

Copyright

emtrion

All rights reserved. Without written permission this documentation may neither be photocopied nor stored on electronic media. The information contained in this documentation is subject to change without prior notification. We do not assume any liability for erroneous information or its consequences. The trademarks of other companies that are used identify the products of these companies exclusively. Microsoft and Windows® are registered trademarks of the company Microsoft.

Manual Revision No.	Changes	Date
1	First revision	09.06.2006 / MI
2	Corrections	26.07.2006 / Ft
3	Updates and new features	25.10.2006 /MI

This document is published by:

emtrion GmbH
Greschbachstr. 12
D-76229 Karlsruhe
Germany

Tel.: +49 (0) 721 / 62725 - 0
Fax.: +49 (0) 721 / 62725 - 19
E-mail: mail@emtrion.de
Internet: www.emtrion.de

Table of Contents

1.	Terms and Conditions of Licence	7
2.	Introduction	8
3.	Package Contents	9
3.1.	Standard Edition	9
3.2.	Professional Edition	10
4.	System Requirements	11
5.	Definitions	12
6.	Starter Kit CD	13
6.1.	Navigation	13
6.2.	Folders	14
7.	Support	15
8.	Starter Kit Components	16
8.1.	Operating System Kernels	16
8.1.1.	Overview of the Operating System Kernel	17
8.2.	Software Development Kit (SDK)	19
8.3.	Supplements to Platform Builder 5.0	19
8.4.	Documentation	20
9.	Initial Operation of the Starter Kit	21
9.1.	Before Installing and Starting Up the System	21
9.2.	Preparing the HiCOARM9_1 for Operation	22
9.3.	Software Installation on the Desktop Computer	23
10.	Data Exchange between the Target and the Desktop Computer	25
10.1.	Preparing the Desktop Computer	25
10.2.	Preparing the Windows CE Device	26
10.3.	USB- Connection	26
10.4.	Connection via Network	27
10.5.	Remarks on Using ActiveSync	29
11.	The Bootloader	35
12.	Working with eMbedded Visual C++ 4.0	36
12.1.	Using eMbedded Visual C++ 4.0	36
12.2.	Creating an Application Project	37
12.3.	Debugging an Application	39
12.3.1.	Preparing the Debug Session	40

12.3.2.	Debugging	40
12.3.3.	Starting the Debugger	41
13.	Remote Tools for Windows® CE	42
13.1.	Remote Spy++	42
13.2.	Remote Registry Editor	43
13.3.	Remote Heap Walker	43
13.4.	Remote Process Viewer	43
13.5.	Remote Zoomin	44
13.6.	Remote File Viewer	44
14.	Starting the Application at the System Start	45
14.1.	OS Kernels with a Persistent Registry	45
14.2.	Start with the Supplied OS Kernels	46
15.	Sample Application	49
15.1.	In General	49
15.2.	How the Demo Application Works	49
15.3.	Changing the Parameters via the Menu	50
15.4.	Changing the Port	50
16.	Persistent Registry	51
16.1.	Deleting the Persistent Registry	53
16.1.1.	Deleting the Registry by means of the Bootloader	53
16.1.2.	Deleting the Registry by means of an Application under Windows CE ...	53
17.	Exchanging the Operating System Kernel	56
18.	The "KernelDownload" Tool	57
18.1.	Description	57
18.2.	Operating Elements	58
18.2.1.	Menus	58
18.2.2.	Toolbar	59
18.2.3.	Configuring the Download Tool	60
19.	emtrion Tools for Windows CE	61
19.1.	Writing the Persistent Registry	61
19.2.	Determining the Version of the Operating System Kernel	61
20.	Modifying the Kernel, BSP and Bootloader	62
21.	FAQs	63
21.1.	In General	63
21.2.	Debugging an Application Using eMbedded Visual C++	65

22.	Licence Agreements	67
22.1.	Software Licence Agreement emtrion GmbH.....	67
22.2.	Licence Agreement Microsoft.....	68
INDEX	70	

1. Terms and Conditions of Licence

The Windows® CE Starter Kit for HiCOARM9_1 comes with a runtime licence for the Windows® CE operating system and the associated drivers. Distribution of any product created by use of the Windows® CE operating system and the associated drivers requires the purchase of separate licences.

For the supplied software, the Licence Agreements shown in section "Licence Agreements" on p. 67ff from emtrion and Microsoft apply.

BY INSTALLING THE STARTER KIT ON YOUR SYSTEM, YOU ARE AGREEING TO BE BOUND BY THE TERMS OF THOSE AGREEMENTS.

2. Introduction

The Windows® CE starter kits HiCOARM9_1ST-CE500 or HiCOARM9_1STPro-CE500 are the ideal solution to acquaint oneself with the Windows® CE 5.0 embedded world. It allows the developer to quickly build flexible and reliable designs based on the Windows® CE operating system platform.

The Starter Kit comes with preconfigured operating system kernels, which eliminates the time-consuming task of creating and testing the operating system kernel. The developer can fully concentrate on creating their application.

The supplied operating system kernels provide virtually all components and functions of Microsoft's popular Windows® CE Win32 APIs and have a size of 17 MB, which allows them to be used for wide range of applications.

A detailed description of the operating system kernels can be found in the section "Starter Kit Components" on [p. 16](#).

Specifically tailored operating system kernels may, for example, include special fonts, keyboard layout or additional drivers. These components will either be supplied by the customer or designed by emtrion, if desired.

3. Package Contents

3.1. Standard Edition

The Starter Kit (Standard Edition) for Windows® CE contains the following components:

- 1 assembly composed of
 - 1 HiCOARM9_1-SBC consisting of
 - 1 HiCOARM9_1 single board computer module with 64 MB SDRAM and 32 MB Flash
 - 1 Color TFT Display TX14D11VM1CAA with Touch (resolution 320x240)
 - 1 extension cord for the Touch (applicable for the Touch of the TX14D11VM1CAA)
 - 1 Backlight-Inverter (applicable for the TFT14D11VM1CAA)
 - 1 power cable for the Backlight-Inverter
 - PC/104 bolts, assembly Kit
- 5V DC power supply with appropriate wall socket adapters for Europe, UK and US
- 1 Null modem cable
- 1:1 ethernet cable for connecting to a hub (blue cable)
- Twisted-pair cable for connecting to the PC (grey cable)
- 1 Runtime Licence Windows® CE¹
- 1 USB Hub
- 1 USB flash device (USB Stick)
- 1 adapter for the serial port COM0
- 1 Secure Digital Memory Device (SD Card)

- 1 CD-ROM containing the Starter Kit software for the Standard Edition
- User Manual
- Platform Builder 5.0 Evaluation Version from Microsoft (the DVD containing the eMbedded Visual C++ 4.0 development environment)

¹ The licence fees for the runtime licence were paid to Microsoft.

3.2. Professional Edition

The Starter Kit (Professional Edition) for Windows® CE contains the same as the Standard Edition except the CD-ROM with the Starter kit software. The CD-ROM with the Starter Kit software for the Professional Edition contains also source code of the Board Support Package and schematics of the HiCOARM9_1.

4. System Requirements

Currently, the system requirements are as follows:

- Desktop computer with at least 200 MB of free hard disk space (when installing all Starter Kit components) and one of the following operating systems:
 - Windows® 2000 workstation with service pack 4 or later
 - Windows® XP Professional with service pack 1 or later

Remark:

If you have not already installed the Windows CE Platform Builder 5.0 you will also need at least 2,0 GB free hard disk space for the Windows CE Platform Builder 5.0 installation

For the HiCOARM9_1 the following peripheral devices are required:

- USB keyboard
- USB mouse

5. Definitions

The following is a definition of terms more or less used throughout this manual:

ActiveSync	A program from Microsoft used for the exchange of data between the Windows® CE computer and a desktop computer.
Applet	Program of the Control Panel which serves to control the system's properties.
Operating system kernel	Complete Windows® CE operating system running on the target platform. During the creation of the Windows® CE OS kernel, modules and components may be omitted. This will create different OS kernels, which can each be found in an image file.
Desktop computer	PC with Windows® 2000 and Service Pack 4 or later; the development environment (eMbeddedVisual C++ 4.0 and/or Platform Builder 5.0) will be installed on this computer.
Image (file)	File that is used by the bootloader and stored in the flash or RAM for execution.
Persistent registry	Allows modifications to the registry which are still in effect after power off/on. A non-persistent registry of Windows® CE is created from the image file every time the system is started up. Modifications made after system startup will not be in effect after a restart.
Software Development Kit (SDK)	Installable collection of header- and library files. The tools will be informed about the API functions supported by the associated OS kernel, and whether MFC, ActiveX, etc. is supported.
Target platform	Target hardware on which Windows® CE is to run. In this Starter Kit, the HiCOARM9_1 module along with accessories like display, keyboard, mouse, additional boards.
HiCOARM9_1	The complete target platform

6. Starter Kit CD

6.1. Navigation

The CD startup page allows you to choose from the following:

- Installation of the Starter Kit components
A detailed description can be found in section "Starter Kit Components" on p.16.
- Documentation
The CD contains this user manual and the documentation for the HiCOARM9_1 as PDF files. To be viewed, these files require an Acrobat viewer such as Acrobat Reader. Acrobat Reader, which is free of charge, can also be found on this CD.
- Information on emtrion and its products

In the professional version the starter kit CD shows also

- Additional Information on the supplied hardware
This version contains the schematics, additional manuals

6.2. Folders

Folder Name		Contents
	Acrobat	Installation program of Acrobat Reader, both the German (AdbeRdr60_deu.exe) and English version (AdbeRdr60_enu.exe)
	ActiveSync	Microsoft ActiveSync 3.7 installation program, German (msasync_V37_deu.exe) and English version (msasync_V37_eng.exe)
	Bootloader	Update application for the bootloader of the HiCOARM9_1
	emtrion_In fo	emtrion internet pages containing information on emtrion and its products. (These pages may also be read offline, there is no need to connect to the internet.)
	HiCOARM9_1	Data sheets and hardware manual of the HiCOARM9_1
	Datasheets	Data sheets of the components installed on the HiCOARM9_1
	Manual	User manual of the HiCOARM9_1 hardware in English
	Schematics	- professional edition of the starter kit only - schematics of the HiCOARM9_1
	Html	The HTML files of the CD's startup page
	Starterkit	Installation- and data files of the Windows® CE Starter Kit
	Kernel	Image files of the supplied operating system kernels
	Manual	This manual as a PDF file (in English)
	Sample	Sources of programming samples
	SampleApp	Source of the starter kit sample application
	MmMapIoSpace	Source of the programming sample for accessing specific hardware addresses
	NulDriver	Source of the programming sample for a stream interface driver
	SDK	Software Development Kits for the supplied operating system kernels
	Source	Project environments to rebuild the supplied operating system kernels
	Specifications	The specifications of the Starter Kit kernel and an empty form for the specification of own kernels
	Support	Form for request for support
	USBFDriver	Installation files for the desktop. With this driver a connection with ActiveSync over USB is possible

7. Support

This product has been thoroughly tested over the development period. Due to its complexity, however, no guarantee can be given that it will seamlessly operate under any circumstances. We are therefore grateful for any feedback regarding an incorrect operation of the boards.

If any problems should occur, have a look at the FAQ section of this manual first. Or visit our website at http://www.emtrion.com/support_en.php for the latest FAQ.

If you cannot find the necessary information, contact our Support Team via e-mail, fax or phone.

To accelerate the process, please fill out the supplied form, which can be found in the Support directory of the CD or on the internet at http://www.emtrion.com/support_form_en.php.

To handle a support request we need at least the following information :

- Where is sending the request (company name and the name of the writer)
- A valid email address, where we can send the answer
- The product name (here :
 - HiCOARM9_1ST-CE500 for the Standard Edition of the starter kit
 - or
 - HiCOARM9_1STPro-CE500 for the Professional Edition of the starter kit)
- The product version and release date, which is printed on the CD (NOT the version of the Windows CE operating system)
- A detailed description of the problem and how it can produced.
- Recommended : all applications which are required for the reproducibility and which are not included in the delivery of this starter kit

8. Starter Kit Components

The Starter Kit consists of an operating system kernel, the corresponding Software Development Kit (SDK) and the Board Support Package (BSP) for Windows® CE Platform Builder 5.0. The components, which are detailed below, can be installed on your PC's hard disk through the installation program.

8.1. Operating System Kernels

The Starter Kit's operating system kernel is the same for the Standard and Professional Edition. There is no difference in their functional scope.

Additionally, customer-specific operating system kernels are available on request. (See page 2 for contact info.)

8.1.1. Overview of the Operating System Kernel

The Operating System Kernel is based on the OS-Design Revision 1.1 and is in English.

HiCOARM9_1SKit_CE500 (Standard and Professional Edition included)

- .Net Compact Framework 2.0
- Library Microsoft Foundation Classes (MFC)
- Autostart mechanism for launching application from Mass-Storage-Devices (e.g. SD Card)
- Graphical User Interface
- Complete WIN32-API, available for Windows CE
- Driver for the on-board Ethernet
- USB Host (supports USB keyboard, USB mouse, USB storage (USB sticks) and USB printer)
- 2 serial ports (COM0: and COM1:)
- USB function (serial interface class for active sync support)
- SD Card support
- Pocket Internet Explorer
- Telnet and FTP server functionality

The operating system kernel contains a persistent registry.
There is one English version available for all kernels.

Support for touch is under preparation. Regarding the touch, the BSP of the Starterkit is still prepared for a later inclusion of the touch driver. For this situation please don't try to add the touch driver component from the catalog to the OS Design of this revision. It will result in errors by building the kernel.

If you need any of these components please contact emtrion for an update. Some parts have some known issues. Please look into the release notes to get the latest information about the known issues for this revision of the starter kit.

For a more detailed overview of all Starter Kit components, please see the \Starterkit\Specifications subfolder on CD.

Remarks

Persistent registry

All operating system (OS) kernels supplied with the CD-ROM currently have a persistent registry. OS kernels without a persistent registry will not keep the modifications made before a restart.

On the other hand, this will prevent the system from being damaged due to possibly wrong settings. OS kernels without a persistent registry can also be ordered at emtrion.

Starting user applications

All OS kernels contain an entry that automatically starts an application when the system is started up. For more details, please refer to section "Starting the Application at the System Start" from p. 45.

Available screen resolutions

At present, the driver integrated with the operating system kernels supports the following resolutions:

1. For the displays from:
 - Hitachi TX14D11VM1CAA; resolution 320 x 240
 - 256 colours, image rotation 0°

8.2. Software Development Kit (SDK)

The CD contains the Software Development Kit for eMbedded Visual C++ 4.0 for all operating system kernels.

Note

An SDK is valid for one specific operating system kernel. Although an application designed for a specific operating system might also run on another OS kernel, this is true only if **all APIs** used by the application are available in both kernels.

To avoid errors caused by functions that are called but unavailable in an OS kernel, we recommend that you always use the appropriate SDK.

8.3. Supplements to Platform Builder 5.0

The Starter Kit contains the board support package (BSP) that will help you build your own operating system kernels for the HiCOARM9_1. In the standard version of the starter kit you the CD contains the BSP with binary files, in the professional version the CD contains the BSP with source code.

The Sample projects subcomponent is required only if you wish to rebuild the supplied operating system kernels.

In the professional version of the starter kit the CD contains also the source code of the bootloader.

Note

This subcomponent can only be installed if the installation program detects an installed version of Platform Builder 5.0.

Remark:

emtrion also offers to make a custom specific kernel

8.4. Documentation

The Starter Kit CD contains this manual as well as additional software- and hardware manuals for the HiCOARM9_1 as PDF files. You may read these manuals from the CD or copy them to your hard drive.

Hint

View the PDF files, Acrobat Reader is required. The setup file for Acrobat Reader can be found on the CD in the ...\\Acrobat folder.

9. Initial Operation of the Starter Kit

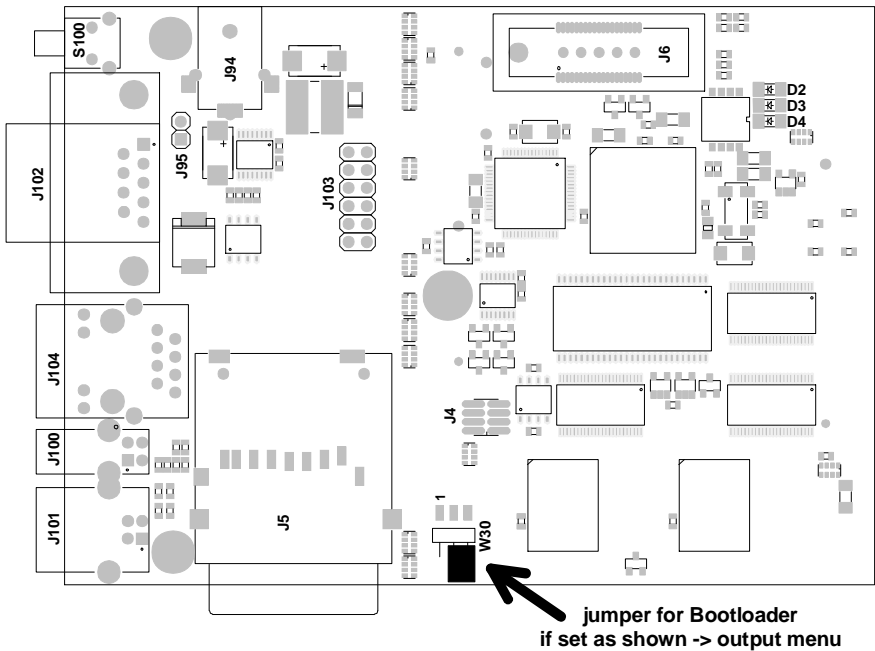
9.1. Before Installing and Starting Up the System

Please read the following notes prior to installing the Starter Kit. They apply to all ESD (electrostatic discharge) components:

- Before installing the module it is recommended that you discharge yourself by touching a grounded object.
- Be sure all tools required for installation are electrostatic discharged as well.
- Before installing (or removing) a board, remove the power cable from your mains supply.
- Handle the board with care and try to avoid touching its components or tracks.

9.2. Preparing the HiCOARM9_1 for Operation

1. Connect the USB hub, USB keyboard and USB mouse to the HiCOARM9_1 board.
2. Connect the serial cable to the COM1 port on the HiCOARM9_1 board (connector J102).



3. Connect the HiCOARM9_1 to your network. For this, use the (blue) 1:1 cable if the connection is established via a hub/switch. For the direct connection between the HiCOARM9_1 and the PC, use the (grey) twisted-pair network cable.
4. Connect the power supply to the HiCOARM9_1.

9.3. Software Installation on the Desktop Computer

For a proper operation, the TCP/IP protocol must be installed on the desktop computer.

Note

To save memory, select the components for the ARMV4I processor only, where is possible during the installation in the steps 1 and 2.

1. If not already installed, install Windows CE Platform Builder on your development workstation. For this, insert the DVD in your DVD drive and run `setup.exe` (unless stated automatically).
2. Install Microsoft eMbedded Visual C++ 4.0. For this, insert the DVD in your DVD drive and run **setup.exe** (unless started automatically).

Note

You have to install the Service Pack 4 for the eMbedded Visual C++ also, because this is required to develop applications for Windows CE 5.0

3. Insert the Starter Kit CD in your CD-ROM drive. This will automatically bring up the HTML browser installed on your computer (e.g. Internet Explorer), which allows you to explore the contents of this CD.

Hints

- If your browser does not start automatically, run **Autorun.exe** from the CD's main folder. Alternatively, you can also call **start.html** from the **Html** subfolder.
 - You may now choose from the following components:
 1. Installation of the Starter Kit components
A detailed description of the individual components can be found in chapter "Starter Kit Components" on [p. 16](#).
 2. Documentation
The CD contains this user manual and the documentation for the HiCOARM9_1 as PDF files. To be viewed, these files require an Acrobat viewer such as Acrobat Reader. Acrobat Reader is free of charge and can also be found on this CD.
 3. Information on emtrion and its products
-

4. Select **Windows CE Starterkit** from the navigation bar on the left.
5. Click the **Start Setup** button.
Select **Start Program Here** from the following dialog box and click **YES**.

6. Follow the instructions on the screen.

Note

- The default settings of the "**select components**" dialog box are sufficient for applications using the preinstalled operating system kernel. If you intend to create applications for another operating system kernel or to build your own kernel(s), you will need to select additional components. A detailed description of the individual components can be found in Chapter "Starter Kit Components" on [p. 16](#).
In any case, be sure to install the SDK associated with the operating system kernel when selecting the components.
 - The options which require the Platform Builder (Board Support Package, Bootloader source code) are only displayed if you use the standard or professional edition of the starter kit and a installation of the Windows CE Platform Builder 5.0 is found by the setup program.
 - During the installation you are prompted to start the selected installation programs of the SDKs. This is recommended as they are required from the beginning of application development.
-

7. Install ActiveSync as is described in section "Preparing the Desktop Computer" on [p. 25](#).

The installation of the Starter Kit is now complete.

Note

The size of the memory space indicated in the system requirements is sufficient only if you perform a user-defined setup and select the codes for the ARMV4I Platform only.

10. Data Exchange between the Target and the Desktop Computer

The Starter Kit's operating system kernels allow the exchange of data between the Windows® CE computer and a computer on which the Windows® 2000 or Windows® XP operating system is running. For this, Microsoft's ActiveSync program is provided. The installation program for ActiveSync on a Windows® 2000 or Windows XP machine can be found on the starter kit CD. The latest version is available on the internet at:

<http://search.microsoft.com/search/results.aspx?st=b&qu=ActiveSync&view=en-us>.

Both the Starter Kit components and the components that come with Platform Builder allow the exchange of data via a serial connection or via network.

If you have not worked with Microsoft's ActiveSync yet, we recommend that you have a look at section "Remarks on Using ActiveSync" on [p. 29](#).

10.1. Preparing the Desktop Computer

If Microsoft ActiveSync has not yet been installed on your desktop computer, it must be prepared as follows:

Windows 2000:

1. From the Control panel choose "Modem" and install the communications cable between two computers.
2. Start **msasync_V3.7_eng.exe** or **msasync_V3.7_deu.exe** in the **ActiveSync** directory of starter kit CD .

With all other Windows versions:

Start **msasync_V3.7_eng.exe** or **msasync_V3.7_deu.exe** in the **ActiveSync** directory of the starter kit CD.

10.2. Preparing the Windows CE Device

Communication via a serial COM-port is possible but not recommended. Instead use USB-Function for ActiveSync. This way of communication is the default setting of the delivered kernel.

10.3. USB- Connection

1. Establish a connection between HiCOARM9_1 and the desktop computer via the supplied USB cable.
2. Connect the board with the desktop computer and then switch on HiCOARM9_1. The Windows® CE OS kernel stored in the Flash will be automatically started. When HiCOARM9_1 is unknown as USB-Device for the desktop computer it will ask you for the corresponding USB-function-driver which has to be installed. The driver is located in the directory USBFdriver on the delivered CD.
3. Start ActiveSync on the desktop computer (unless automatically started at the system start). Please note, USB has to be selected in the connectivity settings of ActiveSync

HiCOARM9_1 establishes a connection with the desktop computer and possibly prompts you to login and enter your password. Enter the same name and password as you are using for your desktop computer.

Note


If the connection between the HiCOARM9_1 and the PC is **not** established automatically, you should start the “repllog” program on the HiCOARM9_1 platform manually. To do this, select “Run” from the “Start” menu, enter **repllog** and click the **OK** button.

The **New Partnership** dialog appears on the desktop computer.

5. Select the options as described in section "Remarks on Using ActiveSync" from [p. 29](#).
6. After establishing a connection, click **Explore** to open a window. Via this window, you may then exchange files with the HiCOARM9_1 target platform using drag and drop. For this, drag the files onto the desired directory on the HiCOARM9_1.

Note


The folder contents of the HiCOARM9_1 target platform will be created from the flash when Windows® CE is started. If you copy any files to these folders, they are not available any more when Windows® CE is restarted. This does not apply to folders which directly represent mass storage. This could be the ..\FlashFX Disk or the ..\SD Card. These folders are available only if the corresponding devices are detected by the system. An exception are also all subfolders of the "Network" folder. These folders represent drives with which a connection was established via network.

7. To terminate the connection, click the  symbol on the Windows® CE's task bar. The connection may be separated in the dialog that appears.

10.4. Connection via Network

Note

The steps 1 to 2 will need to be performed only once with OS kernels that have a persistent registry.


1. Establish a connection with the desktop computer as described in section "Serial Connection" on [p. 26](#). In doing so, note that a partnership will have to be created.
2. Disconnect now by double-clicking the  symbol in the task bar of the Windows® CE computer, and selecting "Disconnect" in the following dialog.
3. On the HiCOARM9_1 platform select "Run" from the "Start" menu. Enter **repllog /remote** and click the "OK" button.

Select the network connection from the following dialog and click the "Connect..." button.

HiCOARM9_1 establishes a connection with the desktop computer and possibly prompts you to login and enter your password. Enter the same name and password as you are using for your desktop computer.

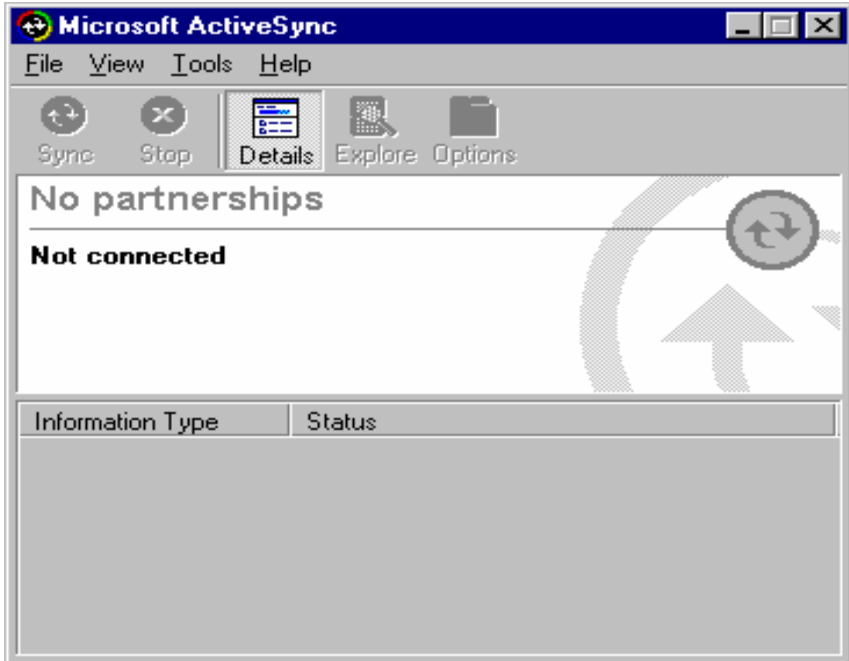
5. After establishing a connection, click **Explore** to open a window. Via this window you may then exchange files with the HiCOARM9_1 target platform using drag and drop. For this, drag the files onto the desired directory on the HiCOARM9_1.

Note: The folder contents on the HiCOARM9_1 platform are created from the Flash when Windows® CE is started. Files copied to these folders will not be available any more after Windows® CE is restarted. This could be the ..\FlashFX Disk or the ..\SD Card folder. These folders are available only if the corresponding devices are detected by the system. An exception are also all subfolders of the "Network" folder. These folders represent drives with which a connection was established via network.

6. To terminate the connection, click the  symbol on the Windows® CE's task bar. The connection may be separated in the dialog that appears.

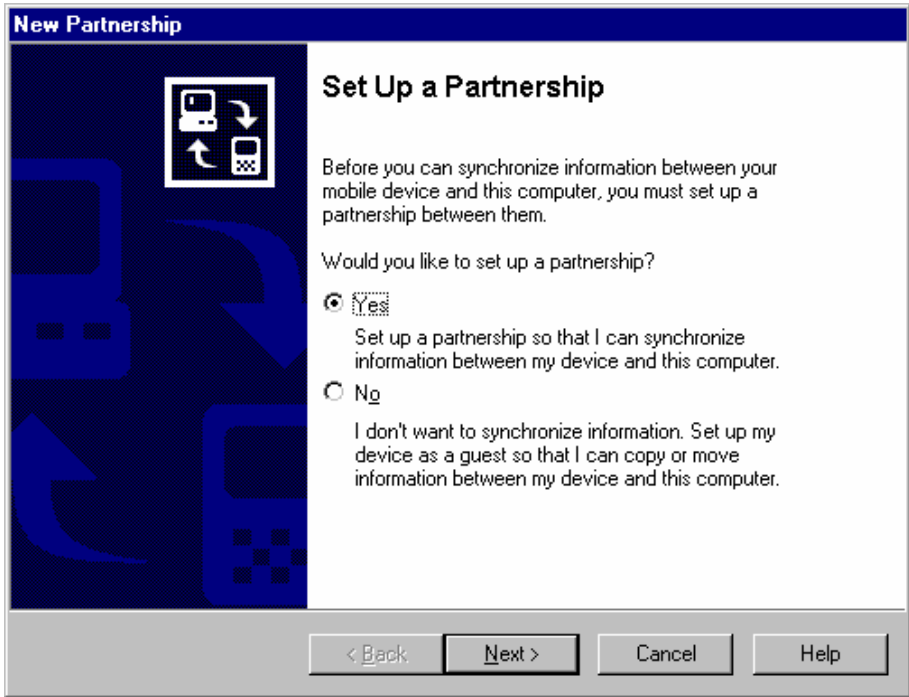
10.5. Remarks on Using ActiveSync

After installing ActiveSync, there is a "Microsoft ActiveSync" link in the "Programs" group of the Windows "Start" menu. When pointing to this link, the following window appears:



Please verify that the serial communication is enabled and the proper port has been selected (**Connection Settings** menu item of the **File** menu).

If a connection has been established for the first time after power on, the following dialog box displays:



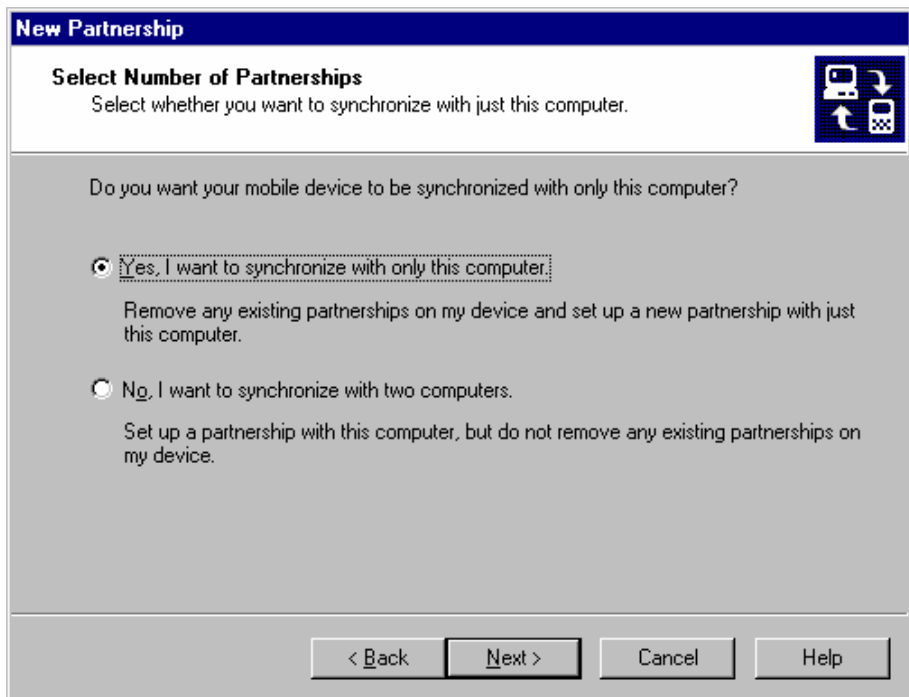
The window shows that no partnership has been established between the Windows® CE computer and the desktop PC. A partnership is required for network connections, or to synchronize data with Pocket Word or Pocket Inbox (Outlook).

Hint

If a partnership has already been created with a device with persistent registry, the dialogs displayed here do not show up (except for the dialog box on [p. 34](#)).

If you wish to use the serial or USB interface only, the selection made here is of no significance. If you wish to establish a connection via network, select "Yes". "No" means that the Windows® CE computer is connected with the desktop computer as a guest only. The connection as a guest is sufficient if you wish to exchange data or to debug an application via a serial or USB connection. Selecting the "Yes" radio button will establish a partnership, which is necessary for a

network connection. The following dialogs will appear:




This dialog specifies the number of desktop computers you wish to synchronize with the Windows® CE device. Accept the default setting and click "Next".

The following two dialogs let you select a Personal Information Manager. Accept the settings by clicking the "Next" button twice.

New Partnership

Select a Personal Information Manager
Select a program for synchronizing personal information.



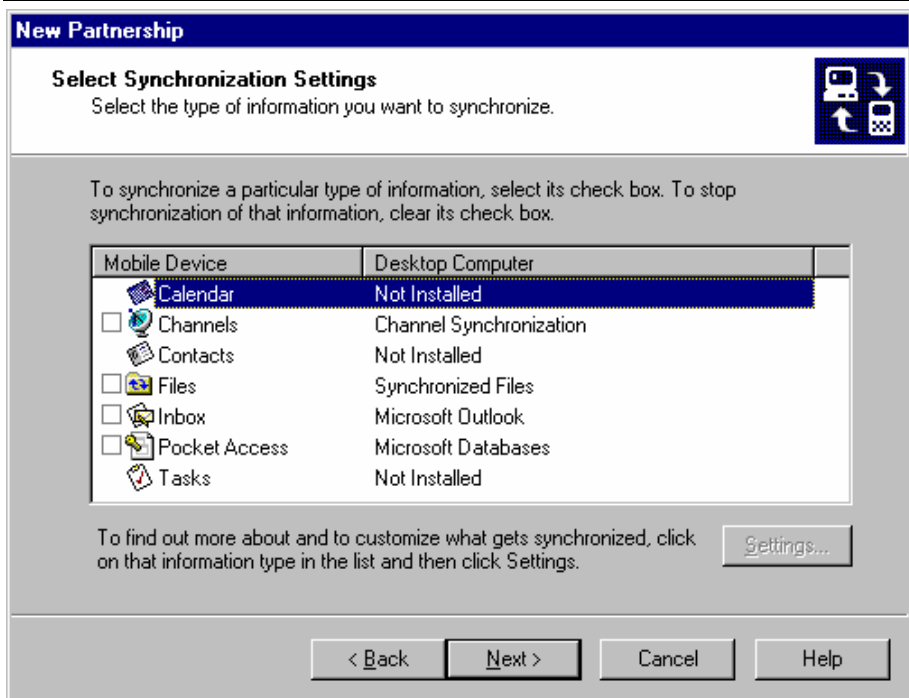
This computer has more than one personal information manager (PIM) that can be used to synchronize personal information on your mobile device. For each of the following mobile device programs, select the PIM you want to use for synchronization.

Calendar

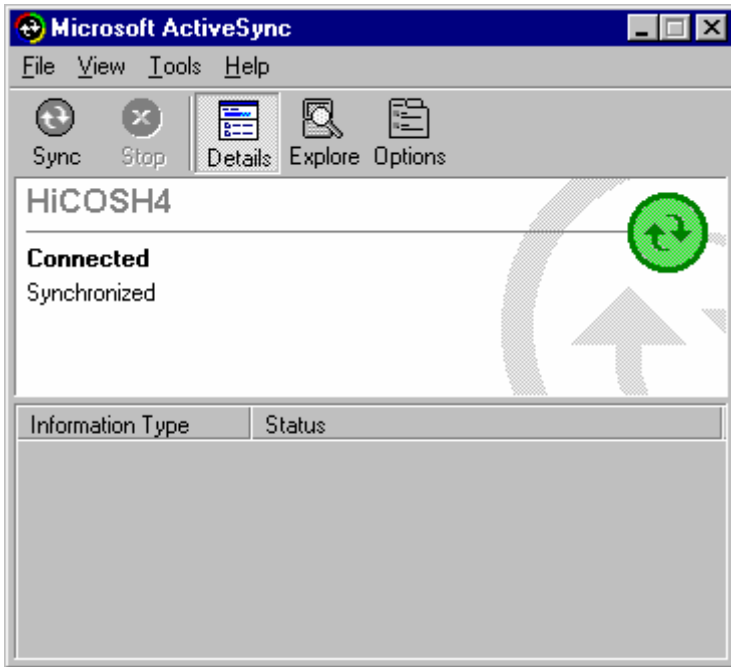
Contacts

Tasks

< Back Next > Cancel Help



After that, a dialog box appears telling you that the partnership has been successfully established. After clicking the **Finish** button, the ActiveSync program will open on your desktop:



The connection has been established.

Data transfer

Clicking the "Explore" button will bring up a window where you can search the Windows® CE computer.

The data exchange between the Windows® CE computer and the desktop computer can be initiated by dragging a file from the window to another window that does not belong to ActiveSync, or vice versa.

11. The Bootloader

The HiCOARM9_1 accommodates a bootloader, which is run after a reset. The bootloader serves to start (OS) software stored in the Flash, to reload and/or debug (OS) software, especially Windows® CE kernels.

The separate manual "Bootloader for HiCOARM9_1" describes the bootloader functions required for working with this Starter Kit. :

- Controlling the bootloader
- Supported file formats
- Memory allocation
- Bootloader functions that can be used by the (OS) software. These are, among others, the flash functions.

This separate manual is on the CD of the starter kit.

*For the professional Edition of the starter kit **only** :*

You will find instructions how to re-build the bootloader in the additional manual HiCOARM9_1STPro-CE500. This manual is on the CD of the starter kit professional.

12. Working with eMbedded Visual C++ 4.0

The Microsoft eMbedded Visual C++ 4.0 deliver a complete desktop development environment derived from the Visual C++ 6.0 development environment. Using the eMbedded Visual C++ 4.0, you are able to create Windows® CE-based applications and system components. eMbedded Visual C++ comes with remote tools, which are detailed in section "Remote Tools for Windows CE" on [p. 42](#).

Important note :

To develop applications for Windows CE 5.0 the Service Pack 4 is required. This service pack is also on the Platform Builder DVD

12.1. Using eMbedded Visual C++ 4.0

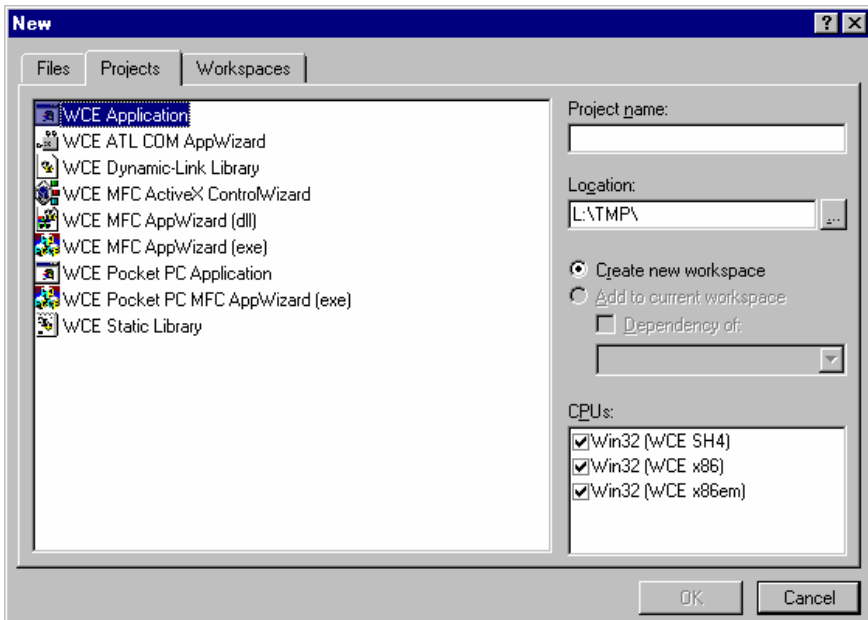
Using eMbedded Visual C++ 4.0, applications will be developed in the same manner as with Visual C++ 6. However, the following must be taken into account:

- Since Win32-API functions may be omitted in the creation of operating system kernels under Windows® CE, there is no SDK (Software Development Kit) which suits all Windows® CE operating system kernels. Actually, each operating system kernel has an SDK of its own which must be installed in addition to the development environment. If you have not installed the SDK during the installation of the Starter Kit, the corresponding setup program can be found in the SDKs subfolder of the Starter Kit installation (default folder: C:\Programs\Starterkit for Windows CE 5.0 with HiCOARM9_1). The name of the setup program for OS kernel 1 is HiCOARM9_1Skit_CE500.
- Please consult the online help (search for "Requirements") to obtain information on whether a specific function is available in an SDK. If the name of the OS kernel (e.g. HiCOARM9_1Skit_CE500 for Starter Kit kernel 1) is listed, the function will be supported by this OS kernel.

- The list of parameters of the Win32-API functions is identical with that of the corresponding API functions of Windows® 2000/XP. With some functions certain parameters must be set to 0. For more information, please refer to the online help.
- The MFC (Microsoft Foundation Class) for Windows® CE is a subset of the MFC for Windows® 2000/XP, that is, not all MFC classes are supported. More detailed information can be found in the MSDN library at:
<http://msdn.microsoft.com/library/wcedoc/wcemfc/guide.htm>

12.2. Creating an Application Project

Select **New...** from the **File** menu. The following dialog box appears:



Select the desired project type (WCE Application, WCE ATL COM AppWizard, etc.).

Enter a name for the new project in the **Project name** text box.

Select **Win32 (WCE ARMV4I)** from the list of **CPUs** and click the **OK** button. The corresponding wizard will be started.

Here, the project-specific settings can be made. After this, you may start to develop your application.

12.3. Debugging an Application

For debugging an application, Microsoft's ActiveSync 3.7 is used. Microsoft ActiveSync establishes a connection via network or the USB port. Whether this is done via network or the USB port depends on how ActiveSync has been installed. For more details, please refer to chapter "Data Exchange between the Target and the Desktop Computer" on [p. 25](#).

We recommend that you use a network connection, since large amount of data might be transferred via this connection.

Note

If there is no network, a 'network solution' can be established by connecting the target platform to your desktop computer via a twisted-pair cable.

12.3.1. Preparing the Debug Session

Before starting the debug session, the following steps will have to be performed **once**. In addition, a connection via Microsoft ActiveSync 3.7 must be established.

1. In the development environment select **Configure Platform Manager...** from the **Tools** menu.
2. In the following dialog select the name of your operating system kernel (e.g. HiCOARM9_1Skit_CE500 for the OS kernel 1).
3. Click the 'Plus' sign in front of the name and highlight the subentry **<NAME> (Default Device)**, where <NAME> denotes the name of the OS kernel.
4. Click the **Properties...** button.
5. In the following dialog box click **Microsoft ActiveSync** from the **Available Transport Components** list.
6. Click the **Advanced** button.
7. In the following dialog box select **Microsoft ActiveSync** from the **Available Server Components** list box.
8. Click on **OK**.
9. Click the **Test** button.
The communications line's functionality is being tested. The following message should appear: **Connection to device established**.
10. Close all dialog boxes by clicking the **OK** button.


12.3.2. Debugging

For the debugging of Windows® CE-based applications on the target platform, the application plus a client of the debugger will be copied to the target. To start the debug process, please proceed as shown below.

Hint

The symbol bar of your development environment provides a drop-down list box showing the name of the operating system kernel on which the application is to be debugged. That is, if the Starter Kit's OS kernel 1 runs on the target platform, HiCOARM9_1Skit_CE500 must appear in the drop-down list box.

12.3.3. Starting the Debugger

Click the  button on the symbol bar of your development environment.

The development environment checks whether the application to be debugged is up to date, that is, whether or not changes were made to the source code since the last session.

After that, CEMON.EXE, the client part of the debugger, will be searched for on the target platform. If it cannot be found, it will be copied to the target.

The .EXE file of your application together with the DLLs will also be copied to the target.

After that, your application is started and you may start the debug process.


Note on debugging

When a breakpoint takes effect, or after performing a single step, it will take some time until the debugger responds to your inputs, since the values of the variables shown in the debugger need to be transferred from the target to your desktop computer.

13. Remote Tools for Windows® CE

eMbedded Visual C++ 4.0 provides a set of remote tools, which can be very useful in developing your application.

- These tools can only be called from within the development environments. They can be found with the eMbedded Visual C++ 4.0 and with the Windows CE Platform Builder 5.0 in the **Tools** menu.
- After the corresponding tool is called, the main window and a **Select a Windows CE Device** dialog box will appear.

If the dialog box does not appear, click the  button in the symbol bar of the main window.

Click the 'Plus' sign in front of the name of the OS kernel you are using and highlight the subentry (e.g. **HiCOARM9_1Skit_CE500 (Default Device)** for the OS kernel 1).

Click the **OK** button. The selection of whether the connection is established serially or via network is the same as under eMbedded Visual C++ (see section "Debugging an Application" on [p. 39](#)).

Note

The **Configure Windows CE Platform Manager** menu item can be found in the **Connection** menu with the Remote Tools.

13.1. Remote Spy++

The Remote Spy++ tool displays the windows and the messages on a Windows® CE-based device.

13.2. Remote Registry Editor

Using the Remote Registry editor, the Windows® CE computer's registry can be read and modified.

Important note

By default, this tool provides the **My Computer** entry, which stands for the desktop computer. Changes made here have the same effect on the registry as when using the "regedit" tool.

Note

The Remote Registry editor does not call the Win32-API function RegFlushKey(). It is therefore necessary for writes to the persistent registry to run WriteReg.exe or another program that calls the RegFlushKey() function, after the Remote Registry editor has finished.

13.3. Remote Heap Walker

This tool displays data that were stored in the main memory by the individual applications at run time.

13.4. Remote Process Viewer

Remote Process Viewer displays the currently active processes (applications) and the number of threads belonging to a process including their priority.

All loaded DLLs are listed.

In addition, Remote Process Viewer allows you to "kill" any process.

13.5. Remote Zoomin

Remote Zoomin allows you to grab screen shots from a Windows® CE-based device. These can then be saved as Windows bitmap files on your desktop computer.

13.6. Remote File Viewer

Remote File Viewer views the file system on a Windows® CE-based device.

14. Starting the Application at the System Start

Your application may be installed on the HiCOARM9_1 target platform in such a way that it is automatically started at the next system start. To achieve this, the following must be performed:

- Every time the computer is (cold-)started, the folder structure is created a new from the Flash. This means that files that were copied to the Windows® CE computer will be lost after a restart.

This does not apply to files that were copied to mass storage folders, such as Secure Digital Memory Card, USB storage or flash file system drives. With the Starter Kit this affects one folder:

- Folder ...\\SD Card
- Folder ...\\Hard disk
- Folder ...\\FlashFX Disk (not supported in revision 1.1)

The fact that all files created during runtime will be lost after a restart means that a shortcut created in the ...\\Windows\\Autostart folder during runtime will not be available any more after Windows® CE has been restarted.

- This is also true for the registry, i.e., it will be created a new from the image file each time Windows® CE is started.

Changes made to the registry of a Windows® CE OS kernel will be kept. For details on the persistent registry, refer to section "Persistent Registry" on [p. 51](#).

14.1. OS Kernels with a Persistent Registry

To automatically start your application with operating system kernels that have a persistent registry, the following has to be done:

- Copy your application and all required DLLs to a directory which represents a mass storage (e.g. MMC Disk). Additionally required DLLs are those that are not built in the OS kernel.
- Make the following settings in the registry's **[HKEY_LOCAL_MACHINE\\init]** key:
 - Entry 1:
 - Name: LaunchXX, where XX stands for a two-digit number. This number must be between 51 and 99 with the operating system kernel belonging to this Starter Kit.

Type: REG_SZ

Value: File name of the file to be executed

- Entry 2:
Name: DependXX, where XX must be the **same** two-digit number as specified with entry 1.
Type: REG_BINARY
Value: Hexadecimal digits that specify the dependency on a previously started program.

Please also note section "Persistent Registry" on [p. 51](#).

- Save the registry by calling the RegFlushKey system function with any application.

Note

Entry 2 specifies the dependency of the application. An application stored on a mass storage cannot be started until the corresponding entry has been made in the folder structure. It may happen in this context that the registry entries in the [HKEY_LOCAL_MACHINE\init] key will be processed faster than the mass storages are entered in the folder structure. This must be taken into account in the start sequence.

14.2. Start with the Supplied OS Kernels

Note

The mechanism described in this chapter is only implemented in the kernel HiCOARM9_1Skit_CE500. This kernel is only provided with the Standard or the Professional Edition of the starter kit.

The supplied OS kernels include a link in the autostart folder that searches the registry for an application to be started. If an application is found, it will be started.

The program is configured by the registry.

The program uses the following registry entries:

The application searches for an entry "**CustomerApp**" in the registry key [HKEY_LOCAL_MACHINE\Software\emtrion\AutoStart]. This entry, of the type REG_SZ, contains the filename of the application to be started and possibly the path. The specified file is searched for in these directories (if available):

- Windows system directory
- SD Card directory (Secure Digital Memory Card).
- FlashFX Disk
- Hard Disk directory (represents a USB storage device)

The "AppParameter" entry can be found in the same registry key. This parameter, which is also a REG_SZ type, contains the list of parameters to be passed to the application.

Optionally, the same registry key may also contain an entry called "WaitCycles". This entry, a REG_DWORD type, lets you extend the time that the emtrion program is waiting for the directories to be linked, e.g. a CF disk. If the entry is not found, a default value of 10000 taken.

Optionally, the same registry key may also contain an entry called "FailureOptions". This entry is a REG_DWORD type and is bit-coded. The significance of the individual bits is shown in the table below:

Bit No.	Significance
0	If an error occurs, the following message will be displayed: "Cannot start the application which was specified to launch at startup."
1	If an error occurs, a detailed message will appear. In addition to the text displayed by bit no. 0, the message includes the name and search path of the program to be started. This bit is only analyzed if it is set to 0.
2	The error message is always displayed. That is, it displays again as soon as the message has been acknowledged. This bit is only analyzed if it is set to 0.
3	Explorer will be started in place of the application. This bit is of significance only if bit 2 is set to 0.
4..31	Reserved for future extensions

Note

Set bit 3 if you wish to start an application without previously opening Explorer.

15. Sample Application

IMPORTANT NOTE:

This sample uses the serial port which will be available in the next release of the starter kit. Currently you can use this application to see how a Windows CE application works.

15.1. In General

The demo application is a simple terminal program that transmits all user inputs via the selected serial port. At the same time, the data received will be displayed in the application's window.

The parameters used for communication (baud rate, number of data- and stop bits, parity, handshake) are displayed in the so-called 'Command bar' (=combination of menu bar and tool bar) (see Figure). These parameters can be set via the menu:



The sample application was created with the eMbedded Visual Tools 3.0 or eMbedded Visual C++ 3.0 and can be found on the Starter Kit CD in the ..\Starterkit\Samples\SampleApp folder. This application works also with the eMbedded Visual C++ 4.0.

15.2. How the Demo Application Works

After the application has been started, the window will be created and the global variables initialized. The application then determines (via registry entries) the ports available (COM0: to COM4:). The application is now ready for user inputs.

Select the desired port from the "Parameters" -> "Port" menu.

As soon as a port is selected, it will be opened by calling the **OpenPort** function. The parameters chosen and the timeout time will be set. When the port is opened, a receive thread is created that receives the characters from the selected port and stores them in the receive buffer. Whenever a character has been received, a message with the **WM_USER** ID is transmitted to the main thread (primary thread). These **WM_USER** messages have the application repaint your window and so display the characters received.

A character is directly transmitted from within the main thread if it receives a message of the type **WM_CHAR**. This message is always received by the operating system when any key is pressed. The message handler calls the **WritePort** function that converts the character to be transmitted into ASCII code and then transmits it.

15.3. Changing the Parameters via the Menu

You are able to alter the serial port and the communications parameters in the "Parameters" menu. Each time a parameter is changed, the main thread gets a message of the type **WM_COMMAND**. The sub ID transmitted to the **wParam** parameter causes the corresponding parameters in the DCB to be modified. After this, the **UpdateTheSettings** function is called. This function sets the corresponding check marks with the menu items and updates the parameter string ("User Settings:") displayed in the menu bar. **UpdateTheSettings** also calls the **UpdatePortSettings** function, in order to return the parameters of the port in use.

15.4. Changing the Port

The port can be changed any time via the "Parameters" menu. If another interface is selected, the open port is closed (by calling **ClosePort**). After this, the global variable is corrected with the port number, and the new port will be opened. When the new port is opened, it will be initialized with the same parameters as the last port used.

16. Persistent Registry

Originally, Microsoft's Windows® CE does not support a persistent registry. That is, all modifications made to the registry while Windows® CE is running will not be kept when Windows® CE is restarted.

To permanently keep these modifications, two functions are available which have to be implemented by the hardware manufacturer. These functions are already implemented in the HiCOARM9_1. They will be called by the operating system, in order to write the registry onto the Flash in the last 512 kBytes.

Note

Systems with a non-persistent registry have the advantage that a system cannot be damaged due to possibly wrong registry settings made while the system is running.

In order to have the registry write to the flash disk, the "RegFlushKey" function must be called by the application. However, this function should not be called after each modification to avoid performance bottlenecks. Call this function after having made several changes to the registry.

The starter kit's operating system kernels include a tool called WriteReg.exe. This tool calls the RegFlushKey function to make the desired changes. For example, it can be used for system modifications that are to be kept permanently.

A programming sample:

```
HKEY  hRegKey;
DWORD  retWert;

/* open the registry key */
retWert = RegOpenKeyEx(HKEY_LOCAL_MACHINE, _T("Software"), 0,
                      0, &hRegKey);
if (retWert != ERROR_SUCCESS)
{
    /* Error handling */
}

/*
:
:
Here new entries are made, entries changed or deleted.
:
:
*/

/* Save registry to Flash */
retWert = RegFlushKey ( hRegKey );
/* Hint: Here a valid handle for the registry key will need to
be specified (see the online help).
The complete registry will always be saved to Flash. */
if (retWert != ERROR_SUCCESS)
{
    /* Error handling */
}

/* Close registry key */
retWert = RegCloseKey( hRegKey );
if (retWert != ERROR_SUCCESS)
{
    /* Error handling */
}
```

16.1. Deleting the Persistent Registry

The persistent registry can be deleted in either of the following ways:

- 1.) Using the bootloader
- 2.) Via an application under Windows CE

Note

These methods should be applied only if the installed OS kernel makes use of, or at least supports a persistent registry. Otherwise, parts of the operating system can be damaged.

16.1.1. Deleting the Registry by means of the Bootloader

The persistent registry can be deleted via menu item 4 of the second bootloader menu. For more detailed information, please refer to the user manual of the bootloader.

16.1.2. Deleting the Registry by means of an Application under Windows CE

A Windows CE application can be enabled to use the persistent registry by calling the Win32-API function `KernelIoControl`. For this purpose, emtron has created the following device-specific I/O control code:

```
#define IOCTL_HAL_SETREGISTRYTODEFAULT
        CTL_CODE(FILE_DEVICE_HAL, 0x810, METHOD_BUFFERED,
FILE_ANY_ACCESS )
```

In this code, `CTL_CODE` is a macro, `FILE_DEVICE_HAL`, `METHOD_BUFFERED` and `FILE_ANY_ACCESS` are specified as `#define` statements. Both the macro and the definitions were defined by Microsoft in the `winiocctl.h` header file. This file is usually linked to applications with the `#include <windows.h>` statement.

The `KernelIoControl` function is not described in the online help of embedded Visual C++ 4. Therefore, an extract from the online help of the CE Platform Builder that describes this Win32-API function is printed at the end of this section.

IOCTL_HAL_SETREGISTRYTODEFAULT

This control code will delete the currently stored persistent registry. As a result, the default registry will be used when the system is started again. However, a reset will **not** be issued automatically. Such a reset can be performed by using the IOCTL_HAL_REBOOT control code.

IOCTL_HAL_SETREGISTRYTODEFAULT expects the following parameters of the KernelIoControl function:

lpInBuf, nInBufSize, lpOutBuf, nOutBufSize	These parameters will not be analyzed
lpBytesReturned	Pointer to a DWORD variable containing the number of bytes returned (here 0)

The control code deletes the flash sectors (where the registry is saved) by means of the erase function supplied by the bootloader.

Documentation of the KernelIoControl function (extract from the online help of Platform Builder):

This function provides the kernel with a generic I/O control for carrying out I/O operations.

```

BOOL KernelIoControl( DWORD dwIoControlCode,
LPVOID lpInBuf, DWORD nInBufSize,
LPVOID lpOutBuf, DWORD nOutBufSize,
LPDWORD lpBytesReturned );
    
```

Parameters

dwIoControlCode

I/O control code, which should support the OAL I/O controls. For a list of these I/O controls, see [Supported OAL APIs](#).

lpInBuf

Pointer to the input buffer.

nInBufSize

Size, in bytes, of *lpInBuf*.

lpOutBuf

Pointer to the output buffer.

nOutBufSize

Maximum number of bytes that can be returned in *lpOutBuf*.

lpBytesReturned

Address of a **DWORD** that receives the size, in bytes, of the data returned.

Return Values

TRUE indicates success; FALSE indicates failure.

Remarks

The kernel calls the OEMIoControl function when a device driver or application calls the kernel function **KernelloControl** and passes an I/O control code. This function is also called when the [SystemParametersInfo](#) function is called with SPI_GETOEMINFO or SPI_GETPLATFORMINFO. The system is fully preemptible when this function is called. The kernel does no processing, but it passes all parameters directly to the function supplied by you. This function is provided solely to allow your device driver or application to communicate with an OAL and its specific functionality.

17. Exchanging the Operating System Kernel

To exchange the OS kernel, the bootloader is required. A detailed description of the bootloader's functionality can be found in the separate bootloader manual on CD.

The required image files can be found on the CD in folder ...\\Starterkit\\Kernel. If you have selected an OS kernel during installation, this kernel can be found in the installation folder on your hard disk.

As communications partner for the download, the "KernelDownload" tool from emtrion can be used. A description of the tool can be found in chapter "The KernelDownload Tool" on [p. 57](#).

Important note

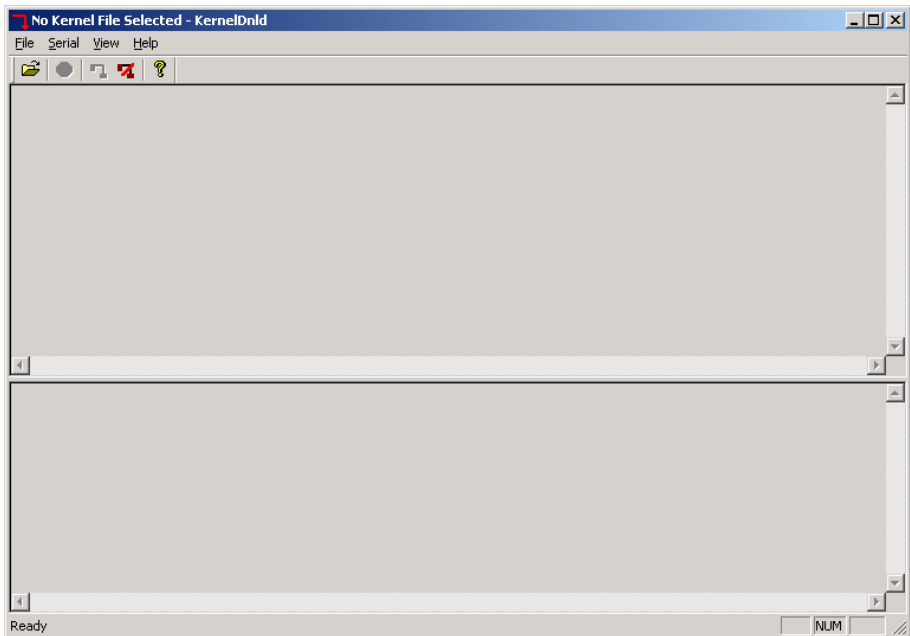
Whenever the operating system kernel is exchanged, the persistent registry should also be deleted by means of the bootloader. If the persistent registry is not deleted, the operating system may not start up correctly, or it may not start at all.

18. The "KernelDownload" Tool

To enable the user to download an image file via network without using Windows CE Platform Builder, emtrion has developed the "KernelDownload" tool.

18.1. Description

The main window consists of two parts. The upper part shows the progress made, while the lower part displays the output sent by the bootloader and received by the program:



The tool awaits a Bootme message at port 980. This message is broadcast by the bootloader when the kernel is to be loaded via network. When the program receives a Bootme message, it checks whether it has to respond to the Bootme message (for details, see section "Configuring the Download Tool" on [p. 60](#)). If it is to respond to the message, it transfers the selected file to the desired device. After the download is finished, the tool is waiting for the next Bootme message.






If a serial port has been successfully opened, the characters received via this port will be automatically displayed in the lower part. When the lower part has been selected via mouse, the characters entered with the keyboard will be transmitted via the serial port.

18.2. Operating Elements

18.2.1. Menus

Menu	Menu item	Comments
File	– Stop download	Terminates the current download process.
	– Select...	Selects the image file to be downloaded.
	– Exit	Exits the program.
Serial	– Select Port...	Selects the port via which the bootloader is controlled.
	– Connect	Opens the selected port for communication with the bootloader.
	– Disconnect	Closes the port previously opened for communication with the bootloader.
View	– Toolbar	Toggles the toolbar on/off.
	– Status bar	Toggles the status line on/off.
	– Split	Changes the sizes of the two windows.
	– Font	Sets the font and font size of the texts displayed by the bootloader.
Help	– About KernelDnld	Provides information on the version number of the download tool.

18.2.2. Toolbar

Symbol	Name of Button	Function
	"Select..."	Allows the selection of the image file to be downloaded.
	"Stop download"	Stops the current download.
	"Connect"	Opens the selected port for communication with the bootloader.
	"Disconnect"	Closes the port previously opened for communication with the bootloader.
	"Infodialog"	Provides information on the version number of the download tool.

18.2.3. Configuring the Download Tool

The download tool is configured in the Dnload.ini file. This file is read once at the start of the download tool. The .INI file contains the following default entries:

- Default port for the communication with the bootloader, e.g. COM1.
- Name of the devices served by the download tool. This setting is to be entered in the sections [Devices] and [RefuseDevices]. Each of these sections may contain one or more entries. An entry in the [Devices] section specifies the comparison constant with which the device name must begin. Only in this way, the corresponding device can be served by the download tool. An entry in the [RefuseDevices] section specifies the beginning of a device's name with which the device **must not** begin.

Example:

The sections in the Dnload.ini file look as follows:

```
[Devices]
DevNo=1
DevName1=HiCOARM9_1_

[RefuseDevices]
DevNo=2
DevName1=HiCOARM9_1_489
DevName2=HiCOARM9_1_532
```

So the download tool will serve all devices whose names start with "HiCOARM9_1_". This does not refer to devices whose names begin with "HiCOARM9_1_489" and "HiCOARM9_1_532".

19. emtrion Tools for Windows CE

In order to do various settings or get information, emtrion provides a variety of tools running on the Windows CE computer. All these tools can be found here:

Start Menu -> Programs -> emtrion Tools.

19.1. Writing the Persistent Registry

In order to enable the user to selectively write to the registry, emtrion has implemented a tool called WriteReg in the operating system kernel. When started it writes the persistent registry by calling the Win32-API function RegFlushKey().

Note

User inputs will be of no significance while the persistent registry is being written.

19.2. Determining the Version of the Operating System Kernel

The RevisionInfo tool implemented in the operating system kernel helps the user determine the currently running operating system kernel.

The following information are displayed in a dialog box:

- Name of the operating system kernel, e.g. HiCOARM9_1Skit_CE500
- Version number of the operating system kernel, e.g. 1.0
- Release date of the operating system kernel, e.g. 2006-05-17, 12:53:52

20. Modifying the Kernel, BSP and Bootloader

The standard edition of this starter kit includes the project workspace of the included kernel(s). It also includes the BSP with a binary version of all components. With this you can rebuild the delivered kernels, but you cannot modify the drivers.

The professional edition also includes the source code of the Board Support Package and the source code of the Bootloader. This version of the starter kit enables you to modify the bootloader, the hardware adaptation layer and/or the drivers.

More details on how you can do this can be found in the separate manual HiCOARM9_1STPro-CE500. You will find this manual on the CD of the professional version of the starter kit in the folder \Starterkit\Manual.

NOTE

emtrion also offers to make a customized operating system kernel for you.

21. FAQs

This section provides FAQs on the following:

- In General
- Debugging an Application Using eMbedded Visual C++

The FAQs presented here were up to date at the time this manual was being printed. Since they are continuously updated, you may wish to check the web at: <http://www.emtrion.de/support/index.html> for the latest FAQ on a variety of products. If you cannot find your product, or if the information provided is not sufficient, contact our Support Team (see also section "Support" on [p. 15](#) for details).

21.1. In General

- **How to determine the MAC address (=unique identification of each node in a network) of the HiCOARM9_1?**

The MAC address can be determined as follows:

- Activate the bootloader by setting the jumper to position 1-2 and then switching on or resetting the device. The bootloader menu appears.
- Select menu item 4 or 5 by clicking the corresponding key. The bootloader now tries to establish a connection via network. At the same time, several status messages are output. The following line of the messages includes the MAC address in the hexadecimal format:

```
Ethernet Address: 00:30:6C:90:00:10
```

- **How to establish a network connection between a CE computer and a Windows® NT / 2000 / XP computer?**

There are two methods to do this:

Method 1

1. Open Internet Explorer on the Windows® CE computer.
2. Example:
The Programme directory is to be mapped. It is located on drive C: on a computer named NTRechner. The drive C: has the share name LW_C. In this case, enter [\\NTRechner\LW_C\Programme](#).
3. If the network is running, the contents of that drive appear.

Method 2

1. Open a prompt on the CE computer.
2. Enter the following command line:

```
net use <localname> <remotename> /user:<uname> /password:<pwd>
```

The command line's parameters have the following meaning:

- <localname> : Name with which the mapped subdirectory appears in the \Network directory.
Example: If Workstation is entered for <localname>, a workstation directory appears under Windows® CE.
- <remotename>: Network path in accordance with the UNC naming convention for the drive to be mapped.
Example: The \Programme directory is to be mapped. It is located on drive C: on a network computer named NTRechner. Drive C: has the share name LW_C. The entry for <remotename> is \\NTRechner\LW_C\Programme.
- <uname>, <pwd>: User name (Login) and password on the computer connected to the network. Enter the same name and password as you are using for your desktop computer.
3. In the \Network directory on the Windows® CE computer, there is now a new directory with the specified name. This directory represents the corresponding directory on the NT computer.

21.2. Debugging an Application Using eMbedded Visual C++

- The following message appears on the screen: "Auto Download Failed: Insufficient memory in object store to download to target device"

Dividing the RAM into a data- and program memory is not sufficient for your application. Open the System applet of Control Panel. Select the Main Memory tab. Move the slider to the right. This will increase the data memory. After this, you should be able to download your application.

- **The application debugging via network in a Peer-to-Peer network without server does not work.**

By default, the operating system kernels of the Starter Kit are configured in such a way that the settings (IP address, subnet mask, gateway) are taken from a DHCP server.

Change these settings as follows:

1. Open the "Network and Dial-up Connections" applet in Control Panel.
2. Select the network card "AT91ETH1".
3. Press the right mouse button and select "Properties" in the context menu.
4. Select the "IP Address" tab.
5. Select "Specify an IP address"
6. Enter an IP address and the related subnet mask. Enter the IP address of your desktop computer as default gateway.
7. Close the dialog box with OK.
8. Save the persistent registry.
9. Press the reset button on the HiCOARM9_1 board.
10. Check the network connection as follows:
 - Open Internet Explorer
 - Enter the following address: [\\<name of your desktop computer>\<share name of one of your drives>](#)

Example:

Name of the desktop computer: NTRechner

Share name of your C: drive: C\$

Address to be entered: \\NTRechner\C\$

The name of the desktop computer can be obtained by entering "IPCONFIG /ALL" in the DOS prompt of the desktop computer.

22. Licence Agreements

22.1. Software Licence Agreement emtrion GmbH

emtrion GmbH (in the following named emtrion), provides the guarantee, and the licensee, according to the following terms, accepts a non-exclusive licence or sublicense to utilize and reproduce software in binary format, developed by emtrion or a third party and distributed by emtrion.

Conditions of Use

By agreeing to be bound by the terms of this agreement, the user is allowed to use the programs ordered. The licensee is not allowed to utilize the programs on more than one system. If a program is to be used on more than one system, the corresponding number of software licences must be ordered. It is multiple usage and thus illegal as well, if the program ordered is used by several workstations on a network.

Only one backup copy of each program ordered may be made for safety purposes. The licensee is prohibited from making further-reaching usage of the program than explained above. It is particularly inadmissible to transmit the programs electronically via dataline from one computer to another.

The licensee may not reverse engineer, decompile or disassemble the software product in any form without prior written consent of emtrion, nor is the purchaser allowed to transfer all the rights under this Software Licence Agreement to a third party.

If the emtrion product can only be used in conjunction with a licenced software product of a third party, the licensee also agrees to the terms of the respective licence agreement.

Copyright

Each single program is copyrighted. The licensee may take a backup copy of the program for safety purposes. Each copy is to be affixed with a copyright note.

Each licensed program and any part of it are trademarks of emtrion or the licensor, respectively. The usage of a software product is subject to all conditions and agreements stated in this Software License Agreement, at all times.

Period of Validity and Expiry of the Licence Agreement

The period of validity of both this Software Licence Agreement and the licences granted is valid as long as it will be terminated by the licensee. This Software Licence Agreement shall automatically terminate if the licensee fails to comply with its terms and conditions. On the expiry of the licence agreement, the licensee returns the original of a software program to emtrion or at least confirms that he will destroy the original as well as all other copies that fall under this Software Licence Agreement.

Guarantee

emtrion gives the guarantee that there is no defect in the material or fabrication default with regard to the data medium of the program. emtrion makes the warranty that, within 90 days from the date of receipt, the programs ordered do basically work according to the documentation enclosed.

Limitation of Liability

emtrion is not liable for any profit losses, business interruption, loss of business information or any other pecuniary loss. emtrion is not liable for indirect, particular, accidental or any sort of damage either, neither with respect to this licence agreement nor otherwise. emtrion makes no warranty of any kind with regard to this material, including but not limited to, the implied warranties of merchantability and fitness for a particular purpose.

Governing Law and General Provisions

Validity and implementation of this agreement are subject to the jurisdiction of Germany. If any part of this Licence Agreement is found void and unenforceable, it will not affect the validity of the balance of the Agreement, which shall remain valid and enforceable according to its terms. Place of jurisdiction is Karlsruhe.

22.2. Licence Agreement Microsoft

You have acquired a device ("HiCO") that includes software licensed by emtrion GmbH from Microsoft Licensing Inc. or its affiliates ("MS"). Those installed software products of MS origin, as well as associated media, printed materials, and "online" or electronic documentation ("SOFTWARE") are protected by international intellectual property laws and treaties. The SOFTWARE is licensed, not sold. All rights reserved.

IF YOU DO NOT AGREE TO THIS END USER LICENSE AGREEMENT ("EULA"), DO NOT USE THE DEVICE OR COPY THE SOFTWARE. INSTEAD, PROMPTLY CONTACT EMTRION GMBH FOR INSTRUCTIONS ON RETURN OF THE UNUSED DEVICE(S) FOR A REFUND. **ANY USE OF THE SOFTWARE, INCLUDING BUT NOT LIMITED TO USE ON THE DEVICE, WILL CONSTITUTE YOUR AGREEMENT TO THIS EULA (OR RATIFICATION OF ANY PREVIOUS CONSENT).**

GRANT OF SOFTWARE LICENSE. This EULA grants you the following license:

You may use the SOFTWARE only on the DEVICE.

NOT FAULT TOLERANT. THE SOFTWARE IS NOT FAULT TOLERANT. EMTRION GMBH HAS INDEPENDENTLY DETERMINED HOW TO USE THE SOFTWARE IN THE DEVICE, AND MS HAS RELIED UPON EMTRION GMBH TO CONDUCT SUFFICIENT TESTING TO DETERMINE THAT THE SOFTWARE IS SUITABLE FOR SUCH USE.

NO WARRANTIES FOR THE SOFTWARE. THE SOFTWARE IS **provided "AS IS" and with all faults. THE ENTIRE RISK AS TO SATISFACTORY QUALITY, PERFORMANCE, ACCURACY, AND EFFORT (INCLUDING LACK OF NEGLIGENCE) IS WITH YOU. ALSO, THERE IS NO WARRANTY AGAINST INTERFERENCE WITH YOUR ENJOYMENT OF THE SOFTWARE OR AGAINST INFRINGEMENT.** IF YOU HAVE RECEIVED ANY WARRANTIES REGARDING THE DEVICE OR THE SOFTWARE, THOSE WARRANTIES DO NOT ORIGINATE FROM, AND ARE NOT BINDING ON, MS.

Note on Java Support. The SOFTWARE may contain support for programs written in Java. Java technology is not fault tolerant and is not designed, manufactured, or intended for use or resale as online control equipment in hazardous environments

requiring fail-safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines, or weapons systems, in which the failure of Java technology could lead directly to death, personal injury, or severe physical or environmental damage. Sun Microsystems, Inc. has contractually obligated MS to make this disclaimer.

No Liability for Certain Damages. EXCEPT AS PROHIBITED BY LAW, MS SHALL HAVE NO LIABILITY FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL OR INCIDENTAL DAMAGES ARISING FROM OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE SOFTWARE. THIS LIMITATION SHALL APPLY EVEN IF ANY REMEDY FAILS OF ITS ESSENTIAL PURPOSE. IN NO EVENT SHALL MS BE LIABLE FOR ANY AMOUNT IN EXCESS OF U.S. TWO HUNDRED FIFTY DOLLARS (US.\$250.00).

Limitations on Reverse Engineering, Decompilation, and Disassembly. You may not reverse engineer, decompile, or disassemble the SOFTWARE, except and only to the extent that such activity is expressly permitted by applicable law notwithstanding this limitation.

SOFTWARE TRANSFER ALLOWED BUT WITH RESTRICTIONS. You may permanently transfer rights under this EULA only as part of a permanent sale or transfer of the DEVICE, and only if the recipient agrees to this EULA. If the SOFTWARE is an upgrade, any transfer must also include all prior versions of the SOFTWARE.

EXPORT RESTRICTIONS. You acknowledge that SOFTWARE is of US-origin. You agree to comply with all applicable international and national laws that apply to the SOFTWARE, including the U.S. Export Administration Regulations, as well as end-user, end-use and country destination restrictions issued by U.S. and other governments. For additional information on exporting the SOFTWARE, see <http://www.microsoft.com/exporting/>.

INDEX

- ActiveSync 13, 26, 30
 - debugging an application 40
 - description 30
 - Installation on the desktop computer 26
 - network connection 28
 - setting up a partnership 31
 - USB connection 27
- Applet 13
- Application
 - automatic start 46
- Autostart 46
- Board Support Package 17
- Bootloader 36, 57
- C++ *see* eMbedded Visual C++
- Connector
 - Location of the connectors 23
- Data exchange
 - desktop computer and WinCE computer 26
- Debugging
 - with eMbedded Visual C++ 40
- Definitions 13
- DHCP 66
 - Windows CE 66
- Downloadtool 58
- Drivers
 - in supplied OS kernels 18
- eMbedded Visual C++ 37
 - creating a new project 38
 - debugging an application 40
 - FAQ 66
 - function of the SDKs 20
- eMbedded Visual Tools 37
- emtrion Tools 62
- ESD 22
- Ethernet address
 - determining ~ 64
- Ethernet connection cable 9, 10
- FAQs 64
- eMbedded Visual C++ 66
 - In General 64
- Glossary *see* Definitions
- Installation 22
- IP address
 - in Windows CE 66
- MAC address
 - determining ~ 64
- MFC 38
- Modify the Kernel, BSP and Bootloader 63
- Network cable 9, 10
- Network drive
 - net use 65
- New project
 - creating
 - in eMbedded Visual C++ 38
- Operating System Kernel
 - determining the version 62
- Operating system kernels
 - exchanging ~ 57
 - functional scope 18
 - supplied ~ 17, 18
- Persistent registry 13, 46, 52
 - in the supplied operating system kernels 19
 - RegFlushKey 52
 - WriteReg Tool 62
 - writing 62
- Platform Manager 41
- Platform Builder
 - supplements 20
- RegFlushKey 52, 53
- Registry
 - permanent ~ 52
 - persistent ~ 52
 - Remote Registry Editor 44
- Remote File Viewer 45
- Remote Heap Walker 44
- Remote Process Viewer 44

Remote Registry Editor	44
Remote Spy++	43
Remote Tools	43
Remote Zoomin	45
repllog	27, 28
Revision info	62
Screen resolutions	
available ~	19
SDK	see Software Development Kit
Software Development Kit (SDK)	13, 20, 25, 37
Starter Kit components	17
Starting the application	
automatically at system startup	46
Support	16
System requirements	12
Target platform	13
Tools	62
eMbedded Visual Tools	37
KernelDownload	58
revision info	62
WriteReg	62
writing the registry	62
Version	
of the operating system kernel	62
Visual C++	see eMbedded Visual C++