DIMM-MX53 Developer Kit for Windows Embedded Compact 7

User manual

Rev5 / 21.01.2015

emtrion GmbH

Revision: **5 / 21.01.2015**

Rev	Date/Signature	Changes
1	14.06.2011/Bi	--
2	09.03.2012/Bi	Updated sections about display driver with DVI suppot Added section for Camera Interface Added section for CAN Interface
3	07.09.2012/Ft	Updated sections to reflect new parameters added in kernel image v201: Added the description of the optional parameter –headless in the chapters of the tools Writereg and RevisionInformation
4	20.02.2013/Ft	Added/Updated sections to reflect the Multi-Touch feature added to kernel image v300 (Multi-touch in conjunction with display adapter ET0700M06)
5	21.01.2015/Ml	Revised chapter NAND due to supporting ONFI New chapter 5

This document is published by:

emtrion GmbH
Alter Schlachthof 45
D-76131 Karlsruhe
Germany

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

1 Contents

2 Definitions

The table below lists some definitions of terms in this manual.

ActiveSync	An application from Microsoft used for the exchange of data between the Windows ® Embedded Compact device and a desktop computer.
Applet	Application of the Control Panel which serves to control the system's properties.
Operating system kernel (OS kernel)	Complete Windows ® Embedded CE operation system running on the target platform. During the creation of the Windows ® Embedded CE OS kernel, modules and components may be omitted. This will create different OS kernels, which can each be found in an image file.
OSDesign	Project overview about all of the software components included in the kernel.
Desktop computer, Host	PC which fulfills the requirements according System Requirements (chapter 7.1)
Image, WINCE-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 ® Embedded 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 where Windows ® Embedded CE is running.
DIMM-MX53	The target platform

3 Terms and Conditions of License

The DIMM-MX53 Developer Kit for Windows ® Embedded Compact (also called Windows® CE in this document) comes with a runtime license 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.

The License Agreements to the software from Emtrion and Microsoft are stated in section License Agreements.

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

4 Introduction

The Developer Kit has been created to allow a simple and fast start-up of application development or building kernels with Windows ® Embedded Compact 7 hardware platform.

Furthermore, it offers the possibility to reduce the time to market phase for software basing Windows ® Embedded Compact 7 on DIMM-MX53.

For developing applications, the Developer Kit contains a preconfigured operating system kernel. This kernel provides a wide range of components and functions of Microsoft's popular Windows ® Embedded CE Win32 APIs and has a size of about 50MB, which allows it to be used for a wide range of applications.

The preconfigured kernels allows to write applications without creating own operating system kernels.

A detailed description of the operating system kernels can be found in the section Features of the Developer Kit.

For building kernels, the Developer Kit contains all the necessary software components that are needed for build kernels on DIMM-MX53.

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

5 OS Revision

This document refers to the functionality of the WINCE-Image with at least revision V03.05. The difference to the previous is in supporting the ONFI-Interface of NAND flashes.

The implementation of the feature ONFI also effects the supported module revisions. Theses revisions have to be greater or equal r3a.

For a less module revision you have to contact emtrion.

6 The Bootloader

This section gives a briefly description of the bootloader used in this Developer Kit. When you are more interested in the function scope of the bootloader, please refer to the bootloader manual. The bootloader manual is available on the DVD or on the emtrion support pages http://www.support.emtrion.de.

The main task of the bootloader is to download a WINCE-Image to the target and starting it. To achieve this work the bootloader can be services through a terminal running on a host. To do so, the host has to be connected to the serial port UART A at the target.

6.1 Default startup of the bootloader

The bootloader is located in the lower address area of the NAND-Flash and starts running at power ON or a reset. The bootloader is evaluating the environment variable “bootcmd”, which is set to the command “bootx wce flash” in the Developer Kit setup. This command boots the installed Windows CE Image from flash.

The bootloader waits for a few seconds before booting Windows CE. If you press a key, you can enter the bootloader console prompt to change bootloader settings or load a new image via tftp.

6.2 Communication settings

The bootloader’s communication settings are:

Baudrate	115200 bps
Data bits	8
Stop bits	1
Parity	none
Handshake	none

6.3 Dip switch setup

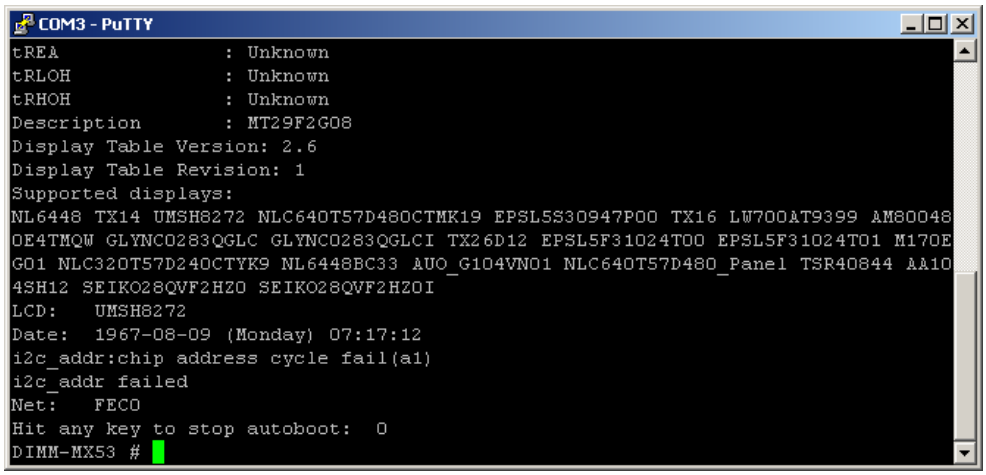
The DIMM-MX53 module carries two dip switches, which have to be setup as follows for a successful start up of the bootloader.

DIP Switch setting for successful start up:

	2	1
off	X	X
on		

6.4 Bootloader prompt

The bootloader prompt is reached if you press a key in the console window when the boot delay is counted down. The bootloader prompt allows you to change settings of the bootloader , to update the Windows CE image in flash or boot directly via tftp.



A detailed description of the bootloader can be found in the bootloader manual. You can find it on the Developer Kit DVD or on our support sites in the web: http://www.support.emtrion.de . Here we just show you some basic commands to update your Windows CE Image.

6.4.1 Print/Change environment variables

The environment variables are handled by using 3 commands: *printenv, setenv, saveenv*. *Printenv* shows you the current setting of all environment variables. *Setenv <variable> <value>* changes the value of an environment variable. This change is only in RAM and will be lost after reset. The changes can be made permanent by using *saveenv.* The following example shows how the boot command is set up.

```
DIMM-MX53 # setenv bootcmd bootx wce flash
DIMM-MX53 # saveenv
```

6.4.2 Network setup

The network setup of the bootloader is also handled by environment variables:

ipaddr	IP address of the device. Only effective if dhcp is deactivated
dhcp	Set “y”/”n” to enable/disable fetching an ip address from a DHCP server
serverip	IP address of the host PC which acts as TFTP server
netmask	Subnet mask of the device

To test your network setting you can ping the host PC from the device running the bootloader. To do so use the command *ping <ip address>*. **Please note, that the device running the bootloader can not be pinged**.

6.4.3 Update Bootloader or Windows CE image

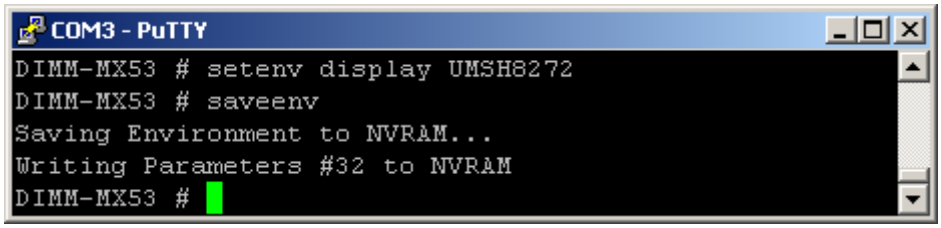
If the network connection to a tftp server is established, the bootloader in flash can be updated using this command: *update uboot tftp <uboot-image-name>*

Windows CE can be updated like this: *update wce tftp <wince-image-name>*

6.4.4 Changing the display

The kernel supports some different displays. To offer a wide flexibility the used display has to be selected in the bootloader. For this purpose the bootloader holds the environment variable "display". Set the variable to the corresponding name of the used display and then save the environment variable. However, the environment variable is already set to the connected display by delivery. Also many of our supported displays can be detected automatically. To activate auto detection you have to reset the environment variable by using: *setenv display*

The following sample shows the setting for the display UMSH8272 on the bootloader output. DIMM-MX53 # represents the prompt of the bootloader.




```
Remark:
You can get a list of the supported displays when you set the environment
variable display to the value "dummy", save the environment variables
using the command "saveenv" and restart the device. Each time if the
environment variable display is set to a value which is not existing in
the display table the bootloader prints the available displays.
```

6.4.5 Reset Windows CE Registry

The hive-based registry of Windows CE can be reset to its default values by setting the environment variable *erase_hive_registry* to "*y*" before starting the boot process of Windows CE.

Attention: If you set this variable permanently to "y", the registry will always be reset at startup.

6.5 Supported File Formats

The bootloader requires a special file. It contains raw binary such as in the file nk.nb0, extended by a special header. The special header contains all the information needed by the bootloader, like length, start address etc.

7 Requirements

7.1 Requirements of the host

The requirements are dependent on whether you want to develop applications only or build also kernels. The minimal requirements on the host are listed in the following subchapters.

7.2 SW-Requirements

- Operating system
    - Windows® XP SP3 or later
    - Windows® Vista with SP2
    - Windows® 7 with any available service packs
- Microsoft ActiveSync Version 4.2 for development purpose and data exchange between target and host
- Microsoft Visual Studio 2008. A 180-Day Trial Version is delivery by the Developer Kit.
- When you want to update the image a TFTP-Server is required too. The TFTP-Server is used for downloading of the WINCE-Image to the target. A TFTP-Server is not part of the Developer Kit, but can be downloaded for free for example at http://tftpd32.jounin.net/tftpd32_download.html. When you are using this tftp server, be sure its DHCP service is disabled to avoid conflicts with the current DHCP server at you network. We added the automatic creation of a U-Boot compatible kernel image to the Windows EC build process. This image is created in "C:\WINCE700\OSDesigns\DIMM_MX53DevKit_CE700\DIMM_MX53DevKit_CE700\misc". Just put the executable of the tftp server in this directory.

For building kernels you additionally require

- Windows Embedded Embedded Compact Platform Builder. A 180-Day Trial Version is also delivered with the Developer Kit.

7.3 HW-Requirements

- CPU-Speed: 2,4 GHz or higher
- RAM: 1024 MB, recommended 4 GB
- Interfaces
    - 1x Ethernet port
    - 1x RS232 (serial port)
    - 1x USB 1.1 or 2.0
- DVD-ROM drive
- If you want to create your own OS images, up to 100GB free hard disk space for
    - Installation of BSP of the Developer Kit
    - Platform Builder Development environment (Microsoft Windows® Embedded CE 6 R3) if not installed already

7.4 Requirements of the target

For the target the following devices are recommended:

- USB keyboard
- USB mouse

8 Host Setup

At this point we assume that the trial or full version of Microsoft® Visual Studio® 2008 with SP1 and Microsoft® ActiveSync (Mobile Device Center on Windows Vista and Windows 7))are already installed on the host.

8.1 Installing the SDK for application development

The SDK allows you to develop applications based on the delivered kernel. To make this possible install the SDK when VS2008 is not open. You can install it from the DVD. During the installation, the SDK is added to the list "Installed SDKs" from VS2008. At this point DIMM-MX53 can be choosen in VS2008 for generating applications on the DIMM-MX53.

8.2 Installing the Platform Builder for kernel development

```
Note:
This step is only necessary when you want to build kernels.
```

When you install Microsoft® Windows Embedded Compact 7 it is recommended to install only the features for the used CPU. For this Developer Kit you need the CPU type ARMV7.

First install the trial or full version of Microsoft® Windows Embedded Compact 7 on the host. Follow the instructions displayed by the installer of Microsoft Windows Embedded Compact 7. Platform Builder 7.0, which is the development environment, is also installed when Microsoft Windows Embedded EC 7.0 is installed. The Platform Builder 7.0 is a Plug-In for Microsoft Visual Studio 2008.

8.3 Setting up the Ethernet (TCP/IP)

TCP/IP is used for Ethernet connection to DIMM-MX53. Make sure that TCP/IP is already setup for the network communication at the host.

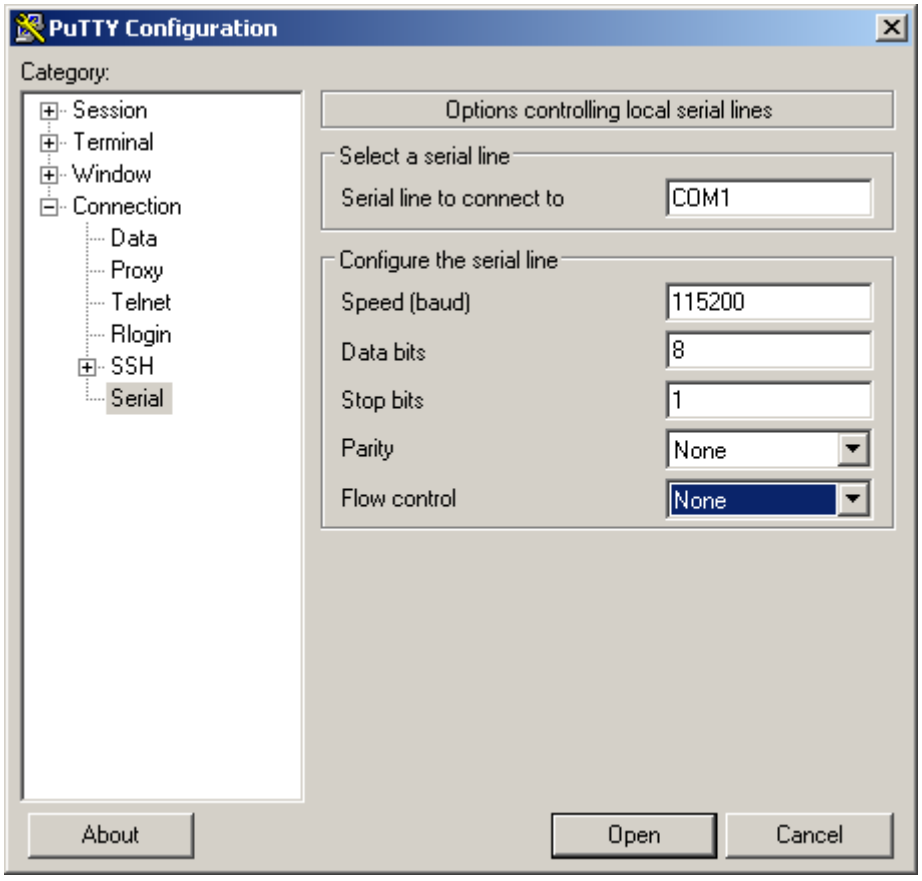
8.4 Setting up a TFTP server

A TFTP server is required to download or update the image on the target. Make sure the TFTP is running and its search directory points to the location of the image. A link for a TFTP server is stated in the section "SW-Requirements".

8.5 Setting up a terminal

As terminal software, use the terminal software that comes with your host or any software you are already using. In the terminal setting window, set the communication parameters as follows:

Baud rate	115200 bps
Data length	8 bits
Partiy	none
Stop bit	1
Flow control	none



Use the supplied serial null-modem cable to connect serial port COM1 (UART A) on the target to the host.

9 Setting up the target

This Developer Kit is delivered with the core module DIMM-MX53 and dependent on the scope of the delivery with one of the available emtrion DIMM-base boards. To make a fast operation possible the target is already assembled. For developing purpose only a few steps are necessary.

```
Please note that everything is discharged when touching the target.
```

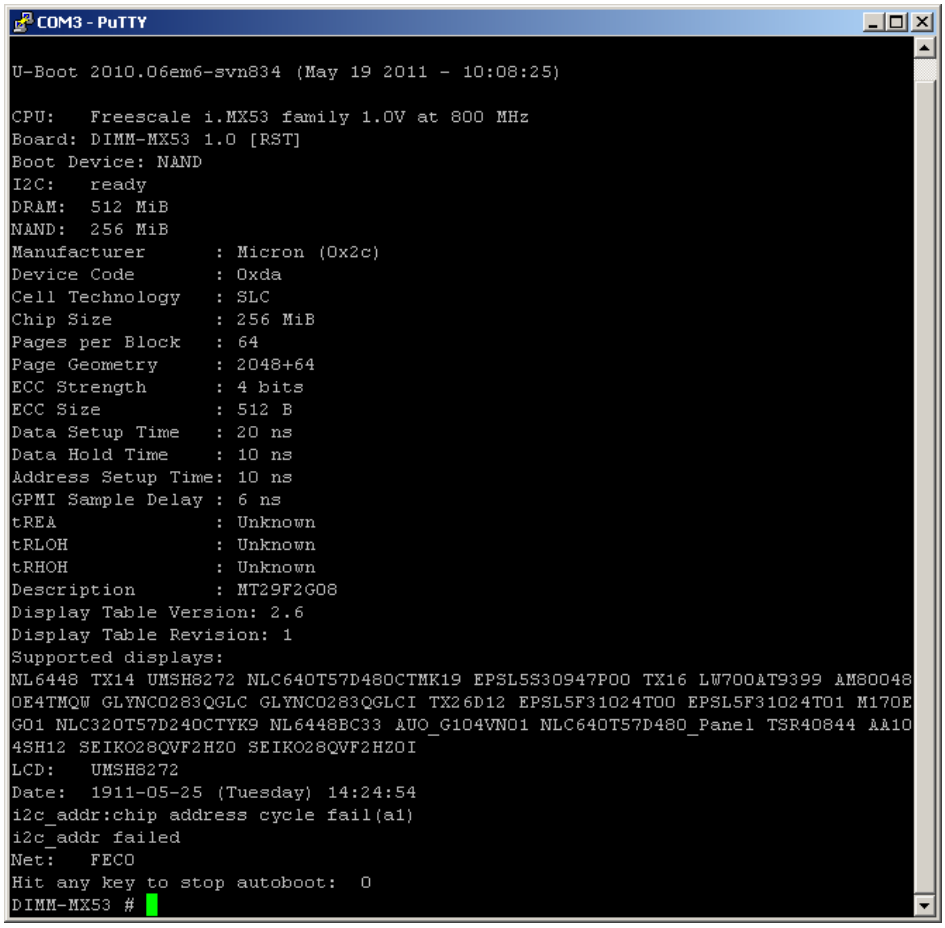
9.1 Setting up for application development

There are only two steps to do:

- Connect the target to the host via the supplied USB cable A-B to allow a Microsoft® ActiveSync (Mobile Device Center) connection. The USB-B type connector is located at the front of the target.
- Use the Ethernet cable to connect the target to the network where the host is linked to.

9.2 Setting up kernel building or updating the image

- A Microsoft ActiveSync connection is not necessary
- Enter the bootloader prompt as described in section Bootloader prompt
- Connect the target to the host via the serial cable. Use the 9-pin D-Sub jack of the serial port UART A at the target.
- Power ON or reset the target. When a terminal application is running on the host the bootloader outputs the following information:



- Make a physical Ethernet connection, like in application development above. Set the network environment variables to your requirements. How to update the installed images is described in section The Bootloader.

10 DVD of the Developer Kit

The supplied DVD contains all the software and documentation you need to start developing applications or building kernels based on Windows® Embedded EC 7 for DIMM-MX53.

The DVD contains all data required for development of applications and/or own images for the platform DIMM-MX53. In the root directory of the DVD you will find a html-file named start_here.html. This file can be opened in a html browser such as Internet Explorer or Firefox. The page which is shown gives an overview about the DVD contents.

10.1 Installation of the BSP

Before installing the BSP of the Developer Kit, please make sure that Microsoft® Windows Embedded EC 7 is already installed on the host (see chapter "Installing the Platform Builder for kernel development").

The installation of the BSP is a copy operation. Use the Windows Explorer (or another file explorer) to copy the contents of the directory BSP from the DVD to the location of your Windows Embedded CE installation. The folder WINCE700 is the same folder as the folder WINCE700 which was created during the installation of the Platform Builder. You should preserve the directory structure of the copied data.

Each folder in the copied structure contains a subfolder with the name .svn. These folders are not used by the Platform Builder but are required if you want to download updates from the emtrion update server (see next section).

10.2 Updating the BSP to the newest release

Emtrion provides the possibility to update your BSP to the newest revision which was released from emtrion. This can be done using a subversion client. Subversion is a open-source version control system. More details you can find in the document "How to update your product from the update repository". This document is also on the DVD.

11 Features of the Developer Kit

The Developer Kit is delivered with the DIMM-MX53 and a base board. Together they provide a wide range of functionality. The following sections describes what the software is supporting at release time of this manual.

11.1 Features presented by the BSP

The Board Support Package (BSP) presents the foundation for building OSDesigns on DIMM-MX53. It provides the support of features on DIMM-MX53. The BSP is delivered with binary files. The supported features are listed below.

11.1.1 Clocks

The input clock of the processor is 24,576MHz . This results in the following clocks.

CPU -> 800 MHz

DDR SDRAMs -> 400 MHz

Bus -> 200 MHz

11.1.2 RAM

DIMM-MX53 is equipped with 512MB DDR2 SDRAM. The data width is 32 bit. The following figure shows the memory layout under Windows EC.

Physical address	Usage of the RAM areas
0x70000000	Reserved
0x70001000	Structure _DRIVER_GLOBALS used as interface Bootloader/Wince
0x70002000	Reserved
0x70200000	WINCE-Image(kernel)
0x76200000	Object-Store, RAM for WINCE
0x8FC00000	Bootloader/Frame Buffer
0x8FFFFFFF	

11.1.3 NAND

11.1.3.1 ONFI

The NAND driver provides the feature ONFI. This feature allows replacing of NANDs without changing the software respective the WINCE-Image. This results in reducing the effort of administration. No any additional revisions have to be managed for this reason.

To satisfy this behavior the following conditions or restrictions have to apply by the NAND.

1. Supporting of the ONFI interface
2. Interleave mode is not supported
3. Only one chip select
4. Existing of one BBI marker to mark a bad block
5. Location of the BBI marker has to be in the first page of a block
6. Parameters have to conform to the layout of the NFC, described in the chapter 51 of the i.MX53 Multimedia Applications Processor Reference Manual

11.1.3.2 Layout

The size of the used NAND can be 256, 512 or 1024MByte. The storage of the NAND is used for the bootloader and its configuration, holding the WINCE-Image and a FlashFile-System. The general structure is shown below.

NAND address	Usage of the NAND
0x00000000	Bootstrapper
0x00080000	U-Boot
0x00100000	NVRAM Environment variables of the bootloader
0x00200000	Splash-Screen
0x00280000	WINCE-Image
0x07000000	FlashFile-System (exFAT)
0x0FFFFFFF	

At power on or following a reset the WINCE-Image is copied from NAND to RAM by the bootloader to be started.

The storage of the Flash Filesystem appears as folder “NAND Flash” inside the WINCE.

11.1.4 Display

The display driver is just prepared to use the same displays as it is supported by the bootloader. To offer a wide flexibility in changing display the display has to be selected by the bootloader. For them look to the chapter “Changing the display”. If you have a base board which supports it, it is also possible to use a DVI Monitor as display. You can chose between parallel and DVI output in the Platform Builder catalog. Furthermore it is possible to use a dual display mode by using an additional

LVDS display. For the exact description of this mode please look into Freescale BSP Reference Manual. (2)

11.1.5 Graphical Hardware Acceleration

Hardware acceleration reduces the CPU usage and increases the performance of the system. The i.MX53 processor offers hardware accelerators for 2D and 3D graphics.

1.) 2D acceleration is provided by an OpenVG 1.01 compatible IP core. A library is provided for Windows EC
2.) 3D acceleration is provided by an IP core which is designed to support OpenGL ES 1.1 & 2.0 as well as Direct3D Mobile 1.2. For both standards drivers and libraries are provided.

11.1.6 Video processing

The i.MX53 features a Video Processing Unit VPU core. This IP core offers the acceleration of en-/decoding for several video formats:

• H.264/AVC decoder for baseline profile, main profile and high profile
• VC-1 decoder for simple profile, main profile and advanced profile
• MPEG-4 decoder for simple profile, advanced simple profile except GMC
• H.263 decoder for baseline profile
• Divx decoder for home theater and high definition profile (version 3.x, 4.x, 5.x, 6.x) and Xvid
• MPEG-2 decoder for main profile @ main and high level
• RV decoder for profile 8/9/10
• MJPEG decoder for Baseline profile
• H.264/AVC encoder for baseline profile
• MPEG-4 encoder for simple profile
• H.263 encoder for baseline profile
• MJPEG encoder for baseline profile
• Multiple codec: supports up to 4 decoding/encoding

The IP Core is supported under Windows EC 7. For a detailed description on how the different codecs are supported please look into the i.MX MX53 Multimedia Applications Processor Reference Manual. (1)

11.1.7 I2C

The DIMM-MX53 possesses 1 I^2C-Interface connected to the SODIMM connector. Several slave devices can be connected to the I2C-Bus, like audio codec and external RTC, depending on the base board. They are serviced by their own device driver.

But it is possible to connect additional devices to the bus. In this case an API is available. Since the driver is provided from Freescale please refer to their documentation. (2)

11.1.8 Touch

A four wire touch is fixed on the display delivered by the Developer Kit and is connected to the touch controller AR1020. The touch driver communicates via the I^2C bus to the touch controller. The slave address is assigned to 0x4d.

The touch driver reads user input from the controller and send it to the GWES. In order that the touch driver can report calibrated points to the GWES, calibration data is required. By default, the calibration data is provided by the U-Boot bootloader.

The bootloader data can be overridden if the touch screen is recalibrated under Windows CE.

Starting with the revision v0300 of the BSP also the Multi-Touch feature is supported (in conjunction with the capacitive touch on the display adapter ET0700M06). The support is automatically enabled during startup if the display adapter ET0700M06 is detected during system startup. Please consider the remark about the CETouchView tool in section 20.6.

11.1.9 Real-Time Clock (RTC)

Due to the RTC of the DIMM-MX53 is not battery buffered, there is an external RTC on the DIMM-MX53 which is buffered by the battery on any suited base board. The external RTC is connected to the I2C-Bus. Its slave address is 0x68.

When starting WINCE, the contents of the external RTC are read and then written to the internal RTC. While WINCE is running the internal RTC is used. But if a new time/date is set, the external RTC is also updated.

Using the WIN32-API functions GetSystemTime and SetSystemTime you can read and set the time of the RTC from your application.

11.1.10 Ethernet

With the implemented ethernet driver the provided network functionality of WINCE is available. The Ethernet driver supports in detail

- 100/10MBit full- and half-duplex
- Full-duplex flow control
- Auto-negotiation

11.1.11 USB Host

The USB Host controller was successfully tested with following devices under Windows CE.

- keyboard
- mouse
- memory stick
- hub

11.1.12 USB Function (USB Device)

The i.MX53 has an integrated USB OTG controller. On the DIMM-MX53 it is setup to be used as USB Function-only. It delivers the functionality for application developers to use Microsoft ActiveSync for debugging purposes.

Microsoft ActiveSync offers to you

- Debugging in combination with Visual Studio 2005/2008
- Using the Remote Tools of VS2005/VS2008
- Exploring the target
-

11.1.13 Serial Ports

The DIMM-MX53 has five serial communication interfaces. One features RTS/CTS signals and RS232 signal level. The others only provide the RX/TX signals and LVTTL level.

The relationship between the names and serial ports from the hardware and software point of view and some more supported features are summarised in the table below.

Software	COM1	COM2	COM3	COM4	COM5
Channel/ interface	UART1	UART2	UART3	UART4	UART5
Signals	RS232	LVTTL*	LVTTL*	LVTTL*	LVTTL*
HW-Handshake RTS/CTS	Yes	No	No	No	No
Hardware Flow Control RTS/CTS	No	No	No	No	No
HW-FIFO					
- Receive	64 Byte	64 Byte	64 Byte	16 Byte	16 Byte
- Transmit	64 Byte	64 Byte	64 Byte	16 Byte	16 Byte
Baudrates					
- 1200	yes	yes	yes	yes	yes
- 2400	yes	yes	yes	yes	yes
- 4800	yes	yes	yes	yes	yes
- 9600	yes	yes	yes	yes	yes
- 14400	yes	yes	yes	yes	yes
- 19200	yes	yes	yes	yes	yes
- 38400	yes	yes	yes	yes	yes
- 57600	yes	yes	yes	yes	yes
- 115200	yes	yes	yes	yes	yes
- > 115200	***	***	***	***	***

(*) Suitable RS232 adapters are available by emtrion

(**) For more details please refer to the corresponding hardware manual on the DVD

(***) Only on request

All the serial ports can be accessed by the Win32-API of Windows Embedded CE. But with the debug version of the Developer Kit kernel image COM1 is reserved for debugging purposes.

11.1.14 Audio

The SSI-interface of the i.MX53 is used to make audio available with any of the suited base boards from emtrion. The base boards are equipped with the audio codec TLV320AIC23B. The codec is connected to the I2C-Bus. Its slave address is 0x1B.

The driver can process 8- and 16bit samples in mono and stereo for sample rates up to 44100Hz.

11.1.15 SPI

The DIMM-MX53 features 1 SPI interface on the DIMM connector. The driver for it is provided by Freescale. To learn about the API please refer to Freescale's documentation. (2)

11.1.16 Camera Interfaces

The DIMM-MX53 features 2 camera interfaces. If you have our base board Verno, you can only use the first camera interface. The base board Lothron features connectors for both camera interfaces. There is also an ADV7180 Analog/Digital Converter on this base board. It can be used to interface an analog PAL/NTSC camera. The BSP includes drivers for ADV7180 and VM009 CMOS camera from

emtrion. The drivers can be chosen in the Platform Builder catalog. If you are using the VM009 camera make sure that its jumper is set to I^2C address 0x48 when connecting it to camera interface 1 or I^2C address 0x5D when connecting it to camera interface 2. To learn about the camera interface API please have a look at the corresponding documentation from Freescale. (2)

11.2 HiCOCAN CAN Bus Interface

The i.MX537 features two internal CAN-Controllers. Freescale offers an own driver for these controllers, but for compatibility reasons regarding our existing CAN Architecture and for performance reasons emtrion developed an own driver.

This driver supports the HiCOCAN API, and its usage is documented in detail in the DIMM-MX537 CAN-SW manual which you also have received with this Developer Kit. **Please note, that the CAN Interface is only available on DIMM-MX537 and not on DIMM-MX535.**

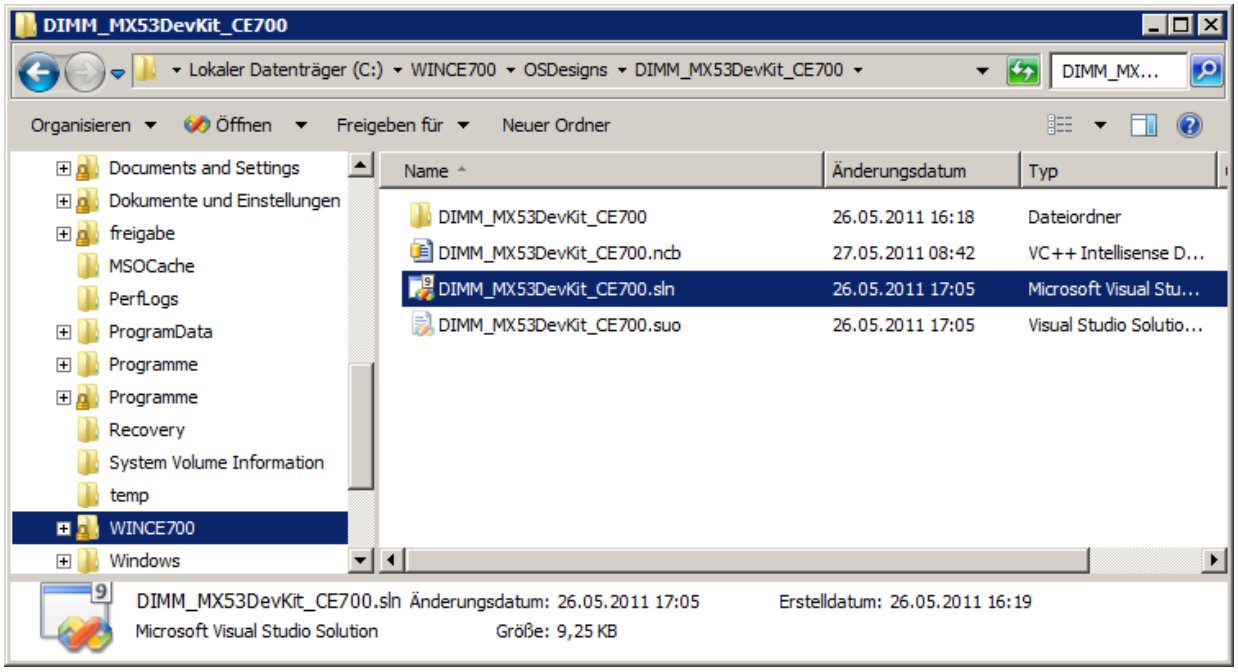
11.3 Features presented by the OS Design

The OS Design is the summary of all the software components that have been included into the kernel of the Developer Kit and reflects its functionality as well. The OS Design comprises the BSP, some emtrion tools and selective functions delivered by Windows Embedded Compact 7. Here are some highlights of the kernel.

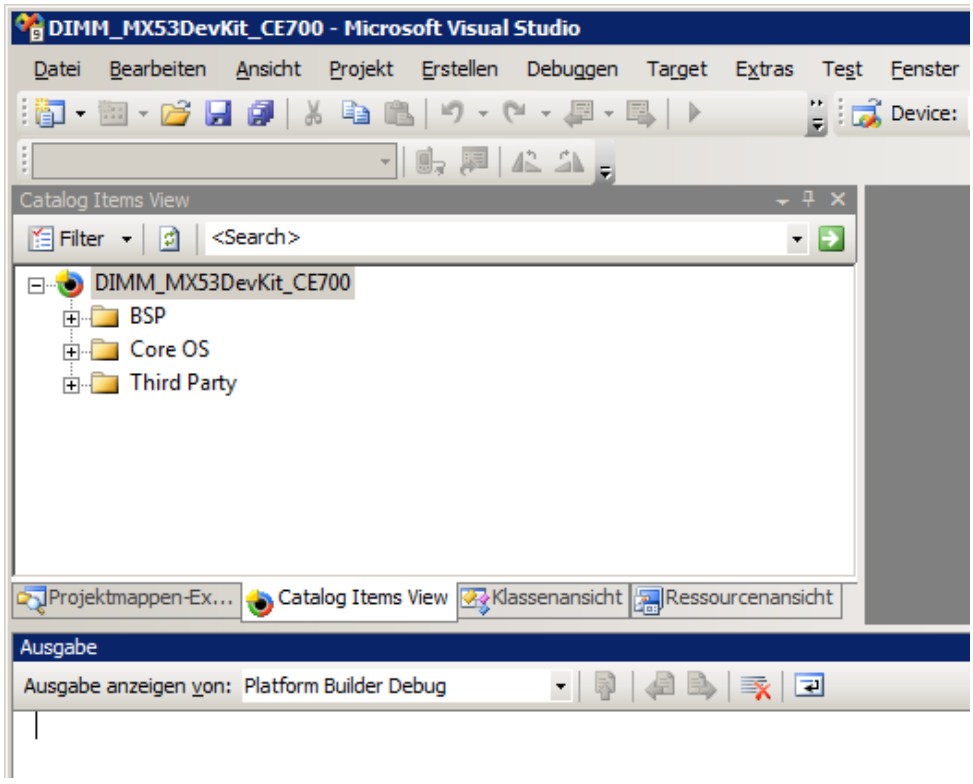
Highlights of the Developer Kit kernel
.NET Compact Framework 3.5
Silverlight for Windows Embedded
CAB File Installer/Uninstaller
Microsoft Flash File System
Autostart of a custom application from Mass-Storage-Devices (e.g. Flash File System, USB Stick, …)
Graphical User Interface
Complete Win32-API, available for Windows CE
Wired Local Area Network
USB Host (USB keyboard, USB mouse, USB memory stick and USB printer)
Touch
5 serial ports (COM1:, COM2:, COM3:, COM4: and COM5:)
USB function (ActiveSync support)
Internet Explorer 7.0 for Windows Embedded Compact
Telnet and FTP server functionality
Software development and debugging with Visual Studio 2005 via Ethernet is prepared (ConmanClient2.exe and CMAccept.exe are integrated into the kernel)

The exact functionality what is containing the kernel can be checked in the window “Catalog Items View” of the OSDesign DIMM_MX53DevKit_CE700.

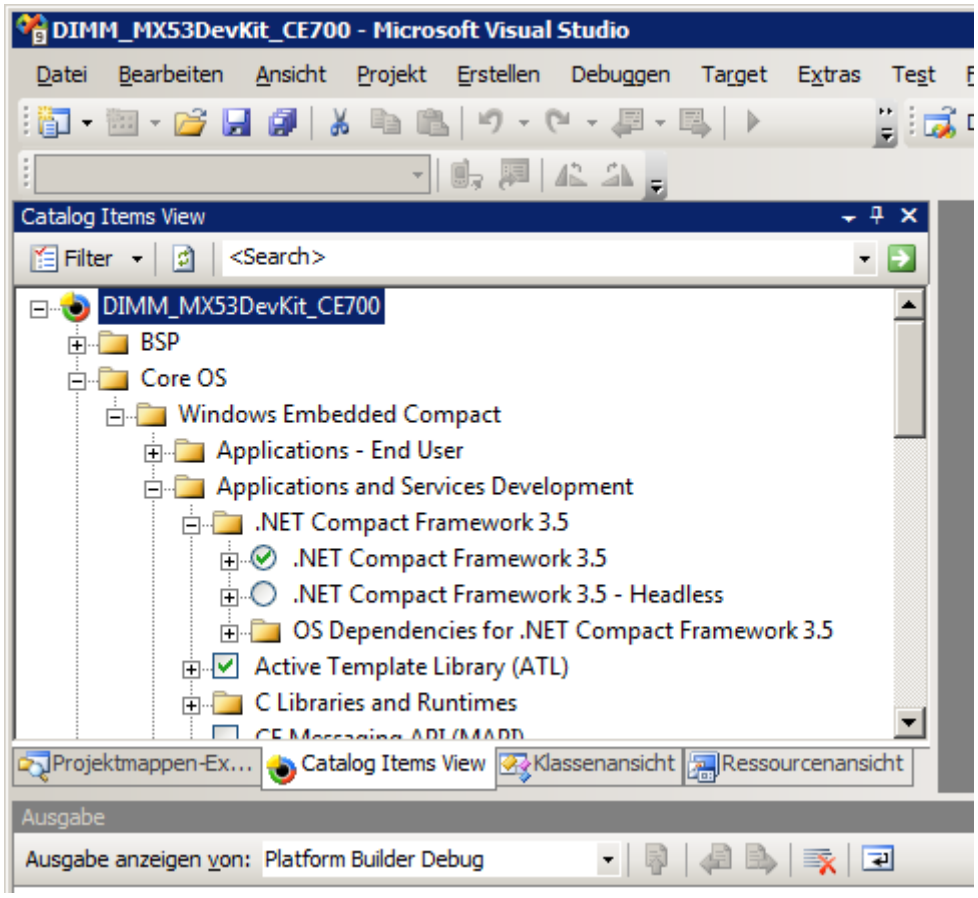
The OSDesign can be opened either out of the VS2008 by navigating to the location of the solution or by double click the solution file DIMM_MX53DevKit_CE700.sln.



The figure below presents the opened solution of the OSDesign (DIMM_MX53DevKit_CE700) of the Developer Kit and the "Catalog Items View" window in Visual Studio 2008. There are four folders visible. The OSDesign of the Developer Kit consists of the components from the folders Core OS, Device Drivers and Third Party. Look in any of the folders for selected items to get an overview of the components in the kernel.



The next screen shot shows a part of the opened folder CEBASE and some of its items. The selected components are part of the kernel.



11.3.1 Persistent Registry

The OSDesign of the Developer Kit kernel comes with a hive-based persistent registry. At such a kernel, changes on the registry can be made persistent. This means the changes are kept at power off or at reset.

But be careful with wrong settings on the registry. This can cause the kernel to not start correctly the next time. When this behaviour occurs deleting the persistent registry is the only way to leave this issue. This can be done by setting a flag in the bootloader's environment variables. To avoid such situations, kernels without a persistent registry can also be ordered at emtrion.

For more details to the persistent registry see at the section The Persistent Registry.

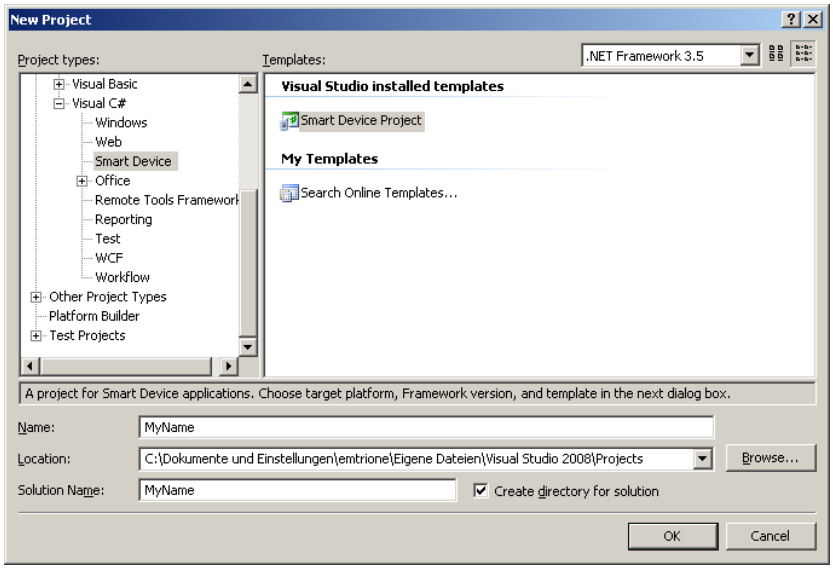
11.3.2 Autostart Mechanism

In the OSDesign an autostart feature is implemented. It allows the execution of a specified application at system start. There are two options, starting the application at launch time of the explorer or at launch time of the autostart application. More information at this topic is given in the chapter Starting an application at system start.

11.3.3 Creating a new Managed Project

The first step is starting a new instance of VS2008.

1. Select **File/NewProject** ... from the Visual Studio menu.
2. In the **NewProject** window select **Visual C#/SmartDevice/WindowsCE5.0**
3. Name your project **MyName** and click **ok**.



In the second windows you now select **DIMM-MX53DevKit_CE700_SDK** as **Target Platform**. Also select **.NET CF Version 3.5** and the **Device Application** template. Acknowledge by clicking **ok.**

Now you can start developing your application.

11.3.4 Deploying to the target

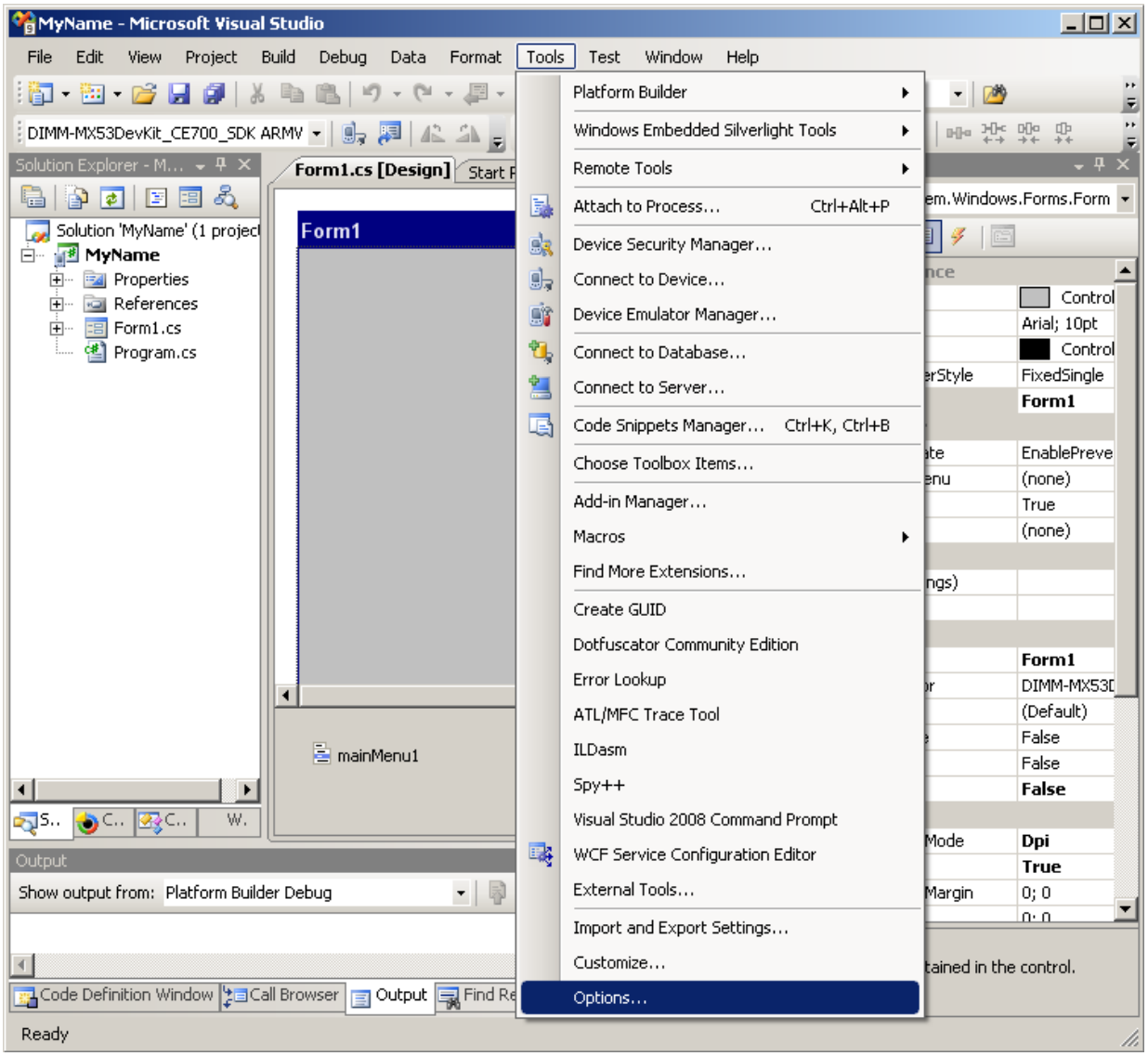
The connection between VS2008 and the target is made by network. For this reason the IP address of the target is necessary. To make the IP address known to VS2008 there are two options.

The first option is to ask manually by means of the function „ipconfig“ on the device. For that “ipconfig” must be executed from the command shell.

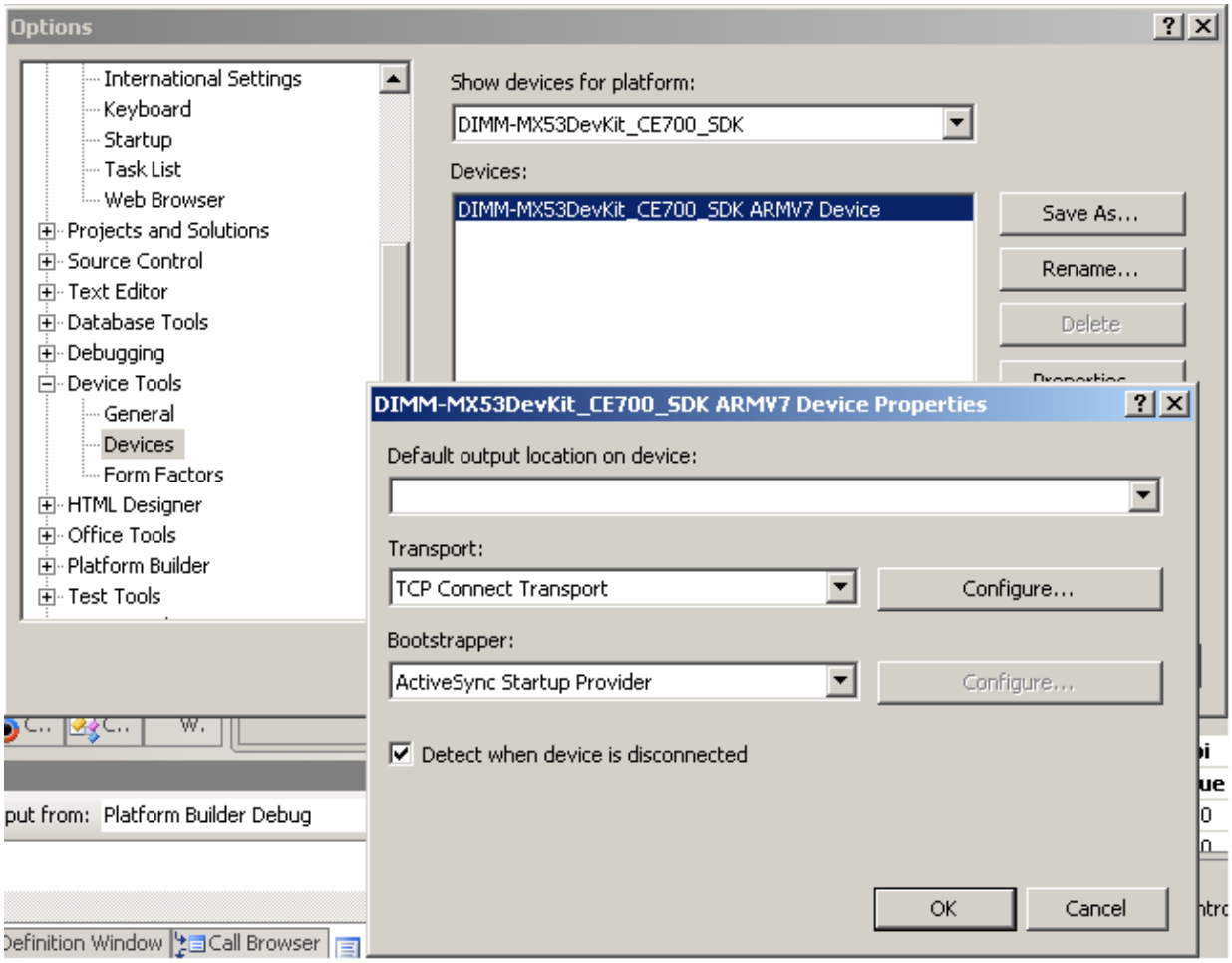
The second is receiving the IP address automatically by an existing USB ActiveSync connection.

After getting this address, the next step is preparing the managed application development environment for deployment.

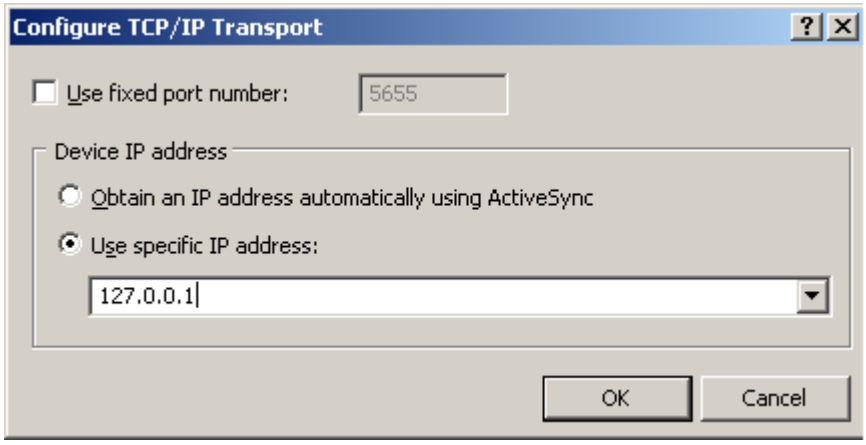
1. In VS2008 select **Tools/Options** from the menu.



2. In the options window, expand the **Device Tools** node and select **Devices**.
3. In the **Show device for platform**: drop down box select **DIMM-MX53DevKit_CE700_SDK**.
4. Click on **DIMM-MX53DevKit_CE700_SDK** and select **Properties.**



5. Click the **Configure** button beside the **Transport** drop down box. We are going to configure the TCP Connect Transport.



6. In the case when an USB ActiveSync connection exists between the workstation and the device, the option “Obtain an IP address … is to select. In the other case the option “Use specific IP address” is to select and the IP address you got on the device by ipconfig has to be inserted.
7. Click **OK** through all of the dialogs.

Preparing the target:

1. At the command shell, type **ConmanClient2**.
2. Then, type **cmaccept**. You have 3 minutes to establish a connection with your managed application.

Deploying the managed application:

1. Select **Debug/Start Debugging** from VS2008.
2. Select **DIMM-MX53DevKit_CE700_SDK** from the list of devices in the **Deploy MyName** box and click **Deploy**. VS2008 will eventually deploy several cab files to the device in addition to your application. Your application is executed on the target device.

11.4 Software Development Kit (SDK)

Together with the Developer Kit kernel, the suitable Software Development Kit for Visual Studio 2008 is delivered too. This allows you to develop applications based on the Developer Kit kernel.

Note:
Normally, a SDK is valid for a specific OSDesign. But it is possible to develop applications based on a SDK different to an OSDesign. In such a case you must be sure the application uses only APIs that are supported by the different OSDesign.

To avoid errors caused by used functions that are not supported by the OSDesign, we recommend to use the appropriate SDK at any time.

You can install the SDK from the Developer Kit DVD.

11.5 Manuals

There are some manuals delivered by this Developer Kit. They are stored in the folder “documents” of the DVD.

11.6 Features of the preconfigured OS Design

This chapter describes briefly the contents of the kernel delivered on the DVD or available online in the product update repository or on our support site.

Note:
The image on our support site is not licensed in contrast to the image which can be found on the DVD or in the product update repository.

The kernel is a typical one that should be suitable for most of applications. If modifications are necessary, you may consider to make your own image or just asking emtrion to design an individual kernel for your needs.

11.6.1 Applications – End User

	Included
ActiveSync	X
File Sync	X
CAB File Installer/Uninstaller	X

11.6.2 Applications – End User

	Included
.NET Compact Framework 3.5	
.NET CompactFramework 3.5	X
.NET CompactFramework 3.5 – headless	-
OS Dependencies for .NET CF 3.5	X
OS Dependencies for .NET CF 3.5 - headless	-
Active Template Library (ATL)	X
C Libraries and Runtimes	
C++ Runtime Support for Exception Handling and Runtime Type Information	X
Full C Runtime	X
Standard I/O (STDIO)	X
Standard I/O ASCII (STDIOA)	X
Standard String Functions - ASCII	X
COM and DCOM	
DCOM, incl. COM Storage	X
Minimal COM (No OLE Support)	-
String Safe Utility Functions	X
XML	
XML Core Services and Document Object Model	X
XML Query Languages (XQL)	X

11.6.3 Communication Services and Networking

	Included
Networking General	
NDIS User-mode I/O Protocol Driver	X
Network Driver Architecture (NDIS)	X
Network Utilities (Ipconfig, ping, route, netstat)	X
TCP/IP	X
IP Helper API	X
TCP/IPv6 Support	X
Windows Networking API/Redirector (SMB/CIFS)	-
Winsock Support	X
Networking – Local Area Network (LAN)	
Wired Local Area Network (802.3, 802.5)	X
Networking – Wide Area Network (WAN)	
Dial Up Networking (RAS/PPP)	X
AutoDial	X
Standard Modem Support for Dial Up Networking	X
Telephony API (TAPI 2.0)	X
Unimodem Support	X
Servers	
Core Server Support	X
File Server (SMB/CIFS)	-
FTP Server	X

RAS Server/PPTP Server (Incoming)	-
SNTP Client with DST	X
Telnet Server	X

11.6.4 Core OS Services

	Included
Toolhelp API	X
Device Manager	X
Display Support	X
Serial Port Support	X
UI Proxy for Kernel-Mode Drivers	X
Windows Embedded CE Driver Development Kit Support Libary	X
Kernel Functionality	
Fiber API	X
FormatMessage API	X
FormatMessage API – System Error Messages	X
Memory Mapped Files	X
Message Queue – Point-to-Point	X
Target Control Support (Shell.exe)	X
Notification	
Non UI based Notification	-
UI based Notification	X
Power Management	
Power Management (Full)	X
Power Management (Minimal)	-
USB Support	
USB Function Driver	X
USB Host Support	X
USB Human Input Device (HID) Class Driver	X
USB HID Keyboard and Mouse	X
USB HID Keyboard only	X
USB HID Mouse only	X
USB Printer Class Driver (PCL3)	X
USB Storage Class Driver	X

11.6.5 Device Management

	Included
Simple Network Management Protocol (SNMP)	X

11.6.6 File System and Data Store

	Included
Compression	**x**
Database Support	
CEDB Database Engine	X
File and Database Replication, bit-based	X
File Cache Manager	-

File System - Internal	
RAM and ROM File System	X
ROM-only File System	-
Registry Storage	
Hive-based Registry	-
RAM-based Registry	X
Storage Manager	
FAT File System	X
Partition Driver	X
Release Directory File System	X
Storage Manager Control Applet	X
System Password	X

11.6.7 Fonts

	Included
Arial (Subset 1_30)	X
Courier New (Subset 1_30)	X
Symbol	X
Tahoma (Subset 1_30)	X
Tahoma Bold	X
Wingding	X

11.6.8 Graphics and Multimedia Technologies

Note: Many codecs are not installed from this section because there are equivalent codecs provided by Freescale which are accelerated in HW. These codecs are mention in section Multimedia .

	Included
Audio	
Audio Compression Manager	X
Waveform Audio	X
Graphics	
Alphablend API (GDI version)	X
DirectDraw	X
Gradient Fill Support	X
Still Image Codec Support (Encode and Decode)	X
Still Image Encoders and Decoders	X
BMP Decoder	X
GIF Decoder	X
JPG Decoder	X
PNG Decoder	X
Windows Codecs	X
Media	
Audio Codecs and Renderers	
MP3 Codec	X
Wave/AIFF/au/snd File Parser	X
WMA Codec	X
WMA Voice Codec	X
DirectShow	

ACM Wrapper Filter	X
DirectShow Core	X
DMO Wrapper Filter	X
HTTP Streamer Filter	X
Local File Streamer	X
Media Formats	
MPEG-1 Parser/Splitter	X
Windows Media Player	
Windows Media Technologies	X
WMA and MP3 Local Playback	X
WMA and MP3 Streaming	X

11.6.9 International

	Included
Input Method Manager (IMM)	X

11.6.10 Internet Client Services

	Included
Internet Explorer 7.0 for Windows Embedded Compact – Standard Components	X
Internet Explorer 7.0 Sample Browser	X
Internet Explorer 7.0 for Windows Embedded CE Components	X
Internet Explorer Browser Control Host	X
Internet Explorer HTML/DHTML API	X
Internet Explorer Multiple-Language Base API	X
URL Moniker Services	X
Windows Internet Services	X
Internet Options Control Panel	X
JScript 5.8	X
VBScript 5.8	X

11.6.11 Security

	Included
Authentication Services	X
NTLM	-
Schannel (SSL/TLS)	X
Credential Manager	X
Cryptography Services (CryptoAPI 1.0) with High Encryption Provider	X
Certificates (CryptoAPI 2.0)	X

11.6.12 Shell and User Interface

	Included
Graphics, Windowing and Events (GWES)	

Minimal GDI Configuration	X
Minimal GWES Configuration	X
Minimal Input Configuration	X
Minimal Window Manager Configuration	X
Command Shell	
Aygshell API Set	X
Command Processor	X
Console Window	X
Graphical Shell	
Standard Shell	X
Common Controls	
Common Control	X
Common Dialog Support	X
Control Panel Applets	X
Mouse	X
Network User Interface	X
Software-based Input Panel (SIP)	X
SIP for Small Screens	X
Software-based Input Panel Driver	X
Touch Screen (Stylus)	X

11.6.13 Device Driver

	Included
Input Devices	
Keyboard/Mouse - Layout Manager	X
Networking	
WAN devices	
Asyncmac NDIS Driver	X
SD Bus Driver	X
SD Memory	X
Flash MDD	
USB Function Clients	X
Composite Function Driver	X
USBFN Mass Storage Client	X
USBFN RNDIS Client	X
USBFN Serial Client	X
USB Host Class Drivers	
USB Human Input Device (HID) Class	X
USB HID Keyboard and Mouse	X
USB Printer Class Driver	X
USB Storage Class Driver	X

11.6.14 BSP

	Included
TLV320AIC23 Audio Driver	X
Camera CSI0	
CMOS MTM9M131 Support (emtrion VM009)	X
TVin ADV7180 Support	

Camera CSI1	
CMOS MTM9M131 Support (emtrion VM009)	X
CANBUS HiCOCAN Platform Driver for DIMM-MX537	X
Enhanced CSPI1 Support	X
IPU Support for U-Boot display parameters	X
Fast Ethernet Controller (FEC)	X
Graphics Processing Unit (GPU)	
Direct 3D Mobile support	X
OpenGL ES support	X
OpenVG support	X
I2C Bus 3	X
Enhanced SD Host Controller 1 & 2	X
UART 1-5	X
USB High Speed Host	X
USB High Speed OTG Pure Client	X
Video Processing Unit Support	
OpenGL XAML Render Plug-in	X
Micron MT29F2G08 NAND Flash Support	X
AR1020 touch controller driver	X

11.6.15 Emtrion Tools

	Included
autostart	X
Corecon_1	X
Emtrion Remote Services	X
Revision Info	X
Writereg	X

11.6.16 Multimedia

	Included
App	
FSL A/V Capture App	X
Audio	
AAC Decoder	X
AAC Plus Decoder	X
AMR Decoder	X
Audio Decoder Filter	X
BSAC Decoder	X
FLAC Decoder	X
MP3 Decoder	X
MP3 Encoder	X
Ogg Vorbis Decoder	X
PEQ Post-Processor	X
SBC Encoder	X
TSM Post-Processor	X
Image	
BMP Decoder	X
GIF Decoder	X
JPEG Decoder	X

PNG Decoder	X
Parser	
FSL AVI Parser	X
FSL MP4 Parser	X
FSL MPEG2 Parser	X
FSL OGG Parser	X
FSL Source Filter	X
Video	
FSL VPU Decoder	X
FSL VPU Decoder with CSC	X

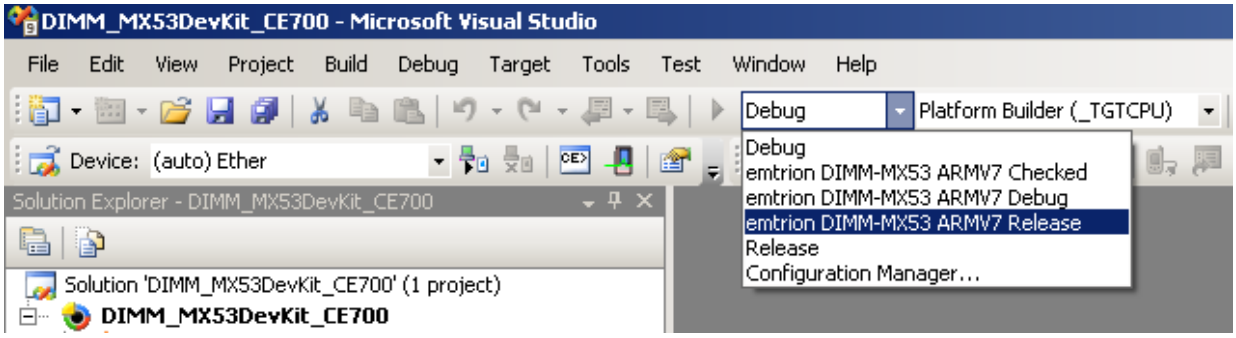
12 Creating a WINCE image (Kernel development)

Normally, when you create your first image after creating or open a new OSDesign, make build-related settings. The settings for the OSDesign of the Developer Kit have already set and differ from release and debug version.

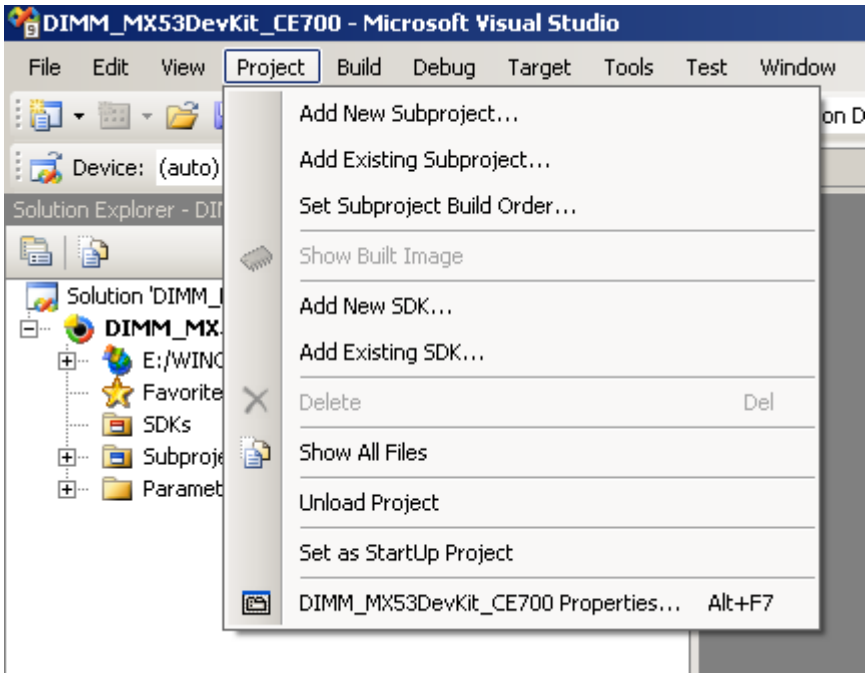
This section describes how to create an image of release of the Developer Kit kernel. If you wish to generate a debug image, refer to the chapter Building a Debug WINCE-Image (Kernel development).

At this point we assume the solution of the OSDesign is already open.

- First, choose the configuration for release as shown:



- Second, navigate to the Project Properties Pages
  Go to menu “Project” of Visual Studio 2008 and select “DIMM_MX53DevKit_CE700 Properties…” to open the Project Properties Page.



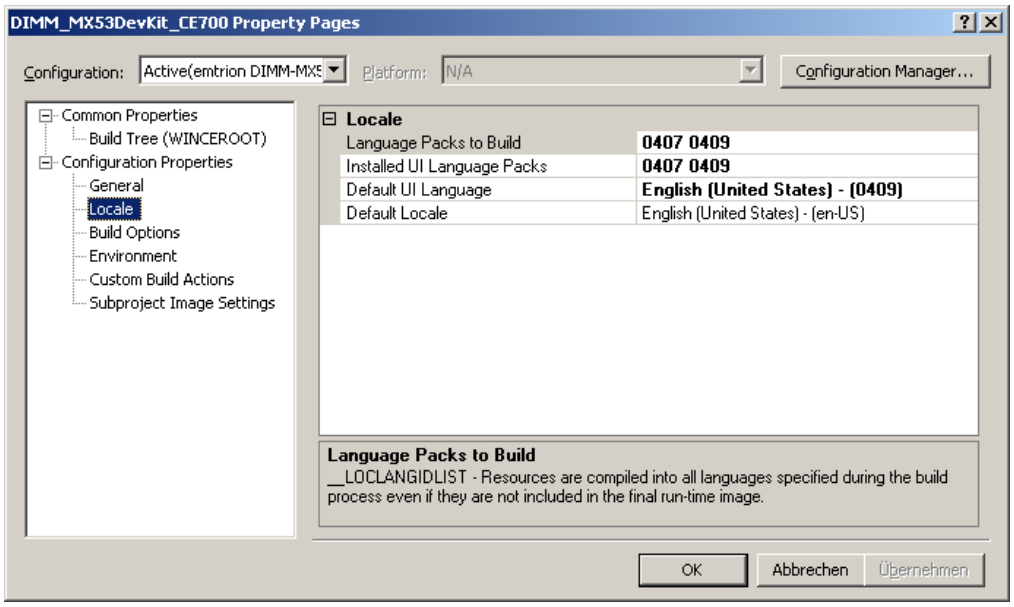
The following subchapters explain some required settings.

12.1 Settings of the Developer Kit kernel

Following the next sections you will get an overview about the settings of the kernel. In general, all settings have to be made before building the kernel.

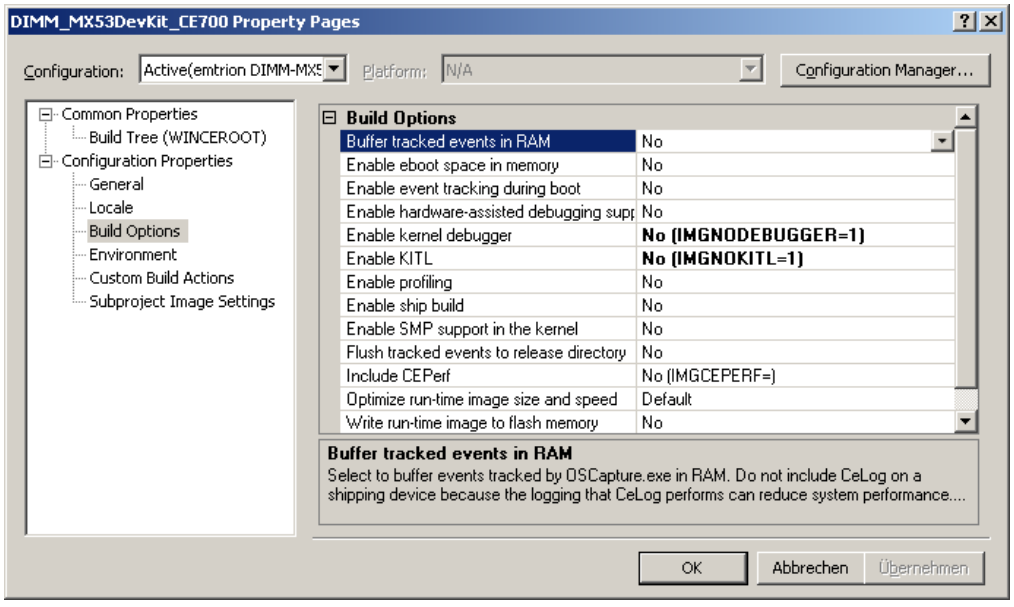
12.1.1 Language settings

The language settings of the kernel can be found in Locale.



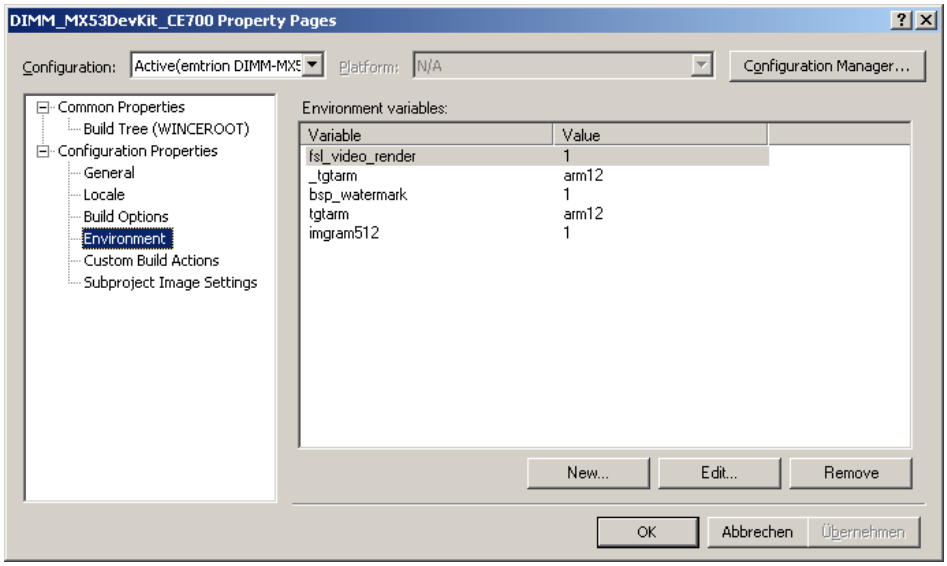
12.1.2 Build Options

There are many build options available. But for release, only one setting is enabled. This setting disables the compiling of RETAILMSG.



12.1.3 Environment settings

While no debug macros are compiled in release, the environment variable for output debug messages is deactivated.

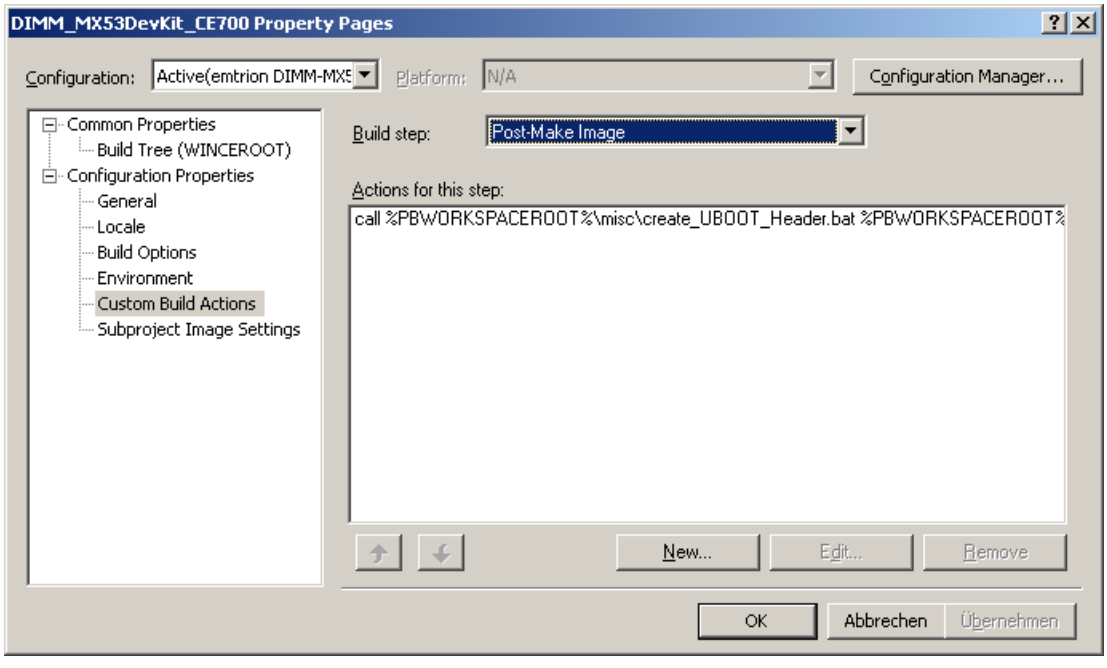


12.1.4 Custom Build Actions

The build process is using some custom batch files for creating the kernel of the Developer Kit. The batch files do some special work in several of the build steps.

In the build steps “Pre-Sysgen” and “Pre-Make image” some files are copied to different locations.

In the build step “Post-Make Image” the batch “create_UBOOT_Header.bat” is called. This batch is using some tools coming with the Platform Builder. The batch file is preparing the original image nk.bin (nk.nb0) to a file which can be handled by the U-Boot bootloader. First, Microsoft’s license key is supplied to the raw binary data file if a specified file containing this product key is present. In the second step the nk.nb0 is extent with a header which is required by the U-Boot.



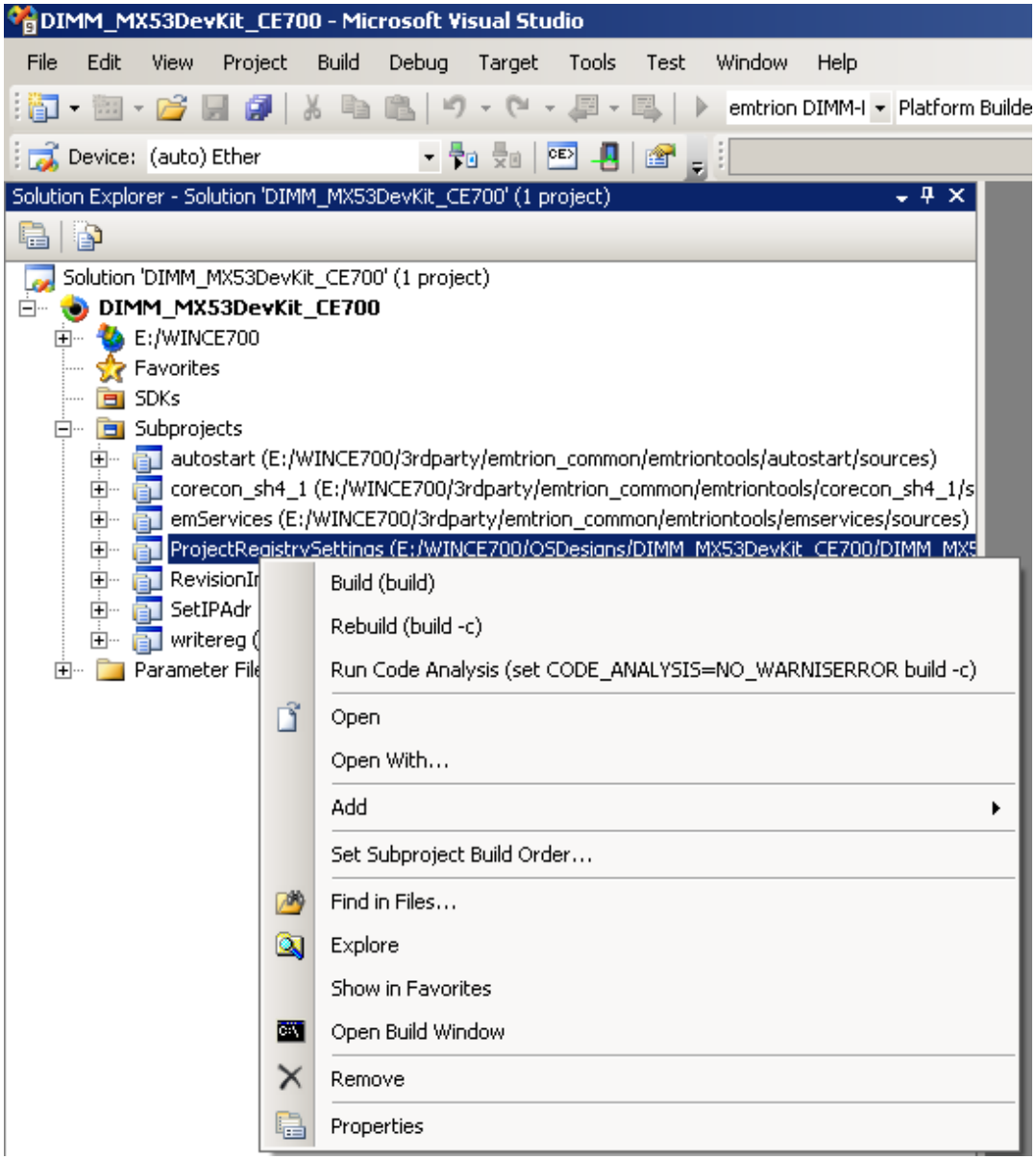
Remark:
If you create your own OSDesign you have to add the execution of the batch file "create_UBOOT_Header.bat" to the "Post-Make Image" step. Otherwise, you can not download the created image to the target device.

12.2 Build

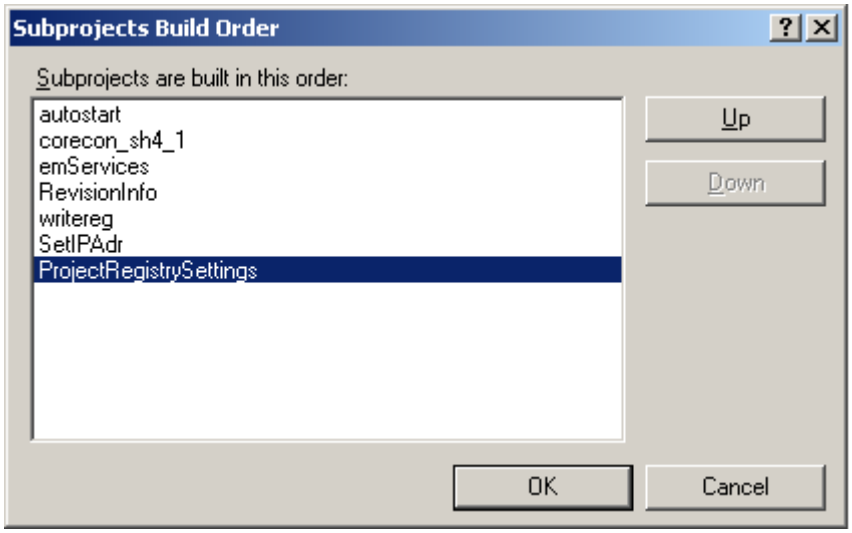
Whenever you are building a kernel of the Developer Kit, the build order of the subproject described in the following sections must be considered.

12.2.1 The subproject ProjectRegistry Settings

The OSDesign of the kernel includes some subprojects and also the subproject ProjectRegistrySettings. This project contains some important registry entries and is provided for adding, removing and overwriting of any existing entry. These registry entries have to merged in a defined build order. This order must be the last at any time.



The subproject ProjectRegistrySettings is built at last:

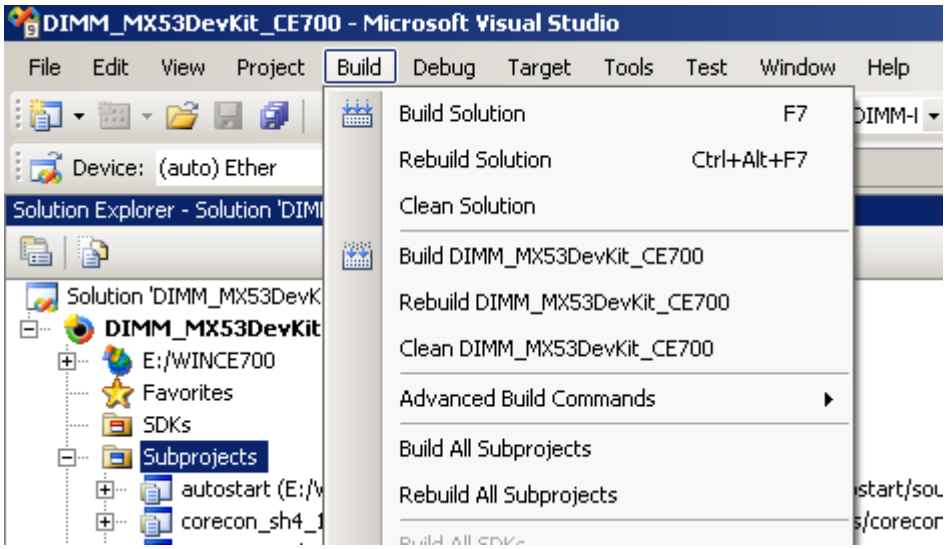


12.2.2 Starting the build process

A complete build process is required in the following cases:

- At first build of the kernel
- Adding or removing drivers
- Adding or removing OS components
- Changing (Swtiching) the OSDesign
- After you have update the BSP from the emtrion update repository

In all cases choose [Build] -> [Clean Solution] and the [Build] -> [Build Solution] from the menu.



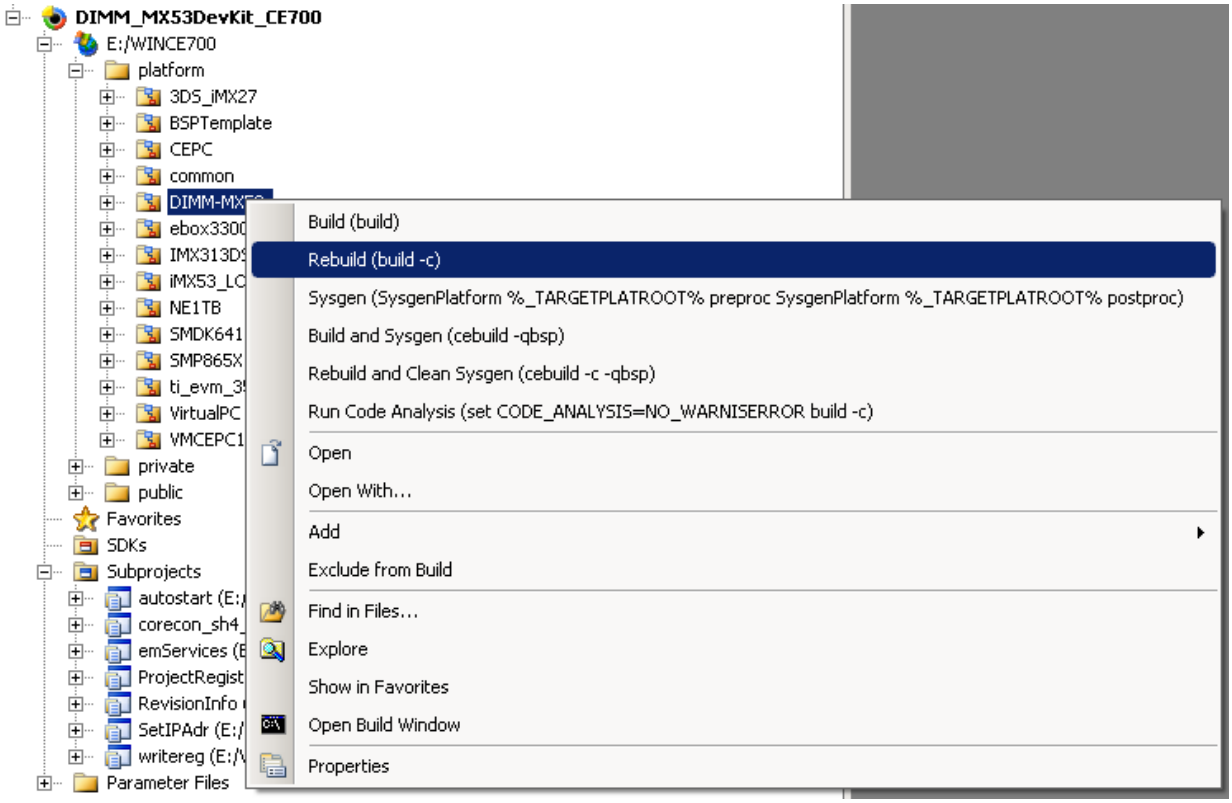
12.2.3 Incremental build

If only source code and build settings are modified an incremental build is sufficient.

At this point the checkboxes of [Global Build Settings] and [Targeted Build Settings] from the "Build" menu must be checked.

From the "Solution Explorer" select the appropriate component and right click on the mouse to open the context menu. Select build or rebuild to start the build process.

The figure below shows the incremental rebuild of the Developer Kit's Platform, for example due to changing of the build settings.



13 Licensing of the WINCE-Image

When a licensed WINCE-Image should be created, it is necessary to use the mechanism created by emtrion. This mechanism stamps the WINCE-Image at the end of the build process automatically before the required header for the bootloader is added. For that, a simple text file must exist. This text file contains the product key which you got from Microsoft's license label (you can use the same key for all items of the same device type if you stick a license label on each device).

If the text file with the product key found the batch file "create_UBOOT_header.bat" creates both types of images: a licensed and an unlicensed one. The output file names are wce-dimm-mx53 for the licensed one and wce-dimm-mx53-eval for the unlicensed one.

The name of the text file is defined to "image_pid_ce600" and its location is the same as for "create_UBOOT_Header.bat". Please look for the location in the figure in section Downloading the image to the target.

Example of the input format for the Product Key in "image_pid_ce600"

AAACD-FG8KJ-KL9NO-P2RST-UV5X3

Remark:
The key shown above is not a not valid key. You must use the key which you get from one runtime license label for Windows Embedded CE 6.

14 Downloading the image to the target

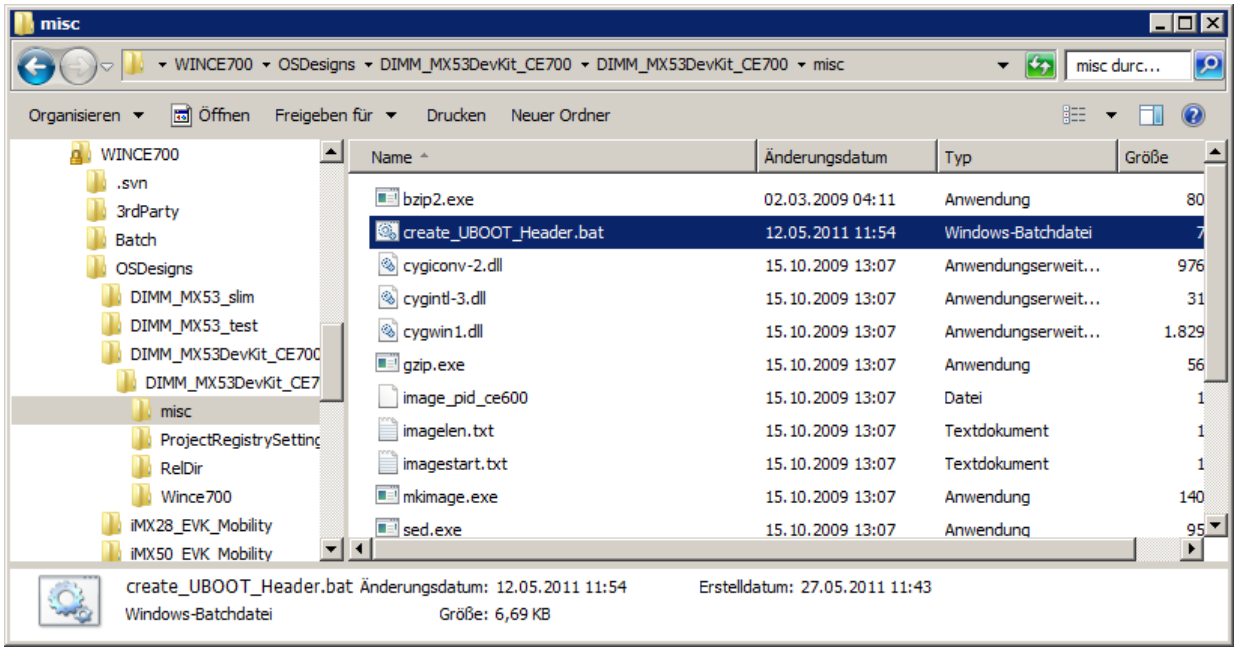
The build process in the previous sections creates the image of the OSDesign. Normally, the image file is nk.bin which will also be created. But this file cannot be downloaded to the target because the UBoot bootloader requires a special header to identify an image.

To satisfy the bootloader, the batch file “create_UBOOT_Header.bat” is running at the end of the build process. See also chapter “Custom Build Actions” above. It creates an extended image based on the origin one and adds a special header. Furthermore the image is renamed according to the conditions in the section “Licensing of the WINCE-Image”.

In the case if only an unlicensed WINCE-Image is created, the output name has manually to be renamed to the definition of bootloader’s environment variable “wimg”. This is normally the name of the licensed WINCE-Image.

Location of the extended image and batch file:

WINCE700\OSDesigns\DIMM_MX53DevKit_CE700\ DIMM_MX53DevKit_CE700\misc



14.1 Perform downloading

Downloading the image from host to the target board is described in section The Bootloader

15 Setting up the Platform Builder for target connection

A debug as well a release version can be debugged by the Platform Builder. For them, the image has to be nuilt with the build options

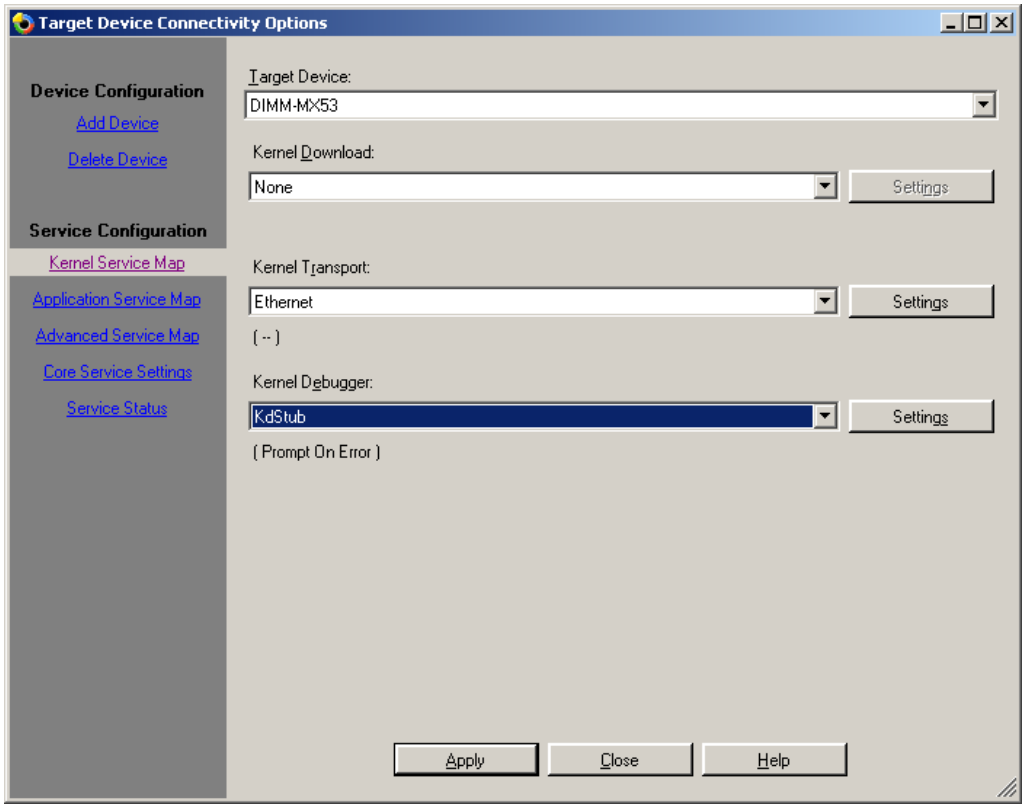
- Kernel Debugger
- KITL

enabled.

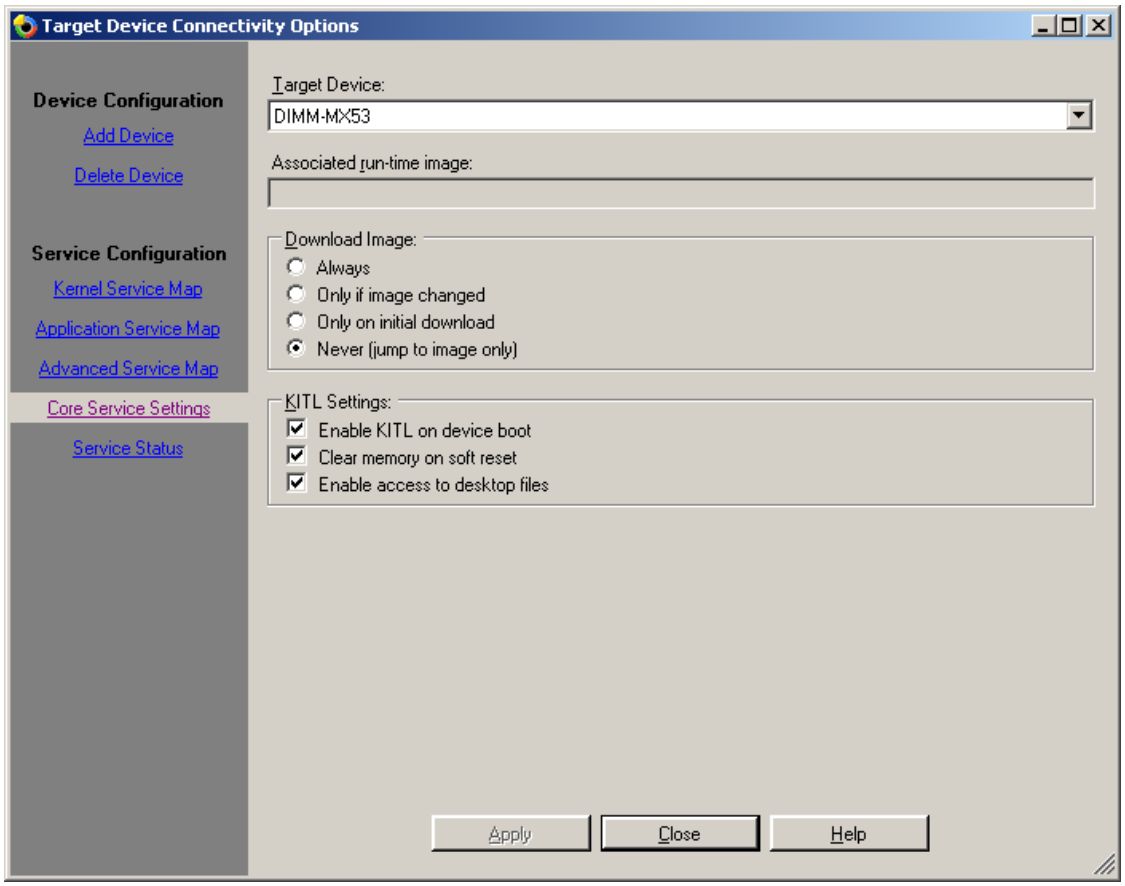
This section informs how you can establish a connection between Platform Builder and target for debugging purposes.

From the steps following below, the steps 4 to 10 have to executed only once.

1.) Make sure the OSDesign of the kernel is open.
2.) Build the image with Kernel Debugger and KITL enabled.
3.) Download the image to the target and store it in RAM
4.) From the "Target" menu of Visual Studio 2008 select "Connectivity Options…" to open the "Target Device Connectivity Options" dialog box.

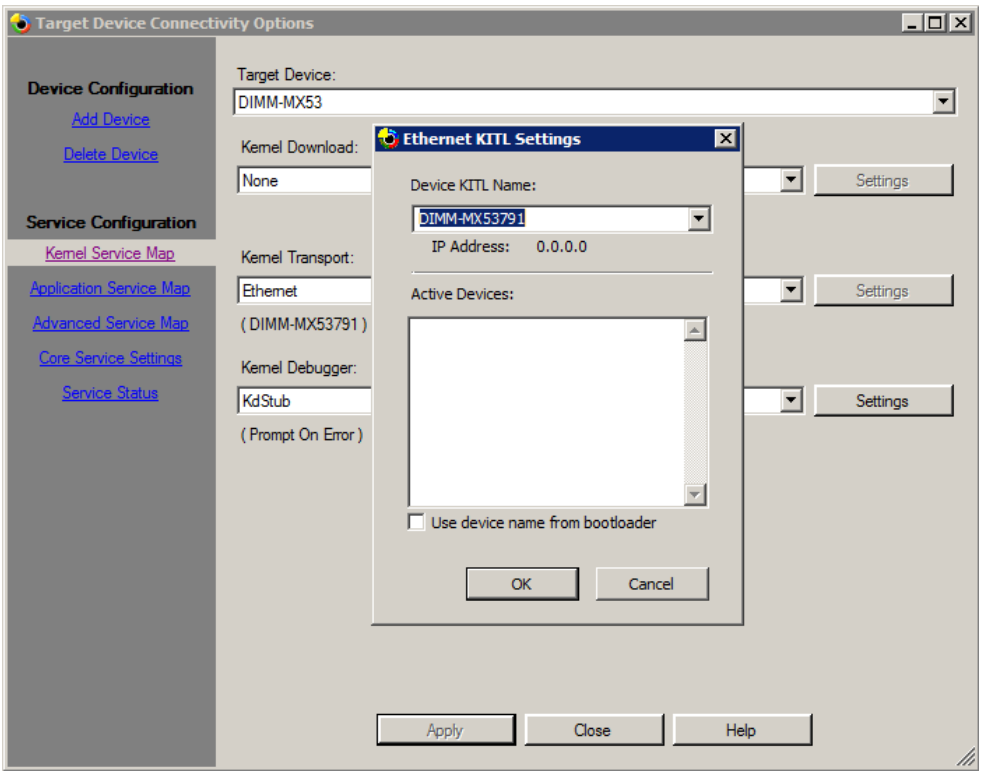


5.) Under “Service Configuration”, choose “Core Service Settings”. Make the settings for “Download Image” and “KITL Settings” as shown below and apply the changes.

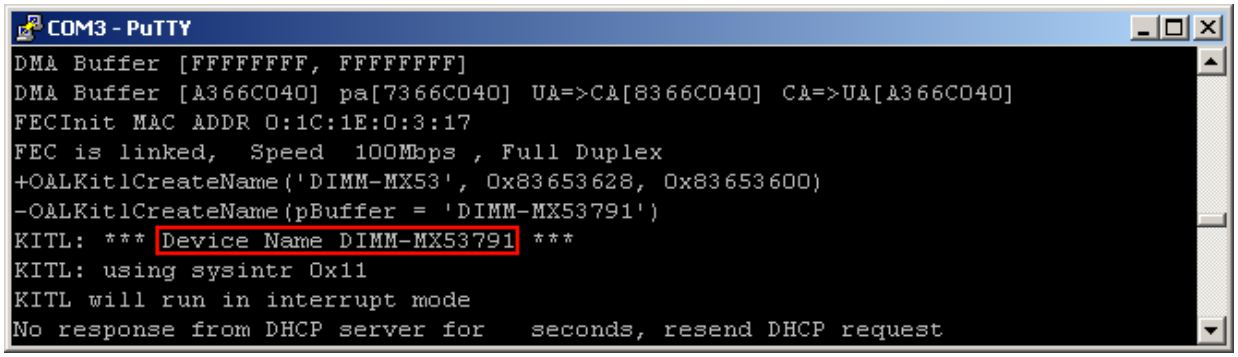


6.) Go to “Kernel Service Map”. Make the settings for the list boxes as in the dialog box in 4.) above.

7.) Choose the associated “Settings” button for “Transport”. The Platform Builder is waiting for the device. From the “Ethernet KITL Settings” dialog the device name of DIMM-MX53… is displayed as “DIMM-MX53” followed by numerals, which are the result of decimal conversion of the lower order two bytes of the LAN controller’s MAC address.



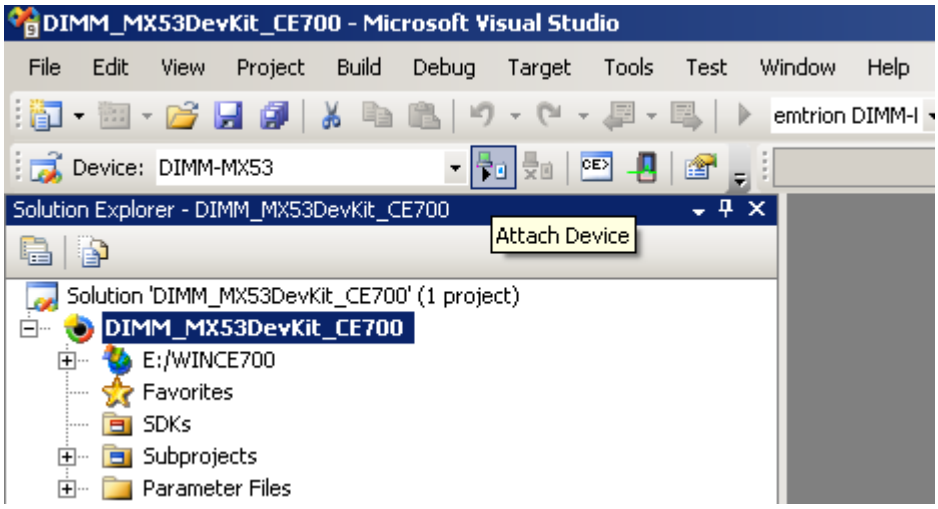
8.) When the KITL on the target has connected to the Debugger of the Platform Builder, the list "Active Devices" is updated by the device name of your target DIMM-MX53… . Then select the device name in the list "Active Devices". The IP address of the target appear below the list "Device KITL_NAME".
If "Active Devices" is not updated, when booting a WinCE Image, which includes KITL, you can also get the device name from the serial debug output as shown in the image below:



9.) Press "OK" button to leave the "Ethernet KITL Settings" dialog box.

10.) Apply the "Target Device Connectivity Options" dialog box and close it.

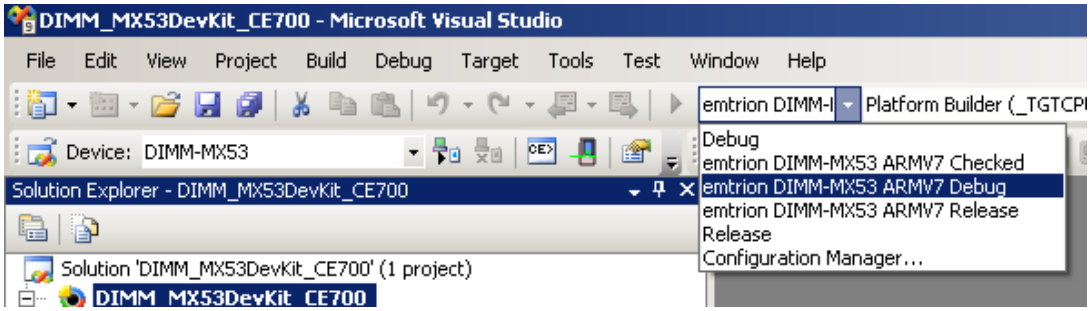
11.) From the “Target” menu or “tools bar” select “Attach Device”. The Platform Builder will connect with the Target.



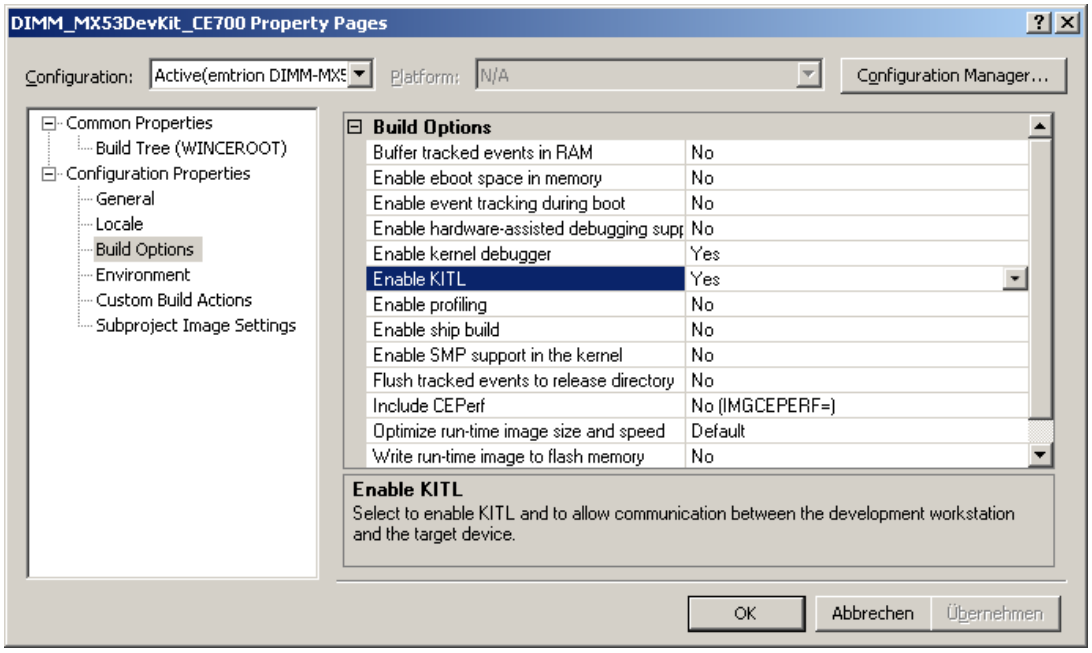
16 Building a Debug WINCE-Image (Kernel development)

This section informs about the option settings made for the Debug version of the OSDesign of the Developer Kit kernel.

1.) Make sure that the OSDesign of the Developer Kit kernel is open
2.) From the "Solution Configurations" field, select "emtrion DIMM-MX53 ARMV7 Debug"



3.) Open the Project Properties pages
4.) The language sections are the same as for release version in "Language settings"
5.) The build options differ to release version. "Kernel Debugger" and "KITL" are enabled.



6.) The "Custom Build Actions" are the same as for release version in "Custom Build Actions"
7.) Leave the Project Properties pages and build the image by following the steps in section Build .

17 Data Exchange between Target and Workstation

The Developer Kit kernel allows the exchange of data between the target and your workstation. For this, Microsoft's ActiveSync application is needed . The latest version is available on the internet at http://search.microsoft.com/search/results.aspx?st=b&qu=ActiveSync&view=en-us

If you have not worked yet with Microsoft's ActiveSync, we recommend to take a look at the section "17.4 Remarks on Using ActiveSync".

Windows Vista and Windows 7 do not support ActiveSync anymore. If you are using these versions of Windows please install the Mobile Device Center.

http://search.microsoft.com/Results.aspx?q=mobile+device+center

If ActiveSync is mentioned in the following sections, for Vista and Windows 7 users Mobile Device Center is meant.

17.1 Preparing the Desktop Computer

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

1.) Start the setup program and follow the shown instructions.

17.2 Preparing the Windows EC Device

USB function is the default setting for communication with ActiveSync. This communication is sufficient for file transfer and for application debugging.

When you do application development by Visual Studio 2008 communication via Ethernet is also possible. In this case USB ActiveSync can be used to get the IP address of the device.

17.3 USB ActiveSync

- Using a USB cable to connect the device to the workstation
- When not yet powered on, turn on the power on the target. The Windows® Embedded EC OS kernel which is stored in the flash will be executed. The running kernel will initiate an ActiveSync connection at the desktop computer. When the target is unknown as USB-Device at the workstation, Windows® will ask for the corresponding USB-fucntion-driver which has to be installed. The driver is located in the subdirectory USBFDriver on the DVD.
- When ActiveSync does not start automatically on the workstation then you must perform this manually. Please note, USB has to be selected in the connectivity settings of ActiveSync.

The target 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:
When the connection is **not** automatically established between the target and desktop, you should start the "repllog" application on the target manually. To do this, select "run" from the "Start" menu, then enter **repllog** and click the **OK** button.

The “New Partnership” dialog appears on the desktop computer.

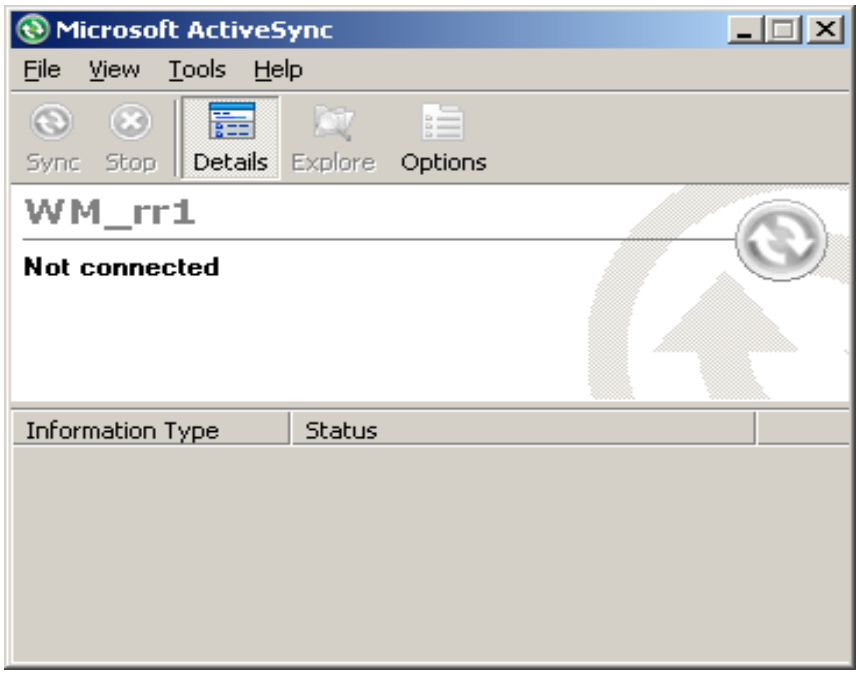
- Select the options as described in section “17.4 Remarks on Using ActiveSync”
- After establishing a connection, click Explore to open a window. Via this window, you may then exchange files with the target using drag and drop. For this, drag the files onto the desired directory on the target.

Note:
The folders of the kernel and their contents are created when Windows® Embedded CE is starting. IF you copy any files to these folders, they are not availabe any more when Windows® Embedded CE is restarted again. This does not apply to folders and files which are directly located at mass storages like “the “\FlashDisk” or the “\Storage Card”. These folders are available only if the corresponding devices are detected in the system during the start up phase. The subfolders of the “Network” folder are a special case. These folders represent drives when a connection via network was established.

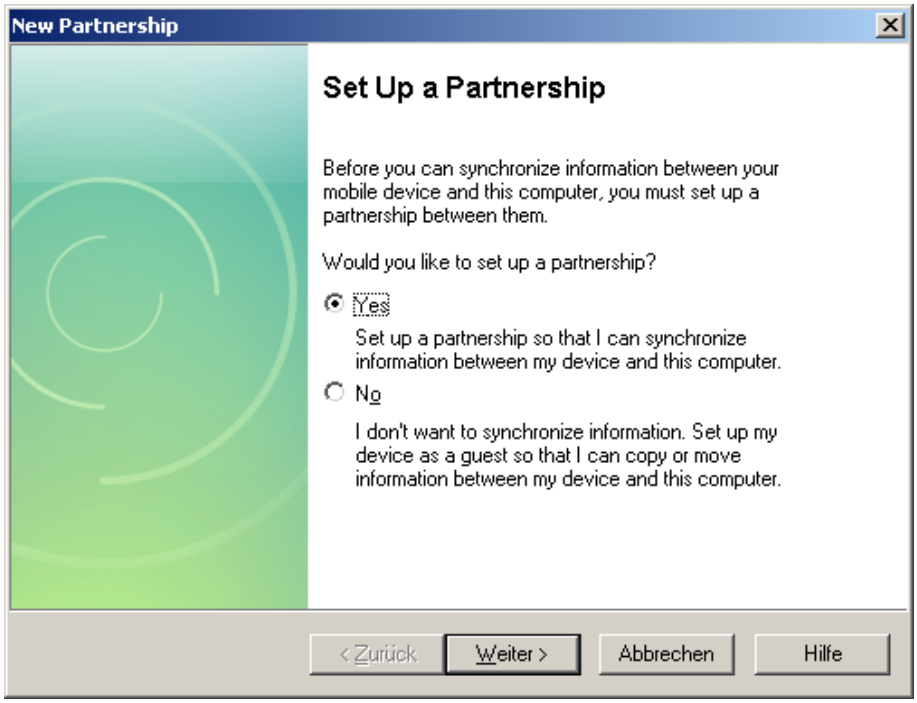
To terminate the connection, click the  symbol on the Windows ® Embedded CE task bar. The connection can be detached in the dialog that appears. Another easy way to stop ActiveSync is just to unplug the USB cable.

17.4 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 USB connection is enabled (“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 is shown:



The window shows that no partnership has been established between the Windows ® Embedded CE computer and the desktop PC.

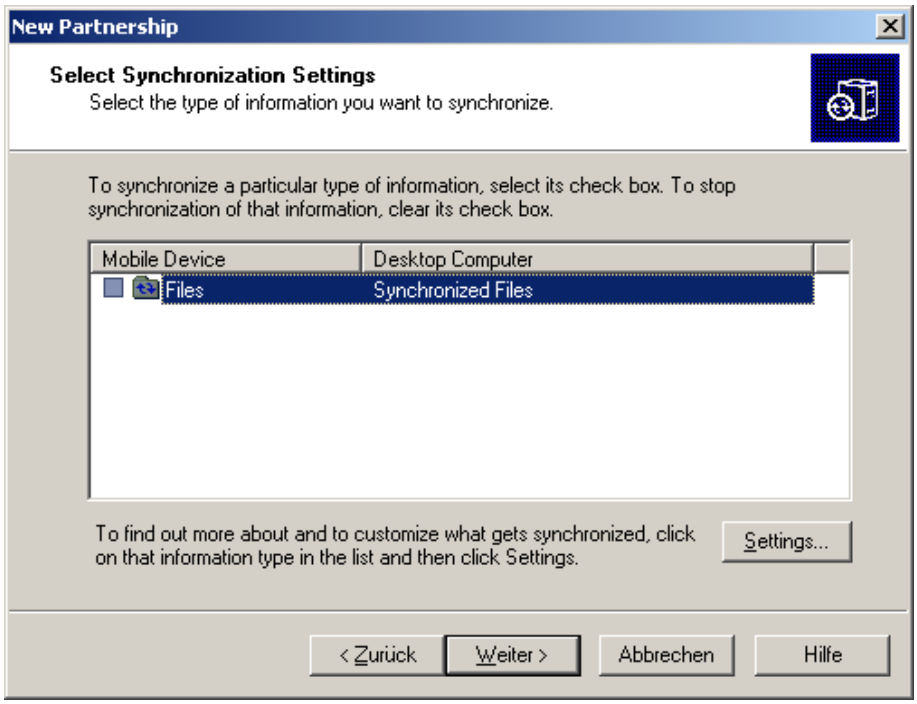
Hint:
If a partnership has already been created with a device with persistent registry, the dialogs displayed here do not show up.

Any of the offered options will be using the USB interface with following property.

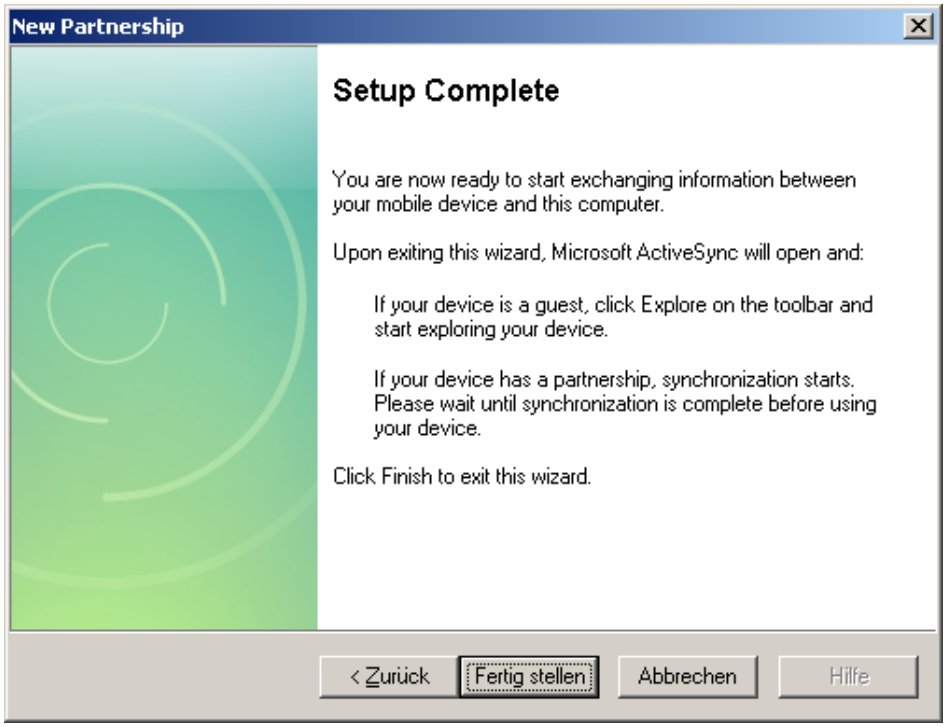
“No” means that the Windows® Embedded CE computer is connected with the desktop computer as a guest only. The connection as a guest is sufficient if you wish to exchange data or to debug an application via an USB connection.

“Yes” will establish a partnership, which is necessary for data synchronization.

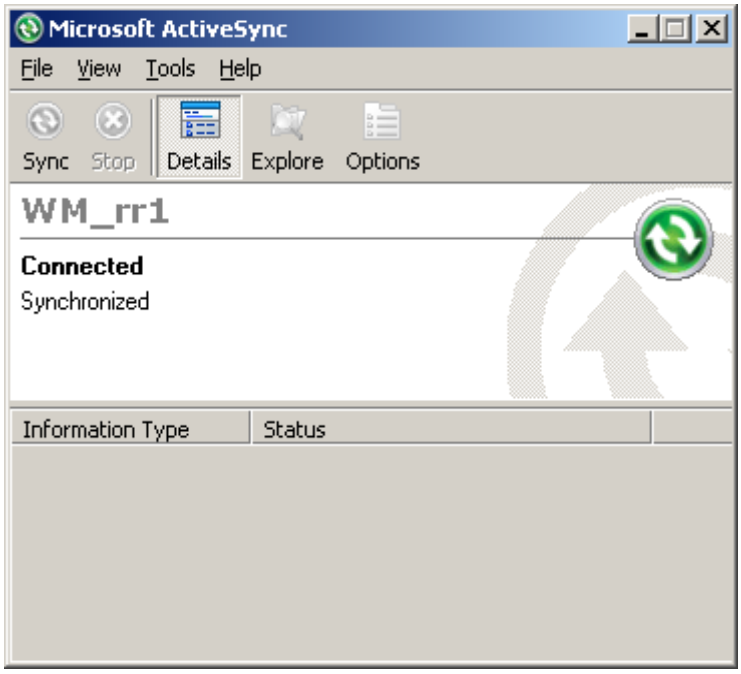
The following dialog appears:



Accept the default setting or deactivate "Files" by clicking on the small box left of "Files" and click "Next".



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



The connection has been established.

17.4.1 Data transfer

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

The data exchange between the Windows® Embedded 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.

18 Application Development

Application development is possible with Visual Studio 2008.

Visual Studio 2008 supports unmanaged and managed code and programming languages like C++, C#, Visual Basic and more are available. The communication interface for debugging can be Ethernet as well as USB ActiveSync. Sole Ethernet communication without a little help from USB ActiveSync is possible, but needs some special tools instead. For a more detailed description see the next chapters. We recommend a combination of Ethernet and USBF and all the kernels are supporting this.

18.1 VS2008 for Application Development

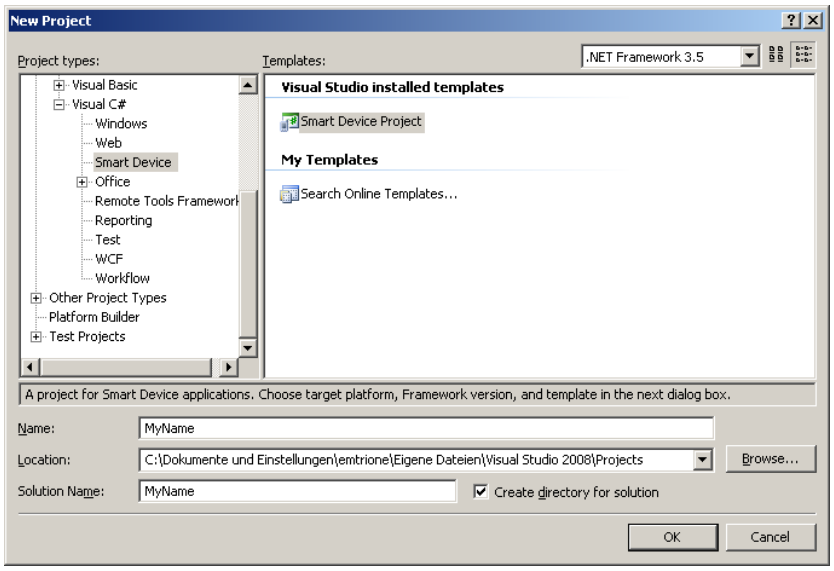
When you using Visual Studion 2005 for application development based on a specific kernel some hints shall be considered.

- Since Win32-API functions may be omitted in the creation of operating system kernels under Windows® Embedded CE, there is no SDK (Software Development Kit) which suits all Windows® Embedded 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.
- Please consult the online help of Visual Studio (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. DIMM-MX53DevKit_CE700_SDK for Developer Kit kernel) 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® 9x/NT/XP/Vista/7. With some functions certain parameters must be set to 0. For more information, please refer to the online help.

18.1.1 Creating a new Managed Project

The first step is starting a new instance of VS2008.

4. Select **File/NewProject** ... from the Visual Studio menu.
5. In the **NewProject** window select **Visual C#/SmartDevice/WindowsCE5.0**
6. Name your project **MyName** and click **ok**.



In the second windows you now select **DIMM-MX53DevKit_CE700_SDK** as **Target Platform**. Also select **.NET CF Version 3.5** and the **Device Application** template. Acknowledge by clicking **ok.**

Now you can start developing your application.

18.1.2 Deploying to the target

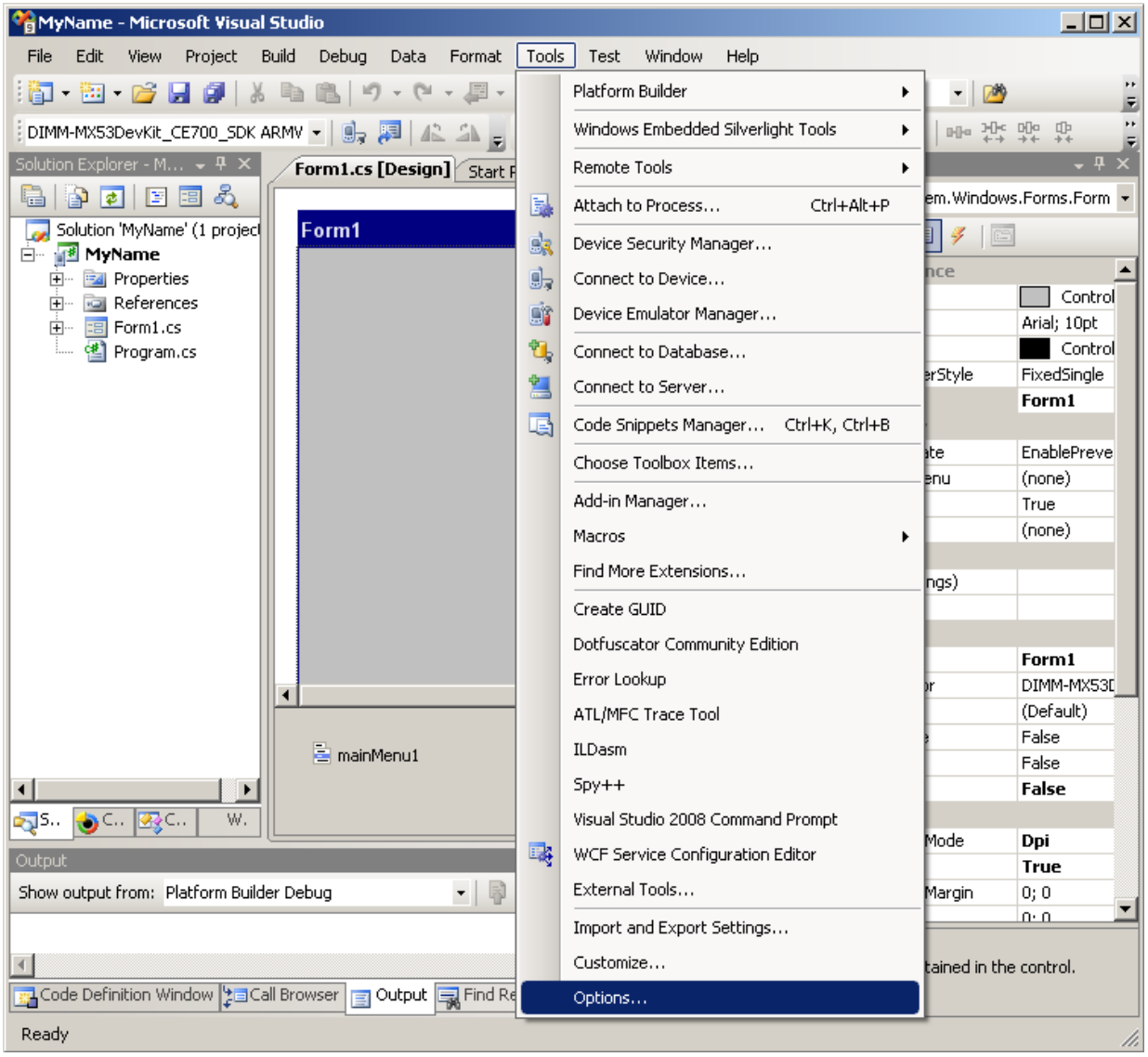
The connection between VS2008 and the target is made by network. For this reason the IP address of the target is necessary. To make the IP address known to VS2008 there are two options.

The first option is to ask manually by means of the function „ipconfig“ on the device. For that “ipconfig” must be executed from the command shell.

The second is receiving the IP address automatically by an existing USB ActiveSync connection.

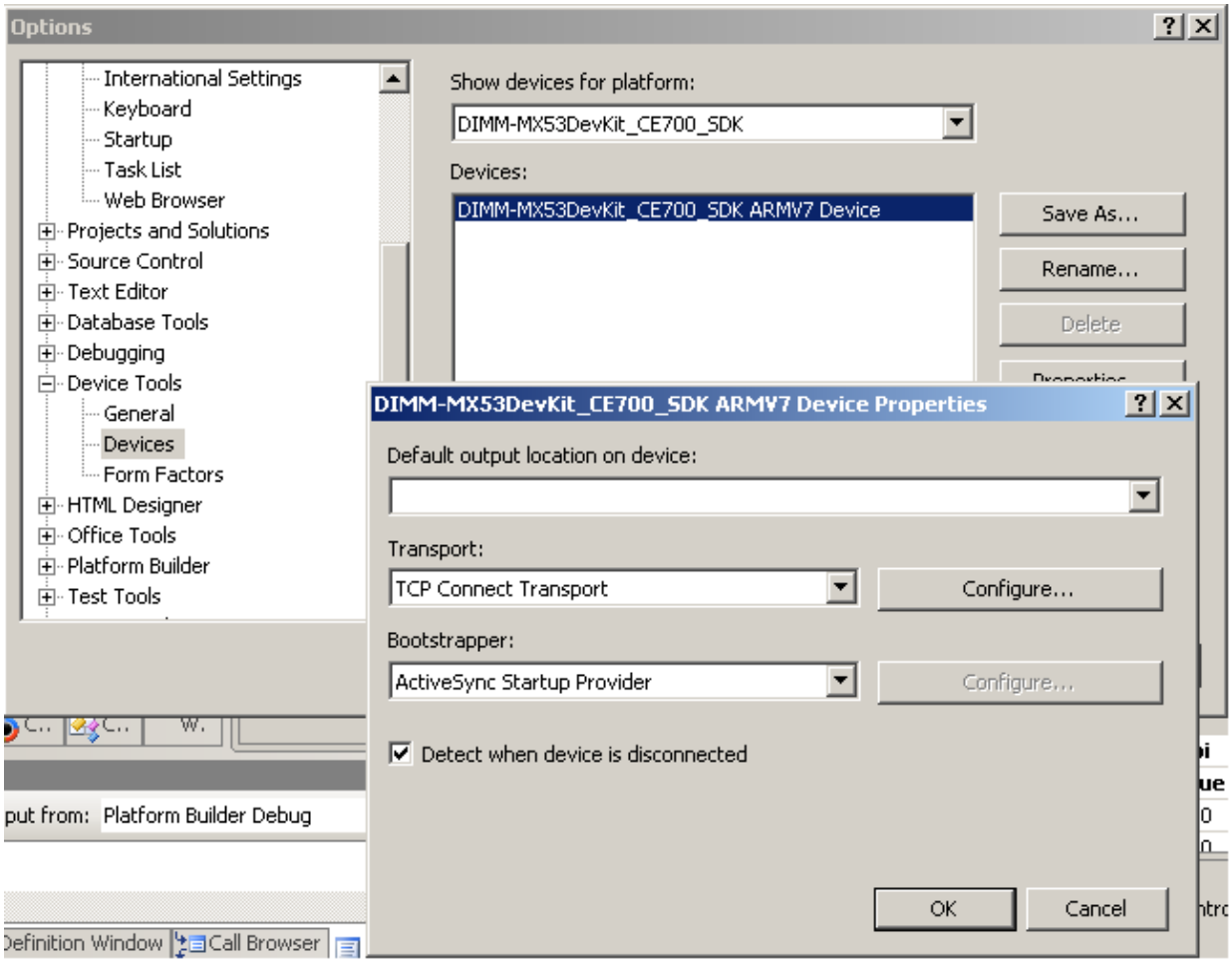
After getting this address, the next step is preparing the managed application development environment for deployment.

8. In VS2008 select **Tools/Options** from the menu.

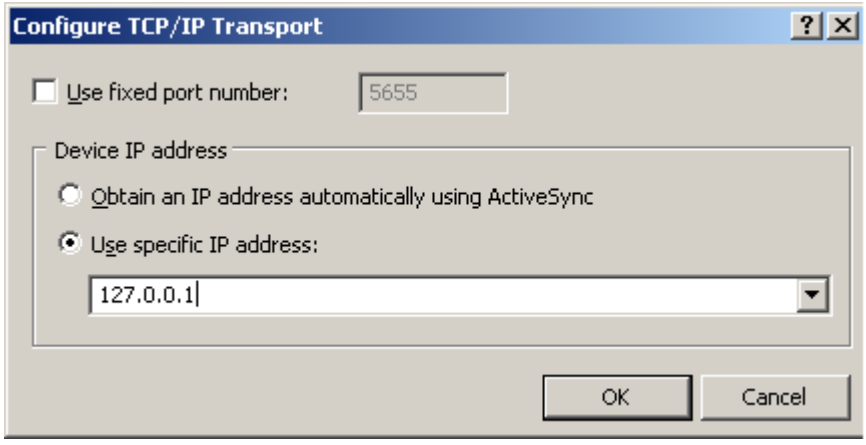


9. In the options window, expand the **Device Tools** node and select **Devices**.
10. In the **Show device for platform**: drop down box select **DIMM-MX53DevKit_CE700_SDK**.

11. Click on **DIMM-MX53DevKit_CE700_SDK** and select **Properties.**



12. Click the **Configure** button beside the **Transport** drop down box. We are going to configure the TCP Connect Transport.



13. In the case when an USB ActiveSync connection exists between the workstation and the device, the option “Obtain an IP address … is to select. In the other case the option “Use specific IP address” is to select and the IP address you got on the device by ipconfig has to be inserted.
14. Click **OK** through all of the dialogs.

Preparing the target:

3. At the command shell, type **ConmanClient2**.
4. Then, type **cmaccept**. You have 3 minutes to establish a connection with your managed application.

Deploying the managed application:

3. Select **Debug/Start Debugging** from VS2008.
4. Select **DIMM-MX53DevKit_CE700_SDK** from the list of devices in the **Deploy MyName** box and click **Deploy**. VS2008 will eventually deploy several cab files to the device in addition to your application. Your application is executed on the target device.

19 Starting an application at system start

There are two options to start an application at start up. Using only one option is allowed at any time. In any case, the execution of the explorer is controlled by emtrion's autostart mechanism described in the section below. This means no launch entry may be existed in the registry for the explorer.

To use the autostart functionality the registry must be modified. For it you can use the "Remote Registry Editor" of Visual Studio. For making the changes persistent, the emtrion tool "WriteReg.exe" is available.

Additonal items have to be considered when you want to realize starting your application at system start:

- Folder structures and shortcuts get lost at power off or reset when it has been created at runtime.
- The same as above happens with files that have been copied to the object store of the kernel.
- Changes to the registry also getting lost when it have not been made persistent.
- Due to supporting autostart of applications from a storage device, the kernel is configured with search paths of any supported storage device.
- Dependent on the supported storage devices of the Developer Kit the storage folders are specified as following:
- \SD Card for storage device microSD or SD card
- \MultiMediaCard for storage device MMC
- \Hard Disk for storage device USB stick
- \FlashDisk for storage device NOR flash
- \NAND flash for storage device NAND

19.1 Emtrion's autostart mechanism

One possibility to start you application at system start is using the autostart mechanism from emtrion. Therefore an autostart application was written and included into the kernel.

The registry of the Developer Kit kernel includes a launch link to that autostart application and is starting it at system start. Additionally, a shortcut of the autostart application is located in the "StartUp" folder of the explorer, so the startup mechanism of the explorer is also useable. Now, the autostart application itself looks in a specific registry key whether or not what application shall be started. The registry key includes several entries.

But is not the only task of the autostart application is also controlling the execution of the explorer.

The autostart application looks in

```
[HKEY_LOCAL_MASCHINE\Software\emtrion\AutoStart]
```

for the entry "CustomerApp". The entry has the type REG_SZ. The entry provides the filename and if necessary the path of the application that shall be started. When the entry is empty no further action is taken. In the case of a valid entry the autostart application tries to start the specified application.

When a path is not stated, the system searches in the folder "Windows" and in the folders of the supported storage devices of the Developer Kit, mentioned in the introduction of this chapter.

The entry "AppParameter" can be found in the same registry key. The entry has also the type REG_SZ. This entry specifies the command line parameters of the application.

Optionally, the same registry key may also contain an entry called "WaitCycles". This entry, a type of REG_DWORD, lets you extend the time that the autostart application is waiting for the directories to be linked, e.g. a FlashDisk. If the entry is not found a default value of 100 is taken.

Optionally, the same registry key may also contain an entry called "FailureOptions". This entry has also the type REG_DWORD and is bit-coded. Please set only the bits that are really necessary. The meaning of the individual bits are shown in the following table:

Bit No.	Description
0	Setting this bit, a message is displaying in a message box, when the custom application cannot be started.
1	With this bit you can specify the message showing in the message box. 0: "Cannot start the application which was specified to the launch at startup" 1: more detailed information is showing, like the name and search path of the application.
2	Setting this bit, the autostart application doesn't exit and is showing continuously the message when the custom application cannot be started
3	Setting this bit, the explorer is started when the custom application cannot be started.
4	Specifies the start up process for the custom application 0 -> the explorer is not started and the custom application is started at launch time of the autostart application 1 -> the explorer is started and the custom application is started by the startup mechanism of the explorer
5	Setting this bit, the explorer will be started in any case
6	Setting this bit when the kernel is a headless one. Due to no graphic exists, the console is started instead.
7..31	Reserved for future extensions.

A further entry is "Timeout". On that you can specify the time between two consecutive WaitCycles. The type of Timeout is REG_DWORD.

19.2 Autostart with LaunchXX and DependXX

This section describes how you can modify the kernel in order to start your application at system start by the two registry entries LaunchXX and DependXX.

Using this option, be sure the registry key "CustomerApp" in "19.1 Emtrion's autostart mechanism" is empty. In the case of using the explorer by the kernel, the registry key "FailureOptions" has to be set to 00000020h for execution.

The application and its own DLLs that shall be started have to be stored in one of the supported storage devices.

Here are the settings to do on the registry key

```
[HKEY_LOCAL_MASCHINE\init]
```

- Entry 1:
  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 the Developer Kit kernel
  2.) Type: REG_SZ
  3.) Value: Name of the file that shall be executed.
- Entry 2:
  1.) Name: DependXX, where XX must be the same two-digit number as specified with entry 1.
  2.) Type: REG_BINARY
  3.) Value: Hexadecimal digits that specify the dependency on a previously started application.

Please also not the chapter “The Persistent Registry”.

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 that the mass storages are entered in the folder structure. This must be taken into account in the start sequence.

20 Sample Applications

20.1 SKitApp

The sample application SKitApp is a simple terminal application that transmits all user inputs via the selected serial port. At the same time, receiving data is possible and is displayed in the main window of the application.

The communication settings (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 developed with Visual Studio 2005 and can be found on the Developer Kit DVD in the folder \Samples\SampleApp\SKitApp.

20.1.1 How the sample application work

After the application has been started, the window will be created and the global variables initialized. The application shows the serial ports (COM1… COMx) but only the available serial ports are highlighted. 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 transmit it.

20.1.2 Changing the parameters via the menu

You are able to alter the serial port and the communication parameters in the "Parameters" menu. Each time a parameters 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. "UpdateThe Settings" also calls the "UpdatePortSettings" function, in order to return the parameters of the port in use.

20.1.3 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 has used.

20.2 Multimedia

To demonstrate the excellent multimedia characteristics of the i.MX53 you can find a Audio/Video-Player application integrated in the delivered Windows EC 7 Image. You can find it in the folder \Windows on the device. It is called **PlayWnd.exe**.

The playback is hardware accelerated, if the codec is among the list of supported codecs. (Please have a look at section Video processing)

20.3 2D and 3D acceleration

There are 2 small demo applications which make use of the integrated OpenGL ES and OpenVG hardware acceleration. Both can be found in \Windows.

20.3.1 cube.exe (OpenGL ES 1.1)

cube.exe renders a rotating cube and shows the frames per second in an additional command shell window.

20.3.2 tiger.exe (OpenVG)

tiger.exe renders a vector graphic which shows the head of a tiger. The head can be rotated and zoomed in and out by using a mouse. A rotation of the head can be performed by pressing "," or "." on the keyboard.

20.4 Camera Interfaces

To display the input of the two camera interfaces, two small applications are included in the Developer Kit: camapp1.exe and camapp2.exe

Both applications show the camera input in a window and can be used simultaneously.

Additionally there is the application emCamCube.exe. It is a combination of an Open GL ES 2.0 and a camera demo and shows a spinning cube with the camera picture mapped on two of its sides.

20.5 HiCOCAN Demo

There is also a demo of the HiCOCAN CAN Bus interface included. For detailed information please have a look at the DIMM-MX537 CAN-SW Manual.

20.6 Multi-Touch Demo

Starting with the OS Design revision v0300 the image includes Microsoft's sample application CETouchView.exe For the full functionality of the sample application it's required that the CETouchFilter.DLL is loaded during startup. The OS Design is equipped to load this DLL. To activate the loading of this DLL during system startup you have to modify three registry entries and reboot the system. These are the required steps:

1.) Modify the value of the registry entry flgnore under
`[HKEY_LOCAL_MACHINE\software\emtrion\AdjustRegistrySettingsDueToCurrentHwConfig\display\ET0700M06\4]`
from 0 to 1

2.) Modify the value of the registry entry flgnore under
`[HKEY_LOCAL_MACHINE\software\emtrion\AdjustRegistrySettingsDueToCurrentHwConfig\display\ET0700M06\98]`
from 1 to 0

3.) Modify the value of the registry entry flgnore under
`[HKEY_LOCAL_MACHINE\software\emtrion\AdjustRegistrySettingsDueToCurrentHwConfig\display\ET0700M06\99]`
from 1 to 0

4.) Execute the tool writereg to make the registry persistent

5.) Repower the device.

6.) After the system is up again you can start the demo tool CETouchView without any restrictions.

21 The Persistent Registry

By default Microsoft's Windows® Embedded CE does not support a persistent registry. That is, all modifications made to the registry while Windows® Embedded CE is running will not be kept when Windows® Embedded 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 kernel. They will be called by the operating system, in order to write the registry in hive-based form onto the Flash File System .

```
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 make the modifications persistent, 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 system also automatically writes changes back to the registry, but no exact statement can be made about this time interval. Therefore, use the "RegFlushKey" to ensure that the changes are saved.

The Developer Kit's operating system kernel includes 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.

The command line

```
writereg -default
```

sets the registry to its default values, this means that all changes are lost and the settings of the default registry are used. 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). But always the complete registry will be saved to
flash */
If (retWert != ERROR_SUCCESS)
{
      /* Error handling */
}

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

21.1 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

21.1.1 Deleting the registry by the bootloader

The persistent registry can be deleted via bootloader as described in section Reset Windows CE Registry .

21.1.2 Deleting the registry by an application under Windows Embedded CE

A Windows CE application can be enabled to use the persistent registry by calling the Win32-API function KernelIoControl. For this purpose, emtrion 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 ware defined by Microsoft in the winioctl.h header file. This file is usually linked to applications with the

```
#include <windows.h>
```

statement.

21.1.2.1 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.

Parameter	Description
lpInBuf, nInBufSize, lpOutBuf, nOutBufSize	These parameters will not be analyzed.
lpBytesReturned	Pointer to a UINT32(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.

21.1.2.2 Documentation of the KernelIoControl function (extraction from the online help of Platform Builder)

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

Function prototype:

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

Parameter	Description
dwIoControlCode	I/O control code, which should supported the OAL I/O controls.
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 value:

Return value	Description
TRUE	indicates success
FALSE	indicates failure

```
Remarks:
The kernel calls the OEMIoControl fucntion when a device driver or
application calls the kernel function KernelIoControl and passes an I/O
control code. 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.
```

22 Emtrion Tools for Windows Embedded CE

In order to do various settings or get information, emtrion provides a variety of tools running on the Windows Embedded CE computer. Some of the tools can be found in “Start” menu -> “Programs” -> “emtrion Tools”.

22.1 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 starting, it writes the persistent registry by calling the Win32-API function RegFlushKey().

When it is called with the option “-default” the persistent registry is deleted and the OS is starting with its internal registry at next start.

Note:
User inputs will be of no significance while the persistent regsitry is being written.

Second Note:
You can call this tool also from a console window. Then you can force to output all messages on the command line interface instead in dialog boxes if you execute this tool with the optional command line parameter *–headless*. Instead using this command line parameter –headless you can set the optional registry entry EnableHeadlessMode under HKLM\Software\emtrion\Writereg to 1 (DWORD value).

22.2 Getting the Version of the Operating System Kernel

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

The following information is displayed in a dialog box:

- Name of the operating system kernel (= name of the OSDesign), e.g. DIMM-MX53_CE700
- Version number of the operating system kernel, e.g. V1
- Release date of the operating system kernel, e.g. 2011/26/02

Note:
You can call this tool also from a console window. Then you can force to output the information to the console window instead in a dialog box if you execute this tool with the optional command line parameter *–headless*. Instead using this command line parameter –headless you can set the optional registry entry EnableHeadlessMode under HKLM\Software\emtrion\RevisionInformation to 1 (DWORD value).

22.3 SetIPAdr

This tool is not displayed within the folder structure stated above. The tool setipadr is a console application and is equipped with some command line options. You can get current information about the network adapter or you can set some properties like the IP address or the subnet mask.

Calling the tool without any options shows a list of all the available network adapters:

Option	Value	Description
-a	Name of the network adapter	Displays the current network settings of the specified adapter
-d	Specify 0 or 1	Disables or enables the DHCP for the specified adapter in -a
-h	No value is required	Available parameters are listed
-i	Specify the IP address in dot notation	Sets the IP address for the specified adapter in –a
-n	Specify the subnet mask in dot notation	Sets the network mask for the specified adapter in –a

Example:

This command line disables the DHCP and set the IP address and network mask for the network adapter TEST:

```
setipadr –a TEST –d 0 –I 192.168.110.98 –n 255.255.255.0
```

23 Frequently Asked Questions (FAQ)

This section provides FAQs on the following

- In General

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 our support site http://www.support.emtrion.de/ to get 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 "24 Support" for details).

23.1 In General

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

  The MAC address of the target is hold in the environment variable "ethaddr" of the bootloader. From the bootloader prompt type printenv and then look for the specified variable. You will see the MAC address assigned to, like

```
ethaddr= 00:30:6C:90:00:10
```

- How to establish a network connection between a CE computer and a Windows® NT/2000/XP computer?
  Method 1
  1.) Open Internet Explorer on the Windows® Embedded CE computer.
  2.) Example:
  The "program files" 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\program files" to the address bar.
  3.) If the network is running, the contents of that drive appear.
- Method 2
  1.) Open a command line prompt under Windows Embedded CE
  2.) Enter the following command line:

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

  The command line parameters have the following meaning:

Parameter	Description
<localname>	Name with which the mapped subdirectory appears in the "\Network" directory
<remotename>	Network path in accordance with the UNC naming convention for the drive to be mapped. Example: The "program files" 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\program files" to the address bar
<uname>, <pword>	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® Embedded CE computer, there is now a new directory with the specified name. This directory represents the corresponding directory on the NT computer.

- 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 the 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.

- No DHCP available
  By default, the operating system kernel of the Developer 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 the Control Panel.
  2.) Select the corresponding network adapter
  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 target.
  10.) Check the network connection as follows:
    - Open the 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

23.2 Changing display and camera settings without (re)building the OSDesign

If you want to change the display settings between parallel display and DVI output you can do it quickly by changing following Registry value:

```
[HKEY_LOCAL_MACHINE\Drivers\Display\DDIPU\DI1]
   "PanelType"=dword:X
```

For **X** following values are valid:

X	Display type
2	Parallel display output using parameter provided by U-Boot
3	DVI output 800x600
4	DVI output 1024x768
5	DVI output 1280x1024
6	DVI output 1600x1200
7	DVI output 1920x1080
8	DVI output 1280x720

It is also possible to setup the camera driver for ADV7180 or VM009 by using the registry:

```
[HKEY_LOCAL_MACHINE\Drivers\BuiltIn\Camera1]
    "CameraId"=dword:X
```

For **X** following values are valid:

X	Display type
3	ADV7180
4	MT9M131 (VM009)

24 Support

This product has been thoroughly tested over the development period. Due to its complexity, however, no gurantee 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 support web site http://www.support.emtrion.de/ for the lastest updates. Please check also if there already updates available in the update repository (see chapter "10.2 Updating the BSP to the newest release" for details).

To accelerate the process, please fill out the supplied from, which can be found on our web site at http://www.emtrion.com/support_form_en.php

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

- Who sent the request (company name and the name of the writer)
- A valid email address, where we can send the answer
- The product name DIMM-MX53DevKit_CE700
- The product version and release date, which is printed on the CD (NOT the version of the Windows CE operating system).
- If you have updated the product from the update repository, the revision number of the version which you have downloaded
- A detailed description of the problem and how it can be reproduced
- Recommended: all applications which are required for reproducing your issue and which are not included in this Developer Kit.

25 License Agreement

25.1 Emtrion License Agreement

End User License Agreement

Emtrion GmbH Greschbachstrasse 12, 76229 Karlsruhe, Germany (hereinafter referred to as "**Emtrion**") provides to businesspersons which are "Unternehmer" in the sense of Sect. 14 BGB German Civil Code (hereinafter: "the **Customer**") certain Software to be run as embedded software on hardware boards (hereinafter: "the Software") subject to Emtrion's Terms and Conditions:

I. Definitions

1. **"Software**" shall mean the operating software to be embedded into the Hardware, including any documentation hereto.
2. "**Confidential Information**" shall mean any information, know-how and data owned or controlled by Emtrion or by Third Parties license Rights.

II. Limited Use Rights

With respect to Software, Customer is granted a non-exclusive, non-transferable, non-sub licensable, worldwide, license to use or have used the Software for validating or testing. Against an extra fee and an advanced EULA Customer can use Software for projects.

Customer shall not - and shall prevent others from doing so - copy, translate, modify, create derivative works, disassemble, reverse engineer, decompile or otherwise use the Software, except as specifically authorized hereunder in this item, or except provided otherwise by compulsory law. Whenever compulsory law permits the above actions, Recipient shall in any case

(i) inform Emtrion of these actions in advance, and

(ii) request support of Emtrion, which Emtrion may offer against reasonable consideration.

All rights granted to Software hereunder shall only apply to the code format to be provided as agreed upon, which is, if not explicitly agreed upon otherwise, solely the binary code to the Software.

Customer is aware that Software may consist of or contains third party software, distributed by Emtrion in altered or unaltered form. The use of such Software may be further restricted by additional terms and conditions (hereinafter also "**Third Parties License Rights**"), with priority to the terms set out herein.

III. ADDITIONAL RESTRICTION ON USE

THE OBJECT CODE, SOURCE CODE AND THIRD PARTY MATERIALS ARE NOT FAULT TOLERANT AND ARE 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 THE OBJECT CODE, SOURCE CODE AND THIRD PARTY MATERIALS COULD LEAD DIRECTLY TO DEATH, PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE.

IV. Confidentiality Restrictions

1. **Restrictions.** Without limiting and in addition to the use restrictions under Sect. II and Sect III above, all Confidential Information delivered pursuant to this agreement shall
   (a) be used by Customer for the Purpose and subject to Section II above only; and

   (b) be treated by Customer with the same degree of care to avoid unauthorized disclosure to any third party as with respect to Customer's own confidential information of like importance but with no less than the degree of care described in this agreement; and

   (c) not be distributed, disclosed, or disseminated in any way or form by Customer to anyone except its own employees, who have a reasonable need to know such Confidential Information with respect to the Purpose and who are bound to confidentiality by written agreements not less stringent than under the obligations of this agreement; and

   (d) be stored in secure areas only that are protected against unauthorized access and are regularly monitored to assure their sufficient security,

   (e) be treated by Customer in compliance with any additional security guidelines the parties may agree.

2. **Exceptions.** The obligations as per Sect. IV. 1 above shall not apply to any information which the Customer can prove,
   (a) is at the time of disclosure already in the public domain or becomes available to the public through no breach by the Customer of this agreement, except that Confidential Information shall not be deemed to be in the public domain merely because any part of

the Confidential Information is embodied in general disclosures or because individual features, components or combinations thereof are now or become known to the public;

(b) is received by the Customer from a third party free to lawfully disclose such information to Customer;

(c) was in the Customer's lawful possession prior to receipt from the Discloser as evidenced by written documentation;

(d) is independently developed by the Customer without the benefit of any of the Confidential Information as evidenced by written documentation;

(e) is approved for release by written agreement of Emtrion;

(f) is required to be disclosed to comply with legal mandatory regulations, a judicial or official order or decree, provided that written advance notice of such judicial action was timely given to Emtrion.

V. Prohibition on Development of Competing Products

Customer shall at no time during the term of this agreement and hereafter for a tem of five (5) years use Confidential Information for the purpose of developing or distributing products which are competing with Emtrion's Hardware as a stand alone product.

VI WARRANTIES.

Emtrion makes no warranties, either express or implied, with respect to the object code or source code or with respect to Software or other products or intellectual property obtained from Third Parties. Emtrion expressly disclaims any such warranties, including but not limited to the implied warranties of merchantability and fitness for a particular purpose.

25.2 Microsoft License Agreement

SOFTWARE LICENSE TERMS
WINDOWS EMBEDDED Compact 7.0 CUSTOMIZED OEM SOFTWARE DEVELOPMENT KIT for
Windows Embedded Compact 7.0 Developer Kit for DIMM-MX53, 02.12

These license terms are an agreement between you and *[emtrion GmbH]*. Please read them. They apply to the software included on this device. The software also includes any separate media on which you received the software.

The software on this device includes software licensed from Microsoft Corporation or its affiliate.

The terms also apply to any Microsoft

- Updates,
- Supplements,
- Internet-based services, and
- Support services

for this software, unless other terms accompany those items. If so, those terms apply. If you obtain updates or supplements directly from Microsoft, then Microsoft, and not *[emtrion GmbH]*, licenses those to you.

As described below, using some features also operates as your consent to the transmission of certain standard computer information for Internet-based services.

By using the software, you accept these terms. If you do not accept them, do not use or copy the software. Instead, contact *[emtrion GmbH]* to determine its return policy for a refund or credit.

If you comply with these license terms, you have the rights below.

1. **Use Rights.**

You may use the software on the device with which you acquired the software.

2. **Additional Licensing Requirements and/or Use Rights.**

   a. **Specific Use.** *[emtrion GmbH]* designed this device for a specific use. You may only use the software for that use.

   b. **Other Software.** You may use other programs with the software as long as the other programs
      - Directly support the manufacturer's specific use for the device, or
      - Provide system utilities, resource management, or anti-virus or similar protection.

3. **Scope of License.** The software is licensed, not sold. This agreement only gives you some rights to use the software. *[emtrion GmbH]* and Microsoft reserve all other rights. Unless applicable law gives you more rights despite this limitation, you may use the software only as expressly permitted in this agreement. In doing so, you must comply with any technical limitations in the software that allow you to use it only in certain ways. For more information, see the software documentation or contact *[emtrion GmbH]*. Except and only to the extent permitted by applicable law despite these limitations, you may not:
   - Work around any technical limitations in the software;
   - Reverse engineer, decompile or disassemble the software;
   - Make more copies of the software than specified in this agreement;
   - Publish the software for others to copy;
   - Rent, lease or lend the software; or
   - Use the software for commercial software hosting services.

   Except as expressly provided in this agreement, rights to access the software on this device do not give you any right to implement Microsoft patents or other Microsoft intellectual property in software or devices that access this device.

4. **Connectivity Software.** Your device package may include either CE Services or Active Sync software. If it is included, then you may install and use it in accordance with the end user license terms that are

provided with it. If no end user license terms are provided, then you may install and use only 1 copy of the software on a single computer.

Note on Compatibility of ActiveSync. ActiveSync software may not operate on your device. You bear all risk of using it.

5. **Product Support.** Contact *[emtrion GmbH]* for support options. Refer to the support number provided with the device.
6. **Backup Copy.** You may make one backup copy of the software. You may use it only to reinstall the software on the device.
7. **Proof of License.** If you acquired the software on the device, or on a disc or other media, a genuine Certificate of Authenticity label with a genuine copy of the software identifies licensed software. To be valid, this label must be affixed to the device, or included on or in *[emtrion GmbH]*'s software packaging. If you receive the label separately, it is not valid. You should keep the label on the device or packaging to prove that you are licensed to use the software. To identify genuine Microsoft software, see http://www.howtotell.com.
8. **Transfer to a Third Party.** You may transfer the software only with the device, the Certificate of Authenticity label, and these license terms directly to a third party. Before the transfer, that party must agree that these license terms apply to the transfer and use of the software. You may not retain any copies of the software including the backup copy.
9. **Not Fault Tolerant. The software is not fault tolerant. *[emtrion GmbH]* installed the software on the device and is responsible for how it operates on the device.**
10. **Restricted Use.** The Microsoft software was designed for systems that do not require fail-safe performance. You may not use the Microsoft software in any device or system in which a malfunction of the software would result in foreseeable risk of injury or death to any person. This includes operation of nuclear facilities, aircraft navigation or communication systems and air traffic control.
11. **No Warranties for the Software. The software is provided "as is". You bear all risks of using it. Microsoft gives no express warranties, guarantees or conditions. Any warranties you receive regarding the device or the software do not originate from, and are not binding on, Microsoft or its affiliates. When allowed by your local laws, *[emtrion GmbH]* and Microsoft exclude implied warranties of merchantability, fitness for a particular purpose and non-infringement.**
12. **Liability Limitations. You can recover from Microsoft and its affiliates only direct damages up to two hundred fifty U.S. Dollars (U.S. $250.00). You cannot recover any other damages, including consequential, lost profits, special, indirect or incidental damages.**

    This limitation applies to:

    - **Anything related to the software, services, content (including code) on third party internet sites, or third party programs, and**
    - **Claims for breach of contract, breach of warranty, guarantee or condition, strict liability, negligence, or other tort to the extent permitted by applicable law.**

    It also applies even if Microsoft should have been aware of the possibility of the damages. The above limitation may not apply to you because your country may not allow the exclusion or limitation of incidental, consequential or other damages.
13. **Export Restrictions.** The software is subject to United States export laws and regulations. You must comply with all domestic and international export laws and regulations that apply to the software. These laws include restrictions on destinations, end users and end use. For additional information, see www.microsoft.com/exporting.

26 Literaturverzeichnis

1. **Freescale semiconductor.** *i.MX53 Multimedia Applications Processor Reference Manual Rev. 1A.* 03 2011.

2. **Freescale semiconductor**. *Windows Embedded Compact 7 BSP for i.MX53 SMD Reference Manual.* 2011.