User's Manual

Microsoft Windows XP Embedded

Syslogic User's Manual

Document Ordercode: IPC/WINXPE-6A

Revision	Datum	Author	Modification
1.0	05.01.2006	Stefan Wyss	-
1.1	06.11.2006	Stefan Wyss	Minor mods
1.2	02.02.2007	Stefan Wyss	Feature Pack 2007 integration
1.3	03.08.2007	Stefan Wyss	WinIO integration
1.4	10.09.2007	Stefan Wyss	Updated trademarks & licenses
1.5	04.02.2008	Stefan Wyss	2GB flash transition
1.6	08.08.2008	Stefan Wyss	NETSBC, NETIPC-6L Support added

Content

List of figures

1 General Remarks

The content and presentation of this document has been carefully checked. No responsibility is accepted for any errors or omissions in the documentation. Note that the documentation for the products is constantly revised and improved. The right to change this documentation at any time without notice is therefore reserved.

Syslogic is grateful for any help referring to errors or for suggestions for improvements.
The following registered trademarks are used:

IBM-PC, PC/AT, PS/2	trademarks of IBM Corporation
Windows XP Embedded, XPe	trademark of Microsoft Corporation
FBWF	trademark of Microsoft Corporation
IPC/NETIPC	trademark of Syslogic Datentechnik AG
TSHARC	trademark of Hampshire Company
FILEMON	trademark of Sysinternals
WinIO	trademark of Yariv Kaplan

2 Introduction

Windows XP Embedded SP2 Feature Pack 2007 (short XPe) is the componentized version of Microsoft Windows XP Professional. In fact, both Operating Systems are based on the same binaries.

XPe is enhanced with special tools to target the needs of embedded systems that run continuously. While a desktop OS like XP Professional is optimized for maximum user interaction, an embedded OS is the opposite. Embedded systems often run in environments where few user interactions are desired. Embedded systems often supervise safety critical facilities and thus need to be very robust, reliable and autonomous.

This user manual covers some tools and techniques that help to reduce possible software/hardware breakdowns concerning the operation of Windows XP Embedded on a IPC/NETIPC or IPC/NETSBC based CPU in an industrial environment. It details some embedded features that can be configured.

2.1. Boards Supported

This release of XPe supports the following boards from Syslogic:

- IPC/NETIPC-6L (128MB RAM, no Audio)
- IPC/NETIPC-6A (256MB RAM)
- IPC/NETSBC-6x (512MB RAM, no Audio)

3 Getting Started

This chapter will give you a "quick start" on how to get XPe running on a IPC/NETPC-6A CPU board. The same procedure applies for IPC/NETSBC and IPC/NETIPC-6L boards.

3.1. Booting XPe

Together with the Starter Kit, you should have received a Compact flash card that is prepared with a full, preinstalled XPe image. Open your industrial PC (IPC) and plug the Compact flash card into the corresponding socket of the IPC/NETIPC CPU board. Close the IPC and connect a PS/2 or USB keyboard and a monitor. Power the IPC and press and hold the "*Delete*" key on the keyboard to enter the system BIOS screen. Select "*Basic CMOS Configuration*", where you have to enter the following settings:

CMOS Setup:

Boot 1st:	Drive C:
Ide 0:	3 = AUTOCONFIG, LBA

Hit "ESC" to leave the configuration screen and select "*Write to CMOS and Exit*" (confirm with Y or Z key). Now, the system should boot into Windows XP Embedded. You do not always have to change these CMOS settings when booting XPe, as these CMOS settings are stored permanently if (and only if) the battery is inserted and operational. If you change the battery, you must redo the correct settings.

3.2. User and Password Settings

The Syslogic XPe image is configured for Autologon, i.e. if the system starts, there's no login screen showing up and the default user "Administrator" is automatically logged on.

User and Password:

XPe Autologon User:	Administrator
XPe Autologon Password:	netipc

3.3. Drive Write Protection

The major drawback of Compact flash cards is their limited write cycles of typically 100'000 operations per block before breakdown. Thus it's extremely important to protect the Compact Flash card in terms of write accesses.

This protection is achieved through the File Based Write Filter (FBWF) shipped with Windows XP Embedded[1]. If FBWF is enabled (which is not the default in the Syslogic XPe Image), all write operations to the system drive C: except the ones to the data directory are redirected to a RAM layer, thus the Compact Flash is protected against writes (see figure 1).

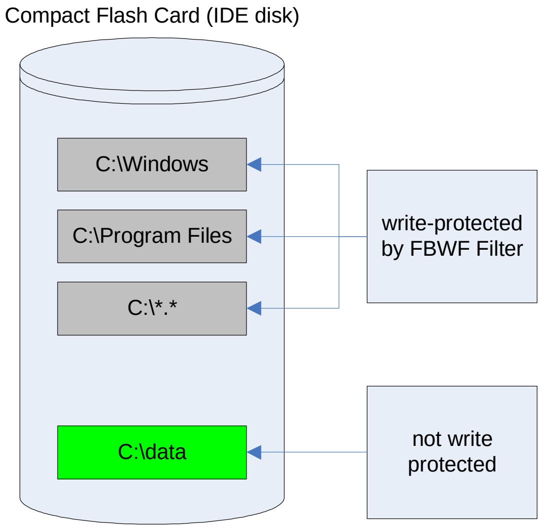


figure 1 – Compact Flash FBWF setup

Per factory default the FBWF is disabled. So if you boot into XPe and change some system settings, these settings will persist between reboots.

Note:
If FBWF is enabled, system and drive C: settings are write protected, except the files in the folder C:\data

If you create a file C:\Windows\myfile.txt with FBWF enabled, this file will not exist after the next reboot. We heavily encourage you to read the following sections, especially the section on the “File Based Write Filter”, to get further information on how to enable/disable and use the FBWF features.

[1] FBWF is the successor of the „Enhanced Write Filter“ (EWF). It was first introduced in Feature Pack 2007 for Windows XP Embedded.

3.4. Touchscreen or PS/2 Mouse (exclusive)

The XPe Image has PS/2 Touch Screen enabled as a default. The touch screen function is implemented by the "Tsharc" touch driver that is installed in the *C:\TouchDriver* folder of XPe. If you want to use a PS/2 mouse instead of the PS/2 touch driver, you must uninstall the touch driver (*C:\TOUCHSCREEN\uninstal.exe*).

Note:

PS/2 Touchscreen and PS/2 mouse can not coexist.

Of course, you must first disable the FBWF drive C: protection (see section 5.2) in order to make the changes permanent. Also don't forget to re-enable the FBWF protection after uninstalling the touch screen driver. Please note that the USB driver for the mouse is not affected and can coexist with either PC/2 mouse driver or PS/2 Touch Screen driver installed.

If you want to calibrate the PS/2 touch screen, you can use the calibration tool C:\TOUCHSCREEN\hwincal.exe. Please note that, again, you have to disable the FBWF drive C protection before doing the calibration.

Note:

Disable FBWF when installing/uninstalling or calibrating the touch screen.

3.5. Audio

Only the IPC/NETIPC-6A has an audio chip assembled and the audio features of XPe must be enabled prior to using the audio features of the IPC/NETIPC-6A.

Open ***Start / Settings / Control Panel / System / Hardware / Device Manager*** and click on the ***Sound, video and game controller*** node.

Then right click on ***the GeodeLX Audio Driver (WDM)*** and select ***enable*** to activate the audio driver in XPe.

4 XP Embedded Standard Image Features

The software included in our Windows XP Embedded image is a subset of the software shipping with Windows XP Desktop edition. From 14'000 selectable packages, there are around 750 currently included into our standard XP Embedded configuration.

So do not expect to have the full Desktop XP Pro feature set included in out standard image. But be also aware of the fact that you can install your own missing drivers and software packages at will like on a regular Windows XP system.

Below you will find a list of features that are included/not included in our standard configuration. This list is not complete, it may only give you an overview of the basic software bundles that are shipped with our XPe standard image.

Features included:

- Windows XP user interface, full Desktop with Taskbar and start menu
- Command Shell, Control Panel, CMI Scripting Interface
- Internet Explorer 6.0, Windows File Explorer
- Microsoft .NET 2.0 CLR
- Microsoft Installer
- Windows Media Player 9.0
- Remote Desktop Connection, Hyperterminal
- Remote Folder Share, Drive Mapping, basic Networking utilities
- Drivers for Board Hardware (Mouse & Keyboard PS/2 & USB, Flash-Memory-Stick, Video, Sound).
- Touch Screen driver and calibration utility.

Features NOT included:

- Java engine
- Most user applications like Microsoft Paint, Games, etc.
- Windows Update Agent
- Language Support for Languages other than English & German**.**
- Drivers for additional hardware devices.

Please consult the XPe build logfile *NETIPC6_XPE_Vnnn.log* from the product DVD-ROM to see the full list of packages which are included into the XPe release.

5 File Based Write Filter

5.1. Overview

Compact flash cards are a very popular storage for data because of their low price, small form factor and robustness (e.g. no moving parts, high temperature operation). The major drawback of compact flash cards is their limited write robustness of typically 100'000 operations per block before breakdown. The Compact flash card is not wear resistant against excessive write cycles.

Thus it's extremely important to protect the Compact Flash card in terms of write accesses. The File Based Write Filter "FBWF" shipped with Windows XP Embedded protects the flash card. If the FBWF is enabled, every data written to drive C: is actually redirected to a RAM layer, thus the Compact Flash is protected against writes (see figure 1).

Important:

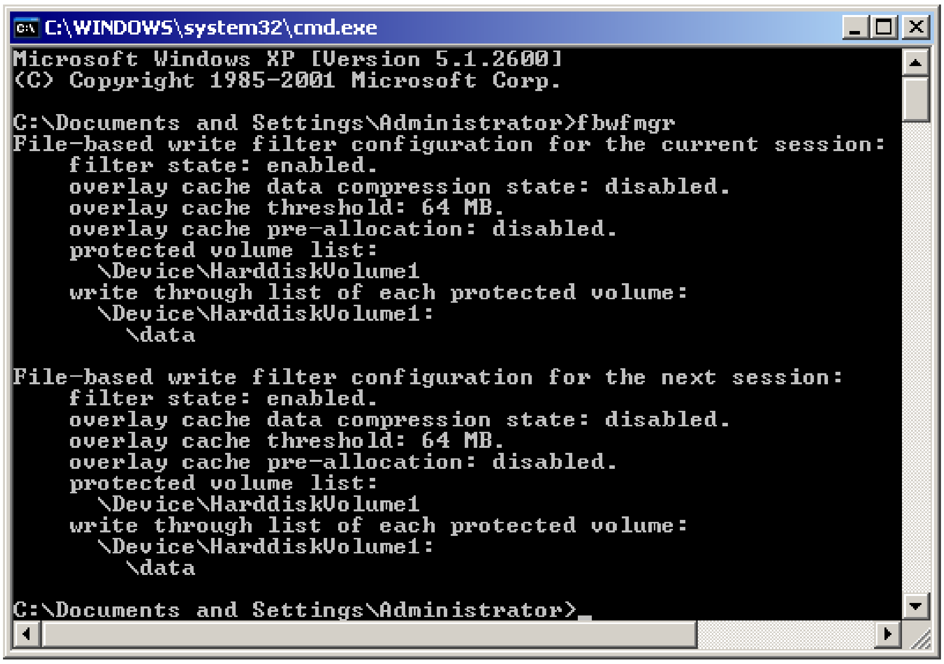
When using Compact flash with Windows XPe, the FBWF must be enabled.

Of course, this setup results in a stateless system. After a reboot, all data previously written will be lost. If FBWF is enabled, the Windows XPe OS is also secured against accidental system file damages (i.e. if an application deleted some important system files.

If you want to store application data permanently, you can use the unprotected folder C:\data or you can disable the FBWF.

5.2. Configuring FBWF – FBWF Manager Tool

XPe includes the "FBWF Manager Console Application", FBWFMGR.EXE, which controls the FBWF configuration. One can query the state of the FBWF feature by simply executing the command *fbwfmgr* in a Windows shell.



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>fbwfmgr
File-based write filter configuration for the current session:
    filter state: enabled.
    overlay cache data compression state: disabled.
    overlay cache threshold: 64 MB.
    overlay cache pre-allocation: disabled.
    protected volume list:
      \Device\HarddiskVolume1
    write through list of each protected volume:
      \Device\HarddiskVolume1:
        \data

File-based write filter configuration for the next session:
    filter state: enabled.
    overlay cache data compression state: disabled.
    overlay cache threshold: 64 MB.
    overlay cache pre-allocation: disabled.
    protected volume list:
      \Device\HarddiskVolume1
    write through list of each protected volume:
      \Device\HarddiskVolume1:
        \data

C:\Documents and Settings\Administrator>_
```

figure 2 – fbwfmgr.exe utility

To enable or disable the FBWF RAM overlay of drive C:, one must open a shell console in Windows XP Embedded and use the following command:

```
C:\> fbwfmgr /enable
C:\> fbwfmgr /disable
```

This will install or remove the RAM overlay of partition drive C:. These operations require a complete system reboot to effect the changes. Don't expect the FBWF being enabled/disabled just by executing the appropriate command. A power cycle is mandatory.

Important:

Each configuration change of the FBWF needs a complete system reboot until the change is effective.

5.3. Committing files and adding exclusions

It is possible to commit a file from the protected region to the Compact flash card. As an example, we will create a file on the protected C:\ root directory called myfile.txt and will commit it to the compact flash card.

```
C:\> echo hello > c:\myfile.txt
C:\> fbwfmgr /commit c: \myfile.txt          (note the space between c: and \myfile.txt)
```

After the next reboot, myfile.txt will still exist.

If you like to add some folders that are permanently writable, you could make use of the “addexclusion” argument of FBWF. The following example will change the directory test state from protected to unprotected.

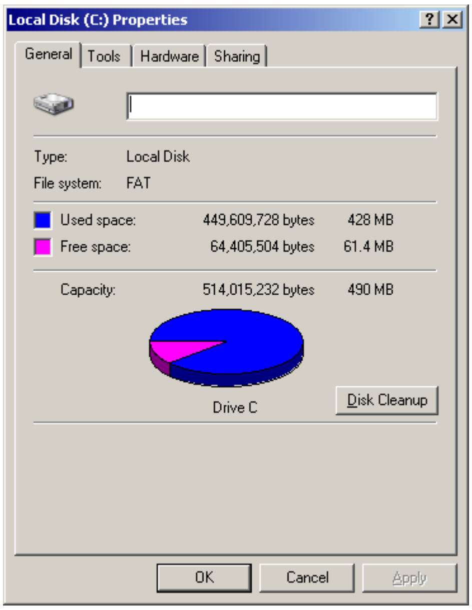
```
C:\> mkdir c:\test
C:\> fbwfmgr /addexclusion c: \test          (note the space between c: and \myfile.txt)
```

Again, reboot the system and note that the directory test still exists.

For further usage examples of the FBWF manager tool, please consult the document *FBWF.pdf* from the DVD-ROM.

5.4. Compact flash drive space

When opening the drive C: properties page with FBWF enabled, one thing could lead to confusion: Although you have a 1GByte Compact flash card inserted, the properties page displays a disk capacity of ~490MBytes with only ~60MBytes free space left.



Local Disk (C:) Properties

Type: Local Disk
File system: FAT

🟦 Used space: 449,609,728 bytes 428 MB
🟪 Free space: 64,405,504 bytes 61.4 MB

Capacity: 514,015,232 bytes 490 MB

figure 3 – disk C: properties

This might be confusing but it is not a system fault. With FBWF enabled, the write cycles to drive C: are filtered into RAM layer which is limited to 64MBytes. So if you download a file of size 10MBytes from the Internet to the Desktop, the free space would decrease 10MBytes. If you delete that file or if you reboot, the 10MBytes would be free again.

With FBWF disabled, you would see the "real" free space on the Compact flash card which would be around 510 MBytes.

Compact Flash Cards

5.5. Wear Leveling Algorithms explained

Let's assume a Compact flash card that has 6 blocks, where 3 blocks are filled with system data (i.e. operating system files on drive C:) that were written once. This might be a typical situation where the FBWF filter is enabled on the first 3 blocks (see section 5). The first 3 system file blocks are protected by the FBWF, where the remaining 3 blocks are unprotected on the Compact flash card. We further assume that a user application is writing a single block of data to the same logical block address of the Compact flash card for 99 times.

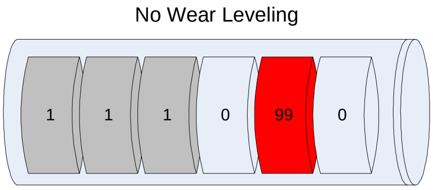


figure 4 – no wear leveling

If the Compact flash card does not implement wear-leveling, then all 99 writes happen on the same block and the flash is destroyed soon (see figure 2 – the numbers represent the write access count of each block).

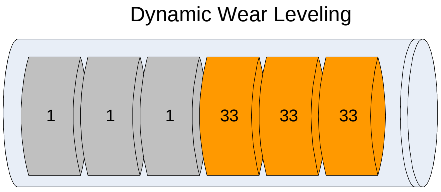


figure 5 – dynamic wear leveling

If the Compact flash does implement a dynamic wear leveling algorithm, then the write accesses are evenly distributed across the free blocks, if the application always writes to the same file. This leads to much better endurance of the flash (see figure 3).

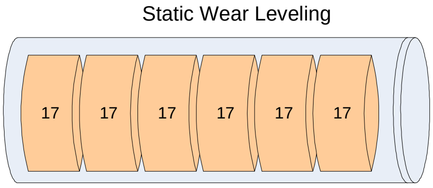


figure 6 – static wear leveling

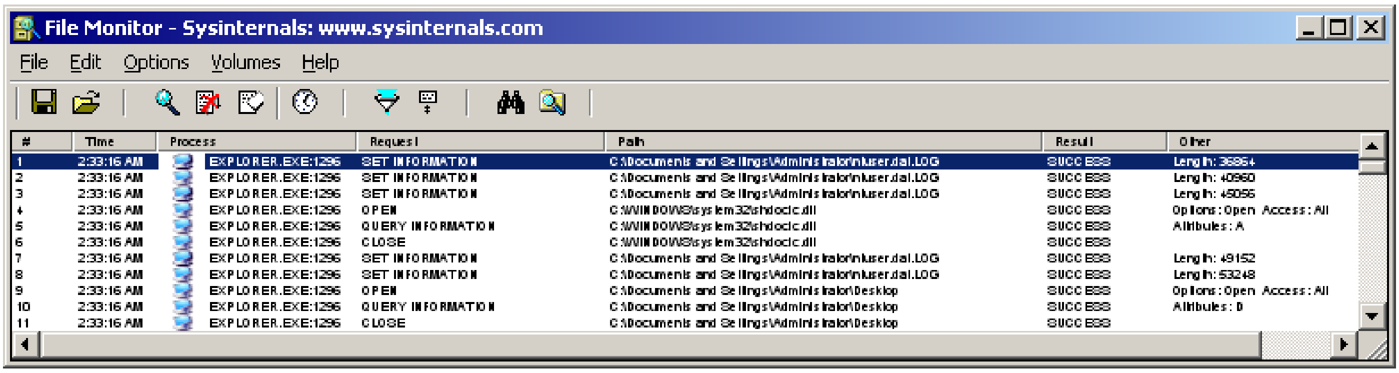
Maximum endurance is achieved with a Compact flash card that implements Static Wear-Leveling. This algorithm spreads the write cycles over the entire flash capacity, including the blocks that are protected by FBWF.

You might wonder where the FBWF protection has gone with static wear-leveling. On the first glance, the FBWF features seem useless with a Compact flash card that implements static wear-leveling.

Let me explain this by introducing two different FBWF protection goals:

- flash protection against user program write accesses
- flash protection against XPe system activity write accesses

If you use a file monitor tool (e.g. FILEMON from www.sysinternals.com), you will notice a lot of write activity caused by the XPe system. E.g. if you open the internet explorer application on the desktop, the window properties will be stored in the file user.dat as shown in the following figure.



#	Time	Process	Request	Path	Result	Other
1	2:33:16 AM	EXPLORER.EXE:1296	SET INFORMATION	C:\Documents and Settings\Administrator\ntuser.dat.LOG	SUCCESS	Length: 36864
2	2:33:16 AM	EXPLORER.EXE:1296	SET INFORMATION	C:\Documents and Settings\Administrator\ntuser.dat.LOG	SUCCESS	Length: 40960
3	2:33:16 AM	EXPLORER.EXE:1296	SET INFORMATION	C:\Documents and Settings\Administrator\ntuser.dat.LOG	SUCCESS	Length: 45056
4	2:33:16 AM	EXPLORER.EXE:1296	OPEN	C:\WINDOWS\system32\shdoclc.dll	SUCCESS	Options: Open Access: All
5	2:33:16 AM	EXPLORER.EXE:1296	QUERY INFORMATION	C:\WINDOWS\system32\shdoclc.dll	SUCCESS	Attributes: A
6	2:33:16 AM	EXPLORER.EXE:1296	CLOSE	C:\WINDOWS\system32\shdoclc.dll	SUCCESS	
7	2:33:16 AM	EXPLORER.EXE:1296	SET INFORMATION	C:\Documents and Settings\Administrator\ntuser.dat.LOG	SUCCESS	Length: 49152
8	2:33:16 AM	EXPLORER.EXE:1296	SET INFORMATION	C:\Documents and Settings\Administrator\ntuser.dat.LOG	SUCCESS	Length: 53248
9	2:33:16 AM	EXPLORER.EXE:1296	OPEN	C:\Documents and Settings\Administrator\Desktop	SUCCESS	Options: Open Access: All
10	2:33:16 AM	EXPLORER.EXE:1296	QUERY INFORMATION	C:\Documents and Settings\Administrator\Desktop	SUCCESS	Attributes: D
11	2:33:16 AM	EXPLORER.EXE:1296	CLOSE	C:\Documents and Settings\Administrator\Desktop	SUCCESS	

figure 7 – filemon output

The main reason for FBWF is to protect the Compact flash card from suffering damage by these XPe internal system activities. As the programmer of the embedded system often has total control over his application and the amount of write accesses to an unprotected file, the flash protection against user program write accesses is not the main concern/problem.

In fact, Compact flash cards implementing static wear leveling have so much better endurance concerning write accesses, that the advantage might overwhelm the fact that the system files on a Compact flash card implementing dynamic wear-leveling are undestroyable.

Big Picture:

dynamic wear-leveling and FBWF:

- XPe system blocks can never be damaged
- application can not write much user data to unprotected files

static wear-leveling and FBWF:

- XPe system file blocks can suffer damage when the flash wears out
- application can write much more data to unprotected files before damage can occur.

Syslogic recommends the use of Compact Flash cards that implement Static Wear-Leveling like the SiliconDrive™ Compact flash card from Silicon Systems.

5.6. Endurance Calculation

To get an idea of how long a Compact flash card with a Static Wear-Leveling algorithm can be used, the following formula could be helpful. This formula assumes a Compact Flash with a flash block endurance of 100'000 write cycles.

$$Years = \frac{CardSize}{WriteRate \cdot 310}$$

Data size unit is ***MBytes*** and ***WriteRate*** is in ***MBytes per Second***. For an application with a Compact flash capacity of 1024 MBytes and a data file of 0.1MBytes that is written 10 times per second, this formula results in a card endurance of 3.3 years.

With SiliconSystems Compact flash cards, the max. write transfer rate is 6MBytes/s, so the minimal endurance for the 1024MByte card is 0.5 years, assuming full write workload.

Please note that the formula above is only an approximation and only correct for Compact flash drives that implement Static Wear-Leveling. Drives that use other algorithms like Dynamic Wear-Leveling or no Wear-Leveling have much less endurance.

Note: The Compact Flash block endurance is assumed as 100k write cycles in the formula above. For other block endurance specifications, the formula can be simply scaled linear with the block endurance enhancement factor.

6 Power Failure

6.1. Overview

Writing a file (consisting of multiple sectors/blocks) to a Compact Flash card is not an atomic operation, so there could be a possible file system corruption upon a power loss. FAT32 file systems typically used under Windows XP can be corrupted. NTFS file systems are less sensitive to corruption, as they implement a journal that keeps track of the directory structure and recovers the inconsistencies while booting. But it is also possible to corrupt an NTFS file system.

6.2. Power Failure - FBWF enabled

When enabling the Enhanced Write Filter according to section 5.2, there is no danger in terms of system file corruption because the system drive partition is write protected. So the system will always boot when the power is switched on again.

Good to know:

FBWF also protects the system files from corruption in case of a power failure.

6.3. Power Failure – FBWF disabled

If you intend not to use FBWF, then an NTFS formatted system drive would be the first choice, although such a setup is not 100% tolerant to a power failure. The chance of getting a corrupted file system is very rare, but exists.

Good to know:

For a system with FBWF disabled, use a NTFS formatted file system.

7 System Errors - Blue Screens

7.1. Overview

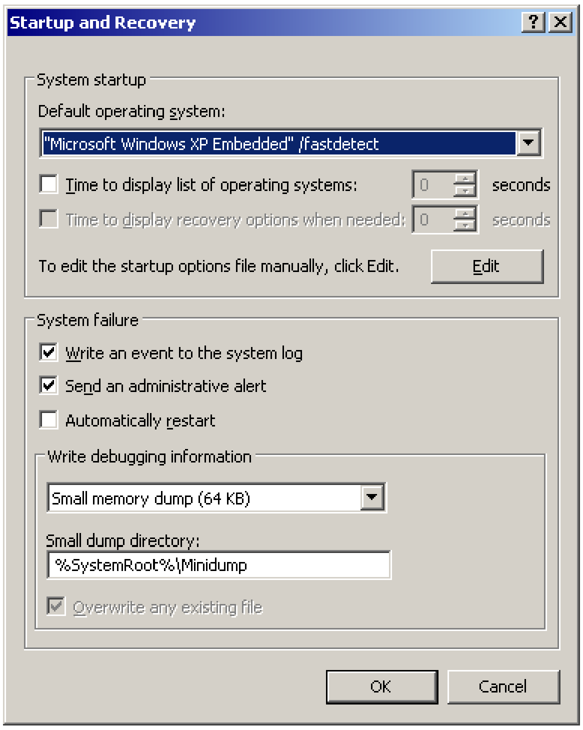
If the Windows XP kernel detects an unrecoverable error (e.g. an NMI interrupt is pending), then the kernel issues a "blue screen" to inform the user of the faulty hardware or malfunctioning software. In most cases there are badly written drivers involved, because drivers can run in "kernel mode" and thus have privileges and rights to access and damage critical kernel sections.

The "blue screen" not only informs the user of a system crash, but also it has the responsibility to halt the processor. Like this, the whole system is physically stopped and there could be no more damage than already has been done.

7.2. Enable Auto Reboot

An embedded system might call for a different behavior than just halting the processor. That's where the AutoReboot feature comes into play. With AutoReboot enabled, the system issues a reboot every time a "blue screen" is happening. This results in a system that might recover from a "blue screen" without user interaction.

To enable AutoReboot, press *WINKEY+Break*, change to *Advanced* Tab and click the "*Startup and Recovery*" button. In the dialog, check "*Automatically restart*". After the next reboot, the "AutoReboot" feature is enabled.



System startup

Default operating system:

"Microsoft Windows XP Embedded" /fastdetect

□ Time to display list of operating systems: 0 seconds

□ Time to display recovery options when needed: 0 seconds

To edit the startup options file manually, click Edit.

System failure

☑ Write an event to the system log

☑ Send an administrative alert

□ Automatically restart

Write debugging information

Small memory dump (64 KB)

Small dump directory:

%SystemRoot%\Minidump

☑ Overwrite any existing file

figure 8 - Auto Reboot feature

8 Remote Debugging

Application debugging can be done from the Visual Studio .NET 200x integrated development environment running on a distant host. The debug session runs over an Ethernet connection. The application and its source files are built, debugged and executed on the target.

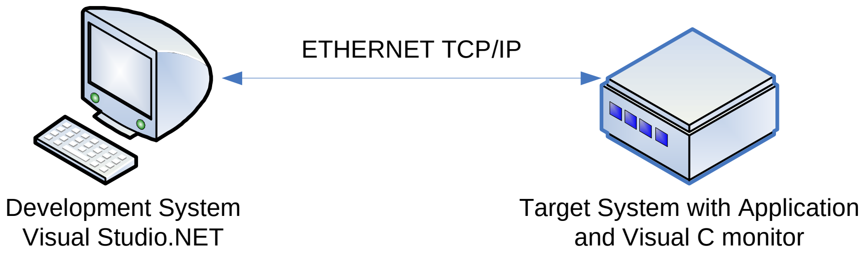


figure 9 - Remote Application Debugging

A monitor application also runs on the target and communicates symbol information back to the Visual Studio debugger. The XP Embedded target must share the application folder with the Development system (map network drive).

8.1. Connect and Debug

The following instructions assume that you know the Embedded XP network identifier (get it from *Control Panel/System/Computer Name*, e.g. XPE) of your embedded system and you have a working Ethernet connection between the Embedded target and the development computer. On the development PC, there's a Visual Studio installed.

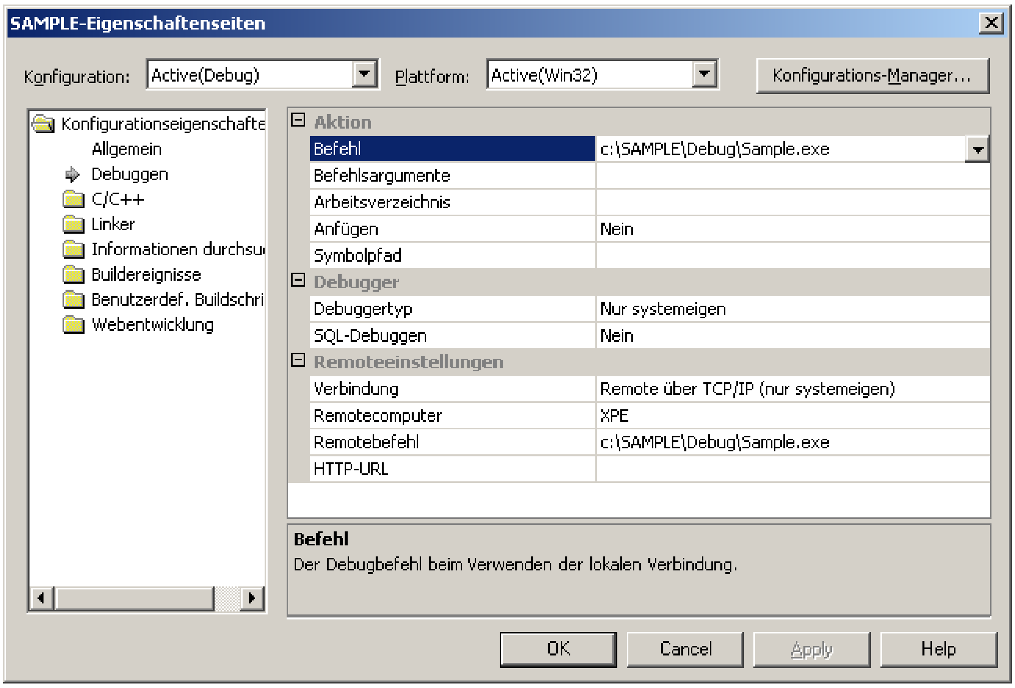
Boot the target and try to ping it from a development system's command window:

```
C:\>ping XPE
Pinging XPE [192.168.1.41] with 32 bytes of data:
Reply from 192.168.1.41: bytes=32 time<10ms TTL=128
```

In the same command window, use NET.EXE to map the remote XPe drive C: as a local drive W: into the development machine.

```
C:\>net use w: \\XPE\c$
The password or user name is invalid for \\XPE\c$.
Type the password for \\XPE\c$: netipc
The command completed successfully.
```

Create your embedded application "Sample" with Visual Studio on the shared network drive W:. In the project settings under "Debugging", make the following configurations:



Konfiguration: Active(Debug)

Plattform: Active(Win32)

Aktion	
Befehl	c:\SAMPLE\Debug\Sample.exe
Befehlsargumente	
Arbeitsverzeichnis	
Anfügen	Nein
Symbolpfad	
Debugger	
Debuggertyp	Nur systemeigen
SQL-Debuggen	Nein
Remoteeinstellungen	
Verbindung	Remote über TCP/IP (nur systemeigen)
Remotecomputer	XPE
Remotebefehl	c:\SAMPLE\Debug\Sample.exe
HTTP-URL	

Befehl
Der Debugbefehl beim Verwenden der lokalen Verbindung.

figure 10 – Remote Debugging Setup

On the embedded XPe device, start the debug monitor from within a command shell.

```
C:>cd MSVCMON
C:\MSVCMON>msvcmon –tcpip -anyuser
```

Now you can compile the Project and start the debug session within Visual Studio on your Development system (menu Debug/Start). Enjoy remote application debugging!

9 Installing the XPe Image

9.1. Overview

If you have lost the Compact flash contents by accident, you must install the XPe image again. Please execute the following instructions to re-install XPe on the Compact flash card.

9.2. Partition the Compact Flash with EDOS-ROM

Power on the IPC/NETIPC with the Compact flash card inserted, press and hold the "*Delete*" key on the keyboard to enter the system BIOS screen. Select "*Basic CMOS Configuration*", where you have to enter the following settings:

- Boot 1st: Drive A:
- Ide 0: 3 = AUTOCONFIG, LBA

Hit "ESC" to leave the configuration screen and select "*Write to CMOS and Exit*" (confirm with Y or Z key). Now, the system should boot into Embedded DOS-ROM drive A:. Execute the "**fdisk**" utility from the command shell to partition the Compact Flash card. The following menu will be displayed on the screen.

```
General Software FDISK Utility (Hard Disk Partitioning System)
Copyright © 1990-1997 General Software. All rights reserved.
…
FDISK Options (Press one digit to select, or ESC to exit):
        1.  Create a new partition (any type).
        2.  Change the active partition.
        3.  Delete a partition (any type).
        4.  Display the fixed disk partitions.
        5.  Select another fixed disk drive to FDISK.
        6.  Display current disk's geometry information.
        ESC. Return to DOS.
```

Eventually you will need to delete all partitions to set up a partition table from a fresh, clean starting point. Select (3.) and choose the partitions to delete.

Basically we must set up one partitions for Windows XPe, system drive C. We will not give a step-by-step guide for handling the fdisk utility here but rather show the final partition table for XPe. You can list this partition table by pressing the key (4.).

```
General Software FDISK Utility (Hard Disk Partitioning System)
Copyright © 1990-1997 General Software. All rights reserved.

Partition Display Menu

Current fixed disk drive is [0-0]: 0

        Partition Status  Type              Start   End    Size
        1         Active  DOS Huge (>32MB)  0       1013   1014   999 MB
        2                 Unused
        3                 Unused
        4                 Unused
Total disk space is 1015 cylinders.
…
```

A new partition can be created with menu entry (1.). Please set up your partition table as shown in the table above. After setting everything up, hit ESC to save the new partition table on the Compact flash card (You are prompted to confirm by pressing Alt-Ctrl-Delete). The partitions need to be formatted next. Restart the IPC and execute the following commands in the shell:

```
[NETIPC] A:\> format c: /s
```

You are now ready to transfer the XPe files onto the Compact flash card. Power off the IPC/NETIPC and remove the Compact flash card.

9.3. Copy the XPe files to the Compact Flash

Now that the Compact Flash is partitioned, you need to transfer the XPe system files. You need a Compact Flash card reader to copy the files from the DVD Drive to the card.

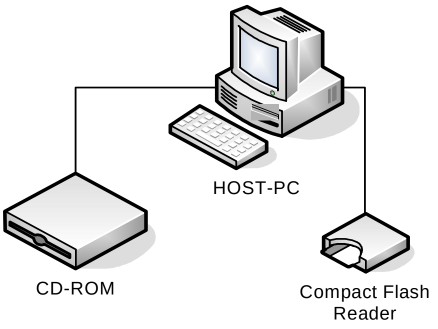


figure 11 – XPe filecopy setup

Copy all the files from the DVD directory **/COMPACTFLASH/FILES_PREFBA/** in Windows Explorer to the Compact Flash card after inserting it into the card reader. Please make sure you copy the files to the first partition of the Compact flash card, not to the second one. The copy process may take a while.

9.4. Make the Compact flash bootable

Next, remove the Compact flash from the card reader and put it back into the IPC/NETIPC card slot. Power on the IPC/NETIPC. There's still Embedded DOS-ROM configured to boot from drive A:

When Embedded DOS-ROM has booted, execute the following commands on the shell:

```
Starting Embedded DOS-ROM…
…
[NETIPC] A:\> C:
[NETIPC] C:\> bootprep /dC
BOOTPREP, Built 807,
Microsoft ® Windows XP Embedded,
…
Are you sure you want to continue (y/N)?
```

Hit y or z here to confirm. Power-cycle the IPC/NETIPC board, press and hold the "*Delete*" key on the keyboard again to enter the system BIOS screen. Select "*Basic CMOS Configuration*", where you have to enter the following settings:

- Boot 1st: Drive C:

Hit "ESC" to leave the configuration screen and select "*Write to CMOS and Exit*" (confirm with Y or Z key). Now, the system should boot into Windows XP Embedded from drive C:.

> **Note:**
> When first booting XPe after installing it, there's a setup program called FBA (First Boot Agent) running that takes a lot of time to complete (~20 minutes). Do not interrupt the FBA or the system will get into an unstable state. After the FBA has set up everything, Windows XPe will boot normally.

9.5. Post FBA setup

After the FBA has finished, it you have to install some components manually. This is the case for the TSharc Touch screen driver. You must install it with C:\TOUCHSCREEN\Setup.exe after disabling the FBWF partition drive C: protection.

10 WinIo Driver Library

10.1. WinIo Library Overview

WinIo is a driver that grants your application access to Windows XP/2000/NT system resources that are normally not accessible. You can use WinIo to simply write a byte to the x86 IO port registers or access physical memory directly. WinIo is written and maintained by Yariv Kaplan (www.internals.com, see 10.5 for license details).

WinIo has the following function interface defined (listing not complete):

bool _stdcall InitializeWinIo()
This function initializes the WinIo library

void _stdcall ShutdownWinIo()
This function performs cleanup of the WinIo library.

bool _stdcall InstallWinIoDriver(PSTR pszWinIoDriverPath, bool IsDemandLoaded);
This function installs the WinIo driver.

bool _stdcall RemoveWinIoDriver()
This function removes the WinIo driver from the system.

bool _stdcall GetPortVal(WORD wPortAddr, PDWORD pdwPortVal, BYTE bSize);
This function reads a BYTE/WORD/DWORD value from an I/O port.

bool _stdcall SetPortVal(WORD wPortAddr, DWORD dwPortVal, BYTE bSize);
This function writes a BYTE/WORD/DWORD value to an I/O port.
<…>

10.2. NETIPC extensions to WinIO

Syslogic has extended the original WinIo Library with some hardware-related functions. These extensions can be accessed like any other WinIo function after properly initializing the WinIo driver:

bool _stdcall GetTemp(float& value)
This function returns the temperature from the on-board LM75 temperature sensor.

bool _stdcall WatchdogGetState(bool& Timeout)
This function returns the state of the Watchdog. 1=Timeout occurred, 0=No Timeout

bool _stdcall WatchdogEnable()

This function enables the watchdog. It will reset the board if not triggered by WatchdogTrigger.

bool _stdcall WatchdogTrigger()

This function triggers the Watchdog.

bool _stdcall WatchdogHardReset()

This function resets the watchdog by hard resetting the board.

bool _stdcall GetCpuFID(DWORD& FID)
bool _stdcall GetCpuOID(DWORD& OID)
bool _stdcall GetCpuRID(DWORD& RID)

These functions return the board identifiers.

10.3. Using WinIo in a Visual C++ application

In order to use WinIo in a Visual C++ application, you should take the following steps (please also have a look at the the TestProg Sample Application found in the TestProg directory):

(1) Place winio.dll, winio.vxd and winio.sys in the directory where your application's executable file resides.
(2) Add winio.lib to your project file by right clicking on the project name in the Visual C++ workview pane and selecting "Add Files to Project...".
(3) Add the #include "winio.h" statement to your source file.
(4) Call InitializeWinIo.
(5) Call the library's functions to access I/O ports and physical memory.
(6) Call ShutdownWinIo.

10.4. TestProg Sample Application

The TestProg sample application demonstrates the use of the extended WinIo Library. After reading out the temperature and the board identifiers, it activates the watchdog and triggers it for about 10 seconds before letting the watchdog generate a timeout.

Copy the WINIO/Bin directory from the DVD-ROM to the Windows XPe device and run TestProg.exe (WinIo.dll and WinIo.sys must be in the same directory as TestProg.exe).

10.5. WinIO End User License Agreement

Software License Agreement for WinIo

The following terms apply to all files associated with the software unless explicitly disclaimed in individual files.

IMPORTANT- PLEASE READ CAREFULLY: BY INSTALLING THE SOFTWARE (AS DEFINED BELOW), OR COPYING THE SOFTWARE, YOU (EITHER ON BEHALF OF YOURSELF AS AN INDIVIDUAL OR ON BEHALF OF AN ENTITY AS ITS AUTHORIZED REPRESENTATIVE) AGREE TO ALL OF THE TERMS OF THIS END USER LICENSE AGREEMENT ("AGREEMENT") REGARDING YOUR USE OF THE SOFTWARE. IF YOU DO NOT AGREE WITH ALL OF THE TERMS OF THIS AGREEMENT, DO NOT INSTALL, COPY OR OTHERWISE USE THE SOFTWARE.

1. GRANT OF LICENSE: Subject to the terms below, Yariv Kaplan ("AUTHOR") hereby grants you a non-exclusive, non-transferable, non-assignable license to install and to use the downloadable version of WinIo ("SOFTWARE").

a. Redistributable Code. You may reproduce and distribute the object code form of the SOFTWARE solely in conjunction with, and as part of, your application ("Permitted Application"); provided that you comply with the following:
If you redistribute any portion of the Redistributable Code, you agree that: (i) you will only distribute the Redistributable Code in conjunction with, and as part of, your Permitted Application which adds significant functionality to the Redistributable Code and that distribution of the Permitted Application does not compete with the AUTHOR's distribution of the SOFTWARE; (ii) you will include a valid copyright notice on your Permitted Application; (iii) you will not permit further redistribution of the Redistributable Code; and (iv) you will indemnify, hold harmless, and defend the AUTHOR from and against any claims or lawsuits, including attorneys' fees, that arise or result from the use or distribution of your Permitted Application.

b. License to use Source Code. You may not sell, lease, rent, transfer or sublicense the source code of this SOFTWARE.

2. MODIFICATION: SOFTWARE Source Code may be modified without the prior written permission of the AUTHOR. Any modifications made to the SOFTWARE will continue to be subject to the terms and conditions of this AGREEMENT.

3. COPYRIGHT: All rights, title, and copyrights in and to the SOFTWARE and any copies of the SOFTWARE are owned by the AUTHOR. The SOFTWARE is protected by copyright laws and international treaty provisions. Therefore, you must treat the SOFTWARE like any other copyrighted material.

4. TITLE: You acknowledge that no title to the intellectual property in the SOFTWARE is transferred to you. Title, ownership, rights, and intellectual property rights in and to the SOFTWARE shall remain the exclusive property of the AUTHOR. The SOFTWARE is protected by copyright laws of the United States and international treaties.

5. LIMITATION OF LIABILITY: You must assume the entire risk of using the SOFTWARE.

IN NO EVENT SHALL THE AUTHOR BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THIS SOFTWARE, ITS DOCUMENTATION, OR ANY DERIVATIVES THEREOF, EVEN IF THE AUTHOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE AUTHOR SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. THIS SOFTWARE IS PROVIDED ON AN "AS IS" BASIS, AND THE AUTHOR HAS NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

11 References

- Syslogic firmware documentation
  (Document Order code: DOC/NETIPCFW-E), available at www.syslogic.ch.

- IPC/NETIPC processor board user documentation
  (Document Order code: DOC/NETIPC-E), available at www.syslogic.ch.

- Windows XP Embedded Advanced, by Sean D. Liming
  RTC Books, San Clemente CA, U.S.A.

- Microsoft Windows Sysinternals Tools & Website
  http://www.microsoft.com/technet/sysinternals/default.mspx

12 Contact

Our distributors and system integrators will gladly give you any information about our products and their use. If you want to contact the manufacturer directly, please e-mail a message containing a short description of your application and your request or use one of our request forms on our homepage.

Syslogic Datentechnik AG
Switzerland

E-Mail:	information	info@syslogic.ch
	technical support	support@syslogic.ch
Homepage:		www.syslogic.ch