

IPC/NETIPCFW-1A

Document Ordercode: DOC/NETIPCFW-E

Revision	Date	Author	Modification
1.0	15.06.1999	U. Müller	
2.0a	16.08.1999	U. Müller	new BIOS configuration options
2.1	31.01.2000	U. Müller	new tools, NT Embedded support
2.2	04.08.2000	U. Müller	updated BIOS
2.3	23.02.2001	U. Müller	new Remote Console tools
2.4	25.07.2001	U. Müller	Support for NETIPC-2 added
2.5	28.09.2001	U. Müller	Corrections in BIOS Int14h reference
3.0	01.12.2004	U. Müller	new BIOS release for NETIPC-5A

Important Notes

The products are intended for measurement, control and communications applications in industrial environments. The products must be assembled and installed by specially trained people. The strict observation of the assembly and installation guidelines is mandatory.

The use of the products in systems in which the life or health of persons is directly dependent (e.g. life support systems, patient monitoring systems, etc.) is not allowed.

The use of the products in potentially explosive atmospheres requires additional external protection circuitry which is not provided with the products.

In case of uncertainty or of believed errors in the documentation please immediately contact the manufacturer (address see chapter 13). Do not use or install the products if you are in doubt. In any case of misuse of the products, the user is solely liable for the consequences.

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, PC/AT Trademark of IBM Corporation USA
- FBCube Trademark of Syslogic Datentechnik AG
- MS-DOS, Windows 3.x/95/98/NT/XP/CE
 Trademark of Microsoft Inc. USA
- Embedded-BIOS Trademark of General Software, Inc. USA
- Embedded DOS-ROM Trademark of General Software, Inc. USA

Contents

1 Introduction.....	6
1.1. Content of this Documentation and Additional Literature.....	6
1.2. Items delivered.....	6
1.3. System Requirements.....	6
1.4. Installation.....	6
1.5. Firmware Version/Revision Numbers.....	7
1.6. Safety Recommendations and Warnings.....	7
1.7. License Policy.....	8
1.8. Additional Hard- and Software.....	9
1.9. Additional Literature.....	9
2 Firmware Description.....	10
2.1. General Firmware Structure.....	10
2.2. BIOS.....	10
2.3. Operating Systems.....	11
2.4. Getting Started.....	11
2.4.1. Standard Configuration from Factory.....	11
3 Boot Loader.....	12
3.1. Boot Loader Program.....	12
3.2. Boot Loader Commands.....	12
3.3. Boot Loader Operation.....	13
4 Embedded BIOS.....	15
4.1. General Information.....	15
4.1.1. Introduction.....	15
4.1.2. Console.....	15
4.1.3. BIOS Console Redirection.....	16
4.1.4. DOS Console Redirection.....	16
4.2. POST (Power-On Self-Test).....	16
4.3. Diagnostic Mode.....	17
4.4. ROM Disk.....	17
4.4.1. General Information.....	17
4.4.2. Make a 704kB ROM-Disk Image.....	17
4.4.3. Make a 1.44MB ROM-Disk Image.....	17
4.4.4. Download the ROM-Disk Image.....	19
4.4.5. Enabling ROM-Drive.....	19
4.5. Binary Configuration Program BCP/BCPX.....	20
4.5.1. Modifiable Options.....	20
4.5.2. Changing the BIOS Image.....	23
4.6. NETIPC board configuration differences.....	24
4.7. Embedded BIOS Reference.....	25

5 BIOS Extensions	31
5.1.General Information.....	31
5.2.VGA BIOS Extension.....	31
6 Development Tools	33
6.1.Serial Remote Console/Disk.....	33
6.2.TCP/IP Remote Console/Disk.....	34
6.3.Using Standard PC Language Compilers.....	34
6.4.Remote Source-Level Debugging.....	34
7 MS-DOS Installation	35
8 Windows NT Embedded Installation	37
9 Linux Installation	39
10 Non-DOS Operating System Installation	41
11 BIOS Configuration for IPC-Modules	43
11.1.NETIPC boards.....	43
12 BIOS Configuration for FBCube-Modules	44
12.1.NETIPC boards.....	44
12.2.CUB/SIC4-1A Serial Interface Module.....	44
12.3.CUB/CARFBC-1R.....	44
13 Manufacturer Information	45
13.1.Contact.....	45
13.2.Warranty.....	45

Tables

Tab. 1 BIOS Console Redirection.....	16
Tab. 2 Fatal Embedded BIOS POST error-code.....	27
Tab. 3 Embedded BIOS POST Progress Codes (in chronological order).....	30
Tab. 4 VGA BIOS Extension.....	32

Figures

Fig. 1 Firmware Structure.....	10
Fig. 2 Boot Loader Operating Modes.....	14

1 Introduction

1.1. Content of this Documentation and Additional Literature

This document describes the basic firmware available for the CUB/NETIPC and IPC/NETIPC processor boards. The firmware is partly already loaded in the Flash-Memory of the NETIPC board when delivered and partly contained in the firmware distribution file *NETIPCFW.EXE*. Please note that IPC/NETIPCFW-1A is absolutely required before you can download or modify firmware or applications program (Note: IPC/NETIPCFW-1A is not automatically delivered with each NETIPC board, but must be downloaded from the internet (www.syslogic.ch)).

The content is based on the following program versions:

Firmware package (IPC/NETIPCFW-1A)	Version 3.0
Boot Loader program	Version 2.5x
Embedded BIOS	Version 4.1.4 and 5.3.0
Embedded DOS-ROM	Version 4.042

All the hardware information is contained in separate manuals for each NETIPC board. Programming information and programming examples are contained in IPC/IOCOMSW-1A and described in DOC/IPC_IOCOMSW. For CUB/NETIPC boards see CUB/IOCOMSW-1A and DOC/IOCOMSW.

1.2. Items delivered

The order code IPC/NETIPCFW-1A includes the basic firmware and utilities for the CUB/NETIPC and IPC/NETIPC boards in a Windows installation program (NETIPCFW.EXE).

1.3. System Requirements

The NETIPC firmware requires a PC/AT compatible computer hardware with at least a 80486-DX processor and a Windows 95/98/NT/2000 operating system with the latest service packs installed. In the current release most development tools run on DOS or in a Windows DOS box or have a DOS counterpart, but the installation requires a Windows operating system.

Syslogic recommends Windows NT 4.0 (service pack 6) or Windows 2000 with a FAT file system partitioned drive C: as a stable development platform.

1.4. Installation

The installation program NETIPCFW.EXE contains a number of files, which must be installed on your PC before you can work with the NETIPC board. For the installation please proceed as follows:

1. Start up Windows on your host PC.
2. Deinstall any earlier version of NETIPCFW
3. Start the installation program NETIPCFW.EXE with a double click.
4. Print and read the instructions and comments in the README.TXT file created in the installation directory (e.g. C:\FBCUBE\NETIPCFW).

Note: the installation program generates a sub-directory structure. Do not change or rename elements of this structure, because other programs expect to find them as defined. The default root directory created is named '*FBCUBE*'.

CAUTION: if you already have installed an older version of the NETIPC firmware and you want to install the new version, you must first completely uninstall the old version using the Windows software registry or the uninstall program UNWISE.EXE in the installation directory (e.g. C:\FBCUBE\NETIPCFW).

1.5. Firmware Version/Revision Numbers

In some cases it may be important to read the installed firmware version/revision numbers. These are available as follows:

Embedded BIOS:	In the BIOS startup screen
Embedded DOS-ROM:	Use DOS-Command VER
Boot Loader:	Target-side: shown after establishing connection (INIT-command) Host-PC side: shown during start-up

1.6. Safety Recommendations and Warnings

Note that some of the information in this manual is only applicable to the software releases mentioned in paragraph 1.1 and not to former revisions !

The README.TXT-file on the NETIPC Firmware installation directory contains important information not contained in this manual. **Print and read the README-file before working with the firmware !**

Please strictly observe all the warnings and safety recommendations described in the hardware user manuals, especially in DOC/IPC_NETIPC, DOC/NETIPC and DOC/CUBINST. They will not be repeated here.

The products require set-up procedures before they can be operated. Before being used in an actual installation, they must be correctly configured. If the products are not or wrongly configured, they may malfunction in the intended application.

If you already have installed an older version of the NETIPC firmware and you want to install the new version, you must first completely uninstall the old files (see paragraph 1.3).

In order to use an already installed NETIPC system (also freshly delivered systems) with the new firmware you must reconfigure/regenerate and download the BIOS, VGA-BIOS extension and the ROM-Drive to the target. Otherwise the new firmware may fail due to version conflicts. Keep the old firmware in a separate directory if you must be able to work with older systems without downloading the new firmware.

The binary BIOS image file delivered with the NETIPCFW installation is preconfigured with the factory defaults for the *CUB/NETIPC-1A* board (BIOS V4) and *IPC/NETIPC-5A* (BIOS V5). Some BIOS options must be changed if you want to download the BIOS image to a

different NETIPC board type (see chapter 12 of this manual). Be very careful when changing BIOS parameters. Make sure that you understand each parameter you change and do not change in case of doubt. A wrongly configured BIOS may result in malfunction or inoperability of the NETIPC or an input-/output-module !

1.7. License Policy

The firmware delivered in the memory of the NETIPC board and the firmware/software delivered with the IPC/NETIPCFW-1A installation file are proprietary firmware/software and their rights belong to the companies as follows:

Syslogic Datentechnik AG, Switzerland:

Boot Loader

Various Utilities and Programming Examples

License Agreement:

The purchase of the NETIPC board includes the right to use this firmware/software without further payments exclusively on the respective NETIPC board and/or on PC's connected to NETIPC boards. Firmware/software delivered in source code form by Syslogic may be changed, adapted or modified by the user without notifying Syslogic. The changed, adapted or modified firmware/software may again only be exclusively used on the NETIPC boards and/or on PC's connected to NETIPC boards. Note that Syslogic cannot support any modified firmware or software.

Technosoftware AG, Switzerland:

TELNET/FTP-Server TNFTP

License Agreement:

The purchase of the NETIPC board includes the right to use this software without further payments exclusively on the respective NETIPC board.

General Software, Inc. USA:

Embedded BIOS

Embedded DOS-ROM

Binary Configuration Program BCP for Embedded BIOS

Embedded DOS-ROM Utilities (Remote Console/Disk, Disk Image Program)

License Agreement:

The purchase of the NETIPC board includes the right to use this firmware/software without further payments exclusively on the respective NETIPC board and to use the Configuration Program for Embedded BIOS and the Embedded DOS-ROM Utilities on development PC's connected to NETIPC boards. Any other use is a violation of the copyright law and may be prosecuted by General Software Inc.

1.8. Additional Hard- and Software

The following additional hardware may be useful for working with the NETIPC processor boards:

SDCFBxxx: SanDisk CompactFlash Memory Card

The following additional software is required for working with the NETIPC processor boards:

IPC/IOCOMSW-1A: Sample program code and utilities for IPC modules

CUB/IOCOMSW-1A: Sample program code and utilities for FBCube modules

1.9. Additional Literature

The following additional documents are required for **correct operation and installation** of the products described in this document:

DOC/EBIOS: Embedded BIOS Reference

DOC/EDOS: Embedded DOS-ROM Reference

DOC/IPC_NETIPC: User Documentation for IPC/NETIPC boards

DOC/IPC_IOCOMSW: User Documentation for IPC Programming Examples and Utilities

DOC/NETIPC: User Documentation for CUB/NETIPC boards

DOC/IOCOMSW: User Documentation for FBCube Programming Examples and Utilities

DOC/CUBINST: User-Documentation for FBCube Installation

DOC/CUBINST also contains the necessary information related to the „*ce*“-**certification** of the products.

2 Firmware Description

2.1. General Firmware Structure

The structure of the firmware is shown in the fig. 1.1. As can be seen, some operating system rely on the BIOS (some may, however, access the hardware directly).

The standard Embedded BIOS configuration may require some changes, if additional boards are used.

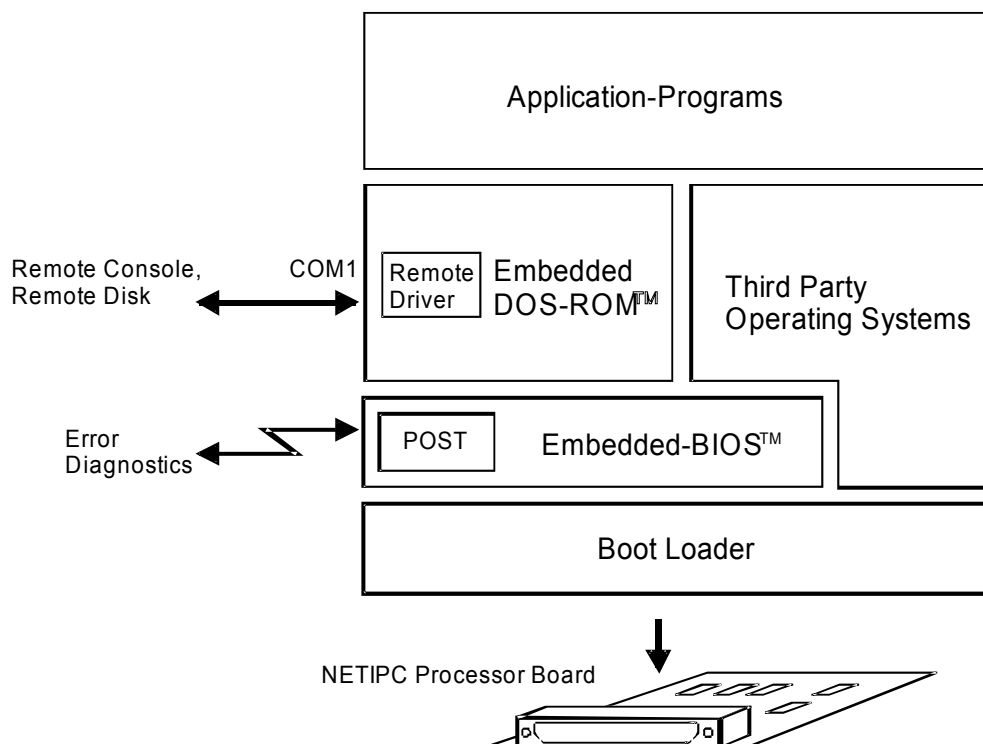


Fig. 1 Firmware Structure

2.2. BIOS

The **BIOS** (= Basic Input/Output System) is the firmware layer, which handles the hardware and offers their functions to the applications software in a standardized mode. On the NETIPC the powerful Embedded BIOS (General Software, USA) is implemented. Embedded BIOS not only offers PC-compatibility, but in addition ROM-drive management as well as WindowsCE direct booting.

2.3. Operating Systems

Operating Systems (especially real-time, multitasking operating systems) are an important part of today's control systems. The NETIPC allows the user to select and load his own operating system (Note: many well-known operating systems - e.g. VxWorks, QNX, Embedded OS-9, Linux, Windows NT/XP Embedded, WindowsCE, RTKernel, RTXDOS-16/32, etc. - are supported by its respective manufacturer or distributor). The NETIPC basic firmware (BIOS, boot loader) are operating-system independent and allow booting of any operating system suited for the NETIPC processor board. Refer to chapter 8 for specific information on downloading non-DOS operating systems to the NETIPC boards.

For less demanding applications the well-known PC operating system **DOS** (Disk Operating System) may be sufficient. One version of DOS - the Embedded DOS-ROM (from General Software, USA) - is included in the delivery and may be used on the NETIPC boards without additional license fees. The user may also load his own copy of MS-DOS (Microsoft). Refer to chapter 7 for specific information on downloading MS-DOS to the NETIPC boards.

The detailed information for the port of a new operating system to the NETIPC is not contained in this manual but can be obtained separately (please ask Syslogic for more information).

2.4. Getting Started

2.4.1. Standard Configuration from Factory

The NETIPC boards delivered from the factory are configured for the **DOS-compatible mode** and are pre-loaded and configured as follows:

- Boot loader target program loaded;
- Embedded BIOS loaded and supporting ROM-drive A: and no console redirection;
Note 1: the console is redirected to 'remote console' in the Embedded DOS-ROM (see chapter 6.1) or can be redirected by BIOS reconfiguration
Note 2: the full BIOS-configuration may be viewed using the 'BIOS Binary Configuration Program' (BCP) with the BIOS image file available in the directory C:\FBCUBE\NETIPCFW\EBIOS (see chapter 4.5 for details)
- VGA BIOS Extension loaded (for onboard VGA controller);
- Embedded DOS-ROM Boot-Drive A: (ROM-Drive) loaded (see chapter 4.4 for details);
- Embedded DOS-ROM remote console/disk driver loaded in CONFIG.SYS (see chapter 6.1 for details).

3 Boot Loader

The boot loader target program is pre-loaded in the NETIPC board and has the following main functions:

- Download of an operating system (first use of the NETIPC);
- Download of a ROM-Drive into the Flash-Memory of NETIPC;
- Upgrade BIOS, BIOS-Extensions (VGA-BIOS) and boot loader firmware in case of new releases.

The NETIPC has three operating modes, which are selected via the configuration of pins on the keyboard connector of the NETIPC board (see the manual DOC/IPC_NETIPC for detailed information). A specially configured dummy keyboard connector may be used for a convenient selection of the operating mode (see NETIPC manual for detailed information). The operating modes of the boot loader are:

RUN Mode:	Starts Up BIOS, operating system and the applications software (see Fig. 1);
Boot-Load Mode:	enters the boot-load dialog with the Boot Loader host-program running on the PC connected to COM1 (RS-232);
Diagnostic Mode:	special factory configuration mode with diagnostic output to a terminal via COM1 (RS-232 at 9'600 bit/s with n/8/1 character format).

3.1. Boot Loader Program

The host-computer part of the Boot Loader program *winld.exe* is installed on the PC (see chapter 1.3) and maintains a dialog with the NETIPC board (i.e. with the target part of the Boot Loader program) via the serial RS-232 Line COM1 of the target. The Boot Loader program can be initialized to use the COM1 or the COM2-port of the PC. The COM-port of the PC will be correctly initialized by executing the INIT-command.

winld.exe is a WIN32 application program running on Windows NT/2000/XP. For host computers running a MSDOS compatible operating system (not a DOS-Box !) the DOS Boot Loader program *dosld.exe* is available in the same directory. The DOS Boot Loader uses the same command syntax and command files as the current Windows version.

3.2. Boot Loader Commands

You can display the available commands of the Boot Loader program by entering „?“ after the prompt: a list of commands is displayed. Additional details (parameters) are displayed in the next window.

The most important commands are:

init	Initializes the host-computer COM-port with the inputted parameters Note: requires e.g. the following parameters: <code>init com1 baud=57600</code>
init1	Initializes the host-computer COM1-port with the following parameters: COM-port 1, baud-rate 57600 bit/s (Note: <code>init1</code> is not a built-in command, but

init2	Initializes the host-computer COM2-port with the following parameters: COM-port 2, baud-rate 57600 bit/s (<u>Note</u> : init2 is not a built-in command, but executes the command file init2.flc)
dir	displays boot block flash drive directory
del	deletes program name from flash-directory
program	path\filename.ext format=bin addr=xxxxxxx: writes the binary file 'filename.ext' to the start address xxxxxxx into the flash-memory (see DOC/IPC_NETIPC for memory layout and address mapping of the NETIPC board)
filename	executes the commands found in the command file 'filename.flc'. Command files are simple ASCII character files containing valid boot loader commands (one per line). Examples of such command files are init1.flc, init2.flc, ebios.flc, stpcvga.flc, edos.flc, edrive_a.flc (all in directory C:\FBCUBE\NETIPCFW\BOOT>)
quit	updates, saves the flash-drive tables and quits the program

WARNING: whenever leaving the Boot Loader program, always use **quit**. Do not switch power off or disconnect the target while the quit-command is in operation. Failing to do so may corrupt the flash-tables and make the FBCube un-bootable !

3.3. Boot Loader Operation

The boot loader for the NETIPC board supports the download of binary and of Intel-HEX file formats into the boot-block flash-memory on the NETIPC. The resident part in the flash-memory has already been downloaded in the factory. After the installation the host PC-part of the boot loader is located in the directory C:\FBCUBE\NETIPCFW\BOOT>. To work with the Boot Loader proceed as follows:

1. Connect the COM1 of the NETIPC to a COM-port of your host-PC using a Null-Modem Cable;
2. Switch the NETIPC to boot load-mode (see DOC/IPC_NETIPC for detailed information);
3. start the Boot Loader host program **winld.exe** with START -> PROGRAMS -> FBCUBE -> NETIPC Boot Loader;
4. Input: **init1** (for COM1 on your PC, init2 for COM2 on your PC);
5. The target now responds with „target system on-line“
6. Input: ?
7. The Boot Loader program displays valid commands and options.

CAUTION: executing commands of the Boot Loader program may overwrite the flash-memory or change options which make operation of the data link impossible. For all the necessary operations with respect to the NETIPC **command-files** (extension *.FLC) are provided. Do only work with these command files.

WARNING: the Boot Loader program must always be correctly terminated by the command „QUIT“. You must wait before switching off or disconnecting until the program has finished updating all its internal tables, otherwise the operation will not be terminated correctly.

The use of the boot loader for different functions (e.g. downloading the operating system, downloading a ROM-drive, etc.) is explained later in this manual.

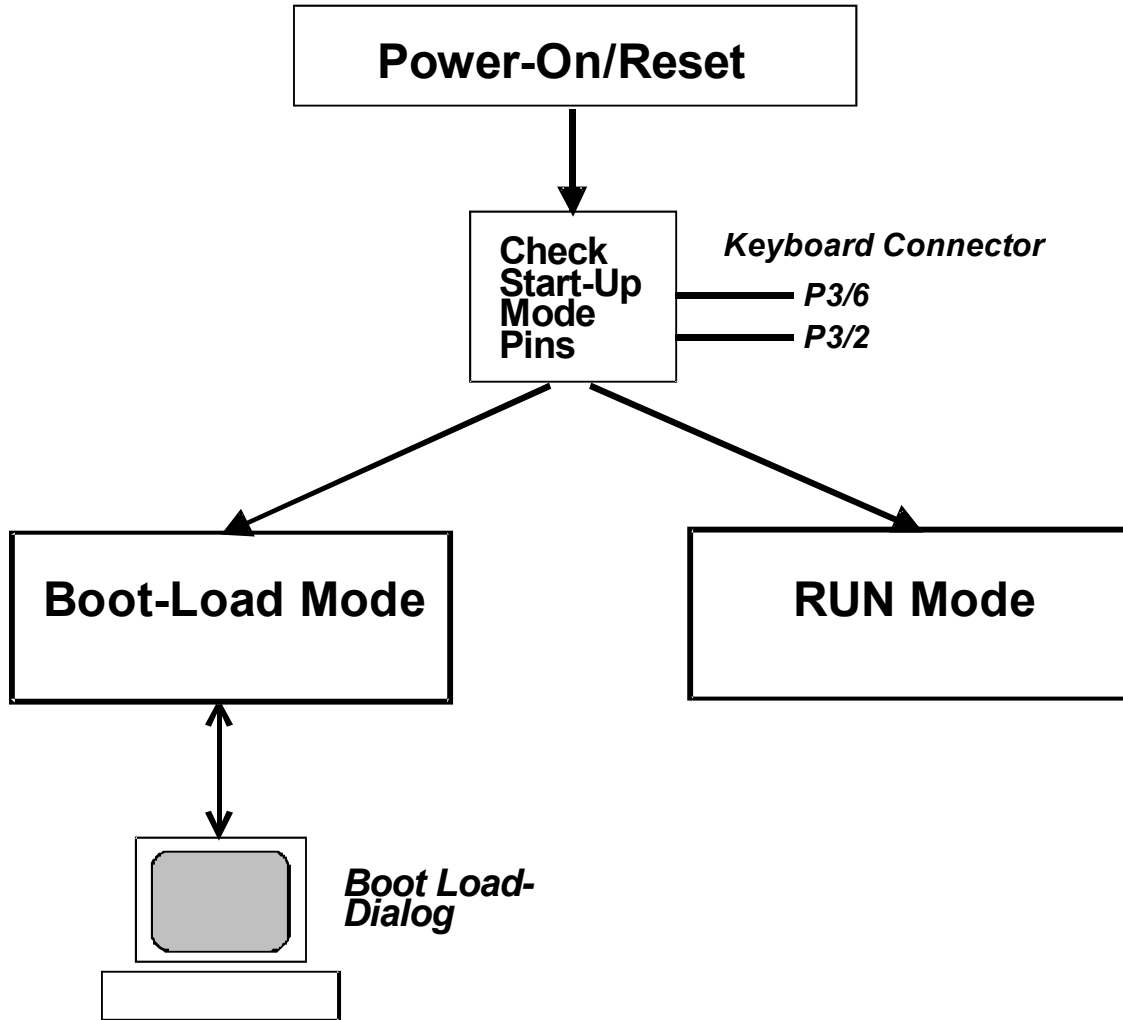


Fig. 2 Boot Loader Operating Modes

4 Embedded BIOS

4.1. General Information

4.1.1. Introduction

The Basic Input/Output System (BIOS) is a standard application programmers interface (API) that hides the details of the hardware on a PC-AT compatible processor system behind a well defined software interface. The NETIPC boards are delivered with Embedded BIOS (General Software, Inc. USA) a BIOS implementation that is specially designed for embedded target platforms, such as the NETIPC. Embedded BIOS comes with an easy to use configuration tool (Binary Configuration Program „BCP“) that allows the system integrator to fine tune the BIOS to his application. In addition to the standard BIOS functions (Power On Self Test, console/keyboard control, COM port control, etc.) it provides support for WindowsCE booting, for a ROM disk (as boot drive A:) accessible from any operating system through the standard BIOS int13 calls, and it allows to redirect the standard console/keyboard output/input over one of the four serial COM ports replacing the expensive video output by a simple serial connection. The full description of the standard BIOS API calls is not repeated here but may be found in standard PC-AT literature.

4.1.2. Console

The „console“ is the system device, through which the operating system and the BIOS (and also the applications programs) handle input/output. The „console device“ is the physical device displaying and accepting information, e.g. the VGA-screen/keyboard or a terminal connected via the serial port.

The console can be assigned to the console device according to the user's need. The linking of the console to the desired console device is called „console redirection“ or „redirection“. The console can be redirected in the BIOS and in the DOS and care must be taken, not to mix or to select contradictory settings.

4.1.3. BIOS Console Redirection

In the Embedded BIOS the console function is separated into a „console I/O“ and a „set up I/O“. The I/O's can be redirected independently and also depend from the start up mode of the NETIPC board. The relationship is as follows:

Start-Up Mode	Console I/O to:	Set-Up I/O to:	Remarks
RUN	0: Video (factory set) 1: COM1 2: COM2 3: COM3 4: COM4	0: Video (factory set) 1: COM1 2: COM2 3: COM3 4: COM4	0 (= Video) requires connection of video/keyboard devices <u>or</u> a redirection in DOS
Boot Loader	-	COM1 (factory set)	cannot be redirected
Boot Loader Diagnostics	-	COM1 (factory set)	cannot be redirected
BIOS Diagnostics	0: Video (factory set) 1: COM1 2: COM2 3: COM3 4: COM4	0: Video (factory set) 1: COM1 2: COM2 3: COM3 4: COM4	0 (= Video) requires connection of video/keyboard devices <u>or</u> a redirection in DOS

Tab. 1 BIOS Console Redirection

4.1.4. DOS Console Redirection

In the operating system (e.g. DOS) the console can additionally be redirected (e.g. to a „remote console“ service via serial link or via Ethernet-TCP/IP). **Warning:** whenever a DOS-console redirection is enabled, the *Embedded BIOS redirection should be 0*.

4.2. POST (Power-On Self-Test)

The POST (= *Power-On Self-Test*) is a resident firmware (part of the BIOS) which is executed each time the NETIPC board is powered on or is RESET. The POST executes a number of hardware tests (memory check, interrupt structure, etc.). If the POST does not find any errors or malfunctions, the operating system and applications programs are started. If, however, the POST finds errors or malfunctions, it communicates via two mechanisms with the user:

Fatal Errors: the RED LED on the front flashes ON-OFF

Notes:

- a) the number of OFF-pulses after the pause indicates the POST error-code (see Tab. 2);
- b) the target will not continue. i.e. it will not boot the BIOS and/or the operating system !

Soft Errors: If a soft error is found, it will be displayed at the end of the POST on the standard system console.

Note: during its execution, the POST writes its progress to the output port 0x80. The sequence of result codes written into this port is listed in the Tab. 3. By using a special hardware listening and logging the arriving progress codes on the port 0x80, the progress can be monitored.

4.3. Diagnostic Mode

The diagnostic mode of the NETIPC can be entered by setting the start-up pins of the keyboard connector to the „diagnostic mode“ (see NETIPC manual) and resetting. In diagnostic mode special code may be executed for factory diagnostics.

4.4. ROM Disk

4.4.1. General Information

The ROM disk is a binary image (sector by sector) of a real floppy disk that is located anywhere in the protected mode address space of the NETIPC processor. If the BIOS is configured to support the ROM disk, it treats it like a floppy drive A:, allowing booting of the operating system out of the ROM disk.

4.4.2. Make a 704kB ROM-Disk Image

On delivery the BIOS is configured to support a ROM disk located at address 3f00000h with a maximum size of 704 kbyte. The binary image is generated using a DOS batch file 'MAKEDRVX.BAT' located in directory C:\FBCUBE\NETIPCFW\EDOSROM (for Embedded DOS-ROM), C:\FBCUBE\NETIPCFW\MSDOS (for MS-DOS) and C:\FBCUBE\NETIPCFW\NONDOSOS (for non-DOS operating systems). These batch files contain the following steps:

1. First an empty diskette image is created with the maximum allowable size (e.g. 704 kbytes).
2. Secondly the system files, utilities and predefined application programs are copied into the disk image using a disk imaging tool located in directory C:\FBCUBE\NETIPCFW\UTIL

To make the ROM-Disk image, please proceed as follows:

1. Edit the command file MAKEDRVX.BAT to include your applications programs.
Note: in the command file delivered, the monitor program CUBMON.EXE is included. CUBMON is part of the CUB/IOCOMSW-1A software package which must be installed before using MAKEDRVX.BAT in order to use the program. CUBMON.EXE may be deleted from the MAKEDRVX.BAT if not required.
WARNING: do not delete the DOS.SYS, IO.SYS, MSDOS.SYS, COMMAND.COM, CONFIG.SYS and AUTOEXEC.BAT entries from MAKEDRVX.BAT!
2. Save MAKEDRVX.BAT
3. Start MAKEDRVX.BAT (Enter
C:\FBCUBE\NETIPCFW\EDOSROM>MAKEDRVX
4. Note that no error messages are allowed during the process of ROM-drive image generation (please observe the running output on your screen carefully).

4.4.3. Make a 1.44MB ROM-Disk Image

On delivery the BIOS is configured to support a ROM disk located at address 3f00000h with a maximum size of 704 kbyte. Some NETIPC boards (e.g. IPC/NETIPC-2A,-3A,-5A) support a full 1.44 MByte ROM-Disk Image.

Note: this image must be located at address 3e00000h in the target and the BIOS must be configured for a 1.44 MByte ROM-Drive.

The binary image is generated using a DOS batch file 'MAKEDRV2.BAT' located in directory C:\FBCUBE\NETIPCFW\EDOSROM (for Embedded DOS-ROM), C:\FBCUBE\NETIPCFW\MSDOS (for MS-DOS) and C:\FBCUBE\NETIPCFW\NONDOSOS (for non-DOS operating systems).

These batch files contain the following steps:

3. First an empty diskette image is created with the maximum allowable size (e.g. 1.44 MBytes).
4. Secondly the system files, utilities and predefined application programs are copied into the disk image using a disk imaging tool located in directory C:\FBCUBE\NETIPCFW\UTIL

To make the ROM-Disk image, please proceed as follows:

5. Edit the command file MAKEDRV2.BAT to include your applications programs.
Note: in the command file delivered, the monitor program CUBMON.EXE is included. CUBMON is part of the CUB/IOCOMSW-1A software package which must be installed before using MAKEDRV2.BAT in order to use the program. CUBMON.EXE may be deleted from the MAKEDRV2.BAT if not required.
WARNING: do not delete the DOS.SYS, IO.SYS, MSDOS.SYS, COMMAND.COM, CONFIG.SYS and AUTOEXEC.BAT entries from MAKEDRV2.BAT!
6. Save MAKEDRV2.BAT
7. Start MAKEDRV2.BAT (Enter
C:\FBCUBE\NETIPCFW\EDOSROM>MAKEDRV2
8. Note that no error messages are allowed during the process of ROM-drive image generation (please observe the running output on your screen carefully).

4.4.4. Download the ROM-Disk Image

The binary image generated by MAKEDRVX.BAT or MAKEDRV2.BAT must be downloaded to the boot block flash memory using the Boot Loader program (see chapter 3 of this manual).

A command file is available, please proceed as follows:

1. Start boot loader program
2. Enter: EDRIVE_X or EDRIVE_2 (command file for download of Embedded DOS-ROM), MDRIVE_X or MDRIVE_2 (command file for download of MS-DOS), NDRIVE_X or NDRIVE_2 (command for download of non-DOS operating system)
3. Quit boot loader program.

Note: xDRIVE_X downloads the ROM-Disk image to address 3f00000h, xDRIVE_2 to address 3e00000h in the target. With xDRIVE_X the maximum allowable image size is 704 kBytes, with xDRIVE_2 it is 1.44 MBytes. Downloading an image with a bigger size may overwrite the BIOS and BIOS extensions and result in unpredictable behaviour of the target system. Check the NETIPC hardware documentation to verify if your target supports only 704 kByte or also 1.44 MByte ROM-Disk images. Both MAKEDRVX.BAT and MAKEDRV2.BAT generate the same output file DRIVE_X.BIN, so be shure to regenerate it when changing from one size to the other before downloading the image.

4.4.5. Enabling ROM-Drive

Enabling the support of the ROM disk is done using the Binary Configuration Program. The ROM-disk is already correctly enabled and defined for a 704 kByte ROM-Drive when the NETIPC is delivered !

When changing to another image size, the BIOS must be adjusted in the 'Drive Assignment' menu of the Binary Configuration Program and downloaded to the target. 'ROM Drive 704 k' selects image start address at 3f00000h (max. size 704 kbytes), 'ROM Drive 1.44 M' selects image start address at 3e00000h (max. size 1.44 MBytes).

4.5. Binary Configuration Program BCP/BCPX

The Binary Configuration Program „*BCP*“ is an interactive DOS program that allows to read, modify and write back the binary BIOS image file on the host PC. The modified BIOS image must be downloaded to the NETIPC board using the Boot Loader program. Currently two versions of this program are available:

- BCP.EXE: binary configuration program for Embedded BIOS 4.x.
- BCPX.EXE: binary configuration program for Embedded BIOS 5.x.

BCP patches the binary BIOS file BIOS.BIN while BCPX patches the binary BIOS file BIOSF.BIN.

WARNING: be very careful when changing BIOS parameters. Make sure that you understand each parameter you change and do not change in case of doubt. A wrongly configured BIOS may result in malfunction or inoperability of the NETIPC or an Input-/Output-module !

4.5.1. Modifiable Options

The following options may be modified using BCP(X):

Menu Target Board

Allows setting of the target board type. This has effects on the setting of some board type dependent options, e.g. maximum memory size.

Note: Always set the Target Board before modifying other parameters.

Menu COM ports

For each of the four COM ports the I/O base address, the BIOS int14 initialization byte, automatic RTS assertion and clock scaling in BIOS int14 initialization routine may be defined. The init byte is set to 9600 baud, no parity, 8 data bits, one stop bit by default. Automatic RTS and DTR assertion is enabled by default and may be disabled in case the COM port is used with half duplex RS485 driver circuit where the transmitter enable is controlled by the RTS line. Clock scaling is used to scale the baud rate divisor in case of a non standard clock frequency attached to the COM ports UART device. It is set to 0 by default for the standard 1.8432 MHz clock. Select 1 for 3.6864 MHz and 2 for 7.3728 MHz clocks.

Note: When defining the COM1..4 I/O ports, no gap (e.g. COM3 = I/O port 0) must be configured.

Menu LPT ports

Allows setting of the I/O base address for each of the three LPT ports supported by the BIOS.

Menu Console Redirection

Allows setting of the COM port number to be used for console redirection. No redirection (i.e. console = standard video output) is specified by 0 (default). See also 4.1.3 for more information.

Note: Any VGA-BIOS-Extension must be removed to enable the BIOS console redirection (e.g. using boot loader command 'vgadel').

Menu ROM Scan/ROM BASIC

Allows setting of the segment addresses for the video and Embedded DOS-ROM segments and the scan range and scan interval for standard BIOS extensions. Note that these values are all segment addresses and must be multiplied by 16 to get the absolute address. The scan of the Embedded DOS-ROM segment must be disabled if any other operating system should be loaded.

Note: If the Embedded DOS-ROM BIOS extension (DOS.BIN) was installed before, it must be deleted manually (using boot loader command 'edosdel').

Menu Misc BIOS Controls

Allows setting of the BIOS equipment byte, and memory test base size for POST. The maximum low memory setting specifies the scan size of the main memory in kbyte (default 640).

Additional options allows the protection from manual CMOS changes by forcing the update of the CMOS parameters from the BIOS-ROM and enable/disable the BIOS configuration box display on BIOS startup.

Menu Disk I/O & Booting

Allows setting of the boot drive.

If WindowsCE binary image loading/booting is enabled, the BIOS searches for a file named NK.BIN (must be upper case !) on any of the boot drives and tries to load and boot it according to the WindowsCE binary image definition. The searched drives must have a FAT file system if this option is enabled. Some WindowsCE startup arguments may be configured in the separate WinCE configuration menu.

Menu Video Configuration:

Allows setting of the video segment addresses. Note that these values are all segment addresses and must be multiplied by 16 to get the absolute address.

If no Video output or an add-on graphics card is used, the on-board graphics controller may be disabled here. This also sets the frame buffer size to 0 freeing some memory for the application and saves power.

On some NETIPC boards (e.g. IPC/NETIPC-2A) a TV/Video output may be enabled and configured for special video formats.

Menu Memory Configuration:

Allows setting of the chip select logic on the NETIPC module. The maximum main memory size depends on the NETIPC board type set in the Board Type menu.

For all NETIPC but the NETIPC-5A the graphics frame buffer memory (1...4MByte) is subtracted from the main memory (Unified Memory Architecture).

Menu ROM Shadowing:

Allows shadowing of the onboard FlashROM for faster code execution. Note that enabling Read and Write shadowing also allows modification of the shadowed code!

Menu Drive Assignment:

Allows assignment of the available drives (ROM drive, IDE drives, etc.) to the drive letters A: to F:.

'ROM Drive 704 k' selects image start address at 3f00000h (max. size 704 kbytes), 'ROM Drive 1.44 M' selects image start address at 3e00000h (max. size 1.44 MBytes).

Note: The ROM drive should not be assigned to any drive letter when using Windows 95/98/NT/2000/XP on the NETIPC. Windows does not support the BIOS built in protected mode ROM driver.

Menu IDE Configuration:

Allows setting of IDE drive parameters. The BIOS supports four IDE drives attached to the primary (onboard) and secondary (offboard) IDE controllers. The following options are supported:

- No IDE drive
- Autodetection (physical geometry)
Recommended for drives with less than 512 Mbyte capacity
- Autodetection (with LBA translation)
Recommended for drives with more than 512Mbyte capacity
- Autodetection (with CHS translation)
not recommended

Note: It is very important that always the same autodetection mechanism is used which was used to partition the drive. Otherwise all data on the drive may be corrupted due to incompatibilities between the translation methods.

Some operating systems use LBA translation per default also for drives with less than 512 Mbyte (e.g. Windows NT). So the BIOS must be configured for this mode if the drive was partitioned by such an operating system.

The fastest supported IDE PIO mode for the onboard primary IDE master and slave drive is autodetected.

Menu WinCE Configuration:

Allows setting of some WindowsCE startup arguments. Please consult the WindowsCE ETK documentation for details. WindowsCE loading/booting must be enabled to use these options.

Note: Support of these argument settings is dependent on the actual WindowsCE configuration used to build the binary image NK.BIN.

4.5.2. Changing the BIOS Image

To make changes to the existing BIOS image proceed as follows:

1. Run the Binary Configuration Program with START -> PROGRAMS -> FBCUBE -> NETIPC BIOS Vx Configuration.
2. Read BIOS Image File (BIOS.BIN for BCP, BIOSF.BIN for BCPX). Press <F1> key to load the file.
3. Edit BIOS Image.
4. Make your changes (saving the options using <F1> key).
5. Go back to the Main Options screen using <ESC> key.
6. Write BIOS Image File (confirm with 'Y').
7. Exit.
8. Run the Boot Loader and download the new BIOS image with the EBIOS or EBIOSX command (see chapter 3 for operation of the Boot Loader).
9. Enter CMOS Setup while restarting the NETIPC and select „Reset CMOS to Factory Defaults“
10. Save updated CMOS parameters and exit

The NETIPC has now registered the new BIOS settings and is ready for operation.

4.6. NETIPC board configuration differences

The binary BIOS image file BIOS.BIN delivered on the NETIPCFW is preconfigured with the factory defaults for the *CUB/NETIPC-1A* board. The binary BIOS image file BIOSF.BIN delivered on the NETIPCFW is preconfigured with the factory defaults for the *IPC/NETIPC-5A* board.

IPC/NETIPC-1A,-1B,-2A,-3A and CUB/NETIPC-1A,-1B are delivered with Embedded BIOS V4.x installed. This BIOS uses BCP with file BIOS.BIN for binary configuration and only requires 64kbyte in the target system (address F0000h to FFFFh).

IPC/NETIPC-5A is delivered with Embedded BIOS V5.x installed. This BIOS uses BCPX with file BIOSF.BIN for binary configuration and requires 128kbyte in the target system (address E0000h to FFFFh).

All NETIPC boards may be used with Embedded BIOS V5.x but the IPC/NETIPC-5A is not supported by Embedded BIOS V4.x.

Two of the BIOS options must be changed if you want to download the BIOS image to another NETIPC board type.

1. The Board Type must be adjusted in the menu 'Target Board'. This also adjusts the main memory size to the maximum supported by this Board Type.
2. The graphics frame buffer size eventually must be reduced when changing to a NETIPC-1B due to the smaller main memory available.

WARNING: if this options are not adjusted correctly, the memory test and board operation may fail.

4.7. Embedded BIOS Reference

This chapter defines the application programming interface (API) to the BIOS services supported by the current Syslogic adaption of Embedded BIOS. Only non-standard API functions are listed. The standard functions reference may be requested from Syslogic as a separate document (DOC/EBIOS-E), it is based on the original IBM PC/AT and PS/2 BIOS.

INT 10h, Video BIOS Services

The video BIOS is called through software interrupt 10H. Many services are provided to modify or inspect the contents of the video display.

INT 11h, Equipment List Service

The Equipment Status Interrupt is invoked to return the device flags as defined in the BIOS Data Area's **DevFlags** field.

INT 12h, Low Memory Size Service

The Low Memory Size Interrupt is invoked to return the size of low memory in kilobytes.

INT 13h, Disk Services

The disk BIOS is called through software interrupt 13H. Services are provided to reset the disk system, read the status of the last operation, read diskette sectors, write diskette sectors, verify diskette sectors, format disk tracks, read drive parameters, read drive types, detect media changes, set the media type, and set the media type for formatting.

INT 14h, Serial I/O Services

The serial BIOS is called through software interrupt 14H. Services are provided to initialize the serial ports, send characters, receive characters, and read the serial port status.

INT 15h, General Services

The general services BIOS is called through software interrupt 15H. Services are provided to support multitasking for device waits, protected mode functions, and access to system configuration information.

Additional services are supported to handle CMOS RAM reading and writing, setting the BIOS CurrIo variable to affect console redirection and return the EMBEDDED BIOS version number.

INT 16h, Keyboard Services

The keyboard BIOS is called through software interrupt 16H. Services are provided to read keystrokes from the keyboard typeahead buffer, peek at the next keystroke in the typeahead buffer, and get the status of the shift keys on the keyboard.

Additional services are provided by the INT 16h service module to set the CPU speed and manipulate the system's cache. These services are historically bound to the INT 16h service because they were first managed by the 8042 keyboard controller.

INT 17h, Parallel I/O Services

The parallel BIOS is called through software interrupt 17H. Services are provided to write a character to the parallel port, initialize the printer attached to a parallel port, and read the status of a printer attached to a parallel port.

INT 1ah, Time Services

The date/time BIOS is called through software interrupt 1aH. Services are provided to read the system time counter, write the system time counter, read the real time clock, write the read time clock, read the real time clock date, and write the real time clock date.

Code	BIOS POST Error
1	memory refresh is not working
2	parity error in 1st 64KB of memory
3	memory failure in 1st 64KB of memory
4	timer T1 not operational
5	CPU test failed
6	gate A20 failure
7	DMA page/base registers
8	video error (nonfatal)
9	keyboard failure
10	CMOS shutdown register failed
11	external cache is not working
12	chipset initialization failure
13	exhaustive low memory test
14	exhaustive extended memory test
15	CMOS restart byte can't hold data
16	address line test failed
17	data line test failed
18	interrupt controller failure

Tab. 2 Fatal Embedded BIOS POST error-code

Code	BIOS POST Progress
00H	POST beginning
01H	CPU register test about to start
02H	NMIs are disabled; delay starts
03H	power-on delay finished
04H	kbd BAT done; reading kbd SYS bit
05H	disabling shadowing & cache
06H	calc'ing ROM cksum, wait kbd ctrlr
07H	cksum okay, kbd ctrlr free
08H	verifying BAT cmd to kbd ctrlr
09H	issuing kbd ctrlr cmd byte
0aH	issuing kbd ctrlr data byte
0bH	issuing pin 23,24 blocking & unblocking
0cH	issuing kbd ctrlr NOP cmd next
0dH	testing CMOS RAM shutdown register
0eH	checking CMOS cksum, updating DIAG byte
0fH	initializing CMOS (if req'd every boot)
10H	init CMOS status reg for date/time
11H	disabling DMA, interrupt ctrlrs
12H	disabling Port B, disabling video display
13H	init chipset, start auto-mem detect
14H	starting timer tests

Code	BIOS POST Progress
15H	testing 8254 T2, for spkr, part B
16H	testing 8254 T1, for refresh
17H	testing 8254 T0, for 18.2Hz
18H	starting memory refresh
19H	testing memory refresh
1aH	testing 15usec refresh ON/OFF time
1bH	testing base 64KB memory
1cH	testing data lines
20H	testing address lines
21H	testing parity (togglng)
22H	base 64KB mem read/write test
23H	system init before vector table init
24H	init vector table
25H	reading 8042 for turbo switch setting
26H	initializing turbo data
27H	any init after vector table init is next
28H	setting monochrome mode
29H	setting color mode
2aH	toggle parity before optional video ROM test
2bH	init before video ROM check
2cH	control passed to video ROM
2dH	video ROM returned control
2eH	checking for EGA/VGA adapter found
2fH	no EGA/VGA found, r/w test of video memory
30H	looking for video retrace signal
31H	retrace failed, checking alt. display
32H	alt found, checking video retrace signal
33H	compare switches w/actual adapter type
34H	setting display mode
35H	check ROM BIOS data area at seg 40h
36H	setting cursor for power-on msg
37H	displaying power-on message
38H	save cursor position
39H	display BIOS ident. string
3aH	display "Hit to ..." msg
40H	preparing vm test. vrfy from display memory
41H	preparing descriptor tables
42H	enter virtual mode for memory test
43H	enable ints for diagnostics mode
44H	init data for checking wraparound at 0:0
45H	checking for wrap, find total memory size
46H	write extended memory test patterns
47H	write conventional memory test patterns
48H	finding low memory size from patterns

Code	BIOS POST Progress
49H	finding high memory size from patterns
4aH	check ROM BIOS data area again
4bH	check for , clear low mem for soft reset
4cH	clearing ext mem for soft reset
4dH	saving memory size
4eH	on cold boot, display 1st 64KB memtest
4fH	on cold boot, test all of low memory
50H	adjust memsize for 1K usage
51H	on cold boot, test high memory
52H	prepare for shutdown to real-mode
53H	saved regs & memsize, entering real-mode
54H	shutdown successful, restoring codepath
55H	disabling A20 line
56H	checking ROM BIOS data area again
57H	checking ROM BIOS data area some more
58H	clear the "Hit " message
59H	test DMA page register
60H	verify from display memory (???)
61H	test DMA0 base register
62H	test DMA1 base register
63H	checking ROM BIOS data area again
64H	checking ROM BIOS data area some more
65H	programming DMA ctrllrs 0 & 1
66H	initializing INT ctrllrs 0 & 1
67H	starting keyboard test
80H	issuing reset cmd & clrng output buffer
81H	check for stuck keys & issue test cmd
82H	initializing circular buffer
83H	check for locked keys
84H	check for memsize mismatch (CMOS/BIOSDATA)
85H	check for pswd or bypass setup
86H	pswd checked. do pgming before setup
87H	call the setup module
88H	back from setup, clr screen
89H	display power-on screen message
8aH	display "Wait..." message
8bH	do system & video BIOS shadowing
8cH	load standard setup params into BIOSDATA
8dH	check and initialize mouse
8eH	check floppy disks
8fH	configure floppy drives
90H	check hard disks
91H	configure IDE drives
92H	checking ROM BIOS data area again

Code	BIOS POST Progress
93H	checking ROM BIOS data area some more
94H	setting base & ext mem sizes
95H	memsize adjusted for 1K, verifying disp mem
96H	initialization before calling C800h
97H	call ROM BIOS extension at C800h
98H	processing after extension returns
99H	configuring timer data area, printer base addr
9aH	configuring serial port base addrs
9bH	initialization before coprocessor test
9cH	initializing the coprocessor
9dH	processing after coprocessor initialized
9eH	check ext kbd, kbdID, numlock settings
9fH	issue keyboard ID command next
a0H	kbd ID flag reset
a1H	do cache memory test
a2H	display any soft errors
a3H	set keyboard typematic rate
a4H	program memory wait states
a5H	clear screen
a6H	enable parity and NMIs
a7H	initialization before calling E000h
a8H	call ROM BIOS extension at E000h
a9H	processing after extension returns
b0H	display system config box
b1H	test low memory exhaustively
b2H	test extended memory exhaustively
00H	call INT 19h bootstrap loader

Tab. 3 Embedded BIOS POST Progress Codes (in chronological order)

5 BIOS Extensions

5.1. General Information

The BIOS provides a method for integrating add-on boards with on-board ROM code, called BIOS extensions, into the system. During the POST, interrupt vectors are established for BIOS calls. After the default vectors are in place, a scan for add-on board BIOS extensions occurs. At this point, an add-on board BIOS extension routine can gain control. The routine can establish or intercept interrupt vectors to hook into the system.

Early in the POST the absolute addresses c0000h-c7fff are scanned in 2 kB blocks in search of BIOS extensions that need to be initialized (for example, valid graphics controller VGA BIOS). Later in the POST, the absolute addresses c8000h-effffh are scanned in 2 kB blocks in search of devices with valid BIOS extensions.

Valid BIOS extensions are defined as follows:

Byte 0: 55h

Byte 1: aah

Byte 2: a length indicator representing the number of 512-byte blocks (limit 7fh) in the BIOS extension (length/512). A checksum tests the integrity of the BIOS extension code. Each byte in the defined BIOS extension is summed modulo 100h. This sum must be 0 for the BIOS extension to be valid.

When POST identifies a valid BIOS extension, it executes a far call to byte 3 of the BIOS extension (which should contain executable code). The device can now perform power-on initialization. The BIOS extension should return to the POST by executing a far return.

If the BIOS extension diagnoses a self-test error, the following should be done before returning to POST:

- Set bit 4 of address hex 40:12 (POST status) to 1
- Set the device number for the supported add-on board into register (AH)
- Set a two-digit error code into register (AL)

If no self-test error is found, the BIOS extension should reset bit 4 of address hex 40:12 (POST status) to 0 before returning.

5.2. VGA BIOS Extension

All standard NETIPC boards include an onboard VGA graphics controller.

If enabled in the BIOS, the graphics controller requires a valid VGA BIOS extension to support the basic video graphics functions defined for VGA. The VGA BIOS extension for the NETIPC boards is part of the IPC/NETIPCFW-1A firmware. It is already programmed into the onboard BIOS flash when delivered. If it is overwritten or deleted, it must be downloaded with the Boot Loader Program using a prepared command file.

If the graphics controller is not needed, it may be disabled in the BIOS and the VGA BIOS extension must be deleted in the onboard BIOS flash using the Boot Loader Program with the command VGADDEL. This overwrites the VGA BIOS extension code with all ffh's.

Disabling the graphics controller will save some power and improve overall system performance.

Actually different VGA BIOS extensions exist for the NETIPC products. To enable correct

Board Identification (see product label)	Product Revision	VGA BIOS download command	Remarks
CUB/NETIPC-1B	all	STPCVGA	STPC Client compatible
CUB/NETIPC-1A	all	STPCVGA	STPC Client compatible
IPC/NETIPC-1B #1,2,3	all	STPCVGA	STPC Client compatible
IPC/NETIPC-1B #4+	4+	STPCVGA2	STPC Consumer II compatible
IPC/NETIPC-1A #1,2,3	1-3	STPCVGA	STPC Client compatible
IPC/NETIPC-1A #4+	4+	STPCVGA2	STPC Consumer II compatible
IPC/NETIPC-2A	all	STPCVGA2	STPC Consumer II compatible
IPC/NETIPC-3A	all	STPCVGA2	STPC Consumer II compatible
IPC/NETIPC-5A	all	LynxVGA	LynxEM+ compatible

Tab. 4 VGA BIOS Extension

6 Development Tools

6.1. Serial Remote Console/Disk

The Embedded DOS-ROM is delivered with a useful development tool: the remote console/remote disk program. This program consists of two parts:

-) the **target system part** (which must be loaded into the NETIPC board), named „SERDRIVE.SYS“.

Note that also the CONFIG.SYS file of the target must be modified. The line:

```
DEVICE=SERDRIVE.SYS
```

must be added to the CONFIG.SYS file in the target. This enables the remote disk. The remote disk is automatically assigned to the next free drive letter of the target system (usually C:, but maybe D: or E: if more target disks are defined);

-) the **host-computer part** (which must be available on your host-PC), named „INTERSVR.EXE“.

The program INTERSVR.EXE can be run with different options: the available options can be displayed by starting „INTERSVR /H“ to be found in the directory C:\FBCUBE\NETIPCFW\UTIL. A self-explaining help screen is then displayed.

WARNING: the host-computer program „INTERSVR.EXE“ must be started *before* the target is powered up ! Quitting INTERSVR requires depressing „CTRL“ and „ALT“ at the same time for approx. 3 seconds.

The remote console/remote disk facility allows you to redirect the target console input/output to your host-computer keyboard and screen and to map your host-computer hard disk (or floppy drive) to a drive letter assigned to the target. The console and disk data exchange is then routed through the serial line (COM-ports), i.e. whenever you e.g. type „dir C:“ the directory of your hard disk C: on the host-PC is transferred to the target and returned via the same serial line as console output to your host-PC screen.

NOTE: the host-computer drive linked to the target must be a local drive with a FAT12 or FAT16 file system installed. Otherwise the INTERSVR program will fail.

If you only want to use the remote disk, but not the remote console (i.e. your target console is connected directly to an independent I/O-terminal), you must not write the option ‘/CON’ in ‘DEVICE=SERDRIVE.SYS’ in the target CONFIG.SYS file (see above). Please make sure that the SERDRIVE.SYS file is located on the drive A: of the target, otherwise the command in the CONFIG.SYS file will fail to find and start the remote console/remote disk driver !

When you disable the remote disk or remote console it may be useful to redirect the targets console input/output to any other free COM-port available in the target (e.g. COM2). To do so, you can use the Embedded BIOS console redirection (see chapter 4.1.3 of this manual).

6.2. TCP/IP Remote Console/Disk

Because the above serial remote console/disk tool was specifically developed for DOS-based host PC's, the NETIPC is delivered with another remote console/disk tool useful for all modern host PC operating systems offering TCP/IP support and FTP and TELNET clients.

This tool consists of a FTP and TELNET server (named TNFTP) running on any DOS compatible operating system on the NETIPC target. It may be configured for Ethernet connection (through an external Packet Driver) or for serial connection (through built in SLIP support). A DOS command batch file (RCONSOLE.BAT) helps to keep the startup easy and may be customized (interrupts, packet driver, IP address, COM port, baud rate etc).

Details on host setup and target configuration may be found in the accompanying user documentation (see x:\FBCUBE\NETIPCFW\RCONSOLE\ directory).

Note: The BIOS console redirection over COM port must be disabled in order to use the TCP/IP remote console with SLIP.

6.3. Using Standard PC Language Compilers

Because of the highly PC-AT compatible design of the NETIPC board most standard PC high-level language compilers can be directly used to generate applications programs for the NETIPC. Note that some of the development tools depend on the target operating system. Sample programming examples are distributed in source code form on the diskette IOCOMSW (containing program examples written for the DOS operating system, BORLAND-C++ compiler and BORLAND Turbo-Assembler).

6.4. Remote Source-Level Debugging

Remote Target Source-Level Debugging is a very helpful and time-saving feature for applications software development. Several standard PC language compilers (e.g. BORLAND C++) include such a remote source level debugger which may be operated via a serial link (COM-port) between the target and the host-PC. Some of them even work over a network connection (e.g. Ethernet-TCP/IP). Please consult the documentation accompanying your language compiler for detailed information about features and installation/operating procedures of the remote source level debugging tools.

7 MS-DOS Installation

Specific operating systems (with the exception of the Embedded DOS-ROM) are not supported by Syslogic. Due to its popularity, however, the installation of the standard DOS (usually Microsoft-DOS or an equivalent implementation) shall be explained here. Note that you must make sure to have the legal right to load and run MS-DOS on the target, i.e. to have a correct license. For the installation of MS-DOS please proceed as follows:

1. Prepare the Embedded BIOS for an MS-DOS boot drive:

- Run the NETIPC BIOS Configuration Program on your host-PC
- Read-in the BIOS image file
- Edit the BIOS-image
- Enter the Disk I/O and Booting menu
- Select drive A: as first boot drive
- Save options using <F1> key
- Return to main menu via <ESC> key
- Write BIOS image file
- Exit (Note: you have now generated a new binary BIOS image)

1. Prepare the MS-DOS system files on the host-PC:

- change to directory C:\FBCUBE\NETIPCFW\MSDOS
- Insert system formatted 1.44 MByte 3.5" MS-DOS boot diskette including your required MS-DOS utility programs (e.g. FDISK.EXE, FORMAT.COM, MODE.COM, etc.) into the floppy drive A: of the host-PC
- Run the DOS batch file PREPBOOT.BAT to transfer MS-DOS system files and utility programs to subdirectories SYSTEM and DOS

2. Create a binary image of a MS-DOS boot disk for the ROM-drive A: of the target:

- change to directory C:\FBCUBE\NETIPCFW\MSDOS
- Remove the files which you do not want to have on your ROM-drive A: from the COPYFILES section in MAKEDRVX.BAT
Note: The IO.SYS, MSDOS.SYS and COMMAND.COM files must be the first files copied to the drive image. Do not remove them or change the order!
- Add the files which you want to have additionally on your ROM-drive A: to the COPYFILES section in MAKEDRVX.BAT
- Modify CONFIG.SYS and AUTOEXEC.BAT (fitting your requirements)
- Run the DOS batch file MAKEDRVX.BAT
- Verify, that the files you specified in MAKEDRVX.BAT were copied completely to the drive image (maximum ROM-drive size is 704 kBytes).

The MS-DOS boot ROM disk image is now ready in the file DRIVE_X.BIN

CAUTION: most of the DOS operating systems running on the PC (including MS-DOS, but excluding Embedded DOS-ROM) initialize all COM-ports to 2'400 bit/s when powering up. If you want to use the console redirection feature of the Embedded BIOS to channel all your input/output of your target applications programs to an RS-232 terminal, then you should add the DOS „MODE-command“ to your AUTOEXEC.BAT file in the

„MODE.EXE“ must be installed on the ROM-drive A: of the target. The default MAKEDRVX.BAT and AUTOEXEC.BAT-files delivered with the firmware are prepared for this case.

3. **Download** the new BIOS and bootable ROM-drive into the target:
 - Connect the target COM1 port via a Null-Modem cable to the COMx-port of your host-PC (x = port number)
 - Run the NETIPC boot loader program
 - Power-on the target in the boot-load mode (see DOC/IPC_NETIPC)
 - Enter INITx after the boot loader program prompt (x = host-PC COM port number)
 - Wait for connection to target (Message „Target connected at“)
 - Enter EBIOS (or EBIOSX) to download the new Embedded BIOS
 - Wait for error-free completion (Note: takes several seconds, do never interrupt !)
 - Enter MDRIVE_X to download the bootable MS-DOS ROM-drive A:
 - Wait for error-free completion (Note: takes several minutes, do never interrupt !)
 - Enter QUIT to automatically update the flash-tables and leave the boot loader program
 - Wait for error-free completion (Note: takes several seconds, do never interrupt !)

4. **Start Up the target** with remote console/remote disk connection:
 - Run the program NETIPC Remote Console
 - Reset and Start Up the target in RUN-mode (see DOC/IPC_NETIPC)
 - Wait for connection message from INTERSVR (Note: takes some seconds, do not interrupt !)
 - target DOS-prompt appears: enter DOS-commands, e.g. DIR A: to check correct operation

MS-DOS is now installed on the target. Please remember that you need a valid run-time license for MS-DOS (not available from Syslogic).

Use MAKEDRV2.BAT instead of MAKEDRVX.BAT and MDRIVE_2 instead of MDRIVE_X for a 1.44 MByte ROM-Drive. See also chapter 4.4 for more information about generating a 704 kbyte or 1.44 Mbyte ROM-Disk image.

8 Windows NT Embedded Installation

Note that you must make sure to have the legal right to load and run Microsoft Windows NT Embedded on the target, i.e. to have a correct license. For the installation please proceed as follows:

1. **Prepare the Embedded BIOS** to start the NT Embedded loader:
 - Run the NETIPC BIOS Configuration Program on your host-PC
 - Read-in the BIOS image file
 - Edit the BIOS-image
 - Enter the Disk I/O and Booting menu
 - Select drive A: as first boot drive
 - Save options using <F1> key
 - Return to main menu via <ESC> key
 - Write BIOS image file
 - Exit (Note: you have now generated a new binary BIOS image)

2. **Download** the new BIOS and the NT Embedded Loader into the target:
 - Connect the target COM1 port via a Null-Modem cable to the COMx-port of your host-PC (x = port number)
 - Run the NETIPC boot loader program
 - Power-on the target in the boot-load mode (see DOC/IPC_NETIPC)
 - Enter INITx after the boot loader program prompt (x = host-PC COM port number)
 - Wait for connection to target (Message „Target connected at“)
 - Enter EBIOS (or EBIOSX) to download the new Embedded BIOS
 - Wait for error-free completion (Note: takes several seconds, do never interrupt !)
 - Enter ENTLDR to download the NT Embedded loader.
 - Wait for error-free completion (Note: takes several minutes, do never interrupt !)
 - Enter QUIT to automatically update the flash-tables and leave the boot loader program
 - Wait for error-free completion (Note: takes several seconds, do never interrupt !)

3. **Copy the Windows NT Embedded Files** on a Flash Disk:
 - Build and Install your custom Windows NT Embedded System with the Windows NT Embedded Target Designer
 - Copy all Files to a Compact Flash Disk (use PCMCIA Slot in a Laptop, a SanDisk Compact Flash Reader or a similar product). Be sure, that the Windows NT Files are located in the directory \WINNT (same as with a standard NT installation).

- Reset and Start Up the target in RUN-mode (see DOC/IPC_NETIPC), the NT Embedded will be loaded from C:\WINNT

9 Linux Installation

Linux operating systems are installed using a bootable Linux Floppy Disk. Depending on the target board type (supported ROM-Drive image size) a standard 1.44 MByte Linux Boot Diskette or a specially prepared Linux Boot Diskette with all required files in the first 704 kByte of a standard 1.44 MByte Floppy disk may be used to generate a Linux ROM-Disk image to start with. The next step for installing a full Linux system may be done either by connecting a Linux CD-ROM to the IDE interface of the NETIPC target or by doing a network installation via the onboard Ethernet connectivity (requires network interface drivers in the ROM-Drive image).

If the target runs Embedded BIOS V5.x you may directly boot and install Linux from a bootable CDROM.

Some Linux companies provide simple to use installation tools for their Linux distribution to put them onto embedded systems without Floppy support (like the NETIPC).

Check the following links to find your solution:

- Linux Org:

www.linux.org

- Sysgo:

www.elinos.com

Note that you must make sure to have the legal right to load and run the selected Linux operating system on the target, i.e. to have a correct license. For the installation please proceed as follows:

1. **Prepare the Embedded BIOS** for a non-DOS boot drive:
 - Run the NETIPC BIOS Configuration Program on your host-PC
 - Read-in the BIOS image file
 - Edit the BIOS-image
 - Enter the Disk I/O and Booting menu
 - Select drive A: as first boot drive
 - Save options using <F1> key
 - Return to main menu via <ESC> key
 - Write BIOS image file
 - Exit (Note: you have now generated a new binary BIOS image)

2. **Create a binary image** of a non-DOS boot disk for the ROM-drive A: of the target:
 - change to directory C:\FBCUBE\NETIPCFW\NONDOSOS
 - Insert a system formatted 1.44 MByte 3.5" Linux boot diskette with all required files included into the floppy drive A: of your host-PC
 - Run the DOS batch file MAKEDRVX.BAT
 - Verify, that the files you need do not exceed the maximum ROM-drive size of 704 kBytes. Do that by examining the binary disk image file DRIVE_X.BIN with a HEX-viewer tool. At the end of the file you should encounter only values

3. **Download** the new BIOS and bootable ROM-drive into the target:
 - Connect the target COM1 port via a Null-Modem cable to the COMx-port of your host-PC (x = port number)
 - Run the NETIPC boot loader program
 - Power-on the target in the boot-load mode (see DOC/IPC_NETIPC)
 - Enter INITx after the boot loader program prompt (x = host-PC COM port number)
 - Wait for connection to target (Message „Target connected at“)
 - Enter EBIOS (or EBIOSX) to download the new Embedded BIOS
 - Wait for error-free completion (Note: takes several seconds, do never interrupt !)
 - Enter NDRIVE_X download the bootable ROM-drive A:
 - Wait for error-free completion (Note: takes several minutes, do never interrupt !)
 - Enter QUIT to automatically update the flash-tables and leave the boot loader program
 - Wait for error-free completion (Note: takes several seconds, do never interrupt !)

5. **Start Up the target:**

- Reset and Start Up the target in RUN-mode

The Linux operating system is now installed on the target. Please remember that you need a valid run-time license (not available from Syslogic).

Use MAKEDRV2.BAT instead of MAKEDRVX.BAT and NDRIVE_2 instead of NDRIVE_X for a 1.44 MByte ROM-Drive. See also chapter 4.4 for more information about generating a 704 kbyte or 1.44 Mbyte ROM-Disk image.

10 Non-DOS Operating System Installation

Specific operating systems (with the exception of the Embedded DOS-ROM) are not supported by Syslogic. However, the necessary steps for a drive A: based installation of a non-DOS operating system shall be explained here shortly. Note that you must make sure to have the legal right to load and run the selected operating system on the target, i.e. to have a correct license. For the installation please proceed as follows:

1. **Prepare the Embedded BIOS** for a non-DOS boot drive:
 - Run the NETIPC BIOS Configuration Program on your host-PC
 - Read-in the BIOS image file
 - Edit the BIOS-image
 - Enter the Disk I/O and Booting menu
 - Select drive A: as first boot drive
 - Save options using <F1> key
 - Return to main menu via <ESC> key
 - Write BIOS image file
 - Exit (Note: you have now generated a new binary BIOS image)

2. **Create a binary image** of a non-DOS boot disk for the ROM-drive A: of the target:
 - change to directory C:\FBCUBE\NETIPCFW\NONDOSOS
 - Insert a system formatted 1.44 MByte 3.5" OS boot diskette with all required files included into the floppy drive A: of your host-PC
 - Run the DOS batch file MAKEDRVX.BAT
 - Verify, that the files you need do not exceed the maximum ROM-drive size of 704 kBytes. Do that by examining the binary disk image file DRIVE_X.BIN with a HEX-viewer tool. At the end of the file you should encounter only values indicating unused (formatted) disk space.
The non-DOS boot ROM disk image is now ready in the file DRIVE_X.BIN

3. **Download** the new BIOS and bootable ROM-drive into the target:
 - Connect the target COM1 port via a Null-Modem cable to the COMx-port of your host-PC (x = port number)
 - Run the NETIPC boot loader program
 - Power-on the target in the boot-load mode (see DOC/IPC_NETIPC)
 - Enter INITx after the boot loader program prompt (x = host-PC COM port number)
 - Wait for connection to target (Message „Target connected at“)
 - Enter EBIOS (or EBIOSX) to download the new Embedded BIOS
 - Wait for error-free completion (Note: takes several seconds, do never interrupt !)
 - Enter NDRIVE_X to download the bootable ROM-drive A:
 - Wait for error-free completion (Note: takes several minutes, do never interrupt !)

- Wait for error-free completion (Note: takes several seconds, do never interrupt !)

4. Start Up the target:

- Reset and Start Up the target in RUN-mode (see DOC/IPC_NETIPC)

The non-DOS operating system is now installed on the target. Please remember that you need a valid run-time license (not available from Syslogic).

Use MAKEDRV2.BAT instead of MAKEDRVX.BAT and NDRIVE_2 instead of NDRIVE_X for a 1.44 MByte ROM-Drive. See also chapter 4.4 for more information about generating a 704 kbyte or 1.44 Mbyte ROM-Disk image.

11 BIOS Configuration for IPC-Modules

For some of the IPC boards special Embedded BIOS configuration options may be necessary. They are listed in this appendix. *Please remember that the Embedded BIOS V4.x file of the IPC/NETIPCFW-1A distribution is exclusively preconfigured for the CUB/NETIPC-1A board and the Embedded BIOS V5.x file is exclusively preconfigured for the IPC/NETIPC-5A board!*

11.1. NETIPC boards

The IPC/NETIPC-1B board provides 8 MBytes, the IPC/NETIPC-1A 32 Mbytes, the IPC/NETIPC-2A and -3A 64 Mbytes and the IPC/NETIPC-5A 128 MBytes of main memory. The correct target NETIPC board must be set in the 'Target Board' screen of the BCP (or BCPX) program to enable correct BIOS operation and the correct VGA BIOS extension must be downloaded to the target (see chapter 5.2).

12 BIOS Configuration for FBCube-Modules

For some of the FBCube modules special Embedded BIOS configuration options may be necessary. They are listed in this appendix. *Please remember that the Embedded BIOS V4.x file of the IPC/NETIPCFW-1A distribution is exclusively preconfigured for the CUB/NETIPC-1A board and the Embedded BIOS V5.x file is exclusively preconfigured for the IPC/NETIPC-5A board!*

12.1. NETIPC boards

The CUB/NETIPC-1B board provides 8 Mbytes and the CUB/NETIPC-1A 32 MBytes of main memory.

The correct target NETIPC board must be set in the 'Target Board' screen of the BCP (or BCPX) program to enable correct BIOS operation and the correct VGA BIOS extension must be downloaded to the target (see chapter 5.2).

12.2. CUB/SIC4-1A Serial Interface Module

The Embedded BIOS can be configured to support the channels of the CUB/SIC4-1A as standard COMx-ports. See paragraph 4.5.1 for more information.

The four channels A, B, C and D are fully compatible to the standard PC 16C550A UART except for the interrupt line, which is common to all four channels.

The following options must be set in the BCP COM port setup screen to define a CUB/SIC4-1A channel as COM-port:

- Set I/O port to the channels base address (e.g. Moduleaddress = 1 -> channels A/B/C/D = I/O ports 1100h/1108h/1110h/1118h).
- Set RTS Enable to NO if you use RS485 drivers to YES if you use RS232 drivers on the CUB/SIC4-1A.
- Set Clock Scale to 1 because the CUB/SIC4-1A uses a 3.6864 MHz UART clock.
- Leave the BIOS int14 init byte at standard int14 values.

12.3. CUB/CARFBC-1R

The CUB/CARFBC-1R features a shared RAM of 8 16 or 32 kbyte for data transfers. To be able to access this memory from the NETIPC board the jumpered address range must be redirected to the expansion bus in case it is occupied by a local memory resource on the CPU module (e.g. the main memory or FlashROM).

This must be done in the BCP(X) Memory Configuration setup screen.

13 Manufacturer Information

13.1. 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 send a fax or email message containing a short description of your application and your request to the following address or use one of the information or technical support request forms on our internet homepage:

Syslogic Datentechnik AG
Bruggerstr. 69
CH-5400 Baden / Switzerland

Email: info@syslogic.ch
Support: support@syslogic.ch
www: <http://www.syslogic.ch>
Fax: +41 (0)56 200 9040
Tel: +41 (0)56 200 9050

13.2. Warranty

Our products are covered by a world-wide manufacturers warranty. The warranty period starts at the delivery time from our official distributor to the customer. The duration of the warranty period is specified in the respective product catalogs and the offers. All products carry a date code and a job number for identification. The manufacturing data and deliveries are registered in a high level Quality Management System.

The warranty covers material and manufacturing defects. All products must be returned via the official distributor to the factory for repair or replacement. The warranty expires immediately if the products are damaged or operation outside of the specified recommended operating conditions. The warranty also expires if the date code or job number listed on the product is altered or rendered unintelligible. The warranty does not include damage due to errors in firmware or software delivered with the products.