Software

Developer Guide

www.tinyboards.com

Maximum Power at Minimum Size

Contact

Bluetechnix Mechatronische Systeme GmbH
Waidhausenstr. 3/19
A-1140 Vienna
AUSTRIA/EUROPE
office@bluetechnix.at
http://www.bluetechnix.at
http://www.tinyboards.com

Version 1.0

2005-05-03

Document No. 099-0001-11

Table of Contents

1 Developing projects with the Blackfin Core Modules 1
1.1 Developing projects using a JTAG device: 1
1.2 Developing projects without JTAG device: 1
1.2.1 Developing projects using BLACKSheep functions: 1
1.2.2 Developing projects without using BLACKSheep functions: 2
2 Standalone applications: 4
2.1 Applications booted by the Blackfin: 4
2.2 Applications booted by the BLACKSheep: 4
2.3 Flashing with a JTAG device: 5
3 Revision History 6
A List of Figures and Tables 7

Edition 2005-03

The information herein is given to describe certain components and shall not be considered as a guarantee of characteristics.
Terms of delivery and rights of technical change reserved.

We hereby disclaim any warranties, including but not limited to warranties of non-infringement, regarding circuits, descriptions and charts stated herein.

Bluetechnix makes and you receive no warranties or conditions, express, implied, statutory or in any communication with you. Bluetechnix specifically disclaims any implied warranty of merchantability or fitness for a particular purpose.

Bluetechnix takes no liability for any damages and errors causing of the usage of this board. The user of this board is responsible by himself for the functionality of his application. He is allowed to use the board only if he has the qualification. More information is found in the General Terms and Conditions (AGB).

Information
For further information on technology, delivery terms and conditions and prices please contact Bluetechnix (http://www.bluetechnix.at).

Warnings
Due to technical requirements components may contain dangerous substances

The Core Boards and Development systems contain ESD (electrostatic discharge) sensitive devices. Electrostatic charges readily accumulate on the human body and equipment and can discharge without detection. Permanent damage may occur on devices subjected to high-energy discharges. Proper ESD precautions are recommended to avoid performance degradation or loss of functionality. Unused core boards and development boards should be stored in the protective shipping package.

1 Developing projects with the Blackfin Core Modules

1.1 Developing projects using a JTAG device:

Our core modules works with every JTAG-device provided by ANLOG DEVICES INC. Please refer to the Analog Site www.analog.com for purchasing a JTAG device.
If you have a JTAG devices connected to the core module create an appropriate target platform with the VDSP++ configurator, depending on the core module you are using. Use this target device to connect your JTAG with the core module. You can develop projects using standard linker description files, and download the executable via the JTAG device. If you make an Intel hex compatible loader file with the VDSP++ you can flash this file, using the integrated flash programming tool. As flash driver you have to use the appropriate *.dxe flash driver file located on your support CD in the directory "*VDSPFlashToolDriver*", depending on witch core module you use. Refer to the VDSP++ manual, how to use the flashing tool.

1.2 Developing projects without JTAG device:

1.2.1 Developing projects using BLACKSheep functions:

If you want to create a output on the BLACKSheep console using the ANSI C printf() function, or you want to use the functions for file IO operations to access the SD- or CF-card if inserted or the RAM-disk, as linker description file you have to use the appropriate linker description file (*.ldf) from the CD and you have to include the stdio.h. Take a look at the sample provided with the CD to see how projects developed for the BLACKSheep looks like. In the same directory as your project resides there must be the executable (*.dxe) of the BLACKSheep running on your target core module. It must be the same version because the file IO functions will be statically linked into your project. And the addresses of the symbols of the functions must match to that of the BLACKSheep running on the core modules, otherwise they will not work.

With the file I/O functions you can access files on the SD-card if inserted or otherwise on the RAM-disk. Following functions are provided

-fopen
-fclose
-fputc
-fgetc
-printf
-Sleep

The printf() function creates a output on the BLACKSheep console.

Make a loaderfile with the VDSP++ using the following settings:

For the CM-BF533: PROM boot with 16 bit, no boot kernel, no splitter options.
For the CM-BF561: PROM boot with 16 bit, no boot kernel, no splitter options.

After creating the loader file you can download it, via your terminal program using the xmr command on the BLACKSheep console. Refer to the BLACKSheep_CommandReference to see how you transfer a file from the host PC to the target board. After the transfer is completed, type exec followed by the filename to execute your program, or use the flash command to flash it. Refer to the BLACKSheep_CommandReference for a description of the appropriate commands.

Be aware do don't switch the processor in the usermode within your code, because otherwise maybe some functions from the BLACKSheep does not work anymore. Like in projects that don't uses BLACKSheep functions you can overwrite within your code all the register settings. If you do, be aware that some of the BLACKSheep functions may be not work anymore. Refer to the BLACKSheep_SoftwareManual to see witch resources the BLACKSheep use.

1.2.2 Developing projects without using BLACKSheep functions:

If you don't want to use BLACKSheep functions you can use your own linker description file, and you are completely free in accessing the hardware register of the BLACKSheep. But you have to take care about the SDRAM memory used by the BLACKSheep. That means that in your linker description you can't define input sections that conflicts with the memory used by the BLACKSheep. Otherwise it is possible, that your code, during the load process by the BLACKSheep program loader, overwrites the code of the loader and the execution process aborts and the processor goes in an undefined state.

After you have created the Intel hex loader file you can download and execute it in the same way as described above in section 1.2.1.

The loader settings in this case are the same like for projects that uses BLACKSheep functions (*See section 1.2.1*). The so created loader file can also be flashed with the flash command from the BLACKSheep. Be aware that if you have overridden the BLACKSheep intentionally or unintentionally the only way to re-flash the BLACKSheep is, using the JTAG interface with the flash programming tool included in the VDSP++ environment.

The BLACKSheep provides commands to write an application in the flash, or erase the applications already written. During the boot process the BLACKSheep looks for applications in the flash and if found he tries to start it, if no key is pressed.

As long as you do not flash at the same section where the BLACKSheep is located and you do not change the default boot mode, after a reset the BLACKSheep starts again.

Be aware that even if you create a standalone project, the BLACKSheep code initializes many of the Blackfin processor registers after a reset. You can overwrite these settings within your own code if you need to do. You have full control over the Blackfin processor, because the BLACKSheep don't switch the processor in the user mode.

See the VDSP++ Manual for information about the VDSP++ compiler and loader.

2 Standalone applications:

2.1 Applications booted by the Blackfin:

The core modules are provided with a 16bit external flash starting at address 0x20000000. You can flash your code to create a standalone application. If you do so, be aware of the boot mode you use. The default boot mode on the core modules is BM0= 1 and BM1= 0 what means boot from external flash. The correct loader settings for creating hex-files for this boot mode are:

For the CM-BF533:

Boot mode	**Boot format**	**Output width**	
PROM	Intel hex	16 bit	Default start address

For the CM-BF561:

Boot mode	**Boot format**	**Output width**	
PROM	Intel hex	16 bit	Default start address

Disable the boot kernel option and the ROM splitter options.

Refer to the VDSP++ manual to see how to use the loader and how to change the loader settings and project options.

After you have created the loader file you can flash the file and after the next reset the application starts.

You can change the boot mode for the core modules by soldering other resistor configurations. Refer to the CM-BF5xx_HardwareUserManual to see where these resistors are located. If you change the boot mode you have to change also the loader settings within your project options to remain compatible with the boot mode selected.

2.2 Applications booted by the BLACKSheep:

As shortly described above you can use the BLACKSheep as boot loader for your application. The settings to create an appropriate loader file are the same as described in the above section. Just flash your application with the flash command provided by the BLACKSheep. The BLACKSheep, prior entering the command shell, looks if there is a valid application in flash. If an application was found, the application will be started, if no key is pressed. Refer to the *BLACKSheep_CommandReference* to see the syntax of the “flash” command.

2.3 Flashing with a JTAG device:

If you have a JTAG device connected on your core module you can use the flash programming tool included in the VDSP++ environment to flash the loader file. First of all load the flash driver into the core module using the appropriate flash driver located on your support CD as flash driver file. After you have loaded the driver choose the file with your application and start the flash procedure. Refer to the VDSP++ manual to see how to use the VDSP++ flash programming tool.

Be attention that flashing a standalone application overwrites the BLACKSheep code. The only way to re-flash the BLACKSheep is using a JTAG device!

For BLACKSheep updates and further documentation for the BLACKSheep and the target boards and core modules please visit the Bluetechnix download site on www.tinyboards.com

3 Revision History

2005 05 03	First release V1.0 of the Document

A List of Figures and Tables