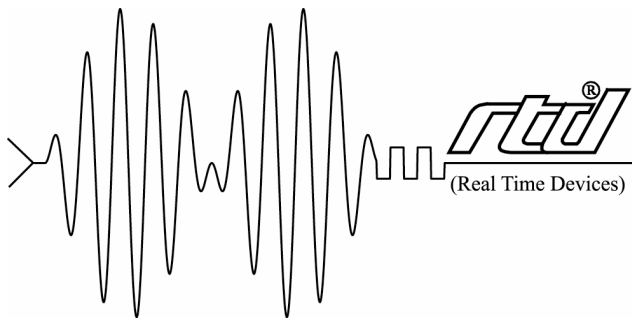


# APPLICATION NOTE

## Using DiskOnChip Under Linux With Memory Technology Device (MTD) Driver



RTD Embedded Technologies, Inc.  
*"Accessing the Analog World"®*

SWM-64000017  
rev A

---

---

# APPLICATION NOTE

## Using DiskOnChip Under Linux With Memory Technology Device (MTD) Driver

---

---



**RTD Embedded Technologies, INC.**

103 Innovation Blvd.  
State College, PA 16803-0906

Phone: +1-814-234-8087

FAX: +1-814-234-5218

E-mail

sales@rtd.com

techsupport@rtd.com

web site

<http://www.rtd.com>

Revision History

---

05/10/2004      Revision A issued

---

Published by:

RTD Embedded Technologies, Inc.  
103 Innovation Blvd.  
State College, PA 16803-0906

Copyright 2004 by RTD Embedded Technologies, Inc.  
All rights reserved  
Printed in U.S.A.

The RTD Logo is a registered trademark of RTD Embedded Technologies. cpuModule is a trademark of RTD Embedded Technologies. MS-DOS is a trademark of Microsoft Corp. All other trademarks appearing in this document are the property of their respective owners.

## Reference Configuration

This procedure is known to work with the following configuration:

- \* Base install of Red Hat Linux 7.3 performed on an RTD CMX27686 cpuModule
- \* RTD Enhanced Award BIOS version 4.51.12.15
- \* 2.4.25 kernel built with source code downloaded from [www.kernel.org](http://www.kernel.org) and with the CPU type set to 386
- \* MTD driver from CVS snapshot dated 04/27/2004
- \* 144 MB M-Systems DiskOnChip 2000
- \* Revision 0.4 of "Booting GRUB on the DiskOnChip 2000 and Millennium" HOWTO
- \* GRUB version 0.94 (obtained through [www.gnu.org](http://www.gnu.org)) with MTD driver patches applied
- \* DiskOnChip DOS utilities and BIOS driver version 5.1.4
- \* Creating a single partition on the device
- \* Using ext2 as the DiskOnChip file system type
- \* Compiling the MTD driver into the kernel rather than as a loadable module

## Introduction

To use a DiskOnChip under Linux, you need a device driver. The Open Source community has developed the Linux Memory Technology Device (MTD) driver for Solid State Devices (SSD). This software, distributed under the GPL, is included in 2.4 versions of the Linux kernel. More information can be found on the project's web site (<http://www.linux-mtd.infradead.org>).

This application note is based upon an online HOWTO (<http://lakeshoremicro.com/diskonchip-grub-howto.html>) called "Booting GRUB on the DiskOnChip 2000 and Millennium". This document functions as an adjunct to, and clarification of, this HOWTO. It is not a step-by-step recipe for configuring DiskOnChip functionality; that's the purpose of the web page. Rather, this application note: 1) adds supplementary information to the HOWTO, 2) makes corrections where needed, and 3) points out some things to watch for. By using the web page, this application note, and other resources as required, you should be able to get Linux to work with a DiskOnChip.

M-Systems, developer of DiskOnChip, recommends that you use their TrueFFS driver instead of the Linux MTD driver. The MTD driver developers recommend that you use the MTD driver instead of the M-Systems driver.

## For More Information

The Linux Documentation Project contains a HOWTO for making Linux boot disks. This document gives some ideas on selecting files for inclusion in a minimal, self-contained boot disk; when such a disk is booted, it provides a bare-bones Linux environment. This information can be generalized to creating a Linux boot environment on a DiskOnChip.

Kernel source code contains instructions on how to build the kernel. These directions can be found in the text file `linux-<kernel version>/README`, where `linux-<kernel version>` is the directory created when the kernel source was extracted.

O'Reilly ([www.oreilly.com](http://www.oreilly.com)) published a book "Building Embedded Linux Systems" written by Karim Yaghmour. This manuscript contains much information pertinent to those seeking to run Linux from Solid State Devices.

Please see the Application Note SWM-64000010 (Using a DiskOnChip Under Linux) on our web site for general information on DiskOnChip and Linux.

## Preliminaries

The hardware manual for your cpuModule contains a section on the setup required to begin using a DiskOnChip. Please refer to this manual for details on BIOS configuration.

You will need to download the DiskOnChip DOS utilities and BIOS driver (current version 5.1.4; zip file) from the M-Systems web site (<http://www.m-sys.com>). Unzip the DOS utilities to a directory. Create a DOS boot floppy disk. From the directory where you unzipped the utilities, copy the dformat.exe, dinfo.exe, and doc514.exb files onto the floppy disk. This boot disk will be used whenever you need to format the DiskOnChip. You may also wish to copy the getimage.exe and putimage.exe files onto the boot floppy.

Thoroughly read the following sections of the "Booting GRUB on the DiskOnChip 2000 and Millennium" HOWTO:

- \* Objectives
- \* Preliminaries: Kernel MTD Options
- \* Section #1: Kernel with latest MTD code [in its entirety]

If your DiskOnChip currently contains firmware, once you enable the extension window in the computer's BIOS you will need to remove the firmware. The extension window needs to be enabled whenever the device will be accessed for any reason, for example writing files to it or booting from it. If you wish to temporarily disable the device, you can simply turn off the BIOS extension window; you will not be able to access it again until you enable the window. To remove the firmware, use the dformat command on the DOS boot floppy you created earlier. This means you will need to boot from this floppy each time formatting the device is required. See the M-Systems DOS utilities documentation for more information about dformat and its options.

## Configuring Linux for DiskOnChip Usage

The organization of this document follows that of the web page HOWTO, with each section representing a section from that procedure. All section numbers below refer to the like-numbered sections from the "Section #1: Kernel with latest MTD code" portion; the "Section #2: Unmodified 2.4.18 Kernel" segment was not used in the creation of this application note.

### ***Objectives***

Please refer to the prior paragraph titled "Reference Configuration" for the software and hardware known to work with this procedure.

### ***Preliminaries: Kernel MTD Options***

When configuring the indicated kernel options, set them so that everything is built into the kernel rather than as loadable modules. Doing so makes the kernel larger but you do not have to worry about module issues and creating initial ramdisk images.

### ***1. Save DiskOnChip contents, Bad Block Table***

Running the commands referred to in this section require the DOS boot floppy created earlier. Boot from this floppy to perform these actions.

To save all the information indicated, you will need a DOS accessible/writable hard disk partition.

## **2. Get the latest MTD sources**

There's really no need to use CVS to access the latest version of the MTD driver. You can download the tar file from the FTP site mentioned in this section of the HOWTO.

Once you have the latest source code and have extracted it somewhere, browse through it and read the documentation to get a general idea of what's going on. These files can be found sprinkled throughout the directory structure.

To patch the kernel source code, use the patches/patchin.sh shell script; your current directory must be patches/ for this. The exact invocation used when creating this document was `./patchin.sh -c /usr/src/linux-2.4.25`. Please see the shell script for more information about the arguments it expects and accepts. The script automatically modifies the kernel source tree but does not modify the `linux-<kernel version>/Documentation/Configure.help` file. In the patches/ directory of the MTD source code exists a file named `Configure.help` which must be manually merged into the `Configure.help` file in the kernel source tree if you would like updated help regarding the MTD kernel configuration options. Once the patches are applied, you can then configure the kernel, build it, and install it.

Once you have a working kernel with MTD support booted, if you wish to use your DiskOnChip only as a storage device then you do not need to proceed any further. Simply format/partition it as desired and create any file system deemed necessary.

## **3. Patch GRUB**

Again, there's no real pressing need to use CVS to download the latest version of GRUB. Version 0.94 obtained via [www.gnu.org](http://www.gnu.org) works just fine as a starting point.

To patch the GRUB source code, change your current directory to the place where you extracted the GRUB source code. Then, issue the command `patch -p0 -i <path to MTD source>/patches/grub-2003-11-10-doc.patch`. The patch command will complain about the AUTHORS and ChangeLog files and not update them, but the rest of the source appears to be updated correctly.

Be very careful about which version of GRUB you elect to use. The GRUB patches supplied with the MTD driver may not successfully update that version's source code. Also, a patch that will correctly update the source code may not support using `dformat` to write the newly compiled GRUB to a DiskOnChip; in this scenario, the MTD `doc_loadbios` and `nftl_format` utilities would be utilized to respectively write GRUB to the device and to format the DiskOnChip.

## **4. Build GRUB**

A successful build of GRUB version 0.94 requires that the following tools/versions be installed on your build system: 1) automake version 1.7 or higher, and 2) autoconf version 2.57 or higher.

Currently, the RTD Enhanced Award BIOS does not support booting from a network device. Therefore, you will need to disable this functionality in GRUB when configuring the build process.

The exact invocation of the configure shell script used when creating this document was

```
./configure --enable-diskonchip-2000 --disable-diskonchip-biosnetboot --enable-diskonchip-ctrlbypass
--enable-ext2fs --disable-ffs --disable-xfs --disable-jfs --disable-vstafs --disable-reiserfs --disable-minix
--disable-fat"
```

On the system used to develop this application note, installing the GRUB firmware on the DiskOnChip did not override the boot sequence options set in the BIOS. More about this later.

## **5. Copy GRUB to DOC**

Copy the `stage1/grub_firmware` file onto the DOS boot floppy created earlier. You might also want to shorten the file name so that DOS does not create some non-mnemonic name because the original is too long. Boot from the DOS boot floppy to issue any of the `dformat` commands indicated.

What you are actually doing by writing GRUB on the DiskOnChip is replacing the M-Systems firmware that normally would be used to make the device appear as another disk drive to the BIOS. In other words, the firmware is not used when running the MTD driver; GRUB becomes the firmware. In contrast, when using LILO and the M-Systems driver, the firmware is used in addition to the boot loader.

## **6. Turn off PC**

On the RTD CMX `cpuModule` used in creating this document, a power cycle was required to fully set up the device after using `dformat` to install GRUB.

## **7. Change Boot Sequence**

Due to conditions dictated by the hardware and software used in developing this application note, booting from the DiskOnChip requires some changes in the BIOS. First, disable any hard drives in your system. Second, set the boot sequence to either "C, A, SCSI" or "A, C, SCSI". Save the changes and reboot.

Setting the boot sequence as directed in the previous paragraph allows the kernel to probe and find hard disks in your system. Since the DiskOnChip isn't set up yet as a fully self-contained bootable entity, you'll need access to a hard drive to copy files from it.

After you disable a hard drive in the BIOS, you will no longer be able to boot from it.

## **8. Verify GRUB boot**

Once the system begins booting from DiskOnChip, you should see some information printed about the configuration of the device and a message similar to "Scanning for NFTL Media Header".

Because hard drives were disabled in the BIOS in order to boot from the device, you will not be able to boot from a hard drive as indicated in (b) of the HOWTO by typing "kernel ..." and "boot" while at the GRUB prompt. Instead, you will need to reboot and enter the BIOS setup to enable the hard drives once again.

## **9. Verify DiskOnChip FAT Partition**

The FAT partition mentioned in this section was created when you wrote GRUB to the DiskOnChip with `dformat`.

## **10. Create ext2 partition on the DiskOnChip**

In (a), `vmlinuz-doc` is the kernel created in "2. Get the latest MTD sources". Replace it with the name of the kernel you built. This is also true in (b).

Note that `menu.lst` is `menu."ell"st` and not `menu."one"st`.

The HOWTO's author suggests under (b) that you can add an entry to `menu.lst` for booting from the hard disk. Because hard disks must be disabled in the BIOS to boot from DiskOnChip, they will not be visible to GRUB as boot devices. Therefore, you cannot add such an entry to the GRUB configuration file.

A discussion of how to populate a bootable Linux file system is beyond the scope of this document.

## Closing Comments

The MTD `doc_loadbios` and `nftl_format` utilities would not compile with kernel 2.4.25 source code. Their use is strongly discouraged. Instead, go the path of the latest kernel and MTD driver source code; this allows you to use `dformat` and thus bypass the MTD utilities altogether.