

[2.6 patch] remove the documentation for the legacy CDROM drivers

[2.6 patch] remove the documentation for the legacy CDROM drivers

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2007-07/msg03932.html>

- *From:* Adrian Bunk <bunk@xxxxxxxx>
 - *Date:* Wed, 11 Jul 2007 20:52:16 +0200
-

This patch removes the documentation for the removed legacy CDROM drivers.

Signed-off-by: Adrian Bunk <bunk@xxxxxxxx>

```
Documentation/cdrom/00-INDEX | 22
Documentation/cdrom/aztcd | 822 -----
Documentation/cdrom/cdu31a | 196 -----
Documentation/cdrom/cm206 | 185 -----
Documentation/cdrom/gscd | 60 -
Documentation/cdrom/isp16 | 100 --
Documentation/cdrom/mcdx | 29
Documentation/cdrom/optcd | 57 -
Documentation/cdrom/sbpcd | 1061 -----
Documentation/cdrom/sjcd | 60 -
Documentation/cdrom/sonycd535 | 122 ----
Documentation/kernel-parameters.txt | 37
MAINTAINERS | 12
13 files changed, 2763 deletions(-)
```

```
--- linux-2.6.22-rc6-mm1/Documentation/kernel-parameters.txt.old 2007-07-11 20:29:15.000000000
+0200
+++ linux-2.6.22-rc6-mm1/Documentation/kernel-parameters.txt 2007-07-11 20:30:31.000000000 +0200
@@ -34,7 +34,6 @@
APIC APIC support is enabled.
APM Advanced Power Management support is enabled.
AX25 Appropriate AX.25 support is enabled.
- CD Appropriate CD support is enabled.
DRM Direct Rendering Management support is enabled.
EDD BIOS Enhanced Disk Drive Services (EDD) is enabled
EFI EFI Partitioning (GPT) is enabled
@@ -319,9 +318,6 @@
```

autotest [IA64]

```
- aztcd= [HW,CD] Aztech CD268 CDROM driver
- Format: <io>,0x79 (?)
```

[2.6 patch] remove the documentation for the legacy CDROM drivers

[2.6 patch] remove the documentation for the legacy CDROM drivers

–

baycom_epp= [HW,AX25]

Format: <io>,<mode>

@@ -364,10 +360,6 @@

possible to determine what the correct size should be.

This option provides an override for these situations.

– cdu31a= [HW,CD]

– Format: <io>,<irq>[,PAS]

– See header of drivers/cdrom/cdu31a.c.

–

chandev= [HW,NET] Generic channel device initialisation

checkreqprot [SELINUX] Set initial checkreqprot flag value.

@@ -421,9 +413,6 @@

hpet= [X86-32,HPET] option to disable HPET and use PIT.

Format: disable

– cm206= [HW,CD]

– Format: { auto | [<io>],[<irq>] }

–

com20020= [HW,NET] ARCnet – COM20020 chipset

Format:

<io>[,<irq>[,<nodeID>[,<backplane>[,<ckp>[,<timeout>]]]]]

@@ -666,9 +655,6 @@

gpt [EFI] Forces disk with valid GPT signature but

invalid Protective MBR to be treated as GPT.

– gscd= [HW,CD]

– Format: <io>

–

gvp11= [HW,SCSI]

hashdist= [KNL,NUMA] Large hashes allocated during boot

@@ -849,9 +835,6 @@

tasks in the system -- can cause problems and

suboptimal load balancer performance.

– isp16= [HW,CD]

– Format: <io>,<irq>,<dma>,<setup>

–

iucv= [HW,NET]

js= [HW,JOY] Analog joystick

@@ -1006,11 +989,6 @@

mcatest= [IA-64]

– mcd= [HW,CD]

– Format: <port>,<irq>,<mitsumi_bug_93_wait>

[2.6 patch] remove the documentation for the legacy CDROM drivers

[2.6 patch] remove the documentation for the legacy CDROM drivers

```
-
- mcdx= [HW,CD]
-
mce [IA-32] Machine Check Exception

md= [HW] RAID subsystems devices and level
@@ -1243,9 +1221,6 @@
oprofile.timer= [HW]
Use timer interrupt instead of performance counters

- optcd= [HW,CD]
- Format: <io>
-
osst= [HW,SCSI] SCSI Tape Driver
Format: <buffer_size>,<write_threshold>
See also Documentation/scsi/st.txt.
@@ -1561,11 +1536,6 @@

sbni= [NET] Granch SBNI12 leased line adapter

- sbpcd= [HW,CD] Soundblaster CD adapter
- Format: <io>,<type>
- See a comment before function sbpcd_setup() in
- drivers/cdrom/sbpcd.c.
-
sc1200wdt= [HW,WDT] SC1200 WDT (watchdog) driver
Format: <io>[,<timeout>[,<isapnp>]]

@@ -1618,10 +1588,6 @@
simeth= [IA-64]
simscsi=

- sjcd= [HW,CD]
- Format: <io>,<irq>,<dma>
- See header of drivers/cdrom/sjcd.c.
-
slram= [HW,MTD]

slub_debug[=options[,slabs]] [MM, SLUB]
@@ -1798,9 +1764,6 @@

snd-ymfpci= [HW,ALSA]

- sonycd535= [HW,CD]
- Format: <io>[,<irq>]
-
sonypi.*= [HW] Sony Programmable I/O Control Device driver
See Documentation/sonypi.txt

--- linux-2.6.22-rc6-mm1/MAINTAINERS.old 2007-07-11 19:46:36.000000000 +0200
+++ linux-2.6.22-rc6-mm1/MAINTAINERS 2007-07-11 19:47:38.000000000 +0200
```

[2.6 patch] remove the documentation for the legacy CDROM drivers

@@ -2657,12 +2657,6 @@

W: <http://www.netlab.is.tsukuba.ac.jp/~yokota/izumi/ninja/>

S: Maintained

-NON-IDE/NON-SCSI CDROM DRIVERS [GENERAL] (come on, crew - mark your responsibility)

-P: Eberhard Moenkeberg

-M: emoenke@xxxxxxx

-L: linux-kernel@xxxxxxxxxxxxxxxxxxx

-S: Maintained

-

NTFS FILESYSTEM

P: Anton Altaparmakov

M: aia21@xxxxxxxxxxx

@@ -3127,12 +3121,6 @@

W: <http://www.mihu.de/linux/saa7146>

S: Maintained

-SBPCD CDROM DRIVER

-P: Eberhard Moenkeberg

-M: emoenke@xxxxxxx

-L: linux-kernel@xxxxxxxxxxxxxxxxxxx

-S: Maintained

-

SC1200 WDT DRIVER

P: Zwane Mwaikambo

M: zwane@xxxxxxxxxxxxxxxxxxx

--- linux-2.6.22-rc6-mm1/Documentation/cdrom/00-INDEX.old 2007-07-11 19:45:12.000000000 +0200

+++ linux-2.6.22-rc6-mm1/Documentation/cdrom/00-INDEX 2007-07-11 19:45:28.000000000 +0200

@@ -2,32 +2,10 @@

- this file (info on CD-ROMs and Linux)

Makefile

- only used to generate TeX output from the documentation.

-aztcd

- - info on Aztech/Orchid/Okano/Wearnes/Conrad/CyCDROM driver.

cdrom-standard.tex

- LaTeX document on standardizing the CD-ROM programming interface.

-cdu31a

- - info on the Sony CDU31A/CDU33A CD-ROM driver.

-cm206

- - info on the Philips/LMS cm206/cm260 CD-ROM driver.

-gscd

- - info on the Goldstar R420 CD-ROM driver.

ide-cd

- info on setting up and using ATAPI (aka IDE) CD-ROMs.

-isp16

- - info on the CD-ROM interface on ISP16, MAD16 or Mozart sound card.

-mcd

- - info on limitations of standard Mitsumi CD-ROM driver.

-mcdx

- - info on improved Mitsumi CD-ROM driver.

-optcd

[2.6 patch] remove the documentation for the legacy CDROM drivers

[2.6 patch] remove the documentation for the legacy CDROM drivers

```
-- info on the Optics Storage 8000 AT CD-ROM driver
packet-writing.txt
- Info on the CDRW packet writing module
-sbpcd
-- info on the SoundBlaster/Panasonic CD-ROM interface driver.
-sjcd
-- info on the SANYO CDR-H94A CD-ROM interface driver.
-sonycd535
-- info on the Sony CDU-535 (and 531) CD-ROM driver.

--- linux-2.6.22-rc6-mm1/Documentation/cdrom/aztcd 2007-04-26 05:08:32.000000000 +0200
+++ /dev/null 2006-09-19 00:45:31.000000000 +0200
@@ -1,822 +0,0 @@
-$Id: README.aztcd,v 2.60 1997/11/29 09:51:25 root Exp root $
- Readme-File Documentation/cdrom/aztcd
- for
- AZTECH CD-ROM CDA268-01A, ORCHID CD-3110,
- OKANO/WEARNES CDD110, CONRAD TXC, CyCDROM CR520, CR540
- CD-ROM Drives
- Version 2.6 and newer
- (for other drives see 6.-8.)
-
- NOTE: THIS DRIVER WILL WORK WITH THE CD-ROM DRIVES LISTED, WHICH HAVE
- A PROPRIETARY INTERFACE (implemented on a sound card or on an
- ISA-AT-bus card).
- IT WILL DEFINITELY NOT WORK WITH CD-ROM DRIVES WITH *IDE*-INTERFACE,
- such as the Aztech CDA269-031SE !!! (The only known exceptions are
- 'faked' IDE drives like the CyCDROM CR520ie which work with aztcd
- under certain conditions, see 7.). IF YOU'RE USING A CD-ROM DRIVE
- WITH IDE-INTERFACE, SOMETIMES ALSO CALLED ATAPI-COMPATIBLE, PLEASE
- USE THE ide-cd.c DRIVER, WRITTEN BY MARK LORD AND SCOTT SNYDER !
- THE STANDARD-KERNEL 1.2.x NOW ALSO SUPPORTS IDE-CDROM-DRIVES, SEE THE
- HARDDISK (!) SECTION OF make config, WHEN COMPILING A NEW KERNEL!!!
-----
-
-Contents of this file:
- 1. NOTE
- 2. INSTALLATION
- 3. CONFIGURING YOUR KERNEL
- 4. RECOMPILING YOUR KERNEL
- 4.1 AZTCD AS A RUN-TIME LOADABLE MODULE
- 4.2 CDROM CONNECTED TO A SOUND CARD
- 5. KNOWN PROBLEMS, FUTURE DEVELOPMENTS
- 5.1 MULTISESSION SUPPORT
- 5.2 STATUS RECOGNITION
- 5.3 DOSEMU's CDROM SUPPORT
- 6. BUG REPORTS
- 7. OTHER DRIVES
- 8. IF YOU DON'T SUCCEED ... DEBUGGING
- 9. TECHNICAL HISTORY OF THE DRIVER
- 10. ACKNOWLEDGMENTS
```

[2.6 patch] remove the documentation for the legacy CDROM drivers

- 11. PROGRAMMING ADD ONS: CDPLAY.C
 - APPENDIX: Source code of cdplay.c
-

-1. NOTE

- This software has been successfully in alpha and beta test and is part of
- the standard kernel since kernel 1.1.8x since December 1994. It works with
- AZTECH CDA268-01A, ORCHID CDS-3110, ORCHID/WEARNES CDD110 and CONRAD TXC
- (Nr.99 31 23 -series 04) and has proven to be stable with kernel
- versions 1.0.9 and newer. But with any software there still may be bugs in it.
- So if you encounter problems, you are invited to help us improve this software.
- Please send me a detailed bug report (see chapter BUG REPORTS). You are also
- invited in helping us to increase the number of drives, which are supported.

- Please read the README-files carefully and always keep a backup copy of your
- old kernel, in order to reboot if something goes wrong!

-2. INSTALLATION

- The driver consists of a header file 'aztcd.h', which normally should reside
- in /usr/src/linux/drivers/cdrom and the source code 'aztcd.c', which normally
- resides in the same place. It uses /dev/aztcd (/dev/aztcd0 in some distri-
- butions), which must be a valid block device with major number 29 and reside
- in directory /dev. To mount a CD-ROM, your kernel needs to have the ISO9660-
- filesystem support included.

- PLEASE NOTE: aztcd.c has been developed in parallel to the linux kernel,
- which had and is having many major and minor changes which are not backward
- compatible. Quite definitely aztcd.c version 1.80 and newer will NOT work
- in kernels older than 1.3.33. So please always use the most recent version
- of aztcd.c with the appropriate linux-kernel.

-3. CONFIGURING YOUR KERNEL

- If your kernel is already configured for using the AZTECH driver you will
- see the following message while Linux boots:
- Aztech CD-ROM Init: DriverVersion=<version number> BaseAddress=<baseaddress>
- Aztech CD-ROM Init: FirmwareVersion=<firmware version id of your I/O-card>>>
- Aztech CD-ROM Init: <drive type> detected
- Aztech CD-ROM Init: End

- If the message looks different and you are sure to have a supported drive,
- it may have a different base address. The Aztech driver does look for the
- CD-ROM drive at the base address specified in aztcd.h at compile time. This
- address can be overwritten by boot parameter aztcd=....You should reboot and
- start Linux with boot parameter aztcd=<base address>, e.g. aztcd=0x320. If
- you do not know the base address, start your PC with DOS and look at the boot
- message of your CD-ROM's DOS driver. If that still does not help, use boot
- parameter aztcd=<base address>,0x79 , this tells aztcd to try a little harder.
- aztcd may be configured to use autoprobng the base address by recompiling
- it (see chapter 4.).

- If the message looks correct, as user 'root' you should be able to mount the
- drive by

[2.6 patch] remove the documentation for the legacy CDROM drivers

- mount -t iso9660 -r /dev/aztcd0 /mnt
- and use it as any other filesystem. (If this does not work, check if
- /dev/aztcd0 and /mnt do exist and create them, if necessary by doing
- mknod /dev/aztcd0 b 29 0
- mkdir /mnt
-
- If you still get a different message while Linux boots or when you get the
- message, that the ISO9660-filesystem is not supported by your kernel, when
- you try to mount the CD-ROM drive, you have to recompile your kernel.
-
- If you do *not* have an Aztech/Orchid/Okano/Wearnes/TXC drive and want to
- bypass drive detection during Linux boot up, start with boot parameter aztcd=0.
-
- Most distributions nowadays do contain a boot disk image containing aztcd.
- Please note, that this driver will not work with IDE/ATAPI drives! With these
- you must use ide-cd.c instead.
-
- 4. RECOMPILING YOUR KERNEL
- If your kernel is not yet configured for the AZTECH driver and the ISO9660-
- filesystem, you have to recompile your kernel:
-
- Edit aztcd.h to set the I/O-address to your I/O-Base address (AZT_BASE_ADDR),
- the driver does not use interrupts or DMA, so if you are using an AZTECH
- CD268, an ORCHID CD-3110 or ORCHID/WEARNES CDD110 that's the only item you
- have to set up. If you have a soundcard, read chapter 4.2.
- Users of other drives should read chapter OTHER DRIVES of this file.
- You also can configure that address by kernel boot parameter aztcd=...
- aztcd may be configured to use autoprobing the base address by setting
- AZT_BASE_ADDR to '-1'. In that case aztcd probes the addresses listed
- under AZT_BASE_AUTO. But please remember, that autoprobing always may
- incorrectly influence other hardware components too!
- There are some other points, which may be configured, e.g. auto-eject the
- CD when unmounting a drive, tray locking etc., see aztcd.h for details.
- If you're using a linux kernel version prior to 2.1.0, in aztcd.h
- uncomment the line '#define AZT_KERNEL_PRIOR_2_1'
- Build a new kernel, configure it for 'Aztech/Orchid/Okano/Wearnes support'
- (if you want aztcd to be part of the kernel). Do not configure it for
- 'Aztech... support', if you want to use aztcd as a run time loadable module.
- But in any case you must have the ISO9660-filesystem included in your
- kernel.
- Activate the new kernel, normally this is done by running LILO (don't for-
- get to configure it before and to keep a copy of your old kernel in case
- something goes wrong!).
- Reboot
- If you've included aztcd in your kernel, you now should see during boot
- some messages like
- Aztech CD-ROM Init: DriverVersion=<version number> BaseAddress=<baseaddress>
- Aztech CD-ROM Init: FirmwareVersion=<firmware version id of your I/O-card>
- Aztech CD-ROM Init: <drive type> detected
- Aztech CD-ROM Init: End
- If you have not included aztcd in your kernel, but want to load aztcd as a

[2.6 patch] remove the documentation for the legacy CDROM drivers

- run time loadable module see 4.1.
- If the message looks correct, as user 'root' you should be able to mount
- the drive by
- mount -t iso9660 -r /dev/aztcd0 /mnt
- and use it as any other filesystem. (If this does not work, check if
- /dev/aztcd0 and /mnt do exist and create them, if necessary by doing
- mknod /dev/aztcd0 b 29 0
- mkdir /mnt
- If this still does not help, see chapters OTHER DRIVES and DEBUGGING.

-4.1 AZTCD AS A RUN-TIME LOADABLE MODULE

- If you do not need aztcd permanently, you can also load and remove the driver
- during runtime via insmod and rmmod. To build aztcd as a loadable module you
- must configure your kernel for AZTECH module support (answer 'm' when con-
- figuring the kernel). Anyhow, you may run into problems, if the version of
- your boot kernel is not the same than the source kernel version, from which
- you create the modules. So rebuild your kernel, if necessary.

- Now edit the base address of your AZTECH interface card in
- /usr/src/linux/drivers/cdrom/aztcd.h to the appropriate value.
- aztcd may be configured to use autoprobng the base address by setting
- AZT_BASE_ADDR to '-1'. In that case aztcd probes the addresses listed
- under AZT_BASE_AUTO. But please remember, that autoprobng always may
- incorrectly influence other hardware components too!
- There are also some special features which may be configured, e.g.
- auto-eject a CD when unmounting the drive etc; see aztcd.h for details.
- Then change to /usr/src/linux and do a
- make modules
- make modules_install
- After that you can run-time load the driver via
- insmod /lib/modules/X.X.X/misc/aztcd.o
- and remove it via rmmod aztcd.
- If you did not set the correct base address in aztcd.h, you can also supply the
- base address when loading the driver via
- insmod /lib/modules/X.X.X/misc/aztcd.o aztcd=<base address>
- Again specifying aztcd=-1 will cause autoprobng.
- If you do not have the iso9660-filesystem in your boot kernel, you also have
- to load it before you can mount the CDROM:
- insmod /lib/modules/X.X.X/fs/isofs.o
- The mount procedure works as described in 4. above.
- (In all commands 'X.X.X' is the current linux kernel version number)

-4.2 CDROM CONNECTED TO A SOUND CARD

- Most soundcards do have a bus interface to the CDROM-drive. In many cases
- this soundcard needs to be configured, before the CDROM can be used. This
- configuration procedure consists of writing some kind of initialization
- data to the soundcard registers. The AZTECH-CDROM driver in the moment does
- only support one type of soundcard (SoundWave32). Users of other soundcards
- should try to boot DOS first and let their DOS drivers initialize the
- soundcard and CDROM, then warm boot (or use loadlin) their PC to start
- Linux.

[2.6 patch] remove the documentation for the legacy CDROM drivers

- Support for the CDROM-interface of SoundWave32-soundcards is directly implemented in the AZTECH driver. Please edit linux/drivers/cdrom/aztcd.h, uncomment line '#define AZT_SW32' and set the appropriate value for AZT_BASE_ADDR and AZT_SW32_BASE_ADDR. This support was tested with an Orchid CDS-3110 connected to a SoundWave32.
- If you want your soundcard to be supported, find out, how it needs to be configured and mail me (see 6.) the appropriate information.

-5. KNOWN PROBLEMS, FUTURE DEVELOPMENTS

-5.1 MULTISESSION SUPPORT

- Multisession support for CD's still is a myth. I implemented and tested a basic support for multisession and XA CDs, but I still have not enough CDs and applications to test it rigorously. So if you'd like to help me, please contact me (Email address see below). As of version 1.4 and newer you can enable the multisession support in aztcd.h by setting AZT_MULTISESSION to 1. Doing so will cause the ISO9660-filesystem to deal with multisession CDs, ie. redirect requests to the Table of Contents (TOC) information from the last session, which contains the info of all previous sessions etc.. If you do set AZT_MULTISESSION to 0, you can use multisession CDs anyway. In that case the drive's firmware will do automatic redirection. For the ISO9660-filesystem any multisession CD will then look like a 'normal' single session CD. But nevertheless the data of all sessions are viewable and accessible. So with practically all real world applications you won't notice the difference. But as future applications may make use of advanced multisession features, I've started to implement the interface for the ISO9660 multisession interface via ioctl CDROMMULTISESSION.

-5.2 STATUS RECOGNITION

- The drive status recognition does not work correctly in all cases. Changing a disk or having the door open, when a drive is already mounted, is detected by the Aztech driver itself, but nevertheless causes multiple read attempts by the different layers of the ISO9660-filesystem driver, which finally timeout, so you have to wait quite a little... But isn't it bad style to change a disk in a mounted drive, anyhow ?!

- The driver uses busy wait in most cases for the drive handshake (macros STEN_LOW and DTEN_LOW). I tested with a 486/DX2 at 66MHz and a Pentium at 60MHz and 90MHz. Whenever you use a much faster machine you are likely to get timeout messages. In that case edit aztcd.h and increase the timeout value AZT_TIMEOUT.

- For some 'slow' drive commands I implemented waiting with a timer waitqueue (macro STEN_LOW_WAIT). If you get this timeout message, you may also edit aztcd.h and increase the timeout value AZT_STATUS_DELAY. The waitqueue has shown to be a little critical. If you get kernel panic messages, edit aztcd.c and substitute STEN_LOW_WAIT by STEN_LOW. Busy waiting with STEN_LOW is more stable, but also causes CPU overhead.

-5.3 DOSEMU's CD-ROM SUPPORT

- With release 1.20 aztcd was modified to allow access to CD-ROMS when running under dosemu-0.60.0 aztcd-versions before 1.20 are most likely to crash

[2.6 patch] remove the documentation for the legacy CDROM drivers

-Linux, when a CD-ROM is accessed under dosemu. This problem has partly been
-fixed, but still when accessing a directory for the first time the system
-might hang for some 30sec. So be patient, when using dosemu's CD-ROM support
-in combination with aztcd :-) !
-This problem has now (July 1995) been fixed by a modification to dosemu's
-CD-ROM driver. The new version came with dosemu-0.60.2, see dosemu's
-README.CDROM.

-6. BUG REPORTS

-Please send detailed bug reports and bug fixes via EMail to

- Werner.Zimmermann@xxxxxxxxxxxxxxxxxxxx

-Please include a description of your CD-ROM drive type and interface card,
-the exact firmware message during Linux bootup, the version number of the
-AZTECH-CDROM-driver and the Linux kernel version. Also a description of your
-system's other hardware could be of interest, especially microprocessor type,
-clock frequency, other interface cards such as soundcards, ethernet adapter,
-game cards etc..

-I will try to collect the reports and make the necessary modifications from
-time to time. I may also come back to you directly with some bug fixes and
-ask you to do further testing and debugging.

-Editors of CD-ROMs are invited to send a 'cooperation' copy of their
-CD-ROMs to the volunteers, who provided the CD-ROM support for Linux. My
-snail mail address for such 'stuff' is
- Prof. Dr. W. Zimmermann
- Fachhochschule fuer Technik Esslingen
- Fachbereich IT
- Flandernstrasse 101
- D-73732 Esslingen
- Germany

-7. OTHER DRIVES

-The following drives ORCHID CDS3110, OKANO CDD110, WEARNES CDD110 and Conrad
-TXC Nr. 993123-series 04 nearly look the same as AZTECH CDA268-01A, especially
-they seem to use the same command codes. So it was quite simple to make the
-AZTECH driver work with these drives.

-Unfortunately I do not have any of these drives available, so I couldn't test
-it myself. In some installations, it seems necessary to initialize the drive
-with the DOS driver before (especially if combined with a sound card) and then
-do a warm boot (CTRL-ALT-RESET) or start Linux from DOS, e.g. with 'loadlin'.

-If you do not succeed, read chapter DEBUGGING. Thanks in advance!

-Sorry for the inconvenience, but it is difficult to develop for hardware,
-which you don't have available for testing. So if you like, please help us.

[2.6 patch] remove the documentation for the legacy CDROM drivers

- If you do have a CyCDROM CR520ie thanks to Hilmar Berger's help your chances
- are good, that it will work with aztcd. The CR520ie is sold as an IDE-drive
- and really is connected to the IDE interface (primary at 0x1F0 or secondary
- at 0x170, configured as slave, not as master). Nevertheless it is not ATAPI
- compatible but still uses Aztech's command codes.
-
-
- 8. DEBUGGING : IF YOU DON'T SUCCEED, TRY THE FOLLOWING
- reread the complete README file
- make sure, that your drive is hardware configured for
- transfer mode: polled
- IRQ: not used
- DMA: not used
- Base Address: something like 300, 320 ...
- You can check this, when you start the DOS driver, which came with your
- drive. By appropriately configuring the drive and the DOS driver you can
- check, whether your drive does operate in this mode correctly under DOS. If
- it does not operate under DOS, it won't under Linux.
- If your drive's base address is something like 0x170 or 0x1F0 (and it is
- not a CyCDROM CR520ie or CR 940ie) you most likely are having an IDE/ATAPI-
- compatible drive, which is not supported by aztcd.c, use ide-cd.c instead.
- Make sure the Base Address is configured correctly in aztcd.h, also make
- sure, that /dev/aztcd0 exists with the correct major number (compare it with
- the entry in file /usr/include/linux/major.h for the Aztech drive).
- insert a CD-ROM and close the tray
- cold boot your PC (i.e. via the power on switch or the reset button)
- if you start Linux via DOS, e.g. using loadlin, make sure, that the DOS
- driver for the CD-ROM drive is not loaded (comment out the calling lines
- in DOS' config.sys!)
- look for the aztcd: init message during Linux init and note them exactly
- log in as root and do a mount -t iso9660 /dev/aztcd0 /mnt
- if you don't succeed in the first time, try several times. Try also to open
- and close the tray, then mount again. Please note carefully all commands
- you typed in and the aztcd-messages, which you get.
- if you get an 'Aztech CD-ROM init: aborted' message, read the remarks about
- the version string below.
-
- If this does not help, do the same with the following differences
- start DOS before; make now sure, that the DOS driver for the CD-ROM is
- loaded under DOS (i.e. uncomment it again in config.sys)
- warm boot your PC (i.e. via CTRL-ALT-DEL)
- if you have it, you can also start via loadlin (try both).
- ...
- Again note all commands and the aztcd-messages.
-
- If you see STEN_LOW or STEN_LOW_WAIT error messages, increase the timeout
- values.
-
- If this still does not help,
- look in aztcd.c for the lines #if 0
- #define AZT_TEST1

[2.6 patch] remove the documentation for the legacy CDROM drivers

- ...
- #endif
- and substitute '#if 0' by '#if 1'.
- recompile your kernel and repeat the above two procedures. You will now get
- a bundle of debugging messages from the driver. Again note your commands
- and the appropriate messages. If you have syslogd running, these messages
- may also be found in syslogd's kernel log file. Nevertheless in some
- installations syslogd does not yet run, when init() is called, thus look for
- the aztcd-messages during init, before the login-prompt appears.
- Then look in aztcd.c, to find out, what happened. The normal calling sequence
- is: aztcd_init() during Linux bootup procedure init()
- after doing a 'mount -t iso9660 /dev/aztcd0 /mnt' the normal calling sequence is
- aztcd_open() -> Status 2c after cold reboot with CDROM or audio CD inserted
- -> Status 8 after warm reboot with CDROM inserted
- -> Status 2e after cold reboot with no disk, closed tray
- -> Status 6e after cold reboot, mount with door open
- aztUpdateToc()
- aztGetDiskInfo()
- aztGetQChannelInfo() repeated several times
- aztGetToc()
- aztGetQChannelInfo() repeated several times
- a list of track information
- do_aztcd_request() }
- azt_transfer() } repeated several times
- azt_poll }
- Check, if there is a difference in the calling sequence or the status flags!
-
- There are a lot of other messages, eg. the ACMD-command code (defined in
- aztcd.h), status info from the getAztStatus-command and the state sequence of
- the finite state machine in azt_poll(). The most important are the status
- messages, look how they are defined and try to understand, if they make
- sense in the context where they appear. With a CD-ROM inserted the status
- should always be 8, except in aztcd_open(). Try to open the tray, insert an
- audio disk, insert no disk or reinsert the CD-ROM and check, if the status
- bits change accordingly. The status bits are the most likely point, where
- the drive manufacturers may implement changes.
-
- If you still don't succeed, a good point to start is to look in aztcd.c in
- function aztcd_init, where the drive should be detected during init. Do the
- following:
- reboot the system with boot parameter 'aztcd=<your base address>,0x79'. With
- parameter 0x79 most of the drive version detection is bypassed. After that
- you should see the complete version string including leading and trailing
- blanks during init.
- Now adapt the statement
- if ((result[1]=='A')&&(result[2]=='Z' ...)
- in aztcd_init() to exactly match the first 3 or 4 letters you have seen.
- Another point is the 'smart' card detection feature in aztcd_init(). Normally
- the CD-ROM drive is ready, when aztcd_init is trying to read the version
- string and a time consuming ACMD_SOFT_RESET command can be avoided. This is
- detected by looking, if AFL_OP_OK can be read correctly. If the CD-ROM drive

[2.6 patch] remove the documentation for the legacy CDROM drivers

- hangs in some unknown state, e.g. because of an error before a warm start or
- because you first operated under DOS, even the version string may be correct,
- but the following commands will not. Then change the code in such a way,
- that the ACMD_SOFT_RESET is issued in any case, by substituting the
- if-statement 'if (...=AFL_OP_OK)' by 'if (1)'.
-
- If you succeed, please mail me the exact version string of your drive and
- the code modifications, you have made together with a short explanation.
- If you don't succeed, you may mail me the output of the debugging messages.
- But remember, they are only useful, if they are exact and complete and you
- describe in detail your hardware setup and what you did (cold/warm reboot,
- with/without DOS, DOS-driver started/not started, which Linux-commands etc.)

-9. TECHNICAL HISTORY OF THE DRIVER

- The AZTECH-Driver is a rework of the Mitsumi-Driver. Four major items had to
- be reworked:

- a) The Mitsumi drive does issue complete status information acknowledging
- each command, the Aztech drive does only signal that the command was
- processed. So whenever the complete status information is needed, an extra
- ACMD_GET_STATUS command is issued. The handshake procedure for the drive
- can be found in the functions aztSendCmd(), sendAztCmd() and getAztStatus().

- b) The Aztech Drive does not have a ACMD_GET_DISK_INFO command, so the
- necessary info about the number of tracks (firstTrack, lastTrack), disk
- length etc. has to be read from the TOC in the lead in track (see function
- aztGetDiskInfo()).

- c) Whenever data is read from the drive, the Mitsumi drive is started with a
- command to read an indefinite (0xfffff) number of sectors. When the appropriate
- number of sectors is read, the drive is stopped by a ACDM_STOP command. This
- does not work with the Aztech drive. I did not find a way to stop it. The
- stop and pause commands do only work in AUDIO mode but not in DATA mode.
- Therefore I had to modify the 'finite state machine' in function azt_poll to
- only read a certain number of sectors and then start a new read on demand. As I
- have not completely understood, how the buffer/caching scheme of the Mitsumi
- driver was implemented, I am not sure, if I have covered all cases correctly,
- whenever you get timeout messages, the bug is most likely to be in that
- function azt_poll() around switch(cmd) case ACD_S_DATA.

- d) I did not get information about changing drive mode. So I doubt, that the
- code around function azt_poll() case AZT_S_MODE does work. In my test I have
- not been able to switch to reading in raw mode. For reading raw mode, Aztech
- uses a different command than for cooked mode, which I only have implemen-
- ted in the ioctl-section but not in the section which is used by the ISO9660.

- The driver was developed on an AST PC with Intel 486/DX2, 8MB RAM, 340MB IDE
- hard disk and on an AST PC with Intel Pentium 60MHz, 16MB RAM, 520MB IDE
- running Linux kernel version 1.0.9 from the LST 1.8 Distribution. The kernel
- was compiled with gcc.2.5.8. My CD-ROM drive is an Aztech CDA268-01A. My

[2.6 patch] remove the documentation for the legacy CDROM drivers

–drive says, that it has Firmware Version AZT26801A1.3. It came with an ISA–bus
–interface card and works with polled I/O without DMA and without interrupts.
–The code for all other drives was 'remote' tested and debugged by a number of
–volunteers on the Internet.

–Points, where I feel that possible problems might be and all points where I
–did not completely understand the drive's behaviour or trust my own code are
–marked with /*??*/ in the source code. There are also some parts in the
–Mitsumi driver, where I did not completely understand their code.

–10. ACKNOWLEDGMENTS

–Without the help of P.Bush, Aztech, who delivered technical information
–about the Aztech Drive and without the help of E.Moenkeberg, GWDG, who did a
–great job in analyzing the command structure of various CD–ROM drives, this
–work would not have been possible. E.Moenkeberg was also a great help in
–making the software 'kernel ready' and in answering many of the CDROM–related
–questions in the newsgroups. He really is *the* Linux CD–ROM guru. Thanks
–also to all the guys on the Internet, who collected valuable technical
–information about CDROMs.

–Joe Nardone (joe@xxxxxxxxxxxxxxxxxx) was a patient tester even for my first
–trial, which was more than slow, and made suggestions for code improvement.
–Especially the 'finite state machine' azt_poll() was rewritten by Joe to get
–clean C code and avoid the ugly 'gotos', which I copied from mcd.c.

–Robby Schirmer (schirmer@xxxxxxxxxxxxxxxxxx) tested the audio stuff (ioctl)s
–and suggested a lot of patches for them.

–Joseph Piskor and Peter Nugent were the first users with the ORCHID CD3110
–and also were very patient with the problems which occurred.

–Reinhard Max delivered the information for the CDROM–interface of the
–SoundWave32 soundcards.

–Jochen Kunz and Olaf Kaluza delivered the information for supporting Conrad's
–TXC drive.

–Hilmar Berger delivered the patches for supporting CyCDROM CR520ie.

–Anybody, who is interested in these items should have a look at 'ftp.gwdg.de',
–directory 'pub/linux/cdrom' and at 'ftp.cdrom.com', directory 'pub/cdrom'.

–11. PROGRAMMING ADD ONs: cdplay.c

–You can use the ioctl–functions included in aztcd.c in your own programs. As
–an example on how to do this, you will find a tiny CD Player for audio CDs
–named 'cdplay.c'. It allows you to play audio CDs. You can play a specified
–track, pause and resume or skip tracks forward and backwards. If you quit the
–program without stopping the drive, playing is continued. You can also
–(mis)use cdplay to read and hexdump data disks. You can find the code in the
–APPENDIX of this file, which you should cut out with an editor and store in a

[2.6 patch] remove the documentation for the legacy CDROM drivers

```
-separate file 'cdplay.c'. To compile it and make it executable, do
- gcc -s -Wall -O2 -L/usr/lib cdplay.c -o /usr/local/bin/cdplay # compiles it
- chmod +755 /usr/local/bin/cdplay # makes it executable
- ln -s /dev/aztcd0 /dev/cdrom # creates a link
- (for /usr/lib substitute the top level directory, where your include files
- reside, and for /usr/local/bin the directory, where you want the executable
- binary to reside )
-
-You have to set the correct permissions for cdplay *and* for /dev/mcd0 or
-/dev/aztcd0 in order to use it. Remember, that you should not have /dev/cdrom
-mounted, when you're playing audio CDs.
-
-This program is just a hack for testing the ioctl-functions in aztcd.c. I will
-not maintain it, so if you run into problems, discard it or have a look into
-the source code 'cdplay.c'. The program does only contain a minimum of user
-protection and input error detection. If you use the commands in the wrong
-order or if you try to read a CD at wrong addresses, you may get error messages
-or even hang your machine. If you get STEN_LOW, STEN_LOW_WAIT or segment violation
-error messages when using cdplay, after that, the system might not be stable
-any more, so you'd better reboot. As the ioctl-functions run in kernel mode,
-most normal Linux-multitasking protection features do not work. By using
-uninitialized 'wild' pointers etc., it is easy to write to other users' data
-and program areas, destroy kernel tables etc.. So if you experiment with ioctls
-as always when you are doing systems programming and kernel hacking, you
-should have a backup copy of your system in a safe place (and you also
-should try restoring from a backup copy first)!
-
-A reworked and improved version called 'cdtester.c', which has yet more
-features for testing CDROM-drives can be found in
-Documentation/cdrom/sbpcd, written by E.Moenkeberg.
-
-Werner Zimmermann
-Fachhochschule fuer Technik Esslingen
-(EMail: Werner.Zimmermann@xxxxxxxxxxxxxxxxxxx)
-October, 1997
-
-----
-APPENDIX: Source code of cdplay.c
-
-/* Tiny Audio CD Player
-
- Copyright 1994, 1995, 1996 Werner Zimmermann (Werner.Zimmermann@xxxxxxxxxxxxxxxxxxx)
-
-This program originally was written to test the audio functions of the
-AZTECH.CDROM-driver, but it should work with every CD-ROM drive. Before
-using it, you should set a symlink from /dev/cdrom to your real CDROM
-device.
-
-The GNU General Public License applies to this program.
-
-History: V0.1 W.Zimmermann: First release. Nov. 8, 1994
```

[2.6 patch] remove the documentation for the legacy CDROM drivers

```
- V0.2 W.Zimmermann: Enhanced functionality. Nov. 9, 1994
- V0.3 W.Zimmermann: Additional functions. Nov. 28, 1994
- V0.4 W.Zimmermann: fixed some bugs. Dec. 17, 1994
- V0.5 W.Zimmermann: clean 'scanf' commands without compiler warnings
- Jan. 6, 1995
- V0.6 W.Zimmermann: volume control (still experimental). Jan. 24, 1995
- V0.7 W.Zimmermann: read raw modified. July 26, 95
-*/
-
-#include <stdio.h>
-#include <ctype.h>
-#include <sys/ioctl.h>
-#include <sys/types.h>
-#include <fcntl.h>
-#include <unistd.h>
-#include <linux/cdrom.h>
-#include <linux/../../drivers/cdrom/aztcd.h>
-
-void help(void)
- { printf("Available Commands: STOP s EJECT/CLOSE e QUIT q\n");
-   printf(" PLAY TRACK t PAUSE p RESUME r\n");
-   printf(" NEXT TRACK n REPEAT LAST l HELP h\n");
-   printf(" SUB CHANNEL c TRACK INFO i PLAY AT a\n");
-   printf(" READ d READ RAW w VOLUME v\n");
- }
-
-int main(void)
- { int handle;
-   unsigned char command=' ', ini=0, first=1, last=1;
-   unsigned int cmd, i,j,k, arg1,arg2,arg3;
-   struct cdrom_ti ti;
-   struct cdrom_tochdr tocHdr;
-   struct cdrom_subchnl subchnl;
-   struct cdrom_tocentry entry;
-   struct cdrom_msf msf;
-   union { struct cdrom_msf msf;
-   } azt;
-   unsigned char buf[CD_FRAMESIZE_RAW];
-   struct cdrom_volctrl volctrl;
-
-   printf("\nMini-Audio CD-Player V0.72 (C) 1994,1995,1996 W.Zimmermann\n");
-   handle=open("/dev/cdrom",O_RDWR);
-   ioctl(handle,CDROMRESUME);
-
-   if (handle<=0)
-   { printf("Drive Error: already playing, no audio disk, door open\n");
-     printf(" or no permission (you must be ROOT in order to use this program)\n");
-   }
-   else
-   { help();
-     while (1)
```

[2.6 patch] remove the documentation for the legacy CDROM drivers

```
- { printf("Type command (h = help): ");
- scanf("%s",&command);
- switch (command)
- { case 'e': cmd=CDROMEJECT;
- ioctl(handle,cmd);
- break;
- case 'p': if (!ini)
- { printf("Command not allowed – play track first\n");
- }
- else
- { cmd=CDROMPAUSE;
- if (ioctl(handle,cmd)) printf("Drive Error\n");
- }
- break;
- case 'r': if (!ini)
- { printf("Command not allowed – play track first\n");
- }
- else
- { cmd=CDROMRESUME;
- if (ioctl(handle,cmd)) printf("Drive Error\n");
- }
- break;
- case 's': cmd=CDROMPAUSE;
- if (ioctl(handle,cmd)) printf("Drive error or already stopped\n");
- cmd=CDROMSTOP;
- if (ioctl(handle,cmd)) printf("Drive error\n");
- break;
- case 't': cmd=CDROMREADTOCHDR;
- if (ioctl(handle,cmd,&tocHdr)) printf("Drive Error\n");
- first=tocHdr.cdth_trk0;
- last= tocHdr.cdth_trk1;
- if ((first==0)|| (first>last))
- { printf ("—could not read TOC\n");
- }
- else
- { printf("—first track: %d —last track: %d —enter track number: ",first,last);
- cmd=CDROMPLAYTRKIND;
- scanf("%i",&arg1);
- ti.cdti_trk0=arg1;
- if (ti.cdti_trk0<first) ti.cdti_trk0=first;
- if (ti.cdti_trk0>last) ti.cdti_trk0=last;
- ti.cdti_ind0=0;
- ti.cdti_trk1=last;
- ti.cdti_ind1=0;
- if (ioctl(handle,cmd,&ti)) printf("Drive Error\n");
- ini=1;
- }
- break;
- case 'n': if (!ini++)
- { if (ioctl(handle,CDROMREADTOCHDR,&tocHdr)) printf("Drive Error\n");
- first=tocHdr.cdth_trk0;
```

[2.6 patch] remove the documentation for the legacy CDROM drivers

```
- last= tocHdr.cdth_trk1;
- ti.cdti_trk0=first-1;
- }
- if ((first==0)||((first>last))
- { printf ("--could not read TOC\n");
- }
- else
- { cmd=CDROMPLAYTRKIND;
- if (++ti.cdti_trk0 > last) ti.cdti_trk0=last;
- ti.cdti_ind0=0;
- ti.cdti_trk1=last;
- ti.cdti_ind1=0;
- if (ioctl(handle,cmd,&ti)) printf("Drive Error\n");
- ini=1;
- }
- break;
- case 'l': if (!ini++)
- { if (ioctl(handle,CDROMREADTOCHDR,&tocHdr)) printf("Drive Error\n");
- first=tocHdr.cdth_trk0;
- last= tocHdr.cdth_trk1;
- ti.cdti_trk0=first+1;
- }
- if ((first==0)||((first>last))
- { printf ("--could not read TOC\n");
- }
- else
- { cmd=CDROMPLAYTRKIND;
- if (--ti.cdti_trk0 < first) ti.cdti_trk0=first;
- ti.cdti_ind0=0;
- ti.cdti_trk1=last;
- ti.cdti_ind1=0;
- if (ioctl(handle,cmd,&ti)) printf("Drive Error\n");
- ini=1;
- }
- break;
- case 'c': subchnl.cdsc_format=CDROM_MSF;
- if (ioctl(handle,CDROMSUBCHNL,&subchnl))
- printf("Drive Error\n");
- else
- { printf("AudioStatus:%s Track:%d Mode:%d MSF=%d:%d:%d\n", \
- subchnl.cdsc_audiostatus==CDROM_AUDIO_PLAY ? "PLAYING":"NOT PLAYING",\
- subchnl.cdsc_trk,subchnl.cdsc_adr, \
- subchnl.cdsc_absaddr.msf.minute, subchnl.cdsc_absaddr.msf.second, \
- subchnl.cdsc_absaddr.msf.frame);
- }
- break;
- case 'i': if (!ini)
- { printf("Command not allowed - play track first\n");
- }
- else
- { cmd=CDROMREADTOCENTRY;
```

[2.6 patch] remove the documentation for the legacy CDROM drivers

```
- printf("Track No.: ");
- scanf("%d",&arg1);
- entry.cdte_track=arg1;
- if (entry.cdte_track<first) entry.cdte_track=first;
- if (entry.cdte_track>last) entry.cdte_track=last;
- entry.cdte_format=CDROM_MSF;
- if (ioctl(handle,cmd,&entry))
- { printf("Drive error or invalid track no.\n");
- }
- else
- { printf("Mode %d Track, starts at %d:%d:%d\n", \
- entry.cdte_adr,entry.cdte_addr.msf.minute, \
- entry.cdte_addr.msf.second,entry.cdte_addr.msf.frame);
- }
- }
- break;
- case 'a': cmd=CDROMPLAYMSF;
- printf("Address (min:sec:frame) ");
- scanf("%d:%d:%d",&arg1,&arg2,&arg3);
- msf.cdmsf_min0 =arg1;
- msf.cdmsf_sec0 =arg2;
- msf.cdmsf_frame0=arg3;
- if (msf.cdmsf_sec0 > 59) msf.cdmsf_sec0 =59;
- if (msf.cdmsf_frame0> 74) msf.cdmsf_frame0=74;
- msf.cdmsf_min1=60;
- msf.cdmsf_sec1=00;
- msf.cdmsf_frame1=00;
- if (ioctl(handle,cmd,&msf))
- { printf("Drive error or invalid address\n");
- }
- break;
-#ifdef AZT_PRIVATE_IOCTLLS /*not supported by every CDROM driver*/
- case 'd': cmd=CDROMREADCOOKED;
- printf("Address (min:sec:frame) ");
- scanf("%d:%d:%d",&arg1,&arg2,&arg3);
- azt.msf.cdmsf_min0 =arg1;
- azt.msf.cdmsf_sec0 =arg2;
- azt.msf.cdmsf_frame0=arg3;
- if (azt.msf.cdmsf_sec0 > 59) azt.msf.cdmsf_sec0 =59;
- if (azt.msf.cdmsf_frame0> 74) azt.msf.cdmsf_frame0=74;
- if (ioctl(handle,cmd,&azt.msf))
- { printf("Drive error, invalid address or unsupported command\n");
- }
- k=0;
- getchar();
- for (i=0;i<128;i++)
- { printf("%4d:",i*16);
- for (j=0;j<16;j++)
- { printf("%2x ",azt.buf[i*16+j]);
- }
- for (j=0;j<16;j++)
```

[2.6 patch] remove the documentation for the legacy CDROM drivers

```
- { if (isalnum(azt.buf[i*16+j]))
- printf("%c",azt.buf[i*16+j]);
- else
- printf(".");
- }
- printf("\n");
- k++;
- if (k>=20)
- { printf("press ENTER to continue\n");
- getchar();
- k=0;
- }
- }
- break;
- case 'w': cmd=CDROMREADRAW;
- printf("Address (min:sec:frame) ");
- scanf("%d:%d:%d",&arg1,&arg2,&arg3);
- azt.msf.cdmsf_min0 =arg1;
- azt.msf.cdmsf_sec0 =arg2;
- azt.msf.cdmsf_frame0=arg3;
- if (azt.msf.cdmsf_sec0 > 59) azt.msf.cdmsf_sec0 =59;
- if (azt.msf.cdmsf_frame0> 74) azt.msf.cdmsf_frame0=74;
- if (ioctl(handle,cmd,&azt))
- { printf("Drive error, invalid address or unsupported command\n");
- }
- k=0;
- for (i=0;i<147;i++)
- { printf("%4d:",i*16);
- for (j=0;j<16;j++)
- { printf("%2x ",azt.buf[i*16+j]);
- }
- for (j=0;j<16;j++)
- { if (isalnum(azt.buf[i*16+j]))
- printf("%c",azt.buf[i*16+j]);
- else
- printf(".");
- }
- printf("\n");
- k++;
- if (k>=20)
- { getchar();
- k=0;
- }
- }
- break;
-#endif
- case 'v': cmd=CDROMVOLCTRL;
- printf("--Channel 0 Left (0-255): ");
- scanf("%d",&arg1);
- printf("--Channel 1 Right (0-255): ");
- scanf("%d",&arg2);
```

[2.6 patch] remove the documentation for the legacy CDROM drivers

```
- volctrl.channel0=arg1;
- volctrl.channel1=arg2;
- volctrl.channel2=0;
- volctrl.channel3=0;
- if (ioctl(handle,cmd,&volctrl))
- { printf("Drive error or unsupported command\n");
- }
- break;
- case 'q': if (close(handle)) printf("Drive Error: CLOSE\n");
- exit(0);
- case 'h': help();
- break;
- default: printf("unknown command\n");
- break;
- }
- }
- }
- return 0;
-}
---- linux-2.6.22-rc6-mm1/Documentation/cdrom/cdu31a 2007-04-26 05:08:32.000000000 +0200
+++ /dev/null 2006-09-19 00:45:31.000000000 +0200
@@ -1,196 +0,0 @@
-
- CDU31A/CDU33A Driver Info
- -----
-
- Information on the Sony CDU31A/CDU33A CDROM driver for the Linux
- kernel.
-
- Corey Minyard (minyard@xxxxxxxxxxxxx)
-
- Colossians 3:17
-
- Crude Table of Contents
- -----
-
- Setting Up the Hardware
- Configuring the Kernel
- Configuring as a Module
- Driver Special Features
-
-
- This device driver handles Sony CDU31A/CDU33A CDROM drives and
- provides a complete block-level interface as well as an ioctl()
- interface as specified in include/linux/cdrom.h). With this
- interface, CDROMs can be accessed, standard audio CDs can be played
- back normally, and CD audio information can be read off the drive.
-
- Note that this will only work for CDU31A/CDU33A drives. Some vendors
- market their drives as CDU31A compatible. They lie. Their drives are
- really CDU31A hardware interface compatible (they can plug into the
```

[2.6 patch] remove the documentation for the legacy CDROM drivers

-same card). They are not software compatible.

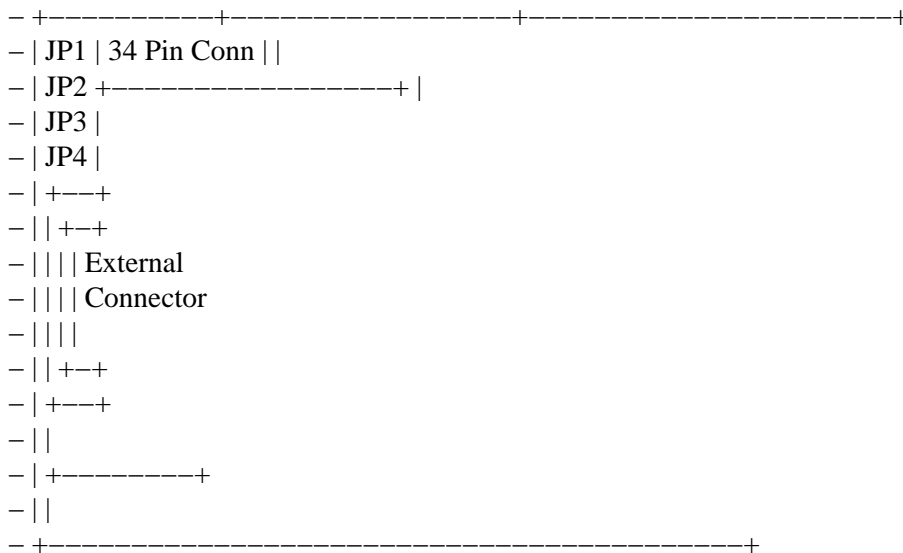
-
-Setting Up the Hardware

-

-The CDU31A driver is unable to safely tell if an interface card is
-present that it can use because the interface card does not announce
-its presence in any way besides placing 4 I/O locations in memory. It
-used to just probe memory and attempt commands, but Linus wisely asked
-me to remove that because it could really screw up other hardware in
-the system.

-
-Because of this, you must tell the kernel where the drive interface
-is, what interrupts are used, and possibly if you are on a PAS-16
-soundcard.

-
-If you have the Sony CDU31A/CDU33A drive interface card, the following
-diagram will help you set it up. If you have another card, you are on
-your own. You need to make sure that the I/O address and interrupt is
-not used by another card in the system. You will need to know the I/O
-address and interrupt you have set. Note that use of interrupts is
-highly recommended, if possible, it really cuts down on CPU used.
-Unfortunately, most soundcards do not support interrupts for their
-CDROM interfaces. By default, the Sony interface card comes with
-interrupts disabled.



-
- JP1 sets the Base Address, using the following settings:

-
- Address Pin 1 Pin 2

- 0x320 Short Short
- 0x330 Short Open
- 0x340 Open Short
- 0x360 Open Open

[2.6 patch] remove the documentation for the legacy CDROM drivers

-
- JP2 and JP3 configure the DMA channel; they must be set the same.
-
- DMA Pin 1 Pin 2 Pin 3
- -----
- 1 On Off On
- 2 Off On Off
- 3 Off Off On
-
- JP4 Configures the IRQ:
-
- IRQ Pin 1 Pin 2 Pin 3 Pin 4
- -----
- 3 Off Off On Off
- 4 Off Off* Off On
- 5 On Off Off Off
- 6 Off On Off Off
-
- The documentation states to set this for interrupt
- 4, but I think that is a mistake.
-
- Note that if you have another interface card, you will need to look at
- the documentation to find the I/O base address. This is specified to
- the SLCD.SYS driver for DOS with the /B: parameter, so you can look at
- you DOS driver setup to find the address, if necessary.
-
- Configuring the Kernel
- -----
-
- You must tell the kernel where the drive is at boot time. This can be
- done at the Linux boot prompt, by using LILO, or by using Bootlin.
- Note that this is no substitute for HOWTOs and LILO documentation, if
- you are confused please read those for info on bootline configuration
- and LILO.
-
- At the linux boot prompt, press the ALT key and add the following line
- after the boot name (you can let the kernel boot, it will tell you the
- default boot name while booting):
-
- cdu31a=<base address>,<interrupt>[,PAS]
-
- The base address needs to have "0x" in front of it, since it is in
- hex. For instance, to configure a drive at address 320 on interrupt 5,
- use the following:
-
- cdu31a=0x320,5
-
- I use the following boot line:
-
- cdu31a=0x1f88,0,PAS
-

[2.6 patch] remove the documentation for the legacy CDROM drivers

–because I have a PAS–16 which does not support interrupt for the
–CDU31A interface.
–
–Adding this as an append line at the beginning of the /etc/lilo.conf
–file will set it for lilo configurations. I have the following as the
–first line in my lilo.conf file:
–
– append="cdu31a=0x1f88,0"
–
–I'm not sure how to set up Bootlin (I have never used it), if someone
–would like to fill in this section please do.
–
–
–Configuring as a Module

–
–The driver supports loading as a module. However, you must specify
–the boot address and interrupt on the boot line to insmod. You can't
–use modprobe to load it, since modprobe doesn't support setting
–variables.
–
–Anyway, I use the following line to load my driver as a module
–
– /sbin/insmod /lib/modules/^uname -r/misc/cdu31a.o cdu31a_port=0x1f88
–
–You can set the following variables in the driver:
–
– cdu31a_port=<I/O address> – sets the base I/O. If hex, put 0x in
– front of it. This must be specified.
–
– cdu31a_irq=<interrupt> – Sets the interrupt number. Leaving this
– off will turn interrupts off.
–
–
–Driver Special Features

–
–This section describes features beyond the normal audio and CD–ROM
–functions of the drive.
–
–2048 byte buffer mode
–
–If a disk is mounted with –o block=2048, data is copied straight from
–the drive data port to the buffer. Otherwise, the readahead buffer
–must be involved to hold the other 1K of data when a 1K block
–operation is done. Note that with 2048 byte blocks you cannot execute
–files from the CD.
–
–XA compatibility
–
–The driver should support XA disks for both the CDU31A and CDU33A. It

[2.6 patch] remove the documentation for the legacy CDROM drivers

–does this transparently, the using program doesn't need to set it.
–
–Multi–Session
–
–A multi–session disk looks just like a normal disk to the user. Just
–mount one normally, and all the data should be there. A special
–thanks to Koen for help with this!
–
–Raw sector I/O
–
–Using the CDROMREADAUDIO it is possible to read raw audio and data
–tracks. Both operations return 2352 bytes per sector. On the data
–tracks, the first 12 bytes is not returned by the drive and the value
–of that data is indeterminate.
--- linux-2.6.22-rc6-mm1/Documentation/cdrom/cm206 2007-04-26 05:08:32.000000000 +0200
+++ /dev/null 2006-09-19 00:45:31.000000000 +0200
@@ -1,185 +0,0 @@
–This is the readme file for the dri–Thanks also to Eberhard Moenkeberg (emoenke@xxxxxxx) for prodding
–me to place this code into the mainstream Linux source tree
–(as of Linux version 1.1.91), as well as some patches to make
–it a better device citizen. Further thanks to Joel Katz
–<joelkatz@xxxxxxxxxxxx> for his MODULE patches (see details below),
–Porfiri Claudio <C.Porfiri@xxxxxxxxxxxxxxxxxxxxxxxx> for patches
–to make the driver work with the older CDU–510/515 series, and
–Heiko Eissfeldt <heiko@xxxxxxxxxxxxxxxxxxxxxxxx> for pointing out that
–the verify_area() checks were ignoring the results of said checks
–(note: verify_area() has since been replaced by access_ok()).
–
–(Acknowledgments from Ron Jeppesen in the 0.3 release:)
–Thanks to Corey Minyard who wrote the original CDU–31A driver on which
–this driver is based. Thanks to Ken Pizzini and Bob Blair who provided
–patches and feedback on the first release of this driver.
–
–Ken Pizzini
–ken@xxxxxxxxxxxx
–

–(The following is from Joel Katz <joelkatz@xxxxxxxxxxxx>.)
–
– To build a version of sony535.o that can be installed as a module,
–use the following command:
–
–gcc -c -D__KERNEL__ -DMODULE -O2 sonycd535.c -o sonycd535.o
–
– To install the module, simply type:
–
–insmod sony535.o
– or
–insmod sony535.o sonycd535=<address>
–
– And to remove it:

[2.6 patch] remove the documentation for the legacy CDROM drivers

-
- rmmmod sony535
-
- The code checks to see if MODULE is defined and behaves as it used
- to if MODULE is not defined. That means your patched file should behave
- exactly as it used to if compiled into the kernel.
-
- I have an external drive, and I usually leave it powered off. I used
- to have to reboot if I needed to use the CDROM drive. Now I don't.
-
- Even if you have an internal drive, why waste the 96K of memory
- (unswappable) that the driver uses if you use your CD-ROM drive infrequently?
-
- This driver will not install (whether compiled in or loaded as a
- module) if the CDROM drive is not available during its initialization. This
- means that you can have the driver compiled into the kernel and still load
- the module later (assuming the driver doesn't install itself during
- power-on). This only wastes 12K when you boot with the CDROM drive off.
-
- This is what I usually do; I leave the driver compiled into the
- kernel, but load it as a module if I powered the system up with the drive
- off and then later decided to use the CDROM drive.
-
- Since the driver only uses a single page to point to the chunks,
- attempting to set the buffer cache to more than 2 Megabytes would be very
- bad; don't do that.

-

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@xxxxxxxxxxxxxxxx

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>