Contents

1 Introduction 1

2 Installing a Linux distribution 3
   2.1 Ubuntu 7.10 ................................................. 3
   2.2 Mandriva 2008 ONE ....................................... 6
   2.3 Knoppix 3.9 ................................................. 10

3 Installing a real-time kernel 17
   3.1 Automatic (Ubuntu only) .................................. 17
      3.1.1 CPU Scaling Settings ................................. 17
   3.2 Manually .................................................. 18
      3.2.1 Startup/shutdown problems ......................... 25

4 EtherCAT for Unix 31
   4.1 Build Sources ............................................ 38
      4.1.1 Alternative timer in the EtherCAT Target .......... 40

5 TUEDACs 43
   5.1 Download software ...................................... 43
   5.2 Configure and build software ........................... 44
   5.3 Test program ............................................. 45

6 Miscellaneous 47
   6.1 Installing ps2 and ps4 printers ......................... 47
      6.1.1 In Ubuntu 7.10 ....................................... 47
      6.1.2 In Mandriva 2008 ONE ............................... 47
   6.2 Configure the internet connection ....................... 48
   6.3 Installing Matlab2007b for Unix ......................... 49
   6.4 Installing JAVA .......................................... 50
   6.5 Installing SmartSVN ..................................... 50
   6.6 Ubuntu 7.10, Gutsy Gibbon freezes every 10 minutes for approximately 10 sec 51
   6.7 Installing Syntek Semicon DC1125 Driver ................ 52

Bibliography 55

A Menu.lst HP desktop computer DCT lab WH -1.13 57
Chapter 1

Introduction

This document describes the steps needed in order to obtain a real-time operating system based on a Linux distribution. The installation of three distributions will be discussed, being Ubuntu 7.10, Mandriva 2008 ONE and Knoppix 3.0. Furthermore, it discusses the installation of programs like Matlab and several useful configuration settings are given.

The document is written for use in the DCT group of the Mechanical Engineering department, Eindhoven University of Technology. Some parts are specific for the working environment in mind and are not generally applicable.

The installation of the Linux distributions is described in Chapter 2. Two ways of installing a real-time kernel, the automatic way in Ubuntu and a more general way, are given in Chapter 3. How to incorporate EtherCAT or TUeDACs with your real-time Linux is explained in Chapter 4 and Chapter 5 respectively. Finally, the settings and installation of printers, internet, Matlab, Java, SmartSVN and more is contained in Chapter 6.
Chapter 2

Installing a Linux distribution

This chapter describes the installation of Ubuntu 7.10 in Section 2.1, Mandriva 2008 ONE in Section 2.2 and Knoppix 3.9 in 2.3 respectively.

2.1 Ubuntu 7.10

This section describes how to install Ubuntu 7.10, Gutsy Gibbon. Therefore, go to the website of Ubuntu [13], and download the distribution. This should be an image file ubuntu-7.10-desktop-i386.iso. Burn the image to a CD and reboot the system. Make sure that it boots from CD. This option can be set in your BIOS settings. During the booting of your system choose ‘Start or install Ubuntu’ which is the default option. After booting the desktop shown in Fig. 2.1 will appear.

![Startup desktop of Ubuntu 7.10.](image)

To install Ubuntu onto your harddisk, double-click the Install icon from the desktop. At the first step choose your language (English) and select ‘Forward’. At the second step choose
your time-zone. The time-zones are sorted by continent. Choose Europe, Amsterdam and select ‘Forward’. At the third step choose your keyboard settings. You can try the settings in the lower area of the setup screen. Default is U.S. English, U.S. English. Select ‘Forward’. At the fourth step you are asked to prepare the disks. Choose for the manual option here and select ‘Forward’. To determine which disks are your windows disks (C: and D:) make use of the following programs which can be started by clicking from the top menubar

- Places → Computer
- System → Administration → System monitor

For this case the following holds:

- `/dev/sda5`, which is the same as `/media/DATA` is the data disk under windows (D:\), see Fig. 2.2.

![Figure 2.2: Partition /dev/sda5.](image)

- `/dev/sda2`, which is the same as `/media/System` is the windows disk under windows (C:\), see Fig. 2.3.

- `/dev/sda1`, which is the same as `/media/RECOVERY` is the recovery disk under windows (not present on all computers).

Consecutively, select each partition, other than the windows partitions that should be kept, and delete those partitions by clicking ‘Delete partition’. In this case that would be the `/dev/sda6` and `/dev/sda7`, see also Fig. 2.4.

After deleting select ‘free space’ and click ‘New partition’. First make a logical partition of 2048 megabytes which is located at the beginning. Use this partition as a swap partition. This can be chosen in the ‘Use as’ drop down menu. See also Fig. 2.5 for the settings. Select ‘Ok’.
2.1. UBUNTU 7.10

Secondly, make a logical partition of the remaining size which is located also at the beginning. Use this partition as an ext3 partition. This can be chosen in the ‘Use as’ drop down menu. The mount point should be ‘/’. See also Fig. 2.6 for the settings. Select ‘Ok’.

Now, all disk space is partitioned. Turn on the ‘Format?’ option for the ext3 partition and select ‘Forward’. At step number five, do not Migrate Documents and Settings and select ‘Forward’. At the sixth step fill in the form about who you are. Make sure that the name of your computer coincides with the TUE name of the computer, in order to prevent
network problems. Select ‘Forward’. At the final step make sure that there is no internet connection and click ‘Install’ to install Ubuntu 7.10 to your harddisk. This may take a while, but is approximately 45 minutes depending on the specifications of your computer. During the installation you will get a message about security updates. Simply accept this message by clicking ‘Ok’. After the installation is completed a screen will pop-up in which you are asked to restart your computer to use the harddisk installed Ubuntu. Therefore click ‘Restart now’. The system will reboot now. During the restart a menu called GRUB will appear in which you can choose which operating system you would like to boot. In this case select ‘Ubuntu 7.10, kernel 2.6.22-14-generic’, which is the first default option. Use your username and password to log on to Ubuntu. After the startup a similar desktop as the previous startup should appear, see Fig. 2.1.

2.2 Mandriva 2008 ONE

The installation of the Mandriva distribution, in this case Mandriva 2008 ONE, will be described in this section. To download the .ISO file of the live CD go to the website of Mandriva [7]. Burn the image to a CD and boot the system from this CD. On boot you are asked to choose a language: ‘English (American)’ and your country: ‘Belgium’. Then ‘Accept’ the license agreement and choose ‘Next’. Select as keyboard layout the ‘US keyboard (international)’, confirm with ‘Next’. Choose as timezone ‘Amsterdam’ and ‘Next’. Choose the correct time, followed by ‘Next’. If the screen stays black, move the mouse. Choose ‘No 3D desktop effects’ and ‘Next’. Now the system will boot. After booting, the desktop of Fig. 2.7 will appear.

The installation of Mandriva on your harddisk can be started by clicking the ‘Live Install’ icon. The Mandriva Live installation wizard, shown in Fig. 2.8, will appear. Choose ‘Next’ to start the installation procedure.
2.2. MANDRIVA 2008 ONE

Since most systems already have a Windows installation, choose ‘Custom disk partitioning’ in the installation options step, as shown in Fig. 2.9. With this option you can resize, delete and create new partitions according to your own needs. You will be asked to backup your data first before proceeding. Choose ‘Exit’ if you want to backup first and do the installation later, otherwise ‘Continue’.

Next the present disk partitioning with sizes and types of the various partitions are shown. See for an example Fig. 2.10, which has two Windows partitions, one ext3 and one swap partition. Select the existing ext3 partition and delete the partition by clicking the button ‘Delete’. Delete also the existing swap partition. The drive partitioning should now contain only the Windows partitions, as shown in Fig. 2.11. If the system has more than one physical harddisk
drive, the different harddisk drives will appear as separate tabs in the disk partitioning screen.

The next step is to create a new swap partition and ext3 partition in the free space of the harddisk drive. Start making the swap partition by selecting the free space in the partitioning table and click ‘Create’. Now a screen appears where the type and size of the new partition can be defined. Choose a size of approximately 2048 MB and partition type a ‘Linux swap’ (see Fig. 2.12). Then click ‘Ok’. Make in the same way an ext3 partition in the free space with size of the remaining available space, type ‘Journalised FS: ext3’ and mount point ‘/’ (see Fig. 2.13).

Your newly made swap and ext3 partitions should be visible in the partition table as
2.2. MANDRIVA 2008 ONE

Figure 2.12: Create swap partition.

Figure 2.13: Create ext3 partition.

shown in Fig. 2.14. If you want to be completely sure the partition is empty, you can format the ext3 partition additionally by selecting it and clicking the ‘Format’ button. When you are satisfied with the partitioning of your disk, click ‘Done’ to proceed.

Mandriva will be installed on the ext3 partition. If you have not already formatted the ext3 partition you will be asked if you want to format a partition before the actual installation starts, as shown in Fig. 2.15.

After the installation is finished configure the GRUB startup menu. Reboot the system without the Live CD. During the startup, a GRUB menu will be started in which you can choose the operating system to boot. In the GRUB menu you will most probably see three options, one Windows startup option and two Mandriva Linux options. Choose the ‘Linux’ option to boot the system with your freshly installed Mandriva ONE 2008 Linux distribution.
2.3 Knoppix 3.9

In this section the installation of the Knoppix 3.9 Linux distribution for the TUE-DAX Linux Live DVD v3.2.6 is discussed. Boot the system from the Knoppix Live DVD. After startup the desktop appears, as shown in Fig. 2.16.

Open a konsole/shell by clicking on the monitor icon in the taskbar. In the shell, become super user by typing

```
$ su
```

Start the installation by typing (see also Fig. 2.17)

```
$ knoppix-installer
```

The Knoppix initialization (v0.3.19.12), as shown in Fig. 2.18 will appear. Click ‘OK’ to open the Knoppix installation menu of Fig. 2.19.

In the installation menu, choose option ‘1. Configure installation: Create a new configuration’. The system type selection dialog opens, choose ‘beginner: Multi-User System with hw-detection’ and proceed to the next step by clicking ‘Next’.

The first time Mandriva ONE 2008 is booted, the internet connection has to be configured and a user account is created. Follow the instructions on the screen to complete these steps. After booting, the desktop should look similar to Fig. 2.7 except without the ‘Live Install’ icon.

In Mandriva 2008 ONE, the sudo command is not standard available. In Mandriva 2008, update the sources by opening the ‘Install & Remove Software’ center. Choose subsequently ‘Options’ and ‘Media Manager’. Update now both the official updates and distribution sources. To make it available, type

```
$ urpmi sudo
```

After installing the sudo package, add the user sudoers file. Open for this the file /etc/sudoers and scroll to the line

```
# User privilege specification
```

Now add

```
<username> ALL=(ALL) ALL
```

where `<username>` is the user account that needs to acquire sudoers rights.
In the following step, choose the partition to install the Knoppix installation as shown in Fig. 2.21. Choose ‘Next’ to proceed to the second configuration step.

Input your name in the configuration step 2/7 dialog, see Fig. 2.22.

In step 3/7 (Fig. 2.23), input a username for your account on Knoppix 3.9. A suggestion
is given by the installation, but any other username can also be chosen, for example the
username of your TU/e account. Proceed by clicking ‘Next’. Now a password for the user
account of the previous step has to be created. Deselect the ‘Hide typing’ box to replace the
‘*’ by the true letters, as shown in Fig. 2.24.

In step 5/7 (Fig. 2.25), the administration password is created. Choose preferably the
same password as for the user account. Choose ‘Next’ to proceed to step six, where the
preferred hostname is created. As a suggestion ‘box’ is provided (see Fig. 2.26), but also another hostname can be chosen.

Figure 2.25: Create the administration password.

Figure 2.26: Create the hostname.

As the final step, the location where the boot-loader (grub) is to be installed has to be specified, as shown in Fig. 2.27. Choose the first option ‘mrb: Master Boot Record’ and choose ‘Next’ to complete the Knoppix Configuration. Next, the Knoppix installation menu of Fig. 2.19 appears. Now choose the second option ‘2. Start installation: Start the installation’ to start the installation. The Installation Summary containing the choices made during the configuration is shown, see Fig. 2.28 for an example.

Figure 2.27: Boot-loader installation.
CHAPTER 2. INSTALLING A LINUX DISTRIBUTION

By clicking ‘Next’ in the Knoppix installation summary, you will start the actual installation of the Knoppix distribution on your hard disk. A wait bar, as shown in Fig. 2.29, appears showing the progress of the installation. After the installation is completed, you will be asked to make a floppy disk, see Fig. 2.30. Choose ‘No’ to complete the installation.

When the installation is finished, reboot the system. In the GRUB menu, choose the option ‘Debian GNU/Linux, kernel 2.6.12.6-fusion-0.9.1 Default’ to boot your system with newly installed Knoppix Linux. To make the system working properly, two additional actions have to be performed.
• Remove `/KNOPPIX/` from the path in the files `/usr/bin/ml` and 
`/usr/local/matlabr14sp2/toolbox/qadscope/qs.m`.

• Add your account to the sudoers group. Open for this the file `/etc/sudoers` and scroll to the line `# User privilege specification`. Now add

```plaintext
<username>  ALL=(ALL)  ALL
```

where `<username>` is the user account that needs to acquire sudoers rights.

If you want to make Matlab run in a terminal, you can do the following.

• In the KDE desktop, open the properties of the Matlab icon.

• Open the Application tab.

• Choose ‘Advanced options’.

• Select ‘Run in terminal’.

If your system does not boot correctly, you might have to change the boot options. Open a shell and go to the folder `/boot/grub`. Now open the file `menu.lst`. An example of the boot options for the HP desktop computer in the DCT laboratorium (WH -1.13) is shown in Appendix A.
Chapter 3

Installing a real-time kernel

In the previous chapter, we installed a Linux distribution. This chapter describes in Section 3.1 an automatic way to get a real-time kernel in Ubuntu only. A more general but elaborate manual way to get a real-time kernel is presented in Section 3.2. But before this, an internet connection should be established (see also Section 6.2).

3.1 Automatic (Ubuntu only)

The automatic way to obtain a real-time Linux kernel, which is only applicable in the Ubuntu distribution, is to download and install a package called ‘linux-rt’. This can be done with the Synaptic Package Manager which can be found by ‘System’, ‘Administration’, ‘Synaptic Package Manager’. In order to be able to download and install more packages you can add repositories. Go to ‘Settings’, ‘Repositories’. Under the tab ‘Ubuntu Software’ select the first four repositories (main, universe, restricted, and multiverse). Close the ‘Software Sources’ window. Click ‘Reload’ to update the downloadable package information. In the right upper part of the screen of Synaptic search for the package ‘linux-rt’. Right-click the package and select ‘Mark for Installation’. Then apply the following windows. The package installs all related packages. More information on the package can be found in [11, 12]. During installation of the package, high resolution timers will be added together with a full preemption of the Linux Kernel. These features are made available by Thomas Gleixner and Ingo Molnar [16]. More information about the real-time preemptive kernel can be found on [17]. A new GRUB option becomes available after rebooting the system: ‘Ubuntu 7.10, kernel 2.6.22-14-rt’.

3.1.1 CPU Scaling Settings

The previous section described how to obtain a real-time Linux kernel in Ubuntu. However, in order to get good real-time performance additional settings must be applied according to [2]. If you installed the ‘linux-rt’ package the CPU scaling is still enabled. CPU scaling creates long-latency events when the CPU is put in a low-power consumption state after a period of inactivity. This section describes how to permanently set your CPU speed to its maximum. You can determine your CPU speed by typing the following in a Terminal

```
cat /proc/cpuinfo
```
CHAPTER 3. INSTALLING A REAL-TIME KERNEL

The actual speed of your CPU can be read from the field ‘cpu MHz’. This might be lower then what your processor is capable of. In order to permanently set it to the maximum CPU speed type the following in a Terminal

```
$ sudo dpkg-reconfigure gnome-applets
```

Accept the warning. Now you can add a panel to your top toolbar. Right-click on an empty space on the toolbar and click add panel. Search for ‘CPU Frequency Scaling Monitor’ and add it to the toolbar. After the icon is displayed you can set the CPU speed to its maximum.

### 3.2 Manually

The manual way to obtain a real-time Linux operation system with a full preemptive kernel is described in this section. Here the source files of the major kernel will be downloaded together with the real-time preemptive patch by Thomas Gleixner and Ingo Molnar [16, 17]. The source files will be unpacked, patched and installed. For more information see [17, 6, 15].

In order to obtain the recent kernel version type the following in a Terminal (open a Terminal by ‘Applications’, ‘Accessories’, ‘Terminal’)

```
$ uname -a
```

The first step is to download the sources files of the major kernel and the patch file to the directory `/usr/src`. Therefore open a Terminal and go to the directory `/usr/src` by typing

```
$ cd /usr/src
```

To become a superuser or root in Linux you need a password. In Ubuntu, this password is unknown and needs to be changed. Therefore, in Ubuntu start a Terminal and type

```
$ sudo passwd root
```

Enter the new password and retype it.

Become a superuser by typing (password required)

```
$ su
```

To get the source files of the major kernel 2.6.22, (no patches as in 2.6.22.1 or 2.6.22.2), type

```
$ cd /usr/src
   linux-2.6.22.tar.bz2
```

If this doesn’t work you can also start Mozilla, go to the website ftp://ftp.kernel.org/pub/linux/kernel/v2.6/ and download the file `linux-2.6.22.tar.bz2`. This file will be stored on your Desktop. You can move this file downloaded with Mozilla in a terminal to `/usr/src` by typing

```
$ mv /home/<username>/Desktop/linux-2.6.22.tar.bz2 /usr/src
```

where `<username>` is your username. Also download the patchfile by typing

```
$ wget http://www.kernel.org/pub/linux/kernel/projects/rt/older/
   patch-2.6.22.1-rt9
```
3.2. MANUALLY

Determine your current hardware by storing the results of

```
$ cd /sbin
$ lspci
```

This command lists the current PCI device information. Also store the results of

```
$ cat /proc/cpuinfo
```

This command lists the information of your CPU.

Having downloaded the necessary files, the source files should be extracted as (password required)

```
$ cd /usr/src
$ tar xfvj linux-2.6.22.tar.bz2
```

A new folder `linux-2.6.22` will be created in the directory `/usr/src`. Before building the kernel we need to check the system requirements. In Mandriva 2008, update the sources by opening the ‘Install & Remove Software’ center. Choose subsequently ‘Options’ and ‘Media Manager’. Update now both the official updates and distribution sources.

Open the Changes file in `/usr/src/linux-2.6.22/Documentation` for example with Open Office Word Processor which can be found under ‘Applications’, ‘Office’. Print this file (see Section 6.1). This file contains a section ‘Current Minimal Requirements’ which looks like.

- Gnu C 3.2 # gcc --version
- Gnu make 3.79.1 # make --version
- binutils 2.12 # ld -v
- util-linux 2.10 # fdformat --version
- module-init-tools 0.9.10 # depmod -V
- e2fsprogs 1.29 # tune2fs
- jfsutils 1.1.3 # fsck.jfs -V
- reiserfsprogs 3.6.3 # reiserfsck -V 2>&1|grep reiserfsprogs
- xfsprogs 2.6.0 # xfs_db -V
- pcmciautils 004 # pccardctl -V
- quota-tools 3.09 # quota -V
- PPP 2.4.0 # pppd --version
- isdn4k-utils 3.1pre1 # isdnctrl 2>&1|grep version
- nfs-utils 1.0.5 # showmount --version
- procps 3.2.0 # ps --version
- oprofile 0.9 # oprofiled --version
- udev 081 # udevinfo -V

The last column contains the Terminal command to check the version of the specific programs.

To determine the version of ‘reiserfsck’ type

```
$ reiserfsck -V
```

To determine the version of ‘isdnutils-base’ type

```
$ isdnctrl --version
```

If some programs are not installed, in Ubuntu you can use the Synaptic Package Manager which can be found by ‘System’, ‘Administration’, ‘Synaptic Package Manager’ to install the missing packages. In Mandriva use the ‘Install & Remove Software’ center. In order to be able to download and install more packages you can add repositories. The package installs all related packages. Preferably, you can use a command line version by typing in Ubuntu 7.10
$ sudo apt-get install <missing packagename>

or in Mandriva 2008 ONE

$ sudo urpmi <missing packagename>

where <missing packagename> is the name of the missing package. Make sure that the
Synaptic Package Manager is not already running. For Ubuntu 7.10 we had to install the
following packages:

- jfsutils,
- xfsprogs,
- quota,
- isdnutils-base, requires Ubuntu 7.10, Gutsy Gibbon Live CD,
- nfs-common, confirm by typing ‘Y’ and hit return,
- oprofile.

For Mandriva 2008 ONE, we had to update/install the following packages:

- gcc
- make
- oprofile

After having checked the requirements and installing the missing programs the sources
files of the kernel should be patched. In Ubuntu, the patch programm should be installed
first by typing in Ubuntu 7.10

$ sudo apt-get install patch

or in Mandriva 2008 ONE

$ sudo urpmi patch

Now patch the source files by typing

$ cd /usr/src/linux-2.6.22
$ sudo patch -p1 <../patch-2.6.22.1-rt9

The patching process takes a few seconds. Once the patching process is finished, it is wise to
make a backup of the patched sources files. This can be done by typing

$ cd /usr/src
$ mv linux-2.6.22 linux-2.6.22.1-rt9
$ tar cvfj linux-2.6.22.1-rt9 linux-2.6.22.1-rt9

Optionally, a unique name can be assigned to the kernel before making it. This can be done
by typing
$ cd /usr/src/linux-2.6.22.1-rt9

Open the Makefile and add ‘EXTRAVERSION = .1-rt9’. Now copy the config file from the running kernel and make a backup of the config file by typing

```
$ cd /boot
$ cp <current kernel> /usr/src/linux-2.6.22.1-rt9/.config
$ cd /usr/src/linux-2.6.22.1-rt9
$ cp .config config.save
```

Here `<current kernel>` is `config-2.6.22-14-generic` for Ubuntu 7.10 and `config-2.6.22.9-desktop586-lmdv` for Mandriva 2008 ONE. In Ubuntu only, additionally the following kernel-packages have to be installed

- `build-essential`
- `ncurses-dev`,

whereas in Mandriva the only package to be installed is `ncurses-dev`. Start the kernel configuration by typing

```
$ cd /usr/src/linux-2.6.22.1-rt9
$ make menuconfig
```

The Kernel Configuration menu, shown in Fig. 3.1 appears. Navigate though this menu using the arrow keys. Submenus can be selected using <Enter> and menus can be exited using <Esc>. With the <? button, the help menu can be opened for each option. Modules can be included using <Y>, excluded using <N> and modularized using <M>

Open the Code Maturity Level options menu and disable the ‘Prompt for development and/or incomplete code/drivers’ option as shown in Fig. 3.2.

Use the <Esc> button to return to the main menu and open the Loadable module support submenu. Change in the submenu the following option, see also Fig. 3.3.

```
- Module versioning support: disable
```

Return to the main menu and open the Processor Type and Features submenu. In this menu, set the following options (see also Fig. 3.4).

```
- Tickless System: enable
- High Resolution Timer Support: enable
- Symmetric multi-processing support: enable
- Subarchitecture type: PC-compatible
- Processor family: Pentium M (Note: depends on the hardware of your own system)
- Generic x86 support: enable
- Max number of CPUs: 8
- Preemption mode: Complete preemption (Real-Time)
- High memory support: 4 GB
- Allocate 3rd level pagetables from highmem: enable
- Boot from EFI support: enable
- Enable kernel irq balancing: enable
```
CHAPTER 3. INSTALLING A REAL-TIME KERNEL

- **Timer Frequency**: max: 1000 Hz

Now go to the Power Management Options submenu and set the following options. See also Fig. 3.5 up to Fig. 3.9 for screen shots of the various menus.

- **Legacy Power Management API**: disable
- **Power Management Debug Support**: disable
- **Driver model**: disable
- **ACPI** -->
  - Enable or module everything except Debug Statements
  - Disable ACPI for system before Jan 1st this year: 2000
- **APM**: module
- **CPU frequency scaling**: disable
- **CPU idle support**: disable

Return to the main menu using <Esc> and open the Bus Options submenu. Change the following settings as in Fig. 3.10.

- **Message Signaled Interrupts**: enable
- **ISA support**: disable
- **MCA support**: disable
- **NatSemi SCX200 support**: module

Next, go back to the main menu and open the Device Drivers submenu. Change the
3.2. MANUALLY

Sound options by choosing ‘Sound’, ‘Open Sound System’ and disable ‘Open sound system’, as shown in Fig. 3.11.

- Open sound system: disable

Return to the main menu and open the File Systems submenu and subsequently the DOS/FAT/NT Filesystems menu. Now change the following option, resulting in settings as shown in Fig. 3.12.

- NTFS write support: enable

Finally, open from the main menu the Kernel hacking submenu. Change the following options according Fig. 3.13.

- Show timing information on printks: disable
- Enable _must_check logic: disable
- Magic SysRq key: disable
- Enable unused/obsolete exported symbols: disable
- Detect soft lockups: disable
- Collect scheduler debugging info: disable
- Verbose bug reporting: disable
- Compile the kernel with debug info: enable
- Compile the kernel with frame pointers: disable
- Force gcc to inline functions marked ‘inline’: disable
- Early printk: disable
CHAPTER 3. INSTALLING A REAL-TIME KERNEL

Figure 3.3: Loadable Module Support menu.

After completing above settings, exit the Linux Kernel Configuration menu while saving
the settings. The real-time kernel is now ready to be build using the following commands in
a shell.

$ make clean
$ make
$ make modules_install
$ make install

Your kernel is now installed and ready to be used. Finally the startup image has to be created
and a new boot option has to be added to the boot GRUB menu. In Ubuntu, a startup image
has to be created from a shell as follows

$ cd /boot
$ mkinitramfs -o initrd-2.6.22.1-rt9.img 2.6.22.1-rt9

For adding the new boot option open the file menu.lst from the folder /boot/grub. Add the
following boot option to the file

title Real-Time Linux, kernel 2.6.22.1-rt9
root (hd0,3)
kernel /boot/vmlinuz-2.6.22.1-rt9
initrd /boot/initrd-2.6.22.1-rt9.img
Figure 3.4: Processor Type and Features menu.

quiet

Now reboot the system and choose the Real-Time Linux option from the GRUB menu to enjoy a fully preemptive real-time Linux kernel.

### 3.2.1 Startup/shutdown problems

Depending on your system, the following errors, warnings or problems may appear on boot or shutdown with the real-time kernel. In Table 3.2.1 several startup problems and the corresponding solutions encountered on the Mandriva 2008 ONE distribution in combination with an IBM Lenovo T60p ThinkPad are given. Note that the list is not intended to be complete, other problems may appear depending on the distribution and/or computer hardware.

If there is no sound after with the new kernel, perform the following steps from a shell console.

```
$ su
$ urpmi alsacof
$ alsacof
```

Choose the correct sound card e.g. ‘Intel 8x0’ and reboot the system after the configuration is finished. Now the sound should work.
CHAPTER 3. INSTALLING A REAL-TIME KERNEL

Figure 3.5: Power Management Options menu.

Figure 3.6: ACPI Support menu.
3.2. MANUALLY

Figure 3.7: APM Bios Support menu.

Figure 3.8: CPU Frequency Scaling menu.
CHAPTER 3. INSTALLING A REAL-TIME KERNEL

Figure 3.9: CPU Idle PM Support menu.

Figure 3.10: Bus Options menu.
3.2. MANUALLY

Figure 3.11: Device drivers -> Open Sound System menu.

Figure 3.12: File systems -> DOS/FAT/NT Filesystems menu.
### Table 3.1: Startup and shutdown problems and solutions.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segmentation fault with APM Deamon</td>
<td>Open the Mandriva Control Center, System tab and choose 'Manage System Services by enabling or disabling them'. Uncheck the 'APMD on boot' box.</td>
</tr>
<tr>
<td>Hal Deamon (bad exit status 8) build failed. Installation skipped</td>
<td>This error is caused by the service before, namely the DKMS. Remove DKMS on boot by opening the Mandriva Control Center, System tab and 'Manage System Services'. Uncheck the 'DKMS on boot' box.</td>
</tr>
<tr>
<td>Shorewall.  fatal module not found ipset v2.2.9a. error from kernel: protocol not available.</td>
<td>Open the Mandriva Control Center, System tab and 'Manage System Services'. Uncheck the ‘Shorewall on boot’ box.</td>
</tr>
</tbody>
</table>
Chapter 4

EtherCAT for Unix

In order to control a system, data acquisition is needed for actuating and measuring from which a feedback loop can be constructed. At Eindhoven University of Technology a well-known data acquisition system is present, called TUeDACS [14, 10], see also Chapter 5. Considering the TUeDACS/1, these systems mainly consists of 2 analogue output ports, two analogue input ports, 2 quadrature input ports and 8 digital I/O ports. These systems are used very commonly in the education program for experiments. If the number of required input and outputs is insufficient more TUeDACS systems can be connected to the host computer. In case of a control system, the control algorithm runs on the host computer itself, instead of the TUeDACS system. When dealing with high sampling frequencies in the order of kHz a lot of data communication is necessary between the host computer and the TUeDACS data acquisition system. This connection is established using PCMCIA or hi-speed USB 2.0, which limits the attainable sample frequency.

In the RoboCup [8] team Tech United [9] of Eindhoven University of Technology another data acquisition system is used. In the soccer playing robots the data acquisition is realized by EtherCAT (Ethernet for Control Automation Technology) [4, 18] which is the real-time Ethernet network originally developed by Beckhoff [1]. As already indicated by the name the data communication is realized by an ethernet connection. Compare the communication speeds of hispeed USB 2.0 (480 mbps) and Gigabit Ethernet (1000 mbps) it can be recognized that the allowable data communication for ethernet is more than twice as much with respect to hispeed USB 2.0. This section describes the steps needed in order to get a Beckhoff EtherCAT system running in combination which Matlab Simulink.

An EtherCAT system consists of a number of EtherCAT terminals like an EtherCAT coupler (the EK1x00), digital input ports (the EL1xxx), digital output ports (like for example EL2xxx), analogue input ports (the EL3xxx), analogue output ports (the EL4xxx), special functions like encoders input ports (the EL5xxx) and a system terminal (the EL9xxx). Examples of these modules are given in Fig. 4.1. Depending on your system requirements you can add these modules multiple times to the EtherCAT system by simply sliding them into each other like in Fig. 4.1. Apply the power to the 24V and 0V pins of the EK1100 unit. Also connect the + with 24V and - with 0V. This can be done as in Fig. 4.2. Obtain the EtherCAT zipfile and safe it to your desktop. Unpack the zipfile by right clicking the zipfile and select ‘Extract Here’. Now a folder is on your desktop called EtherCAT. Use a shell to go to this directory and copy the contents to the right place.

```
$ su <type root password>
```
CHAPTER 4. ETHERCAT FOR UNIX

Figure 4.1: Examples of some EtherCAT modules.

Figure 4.2: Apply the power to the EtherCAT modules.

```bash
$ cd /home/<username>/Desktop/EtherCAT/usr/lib
$ cp * /usr/lib
$ cd ../src
$ cp -r * /usr/src
$ cd ../local/matlabr14sp2
$ cp -r * /usr/local/<matlab>/
```

where <username> is your username and <matlab> is your matlab directory which depends on the matlab version you have installed on your system. On some systems all the sources need to be compiled for the specific system, for example executables which are build on x86 systems will be not recognized as such under a x64 system. In order to build the sources, proceed as in Section 4.1. At this moment it is time to configure the EtherCAT systems. After connecting the (straight) ethernet cable and applying 24 V to the EtherCAT system, this is done as follows.
where $<x>$ is the port of your ethernet that is connected with the EtherCAT system. With the $./ec_config\_eth<x>$ command the modules will be detected and configured. The output should look something like

```bash
$ Socket created: socket id: 3
$ Got interface: index: 0
$ ni = 805f288
$ EtherCAT\_AL: Skip slave initialization
$ EtherCAT\_AL: Ring contains 14 slaves
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ Slave 1 is of type EK1100 (revision 0x110000)
$ Slave 2 is of type EL1008 (revision 0x100000)
$ Slave 3 is of type EL4132 (revision 0x0)
$ Slave 4 is of type EL4132 (revision 0x0)
$ Slave 5 is of type EL4132 (revision 0x0)
$ Slave 6 is of type EL4132 (revision 0x0)
$ Slave 7 is of type EL5101 (revision 0x10000)
```
CHAPTER 4. ETHERCAT FOR UNIX

$ Slave 8 is of type EL5101 (revision 0x10000)
$ Slave 9 is of type EL5101 (revision 0x10000)
$ Slave 10 is of type EL5101 (revision 0x10000)
$ Slave 11 is of type EL5101 (revision 0x10000)
$ Slave 12 is of type EL5101 (revision 0x10000)
$ Slave 13 is of type EL5101 (revision 0x10000)
$ Slave 14 is of type EL5101 (revision 0x10000)

If all the modules where detected properly, the configuration is successful. If a module is detected but the revision is unknown you will receive the following message:

$ Socket created: socket id: 3
$ Got interface: index: 0
$ ni = 805f288
$ EtherCAT_AL: Skip slave initialization
$ EtherCAT_AL: Ring contains 14 slaves
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ Slave 0 is of type EK1100, but revision is unknown!

This can be overcome by editing two files:

1. /usr/local/$<$matlab$>$/rtw/c/ectarget/ec\_new/ec.cpp
First consider the first file: ec.cpp. In Ubuntu, the editing can be done using the program gedit (you can use any another editor in the different Linux distributions)

```
$ su <type root password>
$ cd /usr/local/<matlab>/rtw/c/ectarget/ec_new
$ gedit ec.cpp
```

Change line number 337

```
printf("Slave %d is of type EK1100, but revision is unknown\n", i);
```

in

```
printf("Slave %d is of type EK1100, but revision is unknown: (revision 0x%x)! \n", i, rev[i]);
```

Save the file and leave gedit. Rebuild the configuration tool and run the configuration.

```
$ su <type root password>
$ cd /usr/local/<matlab>/rtw/c/ectarget/ec_new
$ ./buildconfig
$ ./ec_config eth<x>
```

Now the revision is visible:

```
$ Socket created: socket id: 3
$ Got interface: index: 0
$ ni = 805f288
$ EtherCAT_AL: Skip slave initialization
$ EtherCAT AL: Ring contains 14 slaves
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ Slave 0 is of type EK1100, but revision is unknown: (revision 0x110000)!
```
In this case the revision number is 110000. Again edit the `ec.cpp` file. Now go to line number 333 and add an extra case (`case 0x00<revision number>`), where `<revision number>` is the revision number. So in this case the file is edit as follows

```c
331 switch (rev[i]) {
332 case 0x00010000:
333 case 0x00100000:
334 case 0x00110000:
335     printf("Slave %d is of type EK1100 (revision 0x%x)\n", i+1, rev[i]);
336     break;
337     default:
338     printf("Slave %d is of type EK1100, but revision is unknown: (revision 0x%x)! \n", i, rev[i]);
339     return -1;
340     //break;
341 }
```

Save the file, leave gedit. The second file that should be edit is `ec_config.c`. Now go to line number 433 and add an extra case (`case 0x00<revision number>`), where `<revision number>` is the revision number. So in this case the file is edit as follows

```c
428 switch (prod[i]) {
429     case 0x044c2c52:
430         nEK1100++;
431     switch (rev[i]) {
432         case 0x00010000:
433         case 0x00100000:
434         case 0x00110000:
435             fprintf(fp, " slave_db->set_conf(&EC_EK1100_%d, idx);
436             fprintf(fp, " idx++;\n\n");
437             break;
438         default:
439             break;
440     }
```

Save the file, leave gedit, rebuild the configuration tool and run the configuration.

```bash
$ su <type root password>
$ cd /usr/local/<matlab>/rtw/c/ectarget/ec_new
$ ./buildconfig
$ ./ec_config eth<x>
```

Now the output should look something like

```bash
$ Socket created: socket id: 3
$ Got interface: index: 0
$ ni = 805f288
$ EtherCAT_AL: Skip slave initialization
$ EtherCAT AL: Ring contains 14 slaves
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
```
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ EEPROM busy
$ Slave 1 is of type EK1100 (revision 0x110000)
$ Slave 2 is of type EL1008 (revision 0x100000)
$ Slave 3 is of type EL4132 (revision 0x0)
$ Slave 4 is of type EL4132 (revision 0x0)
$ Slave 5 is of type EL4132 (revision 0x0)
$ Slave 6 is of type EL4132 (revision 0x0)
$ Slave 7 is of type EL5101 (revision 0x10000)
$ Slave 8 is of type EL5101 (revision 0x10000)
$ Slave 9 is of type EL5101 (revision 0x10000)
$ Slave 10 is of type EL5101 (revision 0x10000)
$ Slave 11 is of type EL5101 (revision 0x10000)
$ Slave 12 is of type EL5101 (revision 0x10000)
$ Slave 13 is of type EL5101 (revision 0x10000)
$ Slave 14 is of type EL5101 (revision 0x10000)

If all the modules were detected properly, the configuration is successful.
Having configured the EtherCAT system Matlab can be started.

$ matlab

Within matlab go to the folder /usr/local/<matlab>/rtw/c/ectarget. Add this folder to the path of matlab. In this folder run make.m by typing the following in the command window of matlab

$ make

The file make.m compiles the s-functions into executable files which can be called from within matlab. In the folder you will find two simulink files, namely ec_test01.mdl and ec_lib.mdl. The first file contains a model with five blocks in it which are the EtherCAT interface blocks to EL5101, EL2004, EL4132, EL3102 and another EL4132. Hence, you can have more than one block of same module. However, make sure that you assign different links for each block,
which can be done by double clicking the blocks. The other simulink file contains the library of the interface blocks of all supported the EtherCAT modules. For now, add or remove blocks to the ec_test01.mdl such that it matches your EtherCAT configuration and save it to your desktop as test.mdl, so save it to /home/<username>/Desktop. Before building the executable, first the correct ethernet port must be set. This can be done by selecting ‘Simulation’, ‘Configuration Parameters’, ‘Real-Time Workshop’. The make command is make_rtw OPTS=’-DNETIF_ID=<x>’ . Here <x> should be the port of your ethernet that is connected with the EtherCAT system. Apply the settings and click Ok. Set the current directory of matlab to /home/<username>/Desktop by typing the following in the command window of matlab

```matlab
>> cd /home/<username>/Desktop
```

Now we install another custom made library called COIN or COmmand INterpreter, written by Jos Banens. Obtain the coin zipfile and save it to your desktop. Unpack the zipfile by right clicking the zipfile and select ‘Extract Here’. Now a folder is on your desktop called coin. Use a shell to go to this directory and build the coin library

```bash
$ su <type root password>
$ cd /home/<username>/Desktop/coin
$ ./buildlib
```

When trying to build the test.mdl you will notice an error in /usr/local/<matlab>/rtw/c/ectarget/ec_main.c. This is caused by the fact that there are lines commented with // xxx instead of /* xxx */. Replace all comments with /* xxx */ or add ANSI_OPTS=’’ to the make command of the Simulink file, which can be found in ‘Simulation’, ‘Configuration Parameters’, ‘Real-Time Workshop’. After all replacements and trying to build the test.mdl still an error occurs. Now it complains about a hidden symbol. This can be overcome by editing the ec_unix.tmf file. Edit line number 204 such that it becomes

```bash
LDFLAGS = -L/usr/lib -lpthread -ltimer_posix -leml /home/<username>/Desktop/libu1/libu1.c
```

The last thing that should be edited is the file /usr/local/<matlab>/rtw/c/ectarget/ec_main.c. On line number 563 the path of the Matlab is hard coded. Replace it with the correct path, i.e. replace matlabr14sp2 with for example matlab75. Now you are ready to build the executable. Therefore, check if the current directory of Matlab is /home/<username>/Desktop and select the Simulink model test.mdl. Press CTRL+B to build the executable. After successful completion you can start the executable with (make sure that the EtherCAT power is on and that it is connected to your system)

```bash
$ su <type root password>
$ cd /home/<username>/Desktop
$ ./test
```

### 4.1 Build Sources

This section describes how to build the sources needed for getting started with EtherCAT. There are two libraries that should be build:

- Timer-posix, can be found in /usr/src
4.1. BUILD SOURCES

- `eml_linux`, can be found in `/usr/src`

and one executable

- `ec_config`, located in `/usr/local/<matlab>/rtw/c/ectarge/ec_new`.

To build the `Timer_posix` library written by René van de Molengraft type

```
$ su <type root password>
$ cd /usr/src/Timer_posix
$ ./buildlib
```

Now the `Timer_posix` library is build. A shared library called `libtimer_posix.so` is now included in `/usr/lib`.

To build the `eml_linux` (EtherCAT Master Library for Linux) do the following

```
$ su <type root password>
$ cd /usr/src/eml_linux/build_rtnet
$ rm -rf *
$ ccmake ..
```

With the above commands you enter the directory `/usr/src/eml_linux/build_rtnet`, you remove all the content and start ccmake. It might be that this package is not installed on your system. If so, install the package called cmake (be aware: the package is called cmake, whereas the program is ccmake) with the Synaptics package manager. If ccmake runs you enter a configuration screen. Except the defaults by typing 'c', followed by 'c'. Now the option generate becomes available. Type ‘g’ to generate the makefile that is needed to build the `eml_linux` library. You will go back to the Terminal window. Before actually building the `eml_library` you have to modify two files

- `/usr/src/eml_linux/src/arch/RTnet/ethercat_xenomai_drv.c`

- `/usr/src/eml_linux/tests/rtnet/master_test.cpp`

In the first file `ethercat_xenomai_drv.c` comment line number 18 where `rtnet` is included with such that it becomes

```c
/* #include <rtnet.h> */
```

In the second file `master_test.cpp` edit the path of the first line such that it becomes

```c
#include <pthread.h>
```

Now the library can be build with

```
$ su <type root password>
$ cd /usr/src/eml_linux/build_rtnet
$ make
```

The last source that needs to be compiled is the configuration software for the EtherCAT system. Make sure that the path `/usr/lib` is a path of Linux. You can check this by typing the following in a terminal
\texttt{$ $ su <type root password>}

\texttt{$ echo $PATH}

If /usr/lib is not included in the PATH variable then include it by typing

\texttt{$ PATH = /usr/lib:$PATH}

Build the configuration software with

\texttt{$ su <type root password>}

\texttt{$ cd /usr/local/<matlab>/rtw/c/ectarge/ec_new}

\texttt{$ ./buildconfig}

Now you have build all the necessary files and can continue.

4.1.1 Alternative timer in the EtherCAT Target

Previously, the communication with EtherCAT modules is explained. However there might be a conflict between the real-time EtherCAT target and other functions that need timers, which canpopup if both use a timer for periodic tasks such as reading and writing to EtherCAT module. Timers can be set in various ways in Linux. In case of the real-time EtherCAT target a timer is used which is created in C with \texttt{setitimer();}. Periodically, a SIGALRM signal is given. A signal handler which reacts on the SIGALRM signal calls the specific tasks, like reading and writing I/O. However, if the other applications/functions also raise the SIGALRM flag there is conflict. This is solved by using another timer in the real-time EtherCAT target. The timer is created in C with \texttt{timer_create();}. You can manually set the periodic signal which is then set by the timer. In this case the chosen signal is SIGUSR1. This is done in the file \texttt{libtimer_posix.c} located in the folder /usr/src/Timer_posix. First, add a global variable

\begin{verbatim}
static timer_t timerid;
\end{verbatim}

This is the identifier of the timer that will be used. Then in the \texttt{rtc_timer} function add three local variables

\begin{verbatim}
struct itimerspec timer;
struct sigevent evp = \{0,SIGUSR1\};
double ns;
\end{verbatim}

Next, from starting the timer copy the following

\begin{verbatim}
/* start timer */
/* set signal handler */
action.sa_handler=signalhandler;
sigemptyset(&action.sa_mask);
action.sa_flags=0;

/* send SIGUSR1 periodically */
us=dt*1000000.0;
secs=(double)((int)(us/1000000.0));
us=us-secs*1000000.0;
ns=us*1000.0;
//printf("s = %f, us = %f, dt = %f\n", secs, us, dt);
\end{verbatim}
4.1. BUILD SOURCES

```c
status = timer_create(CLOCK_REALTIME, &evp, &timerid);
if (status == -1) {
    printf("Error timer_create\n");
    return -1;
}
/* catch SIGUSR1 */
sigaction(SIGUSR1, &action, NULL);

timer.it_value.tv_sec = timer.it_interval.tv_sec = (long int) secs;
timer.it_value.tv_nsec = timer.it_interval.tv_nsec = (long int) ns;

status = timer_settime(timerid, TIMER_ABSTIME, &timer, NULL);
/* setitimer(ITIMER_REAL, &timer, NULL); */
while (!istop) {
    rtc_usleep(1000000);
}
```

After you are done build the timer library in a shell by typing

```
$ sudo ./buildlib
```

To check whether the timer works you can build the `test_timer` application. However, the buildscript `buildtest` here needs a minor adjustment. It should be

```
gcc -fPIC -lrt -o test_timer test_timer.c -L/usr/lib -ltimer_posix
```

Run the test with

```
$ sudo ./test_timer
```

Now, the conflict is solved and real-time experiments can be performed with the EtherCAT modules with a different implemented timer.
Chapter 5

TUeDACs

In this Chapter, the installation procedure of the TUeDACs software is described [10]. The software is for use with a TUeDACs Microgiant data acquisition device, shown in Fig. 5.1.

Figure 5.1: The TUeDACs Microgiant data acquisition device.

5.1 Download software

The TUeDACs software can be downloaded using SmartSVN. If you do not have SmartSVN installed on your system, you may do so using Section 6.5. SmartSVN also requires JAVA to be installed on the system as described in Section 6.4. Start SmartSVN by typing `smartsvn` in a shell konsole. Configure the project and checkout the TUeDACs software in the following steps.

- Check out project from repository
- Detailed checkout –> manage
- Delete standard repository file
- Add a repository as follows:
5.2 Configure and build software

Before the TUeDACs software can be configured and build, Matlab has to be installed. If you don’t have Matlab installed, refer to Section 6.3. Build the TUeDACs software from a shell konsole by

```
$ su
$ cd /usr/local/TUEDACS
$ ./build-tuedacs
```

The `./build-tuedacs` script is kernel dependent, run it again if the TUeDACs software is to be used with another kernel.

Test whether the drivers are correctly loaded by connecting the Microgiant to your computer via the USB cable. If everything is correct, the Microgiant should have four lights burning (two green and two red) and the command `dmesg` in a shell konsole should display ‘USB TUeDACS Driver 3.1’ on the screen.

Next, install the Matlab TUeDACs software from a shell konsole by

```
$ su
$ cd /usr/local/TUEDACS
$ ./InstallTuedacsMatlabSoftware
```

You will be asked to enter the Matlab rooth path, for example `/usr/local/matlab2007b` or `/usr/local/matlab75`.

Now go to the folder `/usr/local/<matlab>/rtw/c/TUeDACS`, where `<matlab>` denotes the folder name of the Matlab version you have installed on your system. Change in the file `TuedacsTargetUnix.tmf` line 125 to `TUEDACS_ROOT = /usr/local/TUEDACS`.

Remove in the file `/usr/local/<matlab>/bin/gccopts.sh` on line 57 the part `-ansi` and on line 60 the part `-lstdc++`. Start Matlab and go to the `<matlab>/toolbox/TUeDACS` folder. Mex the c-files by
5.3. TEST PROGRAM

When you are asked to choose a compiler, choose the ‘gcc compiler’. You may ignore the
warnings.
Finally, append the folder <matlab>/toolbox/TUeDACS to the Matlab path to finish the
TUeDACS software installation.

5.3 Test program

To test the TUeDACs software and especially the time stamping software, a test program is
available. Start the test program as follows from a shell konsole.

```
$ cd /usr/local/TUEDACS/UtcTestProgram
$ ./UtcTestProgram
```

You can test the TUeDACs device and software by choosing in the UtcTestProgram first
the option ‘5. TUeDACs test’, then ‘1. Standard test’ and ‘6. Execute batch test’. If all is
correct, you get 0 errors. Quit the test by typing ‘q’.

If, after the installation of the TUeDACs software, the following error appears in the
UtcTestProgram

```
tdOpen: TUeDACs - error [1012] - No TUeDACS Interfaces
```

Then type the following in a shell konsole

```
$ cd /usr/local/TUEDACS/UtcDriver
$ ./makenodes
```

If you have to run the ./makenodes command every time you reboot the system, you can
append the command by adding the following lines to the file /etc/rc.local

```
/usr/local/TUEDACS/UtcDriver/makenodes
echo "Makenodes done."
```
Chapter 6

Miscellaneous

This chapter describes the installation of the ps2 and ps4 printers on floor -1 in the mechanical engineering department building, the configuration of the internet connection, how to install JAVA, SmartSVN and Matlab, a solution if Ubuntu 7.10 freezes and how to install a Syntek Semicon DC1125 driver.

6.1 Installing ps2 and ps4 printers

This section describes how to install the ps2 and ps4 printer of the mechanical engineering department.

6.1.1 In Ubuntu 7.10

To install the ps2 printer in Ubuntu, go to ‘System’, ‘Administration’, ‘Printing’. Click ‘New Printer’. Select ‘LPD/LPR Host or Printer’. The hostname of the printer is ‘wfwp2.wfw.wtb.tue.nl’ and the printername is ‘ps2’. Select ‘Printer from Database’ and click ‘Forward’. The model of the ps2 is a ‘Laserjet 9050’ and the drivers should be ‘HP Laserjet 9050 Postscript’. Click ‘Forward’. The name of the printer is ps, the description is ps2 and the location is WH floor -1. Apply the settings and your ps2 printer will be installed. The same procedure can be followed for the ps4, however the hostname is ‘wfwp4.wfw.wtb.tue.nl’, the printername is ‘ps4’, the model is ‘HP Laserjet 8000’, the driver is ‘HP Laserjet 8000 Series Postscript’, the name is ‘ps4’, the description is ‘ps4’.

6.1.2 In Mandriva 2008 ONE

In Mandriva 2008 ONE, the ps2 and ps4 printers can be installed by the following steps.

- Open the Mandriva Control Center.
- Click in the left column on ‘Hardware’.
- Click on ‘Set up the printer(s)’.
- Add a printer.
- Choose the option ‘Enter host name/IP address’. 
• Use the following settings for the ps2 printer

<table>
<thead>
<tr>
<th>Host/IP</th>
<th>wfwp2.wfw.wtb.tue.nl</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol</td>
<td>LPD</td>
</tr>
<tr>
<td>Queue</td>
<td>ps2</td>
</tr>
</tbody>
</table>

For the ps4 printer, replace ‘wfwp2.wfw.wtb.tue.nl’ by ‘wfwp4.wfw.wtb.tue.nl’ in the ‘Host/IP’ setting and replace ‘ps2’ by ‘ps4’ in the Queue setting above.

• Follow the instructions on the screen to select the correct models, HP Laserjet 9050 for the ps2 and HP Laserjet 8000 for the ps4.

### 6.2 Configure the internet connection

This section describes how to setup the internet connection, especially the proxy settings needed for the WFW network of the mechanical engineering department. The first step is to configure the settings of the proxy server. Start Mozilla from the top menubar. Go to ‘Edit’, ‘Preferences’, ‘Advanced’, ‘Network’, ‘Settings’. Select ‘Manual proxy configuration’. The proxy server settings are given in Fig. 6.1. Apply the same settings in ‘System’, ‘Preferences’, ‘Network Proxy’. The internet connection is now established. Check it by trying to connect to a website in Mozilla.

Employees of the department Mechanical Engineering can mount their network disc by typing

```bash
$ sudo mount -t smbfs -o username=<username>,workgroup=TUE //wtbfiler.campus.tue.nl/<username> /mnt/temp
```
6.3. INSTALLING MATLAB2007B FOR UNIX

Here, <username> is your username. The directory /mnt/temp is the directory where the network disc is mounted. Make sure that this location exists. If it does not exist, add it by typing

```
$ sudo mkdir /mnt/temp
```

In Mandriva this requires ‘portmapper’ to be installed, which can be done from a shell konsole by

```
$ su
$ urpmi portmapper
$ portmap restart
```

### 6.3 Installing Matlab2007b for Unix

This section describes how to install Matlab2007b for Unix from the wtbfiler/software present in the mechanical engineering department. First of all become a superuser by typing (password required)

```
$ su
```

Then mount the wtbfiler/software. In order to do that we first have to create a mount point, say /mnt/temp, type

```
$ mkdir /mnt/temp
```

A directory is created called /mnt/temp. In Ubuntu this requires the nfs-common package. In Mandriva, the portmapper package needs to be installed first, see also Section 6.2. You are now able to mount the wtbfiler/software to this mount point by typing

```
$ mount wtbfiler.wtb.tue.nl:/software /mnt/temp
```

Now make a directory in which matlab will be installed

```
$ mkdir /usr/local/matlab75
```

The steps needed for installing Matlab2007b are given in the document located in /mnt/temp/Unix-Software/matlab2007b/DVD1/install but will be explained shortly here. Copy the license file into your Matlab directory

```
$ cd /mnt/temp/Unix-Software/matlab2007b/license
$ cp license.dat /usr/local/matlab75
```

Then run the install script (make sure you are superuser)

```
$ cd /mnt/temp/Unix-Software/matlab2007b/DVD1
$ ./install
```

Choose the Matlab root directory location as /usr/local/matlab75 as in Fig. 6.2. Select the option Create symbolic links to MATLAB scripts in: /usr/local/bin as in Fig. 6.3. After the installation is completed you can start Matlab from the Terminal by typing

```
$ matlab
```
CHAPTER 6. MISCELLANEOUS

Figure 6.2: Matlab root directory.

Figure 6.3: Create symbolic link.

If you cannot type in the command window of Matlab2007b it might be the case that an old java version is installed (see Section 6.4). After installing java you will be able to type in the command window of Matlab2007b. It might be the case that the Matlab interface screen is not appearing in a normal way. Some icons and windows maybe missing. This might be caused by the fact that you have set your Visual Appearance settings, which can be found in Ubuntu in ‘System’, ‘Preferences’, ‘Appearance’, tab ‘Visual Appearance’, to normal, extra or even custom. If you set this option to ‘none’ the Matlab interface screen will appear in a correct way. If you do not prefer this you can add the following line to the file /usr/local/matlab75/bin/matlab after the command lines starting with # (so before, if "$OS"="Windows_NT"; then)

```
export AWT_TOOLKIT = MToolkit
```

In this way the Visual Appearance settings can be kept and the Matlab interface screen will appear correctly. When your license has experid, obtain a new one and copy the file license.dat to /usr/local/matlab75/etc.

6.4 Installing JAVA

You can install JAVA in Mandriva 2008 from a shell konsole by typing

```
$ urpmi java-1.6.0-sun
```

In Ubuntu, install java package using the Synaptic Package Manager. The package is called ‘sun-java6-bin’.

6.5 Installing SmartSVN

Goto the site http://www.syntevo.com/smartsvn/download.html and download the Generic (Linux) archive file smartsvn-generic-3_0_4.tar.gz to the folder /home/<username>. Java needs to be installed for the installation of SmartSVN. If not, refer to Section 6.4. Install SmartSVN by opening a shell konsole and typing

```
$ su
```
6.6. **UBUNTU 7.10, GUTSY GIBBON FREEZES EVERY 10 MINUTES FOR APPROXIMATELY 10 SEC**

```bash
$ cd /usr/local
$ tar xf /home/<username>/smartsvn-generic-3_0_4.tar.gz
$ cd smartsvn_3_0_4/bin
$ ln -s /usr/local/smartsvn_3_0_4/bin/smartsvn.sh
/usr/local/bin/smartsvn
```

Now you can start SmartSVN from a shell konsole from any location by typing `$ smartsvn`.

### 6.6 Ubuntu 7.10, Gutsy Gibbon freezes every 10 minutes for approximately 10 sec

On some computers Ubuntu 7.10 freezes for a couple of seconds and then comes back to life. If you have this problem proceed as follows. To make sure that the problem description is correct open a Terminal and type

```
$ dmesg
```

This prints the kernel ring buffer messages to the screen. If the following messages appear

```
[ 188.960000] ata1 .01: exception Emask 0x0 SAct 0x0 SErr 0x0 action 0x2 frozen
[ 188.960000] ata1 .01: cmd a0 /00:00:00:00:00:00:00/b0 tag 0 cdb 0x0 data 0
[ 188.960000] res 40/00:03:00:00:00:00:00:00/b0 Emask 0x4 (timeout)
[ 188.960000] ata1 .01: port is slow to respond, please be patient (Status 0x0d)
[ 195.964000] ata1 : port failed to respond (30 secs , Status 0x0d)
[ 218.980000] ata1 .00: configured for UDMA /100
[ 218.980000] ata1 .01: configured for UDMA /33
[ 218.980000] ata1 : EH complete
[ 219.900000] SCSI device sda : 78140160 512 - byte hdwr sectors (40008 MB)
[ 219.904000] sda : Write Protect is off
[ 219.904000] sda : Mode Sense : 00 3a 00 00
[ 219.908000] SCSI device sda : write cache : enabled , read cache : enabled , doesn’t support DPO or FUA
```

This problem is caused by wrong firmware of the optical drive of your computer. The problem and the fix for this problem is described by Gareth Fitzworthington in [5]. Go to the website of Samsung, [www.samsungodd.com](http://www.samsungodd.com). To check the Firmware version of your optical drive go to ‘Firmware’, ‘Check F/W version’. Download the ‘Check F/W version’ program. Save it to your Desktop. Install wine by typing

```
$ sudo apt-get install wine
```

Next run the WININQUIRY.exe by typing

```
$ wine WININQUIRY.exe
```

A screen should appear like the one in Fig. 6.4. In this computer was the optical drive

![Figure 6.4: WININQUIRY.exe.](image)

Figure 6.4: WININQUIRY.exe.

an TSSTcorpCD/DVD TS-L632D. The original version on this computer (Asus) was AS05.
From Fig. 6.4 it can be seen that the new firmware version is SC03. The new firmware can be loaded by following the next steps. Download the SC03 firmware from http://www.samsungodd.com/kor/Firmware/FWDownload/FWDownload.asp?FunctionValue=view&no=486&SearchWord=TS-L632D&SearchMode=TOTALSEARCH&PageNumber=1&product_code=You will find the downloadable executable at the bottom of the page called TS-L632D_SC03.exe. Save the file to your desktop. Rename the exe file to a zip file. This can be done in Ubuntu by right-clicking the file TS-L632D_SC03.exe, select Rename, and rename the file. Next, open the zip file with the Ubuntu Archive Manager. Therefore, right-click the zip file and select Open With Archive Manager. You will find two files within this zip file: 1) 200702081154530571_TS-L632D_SC03.BIN and 2) sfdnwin.exe. The BIN file is the firmware and sfdnwin.exe is the firmware flash program. Extract these two files to your Desktop. This can be done by selecting both files in the Archive Manager, right-click those files and select Extract. Extract them to your Desktop. These two files will be necessary in order to update the firmware of your optical drive. Copy these files to your Windows partition (make sure that it is a fat32 partition and not an NTFS partition, since by default NTFS partitions are not writable), or copy them to a memory stick. Reboot your system and start Microsoft Windows. Run the executable sfdnwin.exe from a command line by typing sfdnwin.exe -nocheck. The option nocheck lets you do a crossflash. Load the BIN file and flash the firmware. After the flash program has finished reboot the system and start Ubuntu 7.10 again. The problem should be solved now.

6.7 Installing Syntek Semicon DC1125 Driver

This section describes the steps to be taken in order to install the Syntek Semicon DC1125 Driver STKxx. This driver is preferred if a ASUS built-in webcam is to be used.

First of all, check if the package ctags is installed. If not, install it by typing in a Terminal

```
$ sudo apt-get install ctags
```

or install the package using the Synaptics Package manager of Ubuntu.

Next go to the website [3] and download the file stk11xx-1.2.3.tar.gz and save it to your disc. Unzip the file and go to the unzipped folder stkxxx-1.2.3. To build the documentation type in a Terminal

```
$ sudo make -f Makefile.standalone doc
```

To build the driver type

```
$ sudo make -f Makefile.standalone
```

Add the videodev module to the Linux kernel by typing

```
$ sudo modprobe videodev
```

Insert a module into the Linux kernel by typing

```
$ sudo insmod stk11xx.ko
```

To test your camera you can install the tool Camorama by typing

```
$ sudo apt-get install camorama
```
Start Camorama from Applications/Graphics. Add the color filter by right-clicking the effects tab and add filter, color correction.
Bibliography


Appendix A

Menu.lst HP desktop computer
DCT lab WH -1.13

# menu.lst - See: grub(8), info grub, update-grub(8)
#      grub-install(8), grub-floppy(8),
#     grub-md5-crypt, /usr/share/doc/grub
#   and /usr/share/doc/grub-doc/.

## default num
# Set the default entry to the entry number NUM. Numbering starts from 0, and
# the entry number 0 is the default if the command is not used.
#
# You can specify 'saved' instead of a number. In this case, the default entry
# is the entry saved with the command 'savedefault'.
default 0

## timeout sec
# Set a timeout, in SEC seconds, before automatically booting the default entry
# (normally the first entry defined).
timeout 5

# Pretty colours
color cyan/blue white/blue

## password ['--md5'] passwd
# If used in the first section of a menu file, disable all interactive editing
# control (menu entry editor and command-line) and entries protected by the
# command 'lock'
# e.g. password topsecret
#       password --md5 $1$gLhU0/$aW78kHK1QfV3P2b2znUoe/
#       password topsecret

#
# examples
# title Windows 95/98/NT/2000
# root (hd0,0)
# makeactive
# chainloader +1
#
# title Linux
# root (hd0,1)
# kernel /vmlinuz root=/dev/hda2 ro
#

# Put static boot stanzas before and/or after AUTOMAGIC KERNEL LIST

### BEGIN AUTOMAGIC KERNELS LIST
## lines between the AUTOMAGIC KERNELS LIST markers will be modified
## by the debian update-grub script except for the default options below

## DO NOT UNCOMMENT THEM, Just edit them to your needs

## Start Default Options ##
## default kernel options
## default kernel options for automagic boot options
## If you want special options for specific kernels use kopt_x_y_z
## e.g. kopt=root=/dev/hda1 ro
# kopt=root=/dev/sda3 ro ramdisk_size=100000 init=/etc/init pnpbios=off
lang=us acpi=off noapic lapic noapm xmodule=fbdev nomce quiet vga=791

## default grub root device
## e.g. groot=(hd0,0)
# groot=(hd0,2)

## should update-grub create alternative automagic boot options
## e.g. alternative=true
## alternative=false
# alternative=false

## should update-grub lock alternative automagic boot options
## e.g. lockalternative=true
## lockalternative=false
# lockalternative=false

## altoption boot targets option
## multiple altoptions lines are allowed
## e.g. altoptions=(extra menu suffix) extra boot options
## altoptions=(recovery mode) single
## End Default Options ##

title Debian GNU/Linux, kernel 2.6.12.6-fusion-0.9.1 Default
root (hd0,2)
kernel /boot/vmlinuz root=/dev/sda3 ro ramdisk_size=100000 init=/etc/init
pnpbios=off lang=us acpi=off noapic lapic noapm xmodule=fbdev nomce quiet vga=791
initrd /boot/initrd.img
savedefault
boot

title Debian GNU/Linux, kernel 2.6.12.6-fusion-0.9.1
root (hd0,2)
kernel /boot/vmlinuz-2.6.12.6-fusion-0.9.1 root=/dev/sda3 ro ramdisk_size=100000
init=/etc/init pnpbios=off lang=us xmodule=fbdev nomce quiet vga=791
initrd /boot/initrd.img-2.6.12.6-fusion-0.9.1
savedefault
boot

### END DEBIAN AUTOMAGIC KERNELS LIST

title Windows 2K/XP/2003 (sda1)
chainloader (hd0,0)+1

title Windows 2K/XP/2003 (sda2)
chainloader (hd0,1)+1

title Windows 95/98/ME (sda4)
root (hd0,3)
makeactive
chainloader +1