Debian Linux

Document Order Code: IPC/DEBIAN-50A

Release 5.0 (Lenny)

Revision	Datum	Autor	Modifikation
1.0	22.03.2010	P. Brunner	First release
1.1	17.06.2010	P. Brunner	Note on display resolution settings added

Content

1 Introduction

1.1. Abstract

This user guide describes how use the Debian GNU/Linux Distribution, Release 5.0 (code name “Lenny”) on the x86 industrial computer systems from Syslogic.
Debian is a free operating system. Its primary form, Debian GNU/Linux, is a popular and influential Linux distribution.
Debian is known for its adherence to the Unix and free software philosophies, and for its abundance of options — the current release includes over eighteen thousand software packages for eleven computer architectures, ranging from the ARM architecture commonly found in embedded systems and the IBM eServer zSeries mainframe architecture to the more common Intel x86 architecture found in modern personal computers. Debian GNU/Linux is the basis for several other distributions, including Knoppix, Linspire, MEPIS, Xandros, and the Ubuntu family.
Debian is also known for its package management system (especially APT), for its strict policies regarding its packages and the quality of its releases. These practices afford easy upgrades between releases and easy automated installation and removal of packages.
Debian uses an open development and testing process. It is developed by volunteers from around the world and supported by donations through Software in the Public Interest, Inc., a non-profit umbrella organization for free software projects.

1.2. Sources of Information

We recommend to use this application note combined with the “Debian Linux Anwenderhandbuch” [3] (only German version available). This book is much more detailed and general than this user’s guide, while this user’s guide focuses on the specific parts of our products.

1.3. Supported Hardware

Following Syslogic CPU and extension boards are supported in this release:

Platform / architecture	Syslogic product(s)
Vortex x86 based	IPC/NETIPC-4x
AMD Geode based	IPC/NETIPC-6x IPC/NETSBC-6x
Intel Atom based	IPC/NETSBC-7x

Table 1 – Supported CPU boards

Description	Syslogic product(s)
Relais board	IPC/REL12-1Ax
Digital IO board	IPC/DIO32-1Ax
CAN boards	IPC/COMPACGP-1Fx FBC/CANCORE-2x

Table 2 – Supported expansion boards

1.4. Conventions

In this document, console input is shown with a leading prompt. While both the development and the target system are based on the same the Linux distribution, most of the commands will work the same way on either system. To distinguish to which system a specific command should be applied, the shell prompt will be different and will not necessarily reflect the actual prompt you will see.

Prompt indicating input to the target system:

```
netipc:/#
```

Prompt indicating input to the host system (development PC):

```
host:/#
```

2 Getting Started

This chapter will give you a "quick start" on how to get Debian Linux running on the IPC/NETIPC or IPC/NETSBC CPU boards.

2.1. System preparation

Open your industrial PC (IPC) and plug the Compact flash card into the corresponding socket of the CPU board. Close the IPC and connect an USB keyboard, USB mouse (optional) and a VGA monitor. Power-up the IPC and press and hold the "*Delete*" key on the keyboard to enter the system BIOS screen. Select "*Basic CMOS Configuration*", where you have to enter the following settings:

Parameter	**Value**
Boot 1st:	Drive C:
IDE 0:	3 = AUTOCONFIG, LBA

Table 3 – BIOS Settings

Hit "ESC" to leave the configuration screen and select "*Write to CMOS and Exit*" (confirm with Y or Z key). Now, the system should boot into the GRUB bootloader and eventually start Debian Linux. You do not always have to change these CMOS settings when booting Linux, as these CMOS settings are stored permanently if the battery is inserted and operational. If you change the battery, you must redo the correct settings.

2.2. Booting Debian Linux

GRUB, the bootloader, will show a selection of different kernel versions / configurations. Choose the one corresponding to your hardware configuration.

2.3. Kernel arguments

Default kernel arguments which are passed from GRUB, were chosen in a conservative manner. Nevertheless, some arguments might be useful:

Kernel argument	Description
single	Boot in single user mode (a.k.a. maintenance mode)
quiet	Suppress verbose kernel output; only display error messages
vga=ask	Present a table of display modes before booting the kernel
vga=xxx	Use framebuffer console; e.g.: vga=794 for 1280x1024 @ 16 bit
8250.nr_uarts=x	Number of UARTs (if more than 4), see 9.2
console=x	Put system console on serial port, e.g.: console=ttyS0,57600,8,n,1

Table 4 – Kernel arguments

Important Note:
When booting into the linux kernel, the kernel argument "**acpi=off**" should be passed by the bootloader to the linux kernel. The bootloader GRUB is already configured to boot with this argument.

2.4. User and Password Configuration

The Syslogic Linux release is configured with the following main accounts:

Username	Password	Description
netipc	netipc	Standard / normal user account
root	netipc	Root / administrator account

Table 5 – Preconfigured user accounts

3 Overview of Installed Software

The Debian Linux image on the Compact Flash card contains a standard Debian software setup with base tools and utilities installed. The following list details a few features of the image:

- Linux kernel 2.6.26 (inclusive sources and configuration files)
- GRUB bootloader
- Binutils and GNU Compiler Tools
- Fluxbox window manager
- Iceweasel (Firefox) Web-Browser with Sun Java6 Plugin installed
- FTP/Telnet/SSH Servers installed
- NTP client
- Drivers installed for Syslogic on-board peripherals and add-on boards
- Driver installed for Hilscher cifX add-on boards

4 Syslogic Drivers and Sample Programs

Sample program sources for the CPU boards and extension boards can be found in the directory */usr/src* on your target's filesystem. At the same place, the Linux kernel source is stored. The sources are available on the product DVD-ROM as well.

Important Note:
For infos concerning the Syslogic drivers and sample programs, please read the additional documentation which is contained in the folder /DOCUMENTATION in the Virtual Host.

Please also read the chapter "Building and Installing Syslogic Modules" from the current document.

5 Virtual Host

5.1. General

The IPC/DEBIAN-50A is shipped as a self-contained and complete development environment onto a Compact Flash card. Nevertheless, it might be useful to have a host PC also running Debian Linux, e.g. for testing purposes or for compiling the kernel sources remotely.
This environment is shipped as a VMware virtual PC image.

5.2. Installation

In order to run the VMware virtual PC, you need to download and install the freely available VMware Player from the www.vmware.com website (~80MB).
Next, copy all the files from the "DEBIANHOST" directory your development PC (requires ~9 GB of free HDD space) into a directory called e.g. DEBIANHOST. Unzip the files if necessary. You also need to **re-enable write permission** on all files. Now, start the VMware Player and choose the "*Debian5.vmx*" file from the DEBIANHOST directory on your host PC to boot into the virtual Debian distribution.

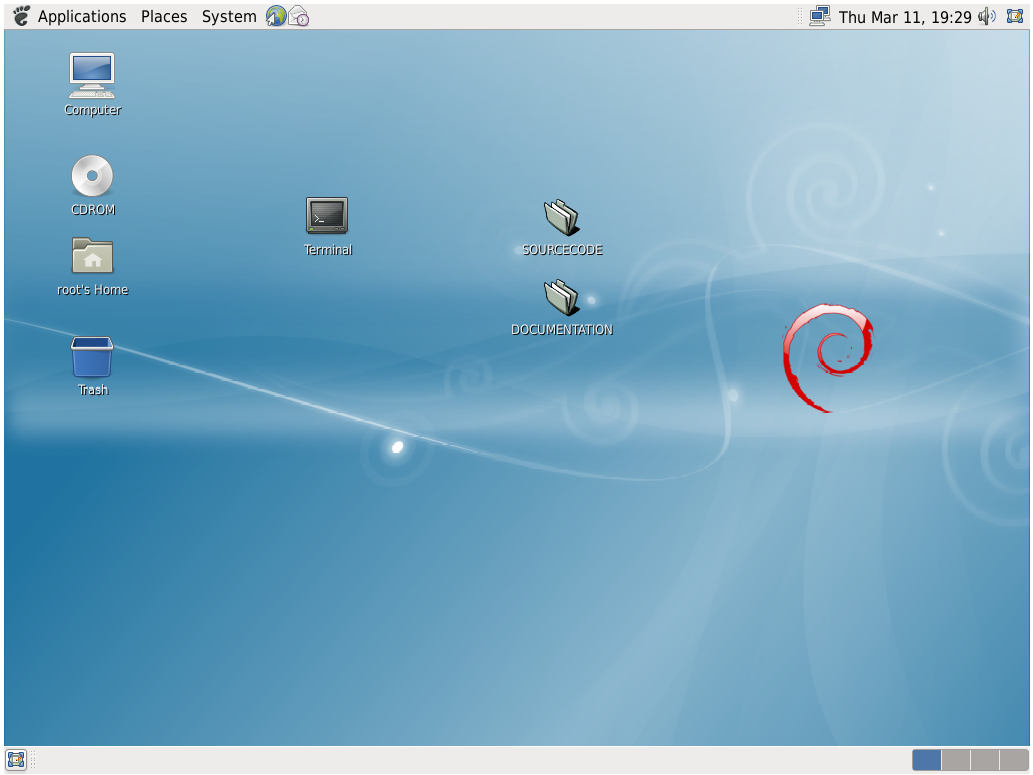

Figure 1 – VMWare virtual Debian Linux Desktop

You might be asked to create a unique identifier when booting, please choose “yes” here. You might also get some warning messages concerning some system peripherals that are missing. This is dependent on your guest computer (development PC). You can just click “ok” to ignore the messages. When the virtual machine presents the login window, login with username “**root**” and password “**root**”.

Important to Know:

You must click into the virtual PC window in order to give the virtual PC the keyboard focus. If you want to transfer the keyboard focus back to the guest PC, press **Alt-Ctrl**.

6 Debian Software Management

We just give a brief overview of the main commands. The basic (dumb) packet manager under Debian is *dpkg*, whereas *aptitude* works on top of *dpkg* and also resolves package dependencies.

6.1. Basic Paket Manager dpkg

Command for package listing :

`dpkg -l`	List installed packages
`dpkg -l "*"`	List all (available and installed packages)

The resulting package list shows the version number and the installation status of the packages:

```
rc  hotplug             0.0.20040329-26 Linux Hotplug Scripts
ii  iamerican           3.1.20.0-4.3     An American English dictionary for ispell
...
```

In the first row you see the desired **paket action**:
***u**=unknown, **i**=install, **r**=remove (delete without config), **p**=purge (delete with config)*
In the second row you see the current **paket status**:
***n**=not installed, **i**=installed, **c**=not installed anymore (but configfiles ok), **u**=unpacked, but unconfigured, **f**=failed configuration, **h**=half installed*
More package commands:

`dpkg -i packet.deb`	Install packet.deb
`dpkg -r packet.deb`	Remove packet.deb, keep configuration files
`dpkg -P packet.deb`	Remove packet.deb, also delete configuration files
`dpkg-reconfigure packet.deb`	Reconfigure packet.deb
`dpkg -S 'filename'`	Search for package that contains 'filename'
`dpkg -s packet`	Show packet status
`dpkg -L packet`	Show files of a packet

6.2. Advanced Paket Manager APTITUDE

The packet manager Aptitude knows how to resolve package dependencies and is the preferred way of installing or removing single packages or package groups.

`aptitude search keyword`	Search in all available packages for a keyword
`aptitude show package`	Show all package infos
`aptitude update`	Update the package list
`aptitude install package`	Install a package
`aptitude remove package`	Remove a package (without config. files)
`aptitude purge package`	Remove a package (with all config. files)
`aptitude clean`	Clears the package cache (/var/cache/apt/archives)
`aptitude autoclean`	Clears old packets from the package cache

6.3. Optional Software Packages

6.3.1. General remarks

There is no intention to give an overview over the several thousand packages available for Debian. Use internet search machines or Aptitude to look for a specific piece of software. Nevertheless, for some popular, optional packages the installation steps are provided below.

6.3.2. Sun Java

See the list of available packages for Sun Java:

```
netipc:/# aptitude search sun-java
```

Install e.g. Sun Java 6 Runtime Environment (approx. 120MB):

```
netipc:/# aptitude install sun-java6-jre
```

6.3.3. MySQL Server

Install latest version of MySQL Server for Debian (approx. 120MB):

```
netipc:/# aptitude install mysql-server
```

6.4. Localization

You might want to change the language of the Debian Software. This can be easily done with the Debian Package reconfiguration tool:

```
netipc:/# dpkg-reconfigure locales
```

7 Network setup

All ethernet interfaces are preconfigured automatically requesting an IP address from your router by means of DHCP. Changes to this configuration, for example using static IP addresses instead of DHCP, have to be applied to the file */etc/network/interfaces* (see example below).
To restart networking after configuration changes, type:

```
netipc:/# /etc/init.d/networking restart
```

Example: /etc/network/interfaces

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo

iface lo inet loopback


auto eth0 eth1

###################################################################
# automatic network configuration (DHCP)
# uncomment to enable, comment out to disable
###################################################################

iface eth0 inet dhcp
iface eth1 inet dhcp

###################################################################
# static network configuration
# uncomment to enable, comment out to disable
###################################################################

#iface eth0 inet static
#    address    192.168.1.40
#    netmask    255.255.255.0
#    network    192.168.1.0
#    broadcast  192.168.1.255
#    gateway    192.168.1.1

#iface eth1 inet static
#    address    192.168.2.41
#    netmask    255.255.255.0
#    network    192.168.2.0
#    broadcast  192.168.2.255
#    gateway    192.168.2.1
```

Important note:
Unconnected ethernet ports assigned to DHCP will delay the boot process considerably due to the long timeout of the DHCP client daemon (up to one minute).
It is recommended to either disable DHCP (i.e. assigning static IP addresses) on unused ports or to disable such an interface altogether to speed up boot time.

8 Window manager / desktop environment

8.1. Default window manager fluxbox

Due to performance and disk capacity considerations, the light-weight window manager fluxbox is pre-installed. Start fluxbox with:

```
netipc:/# startx
```

Important Note:
To maintain maximum compatibility across the different display solutions, the highest common resolution of 640x480 pixels is configured by default, but other resolutions (higher and lower) are defined as well. Cycle through the configured resolutions with <CTRL><ALT>+<NUMPAD_PLUS> / <NUMPAD_MINUS>.

Edit /etc/X11/xorg.conf to change the order of the resolutions, or to add/remove settings. Each bit depth has its own set of resolutions. By default, the X server will choose the first resolution defined in such a line, but will create a virtual desktop which covers the maximum defined resolution. Therefore, only a part of the desktop might be visible. Use the mouse to pan around the whole desktop area.

8.2. Optional desktop environments GNOME and KDE

In case you prefer to install a full desktop environment, this can easily be accomplished by running the commands:
For GNOME desktop environment:

```
netipc:/# aptitude install gnome
```

And for KDE respectively:

```
netipc:/# aptitude install kde
```

In order to un-install any of above listed desktop environments, just re-run aptitude with the option remove instead of install.

Important Note:
Dependencies for GNOME / KDE will download and install a considerable amount of packages (over 1 GB of data). Before confirming the download, make sure disk space is sufficient. Make room on the disk by removing unwanted packages if required.
In case there is still not enough space for the packages you would like to install, contact the manufacturer for ordering a larger compact flash card.

8.3. console-mode only

In case you would like to get rid of the window manager and X server altogether and wish to use the system in console-mode, remove the packages with:

```
netipc:/# aptitude remove fluxbox
netipc:/# aptitude remove xserver-xorg
```

9 Serial Port Configuration

9.1. Driver Configuration

The Syslogic CPU boards are equipped with a 16C550A compatible UART. The driver of the serial ports is compiled into the kernel.

Note
The standard Debian kernel only supports a maximum of 4 serial ports. Syslogic has recompiled the standard debian kernel for you to enable a total of 10 serial ports (e.g. to use the IPC/SIC4 4-port serial cards).

The program *setserial* is used to set or retrieve configuration of the serial devices. The configuration file "/etc/serial.conf" hosts the current configuration settings for all serial ports.
Syslogic has written a "Serial Driver Manual" that details the serial driver setup and configuration.

9.2. Support for more than 4 Serial Ports

If more than 4 serial ports are available in the system, the parameter *8250.nr_uarts=x* has to be appended to the kernel command line, where x states the total number of UARTs to be supported. Additionally, in */etc/modules*, the line *8250_fourport* needs to be added / uncommented and finally in */etc/serial.conf*, the ports have to be configured properly (see examples).

9.3. Console on Serial Port

It might be useful to enable a console on the serial port to test if the driver has correctly initialized all ports. This can be configured with the */etc/inittab* file:

```
netipc:/# cat /etc/inittab

…
# Example how to put a getty on a serial line (for a terminal)
#
T0:23:respawn:/sbin/getty –L ttyS0 57600 vt100
T0:23:respawn:/sbin/getty –L ttyS1 57600 vt100
…
```

With this setup you can connect to the Debian Linux system COM1 and COM2 with a terminal program from a remote host (at 57600 baud).

9.4. Console Keyboard Configuration

If you want another keymap, change directory to /usr/share/keymaps/i386/qwertz and install your favourite keymap with

```
netipc:/# install-keymap ‘keymapname‘.kmap.gz
```

10 CAN Port Configuration

10.1. Driver Configuration

Some Syslogic CPU boards are equipped with up to 2 Philips/NXP SJA1000 CAN controller which are supported by the lincan project. As of lincan version 0.3.4, the generic driver gensja1000io can be used for basic CAN mode.

Either load the kernel module manually:

```
netipc:/# insmod lincan.ko hw=gensja1000io,gensja1000io irq=6,6
io=0x7600,0x7700
```

or edit the file /etc/modules:

```
# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with “#” are ignored.
# Parameters can be specified after the module name.
…


# uncomment and adapt following line to enable CAN support
lincan hw=gensja1000io,gensja1000io irq=6,6 io=0x7600,0x7700
```

The examples show the configuration for a board which is equipped with 2 IO-mapped SJA1000 controllers. Run modinfo to see a full list of options:

```
netipc:/# modinfo lincan
```

For the PeliCAN mode, substitute the module *gensja1000io* with *syslsja1000pio* in the examples above.
For your convinience, the kernel module comes precompiled and installed. The lincan utilities like rxtx, sendburst, and others are stored under /usr/bin.
In case you would like to compile the driver yourself, the source code is contained within the virtual host under /user/src/lincan-0.3.4+syslogic, along with some hints in the file README.SYSLOGIC.

11 Hilscher cifX Cards

11.1. Driver Configuration

For cifX boards from Hilscher, the uio_netx kernel module must be loaded. Either load the kernel module manually:

```
netipc:/# modprobe uio_netx
```

Or append the module uio_netx to /etc/modules if it is not already available. By default, this line is commented out (driver disabled).

11.2. Firmware and configuration files

CifX boards require a strictly defined folder structure where the bootloader, firmware and warmstart parameters are stored.

Example folder structure:

```
netipc:/# tree /opt/cifx

/opt/cifx
|-- NXCIF50-RTE.bin
|-- cifX0.log
`-- deviceconfig
   `-- 1270100
      `-- 20028
         |-- channel0
         |   `-- cifxecm.nxf
         `-- device.conf
```

The top directory /opt/cifx contains the bootloader (e.g. NXCIF50-RTE.bin). Below that, the device configurations are stored, each in a separate folder structure, composed of the board function number (e.g. 1270100) and the serial number (e.g. 20028). In channel0, the firmware for the protocol stack is kept. The example shown above is contained on the target, under /opt/cifx, and must be individually adopted for your specific board.

11.3. Using the driver

A sample application, cifxsample, detects the presence of a card and performs initialisation. For further information regarding programming and usage of Hilscher cards, please contact our support.

```
netipc:/# cifxsample
```

12 Audio Subsystem

12.1. Driver Configuration

Certain variants of IPC/NETIPC6 and IPC/NETSBC6 are equipped with an audio option. Audio support on those boards has to be explicitly enabled by either loading the module:

```
netipc:/# modprobe snd-cs5535audio
```

Or, by removing/commenting out the entry in the modprobe blacklist:

```
netipc:/# vim /etc/modprobe.d/blacklist

# This file lists modules which will not be loaded as the result of
# alias expansion, with the purpose of preventing the hotplug subsystem
# to load them. It does not affect autoloading of modules by the kernel.
# This file is provided by the udev package.

# do not load snd-cs5535audio automatically when CS5535 is detected,
# as it may yield lots of errors when no codec is mounted on the board
# instead, comment out following line if you would like to enable the driver
# blacklist snd-cs5535audio
```

12.2. Installing and using ALSA

Further, you will need to install basic ALSA tools with:

```
netipc:/# apt-get install alsa-base alsa-utils
```

Now you may use tools like alsamixer:

```
netipc:/# alsamixer
```

13 Network Filesystem

13.1. Mounting a Windows Shared Drive

In case you need to copy a file from your Windows host system into the Debian Filesystem, you can setup your host directory as a “shared directory”. Right click on the folder you want to share and click on “Properties”, then choose the “Sharing” tab.

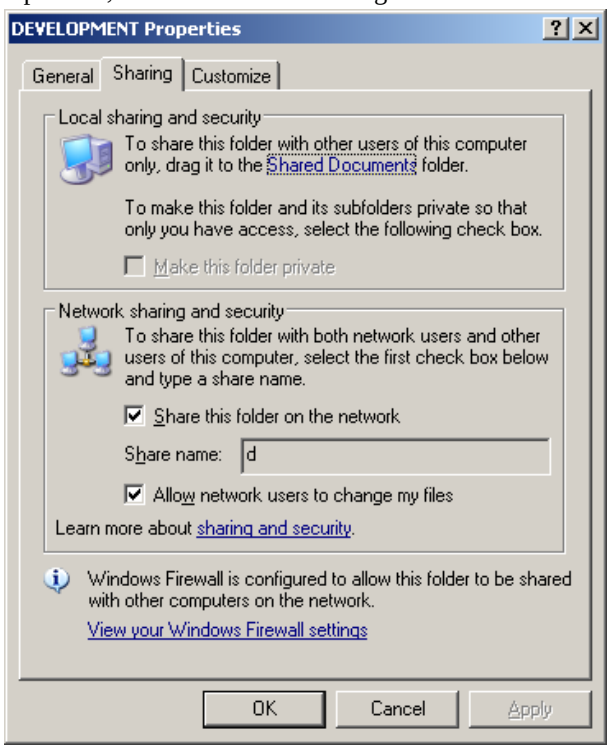

Figure 2 – Setting up a Windows Shared folder

In your VMware Debian virtual machine, you can access the shared drive from a shell and mount it into the Linux directory */mnt/winhost*.

```
netipc:/# mount -t cifs //192.168.1.153/d /mnt/winhost -o
username=Administrator,password=
```

Note: Please substitute the sample IP address from the command above with your Windows host IP address.

13.2. NFS (mounting a Linux Shared Drive)

13.2.1. Installing NFS in Debian

Making your computer an NFS server or client is very easy. A Debian NFS client needs

```
netipc:/# aptitude install nfs-common portmap
```

while a Debian NFS server needs

```
host:/# aptitude install nfs-kernel-server nfs-common portmap
```

13.2.2. NFS Server Configuration

NFS exports from a server are controlled by the file /etc/exports. Each line begins with the absolute path of a directory to be exported, followed by a space-separated list of allowed clients.

```
/etc/exports

/home 192.168.1.*(rw,no_root_squash)
```

A client can be specified either by name or IP address. Wildcards (*) are allowed in names, as are netmasks (e.g. /24) following IP addresses, but should usually be avoided for security reasons.
A client specification may be followed by a set of options, in parenthesis. It is important not to leave any space between the last client specification character and the opening parenthesis, since spaces are interpreted as client separators.
For each options specified in /etc/exports file can be check export man pages.
If you make changes to /etc/exports on a running NFS server, you can make these changes effective by issuing the command:

```
host:/# exportfs -a
```

13.2.3. NFS Client Configuration

NFS volumes can be mounted by root directly from the command line. For example

```
netipc:/# mount 192.168.1.43:/home /mnt/nfs
```

mounts the /home directory from the machine 192.168.1.43 as the directory /mnt/nfs on the client. Of course, for this to work, the directory /mnt/nfs must exist on the client and the server must have been configured to allow the client to access the volume.

13.3. How to exchange files over the Network with SCP

If you want to copy a file from one machine to another machine (target machine has IP 192.168.1.228, user root):

```
netipc:/# scp /filename root@192.168.1.228:/filename
```

14 Linux System Deployment

14.1. Backup and Deploy a System Image

If you need to store the contents of your Compact Flash for backup purposes, it is recommended to compress the image. Start the Virtual Debian host (see section 5), insert the compact flash card into the reader (assumed at /dev/sdb), and compress the flash card into a file on the host Debian linux system: (You don't need to mount the flash card reader).

```
host:/# dd if=/dev/sdb | gzip –9 > myimage.gz
```

The "-9" option is for maximum compression. If you want to rewrite the image file to the compact flash, then use the command:

```
host:/# dd if=myimage.gz | gunzip | dd of=/dev/sdb
```

Tip! If you have a lot of free CF disk space and want to improve compression ratios, you can do a little trick before compression. As the free disk space on the CF card may have randomly assigned bits, the compression result is poor. You can fill up the free disk space with zeros "0", this will lead to much better compression ratios.

```
netipc:/# dd if=/dev/zero of=myzerofile.bin
```

It might be necessary to manually end this process with "Ctrl-C" when the disk is full (use "df" to check for disk space in another console).

Note
This procedure will not work when trying to clone a system to a different compact flash card, as the real sizes of the cards might be slightly different.

14.2. Backup of the Master Boot Record

Before doing some changes to the Compact Flash card (e.g. updating the GRUB bootloader), you might want to backup the master boot record (MBR) of the card. The following command reads the first 446 bytes of a Compact Flash cards MBR attached to the USB card reader of the host system and writes them into a sample file called bootsector.bin for backup purposes:

```
netipc:/# dd if=/dev/sdb of=bootsector.bin bs=1 count=446
```

If you need to reproduce the MBR again, you can issue the following command:

```
Netipc:/# dd if=bootsector.bin of=/dev/sdb bs=1 count=446
```

Note that even if the MBR of the Compact Flash card is actually 512 bytes large, we only take the first 446 bytes because the last bytes contain the partition table that may change between different Compact Flash cards.

14.3. Deploying a System Image in archive format

Important Note:
The procedure described below relies on low-level system tools which are suitable for experienced linux users only. Any typo or mistake might ruin your system!
It is highly recommended to backup your Linux host system / virtual machine before proceeding.

Syslogic provides system images in archived format (TAR compressed with bzip2). Such a system image consists of two files:

1. base /generic image: main system image containing Debian base system and kernel
2. specific image: contains customisation for a specific platform ("delta" image with respect to the base image)

The installation guide below assumes that the compact flash card is available under the device node /dev/sdb. Make sure you are substituting all occurrences with the actual device node for the compact flash card reader on your system (*dmesg* might help you find out). Further, it is assumed that partition 1 (/dev/sdb1) becomes your system disk, while partition 2 (/dev/sdb2) is your swap partition. You will also need a mount point for the compact flash. Any empty directory serves this purpose, for example /mnt/compactflash.

14.3.1. Step 1: Preparing the compact flash card

Partition your compact flash card with the tool of your choice (e.g. fdisk or parted), according to your needs. It is recommended to create a swap partition (size depending on RAM available in your system).

```
host:/# fdisk /dev/sdb or parted /dev/sdb
```

Follow the on-screen instructions. Choose type "Linux" (type 83h) for the system partition (do not forget to set the bootable flag) and "Linux Swap" (type 82h) for the swap partition.

Note:
If using parted, you might face the error *unrecognised disk label*, which points to a corrupt bootsector. In such a case, use fdisk to partition the compact flash. If you insist on using parted, for example due to its better scriptability, use fdisk once for the partitioning, backup the bootsector with dd (see below), and restore/overwrite the bootsector on further un-initialised compact flash cards before starting parted.
Backup bootsector:

```
host:/# dd if=/dev/sdb of=/bootsector_blank.bin bs=512 count=1
```

Restore bootsector:

```
host:/# dd if=/bootsector_blank.bin of=/dev/sdb bs=512 count=1
```

14.3.2. Step 2: Create filesystem and initialise swap disk

Initialise (format) the system partition and initialise swap partition (if available):

```
host:/# mkfs.ext3 /dev/sdb1

host:/# mkswap /dev/sdb2
```

14.3.3. Step 3: Copy base image

First, mount the compact flash card:

```
host:/# mount /dev/sdb1 /mnt/compactflash
```

First, mount the compact flash card. And then, unpack the base image using cfmaster

```
host:/# mount /dev/sdb1 /mnt/compactflash

host:/# /scripts/cfmaster.sh restore base_image.tar.bz2 /mnt/compactflash
```

Alternatively, you could decompress and untar the archive to an empty folder and copy the files manually.

14.3.4. Step 4: Install bootloader

Install the bootloader with grub-install (note the double-dashes preluding the options):

```
host:/# grub-install –recheck –root-directory=/mnt/compactflash /dev/sdb
```

The grub-probe error can be ignored as long as the command shows a list of devices, e.g. (hd0) /dev/sda, at the end. The device.map written to the compact flash is invalid, but will be fixed in the next step.

14.3.5. Step 5: Copy specific image

The specific image will apply platform or device dependent customisation:

```
host:/# /scripts/cfmaster.sh restore specific_image.tar.bz2 /mnt/compactflash
```

14.3.6. Step 6: Finishing

Unmount the compact flash:

```
host:/# umount /dev/sdb1
```

Now, the compact flash can be safely removed from the compact flash card reader.

15 Remote Shell

15.1. Remote Shell with SSH

Secure Shell or SSH is a network protocol that allows data to be exchanged over a secure channel between two computers. Encryption provides confidentiality and integrity of data. SSH is typically used to log into a remote machine and execute commands.
First install the ssh package with apt-get (if it's not already installed):

```
netipc:/# apt-get install ssh
```

SSH (Secure Shell) is the preferred tool for login on your target NETIPC. Do the following for a password-free login from the host PC into the target NETIPC.
Host (PC) setup:

```
host:/# ssh-keygen -t dsa
```

In the dialog, just agree with the default settings. Copy the just created file "id_dsa.pub" to the target system (NETIPC). Maybe this file is located in a subdirectory of root.

Target (NETIPC) setup:

```
netipc:/# cat id_dsa.pub >> ~root/.ssh/authorized_keys2
```

Now you can do a

```
host:/# ssh 192.168.1.228
```

from the host PC without being prompted for login and password. You can even start graphical applications on the remote station with ssh. To do this, you simply type

```
host:/# ssh –X 192.168.1.228
host:/# ./tuxeyes
```

The "-X" stands for something like "graphical session on remote host". Tuxeyes is just an example of a program with graphical output running on the remote station.

16 Date and Time with NTP (Network Time Protocol)

16.1. NTP Setup and Configuration

NTP is a a means of transmitting time signals over a computer network. It can keep your clock up to date, even if there's no battery backing up the system clock. */etc/ntp.conf* is the configuration file for the NTP daemon. Here is a sample one:

```
netipc:/# cat /etc/ntp.conf
driftfile /var/lib/ntp/ntp.drift

server 0.de.pool.ntp.org
server 1.de.pool.ntp.org
server 2.de.pool.ntp.org
server 3.de.pool.ntp.org

# do not make NTP services available for others
restrict default nomodify nopeer
restrict 127.0.0.1
```

The NTP daemon randomly selects a time server from the pool.
Synchronize time manually:

```
netipc:/# ntpd –q -g
```

Lookup NTP time from distant server

```
netipc:/# ntpdate 0.de.pool.ntp.org
14 Nov 13:57:56 ntpdate[15558]: the NTP socket is in use, exiting
```

Eventually you might want to also update the hardware clock, else the system might have a completely wrong time after the next reboot and until it has synchronized over NTP.

```
netipc:/# hwclock–systohc
```

16.2. "set_rtc_mmss" Console Message

It might be that you see the console message "set_rtc_mmss: can't update from 54 to 5" on your system. What does that message mean?
The function set_rtc_mmss() updates minutes and seconds of the CMOS clock from system time. It does not update the hour or date to avoid problems with timezones. The message shown was added to make users and implementers aware of the problem that not all time updates will succeed.
Imagine the system time is 17:56:23 while the CMOS clock is already at 18:03:45. Updating just minutes and seconds would set the hardware clock to 18:56:23, a wrong value. The solution for this problem is to set the hardware clock with the command "**hwclock –systohc**" to the right date/time values.

16.3. Linking your own Startup Scripts into the Boot Process

Debian provides the following mechanism for customisation: Suppose a system needs to execute script foo on start-up, or on entry to a particular (System V) runlevel. Then the system administrator should:

- Enter the script foo into the directory /etc/init.d/.
- Run the Debian command update-rc.d with appropriate arguments, to set up links between the (command-line-specified) directories rc*?*.d and /etc/init.d/foo. Here, *?* is a number from 0 through 6 that corresponds to one of the System V runlevels.
- Reboot the system.

The command update-rc.d will set up links between files in the directories rc*?*.d and the script in /etc/init.d/. Each link will begin with an 'S' or a 'K', followed by a number, followed by the name of the script. Scripts beginning with 'S' in /etc/rc*N*.d/ are executed when runlevel *N* is entered. Scripts beginning with a 'K' are executed when leaving runlevel *N*.
One might, for example, cause the script foo to execute at boot-up, by putting it in /etc/init.d/ and installing the links with update-rc.d foo defaults 19. The argument defaults refers to the default runlevels, which are 2 through 5. The argument 19 ensures that foo is called before any scripts containing numbers 20 or larger.

16.4. Making a Program running faster / Priorities

Of course. If you have a program that normally takes quite a while to run and you really need it to complete faster, you can boost its job priority level so that it uses more processor cycles. This means that all other currently executing programs will run slower. But if this is what you want, the ***nice*** command will do the job for you.
Linux supports priority levels from 19 (lowest) to –20 (highest), with the default being 0. To change the priority to a negative number (increasing the priority), you must first log in as root. Then, to *start* a program called Myprog with a high priority-level of –10, use the command:

```
netipc:/# nice –n –10 myprog
```

If you wish to increase the priority of an *already running* program, use the ***renice*** command, identifying the program by its PID: (the –n argument is not used).

```
netipc:/# renice –10 506
```

17 Linux kernel

17.1. Basics

The pre-installed kernel is derived from the standard Debian kernel 2.6.26 with following changes (non-conclusive list):

- compiled with kernel pre-emption enabled
- patched with Hilscher netX UIO driver
- integration of Syslogic drivers

While in theory the kernel can be built on the target itself too, performance and/or memory limits might become a bottleneck, especially during development when re-compilation has to performed often. Therefore, the recommended method is to use the virtual host in order to perform heavy load tasks. As this is the preferred way, the steps described below will adhere to it.

Kernel sources are available under /usr/src/linux-source-2.6.26. Certain compilation scripts rather expect a path like /usr/src/linux so that creating a symbolic link is recommended:

```
host:/# ln –s /usr/src/linux-2.6.26 /usr/src/linux
```

17.2. Building your own kernel - the Debian way

Debian has its own tools for building a kernel image. Syslogic recommends you to use this tools when making changes to the kernel configuration.

17.2.1. Prerequisites

The following packages might already be installed on your system:

```
host:/# aptitude install build-essential bzip2 fakeroot initramfs-tools
kernel-package libncurses5-dev wget
```

17.2.2. Kernel Configuration

```
host:/# make menuconfig
```

A kernel configuration dialog appears on-screen. It is important to give your kernel package a unique name like shown in the following figure. Note the "**(sys06)**" entry in the "General Setup" submenu of the kernel configuration program.

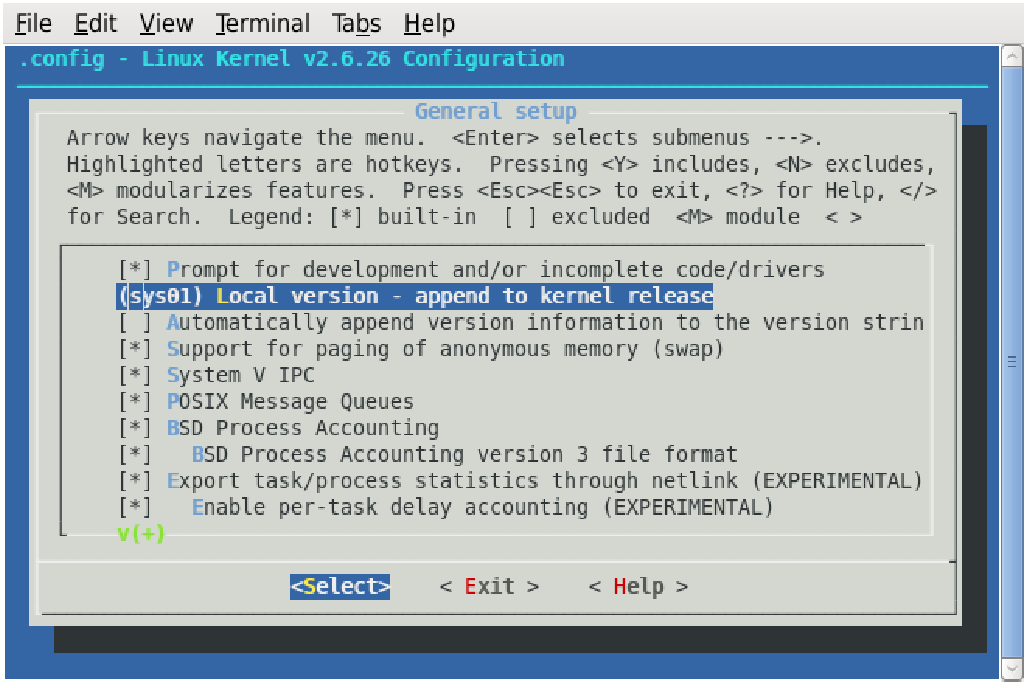

Figure 3 – Kernel Configuration Window

17.2.3. Debian Kernel Package Building

You can start to build the kernel with the following commands. This might take 2-3 hours to complete, depending on your kernel configuration.

```
host:/# make-kpkg clean
host:/# fakeroot make-kpkg --initrd kernel_image
```

This builds a debian archive (.deb package) that can be installed on the target system with dpkg package manager tool.

17.2.4. Debian Kernel Package Installation

In order to install the new kernel, copy the package to the target machine, for example by means of ssh:

```
host:/# scp linux-image-2.6.26_1.0_i386.deb root@192.168.1.37/linux-image-
2.6.26_1.0_i386.deb
```

And install the package on the target PC:

```
netipc:/# dpkg –i linux-image-2.6.26_1.0_i386.deb
```

In case the package system will ask whether you would like to stop installation because there is a kernel with the same version installed already, answer “no”. The installer will now setup the kernel, resolve the module dependencies, build the initial ramdisk and finally update the entries in the GRUB bootloader configuration.

17.3. Building your own kernel - the rough method

This section is here for documentation purposes only. You should not use the “raw” procedure when building a new kernel for your system. Instead, Syslogic recommends you to skip this section and use the instructions from the previous section “Building your own kernel - the Debian method”.
If you already have the kernel sources installed into */usr/src/linux*, so the following subsection “installing the kernel” is not necessary to execute:

Installing the kernel:

```
netipc:/# cd /usr/src
netipc:/# wget ftp://ftp.kernel.org/pub/linux/kernel/v2.6/linux-
2.6.??.tar.bz2
netipc:/# tar –xvjf linux-2.6.??.tar.bz2
netipc:/# ln –s /usr/src/linux-2.6.?? linux
```

Patching the kernel (optional):

```
netipc:/# cd/usr/src/linux
netipc:/# patch –p1 < /‘path_to_patchfile‘
```

Configuring the kernel:

```
netipc:/# make menuconfig
```

Building the kernel:

```
netipc:/# make
```

Installing the kernel modules (drivers):

```
netipc:/# make modules_install
```

The modules will be installed to /lib/modules/‘kernelversion‘.

Make an initial ramdisk image:

```
netipc:/# mkinitramfs –o /boot/initrd.img-KVERSION KVERSION
```

Where KVERSION is the kernel version string found and modified in */usr/src/linux/Makefile.*

Copy kernel files to to boot directory (installing kernel):

```
netipc:/# cd /usr/src/linux
netipc:/# cp arch/i386/boot/bzImage /boot/vmlinuz-KVERSION
netipc:/# cp System.map /boot/System.map-KVERSION
```

Edit Bootloader GRUB to boot into the new kernel image:

```
netipc:/# vi /boot/grup/menu.list
```

Edit this file according to templates.

17.3.1. Hint: Mount Kernel Sources over NFS

When compiling the kernel on the target, it is a good practice to mount the kernel with NFS over TCP/IP from a host PC into the target system PC, because the sources need about 250MB disk space. Doing so, you have also the advantage of compiling the kernel on the fast host PC.

17.3.2. Hint: Log Build Output into a File

Kernel compilation brings a lot of output to the screen. You may want so save this output for debugging purposes and also see the duration of the build process (maybe for benchmarking purposes). Then you could use the following command to build the kernel:

```
netipc:/# time make | tee kernelbuild.log
```

18 Building and Installing Syslogic Modules

In contrast to earlier releases of Debian for Syslogic products, drivers for Syslogic modules and peripherals are integrated into the kernel tree. In the linux kernel configuration dialog, they can be enabled/disabled under Device Drivers -> SYSLOGIC support. Building and installing is done implicitly when compiling the kernel.

On the next reboot, the modules will be loaded according to the modutils configuration file */etc/modules*. You need to edit that file if you want to load a new module. Also the module parameters are stored in this file.

19 Troubleshooting

19.1. Introduction

This distribution was set up and tested with care, however due to the complexity, there is still a small chance that things do not work out as intended. This guide will lead through some potential pitfalls. For other troubles, various internet resources provide advice and several useful books exist (see references). Otherwise, contact our friendly support team to get help.

19.2. General bootproblems

GRUB, the bootloader, provides an inbuilt editor which lets you change the kernel boot parameters on the fly. When the bootloader shows up the various options, just move the selection up or down with the cursor keys and hit ‘e’ to edit any option. Move the cursor to the line starting with “kernel …” and hit ‘e’ again. Edit the line according to your needs and if you are done, hit ‘b’ to boot the current selection. Note: those settings are temporarly only. To make them permanent, edit */boot/grub/menu.lst*.

19.3. How to show all kernel messages during startup

Enter the GRUB boot menu as described earlier and remove the entry *quiet* from the kernel parameters.

19.4. Console is partially visible only/distorted

Framebuffer settings are fixed for a specific display resolution. Enter the GRUB boot menu as described earlier and in the kernel parameters, change the entry *vga=xxx* to *vga=ask* to get an overview of the possible display modes. Or remove the entry *vga=xxx* altogether to get the basic 80x25 console.

19.5. Kernel hanged: waiting for root file system …

Most probably this hang is caused by a wrong value for the kernel parameter root=/dev/xxx. Depending on which driver claims control over the IDE subsystem, either the generic driver or the newer libata-pata driver will be loaded. Unfortunately, latter is using the SCSI subsystem which is using the /dev/sdx naming scheme, while the “old” generic IDE driver uses the /dev/hdx naming. During the transition between those systems ongoing in kernel development, it is difficult to predict which naming will be appropriate. The rule of thumb is: if e.g. root=/dev/hda1 does not work, try root=/dev/sda1. Or vice versa. For the default kernel installed, GRUB is configured correctly.

20 References

[1] **Syslogic IPC/NETIPC-6A User Documentation**
Document Order Code: DOC/IPC_NETIPC6-E
available from Syslogic Datentechnik AG

[2] **Linux Device Drivers (Third edition)**
from http://www.oreilly.com/catalog/linuxdrive3/

[3] **Debian GNU/Linux Anwenderhandbuch (German only):**
from http://debiananwenderhandbuch.de/

[4] **Debian GNU/Linux** (Heike Jurzik), 2. Auflage 2007
ISBN 978-9-8362-1075-1

[5] **Syslogic IPC/CRATER-10A User Documentation**
Document Order Code: DOC/IPC_CRATER-10A
available from Syslogic Datentechnik AG

21 Contact Information / Disclaimer

Our distributors and system integrators will gladly give you any information about our products and their use. If you want to contact the manufacturer directly, please send a fax or email message containing a short description of your application and your request to the following address or use one of the information or technical support request forms on our internet homepage. Syslogic is grateful for any help referring to errors or suggestions for improvements.

The following registered trademarks/licences are used:

- GPL General Public Licence
- GNU GNU is Not Unix, Free Software Foundation

The content and presentation of this document has been carefully checked. No responsibility is accepted for any errors or omissions in the documentation. Note that this application note is constantly revised and improved. The right to change this documentation at any time without notice is therefore reserved.

Syslogic Datentechnik AG
Täfernstrasse 28
CH-5405 Baden-Dättwil / Switzerland

Email: support@syslogic.ch
www: http://www.syslogic.ch
Tel: +41 56 200 90 40
Fax: +41 56 200 90 50