PC/104 & Linux-embedded HOWTO

Heimo Schön *Heimo Schön <heimo.schoen@gmx.at>*

Version 0.1 12 Oct 2003

Setup a *Linux* on a *PC/104* embedded system

1 Overview

1.1 About this HOWTO

Dieses Dokument soll einen kleinen Leitfaden bieten, wie man *Linux* auf einer embedded Hardwareplattform auf Basis des Industriestandards *PC/104* installieren kann und in Betrieb nehmen kann

Wenn Sie zu diesem HOWTO Ideen oder Korrukturen haben, dann fühlen Sie sich frei mich zu kontaktieren. Sie erreichen mich unter:

- Heimo Schön <*heimo.schoen@gmx.at*>
- *http://schoen.priv.at*

Ich wünsche Ihnen viel Erfolg bei der Installation Ihres embedded Systems und viel Spaß bei Ihren Erfahrungen mit einem skalierbaren Betriebssytem das der zeit auf ca. 30 Verschiedenen Prozessorfamilien läuft und ab der Rechenleistung eines 386 hervorragend funktioniert.

1.2 Links

Links zu verwandten Themen:

- *http://www.linux.org*
- *http://www.pc104.org*
- *http://schoen.priv.at/pc104.html*

1.3 Version History

- 0.1 12 Oct 2003 first trial

2 Einleitung

Dieses Kapitel soll eine Einleitung in das Thema dieses HOWTOs bieten und einige Hintergründe und Auslöser beschreiben.

2.1 Geschichte

Bei embedded Anwendungen habe ich in Projekten gerne Prozessoren wie z.B. 8051 oder PIC-Controller eingesetzt. embedded Projekte waren bis vor einigen Jahren damit sehr schön zu realisieren.

Doch die Zeit ist nicht stehen geblieben. Heute wollen Anwender z.B. ihren Videorecorder nicht mehr mit zig undurchschaubaren Knöpfchen bedienen, sondern möchten z.B. via OSD (on-screen-display) menügeführt vom Sofa aus bedienen, oder möchten z.B. die Digitalkamera mit USB Schnittstelle auslesen, oder wollen in den SAT-Receiver mit telnet einloggen können um SW-Updates mit ftp einspielen zu können, uvm.

Genau an diesem Punkt war ich bei einem Projekt im Jahr 2000 und stand vor der Frage ob meine bisherigen Lösungsansätze noch zeitgemäß sind. Nach einiger Suche bin auf die HW-Plattform entsprechend dem Industriestandard *PC/104* gestoßen. Viele, wenn nicht sogar hunderte Hersteller, bieten Prozessorleiterplatten, In-/Outputleiterplatten, Netzwerkkarten, Graphikkarten, uvm. an. Durch diese große Auswahl bestand die Möglichkeit, ohne Leiterplatten entwicklung, auf das Wesentliche zu konzentrieren, nämlich auf die Entwicklung der Applikation und die Lösung der Aufgaben die meine Kunden forderten.

Nach einigen Jahren der Anwendung von PC/104 und Linux, und vielen Versuchen mit verschiedensten Distributionen, hat sich ein SStandardwegäusgebildet, mit dem ich üblicherweise an die Lösung der Aufgaben herangehe. Diesen Weg möchte ich in diesem HOWTO beschreiben, in der Hoffnung das es jemandem weiterhilft ;-)

3 Ein Kernel

3.1 Ansatz

Einleitend: ich bin ein fauler Mensch!

Woher bekommt ein fauler Mensch nun einen Kernel? Man installiert auf einer beliebigen Maschine (möglichst mit der gleichen Prozessorfamilie wie am vorgesehenen Embedded System) eine Standard-Distribution (Suse, RedHat, o.ä.) und kopiert das File /proc/config.gz nach /usr/src/linux.

Danach entpackt man dieses File mit gunzip config.gz und ändert den Namen mit mv config .config und hat nun die Konfiguration des Kernels mit dem man gerade arbeitet.

Sie können mit dieser Methode gefahrlos ßpielenünd verschiedene Distributionen testen oder mit verschiedenen Kernel die angeboten werden experimentieren, denn so lange Sie nich make bzdisk eintippen, ist Ihr lauffähiges System nicht gefährdet.

3.2 Kernel anpassen

Nun haben wir uns eine Kernelkonfiguration beschafft, und müssen diese an unser embedded System anpassen.

Die Änderung der Kerneleinstellung im File /usr/src/linux/.config kann sehr einfach von der graphischen Oberfläche mit dem Befehl make xconfig durchgeführt werden (rootrechte erforderlich oder Schreibrechte des Files .config auf für others setzen)

Ich liste hier die Änderungen nur tabelarisch auf um Ihnen ein Gefühl zu geben, was wichtig sein könnte:

- unnötige Dinge abschalten (bitte mit Ihrer Hardware vergleich), wie z.B.
  - SCSI
  - RAID
  - EXT3, reiserfs, usw.
  - Bluetooth, USB, usw.
  - Sound

  - Multimedia devices
  - uvm.
- watchdog driver aktivieren
- Netzwerkkarten alle als module markierten Treiber abwählen (auf N setzen)
- hingegen den Netzwerkchip der auf Ihrem embedded-system ist, in den Kernel kompilieren (auf Y setzen)
-
- uvm.

Beim ersten Versuch sollten Sie sich mindestens 2 Stunden Zeit nehmen und ggf. auch die Hilfe lesen, die zu den verschiedenen Optionen angeboten wird.

Ein bei mir in einem vergangenen Projekt eingesetztes Konfigfile sieht wie folgt aus:

```
#
# Automatically generated make config: don't edit
#
CONFIG_X86=y
# CONFIG_SBUS is not set
CONFIG_UID16=y

#
# Code maturity level options
#
CONFIG_EXPERIMENTAL=y

#
# Loadable module support
#
CONFIG_MODULES=y
# CONFIG_MODVERSIONS is not set
CONFIG_KMOD=y

#
# Processor type and features
#
# CONFIG_M386 is not set
CONFIG_M486=y
# CONFIG_M586 is not set
# CONFIG_M586TSC is not set
# CONFIG_M586MMX is not set
# CONFIG_M686 is not set
# CONFIG_MPENTIUMIII is not set
# CONFIG_MPENTIUM4 is not set
# CONFIG_MK6 is not set
# CONFIG_MK7 is not set
# CONFIG_MK8 is not set
# CONFIG_MELAN is not set
# CONFIG_MCRUSOE is not set
# CONFIG_MWINCHIPC6 is not set
# CONFIG_MWINCHIP2 is not set
```

```
# CONFIG_MWINCHIP3D is not set
# CONFIG_MCYRIXIII is not set
CONFIG_X86_WP_WORKS_OK=y
CONFIG_X86_INVLPG=y
CONFIG_X86_CMPXCHG=y
CONFIG_X86_XADD=y
CONFIG_X86_BSWAP=y
CONFIG_X86_POPAD_OK=y
CONFIG_X86_L1_CACHE_SHIFT=4
CONFIG_X86_USE_STRING_486=y
CONFIG_X86_ALIGNMENT_16=y
CONFIG_X86_PPRO_FENCE=y
# CONFIG_X86_F00F_WORKS_OK is not set
CONFIG_X86_MCE=y

#
# CPU Frequency scaling
#
CONFIG_CPU_FREQ=y
CONFIG_CPU_FREQ_TABLE=m
CONFIG_CPU_FREQ_PROC_INTF=y
CONFIG_CPU_FREQ_GOV_USERSPACE=m
CONFIG_CPU_FREQ_24_API=y
CONFIG_X86_POWERNOW_K6=m
CONFIG_X86_POWERNOW_K7=m
CONFIG_X86_LONGHAUL=m
CONFIG_X86_SPEEDSTEP=m
CONFIG_X86_P4_CLOCKMOD=m
CONFIG_X86_LONGRUN=m
CONFIG_X86_GX_SUSPMOD=m
# CONFIG_TOSHIBA is not set
# CONFIG_OMNIBOOK is not set
# CONFIG_I8K is not set
# CONFIG_THINKPAD is not set
# CONFIG_MICROCODE is not set
# CONFIG_X86_MSR is not set
# CONFIG_X86_CPUID is not set
# CONFIG_NOHIGHMEM is not set
CONFIG_HIGHMEM4G=y
# CONFIG_HIGHMEM64G is not set
CONFIG_HIGHMEM=y
CONFIG_FORCE_MAX_ZONEORDER=11
CONFIG_1GB=y
# CONFIG_2GB is not set
# CONFIG_3GB is not set
# CONFIG_05GB is not set
CONFIG_HIGHIO=y
# CONFIG_MATH_EMULATION is not set
CONFIG_MTRR=y
# CONFIG_SMP is not set
CONFIG_X86_UP_APIC=y
CONFIG_X86_UP_IOAPIC=y
CONFIG_X86_LOCAL_APIC=y
CONFIG_X86_IO_APIC=y
# CONFIG_X86_TSC_DISABLE is not set
```

```
CONFIG_MXT=y

#
# General setup
#
CONFIG_NET=y
CONFIG_EVLOG=y
CONFIG_EVLOG_BUFSIZE=128
CONFIG_EVLOG_FWPRINTK=y
# CONFIG_EVLOG_PRINTKWLOC is not set
CONFIG_PCI=y
# CONFIG_PCI_GOBIOS is not set
# CONFIG_PCI_GODIRECT is not set
CONFIG_PCI_GOANY=y
CONFIG_PCI_BIOS=y
CONFIG_PCI_DIRECT=y
CONFIG_ISA=y
CONFIG_PCI_NAMES=y
CONFIG_EISA=y
# CONFIG_MCA is not set
CONFIG_HOTPLUG=y

#
# PCMCIA/CardBus support
#
CONFIG_PCMCIA=m
CONFIG_CARDBUS=y
CONFIG_TCIC=y
CONFIG_I82092=y
CONFIG_I82365=y

#
# PCI Hotplug Support
#
CONFIG_HOTPLUG_PCI=m
CONFIG_HOTPLUG_PCI_COMPAQ=m
CONFIG_HOTPLUG_PCI_COMPAQ_NVRAM=y
CONFIG_HOTPLUG_PCI_IBM=m
CONFIG_HOTPLUG_PCI_AMD=m
CONFIG_SYSVIPC=y
CONFIG_BSD_PROCESS_ACCT=y
CONFIG_SYSCTL=y
CONFIG_MAX_USER_RT_PRIO=100
CONFIG_MAX_RT_PRIO=0
CONFIG_KCORE_ELF=y
# CONFIG_KCORE_AOUT is not set
# CONFIG_BINFMT_AOUT is not set
CONFIG_BINFMT_ELF=y
# CONFIG_BINFMT_MISC is not set
CONFIG_PM=y
# CONFIG_APM is not set

#
# ACPI Support
#
```

```
# CONFIG_ACPI is not set

#
# Binary emulation of other systems
#
# CONFIG_ABI is not set

#
# Support for foreign binary formats
#

#
# Memory Technology Devices (MTD)
#
# CONFIG_MTD is not set

#
# Parallel port support
#
CONFIG_PARPORT=y
CONFIG_PARPORT_PC=y
CONFIG_PARPORT_PC_CML1=y
# CONFIG_PARPORT_SERIAL is not set
CONFIG_PARPORT_PC_FIFO=y
CONFIG_PARPORT_PC_SUPERIO=y
# CONFIG_PARPORT_AMIGA is not set
# CONFIG_PARPORT_MFC3 is not set
# CONFIG_PARPORT_ATARI is not set
# CONFIG_PARPORT_GSC is not set
# CONFIG_PARPORT_SUNBPP is not set
# CONFIG_PARPORT_OTHER is not set
CONFIG_PARPORT_1284=y

#
# Plug and Play configuration
#
CONFIG_PNP=y
CONFIG_ISAPNP=m

#
# Block devices
#
CONFIG_BLK_DEV_FD=y
# CONFIG_BLK_DEV_XD is not set
# CONFIG_PARIDE is not set
# CONFIG_BLK_CPQ_DA is not set
# CONFIG_BLK_CPQ_CISS_DA is not set
# CONFIG_BLK_DEV_DAC960 is not set
# CONFIG_BLK_DEV_UMEM is not set
CONFIG_BLK_DEV_LOOP=y
# CONFIG_CIPHER_TWOFISH is not set
# CONFIG_BLK_DEV_NBD is not set
CONFIG_BLK_DEV_RAM=y
CONFIG_BLK_DEV_RAM_SIZE=64000
CONFIG_BLK_DEV_INITRD=y
```

```
CONFIG_BLK_STATS=y

#
# Enterprise Volume Management System
#
# CONFIG_EVMS is not set

#
# Multi-device support (RAID and LVM)
#
# CONFIG_MD is not set

#
# Networking options
#
# CONFIG_PACKET is not set
# CONFIG_NETLINK_DEV is not set
CONFIG_NETFILTER=y
# CONFIG_NETFILTER_DEBUG is not set
CONFIG_FILTER=y
CONFIG_UNIX=y
CONFIG_INET=y
CONFIG_TUX=m
CONFIG_TUX_EXTCGI=y
CONFIG_TUX_EXTENDED_LOG=y
# CONFIG_TUX_DEBUG is not set
CONFIG_IP_MULTICAST=y
CONFIG_IP_ADVANCED_ROUTER=y
CONFIG_IP_MULTIPLE_TABLES=y
CONFIG_IP_ROUTE_FWMARK=y
CONFIG_IP_ROUTE_NAT=y
CONFIG_IP_ROUTE_MULTIPATH=y
CONFIG_IP_ROUTE_TOS=y
CONFIG_IP_ROUTE_VERBOSE=y
CONFIG_IP_ROUTE_LARGE_TABLES=y
CONFIG_IP_PNP=y
CONFIG_IP_PNP_DHCP=y
CONFIG_IP_PNP_BOOTP=y
CONFIG_IP_PNP_RARP=y
CONFIG_NET_IPIP=m
CONFIG_NET_IPGRE=m
CONFIG_NET_IPGRE_BROADCAST=y
CONFIG_IP_MROUTE=y
CONFIG_IP_PIMSM_V1=y
CONFIG_IP_PIMSM_V2=y
# CONFIG_ARPD is not set
CONFIG_INET_ECN=y
CONFIG_SYN_COOKIES=y

#
#   IP: Netfilter Configuration
#
CONFIG_IP_NF_CONNTRACK=m
CONFIG_IP_NF_FTP=m
CONFIG_IP_NF_IRC=m
```

```
CONFIG_IP_NF_QUEUE=m
CONFIG_IP_NF_IPTABLES=m
CONFIG_IP_NF_MATCH_LIMIT=m
CONFIG_IP_NF_MATCH_MAC=m
CONFIG_IP_NF_MATCH_PKTTYPE=m
CONFIG_IP_NF_MATCH_MARK=m
CONFIG_IP_NF_MATCH_MULTIPORT=m
CONFIG_IP_NF_MATCH_TOS=m
CONFIG_IP_NF_MATCH_PSD=m
CONFIG_IP_NF_MATCH_ECN=m
CONFIG_IP_NF_MATCH_DSCP=m
CONFIG_IP_NF_MATCH_AH_ESP=m
CONFIG_IP_NF_MATCH_LENGTH=m
CONFIG_IP_NF_MATCH_TTL=m
CONFIG_IP_NF_MATCH_TCPMSS=m
CONFIG_IP_NF_MATCH_HELPER=m
CONFIG_IP_NF_MATCH_STATE=m
CONFIG_IP_NF_MATCH_CONNTRACK=m
CONFIG_IP_NF_MATCH_IPLIMIT=m
CONFIG_IP_NF_MATCH_UNCLEAN=m
CONFIG_IP_NF_MATCH_STRING=m
CONFIG_IP_NF_MATCH_OWNER=m
CONFIG_IP_NF_FILTER=m
CONFIG_IP_NF_TARGET_REJECT=m
CONFIG_IP_NF_TARGET_MIRROR=m
CONFIG_IP_NF_NAT=m
CONFIG_IP_NF_NAT_NEEDED=y
CONFIG_IP_NF_TARGET_MASQUERADE=m
CONFIG_IP_NF_TARGET_REDIRECT=m
# CONFIG_IP_NF_NAT_LOCAL is not set
CONFIG_IP_NF_NAT_SNMP_BASIC=m
CONFIG_IP_NF_NAT_IRC=m
CONFIG_IP_NF_NAT_FTP=m
CONFIG_IP_NF_MANGLE=m
CONFIG_IP_NF_TARGET_TOS=m
CONFIG_IP_NF_TARGET_ECN=m
CONFIG_IP_NF_TARGET_DSCP=m
CONFIG_IP_NF_TARGET_MARK=m
CONFIG_IP_NF_TARGET_LOG=m
CONFIG_IP_NF_TARGET_ULOG=m
CONFIG_IP_NF_TARGET_TCPMSS=m
CONFIG_IP_NF_ARPTABLES=m
CONFIG_IP_NF_ARPFILTER=m
CONFIG_IP_NF_COMPAT_IPCHAINS=m
CONFIG_IP_NF_NAT_NEEDED=y
CONFIG_IP_NF_COMPAT_IPFWADM=m
CONFIG_IP_NF_NAT_NEEDED=y

#
#   IP: Virtual Server Configuration
#
CONFIG_IP_VS=m
# CONFIG_IP_VS_DEBUG is not set
CONFIG_IP_VS_TAB_BITS=12
```

```
#
# IPVS scheduler
#
CONFIG_IP_VS_RR=m
CONFIG_IP_VS_WRR=m
CONFIG_IP_VS_LC=m
CONFIG_IP_VS_WLC=m
CONFIG_IP_VS_LBLC=m
CONFIG_IP_VS_LBLCR=m
CONFIG_IP_VS_DH=m
CONFIG_IP_VS_SH=m

#
# IPVS application helper
#
CONFIG_IP_VS_FTP=m
# CONFIG_IPV6 is not set
# CONFIG_SHARED_IPV6_CARDS is not set
# CONFIG_KHTTPD is not set
CONFIG_ATM=y
CONFIG_ATM_CLIP=y
CONFIG_ATM_CLIP_NO_ICMP=y
# CONFIG_ATM_LANE is not set
# CONFIG_ATM_BR2684 is not set
# CONFIG_VLAN_8021Q is not set

#
#
#
# CONFIG_IPX is not set
# CONFIG_ATALK is not set

#
# Appletalk devices
#
# CONFIG_DECNET is not set
# CONFIG_BRIDGE is not set
# CONFIG_X25 is not set
# CONFIG_LAPB is not set
# CONFIG_LLC is not set
# CONFIG_NET_DIVERT is not set
# CONFIG_ECONET is not set
# CONFIG_WAN_ROUTER is not set
# CONFIG_NET_FASTROUTE is not set
# CONFIG_NET_HW_FLOWCONTROL is not set

#
# QoS and/or fair queueing
#
# CONFIG_NET_SCHED is not set

#
# Network testing
#
# CONFIG_NET_PKTGEN is not set
```

```
#
# Telephony Support
#
# CONFIG_PHONE is not set

#
# ATA/IDE/MFM/RLL support
#
CONFIG_IDE=y

#
# IDE, ATA and ATAPI Block devices
#
CONFIG_BLK_DEV_IDE=y

#
# Please see Documentation/ide.txt for help/info on IDE drives
#
# CONFIG_BLK_DEV_HD_IDE is not set
# CONFIG_BLK_DEV_HD is not set
CONFIG_BLK_DEV_IDEDISK=y
CONFIG_IDEDISK_MULTI_MODE=y
CONFIG_IDEDISK_STROKE=y
# CONFIG_BLK_DEV_IDECS is not set
CONFIG_BLK_DEV_IDECD=y
# CONFIG_BLK_DEV_IDETAPE is not set
# CONFIG_BLK_DEV_IDEFLOPPY is not set
# CONFIG_IDE_TASK_IOCTL is not set

#
# IDE chipset support/bugfixes
#
CONFIG_BLK_DEV_CMD640=y
CONFIG_BLK_DEV_CMD640_ENHANCED=y
CONFIG_BLK_DEV_IDEPCI=y
CONFIG_BLK_DEV_GENERIC=y
CONFIG_IDEPCI_SHARE_IRQ=y
CONFIG_BLK_DEV_IDEDMA_PCI=y
CONFIG_BLK_DEV_OFFBOARD=y
# CONFIG_BLK_DEV_IDEDMA_FORCED is not set
CONFIG_IDEDMA_PCI_AUTO=y
CONFIG_IDEDMA_ONLYDISK=y
CONFIG_BLK_DEV_IDEDMA=y
# CONFIG_IDEDMA_PCI_WIP is not set
CONFIG_BLK_DEV_ADMA=y
CONFIG_BLK_DEV_AEC62XX=y
CONFIG_BLK_DEV_ALI15X3=y
# CONFIG_WDC_ALI15X3 is not set
CONFIG_BLK_DEV_AMD74XX=y
CONFIG_BLK_DEV_CMD64X=y
CONFIG_BLK_DEV_TRIFLEX=y
CONFIG_BLK_DEV_CY82C693=y
CONFIG_BLK_DEV_CS5530=y
CONFIG_BLK_DEV_HPT34X=y
```

```
CONFIG_BLK_DEV_HPT366=y
CONFIG_BLK_DEV_PIIX=y
CONFIG_BLK_DEV_NS87415=y
# CONFIG_BLK_DEV_OPTI621 is not set
CONFIG_BLK_DEV_PDC202XX_OLD=y
CONFIG_PDC202XX_BURST=y
CONFIG_BLK_DEV_PDC202XX_NEW=y
CONFIG_PDC202XX_FORCE=y
CONFIG_BLK_DEV_RZ1000=y
CONFIG_BLK_DEV_SC1200=y
CONFIG_BLK_DEV_SVWKS=y
CONFIG_BLK_DEV_SIIMAGE=y
CONFIG_BLK_DEV_SIS5513=y
CONFIG_BLK_DEV_SLC90E66=y
# CONFIG_BLK_DEV_TRM290 is not set
CONFIG_BLK_DEV_VIA82CXXX=y
CONFIG_BLK_DEV_CENATEK=y
CONFIG_IDE_CHIPSETS=y

#
# Note: most of these also require special kernel boot parameters
#
CONFIG_BLK_DEV_4DRIVES=y
CONFIG_BLK_DEV_ALI14XX=y
CONFIG_BLK_DEV_DTC2278=y
CONFIG_BLK_DEV_HT6560B=y
# CONFIG_BLK_DEV_PDC4030 is not set
CONFIG_BLK_DEV_QD65XX=y
CONFIG_BLK_DEV_UMC8672=y
CONFIG_IDEDMA_AUTO=y
# CONFIG_IDEDMA_IVB is not set
# CONFIG_DMA_NONPCI is not set
CONFIG_BLK_DEV_PDC202XX=y
CONFIG_BLK_DEV_IDE_MODES=y
# CONFIG_BLK_DEV_ATARAID is not set

#
# SCSI support
#
# CONFIG_SCSI is not set

#
# Fusion MPT device support
#
# CONFIG_FUSION_BOOT is not set
# CONFIG_FUSION_ISENSE is not set
# CONFIG_FUSION_CTL is not set
# CONFIG_FUSION_LAN is not set

#
# IEEE 1394 (FireWire) support (EXPERIMENTAL)
#
# CONFIG_IEEE1394 is not set

#
```

```
# I2O device support
#
# CONFIG_I2O is not set

#
# Network device support
#
CONFIG_NETDEVICES=y

#
# ARCnet devices
#
CONFIG_ARCNET=m
CONFIG_ARCNET_1201=m
CONFIG_ARCNET_1051=m
CONFIG_ARCNET_RAW=m
CONFIG_ARCNET_COM90xx=m
CONFIG_ARCNET_COM90xxIO=m
CONFIG_ARCNET_RIM_I=m
CONFIG_ARCNET_COM20020=m
CONFIG_ARCNET_COM20020_ISA=m
CONFIG_ARCNET_COM20020_PCI=m
# CONFIG_DUMMY is not set
# CONFIG_BONDING is not set
# CONFIG_EQUALIZER is not set
# CONFIG_TUN is not set
# CONFIG_ETHERTAP is not set
# CONFIG_NET_SB1000 is not set

#
# Ethernet (10 or 100Mbit)
#
CONFIG_NET_ETHERNET=y
# CONFIG_HAPPYMEAL is not set
# CONFIG_SUNGEM is not set
# CONFIG_NET_VENDOR_3COM is not set
# CONFIG_LANCE is not set
# CONFIG_NET_VENDOR_SMC is not set
# CONFIG_NET_VENDOR_RACAL is not set
# CONFIG_AT1700 is not set
# CONFIG_DEPCA is not set
# CONFIG_HP100 is not set
CONFIG_NET_ISA=y
# CONFIG_E2100 is not set
CONFIG_EWRK3=y
CONFIG_EEXPRESS=y
CONFIG_EEXPRESS_PRO=y
# CONFIG_HPLAN_PLUS is not set
# CONFIG_HPLAN is not set
# CONFIG_LP486E is not set
# CONFIG_ETH16I is not set
# CONFIG_NE2000 is not set
CONFIG_NET_PCI=y
# CONFIG_PCNET32 is not set
# CONFIG_PCNET32_OLD is not set
```

```
# CONFIG_AMD8111_ETH is not set
# CONFIG_ADAPTEC_STARFIRE is not set
# CONFIG_AC3200 is not set
# CONFIG_APRICOT is not set
# CONFIG_NET_BCM4400 is not set
# CONFIG_CS89x0 is not set
# CONFIG_TULIP is not set
# CONFIG_DE4X5 is not set
# CONFIG_DGRS is not set
# CONFIG_DM9102 is not set
# CONFIG_EEPRO100 is not set
CONFIG_E100=y
# CONFIG_LNE390 is not set
# CONFIG_FEALNX is not set
# CONFIG_NATSEMI is not set
# CONFIG_NETGEAR_GA621 is not set
# CONFIG_NETGEAR_GA622 is not set
# CONFIG_NE2K_PCI is not set
# CONFIG_NE3210 is not set
# CONFIG_ES3210 is not set
# CONFIG_8139CP is not set
# CONFIG_8139TOO is not set
# CONFIG_SIS900 is not set
# CONFIG_EPIC100 is not set
# CONFIG_SUNDANCE is not set
# CONFIG_TLAN is not set
# CONFIG_TC35815 is not set
# CONFIG_VIA_RHINE is not set
# CONFIG_WINBOND_840 is not set
# CONFIG_NET_POCKET is not set

#
# Ethernet (1000 Mbit)
#
# CONFIG_ACENIC is not set
# CONFIG_DL2K is not set
# CONFIG_E1000 is not set
# CONFIG_NS83820 is not set
# CONFIG_HAMACHI is not set
# CONFIG_YELLOWFIN is not set
# CONFIG_R8169 is not set
# CONFIG_SK98LIN is not set
# CONFIG_SK9DLIN is not set
# CONFIG_TIGON3 is not set
# CONFIG_NET_BROADCOM is not set
# CONFIG_FDDI is not set
# CONFIG_NETCONSOLE is not set
# CONFIG_HIPPI is not set
CONFIG_PLIP=m
CONFIG_PPP=y
# CONFIG_PPP_MULTILINK is not set
CONFIG_PPP_FILTER=y
CONFIG_PPP_ASYNC=y
# CONFIG_PPP_SYNC_TTY is not set
CONFIG_PPP_DEFLATE=y
```

```
# CONFIG_PPP_BSDCOMP is not set
CONFIG_PPPOE=y
# CONFIG_PPPOATM is not set
CONFIG_SLIP=y
CONFIG_SLIP_COMPRESSED=y
CONFIG_SLIP_SMART=y
CONFIG_SLIP_MODE_SLIP6=y

#
# Wireless LAN (non-hamradio)
#
# CONFIG_NET_RADIO is not set

#
# Token Ring devices
#
# CONFIG_TR is not set
# CONFIG_NET_FC is not set
# CONFIG_RCPCI is not set
# CONFIG_SHAPER is not set

#
# Wan interfaces
#
# CONFIG_WAN is not set

#
# PCMCIA network device support
#
# CONFIG_NET_PCMCIA is not set

#
# ATM drivers
#
# CONFIG_ATM_TCP is not set
# CONFIG_ATM_LANAI is not set
# CONFIG_ATM_ENI is not set
# CONFIG_ATM_FIRESTREAM is not set
# CONFIG_ATM_ZATM is not set
# CONFIG_ATM_NICSTAR is not set
# CONFIG_ATM_IDT77252 is not set
# CONFIG_ATM_AMBASSADOR is not set
# CONFIG_ATM_HORIZON is not set
# CONFIG_ATM_IA is not set
# CONFIG_ATM_FORE200E_MAYBE is not set

#
# Amateur Radio support
#
# CONFIG_HAMRADIO is not set

#
# IrDA (infrared) support
#
# CONFIG_IRDA is not set
```

```
#
# ISDN subsystem
#
# CONFIG_ISDN is not set

#
# Old CD-ROM drivers (not SCSI, not IDE)
#
# CONFIG_CD_NO_IDESCSI is not set

#
# Input core support
#
# CONFIG_INPUT is not set
CONFIG_INPUT_MOUSEDEV_SCREEN_X=1024
CONFIG_INPUT_MOUSEDEV_SCREEN_Y=768

#
# Character devices
#
CONFIG_VT=y
CONFIG_ECC=m
CONFIG_VT_CONSOLE=y
CONFIG_SERIAL=y
CONFIG_SERIAL_CONSOLE=y
CONFIG_SERIAL_EXTENDED=y
CONFIG_SERIAL_MANY_PORTS=y
CONFIG_SERIAL_SHARE_IRQ=y
# CONFIG_SERIAL_DETECT_IRQ is not set
CONFIG_SERIAL_MULTIPORT=y
CONFIG_HUB6=y
CONFIG_SERIAL_NONSTANDARD=y
CONFIG_COMPUTONE=m
CONFIG_ROCKETPORT=m
CONFIG_CYCLADES=m
# CONFIG_CYZ_INTR is not set
CONFIG_DIGIEPCA=m
CONFIG_ESPSERIAL=m
CONFIG_MOXA_INTELLIO=m
CONFIG_MOXA_SMARTIO=m
CONFIG_ISI=m
CONFIG_SYNCLINK=m
CONFIG_SYNCLINKMP=m
CONFIG_N_HDLC=m
CONFIG_RISCOM8=m
# CONFIG_SPECIALIX is not set
# CONFIG_SX is not set
# CONFIG_RIO is not set
# CONFIG_STALDRV is not set
CONFIG_UNIX98_PTYS=y
CONFIG_UNIX98_PTY_COUNT=256
CONFIG_PRINTER=y
# CONFIG_LP_CONSOLE is not set
# CONFIG_PPDEV is not set
```

```
# CONFIG_TIPAR is not set
# CONFIG_HVC_CONSOLE is not set

#
# I2C support
#
# CONFIG_I2C is not set

#
# Mice
#
# CONFIG_BUSMOUSE is not set
# CONFIG_MOUSE is not set

#
# Joysticks
#
# CONFIG_INPUT_GAMEPORT is not set

#
# Input core support is needed for gameports
#

#
# Input core support is needed for joysticks
#
# CONFIG_QIC02_TAPE is not set
# CONFIG_IPMI_HANDLER is not set

#
# Watchdog Cards
#
CONFIG_WATCHDOG=y
CONFIG_WATCHDOG_NOWAYOUT=y
# CONFIG_ACQUIRE_WDT is not set
# CONFIG_ADVANTECH_WDT is not set
# CONFIG_ALIM7101_WDT is not set
CONFIG_SC520_WDT=y
# CONFIG_PCWATCHDOG is not set
CONFIG_EUROTECH_WDT=m
# CONFIG_IB700_WDT is not set
# CONFIG_WAFER_WDT is not set
# CONFIG_I810_TCO is not set
# CONFIG_MIXCOMWD is not set
# CONFIG_60XX_WDT is not set
# CONFIG_SC1200_WDT is not set
# CONFIG_SCx200_WDT is not set
CONFIG_SOFT_WATCHDOG=m
# CONFIG_W83877F_WDT is not set
# CONFIG_WDT is not set
# CONFIG_WDTPCI is not set
# CONFIG_MACHZ_WDT is not set
CONFIG_DEADMAN=m
# CONFIG_HANGCHECK_TIMER is not set
# CONFIG_AMD7XX_TCO is not set
```

```
# CONFIG_SCx200_GPIO is not set
# CONFIG_AMD_RNG is not set
# CONFIG_INTEL_RNG is not set
# CONFIG_AMD_PM768 is not set
# CONFIG_NVRAM is not set
CONFIG_RTC=y
# CONFIG_DTLK is not set
# CONFIG_R3964 is not set
# CONFIG_APPLICOM is not set
# CONFIG_SONYPI is not set

#
# Ftape, the floppy tape device driver
#
# CONFIG_FTAPE is not set
# CONFIG_AGP is not set
# CONFIG_DRM is not set

#
# PCMCIA character devices
#
# CONFIG_PCMCIA_SERIAL_CS is not set
# CONFIG_SYNCLINK_CS is not set
# CONFIG_MWAVE is not set

#
# IBM Advanced System Management Service Processor
#
# CONFIG_IBMASM is not set

#
# Multimedia devices
#
# CONFIG_VIDEO_DEV is not set

#
# File systems
#
# CONFIG_QUOTA is not set
# CONFIG_AUTOFS_FS is not set
# CONFIG_AUTOFS4_FS is not set
# CONFIG_REISERFS_FS is not set
# CONFIG_ADFS_FS is not set
# CONFIG_AFFS_FS is not set
# CONFIG_HFS_FS is not set
# CONFIG_BEFS_FS is not set
# CONFIG_BFS_FS is not set
# CONFIG_EXT3_FS is not set
# CONFIG_JBD is not set
CONFIG_FAT_FS=y
CONFIG_MSDOS_FS=y
CONFIG_UMSDOS_FS=m
CONFIG_VFAT_FS=y
# CONFIG_EFS_FS is not set
# CONFIG_CRAMFS is not set
```

```
CONFIG_TMPFS=y
CONFIG_RAMFS=y
# CONFIG_ISO9660_FS is not set
# CONFIG_JFS_FS is not set
CONFIG_MINIX_FS=y
# CONFIG_VXFS_FS is not set
# CONFIG_NTFS_FS is not set
# CONFIG_HPFS_FS is not set
CONFIG_PROC_FS=y
CONFIG_PROC_CONFIG=y
# CONFIG_DEVFS_FS is not set
CONFIG_DEVPTS_FS=y
# CONFIG_QNX4FS_FS is not set
# CONFIG_ROMFS_FS is not set
CONFIG_EXT2_FS=y
CONFIG_EXT2_FS_XATTR=y
CONFIG_EXT2_FS_XATTR_SHARING=y
CONFIG_EXT2_FS_XATTR_USER=y
CONFIG_EXT2_FS_XATTR_TRUSTED=y
CONFIG_EXT2_FS_POSIX_ACL=y
# CONFIG_SYSV_FS is not set
# CONFIG_UDF_FS is not set
# CONFIG_UFS_FS is not set
# CONFIG_XFS_FS is not set

#
# Network File Systems
#
# CONFIG_CODA_FS is not set
# CONFIG_INTERMEZZO_FS is not set
CONFIG_NFS_FS=y
CONFIG_NFS_V3=y
CONFIG_NFS_ACL=y
CONFIG_NFS_DIRECTIO=y
CONFIG_ROOT_NFS=y
CONFIG_NFSD=m
CONFIG_NFSD_V3=y
# CONFIG_NFSD_ACL is not set
# CONFIG_NFSD_TCP is not set
# CONFIG_NFSD_FHALIAS is not set
CONFIG_SUNRPC=y
CONFIG_LOCKD=y
CONFIG_LOCKD_V4=y
# CONFIG_CIFS is not set
CONFIG_SMB_FS=y
# CONFIG_SMB_NLS_DEFAULT is not set
# CONFIG_NCP_FS is not set
# CONFIG_ZISOFS_FS is not set
CONFIG_FS_MBCACHE=y
CONFIG_FS_POSIX_ACL=y

#
# Partition Types
#
CONFIG_PARTITION_ADVANCED=y
```

```
# CONFIG_ACORN_PARTITION is not set
# CONFIG_OSF_PARTITION is not set
# CONFIG_AMIGA_PARTITION is not set
# CONFIG_ATARI_PARTITION is not set
# CONFIG_MAC_PARTITION is not set
CONFIG_MSDOS_PARTITION=y
# CONFIG_BSD_DISKLABEL is not set
# CONFIG_MINIX_SUBPARTITION is not set
# CONFIG_SOLARIS_X86_PARTITION is not set
# CONFIG_UNIXWARE_DISKLABEL is not set
# CONFIG_LDM_PARTITION is not set
# CONFIG_SGI_PARTITION is not set
# CONFIG_ULTRIX_PARTITION is not set
# CONFIG_SUN_PARTITION is not set
# CONFIG_EFI_PARTITION is not set
CONFIG_SMB_NLS=y
CONFIG_NLS=y

#
# Native Language Support
#
CONFIG_NLS_DEFAULT="iso8859-1"
CONFIG_NLS_CODEPAGE_437=m
CONFIG_NLS_CODEPAGE_737=m
CONFIG_NLS_CODEPAGE_775=m
CONFIG_NLS_CODEPAGE_850=m
CONFIG_NLS_CODEPAGE_852=m
CONFIG_NLS_CODEPAGE_855=m
CONFIG_NLS_CODEPAGE_857=m
CONFIG_NLS_CODEPAGE_860=m
CONFIG_NLS_CODEPAGE_861=m
CONFIG_NLS_CODEPAGE_862=m
CONFIG_NLS_CODEPAGE_863=m
CONFIG_NLS_CODEPAGE_864=m
CONFIG_NLS_CODEPAGE_865=m
CONFIG_NLS_CODEPAGE_866=m
CONFIG_NLS_CODEPAGE_869=m
CONFIG_NLS_CODEPAGE_936=m
CONFIG_NLS_CODEPAGE_950=m
CONFIG_NLS_CODEPAGE_932=m
CONFIG_NLS_CODEPAGE_949=m
CONFIG_NLS_CODEPAGE_874=m
CONFIG_NLS_ISO8859_8=m
CONFIG_NLS_CODEPAGE_1250=m
CONFIG_NLS_CODEPAGE_1251=m
CONFIG_NLS_ISO8859_1=y
CONFIG_NLS_ISO8859_2=m
CONFIG_NLS_ISO8859_3=m
CONFIG_NLS_ISO8859_4=m
CONFIG_NLS_ISO8859_5=m
CONFIG_NLS_ISO8859_6=m
CONFIG_NLS_ISO8859_7=m
CONFIG_NLS_ISO8859_9=m
CONFIG_NLS_ISO8859_13=m
CONFIG_NLS_ISO8859_14=m
```

```
CONFIG_NLS_ISO8859_15=m
CONFIG_NLS_KOI8_R=m
CONFIG_NLS_KOI8_U=m
CONFIG_NLS_UTF8=m

#
# Console drivers
#
CONFIG_VGA_CONSOLE=y
CONFIG_VIDEO_SELECT=y
# CONFIG_MDA_CONSOLE is not set

#
# Frame-buffer support
#
CONFIG_FB=y
CONFIG_DUMMY_CONSOLE=y
# CONFIG_FB_RIVA is not set
# CONFIG_FB_CLGEN is not set
# CONFIG_FB_PM2 is not set
# CONFIG_FB_PM3 is not set
# CONFIG_UNICON is not set
# CONFIG_FB_CYBER2000 is not set
CONFIG_FB_VESA=y
CONFIG_FB_VGA16=y
# CONFIG_FB_HGA is not set
CONFIG_VIDEO_SELECT=y
# CONFIG_FB_MATROX is not set
# CONFIG_FB_ATY is not set
# CONFIG_FB_RADEON is not set
# CONFIG_FB_ATY128 is not set
# CONFIG_FB_SIS is not set
# CONFIG_FB_NEOMAGIC is not set
# CONFIG_FB_VMWARE_SVGA is not set
# CONFIG_FB_3DFX is not set
# CONFIG_FB_VOODOO1 is not set
# CONFIG_FB_TRIDENT is not set
# CONFIG_FB_VIRTUAL is not set
CONFIG_FBCON_SPLASHSCREEN=y
CONFIG_FBCON_CFB16=y
CONFIG_FBCON_ADVANCED=y
# CONFIG_FBCON_MFB is not set
# CONFIG_FBCON_CFB2 is not set
# CONFIG_FBCON_CFB4 is not set
CONFIG_FBCON_CFB8=y
CONFIG_FBCON_CFB24=y
CONFIG_FBCON_CFB32=y
# CONFIG_FBCON_AFB is not set
# CONFIG_FBCON_ILBM is not set
# CONFIG_FBCON_IPLAN2P2 is not set
# CONFIG_FBCON_IPLAN2P4 is not set
# CONFIG_FBCON_IPLAN2P8 is not set
# CONFIG_FBCON_MAC is not set
# CONFIG_FBCON_VGA_PLANES is not set
CONFIG_FBCON_VGA=y
```

```
# CONFIG_FBCON_HGA is not set
# CONFIG_FBCON_FONTWIDTH8_ONLY is not set
CONFIG_FBCON_FONTS=y
CONFIG_FONT_8x8=y
CONFIG_FONT_8x16=y
# CONFIG_FONT_SUN8x16 is not set
# CONFIG_FONT_SUN12x22 is not set
# CONFIG_FONT_6x11 is not set
# CONFIG_FONT_PEARL_8x8 is not set
# CONFIG_FONT_ACORN_8x8 is not set

#
# Sound
#
# CONFIG_SOUND is not set

#
# USB support
#
# CONFIG_USB is not set

#
# Bluetooth support
#
# CONFIG_BLUEZ is not set

#
# Kernel hacking
#
# CONFIG_DEBUG_KERNEL is not set
# CONFIG_VTUNE is not set

#
# Library routines
#
CONFIG_ZLIB_INFLATE=y
CONFIG_ZLIB_DEFLATE=y
CONFIG_QSORT=y
CONFIG_SUSE_KERNEL=y
```

3.3 Kernel bauen

Nun haben wir unseren Kernel konfiguriert und müssen ihn nun bauen. Dies erfolgt mit dem Befehen:

- make dep
- make clean
- make all
- make bzImage

Das kann man auch mit folgendem Script durchführen. Das nachfolgende Script trägt bei mir den Namen mkkernel und kopiert den gebauten Kernel gleich in die Directorystruktur die später in diesem HOWTO noch beschrieben wird.

```
#! /bin/bash
#
# Build the kernel and copy to gme-development-tree
#
# Copyright (C) 2003 Heimo Schön <heimo.schoen@gmx.at>
#
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
#

cd /usr/src/linux/

make xconfig
#make clean
make dep
#make all
make bzImage
make modules

echo
echo "Kopiere Kernel und Kernelconfiguration ins GME-Directory"
cp -v arch/i386/boot/bzImage /home/heimo/entwicklungen/gme/nanobox/vmlinuz
cp -v .config /home/heimo/entwicklungen/gme/kernel/last-gme-config-$(date "+%y%m%d-%H%M")

cd /home/heimo/entwicklungen/gme/nanobox
echo
echo "$0 done"
```

4 Directory Struktur

4.1 Ansatz

Einleitend: ich bin noch immer ein fauler Mensch und hatte keine Lust eine Directorystruktur zu erfinden.

4.2 Selber machen - oder kopieren

Ich hab seinerzeit eine kleine Distribution namens nanobox verwendet, um eine komplette Dateistruktur zu erhalten, mit /etc, /dev, usw. Diese Methode würde ich Ihnen ans Herz legen, wenn Sie eine Applikation entwerfen, bei der Sie vom Basissystem keine ausgefallenen Spezialitäten wünschen, also gerade mal Kernel und einen einfachen Startup wünschen. In diesem Fall sollten Sie z.B. die Links auf der Homepage von *Busybox* in der Rubrik Products/Projects Using BusyBox"durchsuchen, ob es nicht bereits ein Projekt oder Produkt gibt das Ihren Bedürfnissen sehr nahe kommt. Wie gesagt, bei mir war es seinerzeit das Projekt *Nanobox* das bereits alles inkludiert hat, was für mein Projekt notwendig war.

4.3 Die Dateistruktur meines GME-Projekts

Sie können es sich leichter machen und verwenden z.B. meine beiliegende Directorystruktur *nanobox.gz* und passen diese an Ihre Verhältnisse an. Auch hier können ein bis zwei Stunden nicht schaden. Machen Sie sich vertraut damit was es schon gibt und wie es zusammenhängt.

Hier nocheinmal der Hinweis, daß mein rootfilesystem auf Basis von *Nanobox* aufgebaut wurde.

Laden Sie mein Rootfilesystem, das File *nanobox.gz* mit SHIFT und left-mouse-klick und speichern Sie das File in einem Directory lokal auf Ihrer Disk. (Achtung beim Entpacken wird kein Subdirectory angelegt).

Entpackt wird die Directorystruktur mit:

```
gunzip nanobox.gz
```

Als nächstes müssen wir das entpackte Filesystem mounten. Dazu benutzen wir das Loopback-Device, daß dem Kernel einen Hardwaredriver "vorgaukeltünd in Wirklichkeit eine Device auf einer existierenden Datei mountbar macht. Das Kommando hierzu lautet:

```
mount -o loop ./nanobox /mnt
```

Damit wird mein Rootfilesystem, dass sich in der Datei nanobox befindet, über das Loopback-Device (-o loop) mit dem Mountpoint /mnt verbunden.

Nun sollten Sie folgende Dateien vorfinden im Mountpoint /mnt

```
heimo@ternitz1:~/entwicklungen/gme/nanobox> dir root/
insgesamt 64
drwxr-xr-x    2 root     root         4096 2003-10-05 16:47 bin
drwxr-xr-x    2 root     root        12288 2003-10-05 12:18 dev
drwxr-xr-x    8 root     root         4096 2003-10-05 00:18 etc
drwxrwxrwx    2 heimo    users        4096 2002-07-31 14:55 flashdisk
drwxr-xr-x    5 root     root         4096 2002-07-31 08:26 home
drwxr-xr-x    3 root     root         4096 2003-10-05 16:47 lib
lrwxrwxrwx    1 root     root           11 2003-10-13 06:46 linuxrc -> bin/busybox
drwxr-xr-x    2 root     root         4096 2002-05-02 23:30 lost+found
drwxr-xr-x    2 root     root         4096 2002-04-10 20:55 mnt
drwxr-xr-x    2 root     root         4096 2003-07-01 21:38 proc
drwxr-xr-x    2 root     root         4096 2003-05-03 18:16 root
drwxr-xr-x    2 root     root         4096 2003-10-05 16:47 sbin
drwxrwxrwx    2 root     root         4096 2003-07-01 21:38 tmp
drwxr-xr-x    6 root     root         4096 2003-10-05 16:47 usr
drwxr-xr-x    7 root     root         4096 2002-11-23 23:39 var
```

Machen Sie sich mit dieser Struktur vertraut. Interessant sind vorallem die Dateien im Directory /etc und und der Startup der gesteuert wird durch die Dateien im Directory /etc/init.d

Die Anwenderprogramme dieser Beispielstruktur sind dazu gebaut um von der seriellen Schnittstelle /dev/ttyS1 die Steuerleitungen wie Digitalinputs einzulesen. Diese Applikation misst Geschwindigkeiten im Eisenbahnbereich. Weiters wird am Netzwerk ein ssh-Server und ein Sambaserver angeboten. Insgesamt stellt diese Anwendung eine ganz gute Basis dar von der Sie Ihre eigenen Anwendungen starten können.

Diese Nanobox-Weiterentwicklung benötigt ca. 16 MB Memory und eine Flashdisk mit 16 MB.

Nun können Sie aus dem Mountpoint /mnt die Directorystruktur herauskopieren, an einen beliebigen Punkt in Ihrem Entwicklungstree.

Bei mir hat das Directory, in dem ich die Directorystruktur abgelegt habe, den Namen root, da es das root-directory meines Zielsystems sein wird.

5 Ihre Applikation

5.1 Wohin gehört die Applikation

Im Kapitel 5 (Directory Struktur) finden Sie im Directory /home/heimo die Applikationen die gestartet werden sollen. Im Beispiel der 5 (Directory Struktur) wird der Start aller für die Applikation wichtigen Funktionen durch das Script /home/heimo/gme-start durchgeführt. /home/heimo/gme-start erzeugt ein File im /dev Directory mit Hilfe des Scripts /home/heimo/mkdev, lädt den Kerneldriver der für das Lesen der Handshakeleitungen von der seriellen Schnittstelle dient und startet die Applikation gme-read.

5.2 Wie startet die Applikation

Damit bleibt nun die Frage, wie wird nach dem erfolgreichen Start des Kernels und aller Dämonen, Ihr Applikationsstartscript (im Beispiel /home/heimo/gme-start) gestartet? Dies erfolgt durch das Startscript /etc/init.d/rc.S . Das Script rc.S ist zuständig für den Start aller Dämonen (syslogd, sshd, smbd, nmbd, usw.), für das Mounten aller Dateisysteme, aktivieren der seriellen Schnittstellen, und startet das Script /etc/init.d/rc.net das für das Aktivieren des Netzwerks verantworltlich ist.

Nachfolend das Script rcS:

```
#! /bin/sh

#
# (C) 2002 by Heimo Schön / http://heimo.co.at / heimo.schoen@gmx.at
#
# Bootscript for the GME - Geschwindikeitsmeßeinrichtung
#

mkdir /dev/pts
/bin/mount -t devpts devpts /dev/pts
/bin/mount -t tmpfs none /tmp
/sbin/syslogd

/bin/echo "Mounting filesystems..."
/bin/mount /proc
/bin/mount -a
###mount /dev/proc /proc/ -t proc

###echo "Mounting the flashdisk ..."
###mount -o sync /dev/hda1 /flashdisk 2>&1 > /dev/null
```

```
echo "Activating Serial 2 and 3 ..."
# Devices onboard of the MSM586
/bin/setserial    /dev/ttyS0 irq  4 port 0x3F8 uart 16550A
/bin/setserial    /dev/ttyS1 irq  3 port 0x2F8 uart 16550A
/bin/setserial    /dev/ttyS2 irq  5 port 0x3E8 uart 16550A
/bin/setserial    /dev/ttyS3 irq  7 port 0x2E8 uart 16550A
# port C,D on the AIM104-4-port-serialdevice (if present)
/bin/setserial    /dev/ttyS4 irq 10 port 0x330 uart 16550A
/bin/setserial    /dev/ttyS5 irq 11 port 0x338 uart 16550A
echo ""
echo "Serial ports on this machine:"
/bin/setserial -g /dev/ttyS?
echo ""

/bin/echo "Sourcing the Config-Script ..."
. /etc/init.d/rc.config

# set up motd and issue (and show which kernel we use today)
/bin/echo " [*] $(uname -a) [*]"  >> /etc/nanobox.dpy
/bin/cp /etc/nanobox.dpy /etc/motd
/bin/cp /etc/nanobox.dpy /etc/issue

/bin/echo
/bin/echo "Setting hostname to gme..."
echo $GME_HOSTNAME > /etc/HOSTNAME
/bin/hostname $GME_HOSTNAME

/bin/echo "Starting network configuration..."
/etc/init.d/rc.net

echo "sending bootmessage to ttyS0 ..."
/bin/cat /etc/nanobox.dpy > /dev/ttyS0

/bin/cat /etc/nanobox.dpy

### We moved this stuff to inittab!!!
###/bin/echo "Starting GME Application ..."
###/home/heimo/gme-start &
```

Nachfolend das Script rc.net:

```
#!/bin/ash

#
# (C) 2002 by Heimo Schön / http://heimo.co.at / heimo.schoen@gmx.at
#
# Bootscript for the GME - Geschwindikeitsmeßeinrichtung
#
# abstract:
#    this script brings up the network
#

echo "Bringing up loopback interface"
```

```
/sbin/ifconfig lo 127.0.0.1 up

#echo
#echo Starting interactive network setup...
#echo


#for dev in `/bin/cat /proc/net/dev | /bin/grep eth | /bin/sed  's/^[ \t]*//;s/:/ / ' | /usr/bin/cut -d\  -f1`
#do
#  echo "Use network interface $dev? [y/N] "
#  read  ans
#  if [ "$ans" = "y" -o "$ans" = "Y" ]
#  then
#    echo "Configure $dev using DHCP? "
#    read ans
#    if [ "$ans" = "y" -o "$ans" = "Y" ]
#    then
#      /sbin/udhcpc -n -i $dev -s /etc/udhcpc.script
#    else
#      echo "Please enter the IP address for $dev: "
#      read  ip
#      echo "Please enter the subnet mask for $dev: "
#      read  netmask
#      echo "Configuring eth0...  "
#      /sbin/ifconfig $dev inet $ip netmask $netmask up
#    fi
#    ip=`/sbin/ifconfig $dev | /bin/grep inet | /usr/bin/cut -d: -f2 | /usr/bin/cut -d" " -f1`
#    ifaces="$ifaces $ip/24"
#  fi
#done

ip=$GME_IPADR
echo "Setting up eth0 to $ip"
/sbin/ifconfig eth0 inet $ip netmask $GME_MASK up

#set interfaces line in smb.conf
#####ifaces="$ip/16"
#####echo "interfaces = $ifaces" >> /etc/smb.conf


#echo "Please enter the address of your default gateway [none]: "
#read defaultgw
#if [ "defaultgw" != "" ]
#               then
#               /sbin/route add default gw $defaultgw
#fi
/sbin/route add default gw $GME_GATEWAY


#echo "Please enter your workgroup: "
#read  workgrp
#echo "workgroup = $workgrp" >> /etc/smb.conf
#####echo "workgroup = ARBEITSGRUPPE" >> /etc/smb.conf

echo "Configuring DNS..."
```

```
#echo "Please enter your DNS domain [$workgrp]: "
#read domain
#if [ "$domain" != "" ]
#then
#  echo "search $domain" >> /etc/resolv.conf
#else
#  echo "search $workgrp" >> /etc/resolv.conf
#fi
#echo "search ternitz" >> /etc/resolv.conf

#echo "Please enter the address of your primary DNS server [none]: "
#read dns0
#if [ "$dns0" != "" ]
#then
#  echo "nameserver $dns0" >> /etc/resolv.conf
#fi
#echo "nameserver $GME_GATEWAY" >> /etc/resolv.conf

#echo "Please enter the address of your secondary DNS server [none]: "
#read  dns1
#if [ "$dns1" != "" ]
#then
#  echo "nameserver $dns1" >> /etc/resolv.conf
#fi
#echo "nameserver $GME_GATEWAY" >> /etc/resolv.conf


echo "starting portmapper ..."
/sbin/portmap

#echo "name service cache daemon ..."
#/sbin/nscd

echo "starting sshd ..."
/sbin/sshd

#echo "starting telnetd ..."
#/sbin/tcpd in.telnetd

echo "starting the Samba-Server ..."
/sbin/nmbd
/sbin/smbd

echo
```

5.3 Startup mit init

/etc/init.d/rcS ist also für den Startup des Systems verantwortlich. rcS wird gestartet durch das Programm init. Init wiederum wird gesteuert durch die Init-Tabelle im File /etc/inittab:

```
# /etc/inittab init(8) configuration for BusyBox
#
```

```
# Copyright (C) 1999,2000 by Lineo, inc. and Erik Andersen
# Copyright (C) 1999,2000,2001 by Erik Andersen <andersee@debian.org>
#
#
# Note, BusyBox init doesn't support runlevels.  The runlevels field is
# completely ignored by BusyBox init. If you want runlevels, use sysvinit.
#
#
# Format for each entry: <id>:<runlevels>:<action>:<process>
#
# <id>: WARNING: This field has a non-traditional meaning for BusyBox init!
#
#       The id field is used by BusyBox init to specify the controlling tty for
#       the specified process to run on.  The contents of this field are
#       appended to "/dev/" and used as-is.  There is no need for this field to
#       be unique, although if it isn't you may have strange results.  If this
#       field is left blank, it is completely ignored.  Also note that if
#       BusyBox detects that a serial console is in use, then all entries
#       containing non-empty id fields will _not_ be run.  BusyBox init does
#       nothing with utmp.  We don't need no stinkin' utmp.
#
# <runlevels>: The runlevels field is completely ignored.
#
# <action>: Valid actions include: sysinit, respawn, askfirst, wait, once,
#                                  ctrlaltdel, and shutdown.
#
#       Note: askfirst acts just like respawn, but before running the specified
#       process it displays the line "Please press Enter to activate this
#       console." and then waits for the user to press enter before starting
#       the specified process.
#
#       Note: unrecognised actions (like initdefault) will cause init to emit
#       an error message, and then go along with its business.
#
# <process>: Specifies the process to be executed and it's command line.
#
# Note: BusyBox init works just fine without an inittab. If no inittab is
# found, it has the following default behavior:
#         ::sysinit:/etc/init.d/rcS
#         ::askfirst:/bin/sh
#         ::ctrlaltdel:/sbin/reboot
#         ::shutdown:/sbin/swapoff -a
#         ::shutdown:/bin/umount -a -r
# if it detects that /dev/console is _not_ a serial console, it will
# also run:
#         tty2::askfirst:/bin/sh
#         tty3::askfirst:/bin/sh
#         tty4::askfirst:/bin/sh
#
# Boot-time system configuration/initialization script.
# This is run first except when booting in single-user mode.
#
::sysinit:/etc/init.d/rcS

# /bin/sh invocations on selected ttys
```

```
#
# Note below that we prefix the shell commands with a "-" to indicate to the
# shell that it is supposed to be a login shell.  Normally this is handled by
# login, but since we are bypassing login in this case, BusyBox lets you do
# this yourself...
#
# Start an "askfirst" shell on the console (whatever that may be)
::respawn:-/bin/sh
# Start an "askfirst" shell on /dev/tty2-4
tty2::askfirst:-/bin/sh
tty3::askfirst:-/bin/sh
tty4::askfirst:-/bin/sh

# /sbin/getty invocations for selected ttys
#tty4::respawn:/sbin/getty 38400 tty5
#tty5::respawn:/sbin/getty 38400 tty6

# Example of how to put a getty on a serial line (for a terminal)
#::respawn:/sbin/getty -L ttyS0 9600 vt100
#::respawn:/sbin/getty -L ttyS1 9600 vt100
#
# Example how to put a getty on a modem line.
#::respawn:/sbin/getty 57600 ttyS2

### We moved this stuff to init.d/rcS!!!
#### Start the gme-application
::respawn:/home/heimo/gme-start

# Start the GSM-Connection Utility
#::respawn:/bin/sleep 10 \; /home/heimo/gsm-uplink
::respawn:/home/heimo/gsm-uplink

# Stuff to do before rebooting
::ctrlaltdel:/sbin/reboot
::shutdown:/bin/umount -a -r
#::shutdown:/sbin/swapoff -a
```

rc.config ist ein Applikationsspezifisches Startscript, das für die Aufgabe meiner Applikation Environmentvariablen setzt, die beim Starten der Applikation benötigt werden.

6 Libraries

6.1 Wozu Libraries

Libraries haben den Vorteil, daß Applikationen (Ihre Anwendung, Dämonen, usw.) nicht allen Programmcode enthalten müssen, sondern Funktionen die von verschiedenen Programmen verwendet werden nur in Form einer Library dem System zugänglich gemacht werden müssen. Libraries sparen also Platz auf der Harddisk, indem die Programme viel kleiner sein können und sparen auch zur Laufzeit Platz, da mehrere Programme auf die Libraries zugreifen können die nur noch einmal im Speicher geladen sein müssen.

Dieser Vorteil bringt uns ein bischen Arbeit. Wir müssen immer sicherstellen, daß die richtigen Applikationen und Libraries auf der Flashdisk in Ihrer Directorystruktur vorhanden sein müssen. Hierzu habe ich bei meinen Embedded Systemen folgendes Script im Einsatz, daß in die 5 (Directory Struktur) die notwendigen Applikationen und Libraries implantiert. Damit haben Sie alle Libraries am embedded system verfügbar, die auch auf ihrem Entwicklungssystem verfügbar sind.

6.2 Das Script

Das nachfolgende Script kopiert die Applikationen die in den Variablen BINS und SBINS angegeben sind, in Ihr Zieldirectory, daß Sie zuvor in das Directory rootëntpackt haben. Nach dem implantieren der Programme, werden die notwendigen Libraries ermittelt (mit dem Programm ldd) und in die Director Struktur kopiert.

```
#! /bin/bash
#
# copy all needed files to a Targetdirectory
#
# Original Copyright by:
# Copyright (C) 1998 August Hoerandl <hoerandl@elina.htlw1.ac.at>
#
# Modifications for Nanobox based on a Busybox, made by
# Copyright (C) 2003 Heimo Schön <heimo.schoen@gmx.at>
#
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
#
# $Id: update-cache,v 1.4 1999/04/01 12:09:51 gustl Exp $
# $Log: update-cache,v $
# Revision 1.4  1999/04/01 12:09:51  gustl
# support for glibc
#
# Revision 1.3  1998/11/22 19:07:02  hoerandl
#
# increased ramdisk size
# added printer support
# fixed problem with ld.so.cache
# fixed etc/inetd.conf
#
# Revision 1.1.1.1  1998/09/21 19:28:56  hoerandl
# First Version in CVS
#
#

#set -x
```

```
BINS="clear dmesg fuser gawk login netcat passwd pico rz scp setserial sftp smbclient smbpasswd ssh ssh-add ssh-agent ssh-keyconverter ssh-keygen ssh-keyscan strace stty su sudo sz zip tee"
SBINS="dhcpcd dhcpd hwclock mgetty netdate nmbd nscd portmap pppd smbd sshd tcpd"

function error {
  echo ">>> ERROR: " $* " < ----------------------------------- ERROR"
}

function do_cp {
  cp -puv $LINK $1 $2 || cp -puv $1 $2 || error "cp $1 $2"
}

# ----------- MAIN ----------------

if [ ! -d root ] ; then
  error "couldnot find the target-directory root"
  exit 1;
fi

echo "first we delete old stuff"
rm -rf root/bin/* root/usr/bin/* root/sbin/* root/usr/sbin/*

echo "copy $BINS to /bin:"
for i in $BINS ; do
  if [ -e /bin/$i ] ; then
    do_cp /bin/$i root/bin
    strip root/bin/$i
  elif [ -e /usr/bin/$i ] ; then
    do_cp /usr/bin/$i root/usr/bin
    strip root/usr/bin/$i
  else
    error "cannot locate /bin/$i nor /usr/bin/$i"
  fi
done

echo "copy $SBINS to /sbin:"
for i in $SBINS ; do
  if [ -e /sbin/$i ] ; then
    do_cp /sbin/$i root/sbin
    strip root/sbin/$i
  elif [ -e /usr/sbin/$i ] ; then
    do_cp /usr/sbin/$i root/sbin
    strip root/sbin/$i
  else
    error "cannot locate /sbin/$i nor /usr/sbin/$i"
  fi
done

### echo "copy to /etc:" # ETCS
### for i in $ETCS ; do
###   do_cp /etc/$i root/etc
### done
### echo "set keyboard to $KBD"
### rm ramdisk/etc/default.keyboard
### echo $KBD > ramdisk/etc/default.keyboard
```

```
### echo "copy to /boot:" # BOOT
### for i in $BOOT ; do
###   if [ -e /bin/$i ] ; then
###     do_cp /bin/$i boot
###   elif [ -e /usr/bin/$i ] ; then
###     do_cp /usr/bin/$i boot
###   elif [ -e /sbin/$i ] ; then
###     do_cp /sbin/$i boot
###   elif [ -e /usr/sbin/$i ] ; then
###     do_cp /usr/bin/$i boot
###   else
###     error "cannot locate $i"
###   fi
### done

echo "copy terminfo :"
cp -puv /usr/share/terminfo/l/linux root/usr/lib/terminfo/l
cp -puv /usr/share/terminfo/l/linux root/usr/share/terminfo/l

echo "delete /lib/*"
rm -rf root/lib/*

echo "copy /lib/security/*"
mkdir root/lib/security
cp -r /lib/security/* root/lib/security

echo "copy libraries dynamic ..."
LIBS=$( ldd root/sbin/* root/bin/* root/usr/bin/* root/usr/sbin/* root/lib/security/* 2> /dev/null | grep "[ \t]*lib" | grep -v "ld-linux" | cut -d " " -f 1 | sort -u )
echo "copy $LIBS to /lib :"
for i in $LIBS ; do
  if [ -e /lib/$i ] ; then
    do_cp /lib/$i root/lib
    strip root/lib/$i
  elif [ -e /usr/lib/$i ] ; then
    do_cp /usr/lib/$i root/lib
    strip root/lib/$i
  else
    error "cannot locate /lib/$i nor /usr/lib/$i"
  fi
done

# copy other libaries
echo "copy ld-linux.so.* libnss*"
for i in /lib/ld-linux.so.[0-9] /lib/libnss* ; do
  do_cp $i root/lib
  strip root/$i
done

# busybox updaten
cd ..
cd busybox
make all
make install
cd _install
tar czvf bb.tgz *
```

```
cd ../..
cd nanobox
mv ../busybox/_install/bb.tgz root
cd root
tar xzvf bb.tgz --exclude=tee
mv bb.tgz ../last_bb.tgz
cd ..
```

7 Flashdisk erzeugen

7.1 Einleitung

Dieses Kapitel zeigt mit welchen Scripts ich das im Kapitel 5 (Directory Struktur) gezeigte Filesystem verpacke, so daß es am Zielsystem als Ramdisk dem Kernel zur Verfügung gestellt werden kann.

Hierzu existieren mehrere Scripts, die einzeln gestartet werden können oder gleich in einem Ärbeitsgang"durch Start des nachfolgenden Scripts mit dem Namen mkall:

```
#!/bin/bash

# (W) 2002 by Heimo Schön
#
# abstract:
#   this script creates the cf-card with a nanobox linux
#

if [ $1 ] ; then
  gmenum=$1
else
  echo "You have to specify the Number of the GME in Parameter 1"
  echo "   e.q.: $0 3"
  echo "script $0 stopped."
  exit 1
fi

if [ $2 ]
then
  dev=$2
else
  echo "$0 didn't find your parameter 2"
  echo "   e.q.: $0 $1 /dev/hde1"
  echo "$0 will now write everything to /dev/fd0"
  echo "  So, please insert a floppy to device /dev/fd0"
  echo "  press ^C to cancel operation or press ENTER to proceed"
  read a
  dev=/dev/fd0
fi

# updating all executables and libraries on the target-directory
./update-target-libraries

# build the ramdisk-image-file
./mkrd ${gmenum}
```

```
# write everything to the flashdisk/floppy (on a device)
./mkcfcard ${dev}

echo mkall done.
echo
```

Das Script mkall soll Ihnen nur als Beispiel dienen. Der übergebene Parameter 1 ist in der GME-Applikation erforderlich, der zweite Parameter gibt das Device an, auf dem das Zielsystem angelegt werden soll.

7.2 Ramdisk erzeugen

Das nachfolgende Script mkrd erzeugt das Ramdisk-File. Im Wesentlichen passiert nichts anderes, wie das tarren und zippen der Directorystruktur die Sie im Directory root vorbereitet haben.

```
#!/bin/sh

#
# This script creates the ramdiskfile
#
# Copyright (C) 2002,2003 Heimo Schön <heimo.schoen@gmx.at>
#
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
#

rm -vf ./disk/initrd
rm -vf ./disk/initrd.gz

if [ $1 ] ; then
  echo -n "$1" > gme-num.cfg
else
  echo "You have to specify the Number of the GME in Parameter 1"
  echo "   e.q.: $0 3"
  echo "script $0 stopped."
  exit 1
fi

# RDSIZE ist ein Vielfaches von 1024
# z.B. 25 MB = 25600, usw.
if [ $2 ] ; then
```

```
  RDSIZE=$2
else
  RDSIZE=22528
fi

if [ $3 ] ; then
  RDINODES=$3
else
  RDINODES=4098
fi

echo Begruessung in die ramdisk kopieren
export buildnum=`cat buildnum`
expr $buildnum + 1 > buildnum
echo "Build $buildnum from `date` on $HOSTNAME from $USER"
cp nanobox.dpy.raw nanobox.dpy
echo "       Build $buildnum from `date` on $HOSTNAME from $USER" >> ./nanobox.dpy
echo "       Build $buildnum from `date` on $HOSTNAME from $USER" >> ./nanobox.history
cp ./nanobox.dpy ./root/etc


echo "Application und Kerneldriver in die ramdisk kopieren ..."
cp ../gsm-uplink      root/home/heimo
cp ../gme-start       root/home/heimo
cp ../gme-read        root/home/heimo
cp ../serialint.o     root/home/heimo
cp ../mkdev           root/home/heimo
strip root/home/heimo/gme-read
#strip root/home/heimo/serialint.o -- niemals Kerneldriver stripen
cp getlog             root/bin
cp dellog             root/bin
cp distanz            root/bin

echo ramdisk aufräumen
find -name "*~" -exec rm {} \;

echo "creating an empty file ..."
dd if=/dev/zero of=./disk/initrd bs=1024 count=${RDSIZE}

echo "connecting a loopbackdevice with the empty file ..."
losetup /dev/loop0 ./disk/initrd

echo "creating a minix filesystem in the empty file ..."
mkfs.minix -i ${RDINODES} /dev/loop0

# checken ob der Mountpoint leer ist
umount /mnt &> /dev/null
if [ `ls /mnt | wc -w` -gt 0 ] ; then
  echo
  echo "KRISE !!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
  echo "der Mountpoint /mnt ist nicht leer!!!"
  echo "Möglicherweise ist beim letzten Build etwas daneben gegangen!!!"
  echo
  rm -fv /mnt/*
fi
```

```
echo "coping the files into the new minix filesystem ..."
mount /dev/loop0 /mnt
cd ./root
find . | cpio -pud /mnt &> /dev/null
cd ../
sudo umount /mnt
rm -fv /mnt/*

echo "freeing the new filesystem ..."
losetup -d /dev/loop0

echo
echo "zipping the ramdisk (may take some time) ..."
gzip --fast ./disk/initrd

echo "coping the ramdisk to the build directory ..."
cp ./disk/initrd.gz ./nanobox.gz

echo
echo "ramdisk created."
exit 0
```

7.3 Flashdisk anlegen

Im letzten Schritt wird die Flashdisk formattiert und mit allen notwendigen Files bespielt. Dies erledigt das Script mkcfcard

```
#!/bin/sh

# (W) 2002 by Heimo Schön
#
# this script is based on the ideas of nanobox embedded linux
# from http://www.neonbox.org
#
# abstract:
#   this script creates the cf-card with a nanobox linux
#

if [ $1 ]
then
  dev=$1
else
  # if ypur floppy isn't /dev/fd0, change this...
  dev=/dev/fd0
fi

#echo 'Please insert a blank floppy disk to create the nanobox boot disk.'
#echo 'WARNING: Any data on the floppy will be erased.'
#echo 'Press a key when ready...'
#read -s dummy

echo 'Formatting floppy disk...'
```

```
umount $dev
#fdformat $dev
mkfs.msdos $dev

echo 'Creating Filesystem...'
mkdosfs $dev

echo 'Making disk bootable...'
./syslinux $dev

echo 'Copying Files...'
mkdir /tmp/$$
mount $dev /tmp/$$
cp -fv vmlinuz /tmp/$$/.
cp -fv nanobox.gz /tmp/$$/.
cp -fv syslinux.cfg /tmp/$$/.
cp -fv nanobox.dpy /tmp/$$/.
cp -fv putty.exe /tmp/$$/.
cp -fv gme-num.cfg /tmp/$$/.
echo
echo Your new cfcard:
echo
ls -alp /tmp/$$
umount $dev
rm -rf /tmp/$$

echo
echo "Done."
echo "Your boot floppy/flashdisk is ready for use."
echo "Remove it from $1 and put it to your target system for testing it ..."
echo "                Have a nice day."
echo
echo
```

8 Allgemeines

8.1 Copyright

Dieses Dokument ist urheberrechtlich geschützt. Das Copyright liegt bei Heimo Schön.

Das Dokument darf gemäß der GNU General Public License verbreitet werden. Insbesondere bedeutet dieses, daß der Text sowohl über elektronische wie auch physikalische Medien ohne die Zahlung von Lizenzgebühren verbreitet werden darf, solange dieser Copyright Hinweis nicht entfernt wird. Eine kommerzielle Verbreitung ist erlaubt und ausdrücklich erwünscht. Bei einer Publikation in Papierform ist der Autor zu informieren.

This howto is free documentation; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, BUT NO later version.

This document is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this document; if not, write

to the: Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.