

# Embedded Systems im Heimbereich - Powered by Linux

Johannes Bauer

3. Erlanger Linuxtage

15. Januar 2005

# Embedded Systems

2/31

Wo treten sie auf?

- ▶ Intelligente Steuergeräte für elektronische Geräte aller Art
- ▶ Meist proprietäre Betriebssystemlösungen, kaum veränderbar
- ▶ Schwierigkeit: Spezifikation und Dokumentation der Hardware nicht verfügbar

# Embedded Systems

3/31

## Vielfalt der Prozessoren

- ▶ Einfache 8-bit Mikrocontroller: z.B. Atmel AVR (*Advanced Virtual RISC*), Microchip PIC-Familie (*Peripheral Interface Controller*)
- ▶ Komplexere Mikrocontroller: ARM-Familie (z.B. StrongARM, ARM7, ARM9), *Acorn RISC Machine*
- ▶ Weitere Schwierigkeit: Compilieren des Codes
  - ▶ Gibt es Entwicklungswerkzeuge?
  - ▶ Was passiert bei einer Chiprevision?
  - ▶ Ist der Hersteller zuverlässig?

# Embedded Systems

## Kleincomputer

4/31

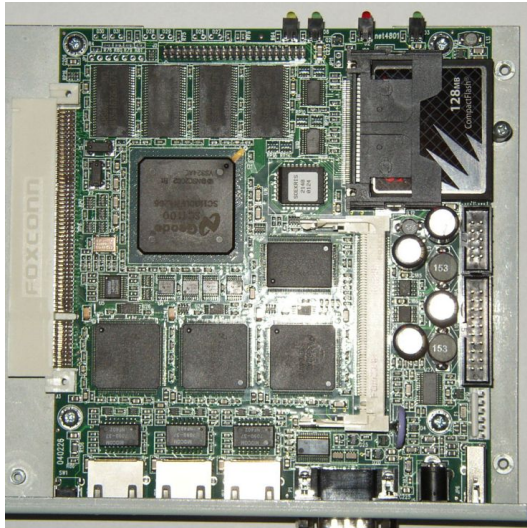
- ▶ Anderer Ansatz: eine bereits bekannte Prozessorfamilie verwenden (x86)
- ▶ Miniaturisierung quasi automatisch durch starke Weiterentwicklung
- ▶ Alle Komponenten zusammensetzen zu einem „vollwertigen“ Computer

# Typische Hardware

Soekris net4801

5/31

- ▶ 266MHz CPU, 128 MB SD-RAM
- ▶ 3 Netzwerkkarten
- ▶ USB
- ▶ Optional: Festplatte
- ▶ Primäres Bootmedium: CompactFlash-Karte



# Typische Hardware

## Bedienung

7/31

- ▶ Bedienung des Systems über den „Console“ Port
- ▶ Serielle Schnittstelle (RS232)
- ▶ Verbindung mit bedienendem PC über Nullmodemkabel
- ▶ Terminal-Programm ist gleichzeitig „Tastatur“ und „Bildschirm“

# Typische Hardware

## Erstinstallation

8/31

- ▶ Beste Option: herausnahme des CF-Speichers und Anschluss an Kartenlesegerät
- ▶ CF-Speicherchip partitionieren (fdisk) und formatieren (mkfs.ext3)
- ▶ Systempartition mounten und System darauf kopieren
- ▶ Wechsel in die entsprechende Umgebung (chroot), Installation des Bootloaders (lilo bzw. grub-install)



# Typische Hardware

## Erstinstallation

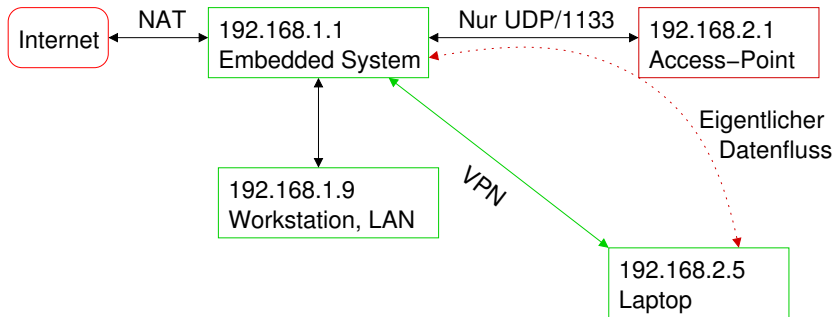
9/31

- ▶ Nachteil des CF-Chips: Flash hat begrenzte Haltbarkeit (in etwa 100.000 Schreibzyklen)
- ▶ Wahl einer „industrial grade“ Speicherkarte (ca. 600.000 Schreibzyklen)
- ▶ Mounten des Dateisystems read only

# Nutzfelder

10/31

## Wireless-Router mit OpenVPN-Funktionalität

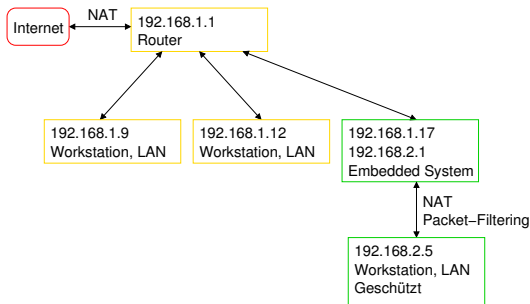


# Nutzfelder

## LAN-Router

11/31

- ▶ Abschotten einzelner Rechner oder Netzwerkabschnitte (Crossover/Switch)



# Nutzfelder

12/31

## Sonstige Möglichkeiten

- ▶ Überwachungseinheit mit Anschluß z.B. an eine Digitalkamera
- ▶ Temperatur/Feuchtigkeitssensor mit Auswertungen und Netzchnittstelle
- ▶ Zusammenführen von drei Teilnetzen, die eventuell nicht gleichberechtigt sind

# Soentoo

13/31

„Soekris running Gentoo“

- ▶ Meta-Distribution, die auf die Soekris Kleincomputer ausgelegt ist
- ▶ Spezielle Hardwareeigenschaften werden berücksichtigt (CPU, Watchdog, Bugs)
- ▶ Relativ einfache Installation, die sich um praktisch alles kümmert

# Soentoo

14/31

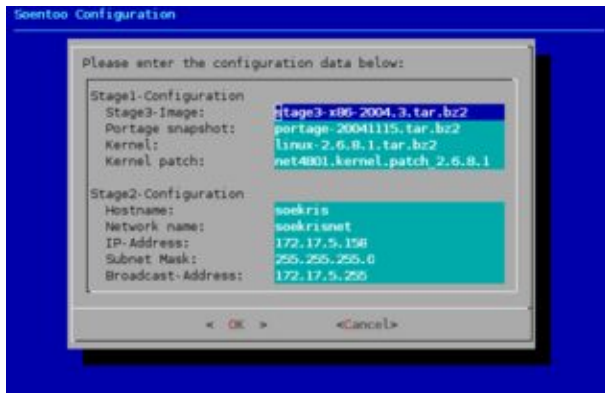
## Die Installationsschritte

- ▶ Konfiguration
- ▶ Stage1
  - ▶ Entpacken der Gentoo-Daten
  - ▶ Entpacken der Kernel
  - ▶ Anpassen der Konfigurationsdateien
- ▶ Stage2
  - ▶ Download und Compilierung der selektierten Pakete
  - ▶ Compilierung und Installation der Kernel
- ▶ Stripping
  - ▶ Entfernen aller nicht notwendigen Pakete
  - ▶ Distribution einsatzfähig machen

# Soentoo

15/31

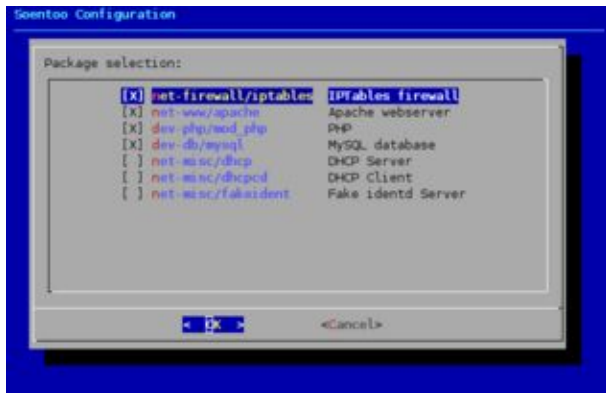
## Die Konfiguration, Schritt 1/3



# Soentoo

## Die Konfiguration, Schritt 2/3

16/31

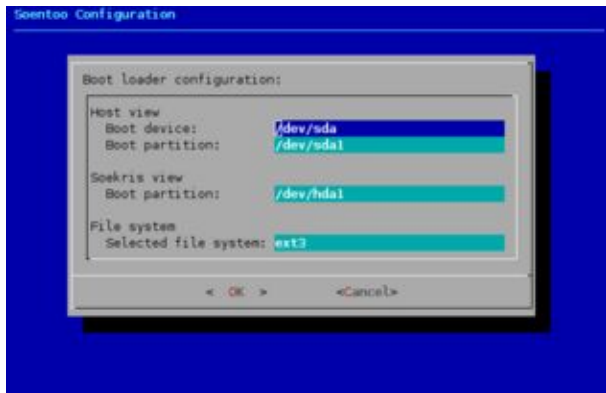




# Soentoo

## Die Konfiguration, Schritt 3/3

17/31



```
joe [/home/joe/Soekris]: ./Create_Image
Extracting Stage3-Image      : 1:09 ms
Extracting Portage snapshot: 0:44 ms
Extracting kernel          : 0:45 ms
Complete Stage1 installation took 2:39 ms.

Now leaving Stage1 and entering Stage2 of installation...
...Stage2 of installation started.

Starting kernel compilation: 5:19 ms
Emerging sys-boot/lilo      : 0:54 ms
Emerging iptables          : 0:34 ms
Emerging dev-db/mysql      : 15:15 ms
Emerging dev-php/mod_php   : 29:19 ms
Emerging net-www/apache    : 5:55 ms
Complete Stage2 installation took 57:20 ms.

Stage 2 installation completed. The image is now bootable and could be
stripped.

joe [/home/joe/Soekris]: []
```

# Soentoo

## Die Installationsschritte

19/31

- ▶ Nach Stage 1 und 2: prinzipiell lauffähiges System auf dem Wirtscomputer
- ▶ Größe allerdings noch jenseits von Gut und Böse (ca. 1,5 GB)
- ▶ Nun können zuerst noch weitere Applikationen installiert werden (emerge)

# Soentoo

## Größenreduktion

20/31

- ▶ Mehrere Optionen, die Größe zu Reduzieren:
  - ▶ Auslagern der zu großen Dateien per NFS
  - ▶ Löschen unnötiger Dateien (Stripping)
- ▶ Stripping ist irreparabel, vorher Sicherheitskopie anlegen

```
joe [/soekris]: ./Strip_Image
To strip an image, you need to be 'chrooted' on it.
Are you absolutely sure this is the correct environment?

You may ruin your system otherwise!

Are you sure (y/n) y
Giving you 10 more seconds to think it over...
5...4...3...2...1...
we're going in...

Unmerging sys-apps/slocate                : 0:01 ms
Unmerging app-editors/nano                 : 0:11 ms
Unmerging app-shells/sash                 : 0:17 ms
Unmerging dev-lang/perl                   : 0:21 ms
Unmerging dev-util/yacc                   : 0:07 ms
Unmerging sys-apps/groff                  : 0:19 ms
Unmerging sys-apps/man                    : 0:18 ms
Unmerging sys-devel/autoconf              : 0:19 ms
Unmerging sys-devel/automake              : 0:20 ms
Unmerging sys-devel/binutils              : 0:22 ms
Unmerging sys-devel/bison                 : 0:20 ms
Unmerging sys-devel/gcc                   : 0:19 ms
Unmerging sys-devel/gcc-config            : 0:08 ms
Unmerging sys-devel/gettext              : 0:20 ms
Unmerging sys-devel/libperl               : 0:07 ms
Unmerging dev-lang/python                  : 0:24 ms
Complete unmerge took                      : 4:15 ms

Removal of remaining directories           : 0:19 ms

Complete stripping took                    : 4:34 ms
joe [/soekris]: []
```

# Erste Schritte

## Bootvorgang

22/31

- ▶ Anstecken des Nullmodemkabels
- ▶ Überwachen des Bootvorgangs
- ▶ Erster Login auf der seriellen Konsole: „root“,  
Passwort „soekris“

```
comBIOS ver. 1.24 20040312 Copyright (C) 2000-2004 Soekris Engineering,  
net4801
```

```
0128 Mbyte Memory CPU Geode 266 Mhz
```

```
Pri Mas SanDisk SDCFB-12B LBA 980-B-32 125 Mbyte
```

```
PXE-M00: BootManage UEFI, PXE-2.0 (build 082)
```

Slot	Vend	Dev	Class	Rev	Cnd	Stat	CL	LT	HT	Base1	Base2	Int
0:00:0	1078	0001	06000000	0107	0280	00	00	00	00	00000000	00000000	00
0:06:0	100B	0020	02000000	0107	0290	00	3F	00	00	0000E101	A0000000	10
0:07:0	100B	0020	02000000	0107	0290	00	3F	00	00	0000E201	A0001000	10
0:08:0	100B	0020	02000000	0107	0290	00	3F	00	00	0000E301	A0002000	10
0:18:2	100B	0502	01018001	0005	0280	00	00	00	00	00000000	00000000	00
0:19:0	DE11	ADF8	0C031008	0117	0280	08	38	00	00	A0003000	00000000	11

```
1 Seconds to automatic boot. Press Ctrl-P for entering Monitor.
```

```
LIL0 22.6 boot:
```

```
Loading Soentoo.....
```

```
BIOS data check successful
```

```
[
```

```
* Activating (possible) swap... [ ok ]
* Remounting root filesystem read-only (if necessary)... [ ok ]
* Checking root filesystem.../dev/hda1: clean, 12252/31360 files, 111980/125424 blocks
[ ok ]
* Remounting root filesystem read/write... [ ok ]
* Setting hostname to soekris... [ ok ]
* Calculating module dependencies... [ ok ]
* Using /etc/modules.autoload.d/kernel-2.6 as config:
* Loading module errorled_control... [ ok ]
* Autoloaded 1 module(s)
* Checking all filesystems... [ ok ]
* Mounting local filesystems... [ ok ]
* Mounting USB device filesystem (usbfs)... [ ok ]
* Activating (possibly) more swap... [ ok ]
* Caching service dependencies... * Setting system clock to hardware clock [UTC]... [ ok ]
* Configuring kernel parameters... [ ok ]
* Updating environment... [ ok ]
* Cleaning /var/lock, /var/run... [ ok ]
* Cleaning /tmp directory... [ ok ]
* Setting user font... [ ok ]
* Loading key mappings... [ ok ]
* Bringing lo up... [ ok ]
* Initializing random number generator... [ ok ]
INIT: Entering runlevel: 3
* Bringing eth0 up (172.17.5.158)... [ ok ]
* Starting mysqld... [ ok ]
* Mounting network filesystems... [ ok ]
* Starting sshd... [ ok ]
* Starting apache2... [ ok ]
* Starting local... [ ok ]

This is soekris.soekrisnet (Linux i586 2.6.8.1) 06:36:21
soekris login: █
CTRL-@ & Z for help | 19200 BNI | NDR | Minicom 2.00.0 | VT102 | Online 00:01
```



# Erste Schritte

25/31

## Netzwerkumgebung prüfen

- ▶ Ist mit *ifconfig* das Interface richtig gesetzt worden?
- ▶ Ist sshd gestartet worden?
- ▶ Wenn ja, Switch an eth0 anstecken, ssh zur Box versuchen

```
acer joe [~]: ssh root@172.17.5.158
Password:
Last login: Sat Jan  5 06:38:26 1980 from joe.alex.stv.uni-erlangen.de

soekris [~]: ls
total 3.0K
-rw-r--r-- 1 root root 1.3K 05.01.1980 06:38:40 .bash_history
-rw-r--r-- 1 root root  29  20.10.2004 15:05:10 .inputrc
-rw-r--r-- 1 root root  0  18.07.2004 02:47:19 .keep
soekris [~]: echo 'Hello world, Soentoo!'
Hello world, Soentoo!
soekris [~]: uptime
 06:38:56 up 4 min,  1 user,  load average: 0.11, 0.30, 0.15
soekris [~]: cat /proc/cpuinfo
processor       : 0
vendor_id     : Geode by NSC
cpu family    : 5
model         : 9
model name    : Unknown
stepping      : 1
cpu MHz       : 266.771
fdiv_bug      : no
hlt_bug       : no
f00f_bug      : no
coma_bug      : no
fpu           : yes
fpu_exception : yes
cpuid level   : 2
wp            : yes
flags         : fpu tsc mtr cx8 cmov mmx cxmm
bogomips     : 524.28

soekris [~]: df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/hda1       115M  106M   7.0M   94% /
none            62M   0    62M   0% /dev/shm
soekris [~]:
```

# Selber ein System einrichten

27/31

## Serielle Konsole

- ▶ *Serial Console Support* in Kernel einkompilieren
- ▶ Parameter herausfinden (Geschwindigkeit, Parität, Datenbits, Stoppbits)
- ▶ In Kernel-Boot-Zeile z.B. anhängen:  
„console=ttyS0,19200n81“ (19200 Baud, keine Parität, 8 Daten- und 1 Stoppbit)

# Selber ein System einrichten

28/31

`/etc/inittab`

- ▶ Nur ein *agetty* starten, das auf `ttyS0` läuft
- ▶ `ttyS0` in die Liste der „sicheren“ Terminals aufnehmen (`/etc/securetty`)

# Selber ein System einrichten

29/31

## Bootloader

- ▶ Flash-Speicher ist in der Box `/dev/hda`, im Wirtssystem z.B. `/dev/sda`
- ▶ Bootloader, z.B. lilo muss wie folgt konfiguriert werden:
  - ▶ `boot = /dev/sda`
  - ▶ `root = /dev/sda1`
  - ▶ `append = "root = /dev/hda1"`

# Zusammenfassung

30/31

- ▶ Geringer Platz- und Strombedarf
- ▶ Lautlos durch Flash-Speicher und passiv gekühlte CPU
- ▶ Sehr flexibel einsetzbar
- ▶ Linuxinstallation sehr einfach

# Abschluß

31/31

- ▶ Danke für die Aufmerksamkeit!