

# Architekturen eingebetteter Linux-Systeme - eine Einführung

## Überall Linux

von René Rebe

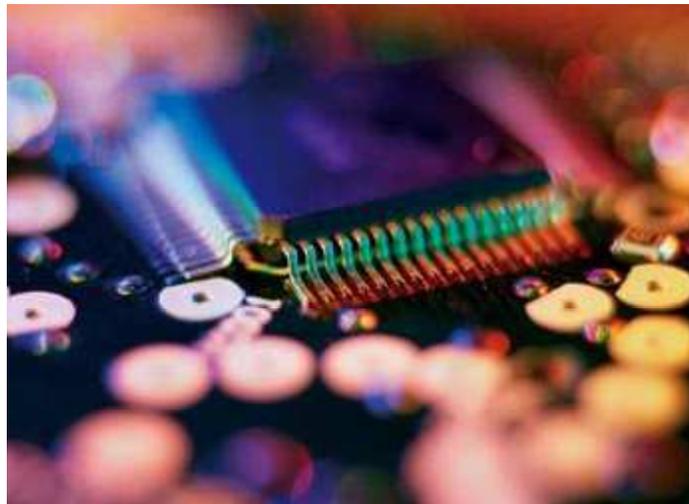
Ob Accesspoints, Firewalls, Network Attached Storage, Navigationsgeräte oder Handys - auf unendlich vielen Geräten läuft Linux. Über das Warum, das Wie und über die Eigenschaften typischer Embedded-Architekturen klärt dieser Artikel auf.

Als Linus Torvalds 1991 als Hobby die Arbeit am Unix-artigen Betriebssystem Linux begann, legte er es nur für Intel-80386-Prozessoren aus. Es unterstützte nur AT-Festplatten und Linus selbst war der Meinung, dass es wohl nicht sehr portabel sei. Doch mit zunehmender Verbreitung folgten in den 90er Jahren erste Linux-Portierungen auf DEC's Alpha, Sun's Sparc und andere CPU-Familien.

Heute unterstützt Linux neben I-386 nicht nur viele andere CPUs wie ARM, AVR32 [1], Mips oder PowerPC, sondern auch Architekturen ohne Memory Management Unit (MMU), beispielsweise Motorola 68000 und Derivate sowie digitale Signalprozessoren wie die Blackfin-Familie [2] von Analog Devices. Linux hat sich zu einem portablen und skalierbaren Betriebssystem entwickelt, das auf fast jeder CPU zu Hause ist und auch durch die monolithische Architektur mit sehr guter Performance glänzt (siehe Kasten "Embedded-Linux-Hardware").

Monolithisch bedeutet, dass alle Kernelkomponenten, Subsysteme und Treiber in einem Adressraum residieren und sich direkt aufrufen. Im Unterschied dazu lagern Microkernel alle weniger zentralen Komponenten in eigene Adressräume aus, was zwar das Speicherschutz-Niveau erhöht, aber den Datenaustausch erschwert und zusätzliche Kontext-Switches erzwingt.

Der monolithische Aufbau macht Linux-System jedoch nicht groß und unbeweglich: Der Quellcode [3] bietet Konfigurationsmöglichkeiten, um nicht benötigte Subsysteme und Funktionen zu deaktivieren und so das System genau an den jeweiligen Verwendungszweck anzupassen.



Anzeige:

### Mythos und Wahrheit über Open-Source-Sicherheit

Ist es riskant, Open-Source-Software in geschäftskritischen Infrastrukturen einzusetzen? Warum sollten wir für einen Open-Source-Anbieter zahlen, wenn es sich doch um "freie" Software handelt? Dieses Whitepaper von ASTARO räumt mit Mythen über Open-Source-Sicherheit auf.

[Whitepaper-Download](#)



Abbildung 1: NGW 100, ein Beispiel für ein Linux-Board mit AVR32-Prozessor, 140 MHz Takt und 32 MByte RAM.



Abbildung 2: Als weiteres Beispiel das Linux-Board Blackfin Stamp mit einem 533-MHz-DSP, aber ohne MMU.

### Extrem einfach portierbar

Der modulare Aufbau sowie der hohe Anteil an C-Code sorgen dafür, dass sich Linux einfach auf beliebige Prozessoren portieren lässt, solange sie über mindestens 32 Bit breite Register verfügen. Mit Portierungen für Renesas H8 und Intel 8086 (ELKS, Embeddable Linux Kernel Subset) existieren sogar spezielle Portierungen für 16-Bit-Systeme. Derlei Anpassungen sind allerdings mit größerem Aufwand verbunden und eher selten.

Der Befehlssatz jeder Prozessorfamilie ist anders. Grob unterscheidet man zwischen komplexen Instruktionen (Cisc) und einfacheren, reduzierten (Risc). Durch den einfacheren Aufbau sind Risc-CPU's in der Regel günstiger zu fertigen und erreichen einfacher höhere Taktfrequenzen. Die einzelnen CPU-Familien sind sehr unterschiedlich (siehe Tabelle 1). Einige, etwa Intel x86, adressieren neben dem Speicher ihre I/O-Ports separat.

<b>Tabelle 1: CPUs mit Linux-Unterstützung</b>			
<b>Prozessorfamilie</b>	<b>Bits</b>	<b>MMU</b>	<b>Byte-Order</b>
Alpha AXP	64	ja	Big-Endian
Sun Sparc	32/64	ja	Big-Endian
Motorola 68000	32	teilweise	Big-Endian
PowerPC	32/64	ja	Big-Endian
ARM	32	ja	Big-/Little-Endian
Hitachi Super H	32/64	ja	Little-Endian
Mips	32/64	teilweise	Big-/Little-Endian
Microblaze	32	teilweise	Big-Endian
HP PA-Risc	32/64	ja	Big-Endian
Intel IA-64	64	ja	Little-Endian
x86	32/64	ja	Little-Endian
Axis CRIS	32	ja	Big-Endian
Renesas M32R	32	teilweise	-
Atmel AVR32	32	ja	Big-Endian
Renesas H8/300	8/16	nein	-
NEC V850	32	nein	Little-Endian
Tensilica Xtensa	32	teilweise	Big-/Little-E.

Analog Devices Blackfin	32	nein	Little-Endian
-------------------------	----	------	---------------

Neben der Breite der Register, die sich in der Regel zwischen 8 und 64 Bit bewegt und entscheidet, wie viel Daten der Prozessor mit einem Befehl bewegen und wie viel Speicher er adressieren kann, gibt es das Unterscheidungsmerkmal der Byte-Reihenfolge (Byte-Order oder Endianness).

Legt eine CPU Daten, die länger als 8 Bits sind, mit den höchstwertigen Bytes zuerst ab, bezeichnet man sie als Big-Endian. Prozessoren, die niederwertige Bytes vor den höherwertigen speichern, heißen Little-Endian. Einige Prozessoren beherrschen sogar beide Adressierungsarten.

### Auch ohne MMU

Die für Schutz und Paging zuständige Memory Management Unit fehlt den kleinsten Embedded-CPU's aus Komplexitäts- und Kostengründen. Kernel und Anwendungen laufen bei ihnen nicht voneinander getrennt, Programmierfehler führen dann schnell zum Systemabsturz. Bei Geräten mit fest definierter Funktion und ohne die Möglichkeit externer Software-Installation, zum Beispiel Digitalkameras oder Drucker, ist eine solche Systemkonfiguration aber durchaus die Regel. Einige Prozessoren bieten statt einer MMU eine Memory Protection Unit (MPU), die zwar kein Paging und somit keinen virtuellen Speicher kennt, aber die einzelnen Anwendungen und den Kernel voneinander schützt.

Wegen des Miniaturisierungs- und Kostendrucks sind die meisten Embedded-Chips keine reinen CPUs, sondern fungieren als System-on-Chip (SoC). Sie beherbergen nicht nur den Rechenkern, sondern zugleich den Speichercontroller und eine Fülle von Peripherie-Anschlüssen, etwa serielle USB- und Grafik-Schnittstellen, Ethernet und Audio. Zur praktischen Funktion benötigen die meisten SoC nur noch Speicherbausteine (ROM, RAM) und ein wenig passive Interface-Beschaltung. Eine andere Spielart stellen FPGAs mit CPU-Kernen (etwa mit PowerPCs) dar sowie synthetisierbare CPU-Cores wie Microblaze und M32R, die es erlauben, Linux mit Zusatzlogik in FPGAs zu verwenden.

Linux, speziell in eingebetteten Geräten, bietet viele Vorteile - allen voran natürlich die Freiheit des Quellcode. Durch die Verfügbarkeit der Quellen und damit auch aller Treiber ist es in der Entwicklung eines Produkts möglich, beliebige Veränderungen an der Hardware vorzunehmen und Fehler zeitnah zu lokalisieren. Linux bietet einen ausgereiften Netzwerkstack samt Firewall, IPv6 und Bridge-Funktionalität. Viele fertige Treiber und die CPU-Unterstützung erlauben es, die Hardware für ein Produkt frei nach den Funktionsvorgaben zu wählen. Eine intelligente Wahl der Hardwarekomponenten reduziert auch die Produktionskosten. Dank der Erweiterungen RTAI und RT Linux lassen sich zudem Echtzeitanforderungen abdecken. (Dazu erscheint in einem der nächsten Linux-Magazine ein Artikel.)

## Embedded-Linux-Hardware

Das Angebot an SoC-Microprozessoren, die Linux unterstützt, ist sehr vielfältig. Hier ein grober Überblick: Die AVR32-Produktreihe von Atmel [1] ist vielleicht das bekannteste Beispiel für SoC (Abbildung 1). AVR32 zeichnet sich durch guten Durchsatz bei niedriger Stromaufnahme aus. Der AT32AP7002 beispielsweise schafft 210 Dhrystone-Mips bei einer Taktrate von 150 MHz. Ausstattungsmerkmale sind 16-Bit-Stereo-DAC, LCD/TFT-Interface, Image Sensor Interface, 10/100-MBit/s-Ethernet, USB, RS232, USART, I2C, SPI und PS2. Der Chip unterstützt zudem gängige Speicherkartenformate. Für portable Geräte bieten sich Prozessoren mit dem AVR-UC3-Kern an. Die ARM-Baureihe AT91CAP erlaubt es dem Entwickler, zusätzliche Logik im Chip zu realisieren, für die er sonst einen extra FPGA oder Asic vorsehen müsste.

### Power-Serie verdient ihren Namen

AMCC zielt auf den Embedded-Markt mit der PowerPC-400-Produktlinie [4]. Je nach Modell legt der Hersteller den Schwerpunkt auf Bildverarbeitung, Wireless, industrielle Lösungen oder Speicher- und Netzwerkanwendungen. Eine Übersicht liefert der Product Selector Guide auf der Webseite. Die Prozessoren bieten unter anderem je nach Modell 10-MBit- bis 1-GBit/s-Ethernet, PCI-Express-Interfaces, SATA-Controller mit Raid-5- und Raid-6-Support. Optional enthalten die beiden Modelle 405EX und 460EX eine Crypto-Engine, die den Prozessorkern beim Verarbeiten derartiger Daten entlastet.

Auch Broadcom spezialisiert sich mit den auf Mips basierenden Prozessoren [5] der BCM-Baureihe auf den Netzwerkbereich. Die Baureihe umfasst je nach Modell bis zu vier Prozessorkerne. Typische Ausstattungsmerkmale sind Gigabit Ethernet, DDR2, PCI und Hyper Transport (HT).

Freescale [6] hat gleich drei verschiedene Prozessorarchitekturen im Sortiment: ARM, Coldfire und Power Architecture, wobei die letztere das breiteste Produktportfolio bietet. Erwähnenswert ist hier die QUICC-Engine-

Technologie: Die chipinterne Kommunikationseinheit entlastet den Hauptkern von Netzwerk-Kommunikationsaufgaben [7].

Intel hat mit seiner Xscale-Architektur [8], die auf dem ARM-5TE-Befehlssatz basiert, ebenfalls Produkte für SoC im Programm. Ein aktuelles Topmodell ist der CE-2110-Mediaprozessor [9], konzipiert für Geräte der Unterhaltungsindustrie, etwa digitale Settop-Boxen und netzwerkfähige Mediaplayer. Er beinhaltet eine 1-GHz-CPU, Mpeg-2- und H.264-Hardware-Videodecoder, einen DSP für Audio- und Videocodecs, 3D-Grafikbeschleuniger, Hardwarebeschleuniger zur Ver- und Entschlüsselung von DVB-CSA, DES, 3DES, AES sowie verschiedene Video- und Audio-Ein- und Ausgänge. Schnittstellen für PCI, ATA, USB, GPIO, UART, I2C, SPI und Smartcards sind ebenfalls vorhanden.

### **Multimedia-Spezialisten**

Die Multimedia-geeignete Blackfin-Serie von Analog Devices [2] enthält einen 16-/32-Bit-Prozessorkern mit DSP (Abbildung 2). Die mit 200 bis 750 MHz getakteten Blackfins besitzen ein Powermanagement, das nach Vorgabe der Software auf Stromaufnahme oder Performance optimiert. Viele Varianten arbeiten bei Umgebungstemperaturen zwischen -40 und +85 °C, was sie für raue industrielle Anwendungen interessant macht. Es gibt auch Low-Cost-Ausführungen, die zu den anderen Pin- und Software-kompatibel sind.

Die Blackfin-Familie wird gern für Kryptographie, IP- und Mobil-TV, Fahrerassistenzsysteme, Biometrie, VoIP bis hin zu GSM eingesetzt. Labview, ein auch unter Linux laufendes grafisches Programmiersystem von National Instruments, das in der Mess- und Automatisierungstechnik gängig ist, unterstützt diese Produktreihe mit dem "Labview Embedded Module for ADI Blackfin Processors" [10].

P.A. Semi bietet mit der PWRficient-Familie einen - nach Embedded-Maßstäben - Hochleistungsprozessor der IPowerPC-Architektur an [11]. Der 2-GHz-Chip hat je nach Applikation eine Leistungsaufnahme von 5 bis 13 Watt. Auf dem Chip finden sich 2 MByte L2-Cache, zwei DDR2-Speichercontroller, Hardwarebeschleunigung für TCP/IP und Kryptographie (IPsec und SSL) sowie ISCSI- und Raid-Funktionen. Des Weiteren besitzt der Chip ein konfigurierbares I/O-System mit insgesamt acht PCI-Express-Interfaces, vier 1-GBit/s- und zwei 10-GBit/s-Ethernet-Schnittstellen.

Renesas ist mit den Chip-Architekturen Super H und M32R vertreten [12]. Die Baureihe SH-Mobile ist für Applikationen in Mobiltelefonen ausgelegt. SH7763, die neueste Super-H-Entwicklung, zielt auf anspruchsvolle Applikationen wie Highend-Audio- und -Video-Equipment, Settop-Boxen und Medienserver, aber auch auf Telekommunikationshardware wie Residential Gateways, Router und IP-Telefone. Der SoC hat einen 32-Bit-SH-4A-CPU-Kern und eine Floatingpoint-Unit, die 1,9 Gflops verarbeitet bei 266 MHz. Der Chip enthält eine Multimediacard-Schnittstelle, ein Simcard-Interface, ein vierkanaliges Serial Sound Interface (SSI), einen A/D- und einen D/A-Wandler, einen PCMCIA-Card-Controller, zwei 1-GBit/s-Ethernet-Ports und einen TFT-Displaycontroller.

ST Microelectronics, ebenfalls ein gut eingeführter SoC-Mitspieler, verkauft auf ARM-Architektur basierende SoC-Prozessoren [13]. Mit der STM32-Produktfamilie ist ST Microelectronics einer der ersten Hersteller, der Prozessoren mit ARM-Cortex-M3-Kern [14] anbietet. Deren Architektur versucht den Zielkonflikt von Performance, Energieverbrauch und Kosten möglichst gut zu lösen. Axel Bauer

### **Standardkonform und grafisch**

Linux\ Posix-Konformität erleichtert die Portierung bestehender Programme. Im Gegensatz zu proprietären Embedded-Systemen lassen sich Programme und Bibliotheken auch auf einem normalen Linux-Desktop beziehungsweise auf Workstations im Sinne von Rapid Prototyping entwickeln. Ein im Netzwerk transparentes GUI (zum Beispiel X11) erlaubt es, die Darstellung der Applikation komfortabel auf dem Zielgerät zu prüfen und erst in einer späteren Phase die Applikation nativ für das Zielsystem zu übersetzen.

Historisch gewachsen und speziell durch Neuentwicklungen im Embedded-Bereich existiert eine Reihe von Subsystemen zur Grafikdarstellung und Programmierung von Linux (siehe Abbildung 3). Die Palette reicht vom klassischen X-Window-System (X11), das auch auf Linux-Desktops Verwendung findet, bis hin zur direkten Pixel-Manipulation im Grafikspeicher (Frame Buffer, FB). Direct FB nimmt eine Art Mittelweg, der den Kernel-FB-Treiber um chipspezifische Hardwarebeschleunigung und Managementfunktionen erweitert.



Abbildung 3: Ein Embedded-Linux-System ist in Schichten aufgebaut. Auf die Hardware legt sich der Kernel, in dem auch die Treiber enthalten sind. Eine C-Bibliothek macht die Kernelfunktionen für die darüber liegenden Schichten nutzbar.

Auf den einzelnen Systemen bauen dann Userinterface-Bibliotheken auf, welche die Widgets der Applikationen implementieren. Zur Auswahl stehen mehrere C- und C++-Bibliotheken. Die bekanntesten Vertreter sind GTK+ des Gnome-Projekts [15] und QT [16] der Firma Trolltech (beide für X11 und FB), die Enlightenment Foundation Libraries (Evas, Edge) sowie Fox, Fltk und Wx Widgets. Die freie OpenGL-Implementierung Mesa [17] sorgt entweder im Rahmen von X11 oder einzeln als OpenGLES (OpenGL Embedded Systems) für 3D-Darstellung.

### Aufbau in Schichten

Ein Linux-System besteht aus dem Linux-Kernel und Anwendungsprogrammen, die ein Dateisystem zur Verfügung stellt. Das Root-System residiert bei normalen Linux-PCs auf einer Festplattenpartition, auf Embedded-Systemen im ROM oder Flashspeicher. Als Dateisysteme kommen hier meist das Journaling Flash Filesystem in der Version 2 (JFFS2), das nur lesbare Squash-FS oder andere, komprimierte Standard-Dateisysteme zum Einsatz. Über die Verzeichnisstruktur entscheidet der Entwickler selbst, im einfachsten Fall reicht ein einzelnes Programm als Init-Prozess. In der Regel wird man aber ein minimales System, das aus Bibliotheken und einzelnen Programmen besteht, unterbringen.

Es bietet sich an, neben den ausgewachsenen GNU-Programmen, zum Beispiel »awk«, Bash, den Coreutils oder »sed«, speziell auf geringe Größe optimierte Programme zu verwenden, um eigenen Skripten eine gewohnte Unix-Umgebung zu bieten. Größenoptimierte Tools liefert zum einen das Busybox-Projekt als Paket, zum anderen gibt es sie einzeln, etwa als Embutils, Minised und so weiter.

### Sprachen und Programme

Im Gegensatz zu einigen proprietären Systemen hat der Entwickler unter Embedded Linux die freie Wahl der Programmiersprache. Hier kann er nach den Projektvorgaben oder eigenem Belieben zwischen C, C++, Objective C, D, Ada, Pascal, Java und vielen anderen aussuchen.

Auch die aus dem Unix-Umfeld gewohnten Skriptsprachen, etwa Perl, Python, Ruby, PHP, Lua und so weiter, stehen weitgehend zur Verfügung. Einige benötigen lediglich etwas Hilfe bei der Cross-Kompilation. (Tipps dazu gibt ein demnächst im Linux-Magazin erscheinender Artikel).

Für Projekte, die auch Standardfunktionen enthalten, bietet es sich an, auf etablierte Linux-Software, Tools und Application Stacks zurückzugreifen, beispielsweise Apache, Samba, Squid, Asterisk oder Callweaver. Sie bieten bewährte Funktionalität und Qualität und beschleunigen die Programmierung. Der Entwickler kann sich so auf die eigenen Produktmerkmale konzentrieren.

Für die Übersetzung der Linux-Software steht der GCC, zum Debuggen der GDB bereit [18]. Beide sind den meisten Entwicklern bekannt und kommen auch für viele proprietäre Embedded-Systeme zum Einsatz. Der GNU-Debugger lässt sich remote über eine serielle Schnittstelle oder Ethernet verwenden.

In letzter Zeit kommt es in Mode, mit dem Emulator Qemu [19] komplette Embedded-Systeme zu virtualisieren. Auf diese Weise testen Entwickler ihre Programme und Systeme auch ohne Zielhardware, also direkt am Arbeitsplatz. Zum komfortablen Arbeiten bietet sich Eclipse an, für das einige kommerzielle Hersteller auch dedizierte Embedded-Linux-Plugins im Programm haben.

### Bootloader und Funktion überwachen

Die im Desktop-Linux-Bereich üblichen Bootloader Lilo und Grub sind derzeit nur für x86-Systeme ausgelegt. Nur Grub wird in der kommenden Version 2 auch PowerPC und Sparc unterstützen. In Embedded-Geräten finden sich hauptsächlich der aus Ppboot hervorgegangene Bootloader "Das U-Boot" [20], der sehr viele

CPU-Architekturen unterstützt, und Redboot [21] aus dem von Red Hat gekauften Embedded-Betriebssystem E-Cos von Cygnus.

Neben klassischen Watchdogs, die nach Ablauf eines Timers das System neu starten, bieten sich Kexec und Kdump in eingebetteten Systemen an. Diese ermöglichen zum einen, ein Linux-System ohne Umweg über den Bootloader zu starten, und zum anderen ein komplettes Speicherabbild zu sichern. Beide Systeme machen es möglich, vor dem Neustart ein Fallback-System zu laden und so Diagnosemeldungen und Protokolle zur Fehlerbereinigung und Produktverbesserung - auch von Geräten im Produktiveinsatz - zu erstellen.

### Fazit: Pinguin die beste Wahl

Rund um Linux findet sich eine aktive Embedded-Gemeinde, die komfortable Systemlösungen entwickelt und auch Support dafür anbietet. Die Linux-Lösungen reichen nicht nur an den Funktionsumfang kommerzieller Systeme heran, sondern übertreffen diese mit interessanten Features und Funktionen.

Ob O(1)- oder CFQ-Scheduler, die Native Posix Thread Library (NPTL), Futex oder Inotify - Linux skaliert hervorragend und API-Innovationen sind kurzfristig realisierbar. Jede Komponente des Systems, sogar den Kernel, detailliert anpassen zu können sowie die Entwicklung von Treibern für neue Hardwarekomponenten, das ist in der Embedded-Entwicklung wichtig und üblich - beste Voraussetzungen also, um gerade den Pinguin einzubetten. (hej/jk)

Infos
[1] Atmel Corporation: [ <a href="http://www.atmel.com">http://www.atmel.com</a> ] [2] Blackfin-Prozessoren: [ <a href="http://www.analog.com/blackfin">http://www.analog.com/blackfin</a> ] [3] Linux-Kernel: [ <a href="http://www.kernel.org">http://www.kernel.org</a> ] [4] AMCC PowerPC-Produktlinie: [ <a href="http://www.amcc.com">http://www.amcc.com</a> ] [5] Broadcom Communication Processors: [ <a href="http://www.broadcom.com/products/Enterprise-Networking/Communications-Processors">http://www.broadcom.com/products/Enterprise-Networking/Communications-Processors</a> ] [6] Freescale: [ <a href="http://www.freescale.com">http://www.freescale.com</a> ] [7] Freescale QUICC-Engine-Technologie: [ <a href="http://www.freescale.com/webapp/sps/site/overview.jsp?nodeId=02VS0I81285985">http://www.freescale.com/webapp/sps/site/overview.jsp?nodeId=02VS0I81285985</a> ] [8] Intel Xscale: [ <a href="http://www.intel.com/design/intelxscale/index.htm">http://www.intel.com/design/intelxscale/index.htm</a> ] [9] Intel CE 2110 Mediaprozessor: [ <a href="http://www.intel.com/design/celect/2110/">http://www.intel.com/design/celect/2110/</a> ] [10] NI Development Kit: [ <a href="http://sine.ni.com/nips/cds/view/p/lang/en/nid/202659">http://sine.ni.com/nips/cds/view/p/lang/en/nid/202659</a> ] [11] P.A. Semi: [ <a href="http://www.pasemi.com">http://www.pasemi.com</a> ] [12] Renesas Technology: [ <a href="http://www.renesas.com">http://www.renesas.com</a> ] [13] Microcontroller von ST Microelectronics: [ <a href="http://www.st.com/mcu/">http://www.st.com/mcu/</a> ] [14] ARM Cortex-M3: [ <a href="http://www.arm.com/products/CPUs/ARM_Cortex-M3.html">http://www.arm.com/products/CPUs/ARM_Cortex-M3.html</a> ] [15] GTK+: [ <a href="http://www.gtk.org">http://www.gtk.org</a> ] [16] QT und Qtopia: [ <a href="http://trolltech.com">http://trolltech.com</a> ] [17] Mesa: [ <a href="http://mesa3d.sourceforge.net">http://mesa3d.sourceforge.net</a> ] [18] GNU Compiler Collection: [ <a href="http://gcc.gnu.org">http://gcc.gnu.org</a> ] [19] Qemu: [ <a href="http://fabrice.bellard.free.fr/qemu/">http://fabrice.bellard.free.fr/qemu/</a> ] [20] Das U-Boot: [ <a href="http://sf.net/projects/u-boot">http://sf.net/projects/u-boot</a> ] [21] Redboot: [ <a href="http://ecos.sourceware.org/redboot/">http://ecos.sourceware.org/redboot/</a> ]
Der Autor
René Rebe ist einer der Geschäftsführer der Exact CODE GmbH in Berlin und durch die tägliche Arbeit mit Linux in diverse Open-Source-Projekte involviert.

Dieser Online-Artikel kann Links enthalten, die auf nicht mehr vorhandene Seiten verweisen. Wir ändern solche "broken links" nur in wenigen Ausnahmefällen. Der Online-Artikel soll möglichst unverändert der gedruckten Fassung entsprechen.

[Impressum](#) | © 2008 [Linux New Media AG](#)

Partner-Sites

Deutschland: [[LinuxUser](#)] [[EasyLinux](#)] [[Linux-Community](#)] [[Linux-Nachrichten](#)] [[Linux Events](#)] [[OpenBytes](#)]

Europa: [[EasyLinux Polen](#)] [[Linux Magazine Polen](#)] [[Darmowe Programy](#)] [[Linux Magazin Rumänien](#)] [[Linux Magazine Spanien](#)]

International: [[Linux Magazine International](#)] [[Linux Pro Magazine](#)] [[Linux Magazine Brasilien](#)]