Embedded Linux lernen mit dem Raspberry Pi

Jürgen Quade studierte Elektrotechnik an der TU München. Danach arbeitete er dort als Assistent am Lehrstuhl für Prozessrechner (heute Lehrstuhl für Realzeit-Computersysteme), promovierte und wechselte später in die Industrie, wo er im Bereich Prozessautomatisierung bei der Softing AG tätig war. Heute ist Jürgen Quade Professor an der Hochschule Niederrhein, wo er u.a. das Labor für Echtzeitsysteme betreut. Seine Schwerpunkte sind Echtzeitsysteme, Embedded Linux, Rechner- und Netzwerksicherheit sowie Open Source. Als Autor ist er vielen Lesern über das dpunkt-Buch »Linux-Treiber entwickeln« und die regelmäßig erscheinenden Artikel der Serie »Kern-Technik« im Linux-Magazin bekannt.

Jürgen Quade

Embedded Linux lernen mit dem Raspberry Pi

Linux-Systeme selber bauen und programmieren

Jürgen Quade
quade@hsnr.de

Lektorat: René Schönfeldt
Copy Editing: Ursula Zimpfer, Herrenberg
Satz: data2type GmbH, Heidelberg
Herstellung: Frank Heidt
Autorenfotos: privat (Quade)
Umschlaggestaltung: Helmut Kraus, www.exclam.de
Druck und Bindung: M.P. Media-Print Informationstechnologie GmbH, 33100 Paderborn

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

ISBN 978-3-86490-143-0

1. Auflage 2014

Wieblinger Weg 17
69123 Heidelberg

5 4 3 2 1 0

Vorwort

Embedded Systems

Die Zahlen der Marktforscher sind beeindruckend: Bei einer Weltbevölkerung von über 7 Milliarden Menschen werden pro Jahr mehr als 13 Milliarden Prozessoren hergestellt. Ein vergleichsweise kleiner Anteil davon (etwa 350 Millionen) landet in Form eines PCs oder Notebooks auf unserem Schreibtisch. Der erheblich größere Teil wird aber in Waschmaschinen, Autos, TV-Geräten, Digicams, Smartphones oder Automatisierungsanlagen eingebettet.

Embedded Linux

Der Markt dieser eingebetteten Systeme kann grob in zwei Lager eingeteilt werden. Sogenannte Deeply Embedded Systems setzen auf einfache 8- oder 16-Bit-Prozessoren, wie der bekannte Atmega, der im populären Arduino-Board steckt. Anwendungs- und Systemsoftware ist häufig proprietär und die Geräte erledigen vorwiegend einfache Aufgaben, so beispielsweise die Spiegelsteuerung in einem Auto. Werden demgegenüber komplexere Funktionen — allem voran Vernetzung — gefordert, greift der Entwickler für ein Open Embedded System zu 32-Bit-Prozessoren und immer häufiger zu einem Standardbetriebssystem, insbesondere Linux. Linux treibt heute Fernseher, Digicams, Router-Hardware, Uhren und vieles mehr an. Das auch aus gutem Grund, schließlich ist es funktional, bekannt und vor allem Open Source.

Ein eigenes Embedded Linux konfektionieren

Allerdings wird Linux in einem eingebetteten System nur selten über eine Standarddistribution, wie beispielsweise Ubuntu, installiert. Das liegt nicht nur an der meist nicht konformen Hardware. Vielmehr wird das System auf die leistungsschwächere Hardware, auf andere Hardwareplattformen (ARM statt x86) und auf die auszuführende Funktionalität abgestimmt. Auch werden andere Update-Zyklen benötigt als auf dem Desktop üblich. Hinzu kommt die Notwendigkeit, Sensoren und Aktoren anzukoppeln und softwaretechnisch anzusprechen.

Hierfür benötigen Entwickler ein umfangreiches Know-how. Das beginnt beim Entwicklungsprozess, der typischerweise in Form einer Host-/Target- und Cross-Entwicklung abläuft, der Entwicklungsumgebung, die linuxspezifisch kommandozeilenorientiert ist, geht über notwendige Kernelerweiterungen, um damit eigene Hardwareerweiterungen anzusprechen, und endet bei den Applikationen, die nicht zuletzt

aufgrund minimaler Ressourcen nur bedingt auf vorhandene Funktionen aufsetzen können.

Dieses Know-how möchte das vorliegende Buch in praxisorientierter und kompakter Weise vermitteln.

Der Raspberry Pi als Praxisbeispiel

Damit Sie die Inhalte nachvollziehen können, werden viele Techniken mithilfe des Raspberry Pi vorgestellt. Der Raspberry Pi ist ein Anfang 2012 auf den Markt gekommener Kleincomputer, der sich nicht zuletzt wegen seiner leichten Erweiterbarkeit gut als Herzstück eines eingebetteten Systems einsetzen lässt. Mit einem ARM-Prozessor ausgestattet, ist er zudem bei einem Preis von unter 40 € (ohne Speicherkarte, ohne Netzteil) preiswert. Dieser günstige Preis zusammen mit einem »Fun-Faktor« haben für eine hohe Verbreitung gesorgt. Aus diesem Grund eignet er sich besonders gut als Basis für eigene Experimente im Bereich Embedded Linux.

Ziel des Buchs: Ein eigenes Embedded-Linux-System

Methodisch werden Sie an das Thema durch den Aufbau und die Konfektionierung eines komplett eigenen Systems herangeführt. Neben dem Lerneffekt steht am Ende ein eingebettetes System für den Raspberry Pi, das effizienter, schneller und vor allem auch sicherer als eine Standarddistribution ist.

Vom Systemanwender zum Systementwickler: Während die meisten Bücher rund um den Raspberry Pi zeigen, wie Sie — häufig auf Basis der Linux-Variante Raspbian — Systeme unterschiedlicher Funktionalität aufbauen, entwickeln Sie mithilfe des vorliegenden Mitmach-Buches und des Raspberry Pi Ihr eigenes Embedded Linux. Mit der Anwendung im Blick zeigt das Buch, woraus ein Embedded Linux besteht und wie es funktioniert. Es erläutert Hintergründe und zeigt Lösungen, die sich auch in zeitkritischeren Umgebungen einsetzen lassen. Der Raspberry Pi ermöglicht den schnellen und einfachen Einstieg in die Welt eingebetteter Linux Systeme.

Das Thema Embedded Linux lässt sich in einem Buch von rund 300 Seiten auch als Einführung nicht annähernd vollständig abhandeln. Die Auswahl und Tiefe der dargebotenen Aspekte orientieren sich primär an deren Wichtigkeit, an der Aktualität, aber auch an meinen eigenen Fachkenntnissen. Sie erfolgen in eher kompakter Form.

Kempen, im März 2014

Inhaltsverzeichnis