Inhaltsverzeichnis

1 Einleitung 1

2 Gut zu wissen 9

2.1 Die Architektur eingebetteter Systeme 11
2.1.1 Hardware 11
2.1.2 Software 14
2.1.3 Auf dem Host für das Target entwickeln 19
2.2 Arbeiten mit Linux 21
2.2.1 Die Shell 23
2.2.2 Die Verzeichnisstruktur 24
2.2.3 Editor 25
2.3 Erste Schritte mit dem Raspberry Pi 26
2.3.1 System aufspielen 27
2.3.2 Startvorgang 29
2.3.3 Einloggen und Grundkonfiguration 30
2.3.4 Hello World: Entwickeln auf dem Raspberry Pi 30

3 Embedded von Grund auf 33

3.1 Der Linux-Kernel 34
3.2 Das Userland 41
3.2.1 Systemebene 43
3.2.2 Funktionsbestimmende Applikationen 59
3.3 Cross-Development für den Raspberry Pi 64
3.3.1 Cross-Generierung Kernel 64
3.3.2 Cross-Generierung Userland 67
3.3.3 Installation auf dem Raspberry Pi 71
3.4 Bootloader »Das U-Boot« 76
3.4.1 Kernel von der SD-Karte booten 80
3.4.2 Netzwerk-Boot 84
3.5 Initramfs: Filesystem im RAM 86

4 Systembuilder Buildroot 95

4.1 Überblick 95
4.2 Buildroot-Praxis 99
4.2.1 Installation auf der SD-Karte 101
4.2.2 Netzwerk-Boot per U-Boot 104
4.3 Systemanpassung 110
4.3.1 Postimage-Skript 111
4.3.2 Postbuild-Skript 113
4.4 Eigene Buildroot-Pakete 131
4.4.1 Grundstruktur 131
4.4.2 Praxis 137
4.5 Hinweise zum Backup 141

5 Anwendungsentwicklung 143

5.1 Cross-Development 144
5.2 Basisfunktionen der eingebetteten Anwendungsprogrammierung 147
5.2.1 Modularisierung 148
5.2.2 Realzeitaspekte 150
5.3 Hardwarezugriffe 155
5.3.1 Systemcalls für den Hardwarezugriff 156
5.3.2 GPIO-Zugriff über das Sys-Filesystem 162

6 Gerätetreiber selbst gemacht 167

6.1 Einführung in die Treiberprogrammierung 168
6.1.1 Grundprinzip 169
6.1.2 Aufbau eines Gerätetreibers 170
6.1.3 Generierung des Gerätetreibers 173
6.2 Schneller GPIO-Treiberzugriff 176
6.2.1 Digitale Ausgabe 177
6.2.2 Digitale Eingabe 185
6.2.3 Programmierhinweise zum Hardwarezugriff 192

7 Embedded Security 197

7.1 Härtung des Systems 199
7.1.1 Firewalling 200
7.1.2 Intrusion Detection and Prevention 212
7.1.3 Rechtevergabe 213
7.1.4 Ressourcenverwaltung 219

7.1.5 Entropie-Management 224
7.1.6 ASLR und Data Execution Prevention 225
7.2 Entwicklungsprozess 226
7.3 Secure-Application-Design 229
7.3.1 Sicherheitsmechanismen in der Applikation 230
7.3.2 Least Privilege 231
7.3.3 Easter Eggs 233
7.3.4 Passwortmanagement 233
7.3.5 Verschlüsselung 235
7.3.6 Randomisiertes Laufzeitverhalten 236

8 Ein komplettes Embedded-Linux-Projekt 237

8.1 Hardware: Anschluss des Displays 238
8.2 Software 240
8.3 Systemintegration 249

Anhänge

A Crashkurs Linux-Shell 259
B Crashkurs vi 269
C Git im Einsatz 273
D Die serielle Schnittstelle 279
Literaturverzeichnis 283
Stichwortverzeichnis 287