

Inhaltsverzeichnis

Einleitung	11
Teil I Einführung und Einrichtung einer Entwicklungsumgebung	17
I Embedded Linux	19
I.1 Desktop-Betriebssysteme	19
I.2 Bare-Metal vs. Betriebssystem	20
I.2.1 Mikroprozessoren vs. Mikrocontroller	21
I.3 Embedded Betriebssysteme	22
I.4 Die Architektur von Linux	23
I.4.1 Erläuterungen	23
I.5 Beliebte Linux-Distributionen	27
I.6 Linux installieren	29
I.6.1 Parallele Installation von Linux Mint zum vorhandenen Betriebssystem	30
I.6.2 Installation von Linux Mint in VirtualBox	32
I.7 Erfahrungen Linux Mint + VirtualBox	39
I.7.1 Größe der virtuellen Festplatte ändern	39
I.7.2 Zielpartition für VirtualBox festlegen	40
I.7.3 Umziehen der virtuellen Maschine	40
I.7.4 Nicht genügend Arbeitsspeicher	42
I.8 Weiterführende Literatur	42
2 Netzwerkanbindung	43
2.1 Datenaustausch zwischen Host und Embedded System	43
2.1.1 Samba	43
2.1.2 FileZilla	52
2.1.3 Daten austauschen mit scp	55
2.2 Verzeichnisstruktur von Linux/Raspbian	59
2.2.1 /bin	61
2.2.2 /boot	61
2.2.3 /dev	62
2.2.4 /etc	63
2.2.5 /home	63
2.2.6 /lib	64
2.2.7 /lost+found	64
2.2.8 /media	64
2.2.9 /mnt	65
2.2.10 /opt	65

2.2.II	/proc	65
2.2.I2	/root	66
2.2.I3	/run	66
2.2.I4	/sbin	66
2.2.I5	/selinux	66
2.2.I6	/srv	67
2.2.I7	/sys	67
2.2.I8	/tmp	67
2.2.I9	/usr	67
2.2.20	/var	67
2.3	Neue Benutzer und Gruppen einrichten	67
2.3.1	Einen User hinzufügen bzw. entfernen	68
2.3.2	Gruppen hinzufügen bzw. entfernen	68
2.4	Weiterführende Literatur	69
3	Shell-Programmierung	71
3.1	Erste Schritte	72
3.1.1	Die Kommandos »man« und »info«	72
3.2	Geschichte der Shells	73
3.3	Die Bourne-again-Shell – bash	73
3.3.1	Ein- und Ausgabeumleitung	74
3.3.2	Shell-Variablen	78
3.3.3	Kommentare	85
3.3.4	Systemkommandos in Shellscrip	85
3.3.5	Mehrere Kommandos in einer Zeile	87
3.3.6	Bedingungen/Vergleiche	88
3.3.7	Funktionen in Shellscrip	92
3.3.8	Schleifen	102
3.3.9	Professionelle Übergabe von Argumenten	106
3.3.10	Einschränkungen bei Shellscrip	111
3.4	Weiterführende Literatur	112
4	Cross-Toolchains	113
4.1	Cross-Toolchains für Raspberry Pi B+	113
4.1.1	Toolchain und IDE für Windows	115
4.1.2	Toolchain und IDE für Linux (Mint)	116
4.1.3	Toolchain und IDE für Mac OS X	118
4.2	Die Bibliothek wiringPi	118
4.2.1	Herunterladen von wiringPi	119
4.2.2	wiringPi »bauen«	120
4.2.3	Funktionen in wiringPi	121
4.2.4	Weitere Informationen zu wiringPi	121
4.2.5	Anschlussbelegung des Raspberry Pi B+	122
4.3	Konfiguration von Code::Blocks	123
4.3.1	Auswahl des Compilers	124
4.3.2	Einstellen der Compiler-Optionen	124
4.3.3	Bibliothek(en) hinzufügen	125
4.3.4	Erweitern des Suchpfades	126

4.3.5	Toolchain executables	127
4.3.6	Testen der Toolchain	128
4.4	crosstool-ng	134
4.4.1	Vorarbeiten	135
4.4.2	Erstellen und installieren von crosstool-ng	136
4.4.3	Toolchain konfigurieren	137
4.5	Weiterführende Literatur	141
Teil II Techniken zur Programmierung von Kernel und rootfs		143
5	Raspbian – der Kernel	145
5.1	Überblick	146
5.1.1	Einmalig durchzuführende Schritte	146
5.1.2	Zu wiederholende Schritte	146
5.2	Kernel erzeugen – detaillierte Anleitung	147
5.2.1	Einmalig durchzuführende Schritte – Details	147
5.2.2	Zu wiederholende Schritte	150
5.3	Das Shellscript mkrpi	166
5.3.1	mkrpi – das Listing zum Shellscript	167
5.3.2	Funktion und Anwendung von mkrpi	175
5.4	Weiterführende Literatur	180
6	Das root-Dateisystem – rootfs	181
6.1	rootfs erzeugen	182
6.1.1	Benötigte Software	183
6.1.2	Die nächsten Schritte	185
6.1.3	Imagedatei erzeugen	198
6.1.4	Schreiben der Boot-Partition	203
6.1.5	Schreiben des root-Dateisystems	205
6.1.6	Das Ende naht	206
6.2	Alternative Methode	207
6.2.1	Beschaffung und Anwendung von Buildroot	208
6.3	Weiterführende Literatur	215
7	Der Bootprozess	217
7.1	Bare-Metal-Systeme	217
7.2	Geräte mit Betriebssystem	218
7.2.1	Der Bootprozess des Raspberry Pi	218
7.2.2	Der Bootprozess beim BeagleBone Black BBB	219
7.2.3	Der Bootprozess beim Cubieboard	219
7.3	Allgemeine Beschreibung des Bootvorgangs	220
7.3.1	Bootloader	220
7.3.2	Die Aufgabe von Bootloadern	221
7.4	Das U-Boot und der Raspberry Pi	224
7.4.1	Sourcecode von »Das U-Boot«	224
7.5	Weiterführende Literatur	241

Teil III Grundlagen der Treiberentwicklung

243

8	Treiber und Module I	245
8.1	Auffrischung	246
8.2	»Normale« Dateien und Gerädateien	247
	8.2.1 Schnittstellen zwischen User Space und Kernel	248
	8.2.2 Schnittstellen zwischen Kernel und Hardware	249
	8.2.3 Wichtige Programme im User Space	251
8.3	Weitere Voraussetzungen	255
8.4	Das erste Kernelmodul	256
	8.4.1 Quelltext des Moduls und Makefile	257
	8.4.2 Kompilieren des Moduls	258
	8.4.3 Modul testen	258
	8.4.4 Details zu nix.c/nix.ko	261
	8.4.5 kbuild	261
8.5	Ein weiteres einfaches Kernelmodul	263
	8.5.1 Der Sourcecode	263
	8.5.2 Log-Level	265
	8.5.3 Kernelmodul ausprobieren	265
8.6	Moderne Variante von hellodriver	266
	8.6.1 Moderne Variante des hellodriver-Moduls	267
8.7	Ende der Einführung	268
8.8	Weiterführende Literatur	269
9	Treiber und Module II	271
9.1	Auf dem Weg zu einem richtigen Gerät	271
	9.1.1 Funktionen, Makros, Datentypen	271
	9.1.2 Der Sourcecode	279
9.2	Weiterführende Literatur	290
10	Treiber und Module III	291
10.1	Checkliste für die Treiberentwicklung	291
	10.1.1 Headerdateien	291
	10.1.2 Die Struktur file_operations	292
	10.1.3 Initialisierung eines Treibers/Moduls	293
	10.1.4 Entfernen von Treibern/Modulen	295
	10.1.5 Funktion mydevice_open	296
	10.1.6 Funktion mydevice_close	297
	10.1.7 Schreiben und Lesen	298
	10.1.8 The End	300
	10.1.9 Generelle Erklärung einiger Funktionen	301
10.2	Ansteuerung »echter« Hardware	303
	10.2.1 GPIO-Funktionen	304
	10.2.2 GPIOs anwenden	306
	10.2.3 (Mögliche) Erweiterung des Treibers	310
10.3	Weiterführende Literatur	312

Teil IV Treiberentwicklung in der Praxis

313

II	Praxis I	315
II.1	Das serielle Schieberegister SN74HC595	315
I2	Praxis II	329
I2.1	Der Baustein Maxim 7219	329
I2.1.1	Beschreibung des Maxim 7219	330
I2.1.2	Zeitverhalten bei der Ansteuerung	331
I2.1.3	Kaskadieren mehrerer Maxim 7219/7221	332
I2.2	Ansteuerung einer 8 x 8-LED-Matrix	333
I2.2.1	Der Schaltplan	333
I2.3	Die Treibersoftware	335
I2.3.1	Das Makefile	336
I2.3.2	Die Headerdatei max7219.h	337
I2.3.3	Der C-Sourcecode max7219imp.c	338
I2.3.4	Das Testprogramm für den Treiber	346
I2.3.5	Verbesserungsvorschläge	351
I2.4	Ansteuerung von 7-Segment-Anzeigen	352
I3	Praxis III	353
I3.1	Der HD44780 – Aus dem Datenblatt	353
I3.2	Die Hardware	355
I3.3	Die Headerdatei hd44780.h	356
I3.3.1	Einige Erläuterungen	359
I3.4	Der Treiber hd44780.c	361
I3.4.1	Erläuterung des Programms	371
I3.5	Das Testprogramm im User Space	375
I3.6	Weiterführende Literatur	380
A	Literaturverzeichnis	381
A.1	Embedded Systeme, Architektur etc.	381
A.2	VirtualBox	381
A.3	Samba	381
A.4	Shell-Programmierung	381
A.5	Toolchains und Bibliotheken	382
A.6	Bootstrapping, Buildroot etc.	382
A.7	Der Bootprozess	382
A.8	Pointer und Strukturen in C	382
A.9	Das Kernel-Buildsystem und Treiber	382
A.10	Bücher	383
B	Belegung der GPIO-Ports	385
B.1	GPIO-Belegung gemäß wiringPi	385
B.2	GPIO-Belegung gemäß Broadcom	386
C	Safety und Security	387
C.1	Security	387

C.2	Safety	388
C.2.1	Maßnahmen in der Automobilindustrie.....	388
C.2.2	Umsetzung in der Programmierung	389
C.3	Ergänzende Literatur	390
D	Kopieren mit scp	391
E	Code::Blocks	393
E.1	Projekteinstellungen	393
E.1.1	Properties	394
E.1.2	Build options	395
	Stichwortverzeichnis	397