

Raspberry Pi Handbuch für Einsteiger

Michael Bonacina

2. Auflage: Februar 2018

© dieser Ausgabe 2019 by BMU Media GmbH

ISBN 978-3-96645-013-3

Herausgegeben durch:

BMU Media GmbH

Hornissenweg 4

84034 Landshut

Raspberry Pi Handbuch für Einsteiger

Inhaltsverzeichnis

| | | |
|-----------|---|-----------|
| 1. | Vorwort..... | 9 |
| 2. | Der Raspberry Pi - ein besonderer Embedded PC..... | 11 |
| 2.1. | Eine kreative Plattform zum Experimentieren..... | 12 |
| 2.2 | Bauweisen, Versionen und Modelle des Raspberry Pi..... | 13 |
| 2.3. | Ein mobiler SoC CPU als Prozessor | 15 |
| 2.4. | Universelle Schnittstellen für Multimedia und Automation..... | 15 |
| 2.5. | Unterstützte Speichermedien | 17 |
| 2.5.1. | Virtuelle Laufwerke durch Partitionen | 18 |
| 2.6. | Drahtlose und kabelgebundene Netzwerke..... | 18 |
| 2.7. | Stromversorgung und USB Netzteile..... | 19 |
| 3. | Betriebssysteme für den Raspberry Pi | 21 |
| 3.1 | Raspbian - Debian Linux für den Raspberry Pi..... | 22 |
| 3.2. | Spezialisierte Betriebssysteme für bestimmte Umgebungen | 23 |
| 3.2.1. | Android | 23 |
| 3.3. | Individuelle Linux Systeme - Arch und LFS | 26 |
| 3.4. | Ubuntu Mate und Snappy Core | 27 |
| 3.5. | Microsoft Windows 10 IoT für den Raspberry Pi 3..... | 28 |
| 3.6. | Installation von unterschiedlichen Betriebssystemen | 29 |
| 3.6.1. | NOOBS: Eine schnelle, komfortable Installation mit grafischer Oberfläche..... | 29 |
| 3.6.2. | Manuelle Installation mit Etcher, Win32DiskImager oder dd.... | 32 |
| 3.7. | Back-up und Wiederherstellung von Betriebssystemen mit fsarchiver | 34 |
| 4. | Einführung in Linux und Raspbian | 37 |
| 4.1. | Dateistruktur von Linux - feste Verzeichnisse für verschiedene Aufgaben | 38 |

| | |
|---|-----------|
| 4.1.1. Die wichtigsten Verzeichnisse für die Administration | 41 |
| 4.1.2. Laufwerke, externe Speichermedien und deren Einbindung ... | 42 |
| 4.2. Rechte und Nutzer | 48 |
| 4.2.1. Allgemeine und spezielle Nutzer | 48 |
| 4.2.2. Gruppen und deren Bedeutung | 49 |
| 4.3. Textbasierte und grafische Oberflächen | 51 |
| 4.3.1. Terminal, Konsole oder Kommandozeile | 52 |
| 4.3.2. Grafische Darstellung und Desktop Umgebungen | 53 |
| 4.4. Erste Schritte: Login und Konfiguration von Raspbian | 54 |
| 4.4.1. Das textbasierte Konfigurationstool raspi-config | 54 |
| 4.4.2. Das grafische Interface Raspberry Pi Configuration | 57 |
| 4.5. Software, Installation von Programmen und die Paketverwaltung .. | 58 |
| 4.5.1. PiPackages - ein grafischer Paketmanager | 59 |
| 4.5.2. apt-get - Pakete ohne grafische Oberfläche installieren | 60 |
| 4.6. Headless Server oder Pi als Desktop PC | 62 |
| 4.6.1 Fernzugriff für Headless Server - SSH und Webinterfaces | 62 |
| 4.7. Grafische Werkzeuge zur Administration von Servern | 62 |
| 4.8. Fehleranalyse und Logdateien | 67 |
| 5. LXDE - die Desktop Umgebung von Raspbian | 70 |
| 5.1. Konfiguration und individuelle Anpassung | 71 |
| 5.2. Optionale Software für mehr Funktionen | 74 |
| 5.3. Besonderheiten von LXDE | 76 |
| 5.4. Alternative Desktop Umgebungen | 77 |
| 6. Steuerung und Eingabe | 80 |
| 6.1. Konventionelle USB Geräte für den PC | 80 |

| | |
|--|------------|
| 6.2. Die Verwendung von Smartphones und Fernbedienungen | 81 |
| 6.3. Die Sprachsteuerung über Google Assistant, Amazon Alexa oder Apple Siri..... | 83 |
| 6.3.1. Voraussetzung für die Installation von Google Assistant..... | 83 |
| 6.3.2. Registrierung für die Verwendung von Alexa..... | 84 |
| 6.3.3. Wahl der geeigneten Software | 84 |
| 6.4. Netzwerkzugriff über unterschiedliche Protokolle..... | 85 |
| 6.4.1. Fernadministration über ein Terminal - SSH..... | 86 |
| 6.4.2. Grafische Oberflächen durch Webinterfaces | 89 |
| 6.5. Von der Tür bis zur Steckdose - RFID, Smartcard und Funk..... | 91 |
| 7. Schnittstellen und modulare Erweiterung | 92 |
| 7.1. Die Auswahl zusätzlicher Module..... | 92 |
| 7.2. Netzwerk und digitale Peripherie: LAN und USB Ports | 93 |
| 7.2.1. LAN und WLAN | 93 |
| 7.2.2. Externe Hardware über USB | 94 |
| 7.3. GPIO - eine universelle Schnittstelle nach außen | 95 |
| 7.3.1. Aufbau und Verwendung der GPIO Schnittstelle | 95 |
| 7.4. Displays, Touchscreens und Kameras - CSI und DSI Schnittstelle .. | 99 |
| 7.5. Automation und Steuerung externer Geräte..... | 100 |
| 7.5.1. Sensoren, I/O Erweiterung und IoT - die zuverlässige Erfassung von Daten..... | 100 |
| 7.5.2. Praktisches Beispiel: Die Steuerung über eine Fernbedienung | 101 |
| 7.5.3. Schaltkreise mit dem Raspberry Pi steuern..... | 105 |
| 8. Media Center, Server, Spielekonsole oder Smart Home - Der Raspberry Pi im Einsatz..... | 107 |
| 8.1. Network Attached Storage (NAS) - lokale und globale Netzlaufwerke | 107 |

| | |
|---|-----|
| 8.1.1. Wahl des Protokolls | 108 |
| 8.1.2. Installation und Einrichtung des Samba Servers..... | 109 |
| 8.1.3. Zugriffsrechte und Sicherheit bei Samba | 113 |
| 8.1.4. Grafische Hilfsmittel für die Konfiguration von Samba | 116 |
| 8.1.5. Network File Shares (NFS) - Netzwerkfreigaben für Linux.... | 118 |
| 8.1.6. FTP Server - Schnelle und komfortable Übertragung von Dateien | 121 |
| 8.1.7. Der Raspberry Pi als Private Cloud | 123 |
| 8.2. Videos und Ton - Einsatz als Media Player und HTPC | 126 |
| 8.2.1. Flexibel, benutzerfreundlich und leistungsfähig: Kodi (XBMC) | 128 |
| 8.2.2. Einrichtung und Add- ons..... | 128 |
| 8.2.3. OpenELEC/LibreELEC - Kodi als Betriebssystem | 130 |
| 8.3. Automation und IoT - Smart Home im Eigenbau | 132 |
| 8.3.1. Informationen und Steuerung - Grundlagen der Automation. | 132 |
| 8.3.2. Smart Home mit wenig Aufwand - Steuerung einer Funksteckdose | 133 |
| 8.3.3. Home Assistant: Ein konfigurierbarer IoT Server für dein Smart Home | 136 |
| 8.4. Privates Hosting: Ein Webserver im eigenen Haus | 143 |
| 8.4.1. Ein Webserver für das Hosting von Internetseiten | 144 |
| 8.5. IT Sicherheit und VPN - den Pi und das lokale Netz aus dem Internet nutzen..... | 154 |
| 8.5.1. Virtuelle Private Netzwerke (VPN) | 154 |
| 8.6. Der Raspberry Pi als Spielekonsole | 158 |
| 8.6.1. RetroPie..... | 158 |
| 8.6.2. Recalbox OS | 161 |
| 9. Grundlagen der Programmierung | 164 |

| | |
|---|------------|
| 9.1. Grundwissen für alle Programmiersprachen: Variablen, Bedingungen und Schleifen | 165 |
| 9.2. IDE, Entwicklungsumgebungen und andere Hilfsprogramme | 166 |
| 9.3. Shell Skripte - Programmieren für die Kommandozeile | 169 |
| 9.3.1. Arbeiten mit heruntergeladenen Bash-Skripten | 172 |
| 9.4. Python - eine flexible, übersichtliche und universelle Programmiersprache | 172 |
| 9.4.1. Variablen unter Python | 173 |
| 9.4.2. Die Entwicklungsumgebung IDLE und eine einfache Schleife | 174 |
| 9.4.3. GPIO mit Python - eine blinkende LED..... | 176 |
| 9.5. Weitere Sprachen: Perl, Java, C | 178 |
| 9.5.1. Perl - Flexible Skriptsprache mit vielen Anwendungen | 178 |
| 9.5.2. Java / JavaScript - Weit verbreitet, aber nicht verwandt | 179 |
| 9.5.3. C/C++ - Hochsprachen für maximale Effizienz | 180 |
| 9.6. Download von Quellcode - git und svn (Apache Subversion) | 182 |
| 9.7. Debuggen und Fehleranalyse | 184 |
| 10. Nachwort..... | 185 |

1. Vorwort

Der Raspberry Pi gehört ohne Zweifel zu den erfolgreichsten Computern, die bislang entwickelt wurden. Innerhalb von lediglich drei Jahren nach seiner Einführung verkaufte die dahinterstehende, gleichnamige Stiftung mehr als fünf Millionen Exemplare des kleinen Embedded PC. Damit entwickelte sie sich von einem Neueinsteiger zum absatzstärksten Computer-Hersteller in Großbritannien. Die für alle Beteiligten überraschend hohe Nachfrage war so groß, dass Kunden in den ersten Monaten teilweise wochenlang auf ihre Lieferung warten mussten. Erst nachdem die Produktionskapazitäten massiv erweitert wurden, war es der Raspberry Foundation möglich, den Rückstau abzubauen und ausreichend Lagerkapazitäten anzuschaffen.

Hinter diesem Erfolg steht nicht zuletzt die ungewöhnliche Vielseitigkeit des Mini Computers, die von Anfang an im Fokus der Entwickler stand. Mit seiner umfangreichen Auswahl an Schnittstellen eignet er sich für unterschiedlichste Aufgaben - vom IT Security Router über die Gebäudeautomation und dem Smart Home bis hin zu technischen Messungen, Robotersteuerung oder dem Einsatz als HTPC Mediacenter.

Angesichts der Möglichkeiten und des geringen Preises bildete sich in kürzester Zeit eine aktive und überaus produktive Community, die sich in weiten Teilen auf Open Source als Lizenzmodell stützt. Aus ihr sind unterschiedlichste Projekte für Soft- und Hardware

hervorgegangen, die für jeden Zweck Baupläne und die notwendigen Anwendungen zur Verfügung stellen. Für ein spezielles Vorhaben sind deshalb in der Regel keine umfangreichen Spezialkenntnisse notwendig, sondern lediglich die Freude am Experimentieren und die Bereitschaft, ein wenig Zeit in die Umsetzung zu investieren.

Mithilfe dieses Buches werden Sie schrittweise in die Geschichte, die Technik und die Möglichkeiten des Raspberry Pi eingeführt. Anhand konkreter Anleitungen lernen Sie einige der häufigsten Verwendungszwecke kennen und können diese sofort in die Praxis umsetzen. Es widmet sich gleichermaßen der Hard- und der Software und unterlegt theoretische Erklärungen anschaulich mit praktischen Beispielen. Diese führen Sie Schritt für Schritt durch die Installation, Montage und Konfiguration populärer Anwendungen.

2. Der Raspberry Pi – ein besonderer Embedded PC

Embedded PCs werden bereits seit den 70er Jahren für die Automation eingesetzt. Sie wurden zunächst in der Industrie und seit Mitte der 80er im Automobilbau - beispielsweise für die Motorsteuerung und die Fehleranalyse - eingesetzt. Es handelt sich um kompakte, effiziente und in der Regel sehr spezialisierte Computer, die in einer komplexen Umgebung „eingebettet“ (embedded) arbeiten. Über spezielle Schnittstellen steuern sie etwa die Bewegung von Robotern, zeichnen Messdaten auf oder optimieren bestimmte Prozesse. Im Gegensatz zu den späteren Heimcomputern, Desktop PC und Notebooks verwendet ein Embedded PC eine minimale, oft spezialisierte Hardware und verzichtet auf einen Bildschirm, Eingabemedien und eine direkte Nutzersteuerung.

Wegen ihrer Bauweise und ihrer Spezialisierung waren Embedded PCs vergleichsweise teuer und für private Nutzer nur eingeschränkt interessant. Die Softwareentwicklung und das Testen in unterschiedlichen Szenarien erfolgt oft über kostspielige Developer Kits, die einen Embedded PC mit erweiterter Auswahl an Schnittstellen und Umgebungen für die Programmierung zur Verfügung stellen. Diese Situation änderte der Raspberry Pi dramatisch, der sich als erster Embedded PC ausdrücklich auch an private Haushalte, Schüler, Studenten und experimentierfreudige Bastler richtete.

2.1. Eine kreative Plattform zum Experimentieren

Die Idee des Raspberry Pi entstand etwa 2005 an der Universität Cambridge in England. Obwohl der Bedarf an Informatikern stetig anstieg, sank das Interesse an dem Studiengang und die praktischen Kenntnisse der Studenten waren in den Augen der Professoren mangelhaft. Einen der Gründe sah die Universität in der Tatsache, dass Schülern und Studenten keine günstige Hardware zum Experimentieren zur Verfügung stand. Die teuren Embedded Systeme blieben ausgebildeten Experten an ihrer Arbeitsstelle vorbehalten und die eigenen Desktop PCs erwiesen sich entweder als ungeeignet oder zu wichtig, um damit potentiell kritische Projekte zu erproben.

Um diese Situation zu ändern, entwickelte eine Gruppe aus Professoren, Ingenieuren und erfahrenen Programmierern die Idee, einen sehr kostengünstigen Embedded Computer mit einer universell nutzbaren Auswahl an Schnittstellen anzubieten. Sowohl die Hardware wie auch das Betriebssystem und die Programme sollten nach Open Source lizenziert sein, so dass alle Komponenten einfach und ohne rechtliche Einschränkungen modifiziert werden können. Erste Prototypen erwiesen sich jedoch entweder als zu leistungsschwach oder zu kostspielig.

Erst mit der Verbreitung von Tablet PC und Smartphones standen Prozessoren zur Verfügung, die aufgrund der Massenproduktion und der technischen Entwicklung den Anforderungen an Preis und Leistung entsprachen. 2009 gründeten Mitglieder der Gruppe die

Raspberry Pi Foundation. August 2011 begann die Auslieferung erster Alpha Boards zu Testzwecken. Bereits vor der Serienreife nahm die Stiftung Bestellungen entgegen, um die Entwicklungs- und Produktionskosten zu finanzieren.

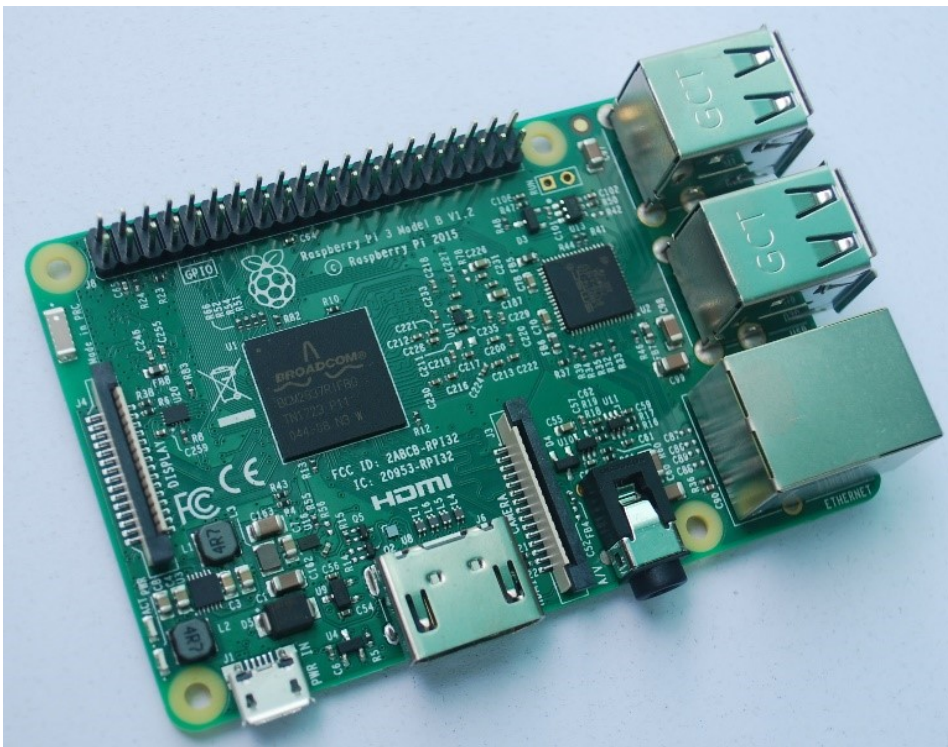
Die Nachfrage übertraf alle Schätzungen um ein Vielfaches und stieg nach Beginn der Auslieferung des überarbeiteten Testmodells im Herbst 2012 stetig an. Um die notwendigen Stückzahlen zu erreichen, musste die Produktion schon vorher auf eine weitere Fabrik in China ausgeweitet werden. Gleichzeitig erlaubten die hohen Bestellmengen eine Verbesserung der Hardware - allen voran die Erweiterung des RAM Speichers von 256 MB auf 512 MB.

2.2 Bauweisen, Versionen und Modelle des Raspberry Pi

Die erste Version des Raspberry Pi mit der Bezeichnung Model A wurde im Februar 2012 ausgeliefert und bereits wenige Monate später durch das Model B ersetzt. Der wichtigste Unterschied ist die Verdopplung des Arbeitsspeichers von 256 MB auf 512 MB RAM bei allen Geräten, die nach Oktober 2012 hergestellt wurden. Außerdem erweiterte die Version B die Anzahl der USB Schnittstellen von einer auf zwei. Die zweite Generation Pi 2 Model B kam im Februar 2015 in den Handel. Sie verwendet statt einem Single Core CPU einen deutlich leistungsstärkeren Quad Core Prozessor, wodurch die Performance deutlich verbessert wurde.

Seit dem Februar 2016 ist auch die dritte Generation erhältlich, die zwar denselben Prozessor einsetzt, dessen Taktfrequenz aber von

900 MHz auf 1200 MHz erhöht. Außerdem besitzt der Raspberry Pi 3 Model B erstmals einen integrierten Chip für WLAN und Bluetooth. Wegen seiner Unterstützung für Betriebssysteme mit 64 Bit, seiner Performance und der besseren Vernetzung solltest du in jedem Fall dieses Modell bevorzugen.



Raspberry Pi 3 mit 1 GB RAM und 1,2 GHz Quad Core CPU

Neben dem konventionellen Raspberry Pi bietet die Stiftung auch Compute Module aller Versionen an. Dabei handelt es sich um kleine Steckkarten, die für den Einbau beispielsweise in Steueranlagen ausgelegt sind und selbst über keinerlei Anschlüsse verfügen. Sie eignen sich wegen ihrer extrem kompakten Bauweise von der Größe eines Speicherriegels für fertige Embedded Systeme, benötigen aber ein zusätzliches Trägerboard als Schnittstelle für die Stromversorgung und die I/O.

2.3. Ein mobiler SoC CPU als Prozessor

Wie Smartphones und Tablet PC verwendet der Raspberry Pi einen Prozessor mit einer ARM Architektur. Dabei handelt es sich um einen Befehlssatz, der besonders energieeffizient arbeitet und sich grundlegend von demjenigen in Desktop Computern oder Notebooks unterscheidet. Wie nahezu alle ARM Computer ist der Pi ein System-on-a-Chip (SoC) System, bei dem bereits alle wesentlichen Funktionen wie Netzwerk Controller, GPIO oder USB Port in dem Prozessor integriert sind.

Ein LPDDR2 RAM dient bei allen Versionen des Raspberry Pi als Arbeitsspeicher. Je nach Modell beträgt die Kapazität zwischen 256 MB und 1 GB. Der Speicher wird sowohl von der Grafik wie auch vom Betriebssystem genutzt, so dass sich bei datenaufwendigen Anwendungen und gleichzeitigem Multitasking Engpässe ergeben können. Da es sich um ein Onboard Modul - also einen fest verlöteten Chip handelt - kannst du die Kapazität nicht wie bei einem Desktop PC oder einem Server erweitern. Deshalb solltest du einen Raspberry Pi grundsätzlich für eine spezielle Aufgabe verwenden und die Performance auf diesen Zweck optimieren.

2.4. Universelle Schnittstellen für Multimedia und Automation

Obwohl der Raspberry Pi ursprünglich als Plattform für die Automation und für industrielle und technische Anwendungen

entwickelt wurde, besitzt er ebenfalls Schnittstellen für die Ausgabe von Multimedia. Durch seine integrierte Hardwarebeschleunigung für 2D/3D Grafik, unterschiedliche Video Codecs und eine Full HD 1080p Auflösung ist er in der Lage, Inhalte inklusive hochauflösende Videos flüssig darzustellen.

Für die Kommunikation mit externer Hardware - beispielsweise einer Tastatur oder zusätzlichen Laufwerken - nutzt der Pi wie die meisten Computer USB. Leider unterstützt auch die neueste Version 3 immer noch nur den älteren Standard 2.0, der eine maximale Stromstärke von 0,5 Ampere und eine Bandbreite von maximal 480 Mbit/s - das entspricht etwa 40 MB/s - ermöglicht. Wie du später noch sehen wirst, kann sich dies unter Umständen als kritisch erweisen. Zu beachten ist außerdem, dass sich alle USB Geräte technisch einen einzigen Port und somit dessen Datentransfer teilen - der vierfache Anschluss ist in Wahrheit ein Hub, der die Anbindung lediglich auf mehrere Stecker verteilt.

Die Ausgabe von multimedialen Inhalten erfolgt über Grafikports in HDMI für konventionelle Monitore und DSI für kleine Bildschirme und Touch Screens. Für Ton stellt der Raspberry einen Audioausgang in 3.5 mm Klinke, wie er für Kopfhörer genutzt wird. Ein CSI Eingang für eine Kamera rundet die Auswahl ab. Bei vielen Anwendungen wird der Raspberry Pi jedoch als sogenannter Headless Server betrieben - das bedeutet, er besitzt selbst weder ein Display noch eine Eingabemöglichkeit, sondern wird ausschließlich über das Netzwerk gesteuert.

Die universellste Schnittstelle ist die GPIO Leiste, die der Raspberry Pi für eine Vielzahl von Sensoren oder die Steuerung nutzen kann. Sie umfasst ab dem Raspberry Pi 1 Modell B+ 40 Pins in einer Doppelreihe. Von diesen dienen 26 als GPIO In- oder Output, weitere stellen eine Ausgangsspannung mit 3.3 Volt oder 5 Volt Gleichstrom oder den negativen Pol (Ground, Erdung) des Stromkreises. Das Besondere der GPIOs ist, dass diese frei programmiert werden können. Du kannst einen Pin nach Belieben als Ein- oder Ausgang verwenden, um Messwerte von Sensoren zu empfangen, über ihn als Aktor Schaltkreise steuern oder digitale Daten senden.

2.5. Unterstützte Speichermedien

Der Raspberry Pi unterstützt mit seiner Hardware keine Festplatten oder SSD, wie sie in Desktop PC oder Notebooks zum Einsatz kommen. Die dafür üblichen Standards SATA, mSATA und das neuere M.2 können nur über den Umweg eines separaten Controllers genutzt werden. Dies geschieht in der Regel als USB Laufwerk.

Lediglich einige Compute Module besitzen ein internes Laufwerk in Form eines eMMC Flash Speichers, wie er bei Smartphones und Tablet PCs zum Einsatz kommt. Als Speichermedium für das Betriebssystem dient dem Raspberry Pi deshalb eine konventionelle Micro SD Karte. Im praktischen Umgang hat das für dich zwei Konsequenzen: Erstens benötigst du eine sehr schnelle Karte, die mindestens der Klasse 10 entsprechen sollte, um die

Leistung nicht durch einen langsamen Zugriff zu beeinträchtigen. Zweitens besitzen SD Karten nur eine gewisse Lebenszeit und neigen zu spontanem Versagen. Deshalb solltest du sicherheitshalber regelmäßig ein Back-up deines Systems erstellen und auf einem sicheren Medium lagern.

2.5.1. Virtuelle Laufwerke durch Partitionen

Ein Laufwerk bietet die Möglichkeit, es in verschiedene, voneinander getrennte Bereiche - die Partitionen - aufzuteilen. Diese werden wie unabhängige Speichermedien behandelt und können mit unterschiedlichen Dateisystemen formatiert werden. Der Raspberry Pi verwendet mehrere Partitionen, um etwa die Daten für den Bootvorgang vom System und den Nutzerdaten zu trennen. So lässt sich beispielsweise ausschließen, dass ein Nutzer den Speicherplatz des Betriebssystems vollständig belegt und dessen Stabilität beeinträchtigt.

2.6. Drahtlose und kabelgebundene Netzwerke

Bis zum Model B besaß der Mini Computer in seiner Basisausstattung keine Unterstützung für drahtlose Netzwerke. Sowohl WLAN wie auch Bluetooth mussten über USB Module nachgerüstet werden, was einerseits den Stromverbrauch erhöht und andererseits die Bandbreite des USB Ports belastet. Der Raspberry Pi 3 verwendet hingegen einen integrierten Chip für WLAN 802.11 b/g/n und Bluetooth 4.1 und bietet Anschlüsse für externe Antennen. Du kannst den Raspberry Pi sowohl als Client in

einem bestehenden Netzwerk wie auch als Wireless Access Point einsetzen, mit dem sich die anderen Geräte verbinden.

Alle Versionen besitzen als primären Netzwerkanschluss einen Ethernet LAN Port mit einer Transferrate von 10/100 Mbit/s, was etwa 12,5 MB/s inklusive den zusätzlichen Protokoll Daten (Overhead) entspricht. Bei dem Anschluss handelt es sich um einen gewöhnlichen RJ45 Stecker, wie du ihn mit großer Wahrscheinlichkeit von einem LAN Kabel kennst. Die Konfiguration des Netzwerks - unabhängig ob als Server oder Client - erfolgt über das Betriebssystem.

2.7. Stromversorgung und USB Netzteile

Die Stromversorgung des Raspberry Pi gehört zu einer der am häufigsten unterschätzten Fehlerquellen des Gerätes. Grundsätzlich liefert ihm ein Micro USB Eingang die notwendige Spannung. Seit vielen Jahren wird diese Schnittstelle auch von Smartphones verwendet, so dass du wie viele andere Nutzer erst einmal versucht sein wirst, einfach ein konventionelles Ladegerät als Quelle zu benutzen.

Aus einfachen, technischen Gründen ist dies jedoch nur in Ausnahmefällen ratsam. Zunächst einmal liefern viele Ladegeräte nicht ausreichend Leistung für einen stabilen Betrieb. Sie besitzen eine Stromstärke deutlich unter 1 A, während der Raspberry Pi bereits ohne jede zusätzliche Hardware bis zu 1,2 A benötigt. Jedes

angeschlossene USB Gerät kann den Verbrauch um weitere 0,5 A erhöhen. Deshalb ist es grundsätzlich empfehlenswert, dass du einen USB-Hub mit separater Stromversorgung verwendest. Dies gilt besonders bei der Verwendung mehrerer USB Geräte und solchen mit einer hohen Spannungsaufnahme - beispielsweise externen Festplatten.

Ein weiteres Problem ist, dass Ladegeräte schlicht nicht dafür ausgelegt sind, dauerhaft eine stabile Spannungsversorgung zu gewährleisten. Die Stromaufnahme wird von der Ladeautomatik eines Gerätes geregelt, die der Raspberry Pi nicht besitzt - ihm dient der Eingang schließlich direkt für die Stromzufuhr. Bei Schwankungen kann die kritische Grenze schnell unterschritten werden, was zu einem vollständig unvorhersehbaren Verhalten führt. Es ist sehr schwer bis unmöglich, solche Fehler zweifelsfrei auf die Stromversorgung zurückzuführen.

Bei dem Raspberry Pi 3 kommt noch ein besonderer Umstand hinzu: Einige Modelle leiden unter einem Konstruktionsfehler, für den das Design und Bauteile einer bestimmten Serie verantwortlich sind. Er führt dazu, dass die Spannung zwischen Eingang und Prozessor um bis zu 0,6 Volt abfällt. Selbst bei einem geeigneten Netzteil arbeitet der Computer deshalb bereits dauerhaft mit Unterspannung. Um diesen Effekt zu kompensieren, liefert das offizielle Netzteil eine leicht erhöhte Spannung von 5,2 Volt. Wichtig für dich ist, dass von diesem Fehler glücklicherweise nur ein kleiner Teil der Produktion betroffen ist. Du hast also gute Chancen, keines dieser „Montagsfabrikate“ zu erhalten.

3. Betriebssysteme für den Raspberry Pi

Insgesamt stehen dir etwa 40 unterschiedliche Betriebssysteme zur Auswahl, die du auf dem Raspberry Pi installieren kannst. Die überwiegende Mehrzahl davon sind unterschiedliche Distributionen und Varianten von Linux, doch auch Microsoft bietet mittlerweile ein Windows 10 IoT an, das für den Raspberry Pi 3 entwickelt wurde. Viele der angebotenen Systeme werden allerdings nicht regelmäßig aktualisiert, eignen sich lediglich für sehr spezielle Anwendungen oder erfordern ein entsprechendes Fachwissen für die Installation.

Neben „gewöhnlichen“ Linux Systemen gehören auch viele spezialisierte Versionen zu der Auswahl. So kannst du beispielsweise beim Einsatz als Media Center beziehungsweise HTPC die Performance durch OpenELEC optimieren oder über das Echtzeit-Betriebssystem (Real Time Operation System, RTOS) ChibiOS/RT Prozesse steuern, die eine extrem niedrige Reaktionszeit voraussetzen.

Die meisten spezialisierten Betriebssysteme richten sich allerdings ausschließlich an erfahrene Entwickler und bieten im Unterschied zu den populären Distributionen keine vorgefertigten Installationen (Images) an. Wegen ihres geringen Nutzerkreises sind einige Funktionen und viele Verwendungen zudem nur unzureichend dokumentiert. Häufig verwendete Betriebssysteme wie Rasbian besitzen hingegen eine aktive und mitgliederstarke Community, die dir bei Problemen schnell und bereitwillig hilft, sofern du sie freundlich um Unterstützung bittest.

3.1 Raspbian – Debian Linux für den Raspberry Pi

Das Betriebssystem Raspbian ist eine Variante von Debian Linux und für dich als Einsteiger besonders geeignet. Es wird seit 2012 permanent weiterentwickelt und bietet neben einem leichten Einstieg eine sehr vielseitige Auswahl an Programmen, die sich direkt aus offiziellen, sicheren und überprüften Quellen installieren lassen.

Das aktuelle Raspbian beruht auf der Debian Version 9 mit dem Codenamen Stretch, die im Juni 2017 als Stable Release veröffentlicht wurde. Das bedeutet, dass alle Pakete intensiv getestet wurden und regelmäßig mit Updates versorgt werden, die beispielsweise Sicherheitslücken schließen oder Programmierfehler (Bugs) beseitigen. Raspbian steht dir auf der Homepage des Raspberry Projekts unter <https://www.raspberrypi.org/downloads/> als freier Download ohne Kosten oder Registrierung zur Verfügung.



Die Logos von Raspberry Pi und Linux Debian dienen als Symbole für Raspbian

3.2. Spezialisierte Betriebssysteme für bestimmte Umgebungen

Wahrscheinlich bist du daran interessiert, auch andere Betriebssysteme zu testen. Wenn du vorher ein Back-up anfertigst, kannst du deinen Versuchen freien Lauf lassen und anschließend innerhalb weniger Minuten den Ausgangszustand wieder herstellen.

3.2.1. Android

Schon mit dem Erscheinen des Raspberry Pi Model 1 begannen die Versuche, Android auf den neuen Embedded PC zu portieren. Dabei zeigten sich jedoch teilweise erhebliche Probleme, die Hardware effizient zu nutzen. Einige der ersten Anläufe stammen von der Entwicklergruppe Razdroid, die unter anderem Android 2.3 und 4.4.2 auf dem Raspberry Pi 1 installierten. Wegen der fehlenden Hardwarebeschleunigung sind diese Systeme jedoch für einen praktischen Einsatz nahezu unbrauchbar.

Falls du wirklich Android auf dem Raspberry nutzen möchtest, solltest du deshalb einen Pi 2 oder besser Pi 3 und das Betriebssystem RaspAnd einsetzen. Es bietet unter anderem eine Full HD Auflösung, unterstützt den offiziellen 7“ Touch Screen und ist mit dem Google Play Store kompatibel. Leider ist der Download nicht gratis, sondern kostet etwa 9 US-Dollar.

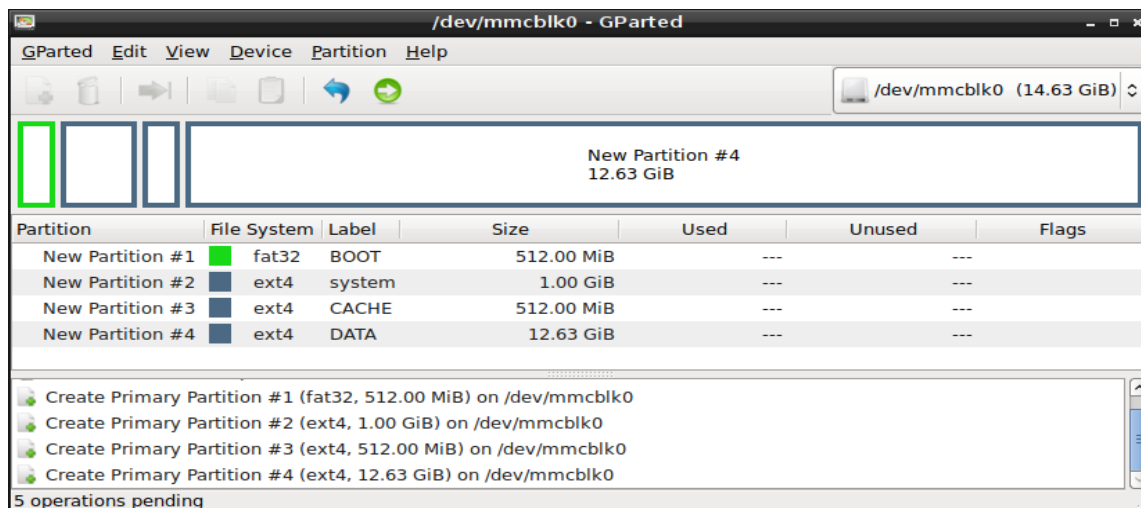
3.2.1.1. Installation von RaspAnd

Grundsätzlich solltest du für die Installation ein Linux Betriebssystem nutzen. Es existiert zwar ein Windows Installer, der jedoch noch in der Testphase (Beta) steckt und unter anderem den Speicherplatz für die Nutzerdaten auf nur 5 GB begrenzt. Falls du kein Linux nutzt, kannst du es in einer virtuellen Maschine etwa unter VMWare oder VirtualBox laufen lassen, ohne es direkt auf deinem Computer installieren zu müssen. Für beide Programme existieren zahlreiche vorgefertigte Linux-Images, die du im Internet herunterladen und direkt verwenden kannst.

Für die Installation von Android lade zunächst den aktuellen RaspAnd Build (http://raspex.exton.se/?page_id=25) herunter. Anschließend startest du Linux und rufst mit `sudo gparted`

den grafischen Partitionsmanager auf. Vergewissere dich, dass deine Micro SD Karte als Laufwerk angezeigt wird und erstelle die folgenden vier Partitionen in exakt dieser Reihenfolge:

- Boot mit 512 MB im Dateisystem FAT32
- system mit 1024 MB als Ext4
- CACHE mit 512 MB als Ext4
- DATA als Ext4 mit dem restlichen Speicherplatz für Nutzerdaten



Partitionen für RaspAnd in GParted

Wenn du die Partitionen geschrieben hast, merke dir die Laufwerksbezeichnung der SD-Karte, markiere die Partition BOOT mit der Boot Flag (Partition → Manage Flags), schreibe die Änderung und schliesse gparted. Anschließend öffne ein Terminal und kopiere mit

```
sudo dd if=/Pfad/zum/entpackten/Download/system.img
of=/dev/sdX2 bs=1M
```

das System auf die Partition. Den Pfad und die Laufwerksbezeichnung sdX musst du natürlich entsprechend deiner Umgebung anpassen. Bei vielen Systemen werden SD Karten auch als /dev/mmcblkYpX eingebunden - in diesem Fall bezeichnet Y die Anzahl der Karten (beginnend mit 0), gefolgt von pX für die Partition. Bei einer SD Karte wäre die korrekte Adresse also zum Beispiel /dev/mmcblk0p2. Als letzten Schritt kopierst du noch alle Daten aus dem entpackten Verzeichnis boot/ auf die Boot Partition 1 der SD

Karte, fährst Linux herunter und kannst anschließend RaspAnd auf deinem Raspberry Pi booten.

3.3. Individuelle Linux Systeme – Arch und LFS

Wenn du einen besonderen Wert auf Effizienz legst, den hohen Arbeitsaufwand nicht scheust und obendrein eine Menge über Linux lernen möchtest, kannst du dir ein eigenes System erstellen. Dafür bietet sich Arch Linux an, während Hardcore-Programmierer mit Linux-from-the-Scratch (LFS) das Betriebssystem vollständig neu und in Eigenarbeit aus dem Quellcode zusammenstellen und kompilieren. Diese Möglichkeit ist für Entwickler interessant, erfordert aber ein sehr hohes Know-how.

Bei LFS startest du vollständig von Null und musst zuerst eine lauffähige Umgebung in einem Emulator erstellen. Im Gegensatz zu gewöhnlichen Linux Distributionen existieren keine fertigen Pakete, die dir fertige Programme liefern. Stattdessen musst du alle Komponenten und Anwendungen als Quellcode herunterladen und selbst kompilieren. Für den Raspberry Pi existiert das PiLFS Projekt (<http://intestinate.com/pilfs/>), das notwendige Konfigurationen und Anleitungen dafür anbietet.

Arch Linux ist eine Distribution, die sich nach dem KISS Prinzip orientiert. Dieses steht für maximale Übersichtlichkeit und einfachen Aufbau ohne überflüssige Funktionen. Aus diesem Grund bietet Arch auch keine grafischen Interfaces für die Konfiguration, die ausschließlich über geschriebene Befehle erfolgt.

3.4. Ubuntu Mate und Snappy Core

Auf Desktop PC ist Ubuntu wegen seiner Benutzerfreundlichkeit wahrscheinlich die populärste Linux Distribution. Sie beruht auf Debian, legt jedoch eine besondere Priorität auf leichte Bedienbarkeit und aktuelle Software durch häufige Aktualisierungen. Für den Raspberry Pi stehen zwei unterschiedliche Versionen betriebsbereit zur Verfügung, aus denen du die geeignete wählen kannst.

Ubuntu Mate entspricht dem klassischen Ubuntu Desktop System und verwendet denselben Paketmanager und Anwendungen aus den offiziellen Quellen für ARMHF Prozessoren. Es verwendet die effiziente grafische Oberfläche Mate als grafisches Benutzerinterface, um den Ressourcengebrauch niedrig zu halten und gleichzeitig eine vertraute, intuitive Bedienung zu ermöglichen.

Snappy Ubuntu Core verzichtet im Gegensatz zu Mate vollständig auf eine grafische Oberfläche und ist für Embedded Devices, IoT Anwendungen und Cloud Computing optimiert. Die Steuerung erfolgt nahezu ausschließlich über geschriebene Befehle in einem Terminal. Der Zugriff erfolgt in der Regel über das Netzwerk oder Konfigurationsseiten, die über den Browser aufgerufen werden können. Snappy verwendet ein eigenes Paketformat, das auf Web-Apps zugeschnitten ist.

Beide Varianten lassen sich sehr einfach von einem Ubuntu System über die integrierte Laufwerksverwaltung installieren. Dazu musst

du nur die entsprechende Image Datei von der offiziellen Ubuntu Homepage herunterladen, deine SD Karte auswählen und Wiederherstellung aus einer Image Datei aus dem Menü aufrufen.

3.5. Microsoft Windows 10 IoT für den Raspberry Pi 3

Wegen der ARM Architektur des Prozessors ist der Raspberry Pi nicht mit konventionellen Microsoft OS Versionen wie Windows XP, 7 oder 10 kompatibel. Seit Sommer 2015 bietet der Konzern jedoch Windows 10 Core IoT für ausgewählte Embedded Computer an - darunter den Raspberry Pi 2 und 3 sowie einige ebenfalls populäre Development Boards. Du kannst das Angebot kostenlos und ohne Registrierung nutzen, falls du einen PC mit Windows 10 in demselben LAN Netzwerk betreibst. Im Vergleich zu Linux als Betriebssystem birgt die Spezialisierung auf IoT Anwendungen jedoch bestimmte Einschränkungen.

Zunächst einmal eignet sich dieses Betriebssystem nicht für multimediale Anwendungen wie einen HTPC. Darüber hinaus sind einige Werkzeuge für die Entwicklung von IoT Applikationen nur in Verbindung mit einer Registrierung bei Microsoft zugänglich. Entscheidend für dich dürfte aber sein, dass das Angebot an betriebsbereiter Software für viele private Zwecke nur unzureichend ist. Das Betriebssystem ist in erster Linie darauf ausgelegt, konkrete Projekte nachzubauen oder eigene IoT Applikationen zu entwickeln.

3.6. Installation von unterschiedlichen Betriebssystemen

Beim Start eines Raspberry Pi sucht der im Prozessor integrierte Boot Loader nach einem geeigneten Betriebssystem auf der Speicherkarte. Wird er fündig, lädt er die weiter benötigten Daten und startet anschließend das Betriebssystem.

Das Booten von einem USB Laufwerk ist bislang nur experimentell auf einem Pi 3 möglich und befindet sich in einem frühen Entwicklungsstadium. Da für diesen Zweck bisher auch noch der Chip irreversibel umprogrammiert werden muss, solltest du lieber warten, bis ein sicheres Verfahren gefunden wurde.

Die SD Karte wird vom Raspberry Pi ausschließlich für die erste Etappe des Bootvorgangs benötigt. Es ist deshalb problemlos möglich, das Betriebssystem selbst auf ein externes Laufwerk auszulagern, solange beim Start eine SD Karte mit den notwendigen Parametern existiert.

3.6.1. NOOBS: Eine schnelle, komfortable Installation mit grafischer Oberfläche

NOOBS ist die Abkürzung für New Out Of The Box Software. Es handelt sich um einen im Sommer 2013 veröffentlichten Installer für den Raspberry Pi, mit dem ein Nutzer ohne technische Kenntnisse ein Betriebssystem direkt über das Gerät aufspielen kann. NOOBS unterstützt ein breites Spektrum an unterschiedlichen Betriebssystemen, die es beim ersten Start zur Wahl stellt. Für dich als Einsteiger ist NOOBS das geeignete Mittel, um ohne manuelle

Konfiguration deinen Raspberry Pi in kurzer Zeit betriebsbereit zu machen.

Du kannst entweder eine bereits vorbereitete NOOBS SD Karte kaufen oder dir selbst eine erstellen. Dazu musst du zunächst einmal NOOBS von der offiziellen Raspberry Pi Seite (<https://www.raspberrypi.org/downloads/noobs/>) herunterladen. Du kannst dich entweder für die vollständige Offline Installation NOOBS mit rund 1.5 GB oder den kleinen Network Installer Lite mit rund 32 MB entscheiden. Erstere Möglichkeit bringt nur dann einen Vorteil, wenn du Raspbian auf einem Pi ohne oder mit sehr langsamem Internetanschluss installieren möchtest. Ansonsten kannst du ohne Nachteile die Lite Version wählen - alle Komponenten werden je nach Bedarf nachgeladen.

Zunächst benötigst du eine leere SD Karte mit einer Kapazität von mindestens 8 GB. Diese formatierst du mit dem Dateisystem FAT32. Unter Linux kannst du gparted oder die Laufwerkverwaltung benutzen, um den gesamten Speicher als primäre Partition zu deklarieren und diese anschließend zu formatieren. Unter Windows ist es wichtig, nicht den Dateimanager zu verwenden, da dessen Methode der Formatierung leider zu Problemen beim Boot führen kann. Eine geeignete und vielfach erprobte - auch vom Hersteller empfohlene - Alternative ist das Programm SD Formatter (https://www.sdcard.org/downloads/formatter_4/index.html).

Ab einer Größe von 64 GB erstellt dieses jedoch automatisch ein exFAT Dateisystem, auf dem NOOBS nicht lauffähig ist. Du kannst

das Format nachträglich ändern, indem du das kleine Programm Guiformat (<http://www.ridgecrop.demon.co.uk/guiformat.htm>) einsetzt. Öffne es einfach ohne vorherige Installation, wähle deine mit dem SD Formatter vorbereitete Speicherkarte und es formatiert diese unabhängig der Größe mit FAT32.

Um NOOBS zu benutzen, musst du lediglich das entsprechende Zip-Paket herunterladen und die enthaltenen Dateien auf deine formatierte SD Karte kopieren. Dabei müssen sich die Dateien direkt auf dem untersten Verzeichnispfad befinden - sollte beim Entpacken automatisch ein Ordner mit dem Namen des Archivs erzeugt worden sein, kopierst du dessen Inhalt.

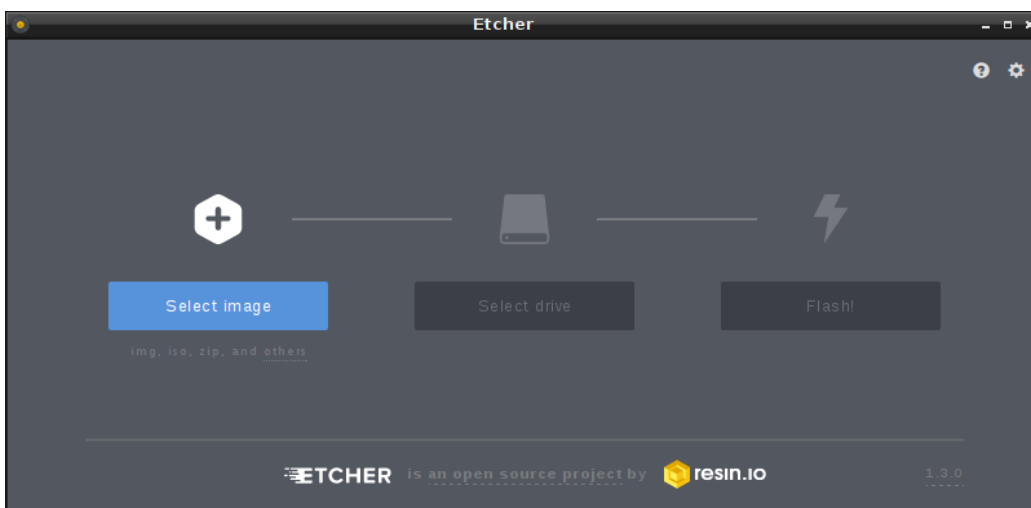
NOOBS verwendet ein grafisches Interface - aus diesem Grund verbinde deinen neuen Raspberry Pi über HDMI mit einem TV oder einem Monitor und schließe eine Maus, ein Keyboard sowie nach Möglichkeit ein Netzkabel an. Nun ist der Pi startbereit und nach einem kurzen Bootvorgang kannst du dein gewünschtes Betriebssystem im Menü auswählen.

Anmerkung:

Wenn du kein Lesegerät für Micro SD Karten besitzt, kannst du notfalls auch ein Android Smartphone einsetzen. Formatiere zuerst die Karte und kopiere anschließend die NOOBS Dateien von deinem Telefon oder deinem Computer direkt in das unterste Verzeichnis der Karte. Diese Methode wird offiziell nicht unterstützt und ist deshalb auch nicht empfehlenswert - in der Regel funktioniert sie aber ohne Probleme.

Mit NOOBS kannst du auch mehrere Betriebssysteme gleichzeitig auf deinem Raspberry Pi installieren. Leider besitzt es jedoch einen gravierenden Nachteil: Nach der Installation sind keine weiteren Änderungen mehr möglich, ohne dass die gesamte SD Karte gelöscht und neu geschrieben wird. Diese Einschränkung kannst du mit dem in Abschnitt 3.7. beschriebenen Vorgehen zur Erstellung und Wiederherstellung von Back-ups umgehen.

3.6.2. Manuelle Installation mit Etcher, Win32DiskImager oder dd
Möchtest du auf NOOBS verzichten, dann stehen dir mehrere Alternativen offen. Die bequemste ist die Anwendung Etcher (<https://etcher.io/>), die für Mac OS, Linux und Windows angeboten wird. Sie schreibt ein Imagefile direkt auf eine SD Karte und erzeugt dabei gleich die notwendigen Partitionen. Da die Software darüber hinaus auch mit Zip komprimierte Archive verarbeitet, kannst du die meisten Betriebssysteme direkt nach dem Download ohne weitere Schritte installieren.



Etcher schreibt Betriebssysteme unter Windows, Linux und OS X direkt auf eine Speicherkarte

Unter Linux oder Mac OS X steht dir zudem das sehr hilfreiche Tool `dd` zur Verfügung, das zur Standardinstallation zählt. Es schreibt Daten unmittelbar ohne weitere Verarbeitung in eine Datei, auf ein Laufwerk oder in diesem Fall auf deine SD Karte. Du musst in jedem Fall sorgfältig überprüfen, dass es sich bei dem unter `of=` angegebenen Ziel um deine SD Karte handelt. Andernfalls kann der Befehl ohne weitere Nachfrage den Inhalt jedes falsch angegebenen Laufwerks löschen und sogar das System zum Absturz bringen. Der entsprechende Befehl in einem Terminal lautet:

```
sudo dd if=/Verzeichnis/Name.img of=/dev/SDKARTE bs=4M  
conv=fsync
```

Um dieses Verfahren zu verwenden, musst du allerdings folgende Bedingungen berücksichtigen. Die Operation überschreibt sämtliche Daten einschließlich einer bestehenden Partitionstabelle auf der SD Karte. Deshalb darf diese niemals in das System eingebunden sein. Zunächst einmal musst du den Namen zweifelsfrei feststellen - bei Linux kann dies entweder `/dev/sdX` oder `/dev/mmcbkX` sein. Mac OS X verwendet für Speicherkarten hingegen in der Regel `/dev/diskX` oder `/dev/rdiskX`. Nutze am besten eine Oberfläche wie die Laufwerksverwaltung bei Ubuntu oder die Systemübersicht bei Apple Computern (Über Diesen Computer → Weitere Info → USB/Kartenleser), um die exakte Bezeichnung zu ermitteln. Als Ziel musst du stets die gesamte Karte angeben - beispielsweise `sdX`, `mmcbkx` oder `diskX` - alle weiteren Informationen beziehen sich auf zusätzliche Partitionen, die gelöscht werden.

Für Windows steht mit dem Win32DiskImager (<http://sourceforge.net/projects/win32diskimager/>) eine weitere Lösung zur Verfügung, die Betriebssysteme direkt auf deine SD Karte übertragen kann. Diese müssen dazu entpackt vorliegen. Wie bei Linux ist auch hier in jedem Fall große Sorgfalt bei der Auswahl des Zieles geboten. Alle Schreibzugriffe finden auf einer Ebene statt, bei der das Betriebssystem keine Warnung über etwaige Fehlkonfigurationen ausgibt.

3.7. Back-up und Wiederherstellung von Betriebssystemen mit fsarchiver

Wenn du neue Betriebssysteme testen möchtest, wirst du in jedem Fall in der Lage sein wollen, die vorherige Installation schnell in ihrem aktuellen Zustand wiederherstellen zu können. Auch aus Sicherheitsgründen ist es ratsam, regelmäßig Back-ups anzufertigen, die auf einem separaten Speichermedium wie einer USB Festplatte gelagert werden sollten. Ein geeignetes Programm dafür ist fsarchiver, das sich bei allen Linux Distributionen als Paket installieren lässt. Es ermöglicht, den gesamten Inhalt deiner SD Karte auszulesen, komprimiert zu speichern und bei Bedarf wieder zu schreiben. Wie viele Linux Anwendungen verfügt es allerdings über keine grafische Oberfläche, sondern ist ausschließlich über die Kommandozeile zu steuern. Diesen Nachteil gleicht es durch seine enorme Effizienz und Leistungsfähigkeit jedoch mehr als wieder aus.

Um fsarchiver zu nutzen, musst du deinen Raspberry herunterfahren und die Speicherkarte in ein Lesegerät einlegen, auf das ein Linux Betriebssystem zugreifen kann. Anschließend ermittelst du deren aktuelle Bezeichnungen etwa mit dem Befehl

```
sudo blkid
```

der dir alle angeschlossenen Laufwerke anzeigt. Stelle über

```
sudo mount
```

sicher, dass die Speicherkarte nicht eingebunden ist. Falls doch, kannst du die betreffenden Partitionen entweder mit einem Dateimanager, der Laufwerksverwaltung oder

```
sudo umount /dev/mmcblkXpY
```

beziehungsweise

```
sudo umount /dev/sdXY
```

freigeben. Durch den Befehl

```
sudo fsarchiver savefs /mein/Backup/Verzeichnis/raspberry-OS.fsa /dev/KartePartition1 /dev/KartePartition2
```

speicherst du anschließend eine oder mehrere Partitionen in einer einzigen Datei ab. Der Vorteil von fsarchiver liegt unter anderem darin, dass es nur die verwendeten Daten ausliest und das Back-up gleichzeitig komprimiert. Je nach freiem Speicherplatz und Art der Daten benötigt es deshalb für den Inhalt einer Speicherkarte nur ein Minimum ihrer eigentlichen Größe.

Die Wiederherstellung erfolgt wahlweise für den gesamten Speicherbereich oder auch gezielt für einzelne Partitionen. Dazu musst du lediglich

```
sudo fsarchiver restfs /mein/Backup/Verzeichnis/raspberry-  
OS.fsa id=0,dest=/dev/KartePartition1  
id=1,dest=/dev/KartePartition2
```

eingeben. Die Kennziffer `id=` bezeichnet dabei die Position der Partition in der Sicherung, `dest=` die Partition, in die diese geschrieben wird.

Mit diesem Vorgehen ist es dir nun möglich, gezielt einzelne Bereiche zu sichern und innerhalb weniger Minuten die SD Karte vollständig oder gezielt Partitionen wie BOOT oder das System zu rekonstruieren.

4. Einführung in Linux und Raspbian

Über Linux sind viele Gerüchte im Umlauf - einige entsprechen den Tatsachen, andere sind frei erfunden und die Mehrzahl schon seit Jahren veraltet. Grundsätzlich bezeichnet der Begriff Linux im eigentlichen Sinn kein Betriebssystem, sondern einen Kernel. Die Aufgabe eines Kernels besteht darin, eine universelle Schnittstelle anzubieten, über die Programme unabhängig von der individuellen Hardware mit dieser kommunizieren. Im Alltag hat der Nutzer wenig mit dem Kernel zu tun, sondern arbeitet ausschließlich mit den darunter angesiedelten Programmen.

Als Betriebssystem wird Linux in unterschiedlichen Distributionen angeboten, die jeweils eine eigene Priorität auf Stabilität, Performance, Einfachheit oder Benutzerfreundlichkeit legen. Teilweise sind sie zusätzlich auf eine besondere Verwendung wie IT Sicherheit, schnelle Reaktionszeiten oder HTPC Media Center spezialisiert. Eine der populärsten Distributionen ist Debian, das als Grundlage für Raspbian dient. Es wird häufig auf Servern eingesetzt, da zu seinen Schwerpunkten Stabilität und Sicherheit zählen. Andere bekannte Distributionen basieren auf Debian, entwickeln dieses jedoch für eigene Zwecke weiter - beispielsweise das auf Desktop PC weit verbreitete Ubuntu, das auf Netzwerksicherheit ausgelegte Kali Linux und das für den Raspberry Pi optimierte Raspbian.

Jede Distribution besteht aus einem mehr oder weniger stark modifizierten Kernel und einer Sammlung aus Programmen.

Wichtige und unverzichtbare Systemwerkzeuge gehören zur Basisausstattung - etwa ein Paketmanagement oder Befehle für die Dateibearbeitung. Alle weiteren Anwendungen werden von den Distributionen über eigene, sichere Archive (Repositories) zur Verfügung gestellt und können bei Bedarf installiert werden. Es handelt sich dabei um lauffähige Programme und die dazu gehörenden Konfigurationsdateien, die zuvor entsprechend den Richtlinien auf ihre Stabilität, Kompatibilität und Sicherheit überprüft wurden.

Die meisten Distributionen schreiben zwingend vor, dass lediglich nach Open Source lizenzierter Quellcode in die offiziellen Archive aufgenommen werden darf. Entwickeln Unternehmen kommerzielle Software für Linux, musst du diese deshalb manuell installieren. Entwickler bieten dafür in der Regel eine einfache Methode an - beispielsweise durch eigene Paketarchive, einen Installer oder ausführbare Skripte. Nur wenige Distributionen nehmen kommerzielle Software in ihre Archive auf, da sie die Sicherheit nicht anhand des Quellcodes kontrollieren können. Die bekannteste Ausnahme ist Ubuntu, das aus Gründen der Benutzerfreundlichkeit Kompromisse eingeht.

4.1. Dateistruktur von Linux – feste Verzeichnisse für verschiedene Aufgaben

In seinem Aufbau orientiert sich Linux an dem von Bell Laboratories entwickelten Unix, das ab 1970 als Standard für Computer galt.

Hat Ihnen diese Leseprobe gefallen?

Bestellen Sie das Buch als Taschenbuch komfortabel auf Amazon!

Taschenbuch: <https://amzn.to/2ExCtSv>



Möchten Sie über neue Bücher und Sonderangebote vom BMU Verlag informiert werden?

Tragen Sie sich jetzt in unseren E-Mail Newsletter ein:
<https://bmu-verlag.de/>