

Rezepte für Raspberry Pi(e)

Teil 1

Eine Serie macht Appetit auf mehr...

Wartet bei Ihnen schon ein Raspberry Pi auf angemessene Aufgaben? Doch womit fängt man an? Nun, Sie können sich zunächst einmal in das folgende Pi(e)-Grundrezept von Elektor vertiefen. Bei diesem ersten Teil der Serie geht es um die Zutaten, die Konfiguration und schließlich um den Start und den Betrieb von Raspberry Pi.



Von **Tony Dixon** (UK)

Zutaten

Es ist eigentlich keine Frage, dass Sie wissen, dass ein Raspberry Pi – kurz: RPi - ein sehr günstiger Computer im Kreditkartenformat ist [1]. Außerdem gehe ich davon aus, dass Sie schon ein Exemplar besitzen oder gerade überlegen, sich eine RPi-Version zu organisieren. Anders als ein normaler PC oder Laptop basiert ein RPi auf einem ARM-Mikroprozessor und läuft unter Linux. Bevor man aber beginnt seinen RPi zu konfigurieren, muss man zwei

Entscheidungen treffen.

Die erste Wahl betrifft das Betriebssystem. Tabelle 1 listet die zur Auswahl stehenden Varianten auf. Wenn man sich entschieden hat, sollte man das Betriebssystem herunterladen. Doch Achtung: diese Downloads sind groß. Bei langsamen Internetverbindungen oder für den Download über Nacht kann man auch einen geeigneten Torrent-Link suchen. Die zweite Entscheidung betrifft die SD-Card. Erforderlich ist eine SD-Card mit einem Spei-

Tabelle 1 - Betriebssysteme

Betriebssystem	Beschreibung
Raspbian „wheezy“	Referenz-Dateisystem basierend auf der optimierten Raspbian-Version von Debian. Enthält LXDE, Midori und Python-Entwicklungs-Tools.
Soft-float Debian „wheezy“	Identisch zum Raspbian-Image „wheezy“, aber zwecks Programm-Kompatibilität mit dem langsameren Fließkomma-ABI kompiliert.
Arch Linux ARM	Basierend auf Arch Linux als einfache und kleine Distribution mit voller Anwender-Kontrolle, die aber für Einsteiger nicht so gut geeignet ist.
RISC OS	Eine moderne Version des originalen Acorn-Archimedes Betriebssystems.
Raspbian	Basierend auf Debian Linux und speziell für die RPi-Hardware [2] konzipiertes Betriebssystem.
Occidentalis	Entwickelt von Adafruit aus der originalen Wheezy-Distribution mit zusätzlichem Hardware-Support für I2C und SPI [6].

cher von minimal 2 GB für das Betriebssystem. 2 GB reichen zwar dafür aus, doch viel Platz für Applikationen und Daten bleibt dann nicht mehr übrig. Eine Kapazität von 4 GB oder mehr ist besser.

Dieses Kochrezept basiert auf Raspbian [2] „wheezy“, das auch von der Raspberry Pi Foundation empfohlen wird. Und los geht’s: Man hole sich zuerst einmal „wheezy“ von der RPi-Webseite. Wir nutzen die (zum Zeitpunkt der Arbeit an diesem Artikel) neueste Version „2012-12-16-wheezy-raspbian.zip“. Nun muss dieses OS natürlich noch auf der SD-Card installiert werden. Hierzu benötigt man einen Laptop oder Desktop-PC plus einen passenden Card-Reader.

Ein Linux-User weiß normalerweise, wie man so eine Linux-Distribution auf eine SD-Card bekommt und mit dem Befehl `dd` expandiert. Doch ein Windows-User (wie ich) nutzt hierzu besser die Hilfe eines kleinen Utility-Programms namens „Win32DiskImager“ [3]. Nach Download und Installation von Win32DiskImager steckt man zunächst die SD-Card in den Reader. Mit Win32DiskImager wählt man dann die zuvor herunter geladene Linux-Distribution aus. Ein Klick auf die Write-Schaltfläche sowie ein OK zur Bestätigung und das Betriebssystem wird auf die SD-Card geschrieben (siehe Bild 1).

Während die SD-Card beschrieben wird, kann man das RPi-Computer-System komplettieren. Man benötigt noch eine USB-Maus, eine USB-Tastatur und entweder einen Monitor mit HDMI-Anschluss (bzw. einen Monitor mit DVI-

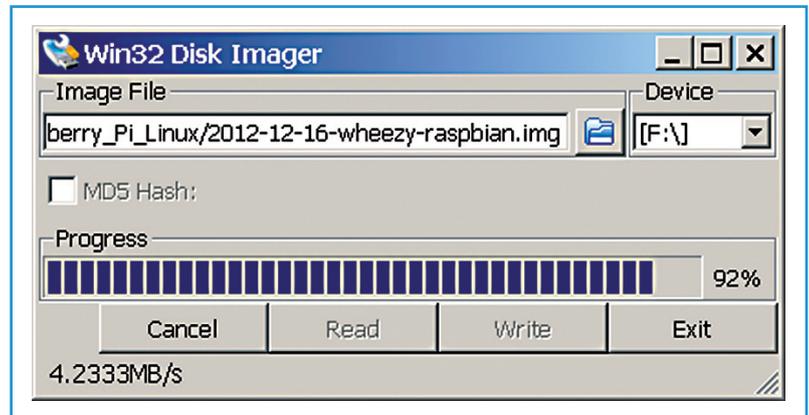


Bild 1. Win32DiskImager in Aktion.

Anschluss und passendem HDMI/DVI-Adapter) oder aber für echtes 80er-Jahre Retro-Fee-ling einen Fernseher mit Composite-Video-Eingang. VGA ist also obsolet, außer man nutzt einen (aufwendigen) aktiven HDMI-VGA-Adapter. Nicht zu vergessen: Ein USB-Kabel mit Micro-USB-Stecker zur Stromversorgung des RPi.

Nach Anschluss von Bildschirm, Tastatur und Maus kann man die fertige SD-Card in den Slot schieben und das versorgende USB-Kabel einstecken, worauf RPi zum ersten Mal booten sollte. Dabei sollte man massenweise Kommandozeilen über den Bildschirm rauschen sehen, während RPi sich selbst konfiguriert. RPi bootet direkt von der SD-Card, was etwas Zeit benötigt. Das Elegante an dieser Lösung ist, dass wenn man ein neues OS ausprobieren oder andere Programme testen will, man einfach die SD-Card (oder ein weiteres Exemplar) damit beschreiben.

Tabelle 2 - Raspi-config

Option	Beschreibung
info	Hilfe zum Tool Raspi-config.
expand_rootfs	Erweitert die Root-Partition auf die Kapazität der SD-Card.
overscan	Passt die Bildschirm-Position an den verwendeten Monitor an.
configure_keyboard	Auswahl des passenden Tastatur-Layouts.
change_pass	Änderung des Default-Passworts „pi“.
change_locale	Festlegung des Orts, an dem RPi betrieben wird.
change_timezone	Auswahl der richtigen Zeitzone passend zum Ort.
memory_split	Änderung der RAM-Verteilung zwischen Anwendung und Display.
ssh	Aktivierung/Deaktivierung des SSH-Servers.
boot_behaviour	Automatisches Starten der GUI nach Bootvorgang.
update	Suche und Durchführung von Updates des Raspi-config-Tools.

Konfiguration

Wenn RPi den Boot-Vorgang abgeschlossen hat, startet automatisch das Raspi-Config-Set-up-Tool (siehe Bild 2). Dieses Tool ähnelt entfernt dem BIOS eines PCs und erlaubt die Konfiguration des RPi. Mit Hilfe der Cursor-Tasten navigiert man durch die Listen mit Optionen und mit der Eingabe-Taste wählt man eine Option aus. Tabelle 2 zeigt die Menü-Struktur von Raspi-config. Welche Optionen man wählt, das hängt von den persönlichen Anforderungen ab – lediglich bei den Einstellungen für Tastatur, Ort und Zeitzone ist man verständlicherweise festgelegt.



Doch zuallererst sollte man die Option „expand_rootfs“ auswählen. Man fährt also mit den Cursor-Tasten zum Eintrag „expand_rootfs“ und betätigt die Eingabetaste. Damit teilt man RPi mit, dass das Dateisystem auf die volle Kapazität der SD-Card erweitert werden soll. Bei der originalen Installation des Betriebssystems auf der SD-Card wurde nur so viel Platz belegt, wie dieses unbedingt benötigt. Wurde z.B. eine 4-GB-SD-Card verwendet, hat das Dateisystem den größten Teil der Karte noch nicht integriert. Nach dieser Einstellung wird RPi beim nächsten Booten ein kleines Script ablaufen lassen, mit dem dann der ganze Speicherplatz der SD-Card inkludiert wird.

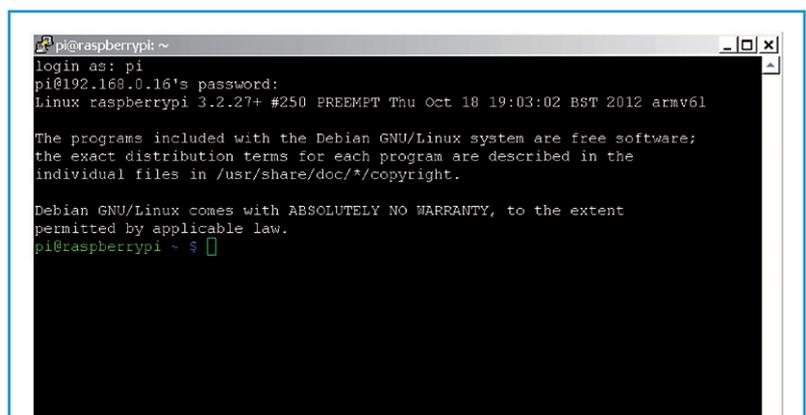
Hierzu muss man natürlich die IP-Adresse von RPi kennen. Der eigene Internet-Router gibt üblicherweise über die aktiven IP-Adressen Auskunft. Alternativ kann man bei RPi das LXTerminal starten und „ip addr“ eingeben. Nach Abschluss dieser grundlegenden Konfiguration kann man RPi resetten und dann die zusätzlichen Optionen konfigurieren, die man für nützlich hält.

Bild 2. Das Tool „Raspi-config“.

Es empfiehlt sich naheliegenderweise, dass man das Default-Passwort ändert. Hierzu navigiert man nach „change_pass“ und betätigt die Eingabetaste, nach der Eingabeaufforderung gibt man das neue Passwort ein. Bei grafisch anspruchsvollen Applikationen oder Spielen kann man das Verhältnis der RAM-Belegung für Anwendungen und Video verändern. Das ist vor allen Dingen dann angesagt, wenn man ein RPi der Revision 2 hat, bei dem 512 MB RAM zur Verfügung stehen. Die Auswahl des automatischen Starts des GUI nach dem Bootvorgang ist ebenfalls sinnvoll, falls man nicht lieber mit der Kommandozeile vorliebnimmt. Eine letzte Option betrifft die Möglichkeit des Fernzugriffs auf RPi. Wenn man das möchte, dann aktiviert man den SSH-Server (Secure Shell). In diesem Fall kann man auf seinem „richtigen“ Computer einen SSH-Client wie z.B. PuTTY for Windows [4] starten und damit dann RPi per Terminal fernsteuern (siehe Bild 3), wenn beide online sind.

Wenn man bei „boot_behaviour“ eingestellt hat, dass das GUI nach dem Booten zur Verfügung steht, wird man nach abgeschlossenem Bootvorgang das Debian-Desktop-Environment bewundern können. Ohne Änderung von „boot_behaviour“ wird man sich nach dem Booten im Kommandozeilen-Modus befinden. In diesem Fall gibt man nun den User-Namen (default „raspberrypi“) und das Passwort (default „pi“) ein. Wenn man dann eingeloggt ist, kann man durch Eingabe von „startx“ und der Eingabetaste das GUI manuell starten. RPi sollte nun eine Oberfläche wie in Bild 4 zeigen.

Bild 3. Verwendung von PuTTY zur Fernsteuerung.



Unter dem GUI kann man dann RPi weiter konfigurieren. Will man Netzzugang per WLAN, klickt man auf das Icon von „WiFi Config“ (siehe Bild 5). Zuvor sollte natürlich ein passender WLAN-Stick in einem der USB-Ports stecken. Hier empfiehlt sich aufgrund des Strombedarfs des Sticks die Verwendung eines USB-Hubs mit eigenem Netzteil, wodurch außerdem mehr USB-Ports zur Verfügung stehen.

Demnächst

Wenn Raspberry Pi so läuft wie gedacht, was dann? Elektor wäre keine Elektronik-Zeitschrift, wenn es demnächst nicht weitere Folgen geben würde, die sich mit dem RPi-Expansion-Header und der Programmierung seiner Pins beschäftigen würden...
 Kommentare? Feedback? Support von der Elektor-Community? Beteiligen Sie sich an der Diskussion in Elektor.LABS [5]!

(130078)

Weblinks

- [1] www.raspberrypi.org
- [2] www.raspbian.org
- [3] www.launchpad.net/win32-image-writer
- [4] www.putty.org
- [5] www.elektor-labs.com/ElektorPOST/2013/03
- [6] www.learn.adafruit.com/adafruit-raspberry-pi-educational-linux-distro/overview

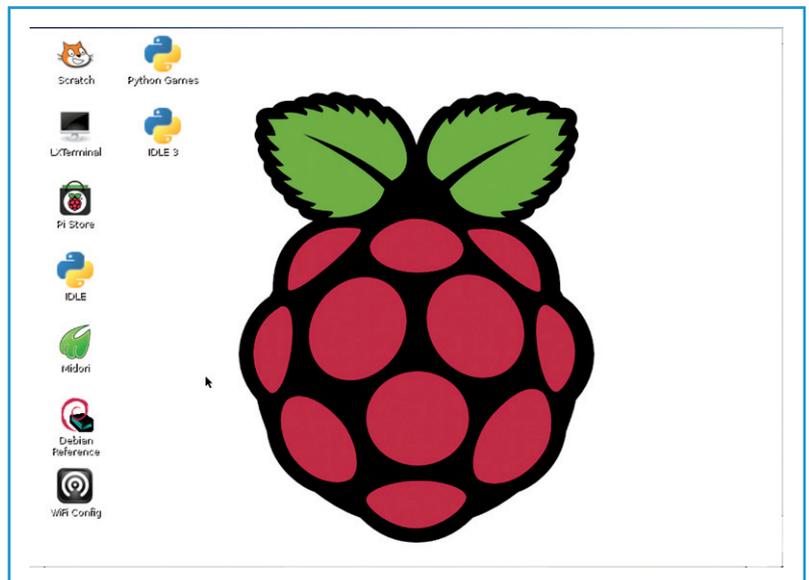


Bild 4. Der grafische Desktop von Raspberry Pi.



Bild 5. Das Icon von „Wifi Config“.

