

Mattias Schlenker

SCHNELLEINSTIEG BANANA PI

ALLE WICHTIGEN
BOARDS

BPI-M1, PRO,
BPI-R1, BPI-M2
UND BPI-D1



160 SEITEN

BOARDS, INSTALLATION,
PROGRAMMIERUNG,
ELEKTRONIKPROJEKTE UND
ZUBEHÖR NUTZEN

FRANZIS

Mattias Schlenker
Schnelleinstieg
Banana Pi

Mattias Schlenker entwickelt Live-Linux-Distributionen, vornehmlich als Notfall- und Rettungssysteme, und ist als Autor für Linux-, Netzwerk- und Sicherheitsthemen für verschiedene Computermagazine tätig. Zu Arduino™ kam der gelernte Kraftfahrzeugmechaniker aus Neugier, als er nach einer günstigen Möglichkeit suchte, Betriebsdaten seines Pkw über die OBD-II-Schnittstelle auszulesen.

Mittlerweile ist er fasziniert von der Möglichkeit, mit Arduino™ die verschiedensten Mess-, Regelungs- und Steuerungsaufgaben einfach, günstig und schnell umzusetzen – dementsprechend viele Arduinos sind in seinem Haushalt im Einsatz. Das in den letzten Jahren neu erworbene Arduino-Fachwissen gibt er gern in Buchform weiter. Auch auf andere Platinen überträgt Schlenker dieses Wissen, in diesem Fall auf den Banana Pi.

Bibliografische Information der Deutschen Bibliothek

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte Daten sind im Internet über <http://dnb.ddb.de> abrufbar.

Alle Angaben in diesem Buch wurden vom Autor mit größter Sorgfalt erarbeitet bzw. zusammengestellt und unter Einschaltung wirksamer Kontrollmaßnahmen reproduziert. Trotzdem sind Fehler nicht ganz auszuschließen. Der Verlag und der Autor sehen sich deshalb gezwungen, darauf hinzuweisen, dass sie weder eine Garantie noch die juristische Verantwortung oder irgendeine Haftung für Folgen, die auf fehlerhafte Angaben zurückgehen, übernehmen können. Für die Mitteilung etwaiger Fehler sind Verlag und Autor jederzeit dankbar. Internetadressen oder Versionsnummern stellen den bei Redaktionsschluss verfügbaren Informationsstand dar. Verlag und Autor übernehmen keinerlei Verantwortung oder Haftung für Veränderungen, die sich aus nicht von ihnen zu vertretenden Umständen ergeben. Evtl. beigefügte oder zum Download angebotene Dateien und Informationen dienen ausschließlich der nicht gewerblichen Nutzung. Eine gewerbliche Nutzung ist nur mit Zustimmung des Lizenzinhabers möglich.

© 2015 Franzis Verlag GmbH, 85540 Haar bei München

Alle Rechte vorbehalten, auch die der fotomechanischen Wiedergabe und der Speicherung in elektronischen Medien. Das Erstellen und Verbreiten von Kopien auf Papier, auf Datenträgern oder im Internet, insbesondere als PDF, ist nur mit ausdrücklicher Genehmigung des Verlags gestattet und wird widrigenfalls strafrechtlich verfolgt.

Die meisten Produktbezeichnungen von Hard- und Software sowie Firmennamen und Firmenlogos, die in diesem Werk genannt werden, sind in der Regel gleichzeitig auch eingetragene Warenzeichen und sollten als solche betrachtet werden. Der Verlag folgt bei den Produktbezeichnungen im Wesentlichen den Schreibweisen der Hersteller.

Ein Dank geht an die ALLNET GmbH für die Unterstützung des Projekts, besonders an Herrn Gerhard Zerwes.

Programmleitung: Dr. Markus Stäuble

Lektorat: Ulrich Dorn

Satz: Nelli Ferderer, nelli@ferderer.de

art & design: www.ideehoch2.de

Druck: CPI-Books

Printed in Germany

ISBN 978-3-645-60390-4

INHALT

VORWORT	10
Konventionen im Buch	11
Schaltpläne	11
Hilfe, ich weiß nicht weiter!	15
1. GRUNDLAGEN	16
1.1 Geschichte des Banana Pi	16
1.1.1 Die frühen Jahre	16
1.1.2 Die ARM-Plattform	17
1.1.3 Der Raspberry Pi	18
1.1.4 Banana Pi – Raspberry Pi weitergedacht	20
1.2 Banana Pi-Boards im Detail	22
1.2.1 Banana Pi M1 (Lamobo M1)	22
1.2.2 Banana Pro	23
1.2.3 Banana Pi R1 (Lamobo R1)	24
1.2.4 Banana Pi M2	24
1.2.5 Banana Pi D1 (Lamobo D1)	25
1.2.6 Orange Pi und Olinuxino	25
1.3 Unterstützte Betriebssysteme	26
1.3.1 Android	27
1.3.2 Bananian Linux	27
1.3.3 Raspbian Linux	28
1.3.4 Ubuntu Linux	29
1.3.5 OpenWRT	30
1.3.6 LeMedia	31
1.3.7 Andere Linux-Distributionen	31
1.3.8 NetBSD, OpenBSD und FreeBSD	31

2. BETRIEBSSYSTEM INSTALLIEREN	32
2.1 Bananian	35
2.1.1 Updates einspielen	38
2.2 OpenWRT (Lamobo R1)	40
2.2.1 Konfiguration per WLAN	41
2.2.2 Konfiguration per LAN	42
2.3 Ubuntu	44
3. BANANIAN IN DER PRAXIS	46
3.1 Netzwerkkonfiguration	46
3.1.1 Statische IP-Adresse	46
3.1.2 Banana Pi im WLAN	48
3.1.3 Banana Pi als WLAN-Access-Point	50
3.1.4 WLAN-Brücke	53
3.1.5 Kompletter Router mit Bananian	54
3.2 Druckerpooler mit CUPS	56
3.3 ownCloud-Server aufsetzen	60
3.3.1 Vorbereitung des Apache-Servers	61
3.3.2 Installation von ownCloud	62
3.3.3 Per Dynamic DNS überall erreichbar	65
3.3.4 Clients fürs Smartphone	66

3.4 MySQL-Datenbank fürs Büro	67
3.4.1 Installation von MySQL	68
3.4.2 Einrichtung einer Datenbank	69
3.4.3 Netzwerkfreigabe	70
3.4.4 Anbindung an LibreOffice	70
3.5 Dateiserver im Heimnetz	73
3.5.1 Vorbereitung	74
3.6 Desktopersatz	77
3.6.1 Auswahl des Desktops	78
3.6.2 Desktop einrichten	79
3.6.3 Office nachinstallieren	80
3.6.4 Kein Sound?	81
3.7 Videorekorder und -streamer	82
3.7.1 TVheadend installieren	84
3.7.2 Clients verbinden	89
4. GPIO KREATIV NUTZEN	92
4.1 Pinbelegung der Banana Pi	93
4.2 Software installieren	94
4.3 Start mit Steckbrett und LED	95
4.4 Schalter in Shell auslesen	99

4.5 Elegantes GPIO mit Python	102
4.6 FS20-Sender ansprechen	104
4.7 Servomotoren nutzen	107
4.8 Eine Webcam anschließen	112
4.9 Servo + Webcam + Webserver = Überwachungs- lösung	115
4.10 Arduino ansteuern	121
4.10.1 Arduino-IDE am PC installieren	123
4.10.2 Software auf dem Banana Pi installieren	126
4.10.3 Arduino-Programme vom PC aus hochladen	127
4.10.4 Schwarze Seite: 5-Euro-Arduino	131
4.11 Monochrome LC-Displays ansteuern	134
4.11.1 Anschluss des Displays	134
4.11.2 Installation der Software	136
4.11.3 Der Code	137
5. ANHANG	140
A.1 Gehäuse selbst bauen	140
A.2 Die serielle Konsole	140
A.2.1 Zugriff mit PuTTY oder screen	143
A.3 SSH-Log-in automatisieren	146

A.4 vi - verhasster und vergötterter Editor	148
A.5 Laufwerke unter Linux	149
A.6 Linux-Einstieg in 15 Minuten	151
A.6.1 Shell	151
A.6.2 Dateien und Ordner	151
A.6.3 Superuser werden	153
A.6.4 Berechtigungen ändern	153
A.6.5 Lange Dateien anzeigen	154
A.6.6 Suchen	154
A.6.7 Archive	155
A.6.8 Manuals	155
INDEX	156

VORWORT

Sie haben einen Banana Pi erstanden oder planen einen zu kaufen? Glückwunsch! Vielleicht, weil Sie sich mit anderen Single-Board-Computern befasst haben und der Banana Pi den besten Kompromiss aus großer Community, intelligent zusammengestellter Hardware und einem günstigen Preis bietet? Vielleicht haben Sie bereits Erfahrung mit dem Raspberry Pi und sind an die Grenzen des beliebtesten Single-Board-Computers gestoßen?

Single-Board-Computer machen einfach Spaß, sie bieten vergleichsweise viel Rechenleistung für kleines Geld bei geringer Leistungsaufnahme. Offene Hardwareplattformen mit freier Software machen doppelt Spaß, denn mit ihnen lassen sich ganz tolle Gerätschaften realisieren, die auf x86-Basis den Stromzähler rotieren lassen würden.

Egal wie groß Ihre Vorerfahrung mit Linux insbesondere auf anderen Plattformen als Intel und AMD von PCs ist: Ich setze keine Grundkenntnisse zwingend voraus und verweise gegebenenfalls auf externe Seiten, die angesprochene Vorgehensweisen konkretisieren. Leider kann ich auf weniger als 200 Seiten keine vollständige Abhandlung der beiden im Buch am intensivsten genutzten Linux-Distributionen Debian und OpenWRT geben. Dennoch sollten Sie sich – auch als Linux-Einsteiger – schnell zurechtfinden und bald erste Erfolge bei der Einrichtung des Banana Pi als Desktopersatz oder als Router mit NAS-Funktionalität vorweisen können.

Konventionen im Buch

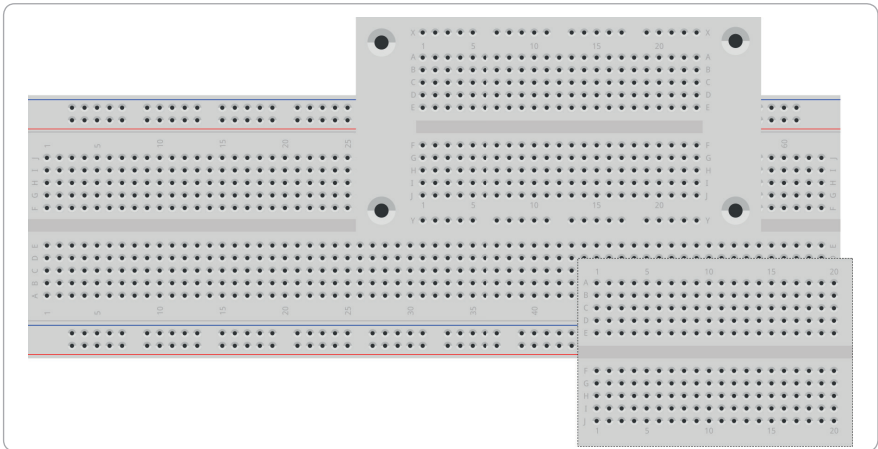
Schaltpläne

Die 26- oder 40-polige GPIO-Sockelleiste des Banana Pi lädt nicht nur dazu ein, Erweiterungsplatinen – sogenannte Shields – aufzustecken, sondern auch direkt Sensoren oder Aktoren zu verbinden. Das müssen keine hochtrabenden Geschichten wie spezielle Funkmodule sein, oft ist bereits ein simpler Taster sehr nützlich, mit dem eine bestimmte Funktion aktiviert oder deaktiviert werden soll. Praktisch sind LEDs zur Statusanzeige: Konnte das letzte Backup richtig durchgeführt werden, liegen ungelesene E-Mails vor – oder habe ich neue Facebook-Freunde?

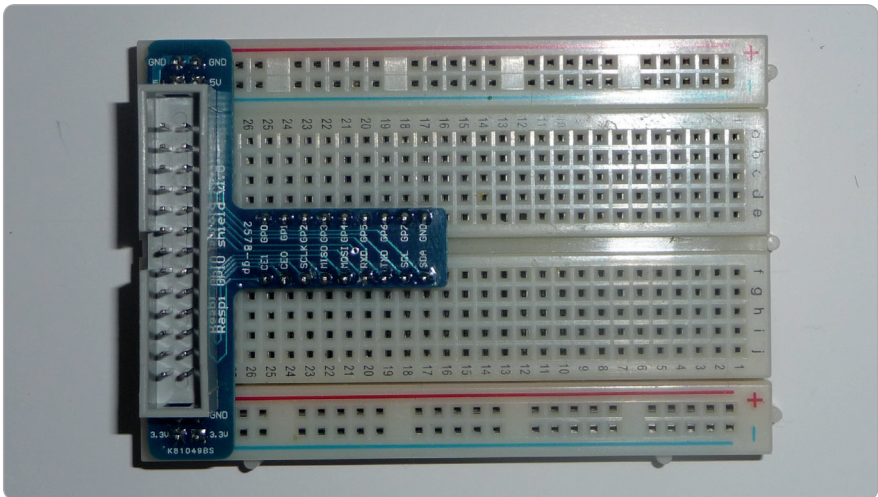
In solch einem Fall kann der Anschluss direkt am Banana Pi erfolgen. Praktisch ist dafür ein sogenanntes Steckbrett oder Breadboard: Besorgen Sie neben dem Breadboard ein Bündel Steckbrückenkabel in der Variante »männlich zu weiblich«, einige mit zwei männlichen Enden und des Weiteren ein Grundsortiment von Widerständen (1 bis 47 kOhm), LEDs und Tastern. Haben Sie bereits mit Arduino experimentiert, werden Sie über diese Bauteile verfügen. Falls nicht, erhalten Sie Einsteigersortimente überall dort, wo es Arduino gibt.

Werden viele der GPIO-Pins des Banana Pi auf dem Breadboard benötigt, hilft ein sogenanntes GPIO-Breakout, wie es Sainsmart im Programm hat (<http://bit.ly/1FVT9L1>). Es führt die 26-GPIO-Pins mit einem Flachbandkabel aufs Breadboard (Achtung, für Banana Pi-Varianten mit 40-poliger GPIO-Leiste müssen Sie ein entsprechendes Flachbandkabel besorgen). Eine Luxusvariante, die zudem mit Arduino verwendet werden kann, hat der deutsche Banana Pi-Distributor Allnet im Programm.

Für die Verdrahtungspläne verwenden wir das kostenlose Programm Fritzing, das Sie von www.fritzing.org herunterladen können. Lassen Sie sich nicht davon verwirren, dass die Grafiken einen vermeintlichen Raspberry Pi Modell B zeigen: Raspberry Pi Modell B und Banana Pi M1 sind bis auf Feinheiten wie die Durchnummerierung der seriellen Schnittstellen pincompatibel. Dort, wo der zusätzliche abgesetzte serielle Port zum Einsatz kommt, zeigt ein Foto den Anschluss.



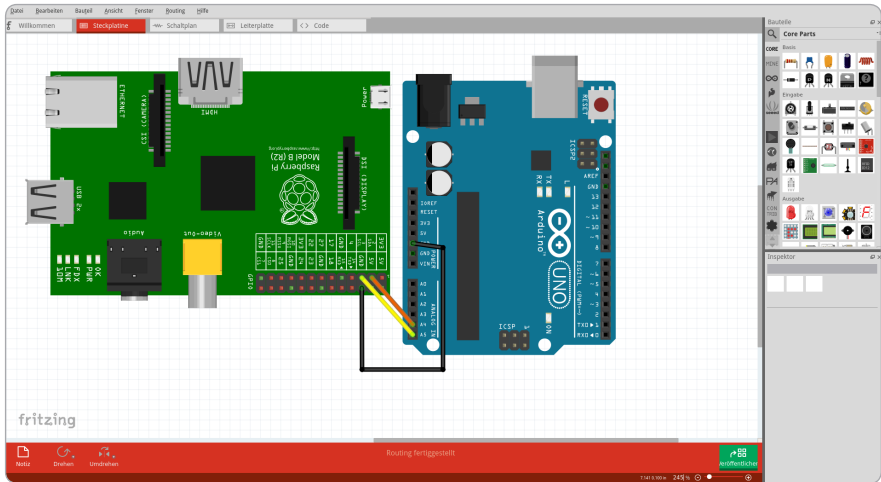
Drei verschiedene Breadboards: In einer vertikalen Reihe sind A bis E und F bis J durchverbunden. Horizontal gibt es die zwei oder vier abgesetzten Kontaktleisten – sie dienen als Schienen für Masse und Versorgungsspannung, das Mini-Breadboard rechts unten verzichtet auf sie.



Das GPIO-Breakout von Sainsmart führt alle 26 GPIO-Pins aufs Breadboard heraus – praktisch, wenn zu viele GPIO-Pins benötigt werden, um einzeln verdrahten zu können.

ACHTUNG!

Die Anschlüsse für Display und Kamera sind nicht kompatibel.



In der Steckplatinenansicht von Fritzing können Sie unsere Entwürfe im Detail ansehen und nach Belieben erweitern.

DOWNLOAD VON BEGLEITMATERIAL

Leider sind im Schwarz-Weiß-Druck gelegentlich Kabelverbindungen auf Fotos oder Fritzing-Skizzen schlecht zu erkennen. Für diesen Fall bieten wir sämtliche Fritzing-Skizzen zum Download an. Sie finden sie unter www.buch.cd oder im GitHub-Repository des Autors unter <https://github.com/mschlenker>. Sie können in Fritzing per Mouseover den Namen der Anschlüsse ermitteln, Bauteile anders stecken und weitere Bauteile hinzufügen.



Im GitHub-Repository des Autors können Sie alle Schaltpläne, Programme und Skripte herunterladen – erstellen Sie einen GitHub-Account, um eigene Änderungen versionieren zu können.



ACHTUNG!

Unsere Fritzing-Skizzen sind lediglich als Schemata und Hilfe beim Stecken von Schaltungen auf dem Breadboard zu verstehen. Wenn Sie Fritzing nutzen, um Erweiterungsplatinen zu entwerfen und bei Fritzing in Kleinstserie (ab Losgröße eins) fertigen zu lassen, müssen Sie Bohrlöcher und Aussparungen an Ihre jeweils verwendete Banana Pi-Platine anpassen!

Auch Quellcodes verwendeter Skripte sind im GitHub-Repository erhältlich. Das erlaubt Änderungen, und die online verfügbaren Versionen haben wir, wo nötig, umfangreicher kommentiert. Noch ein Wort zu GitHub: Zwar können Sie einzelne Dateien im Nur-Text-Format anzeigen lassen und per Copy-and-paste übernehmen oder herunterladen, doch damit verschenken Sie einen der Hauptvorteile von Git: die Versionsverwaltung.

Gerade wenn Sie planen, die von uns erstellten minimalen Beispiele zu erweitern, lohnt es sich, selbst ein GitHub-Konto anzulegen und unser Repository als sogenannten Fork zu übernehmen. Dieses Repository können Sie mit dem GitHub-Client auf Ihren PC synchronisieren und dort vorgenommene Änderungen wiederum zu GitHub synchronisieren. So bleiben Änderungen nachvollziehbar, und Sie können leicht zu jedem Stand eines Programms zurückkehren.

Hilfe, ich weiß nicht weiter!

Zwar ist die Banana Pi-Community nicht ganz so groß wie die des Raspberry Pi, doch gerade weil viele Nutzer des Banana Pi bereits über Erfahrung mit anderen Single-Board-Computern verfügen und sich bewusst für diese Plattform entschieden haben, ist die durchschnittliche Kompetenz der Antworten recht hoch – und damit auch die Qualität.

Ein guter Anlaufpunkt ist das Forum von LeMaker, erreichbar unter <http://forum.lemaker.org>. Sinovoip hat die Google-Plus-Gruppe »Banana Pi Open Source« ins Leben gerufen, die bei Redaktionsschluss über 1.200 Mitglieder stark war: <https://plus.google.com/communities/116770564125019694131> – auch ich bin in dieser Gruppe anzutreffen.

Ein Blog zum Buch pflege ich unter www.bananapi-buch.de. Hier finden Sie Hinweise zu neuen Betriebssystem-Images, zu interessanten Projekten rund um Banana Pi, aber auch Ergänzungen und weitergedachte Projekte. Sollte sich herausstellen, dass Abschnitte des Buchs nähere Erläuterungen und tiefer gehende Erklärungen benötigen, werde ich diese ebenfalls im Blog veröffentlichen. Sie können mich auch per E-Mail unter ms@mattiasschlenker.de kontaktieren – für diesen Fall bitte ich aber darum, dass Sie mir erlauben, die gestellte Frage und meine Lösungsvorschläge (gegebenenfalls anonymisiert) im Blog und der Google-Plus-Gruppe zu veröffentlichen.

1.1 Geschichte des Banana Pi

Single-Board-Computer sind kein neues Phänomen, bereits Ende der 1990er gab es die Minimalcomputer für kommerzielle Kunden. Sie hatten primär zwei Einsatzbereiche: Der erste war die Entwicklung von Steuerungskomponenten für die Industrie. Die hierfür verwendeten Platinen wurden auf breite klimatische Einsatzbereiche optimiert, in kleinen Serien hergestellt, und die Hersteller garantierten eine lange Verfügbarkeit – das sorgte leider für hohe Preise. Bereits für einfache SBCs wurden mehrere Hundert Euro aufgerufen.

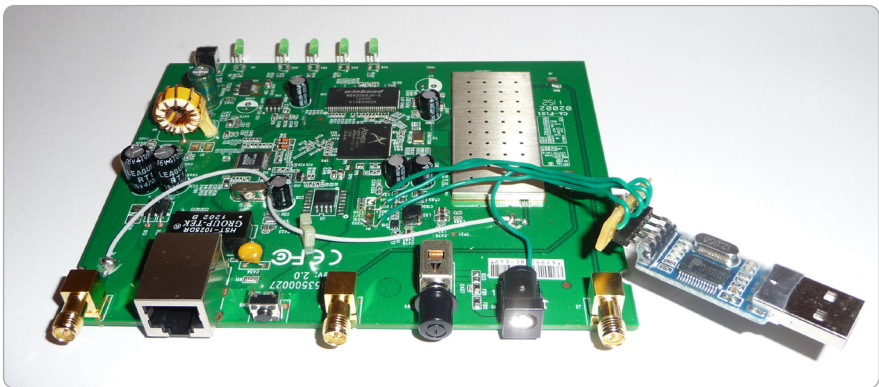
Der zweite Einsatzbereich war der als Musterplatine für die aufkommende Heimvernetzung, wie DSL-Router oder Network Attached Storage (NAS). Basierend auf den Referenzdesigns, konnten Hardwarehersteller Software für DSL-Router und später die abgespeckten Platinen (weniger Schnittstellen, weniger Speicher) entwickeln. In Form von NAS und Router wurden die ersten Single-Board-Computer von der Community entdeckt, die mit eigener Firmware die Funktionalität erweiterte und so vollwertige Server für den Hausgebrauch schuf oder Linux-basierte Settop-Boxen zu Entertainment-Centern ausbaute.

1.1.1 Die frühen Jahre

Die bekanntesten Geräte dieser Ära dürften der Linksys-Router WRT54G(L) und der erste populäre NAS-Server Linksys NSLU2 gewesen sein. Softwareseitig verschrieb sich das OpenWRT-Projekt (und das verwandte DD-WRT-Projekt) der Entwicklung einer schlanken Firmware. Debian arbeitete früh an Paketen für die typischerweise verwendeten MIPS- und ARM-Architekturen, sodass erfahrene Linux-Anwender mit den bekannten Paketwerkzeugen arbeiten konnten. Heutzutage gesellen sich mit Yocto Linux und uClibc Buildroot weitere Embedded Linuxe hinzu, und in Form von NetBSD und mit Einschränkungen FreeBSD stehen »echte Unixe« bereit.

Dennoch galten Single-Board-Computer bis vor ein paar Jahren als reines Nerd-Spielzeug. Das hatte vor allem zwei Gründe: Eine fehlende Anschlussmöglichkeit für Tastatur und Monitor machte das Debugging schwierig und erforderte zusätzliche Kenntnisse sowie zusätzliche Hardware in Form eines 3,3-V- oder 5-V-USB-Seriell-Adapters, um auf den Bootloader und vor Initialisierung der Netzwerkschnittstellen auf eine Konsole zugreifen zu können.

Die zweite Hürde waren fest verlötete »Massenspeicher« – meist Flash-Bereiche mit 2 bis 32 MByte: Schlägt die Installation eines Betriebssystem-Images fehl, muss umständlich aus dem Bootloader heraus per TFTP (*Trivial File Transfer Protocol*) eine erneute Installation gestartet werden – auch hierfür ist zusätzliche Software (im Sinne eines TFTP-Servers) notwendig, zudem brauchen Sie das nötige Expertenwissen im Umgang mit Bootloader und TFTP-Server.



Linux-Experimente auf dem DSL-Router: Weil die Netzwerkschnittstelle nicht initialisiert wurde, musste ich zum Lötcolben greifen, um Zugriff auf eine serielle Konsole zu bekommen – derartiges Ungemach droht beim BananaPi nicht.

1.1.2 Die ARM-Plattform

Ein Meilenstein für die Entwicklung heute omnipräsenter Minicomputer (Smartphones, Tablets, NAS-Geräte, Videoplayer etc.) war die Entwicklung der ARM-Prozessorfamilie. Entstanden ist diese im Umfeld der Universität Cambridge, als die Entwickler des »Volkscomputers« BBC Micros mit dem verbauten Prozessor unzufrieden waren. Mangels guter Rechenleistung zu günstigen Preisen entwickelten sie einfach einen eigenen Prozessor. Der trug zunächst den Namen *Acorn RISC Machines* – RISC wiederum ist ein Akronym für *Reduced Instruction Set Computing*, was dafür steht, dass wenige, dafür hochoptimierte Befehle verwendet werden. Später wurde die Prozessorsparte ausgelagert, die Abkürzung steht seither für *Advanced RISC Machines*.

Für den Erfolg der ARM-Architektur dürften zwei Entscheidungen ausschlaggebend gewesen sein: Zum einen können ARM-Prozessoren mit wenigen Transistoren und damit billig gefertigt werden. Zum anderen baut ARM Holdings selbst keine Prozessoren, sondern entwirft nur die Kerne. Chiphersteller können diese Kerne lizenzieren und mit Peripherie wie einem Grafik-Chipsatz oder Hardware für Schnittstellen (SATA, Ethernet) kombinieren. Derart kombinierte, hochintegrierte Systeme nennt man *System on a Chip*, kurz SoC. Dieses Geschäftsmodell erlaubt die Herstellung günstiger Prozessoren, die exakt die benötigte Hardware mitbringen: ARM-SoCs in NAS-Geräten benötigen keine Grafikhardware, so kann man darauf komplett verzichten. ARM-SoCs in Android-Spielekonsolen dagegen integrieren separat lizenzierte Grafikkerne.

Bei Redaktionsschluss war die 32-bittige ARM-Architektur ARMv7 die am weitesten verbreitete. ARMv8-Prozessoren, die 64 Bit und Virtualisierungsfunktionen kombinieren, waren bis dato nur in High-End-Smartphones und Servern zu finden. Erfahrungsgemäß dürfte es bis 2017 oder 2018 dauern, bis die Vorteile von ARMv8 wirklich auf Single-Board-Computern zum Tragen kommen: Bei 1 GByte Arbeitsspeicher in durchschnittlichen SBCs ist viel Luft nach oben, bis die Grenze von 4 GByte der 32-Bit-Architektur erreicht ist, und Virtualisierung wird eher im Smartphone praktische Anwendungen finden (zur Trennung einer beruflichen und einer privaten Betriebssysteminstanz) als auf Hobby-SBCs.

1.1.3 Der Raspberry Pi

Mit beiden Nachteilen der Routerboards und frühen Evaluationsboards räumte der Raspberry Pi auf: Der im Umfeld der Universität Cambridge entwickelte Einplatinencomputer war von vornherein als Billigcomputer für den Bildungsbereich konzipiert. Ganz oben auf der Agenda stand daher der Anschluss von Fernsehern – weil auch Schwellenländer anvisiert wurden, sogar in Form der gelben analogen Composite-Buchse der frühen Modelle. Organisiert ist das Raspberry Pi-Projekt relativ hierarchisch: Die Raspberry Pi Foundation kontrolliert die Weiterentwicklung der Hardware, deren Fertigung sowie die im »Standardbetriebssystem« Raspbian enthaltenen Programme. Hard- und Software sind erwartungsgemäß eng miteinander verzahnt, was die Einrichtung und Inbetriebnahme stark vereinfacht.

Dennoch gibt es Kritik an dem hierarchischen Entwicklungsmodell des Raspberry Pi: Der verwendete ARM-Prozessorkern (ARMv6) galt bereits bei Veröffentlichung des Computerchens als veraltet – sogar das anderthalb Jahre

Dank der Debian-Basis ist Bananian das am umfangreichsten erweiterbare Linux für den Banana Pi. Die meisten Programme können ohne Änderungen aus den offiziellen Paketquellen von Debian bezogen werden. Auch ist die meist hervorragende Dokumentation im Debian-Wiki und in vielen Blogs fast immer eins zu eins auf Bananian übertragbar. Wegen der engen Verwandtschaft zwischen Debian und Ubuntu gelten die in diesem Kapitel vorgestellten Rezepte beinahe ohne Einschränkungen auch für Ubuntu.

3.1 Netzwerkkonfiguration

Die Standardnetzwerkeinstellung der Bananian-Images ist die automatische Konfiguration per DHCP auf der Ethernet-Schnittstelle. Praktisch für viele Einsatzbereiche als Client, Medienplayer oder Server (wenn der DHCP-Server im Netz davon weiß). Aber bereits wenn der Banana Pi selbst zum Server werden oder als Client per WLAN eingebunden werden soll, ist diese Lösung zu unflexibel. Die Konfiguration der Netzwerkschnittstellen erfolgt bei Debian in der Datei `/etc/network/interfaces`.

Ubuntus »Best Practice« ist mittlerweile, diese Datei auf mehrere Dateien in `/etc/network/interfaces.d` aufzusplitten, doch Sie können weiterhin eine einzelne Konfigurationsdatei pflegen. Eine jeweils aktuelle Übersicht über alle konfigurierten Schnittstellen liefert der Befehl

```
ifconfig
```

Und wenn Sie dem Befehl als Parameter den Schnittstellennamen mitgeben, werden die Informationen zu dieser Schnittstelle angezeigt:

```
ifconfig wlan0
```

3.1.1 Statische IP-Adresse

Ist angestrebt, den Banana Pi als NAS oder Druckerserver im lokalen Netz zu nutzen, ist es oft von Vorteil, den Computer unter einer gleichbleibenden IP-Adresse ansprechen zu können. Um nicht auf das Wohl eines vorhandenen

DHCP-Servers angewiesen zu sein, ist es daher sinnvoll, eine statische Adresse zuzuweisen. Ermitteln Sie zunächst durch einen Blick ins Konfigurationsfrontend Ihres DSL-Routers, in welchem Adressbereich er dynamisch zuweist.

Üblicherweise sind Adressen im niedrigen und hohen Teil des verfügbaren Adressbereichs frei verfügbar und können für Geräte genutzt werden, die konsistent erreichbar sind. Als recht praktische Konvention hat es sich dabei erwiesen, vom DSL-Router ausgehend die letzte Stelle der IP-Adresse neuer Geräte zu inkrementieren oder zu dekrementieren.

Um von dynamischer Adresszuweisung auf statische Adresszuweisung umzustellen, genügt es, den Block

```
001 auto eth0
002 iface eth0 inet dhcp
```

umzustellen auf

```
001 auto eth0
002 iface eth0 inet static
003     address 192.168.1.2
004     netmask 255.255.255.0
005     gateway 192.168.1.1
006     dns-nameservers 192.168.1.1 8.8.8.8
```

Die verwendete Adresse ist natürlich anzupassen, und als Gateway muss die IP-Adresse des DSL-Routers eingetragen werden. Auch einen DNS-Server stellt der DSL-Router in der Regel bereit, der zweite DNS ist ein von Google betriebener. An der Netzmaske sind selten Änderungen nötig, meist kommen im Heim- oder Büronetz Adressbereiche mit maximal 254 nutzbaren Adressen zum Einsatz. Nach der Umstellung starten Sie den Banana Pi neu, er ist nun unter der neuen Adresse erreichbar.

3.1.2 Banana Pi im WLAN

Benötigte Software: wpa_supplicant, iw

Um Debian als Client im Netzwerk nutzen zu können, ist die Installation des Programms `wpa_supplicant` notwendig. Es kümmert sich im Hintergrund um die wechselnden WPA-Schlüssel. Zunächst gilt es, auf der Kommandozeile den »pre shared key« zu erzeugen, mit dem die Kommunikation zwischen Access Point und Client initiiert wird. Dafür verwenden Sie den Befehl:

```
wpa_passphrase 'ssid' 'geheim'
```

Dabei ist `ssid` durch den Namen des Netzes und `geheim` durch die Passphrase (manchmal als »Schlüssel« bezeichnet) zu ersetzen. In der Ausgabe interessiert eine Zeile mit einer sehr langen hexadezimalen Zahlenkolonne, die mit `psk=...` anfängt. In die `/etc/network/interfaces` tragen Sie nun

```
001 auto wlan0
002 iface wlan0 inet dhcp
003     wpa-ssid ssid
004     wpa-psk ccb290f...
```

ein. Starten Sie den Banana Pi neu, um die Verbindung mit Ihrem Netzwerk herzustellen. Zumindest bei Banana Pi R1 und Banana Pro sollte die Netzwerkverbindung jetzt stehen, da deren Treiber getestet wurden und sichergestellt ist, dass die richtige Firmware installiert ist. Dass Sie verbunden sind, zeigt die Ausgabe des Befehls:

```
ifconfig wlan0
```

Hier sollten die Assoziation mit dem Access Point und die IP-Adresse angezeigt werden. Falls Sie einen Banana Pi M1 in Kombination mit einem WLAN-USB-Stick verwenden, kann es vorkommen, dass die Firmware des Sticks nicht enthalten ist. Ob Probleme mit der Firmware vorliegen, zeigt dieser Befehl:

```
dmesg | grep -i firmware
```

Erhalten Sie eine Meldung der Art »Firmware XYZ not found«, installieren Sie zunächst die Pakete `linux-firmware` und `linux-firmware-nonfree` nach –

natürlich über die Kabelverbindung. Hilft auch dies noch nicht und das benötigte Firmwaremodul wird noch immer nicht gefunden, hilft der Blick in das GitHub-Repository der Linux-Entwickler:

<https://git.kernel.org/cgit/linux/kernel/git/firmware/linux-firmware.git/tree/>

Hier können Sie fehlende Firmwaredateien einzeln herunterladen (Klick auf *raw* im Anzeigemodus) und unterhalb von `/lib/firmware` auf dem Banana Pi abspeichern.

```
root@bananapi ~ # cat /etc/network/interfaces
# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback

# Automatische Konfiguration (DHCP)
# auto eth0
# iface eth0 inet dhcp

auto wlan2
iface wlan2 inet dhcp
    wpa-ssid fockeberg
    wpa-psk ac22eb97e543412572031b6658fc82665f0

# Manuelle Konfiguration (schnellerer Start
# wenn WLAN vorhanden)

auto eth0
iface eth0 inet manual

# Statische Konfiguration
#iface eth0 inet static
#    address 192.168.6.241
#    netmask 255.255.255.0
#    gateway 192.168.6.1
root@bananapi ~ # █
```

Die WLAN-Schnittstelle erhält als zusätzliche Parameter den Namen des Netzwerks und den Schlüssel – die Schnittstelle `eth0` ist hier für einen schnelleren Start bereits auf `manual` gesetzt.

Günstige WLAN-Sticks vom Elektronikmarkt sind übrigens meist die beste Wahl, weil sie unmodifizierte Chipsätze der großen Hersteller Realtek, Ralink oder Atheros nutzen und daher in der Regel mit den vom Linux-Kernel mitgelieferten Treibern und Firmwaredateien angesprochen werden können.

Mit abgezogenem Netzkabel verzögert sich der Start um bis zu drei Minuten. Während der Tests mit WLAN-Sticks ist es ganz praktisch, zu wissen, dass Sie durch einen simplen Neustart mit Kabel wieder Netzwerk bekommen, im späteren Dauereinsatz ist es jedoch praktischer, für diesen Fall `eth0` komplett auf manuelle Konfiguration umzustellen. Der gesamte Eintrag für `eth0` besteht dann aus diesen beiden Zeilen:

```
001 auto eth0
002 iface eth0 inet manual
```

INDEX

Symbole

.bin 32

.img 32

A

Allwinner A20 21

Analoge Servos 107

Android 27

apache2 60

Apache-Server vorbereiten 61

apt-get 39

Arch Linux 31

Arduino 121, 132

 ansteuern 121

Arduino-IDE 123

ARM-Architektur 18

ARM-Plattform 17

ARM-Prozessorkern 18

ARMv7 19

ATmega328P 133

audacious 77

Audioausgabe 81

B

Banana Pi 10, 20

 als WLAN-Access-Point 50

 D1 25

 im WLAN 48

 M1 22

 M2 24

 R1 24, 51

 Software installieren 126

Banana Pro 23

bananian-config 37

Bananian Linux 27

Bananian-Projekt 35

bananian-update 38

Betriebssysteme 26, 32

Bootloader 132

bridge-utils 50

C

cifs-utils 73

Clients verbinden 89

cups 56

cups-filters 56

D

Datenbank einrichten 69

dd 34

df 33

dhcp3-server 54

dmesg 83

Download Fritzing-Skizzen 13

Druckerspooler mit CUPS 56

DSL-Router 36

DVB-T-Stick 82

Dynamic DNS 65

E

EPG 88

eth0 49

F

Fedora 31
Flask 115
 absichern 120
Flask-Server 120
Foldersync 67
for-Schleife 98
FreeBSD 31
Fritzing 11
Fritzing-Skizzen 14
FS20-Sender 105
FS20-System 104
FS20-Universalsender 104
fswebcam 112
FTDI-Adapter 31

G

Gehäuse bauen 140
GitHub-Repository 15
gnome-icon-theme 77
GPIO.BOARD 103
GPIO-Breakout 11
GPIO kreativ nutzen 92
GPIO-Leiste 11
GPIO-Pins 11, 93
GPIO.RAW 103
gpio readall 95

H

haveged 50, 53
Heimnetz
 Dateiserver 73
Home-Office-Arbeitsplatz 77
hostapd 50, 52

I

icedove 77
iceweasel 77
ifconfig 46
IoT 30
IP-Adresse 43
 statische 46
iw 48, 50
iwconfig 50

K

Kernelspace 29
KODI 31
Konfiguration
 per LAN 42
 per WLAN 41
Konsole
 serielle 140

L

Lamobo 20
 M1 22
Laufwerke
 Linux 149
LC-Displays ansteuern 134
LEDs 11, 95
LeMaker 77
lemaker.org 20
LeMedia 31
Leuchtdioden 95
libapache2-mod-php5 60
LibreOffice-Anbindung 70
lightdm 77
Linux 33

Linux-Einstieg 151

- Archive 155
- Berechtigungen ändern 153
- Dateien und Ordner 151, 152
- Lange Dateien anzeigen 154
- Manuale 155
- Suchen 154
- Superuser werden 153

M

- Mac OS X 33
- manual 51
- MJPEG-Streamer 114
- mplayer 77
- mysql-client 67
- MySQL-Datenbank 67
- MySQL installieren 68
- mysql-server 67

N

- NetBSD 31
- Netzwerkeinstellung 46
- Netzwerkfreigabe 70

O

- Office nachinstallieren 80
- Olinuxino Micro 25
- OpenBSD 31
- openSUSE 31
- OpenWRT 30, 40
- optional mplayer 112
- Orange Pi 25
 - Mini 25
- ownCloud installieren 62
- ownCloud-Server aufsetzen 60

P

- passwd 44
- php5-curl 60
- php5-gd 60
- php5-imagick 60
- php5-intl 60
- php5-json 60
- php5-mcrypt 60
- php5-sqlite 60
- php-pclzip 60
- php-xml-parser 60
- Plexiglasgehäuse 140
- PowerVR SGX 24
- PPPoE 42
- PuTTY 143
- Python 102
- python-flask 115
- python-serial 104

Q

- Quellcodes 15

R

- Raspberry Pi 18
- Raspbian Linux 28
- reboot 37
- Router mit Bananian 54

S

- samba 73
- samba-common-bin 73
- Samba-Server 74
- samba-tools 73
- screen 143
- SD-Karte 32

SD-Kartenleser 32
Serielle Konsole 140
Servos 107
Shenzhen 20
Single-Board-Computer 16
Sinovoip 20
sinovoip.com 20
Smartphone-Clients 66
smbclient 73
Snappy Core 30
Software installieren 94
SQLite 61
SSH-Log-in 146
ssid 48
SSID 53
SSL 66
Standardnetzwerkeinstellung 46
Statische IP-Adresse 46
Steckbrett 95
SysFS 99

T

tango-icon-theme 77
TFTP 17
Treiber
 nl80211 52
 USB-Webcam 114
TVheadend installieren 84

U

Überwachungskamera 115
Ubuntu 44
 Linux 29
Ubuntu-Images 44
USB-Webcam 114
Userland 29

V

Viderekorder 82
Videstreamer 82
vi-Editor 148
Virtual Hosts 66

W

Watterott RPi Shield Bridge 122
Webcam anschließen 112
Win32 Disk Imager 32
WLAN-Brücke 53
WLAN-Konfiguration 53
WPA-Schlüssel 53
wpa_supplicant 48

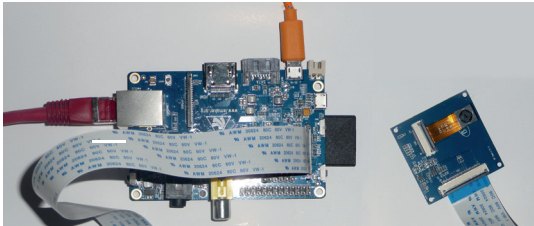
X

xfce4 77
XFCE-Desktop 78
xfonts-75dpi 77
xfonts-100dpi 77
xfonts-scalable 77
xserver-xorg 77

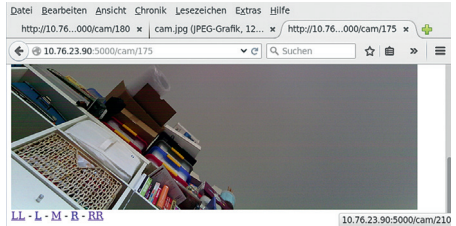
SCHNELLEINSTIEG BANANA PI

ALLE WICHTIGEN
BOARDS
BPI-M1, PRO,
BPI-R1, BPI-M2
UND BPI-D1

Auch wenn der Raspberry Pi als Inbegriff des Single-Board-Computers gilt: Es gibt mehr als nur den Pi – ob als Ein-, Um- oder Aufsteiger, es lohnt sich der Blick auf den Banana Pi. Schon der Name weist auf die Ähnlichkeiten hin, d. h., vorhandene Projekte können Sie gut auch portieren. Profitieren Sie von der besseren Hardware!



Nicht nur die Boards werden erklärt, sondern auch die praktische Nutzung von Zubehör.



Wenig Theorie, dafür viel Praxis: dokumentiert mit Quellcode, Schaltbildern und Screenshots.

Richtiges Board und Betriebssystem wählen, installieren und programmieren

Es gibt nicht den einen Banana Pi. Lernen Sie die unterschiedlichen Boards kennen und wählen Sie das richtige für sich aus. Bananian, OpenWRT oder Ubuntu? Egal welches Sie wählen, die Installation wird Ihnen Schritt für Schritt erklärt. Für die Programmierung nehmen Sie am besten Python. Wie diese Programmiersprache installiert und genutzt wird, wird anhand eines eigenen Projekts gezeigt.

Mit Praxisprojekten den Banana Pi ausreizen

Bananian als Betriebssystem ist sehr mächtig. Nach der richtigen Einbindung im Netzwerk nutzen Sie Ihren Banana Pi als Druckerspooler und wandeln PostScript vor dem Ausdruck in das binäre Format des Druckers um. Cloudserver und Desktopersatz sind genauso dabei wie Dateiserver und Videorekorder. Die Umsetzung von Projekten an der GPIO lernen Sie auch anhand von Praxisbeispielen kennen. Webcam oder Servomotor? Zwei Beispiele zeigen Ihnen, wie es geht.

Aus dem Inhalt:

- Banana Pi-Boards im Detail
- Unterstützte Betriebssysteme
- Bananien, OpenWRT und Ubuntu installieren
- Netzwerkkonfiguration: statische IP, WLAN und Router
- Druckerspooler mit CUPS
- ownCloud-Server aufsetzen
- MySQL-Datenbank fürs Büro
- Datenserver im Heimnetz
- Banana Pi als Desktopersatz
- Videorekorder und -streamer
- GPIO nutzen: Python, Steckbrett und Servomotoren
- Webcam anschließen und Arduino™ ansteuern



Besuchen Sie
unsere Website
www.franzis.de

FRANZIS