All-in-One-OrderSprinter

Installation auf dem Raspberry Pi

(Raspberry Pi Model 3B+)



Stefan Pichel

© 2020 Stefan Pichel

Bei der Zusammenstellung der Texte und Abbildungen wurde mit größter Sorgfalt vorgegangen. Trotzdem können Fehler nicht vollständig ausgeschlossen werden. Weder Autor noch Verlag können für Schäden haftbar gemacht werden, die in Zusammenhang mit der Verwendung dieses Buches stehen. Für Verbesserungsvorschläge und Hinweise auf Fehler ist der Autor dankbar.

In diesem Buch werden eingetragene Warenzeichen, Handelsnamen und Gebrauchsnamen verwendet. Auch wenn diese nicht als solche gekennzeichnet sind, gelten die entsprechenden Schutzbestimmungen.

Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten, auch die der fotomechanischen Wiedergabe und der Speicherung in elektronischen Medien. Nachdruck und Vervielfältigung von Text und Abbildungen, auch auszugsweise, nur mit ausdrücklicher Genehmigung des Autors.

Webseite: www.ordersprinter.de

E-Mail: s.pichel@ordersprinter.de

Inhaltsverzeichnis

1	Inst	allation				
	1.1	Vorbemerkung	5			
	1.2	Die SD-Karte	6			
	1.3	Betriebssystem downloaden	7			
	1.4	Der erste Start	7			
		1.4.1 Installationsvarianten	.0			
		1.4.2 Apache und Datenbank	.1			
		1.4.3 PHP-Erweiterungen	2			
		1.4.4 lightpd	.3			
		1.4.5 Php-Module und -Einstellungen	.4			
		1.4.6 Größe der Logfiles 1	.4			
	1.5	Anlegen der Datenbank	5			
	1.6	OrderSprinter Dateien übertragen 1	5			
	1.7	OrderSprinter installieren 1	.6			
	1.8	Drucker einrichten	.7			
		1.8.1 Manueller Start des Java-Printers	8			
		1.8.2 Autostart vom Javaprinter	20			
	1.9	Raspi ausschalten	!1			
	1.10	Datum einstellen	22			
	1.11	WLAN	23			
		1.11.1 Raspi am Router	23			
		1.11.2 Raspi als Access Point	24			
		1.11.3 Raspi 3 als Access Point	30			
	1.12	Regelmäßiges Backup	33			
	1.13	Anschluss eines Touch-Displays	35			
	1.14	Dateisystemcheck	36			

INHALTSVERZEICHNIS

Kapitel 1

Installation

1.1 Vorbemerkung

Der Raspberry Pi ist ein Gerät mit sehr begrenzter Leistungsfähigkeit und selbst die Version 3B+, welche zum Erstellungszeitpunkt dieser Anleitung das neuste Modell ist, kann niemals einem aktuellen Desktop-PC das Wasser reichen! Daher eignet sich der Raspi gut für erste Tests oder sehr kleine Betriebe, in denen die Geschwindigkeit nicht so maßgebend ist. In allen anderen Fällen sei empfohlen, einen leistungsfähigeren Rechner als Server zu verwenden, z.B. mit SSDs ausgestattete NUC Mini-Computer (wenn der geringer Platzbedarf wichtig ist), oder beispielsweise ein Touch-Notepad, wenn man eine stationäre Kasse aufbauen möchte.

Damit der Raspi trotz seiner leistungsschwachen Hardware bedienbar bleibt, sollte man darauf achten, eine sehr schnelle SD-Karte zu verwenden, da gerade die Lese- und Schreiboperationen einen großen Einfluss auf die Gesamtperformanz ausüben.

Ein sehr wichtiger Punkt ist auch die Verwendung eines leistungsstarken Netzteils, insbesondere wenn noch Peripherie wie Bondrucker oder WLAN-Antenne an den USB-Ports hängen. Die Benutzung von Netzteilen mit weniger als 2,5A ist nicht empfehlenswert, denn sie werden zu einem sehr instabilen System führen! Wenn der Raspi keinen Zugriff auf das Internet bekommen soll, wird er die Uhrzeit nicht beim vorkonfigurierten NTP-Server abfragen können. Zwar besitzt OrderSprinter ein Feature, welches die komfortable Zeit- und Datumseinstellung nach dem Start des Raspis bietet. Wer aber noch bequemer haben möchte, der stattet den Raspi mit einer batteriegepufferten Uhr aus. Diese als Aufsteckzubehör erhältlichen sogenannten RTCs kosten nur wenige Euros.

Diese Anleitung wurde erstmals 2018 erstellt und enthält einige Erweiterungen (z.B. Abschnitt Touch-Display). Es wurden aber nicht alle Stellen ständig aktualisiert. Zum Beispiel enthalten neuere Raspbian-Abbilder mittlerweile aktuelle PHP-Pakete, so dass anders an in dieser Anleitung beschrieben nicht mehr die Versionsnummer 7.2 bei der Paketinstallation angegeben werden muss.

1.2 Die SD-Karte

Der Raspberry Pi holt sich sein Betriebssystem von einer SD-Karte, die in einem *normalen* PC vorbereitet werden muss. Da Raspi-Nutzer eher Linuxaffin sind, wird hier die Vorgehensweise für Linux-Systeme beschrieben. Zunächst einmal wird eine SD-Karte in den PC eingeschoben und nicht gemounted. In diesem Beispiel hat die SD-Karte eine Kapazität von 8GB. Ein Aufruf von *gparted* zeigt die SD-Karte (Abb. 1.1) und als welches Gerät sie ansprechbar ist.



Abb. 1.1: GParted zeigt an, als welches Device sich die Speicherkarte im Rechner angemeldet hat.

Die letzte Kennung p1 steht für Partition 1 und je nach vorheriger Verwendung der SD-Karte kann es weitere Partitionen p2, p3 usw. geben. Wenn man nicht nur eine Partition, sondern die gesamte SD-Karte beschreiben muss, muss man also als Ziel in diesem Beispiel /dev/mmcblk0 angeben. Wie man das macht, wird im nächsten Abschnitt beschrieben.

1.3 Betriebssystem downloaden

Beim Raspi hat man die Auswahl verschiedenster Betriebssysteme und deren unterschiedlichen Versionen. In dieser Anleitung wird auf die Installation von Raspbian Stretch Lite eingegangen. Siehe hierzu

https://www.raspberrypi.org/downloads

Hier lässt sich die Software bequem downloaden und auch Anleitungen finden sich hier. Der Download hat eine Größe von etwa 351 MB und kann je nach Qualität der Internetanbindung ein wenig Zeit beanspruchen. Das heruntergeladene Image muss entpackt werden. Heraus kommt eine 1781MB große Image-Datei (in diesem Fall 2018-10-09-raspbian-stretchlite.img), welche nun auf die Speicherkarte übertragen werden kann. Unter Linux kann dazu das *dd*-Kommando verwendet werden. Unter Windows gibt es dedizierte Software zum Übertragen des Images auf die SD-Karte, hier sei auf die Anleitung auf der Webseite zum Raspberry-Pi verwiesen.

sudo sudo dd bs=4M if=2018-10-09-raspbian-stretch-lite.img of=/dev/m

Auch dieser Vorgang wird je nach Schreibgeschwindigkeit etwa eine Viertelstunde in Anspruch nehmen, weil nun knapp 2 GB auf die SD-Karte übertragen werden müssen. Anschließend sollte eine Statusmeldung ähnlich dieser erscheinen:

445+0 Datensätze ein 445+0 Datensätze aus 1866465280 bytes (1,9 GB, 1,7 GiB) copied, 418,93 s, 4,5 MB/s

Mit GParted kann man sich anschauen, wie die SD-Karte beschrieben wurde (Abb. 1.2). (In einem späteren Schritt wird beschrieben, wie die nutzbare Partition auf den nicht-allokierten Bereich erweitert wird.)

1.4 Der erste Start

Seit dem Update im November 2016 ist SSH nicht mehr voreingestellt aktiviert. Damit später ein Einloggen per SSH möglich ist, muss SSH einmalig

GParted Bearbeit	en Ansicht Gerä	t Partition Hill	Partition Hilfe				
• = ->		~		dev/r	/dev/mmcblk0 (7.21 GiB) 🗸		
/dev/mr 1.69 Git	ncblk0p2			nicht zugeteilt 5.48 GiB			
Partition	Dateisystem	Bezeichnung	Größe	Benutzt	Unbenutzt	Markierungen	
Partition nicht zugeteilt	Dateisystem	Bezeichnung	Größe 4.00 MiB	Benutzt	Unbenutzt	Markierungen	
Partition nicht zugeteilt /dev/mmcblk0p1	Dateisystem nicht zugeteilt fat32	Bezeichnung boot	Größe 4.00 MiB 43.80 MiB	Benutzt 22.66 MiB	Unbenutzt 21.14 MiB	Markierungen	
Partition nicht zugeteilt /dev/mmcblk0p1 /dev/mmcblk0p2	Dateisystem nicht zugeteilt fat32 ext4	Bezeichnung boot rootfs	Größe 4.00 MiB 43.80 MiB 1.69 GiB	Benutzt 22.66 MiB 1008.92 MiB	Unbenutzt 21.14 MiB 723.08 MiB	Markierungen Iba	

Abb. 1.2: GParted zeigt an, wie die SD-Karte beschrieben wurde.

aktiviert werden. Die Boot-Partition der SD-Karte muss dazu gemounted werden. Unter Linux geht das per

```
sudo mount /dev/mmcblk0p1 /mnt
```

Im Wurzelverzeichnis der Boot-Partition muss eine leere Datei namens *ssh* erstellt und die Partition wieder ausgehängt werden. Dadurch wird der SSH-Zugriff automatisch aktiviert, wenn der Raspi später gestartet wird.

sudo echo "" > /mnt/ssh
sudo umount /mnt

Die Karte kann nun dem PC entnommen und in den (noch ausgeschalteten!) Raspi eingesteckt werden. Der Raspi wird nun per LAN-Kabel mit dem eigenen Router verbunden, über den auch das Internet erreicht werden können muss, damit in den nächsten Schritten weitere Pakete zur Installation nachgeladen werden können. Erst jetzt wird das Netzteilkabel in den Raspi eingesteckt und der Raspi somit mit der Stromversorgung verbunden.

In der Router-Oberfläche sollte das neue Gerät nun mit einer zugewiesenen IP-Adresse aufgeführt sein (Abb. 1.3). In diesem Fall lautet die IP-Adresse 192.168.178.35.

9	raspberrypi	💑 LAN 4 mit 1 Gbit/s	192.168.178.55

Abb. 1.3: Die Fritz!Box zeigt den Raspi als Gerät in der Netzwerkansicht. Er hat die IP-Adresse 192.168.178.35 bekommen.

Über ssh kann man sich auf den Raspi einloggen. Das Password ist *raspberry*. Anschließend kommt man in die Konfiguration über das *raspi-config*-Kommando (als Administrator ausführen, d.h. die Kommandozeile heißt *sudo raspi.config*):

```
> ssh pi@192.168.178.55
The authenticity of host '192.168.178.55 (192.168.178.55)' can't be established.
ECDSA key fingerprint is SHA256:FS9JBbMNoBEMD8eg1kJrSH17aRjigbG20NPAX8InUBg.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.178.55' (ECDSA) to the list of known hosts.
pi@192.168.178.55's password:
Linux raspberrypi 4.14.71-v7+ #1145 SMP Fri Sep 21 15:38:35 BST 2018 armv71
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law
SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set a new password.
Wi-Fi is disabled because the country is not set.
Use raspi-config to set the country before use.
sudo raspi-config
```

Nach dem Aufruf von raspi-config erscheint eine Einstellungsoberfläche (Abb. 1.4), über die grundsätzliche Parameter festgelegt werden können.

Raspberry <u>1 Change User Password</u> 2 Network Options 3 Boot Options 4 Localisation Options 5 Enterfaciae Options	Pi Software Configuration Tool (raspi-config) Change password for the current user Configure network settings Configure options for start-up Set up language and regional settings to match your location Configure negrections on explosions
6 Overclock 7 Advanced Options 8 Update 9 About raspi-config	Configure convertions to be typicals Configure advanced settings Update this tool to the latest version Information about this configuration tool
<select< th=""><th><pre><finish></finish></pre></th></select<>	<pre><finish></finish></pre>

Abb. 1.4: Raspi-Config Oberfläche

Damit die gesamte Kapazität der SD-Karte verwendet werden kann, sollte die Datenpartition zunächst erweitert werden. Dazu wählt man Advanced Options und darunter Expand Filesystem aus.

Eine wichtige Einstellung ist auch die Zeitzone: Unter *Localization Options* sollte die Zeitzone per *Change Timezone* korrekt festgelegt werden. Diese Einstellung ist erforderlich, damit beim Systemstart die richtige Uhrzeit vom Zeitserver aus dem Internet geholt werden kann. Unverändert würde die UTC-Zeit verwendet!

Unter Network Options lässt sich auch der Hostname einstellen. Das ist der Gerätename, unter dem der Raspi im Netz angesprochen werden kann. Wenn man den Namen verwendet, muss man sich nicht merken, unter welcher IP-Adresse der Router den Raspi eingespeichert hat. In diesem Beispiel wird der Name raspi gesetzt.

Da die neueren Raspberry Pi einen WLAN-Adapter besitzen, sollte man an dieser Stelle auch die Verbindung zum Router festlegen.

Die anderen Einstellungen können prinzipiell unverändert übernommen werden.

Nach Abschluss der Einstellungen wird man als Benutzer gefragt, ob der Raspi neu gestartet werden soll. Der Neustart ist notwendig, damit die Einstellungen wirksam werden.

Nach dem Neustart und dem erneuten Einloggen über sch sollte die Installation auf den neuesten Stand gebracht werden:

```
sudo apt-get update
sudo apt-get -y dist-upgrade
```

1.4.1 Installationsvarianten

Der Raspi kann so aufgesetzt werden, dass er ein All-in-One-System ist, d.h. der OrderSprinter-Webservice sowie der Java-Printerserver werden auf dem gleichen Raspi nstalliert.

Alternativ kann man natürlich den Raspi auch als reinen Druckerserver verwenden, z.B. um Drucker, die weit entfernt vom OrderSprinter-Server aufgestellt wurden, anzubinden. In dem Fall holt sich der Java-Printserver die Druckdaten vom OrderSprinter-Webserver und leitet sie an den Drucker weiter, den an ihm über USB angeschlossen ist. Auf diese Weise kann jeder Drucker *seinen* Raspi haben, doch ist dabei darauf zu achten, dass die Konfigurationen der Printserver-Instanzen so eingestellt werden, dass es keine Überschneidungen gibt (zwei Druckerserver fragen die für die gleichen Drucker ab). Für die Einrichtung als reinen Druckerserver kann man die folgenden Abschnitte überblättern und direkt zum Kapitel 1.8 gehen.

1.4.2 Apache und Datenbank

Der OrderSprinter benötigt einen Webserver und eine Datenbank. Beim Webserver hat man eine große Auswahl. Im diesem Abschnitt werden der Apache und lighttpd vorgestellt.

Da die alten 5.x-Versionen von PHP nicht mehr unterstützt werden, sollte man auch auf dem Raspi eine aktuelle 7.x-Version von PHP installieren. Für die Installation von PHP 7.2 muss jedoch die Paketquellenkonfiguration umgeändert werden. Dazu muss man mit einem Editor wie vi oder nano die Datei /etc/apt/sources.list abändern, indem die Zeile mit dem Verweis auf stretch einkommentiert und eine Zeile mit Verweis auf die buster-Pakete eingefügt buster wird (Abb. 1.5).



Abb. 1.5: Apt-Get-Sources auf die nächste stabile Version umändern

Anschließend muss man die Pakete auf den buster-Zweig updaten. Dies ist ein sehr zeitaufwändiger Schritt, da sämtliche Pakete aktualisiert werden.

sudo apt-get update && sudo apt-get upgrade

Während des Update-Vorgangs kann es sein, dass Changelogs angezeigt werden, insbesondere wenn man schon etwas weitergelesen und den Apache installiert hat. Mit einem Druck auf die q-Taste und Return verlässt man die Anzeige. Außerdem wird man gefragt, ob man Änderungen für die Samba-Konfiguration vornehmen möchte, dies kann man verneinen. Die Frage, ob man Services automatisch restarten möchte, kann man mit Yes bestätigen.

Im nächsten Schritt kann man eventuelle alte PHP-Pakete per

```
sudo apt-get remove '^php.*'
```

deinstallieren. Wenn man jedoch ein frisches System hat, sollten noch keine deinstallierbaren PHP-Pakete vorhanden sein.

```
sudo apt-get -y install php7.2
```

Mit diesem Installationskommando werden automatisch diese Pakete mit installiert: libapache2-mod-php7.2 libargon2-1 libgpm2 libncurses6 libsodium23 libtinfo6 php-common php7.2 php7.2-cli php7.2-common php7.2json php7.2-opcache php7.2-readline.

Der Apache ist eher ein Schwergewicht unter den Webservern. Er ist sehr umfangreich mit einer riesigen Funktionalität. Der Apache-Webserver wird so installiert:

```
sudo apt-get -y install apache2 apache2-doc apache2-utils
sudo apt-get -y install libapache2-mod-php7.2
sudo usermod -a -G www-data www-data
```

Jetzt gehts zur Installation der Datenbank bzw. der Möglichkeit, mit PHP auf diese zuzugreifen:

```
sudo apt-get -y install php7.2-mysql
sudo apt-get -y install mariadb-server mariadb-client
```

Die Installation erwartet die Vergabe eines Administratorpasswortes für den MySQL-Root-Zugang.

1.4.3 PHP-Erweiterungen

Es ist wichtig, dass einige PHP-Erweiterungen installiert sind. Auf die wichtigsten Erweiterungen wird im Installationsskript geprüft: gd, mysqli, openssl, pdo_mysql, PDO, session, zlib. Diese Erweiterungen lassen sich nachinstallieren, z.B. für die GD-Bibliothek über

sudo apt-get -y install php7.2-gd php7.2-curl php7.2-zip php7.2-xml

Damit die Änderungen im Webserver Apache angewandt werden, muss man den Apache neu starten:

```
sudo /etc/init.d/apache2 restart
```

1.4.4 lighttpd

lighttpd ist eine leichtgewichtige Alternative zum Apache. Bei vielen gleichzeitigen Zugriffen dürfte er auf dem Raspi wesentlich performanter arbeiten, weil er weniger Hardware-Ressourcen benötigt. Außerdem zeichnet er sich durch eine leichte Konfigurierbarkeit aus. Die Installation erfolgt so:

```
sudo apt-get -y install lighttpd
sudo apt-get -y install php7.2-common php7.2-cgi php7.2
```

Wenn bereits der apache installiert ist, wird der Start von lighttpd nicht funktionieren, weil der gleiche Standard-Port belegt ist (Port 80). Das Problem wird später beseitigt.

Den Apache wird man so los:

```
sudo apt-get --purge remove apache2
sudo apt-get autoremove
```

Mit diesen Kommandos wird der Apache nur deinstalliert, der Port ist aber noch geblockt. Nach einem Reboot des Raspis startet aber wie geplan der lighttpd.

Damit lighttpd PHP-Code ausführen kann, muss er entsprechend konfiguriert werden:

sudo lighty-enable-mod fastcgi-php

Die Standard-Konfigurationsdatei /etc/lighttpd/lighttpd.conf sollte bereits so konfiguriert sein, dass die Websourcen aus /var/www geholt werden¹, so dass hier keine weiteren Änderungen vorzunehmen sind. Nach einem Restart des lighttpd sollte die Weboberfläche des Raspis wieder funktionieren:

sudo service lighttpd restart

 $^{^1\,{\}rm Je}$ nach Debian-Version kann das Standardverzeichnis tiefer unterhalb /var/www liegen.

1.4.5 Php-Module und -Einstellungen

Damit die Änderungen wirksam werden, muss der Webserver wieder neu gestartet werden:

```
sudo service lighttpd restart
```

Die PHP-Standardkonfiguration ist so gestaltet, dass PHP auf möglichst allen Rechnern laufen soll. Diese Einstellungen sind jedoch nicht optimal für OrderSprinter. Daher sollte die Datei php.ini mit einem Texteditor (z.BV. vi oder nano) bearbeitet werden:

sudo vi /etc/php/7.2/apache2/php.ini

Bei einer SD-Karte von 8GB sollten folgende Werte eingestellt werden:

max_execution_time = 3000
post_max_size = 5000M
upload_max_filesize = 5000M
memory_limit = 7500M

Die Werte für post_max_size und upload_max_filesize müssen an die Größe des unbenutzten Platzes der Datenpartition der SD-Karte angepasst werden. Die unbenutzte Größe erfährt man über

df -m

(Eintrag für Mounted on /").

Nach einer Änderung der PHP-Einstellungen muss der Webserver neu gestartet werden, damit die Werte übernommen werden!

1.4.6 Größe der Logfiles

Die Clients regelmäßig mit dem Webserver kommunizieren, fallen in der Regel eine Menge Logdaten an. Da so eine SD-Karte nur eine sehr begrenzte Kapazität besitzt, sollte darauf geachtet werden, die Logfiles regelmäßig zu löschen, oder Möglichkeiten wie *logrotate* einzusetzen.

1.5 Anlegen der Datenbank

Die Datenbanksoftware (MariaDB) wurde bereits in einem früheren Schritt installiert. MariaDb ist eine zu MySQL kompatible Datenbanksoftware, die viele Anwender bevorzugen.

Damit OrderSprinter eine Datenbank nutzen kann, muss diese erst einmal angelegt werden. Das geht auf der Kommandozeile sehr einfach.

sudo mysql -u root -p
mysql> CREATE DATABASE ordersprinter
 -> DEFAULT CHARACTER SET utf8
 -> DEFAULT COLLATE utf8_general_ci;
Query OK, 1 row affected (0.00 sec)
mysql> GRANT ALL ON ordersprinter.* TO os@localhost IDENTIFIED BY "secret";

Es wurde nun eine Datenbank mit dem Namen *ordersprinter* angelegt, auf die der Benutzer *os* mit dem Passwort *secret* alle Rechte besitzt. Sicherheitsbewusste Anwender mögen die Rechte einschränken.

Um die Kommandozeilenebene der Datenbank zu verlassen, kann man das Kommandoquitoder die Tastenkombination Strg-D verwenden.

1.6 OrderSprinter Dateien übertragen

Das Übertragen der OrderSprinter-Download-Datei (hier sei angenommen, dass sie als *ordersprinter.tar* gespeichert wurde) von einem anderen PC aus kann nun so erfolgen (Kommando nicht auf dem Raspi eingeben, sondern auf dem PC. Die IP-Nummer des Raspis verwenden!):

```
scp ordersprinter.zip pi@192.168.178.35:/tmp
```

Die Datei ordersprinter.zip wird damit in das Verzeichnis /tmp kopiert (alternativ kann die OrderSprinter-Software auch über den wget-Befehl direkt vom Raspi downgeloaded werden). Es kann von dort aus entpackt und der webapp-Teil nach /var/www kopiert werden².

²Je nach Betriebssystemversion kann der Standardpfad auch /var/www/htdocs sein!

sudo bash
cd /tmp
unzip ordersprinter.zip
rm /var/www/html/index.html
cp -R webapp/* /var/www/html/

Wichtig ist, das Unterverzeichnis php und die Date
i $\rm php/config.php$ beschreibbar zu machen:

```
root@raspi:/tmp# chmod 777 /var/www/html/php
root@raspi:/tmp# chmod 777 /var/www/html/php/config.php
```

1.7 OrderSprinter installieren

Die Vorarbeiten sind erledigt. Jetzt sollte OrderSprinter über den normalen Weg installiert werden können: Wenn man als Hostname den Namen raspi vergeben hat, sollte nach dem Aufruf der Adresse http://raspi im Webbrowser die Installationseinstiegsseite erscheinen (Abb. 1.6).

Für die weiteren Schritte der OrderSprinter-spezifischen Installation wird auf die Webseite vom OrderSprinter verwiesen.



Abb. 1.6: Installationsseite vom OrderSprinter

Es sollten während der Installation keine Warnungen über nicht installierte PHP-Erweiterungen mehr erscheinen und wenn man die richtigen Datenbankparameter eingetragen hat, wird auch der Zugang zur Datenbank als *erfolgreich* markiert (Abb. 1.7).

Die Installation kann bei langsamen SD-Karten bis zu einer halben Stunde in Anspruch nehmen!

Kategorie	Einstellung	Wert		
	Datenbank-Server:	localhost		
	Datenbank-Name:	ordersprinter		
Dutachard	Datenbank-Benutzer:	os		
Datenbank	Datenbank-Passwort:			
	Präfix für die Tabellen:	pos_		
	Teste DB-Zugriff	ок		
	php-Verzeichnis	ок		
Schreibberechtigung	config.php	ок		
	Teste Schreibberechtigung	ОК		
	Administrator-Passwort:			
	Administrator-Passwort (Wdh.):			
Applikation	Währung:	Euro		
	Dezimaltrennzeichen:	Komma 🗸		
PHP-Status		OK!		
	Update -> 1.5.17			
Starte In	stallation			

Abb. 1.7: Konfigurationseinstellungen

1.8 Drucker einrichten

In diesem Beispiel wird vorgestellt, wie man über *Serverprint* Bondrucker direkt am Raspi anschließen kann (der Modus *Serverprint* ist die Standardeinstellung beim OrderSprinter).

Damit der Java-Printserver lauffähig ist, muss Java 8 (oder neuer) installiert werden, das geht über folgendes Kommando:

sudo apt-get update && sudo apt-get -y install oracle-java8-jdk

Anschließend ist der Server-Print im OrderSprinter zu aktivieren, indem in der Administrationsansicht der Druckmodus auf *Server* eingestellt werden muss (ist mittlerweile der Standardmodus) und ein Printcode eingestellt wird. In diesem Beispiel sei dies *123*, welches sich später in der Javaprinter-Configurationsdatei wiederfinden muss.

Das Ziel ist es, dass jeder beliebige USB-Bondrucker an den Raspi an-

geschlossen werden kann und Druckaufträge vom OrderSprinter entgegennimmt. Das Kommando *lsusb* zeigt alle USB-Devices und USB-Hubs an, die vom Betriebssystem gesehen werden. Wenn man einen USB-Bondrucker nun anschließt, sollte dieser auch in der Liste erscheinen (Abb. 1.8). Eine spezielle Druckertreiberinstallation ist nicht erforderlich!

> pi@raspberrupi /var/www \$ lsusb Bus 001 Device 002: ID 0424:9514 Standard Microsystems Corp. Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub Bus 001 Device 003: ID 0424:e500 Standard Microsystems Corp. pi@raspberrupi /var/www \$ lsusb Bus 001 Device 002: ID 0424:9514 Standard Microsystems Corp. Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub Bus 001 Device 003: ID 0424:e500 Standard Microsystems Corp. Bus 001 Device 003: ID 0424:e5584 WinChipHead CH34x printer adapter cable

Abb. 1.8: Das Kommando *lsusb* zeigt nach dem Einstecken des Bondruckers in einen USB-Port einen neuen Eintrag. So sieht das System den Bondrucker.

1.8.1 Manueller Start des Java-Printers

OrderSprinter enthält einen JavaPrinter (im Unterverzeichnis *javaprinter* des Downloads), der in diesem Beispiel nun nach */home/stefan/javaprinter* kopiert wurde (Abb. 1.9). Der Javaprinter erfordert eine Java VM mindestens in der Versin 1.8, die im vorangegangenen Abschnitt installiert wurde.

<pre>pi@raspberrypi /home/stefan/javaprinter \$ ls -ltr total 48</pre>						ar\$ls −ltr	
drwxr-xr-x	2 pi	рi	4096	Jul	12	20:40	.javaprinter_lib
-rw-rr	1 pi	pi	39835	Jul	12	20:40	javaprinter.jar
-rw-rr	1 pi	pi	417	Jul	12	20:40	config.json

Abb. 1.9: Inhalt des Javaprinter-Verzeichnisses

In diesem Unterabschnitt wird erklärt, wie man den Javaprinter manuell starten kann. Erst wenn der Java-Printer in diesem Fall korrekt arbeitet, sollte er als Dienst eingerichtet werden, der direkt nach dem Einschalten startet (Abschnitt 1.8.2).

Die dem javaprinter beiliegende Config-Datei sollte in etwa so aussehen wie in Abb. 1.10.

Die beigelegte config.json kann angepasst werden:

• *instance*: Instanznummer des Printservers. Jeder Printserver sollte eine unterschiedliche Instanznummer haben. In der OrderSprinter-Administrationsansicht lässt sich unter *Druckerinstanzen* einstellen, welche Jobtypen an welche Printserverinstanz geschickt werden soll.

```
{
    "instance" : 1,
    "vendorid" : "4348",
    "productid" : "5584",
    "printersize" : 32,
    "printcode" : "123",
    "baseurl":"http://localhost",
    "basepass" : "",
    "basepass" : [ 27, 64, 27, 116, 0 ],
    "escinits" : [ 27, 64, 27, 116, 0 ],
    "useeveryprintdevice" : 1,
    "verbose_closing_summary" : 1,
    "smallformat" : 0,
    "logoscale" : 1.0
}
```

Abb. 1.10: Inhalt der Datei config.json

- *printersize*: maximale Anzahl der Zeichen pro Zeile, die der Drucker ausgeben kann
- *printcode*: Printcode, wie in der Administrationsoberfläche vom OrderSprinter angegeben ist. Wurde kein Printcode vergeben, können keine Druckjobs ausgegeben werden.
- *logoscale*: Skalierung des Logos auf dem Kassenbon. Mit diesem Wert kann die Abbildung vergrößert oder verkleiner werden.
- *baseurl*: Basis-Webadresse des Webservers, auf dem OrderSprinter läuft: In diesem Fall sollen Javaprinter- und OrderSprinter-Service beide auf dem Raspi laufen, so dass die Einstellung *localhost* richtig ist. Der Javaprinter kann jedoch auch OrderSprinter-Instanzen im Internet bedienen, dann müsste die URL z.B. http://www.meinrestaurant.de/ordersprinter heißen.
- *baseusername* und *basepass*: Ist der OrderSprinter-Webserver über BasicAuthentication geschützt, so können hier Benutzername und Passwort angegeben werden.
- *smallformat*: Ist der Wert auf 1 gesetzt, so werden Arbeitsbons in etwas kompakterer Form gedruckt.

Die restlichen Parameter können normalerweise so belassen werden. Man kann hiermit beispielsweise Drucker-spezifische Bytesequenzen vor und nach einem Druckauftrag senden und bestimmen, welche Drucktypen (Kassenbons, Arbeitsbons Speisen/Getränke, Tageslosungen) ausgedruckt werden sollen.

Mit dem Kommando

sudo java -jar javaprinter.jar -configfile=/home/stefan/javaprinter/config.json

lässt sich der JavaPrinter starten (Abb. 1.11). Es gibt mittlerweile auch die Möglichkeit, Drucker über CUPS anzusprechen - siehe hierzu die ODF-Bedienungsanleitung.

p<mark>l@rasiberrupi /home/stefan/javaprinter</mark> * java -jar javaprinter.jar /home/stefan/javaprinter/config.json Read: /home/stefan/javaprinter/config.json Config: Config [printersez32, vendorid=4348, productid=5584, currency=Euro, decpoint=,, companyinfo=Musterrestaurant

Beispielstrasse 123 12345 Musterort, escinits=[27, 64, 27, 116, 0], escposts=[27, 64], printcode=123, baseurl=http://localhost, baseusername=, basepass=]

Abb. 1.11: Der Start des Java-Printers

Das vorangestellte sudo ist erforderlich, weil ohne weitere Systemkonfiguration nur root direkt auf USB-Ports schreiben darf. Soll der Dienst später unter einem anderen Benutzer als root laufen, so muss im Verzeichnis /etc/udev/rules.d eine Datei 50-usb-stefan.rules (der Datei muss mit .rules enden!) angelegt werden. Diese sollte etwa folgenden Inhalt haben:

SUBSYSTEM=="usb", ATTR{idVendor}=="4348", ATTR{idProduct}=="5584", MODE="0666", GROUP="users"

idVendor und *idProduct* des Druckers können per *lsusb* herausgefunden werden. Alle Benutzer in der Gruppe users können anschließend auf den USB-Port zum Drucker mit der angegebenen VendorId und ProductID schreiben.

Der JavaPrinter scant alle USB-Ports und greift sich den ersten Drucker, den er finden kann. Hat man mehrere Drucker angeschlossen, muss man konkret die ProductId und VendorId des Druckers in der Datei config.json eintragen, sowie dort die Einstellung *useeveruprintdevice* auf 0 setzen.

Autostart vom Javaprinter 1.8.2

Das Ziel sollte es sein, dass der Raspi nur eingeschaltet werden muss und der Javaprinter ohne weiteres Zutun des Benutzers automatisch startet.

Der Javaprinter sollte zunächst in ein sinnvolles Verzeichnis kopiert werden, z.B. bietet sich /usr/local/bin an. Im Folgenden wird davon ausgegangen, dass die Dateien config.json und javaprinter.jar sowie das Verzeichnis javaprinter lib dort liegen.

Es gibt verschiedene Möglichkeiten, Programme nach dem Einschalten zu starten. Hier wird nur die Variante über die Bearbeitung der Datei /etc/rc.local vorgestellt. Vor der letzten Zeile mit der *exit*-Anweisung muss folgende Zeile eingefügt werden:

su root -c '/usr/bin/java -jar /usr/local/bin/javaprinter.jar -configfile=/usr/local/bin/config.json'

1.9 Raspi ausschalten

Damit der Raspi geregelt ausgeschaltet werden kann, kann man auf der Weboberfläche in der Administrationsansicht als Benutzer mit Administrator-Rechten den Server ausschalten lassen (Abb. 1.12). Um Missbrauch zu vermeiden, darf normalerweise nicht jeder Benutzer das Linux-Betriebssystem herunterfahren. Diese Restriktion gilt auch für den Benutzer, unter dem der Webserver betrieben wird. Man kann jedoch das S-Bit vom shutdown-Programm setzen, so dass das möglich wird:

sudo chmod u+s /sbin/shutdown

🖨 Datenbank					
Speisekarte					
Raumplan					
Sicherung und Import					
	Server herunterfahren				
(Herunterfahren nur	möglich, wenn Webserver die erforderlichen Rechte besitzt).				
0	Download Printserver				



Es sei angemerkt, dass es bessere Alternativen gibt, Benutzern Ausführungs-Rechte auf Systemprogrammen einzuräumen (z.B. Visudo), aber für das Ausschalten des Raspis sollte die hier vorgestellte Lösung reichen.

Ist die grüne LED am Raspi erloschen, so ist der Raspi aus.

1.10 Datum einstellen

Der Raspberry Pi hat von Hause keine eingebaute Echtzeituhr. Nach dem Abschalten verliert er seine Uhrzeit. Nach dem Einschalten möchte sich der Raspi die aktuelle Zeit von einem Zeitzserver holen. Ist dieser jedoch nicht erreichbar (z.B. könnte der Raspi stand-alone betrieben werden), so muss diese Zeit erneut eingestellt werden. Damit dies über die Zeitseite vom OrderSprinter (ab Version 1.0.35) möglich ist, muss das *date*-Kommando vom Webserver-Account ausgeführt werden werden. Analog zum Abschnitt 1.9 kann das S-Bit vom date-Kommando gesetzt werden:

sudo chmod u+s /bin/date

Wenn man sich nun in den OrderSprinter mit einem Benutzer mit Admin-Rechten einloggt, wird man auf eine Seite zum Abgleichen der Zeiten umgeleitet, wenn sich Datum/Zeit zwischen Server und Client um mehr als einen Tag unterscheiden (Abb. 1.13). Auf dieser Seite wird sowohl die Server- als auch die Client-Zeit angezeigt. Wenn man den Button Setze Client-Zeit anklickt, so wird versucht, die Zeit auf dem Server mit der Client-Zeit zu überschreiben. War das erfolgreich, sollte die Server-Zeit in der Anzeige oben korrigiert sein.



Abb. 1.13: Zeit kann mit einem Admin-Benutzer eingestellt werden.

1.11 WLAN

In der Ersteinrichtung des Raspis wurde bereits beschrieben, dass man über raspi-config den Zugang zum WLAN-Router einstellen kann. Die ersten Versionen des Raspis hatten keine eingebaute WLAN-Antenne und daher mussten auch die früheren Raspbian-Versionen noch manuell angepasst werden, damit mit einem zusätzlichen WLAN-Adapter eine Verbindung zum WLAN-Netz hergestellt werden konnte.

Dieses Kapitel WLAN ist also nur für diejenigen Anwender relevant, die noch eine ältere Raspbian-Version und einen alten Raspi ohne eigenen WLAN-Adapter verwenden. Alle anderen Anwender können weiterblättern!

In diesem Abschnitt werden zwei verschiedene Set-Ups beschrieben:

- Raspi hängt an einem Router wird über diesen angesprochen
- Raspi als eigenständigen WLAN-Access-Point, der bei Anschluss an einen Internetrouter auch Verbindungen ins Internet weiterleiten kann.

Die beiden Konzepte unterscheiden sich so grundlegend, dass zwei eigene Unterkapitel entstanden sind. Es werden jeweils sinnvolle Inhalte für Konfigurationsdateien vorgestellt, die natürlich stets nur als Muster zu betrachten sind und beliebig an die eigene Umgebung angepasst werden können und sollten.

1.11.1 Raspi am Router

In dieser Konfiguration soll der Raspi über WLAN mit einem Router kommunizieren, aber nicht selbst als Hotspot fungieren.

Die Datei /etc/network/interfaces sollte folgende Zeilen enthalten:

auto lo iface lo inet loopback iface eth0 inet static address 192.168.178.34

```
netmask 255.255.255.0
gateway 192.168.178.1
auto wlan0
allow-hotplug wlan0
iface wlan0 inet static
address 192.168.178.33
netmask 255.255.255.0
wpa-ap-scan 1
wpa-scan-ssid 1
wpa-ssid "SSID-des-Routers"
wpa-psk "WPA2-Password"
```

In dieser Datei ist eingetragen, dass dem Ethernet-Adapter die IP-Adresse 192.168.178.34 und dem WLAN-Adapter die Adresse 192.168.178.33 zugewiesen wird. Diese Adressen müssen gegebenenfalls an die eigene Umgebung angepasst werden. Ebenso muss natürlich die SSID und das Password des eigenen Routers eingetragen werden.

Ein *sudo service networking restart* zeigt, ob die WLAN-Konfiguration so prinzipiell funktioniert. Die WLAN-Funktionalität kann man auch so testen:

sudo ifdown wlan0 sudo ifup wlan0

Nach einer gewissen Zeit kann man von einem anderen Rechner im eigenen Heimnetzwerk testen, ob ein Ping an beide Adressen 192.168.178.33 und 192.168.178.33 erfolgreich ist. Wenn man nun das Netzwerkkabel abzieht, sollte der Raspi weiterhin über 192.168.178.33 anpingbar sein. Ist das nicht der Fall, kann eine Firewall-Einstellung des Routers die Ursache für die Nicht-Erreichbarkeit sein.

1.11.2 Raspi als Access Point

Anmerkung: Dieser Abschnitt gilt nicht für den Raspi Version 3. Das Vorgehen für den Raspi 3 wird im nächsten Kapitel beschrieben.

Soll der Raspi über das WLAN erreichbar sein und als Hotspot dienen (d.h. er ist selbst WLAN-Server), so erfordert dies eine etwas aufwändigere Konfiguration. Die Anleitung geht davon aus, dass ein Raspi-übliches Edimax-WLAN-Modul verwendet wird. Der Autor rät dringend zur Verwendung eines WLAN-Moduls mit großer Antenne (z.B. Modell 8192), weil mit den winzigen USB-Dongles (z.B. Modell 8188) die Reichweite sehr begrenzt ist.

Vor dem Einschalten des Raspi sollte der WLAN-Adapter an einen USB-Port angesteckt werden. Nach dem Einschalten des Raspies sollte der WLAN-Adapter über das Kommando *lsusb* in der Liste erscheinen (Abb. 1.14).

> pi@rasyberrypi * \$ lsusb Bus 001 Device 002: ID 0424:9514 Standard Microsystems Corp. Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp. Bus 001 Device 004:_ID 7392:7822 Edimax Technology Co., Ltd

Abb. 1.14: Der Edimax-WLAN-Adapter taucht in der Liste von Isusb auf

Soll auf einem Linux-Rechner ein WLAN-Access-Point betrieben werden, so wird *hostapd* benötigt, nicht *wpa_supplicant*. Außerdem soll der Raspi als Bridge zu einem bestehenden Heimnetz verwendet werden können (d.h. ist der Raspi über ein Netzwerkkabel an den Router angeschlossen, so können die Geräte per WLAN über den Raspi an die anderen Geräte im gleichen Netzwerk zugreifen als auch ins Internet gehen). Zum Installieren der Tools führt man folgendes Kommando aus:

sudo apt-get install hostapd bridge-utils iw

Wenn man nun

iw list

durchführt, wird wahrscheinlich der fehlende Treiber nl80211 moniert. Das ist im Moment noch ok.

Für die folgenden Schritte müssen wichtige Systemdateien verändert werden. Es empfiehlt sich, vor jeder Veränderung eine Sicherheitskopie anzulegen. Wenn durch eine Fehlkonfiguration der Raspi nicht mehr über das Netz erreichbar ist, kann die Veränderung immer noch über das Zurückspielen der Originalversionen direkt auf der in einem PC eingeschobenen SD-Karte rückgängig gemacht werden! Im nächsten Schritt muss die Datei /etc/default/hostapd editiert werden. Damit hostapd zukünftig nach dem Start des Raspis ausgeführt werden, muss der Daemon-Prozess entsprechend instruiert werden:

```
DAEMON_CONF="/etc/hostapd/hostapd.conf"
RUN_DAEMON=yes
```

Damit *hostapd* weiß, welches Netzwerk es anbieten soll, muss die Datei, auf die soeben der Daemon konfiguriert wurde (/*etc/hostapd/hostapd.conf*), erstellt oder angepasst werden:

```
interface=wlan0
#bridge=br0
driver=rt1871xdrv
country code=DE
ctrl_interface=wlan0
ctrl_interface_group=0
ssid=OrderSprinter
hw_mode=g
channel=1
wpa=2
wpa_passphrase=WPA2Password
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn pairwise=CCMP
beacon int=100
auth algs=3
macaddr acl=0
wmm enabled=1
eap_reauth_period=360000000
```

Der Inhalt kann nach eigenem Geschmack angepasst werden. Beispielsweise sollte ein eigenes WPA2-Passwort statt WPA2Password vergeben werden, die WLAN-SSID OrderSprinter kann ebenso verändert werden. Wenn der WLAN-Kanal 1 bereits sehr belegt ist, sollte ein anderer Kanal gewählt werden. Ganz wichtig ist zu beachten, keine unnötigen Leerzeichen (z.B. am Ende der Zeilen) einzubauen, da hostapd darauf sehr unfreundlich reagiert!

26

Der Eintrag *bridge* ist absichtlich einkommentiert, um darauf hinzuweisen, dass der Raspi nicht als *reine* Bridge sondern als Access Point, betrieben werden soll. Eine Bridge-Konfiguration würde bedeuten, dass der Raspi zwar als WLAN-Hotspot funktionieren würde, aber alle Verbindungen an einen Router weiterleiten würde und auch dem Router wichtige Aufgaben wie die Namenauflösung und IP-Adressvergabe überlasst. In diesem Abschnitt soll aber beschrieben werden, wie der Raspi auch alleine *lebensfähig* gemacht werden kann, d.h. er soll sich um alles kümmern können, was auch ein üblicher Router so macht (Namensauflösung, IP-Adressvergabe).

Das WLAN-Interface muss nun in der Date
i/etc/network/interfacesangepasst werden.

```
auto lo
#auto br0
iface lo inet loopback
iface default inet dhcp
#iface br0 inet dhcp
#bridge fd 1
#bridge_hello 3
#bridge_maxage 10
#bridge_stp off
#bridge_ports eth0 wlan0
#allow-hotplug eth0
iface eth0 inet static
address 192,168,178,36
netmask 255,255,255,0
broadcast 192,168,178,255
gateway 192.168.178.1
dns-nameservers 192.168.178.1
allow-hotplug wlan0
auto wlan0
iface wlan0 inet static
address 192.168.0.1
netmask 255.255.255.0
broadcast 192.168.178.255
# IP-Firewall initialisieren, d.h. die vorhandene Regeln und Ketten zuerst löschen
up /sbin/iptables -F
 up /sbin/iptables -X
  up /sbin/iptables -t nat -F
# Maskieren der LAN-Schnittstelle, Port-Forwarding & Nat aktivieren
up iptables -A FORWARD -o eth0 -i wlan0 -s 192.168.178.0/24 -m conntrack --ctstate NEW -j ACCEPT
 up iptables -A FORWARD -m conntrack --ctstate ESTABLISHED, RELATED -j ACCEPT
  up iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
   up sysctl -w net.ipv4.ip_forward=1
# hostapd und dnsmasq neu starten
   up /etc/init.d/hostapd restart
    up /etc/init d/dnsmasq restart
```

Den Inhalt der Datei /etc/network/interfaces entstammt zu einem großen Teil der Seite http://www.tacticalcode.de/2013/02/raspberry-pi-alsaccesspoint-oder-wlan-bridge.html, auf der Damon Dransfeld die Einrichtung des Raspis als Access Point sehr gut beschrieben hat. Auch in diesem Fall sind die bridge-Anweisungen beabsichtigt einkommentiert, aber nicht entfernt worden. Wenn der Raspi über einen Router eine Verbindung ins Internet erlauben soll (d.h. über ein Ethernet-Kabel an den Router angeschlossen ist), muss als Gateway die Adresse des Routers (in diesem Beispiel: 192.168.178.1) eingetragen sein. Zusätzlich hat das Ethernet-Interface nun die Adresse 192.168.178.36 und das WLAN-Interface die Adresse 192.168.0.1 zugewiesen bekommen.

Die hostapd, die dem letzten Raspbian-Release (Stand Juli 2015) beilag, ließ sich nicht überreden, mit dem Modul nl80211 zu sprechen. Unter http://dl.dropbox.com/u/1663660/hostapd/hostapd liegt jedoch eine Version, die funktioniert. Man kann sie so einbauen:

cd /usr/sbin sudo wget http://dl.dropbox.com/u/1663660/hostapd/hostapd sudo chown root:root hostapd sudo chmod 755 hostapd

Der Raspi soll später selbstständig IP-Adressen an Geräte vergeben, die sich bei ihm anmelden. Dazu müssen zwei wichtige Pakete installiert werden:

sudo apt-get install dnsmasq iptables

Das Paket *iptables* enthält die Firewall, die soeben in der /etc/network/interfaces konfiguriert wurde. Das andere Paket dns-masq enthält DNS- und DHCP-Services (DNS: Namensauflösung, DHCP: IP-Adressvergabe) und dessen Arbeitsweise muss nun in der Datei /etc/dnsmasq.conf eingestellt werden. Folgender Inhalt erklärt das WLAN-Interface wlan0 zum DHCP-Server, verbietet die Adressvergabe für den Ethernetanschkuss (Kabelverbindung) und gibt den Bereich vor, aus dem IP-Adressen an Clients vergeben werden können:

DHCP-Server aktiv für WLAN-Interface
interface=wlan0

DHCP-Server nicht aktiv für bestehendes Netzwerk no-dhcp-interface=eth0 1.11. WLAN

```
# IP-Adressbereich / Lease-Time
dhcp-range=interface:wlan0,192.168.0.100,192.168.0.200,infinite
```

Statt *infinite* lassen sich auch Angaben wie beispielsweise 12h machen. Diese beschreiben, wie lange eine vergebene IP-Adresse gültig sein soll.

Die Konfiguration lässt sich mit folgendem Kommando testen:

```
sudo hostapd -dd /etc/hostapd/hostapd.conf
```

Wichtig ist die Ausführung unter root-Rechten (sudo), damit der Zugriff auf die Treiber erlaubt ist. Im Idealfall sollte man nach wenigen Sekunden ein WLAN-Netz mit dem in der Datei /etc/hostapd/hostapd.conf angegebenen SSID-Namen finden und sich mit diesem verbinden können. Hängt der Raspi per Ethernet-Kabel an einem Router, sollte man zusätzlich damit nun auch im Internet surfen können. Sollte die Verbindung jedoch nicht zustandekommen, hilft ein Blick in die sehr detailreiche Ausgabe des soeben beschrieben Kommandos sudo hostapd -dd /etc/hostapd/hostapd.conf. Sowie der Prozess beendet wird, ist auch das WLAN-Netz wieder verschwunden.

Normalerweise sollte diese Konfiguration nach einem Neustart des Raspis den WLAN-Access-Point bereitstellen, da bereits in der Datei /etc/default/hostapd der Eintrag $RUN_DAEMON=yes$ gesetzt wurde. In der hostapd-Version, die hier beschrieben wurde, gibt es jedoch einen Fehler, der als Bug 729183 ("HostAPD removes IPv6 addresses from interface in manged mode") bekannt ist. Dieser führt dazu, dass hostapd alle nicht-IPv6-Adressen für jene Adapter entfernt, für die es zuständig ist. Das bedeutet im Klartext, dass die dem wlan0-Interface in /etc/network/interface zugewiesene statische Adresse 192.168.0.1 dem wlan0-Interface nach dem Start von hostapd wieder entrissen wird. Als Konsequenz ist zwar eine Verbindung zum Raspi als Access Point möglich, jedoch ist der Webserver auf dem Raspi nicht erreichbar, so dass damit der Zweck eines Stand-alone-Servers natürlich nicht erfüllt werden kann...

Es gibt einen schmutzigen Trick, diesen Bug zu umgehen: Nach dem Start von hostapd muss die Adresse dem wlan0 wieder zugewiesen werden. Mit der eben beschriebenen Konfiguration startet hostapd als Service während des Boot-Vorgangs. Diese Autostart-Funktionalität wird über Skripte eingebettet, die in /etc/init.d abgelegt sind. Dort findet sich eine Datei

hostapd, die beim Startvorgang des Kernels mit dem Argument start aufgerufen wird. Der Trick besteht nun darin, über das Kommando *ifconfig* wlan0 192.168.0.1 als letzte Aktion des Startvorgangs dem wlan0-Interface die statische Adresse 192.168.0.1 wieder zuzuweisen:

Wenn nun alles funktioniert, sollte nach dem Neustart des Raspis auch ohne Netzkabelverbindung zwischen Router und Raspi ein WLAN-Netz vom Raspi bereitgestellt werden, über dass man sich mit jedem Endgerät verbinden kann. Anschließend ist die OrderSprinter-Weboberfläche unter http://192.168.0.1 erreichbar.

Prinzipiell sind noch weitere Konfigurationen denkbar. Beispielsweise könnte man aus Sicherheitserwägungen die Menge der Clients einschränken, die sich mit dem Raspi verbinden können, indem man eine Liste von zulässigen MAC-Adressen übergibt (eine MAC-Adresse sollte jede Netzwerkkarte eindeutig identifizieren, doch dies gilt nicht für Hacker).

Auch wenn der Raspi prinzipiell als Stand-alone-Gerät betrieben werden kann, ist während eines Neustarts eine Verbindung über einen Router ins Internet sinnvoll, da der Raspi seine Uhrzeit nur über einen Zeitserver im Internet beziehen kann. Natürlich kann man ein Zeitmodul nachrüsten oder die Uhrzeit manuell auf der Kommandozeile setzen, das sprengt aber den Rahmen dieser Anleitung.

1.11.3 Raspi 3 als Access Point

Beim Raspi 3 ist die WLAN-Antenne bereits eingebaut und die Version Jessie ist dafür speziell vorbereitet worden. Daher kann hier anders vorgegangen werden. Zunächst müssen die Pakete dnsmasq und hostapd installiert werden:

sudo apt-get install dnsmasq hostapd

Im nächsten Schritt soll der Raspi über WLAN eine statische IP-Adresse bekommen. In den neuen Raspbian-Versionen wird die Interface-Konfiguration standardmäßig über dhcpd verwaltet. Daher müssen wir in einem Konfigurationsfile angeben, dass für WLAN keine dynamische Adresse zugewiesen werden soll: In die Datei /etc/dhcpcd.conf muss als erste Zeile denyinterfaces wlan0 eingetragen werden.

Damit wir dem WLAN-Interface eine statische Adresse zuordnen können, muss die Datei /etc/network/interfaces bearbeitet werden:

Die Zeilen

iface wlan0 inet manual wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf

sind zu ersetzen durch

```
iface wlan0 inet static
address 192.168.0.1
netmask 255.255.255.0
network 192.168.0.0
broadcast 192.168.0.255
```

Der DHCP-Daemon kann nun neu gestartet und die Konfiguration für WLAN angewendet werden:

sudo dhcpcd restart sudo ifdown wlan0; sudo ifup wlan0

Nun muss hostapd konfiguriert werden. Dazu muss eine Datei mit dem Namen /etc/hostapd/hostapd.conf mit folgendem Inhalt angelegt werden:

```
interface=wlan0
driver=nl80211
ssid=OrderSprinter
hw_mode=g
channel=6
```

```
ieee80211n=1
wmm_enabled=1
ht_capab=[HT40][SHORT-GI-20][DSSS_CCK-40]
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_key_mgmt=WPA-PSK
wpa_passphrase=VeryVerySecret
rsn_pairwise=CCMP
```

Als SSID wurde *OrderSprinter* und als Password *VeryVerySecret* eingetragen. Die Werte sind natürlich entsprechend für eine produktive Installation anzupassen.

Damit hostap
d nach dem Starten des Raspis automatisch die eben angelegte Konfiguration einliest, muss man in der Date
i/etc/default/hostapddie Zeile

#DAEMON_CONF=""

ersetzen durch

```
DAEMON_CONF="/etc/hostapd/hostapd.conf"
```

Dnsmasq muss nun konfiguriert werden. Sichern Sie die Originalkonfiguration und legen eine neue an:

```
mv /etc/dnsmasq.conf /etc/dnsmasq.conf.orig
vi /etc/dnsmasq.conf
```

Tragen Sie folgenden Inhalt ein:

interface=wlan0
listen-address=192.168.0.1
bind-interfaces

server=8.8.8.8
domain-needed
bogus-priv
dhcp-range=192.168.0.10,192.168.0.254,12h

Zum Abschluss muss das IP4-Forwarding aktiviert werden. In der Datei /etc/sysctl.conf muss das Routenzeichen (Kommentar) der folgenden Zeile entfernt werden:

#net.ipv4.ip_forward=1

Wenn Geräte, die über WLAN verbunden sind, auch die Internetanbindung des Raspis nutzen können sollen, so müssen folgende Kommandos ausgeführt werden:

iptables -t nat -A POSTROUTING -o ethO -j MASQUERADE iptables -A FORWARD -i ethO -o wlanO -m state --state RELATED,ESTABI iptables -A FORWARD -i wlanO -o ethO -j ACCEPT

Damit diese Kommandos nach jedem Systemstart aufgerufen werden, müssen diese Regeln gesichert werden:

sh -c "iptables-save > /etc/iptables.ipv4.nat"

In der Datei /etc/rclocal muss am Ende vor dem Exit-Kommando nun folgende Zeile eingefügt werden:

iptables-restore < /etc/iptables.ipv4.nat

1.12 Regelmäßiges Backup

Im Download-Zip befindet sich ein Ordner *scripts* mit einer Datei *backup-db-linux.sh* mit der auf einem Linux-System

• ein Backup der Datenbank

• das Kassenbuch (PDF-Zusammenfassung) vom Zeitraum des aktuellen Jahres

auf einem USB-Stick gespeichert werden kann. Der USB-Stick wird vor dem Backup eingehängt, und nachher wieder ausgehängt, so dass er anschließend vom Gerät abgezogen werden kann.

Das Skript kann auf einem Raspberry Pi oder anderen Linux-System eingesetzt werden.

Das Ein- und Aushängen erfordert eigentlich Root-Berechtigungen. Entweder lässt man das Skript als Root laufen, oder nutzt Möglichkeiten des setuid (wird hier nicht beschrieben).

Es werden immer die letzten maximal drei Backups gespeichert bzw. aufbewahrt.

Im Skript müssen am Anfang einige Variablen angepasst werden:

- der Gerätenamen des Sticks (man kann diesen Namen herausfinden, indem man den USB-Stick ansteckt und in /var/log/messages nachschaut)
- die Zugangsparameter der Datenbank (lassen sich aus der config.php auslesen, wenn nicht mehr bekannt).
- die Server-Baseadresse
- der Remoteaccess-Code (zuvor in der Administrationsansicht einstellen!)

Es wird empfohlen, die Ausführung des Skripts über einen cron-Job regelmäßig und automatisisiert zu starten. Um z.B. jeden Morgen um 4 Uhr eine Datensicherung zu starten, legt man den Cronjob wie folgt an:

In der Date
i/etc/crontabmuss ans Ende folgender Eintrag geschrieben und anschließend eine Zeile freigelassen werden:

0 4 * * * root /usr/local/bin/backup-db-linux.sh > /dev/null 2>&1

1.13 Anschluss eines Touch-Displays

Ein Raspi im Headless-Betrieb (d.h. ohne angeschlossenen Monitor und Tastatur) hat aber gewisse Nachteile. Mit einem Raspi-Touch-Monitor in der Größe von 3,5 Zoll, den man für unter 20 Euro zum Aufstecken bekommt, kann man aber einen gewissen Komfort erreichen.



Abb. 1.15: Raspi mit Touch-Display

Wer einen solches Setup aufbauen möchte, sollte die größere Raspbian-Version installieren, damit die grafische Oberfläche enthalten ist. Für die Installation des Touch-Displays halten die Hersteller die benötigten Gerätetreiber und die Dokumentation bereit. Darum wird dies hier nicht beschrieben.

Die standardmäßig eingebundene grafische Oberfläche erlaubt das Herunterfahren des Raspi über ein Menü. Bei einfachen Administrationsaufgaben kann man diese ohne externe Hardware erledigen. Hilfreich ist dabei die Nutzung einer virtuellen Tastatur.

1.14 Dateisystemcheck

Sollte das Dateisystems der SD-Karte im Raspi einmal durch einen Stromausfall oder hartes Ausschalten Schaden genommen haben, startet der Raspi in der Regel nur bis zur Frage, ob man einen manuellen Dateisystemcheck durchführen möchte. Entweder hat man nun einen Bildschirm und eine virtuelle Tastatur installiert oder eine reale Tastatur zur Hand, um diesen Schritt anzustoßen, oder man fügt in die /boot/cmdline.txt folgende Zeilen hinzu:

fsck.mode=force
fsck.repair=yes