1

Smart-Home-Bausteine

Die Schnittstelle zwischen dem Internet und dem Heimnetz ist das heimische Internet-Zugangsgerät – in der Regel der DSL-WLAN-Router. Gemeinsam mit dem Raspberry Pi verfügt der DSL-WLAN-Router nicht nur über performante Prozessorleistung zur Verarbeitung der Daten, die auch in Sachen Hausautomation im Bereich Messen, Steuern, Regeln anfallen, er ist in der Regel auch dauerhaft online und damit an sieben Tagen 24 Stunden im Einsatz. In Ihrem Heimnetz können Sie noch viel mehr machen als Daten im Internet bereitstellen oder simple Dateien hin- und herschieben und den NAS-Server mit Multimedia-Dateien befüllen: Sie können den Raspberry Pi in Ihrem Heimnetz als Mastermind betreiben, das sämtliche Geräte im Haushalt steuert und überwacht.

Heute ist das Thema Netzwerkeinrichtung zu Hause eigentlich keine große Sache mehr – knifflig wird es erst, wenn unterschiedliche Computer vernetzt und mit gewöhnlichen Haushaltsgeräten gekoppelt werden sollen. Dann muss man ein wenig Hand anlegen, damit es klappt. Anschließend können Sie mit dem Raspberry Pi über das Kabel- oder Funknetzwerk weitere Geräte, etwa Heizung, Lichtschalter, Waschmaschine, Klingelanlage und was noch alles in einem Haushalt an Gerätschaften benötigt wird, bequem steuern und kontrollieren.

1.1 LAN/WLAN-Router: Der Datenverteiler

Um die Verteilung der Daten in Ihrem Heimnetzwerk kümmert sich in der Regel ein Router, der den Datenverkehr gezielt steuert und die Netzbelastung in Grenzen hält.

Der Router wickelt sozusagen alle Aufträge ab, die von den Clients an ein anderes Netz geschickt werden. Ob es sich beim adressierten Netz um ein weiteres Unternehmensnetz handelt oder um das Internet, spielt keine Rolle.

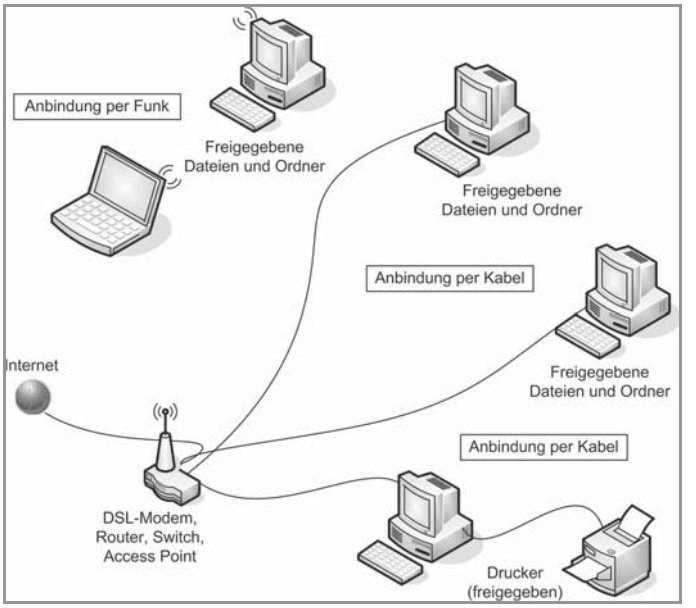

Bild 1.1: Beispiel eines Netzwerks, bestehend aus Kabel- und WLAN-Verbindungen mit Datei- und Druckerfreigaben

Wie auch immer in Ihrem Netzwerk Daten übertragen werden und welches Betriebssystem Sie auch einsetzen, an TCP/IP, der Internetprotokollfamilie, kommen Sie nicht vorbei. Jetzt brauchen Sie sich aber nicht mit so diffizilen Dingen wie Protokollschichten, Headern oder dergleichen herumzuschlagen, für Sie genügen die Basics der Adressierung. Außerdem müssen Sie wissen, dass TCP/IP festlegt, wie Daten im Internet und im Netzwerk übermittelt werden. Bei einer Netzwerkverbindung oder einer Internetverbindung wird keine direkte Verbindung zwischen zwei Punkten hergestellt, wie das beispielsweise beim Telefonieren der Fall ist.

1.1.1 TCP/IP-Protokoll als gemeinsamer Nenner

Die Daten werden vielmehr in kleine Pakete zerlegt und auf den Weg zum Ziel geschickt. Wo sie hinmüssen, steht in der Adresse. Am Ziel werden die Pakete wieder in der richtigen Reihenfolge zusammengesetzt. Auch das wird über TCP/IP gesteuert, denn Reihenfolge und Anzahl der Pakete werden ebenfalls übermittelt. Dazu kommen noch ein paar Prüfgeschichten und sonstige Informationen – das muss Sie aber nicht interessieren. Damit ein Rechner über TCP/IP angesprochen werden kann, muss seine Adresse, die sogenannte IP-Adresse, bekannt sein. Die Adressierung ist bei TCP/IP in ihrer Struktur festgelegt. Auf der Basis von Version IPv4 können bis zu 4.294.967.296 Rechner in ein Netzwerk integriert werden. IPv4 nutzt 32-Bit-Adressen, die Weiterentwicklung IPv6 hingegen setzt auf 128-Bit-Adressen.

Eine TCP/IP-Adresse ist immer gleich aufgebaut: Sie setzt sich zusammen aus einem Netzwerkteil und einem Hostteil (Adressenteil). In der Regel ist die 32-Bit-Adresse in einen 24-Bit-Netzwerkteil und einen 8-Bit-Hostteil aufgeteilt. Der Hostteil wird im LAN (im lokalen Netzwerk) zugeteilt, während der Netzwerkteil von der IANA (Internet Assigned Numbers Authority) vergeben wird, die über die offiziellen IP-Adressen wacht.

Für die Konfiguration des Hostteils sind in einem sogenannten Class-C-Netzwerk – das ist ein typisches privates Netz – 254 Geräteadressen für angeschlossene Clients verfügbar. Die Endadresse *255* ist für den Broadcast (zu Deutsch: Rundruf, also Übertragung an alle) reserviert, während die Adresse *0* für das Netzwerk selbst reserviert ist.

Für die Aufteilung des Netzwerk- und Hostteils ist die Netzmaske zuständig: Im Fall eines Class-C-Netzwerks gibt die Adresse *255.255.255.0* eine sogenannte Trennlinie zwischen beiden Teilen an. Die binäre *1* steht für den Netzwerkteil, und die *0* steht für den Adressteil.

So entspricht die Netzwerkmaske

```
255.255.255.0
```

binär:

```
11111111.11111111.11111111.0000000
```

Die ersten 24 Bit (die Einsen) sind der Netzwerkanteil.

Sie müssen sich aber gar nicht mit der Adressvergabe herumschlagen, denn der heimische Rechner ist immer mit den folgenden Daten ansprechbar. Einige Klassen von Netzwerkadressen sind für spezielle Zwecke reserviert. Man kann an ihnen ablesen, mit welchem Netzwerk man es zu tun hat. Beispielsweise ist eine IP-Adresse beginnend mit *192.X.X.X* oder *10.X.X.X* ein internes, in Ihrem Fall ein Heimnetzwerk.

Adressbereich	Netzwerk
192.168.0.0	Heimnetz, bis zu 254 Clients
172.16.0.0	Unternehmensnetz, bis zu 65.000 Clients
10.0.0.0	Unternehmensnetz, bis zu 16 Mio. Clients

Sobald aus einem heimischen Rechner ein Netz mehrerer Computer wird, beginnt die IP-Adresse mit *192.168.0*. Auf dieser Basis können in das Netz bis zu 254 Geräte eingebunden werden, indem die letzte Zahl von 0 bis 254 hochgezählt wird. Allerdings hat kaum jemand zu Hause so viele Geräte im Einsatz, es wird bei überschaubaren Adressbereichen bleiben.

1.1.2 Über die Vergabe der IP-Adressen

Gewöhnen Sie sich für die Vergabe der IP-Adressen entweder die automatische Zuweisung via DHCP oder eine statische Zuweisung mit festen Adressen an. Wenn Sie mit

festen Adressen arbeiten, sollten Sie gegebenenfalls nur ausgewählte, leicht merkbare IP-Adressen verwenden, also *192.168.0.1* für den Router, *192.168.10* für den zentralen Rechner und für weitere die Endnummern *20*, *30* etc. Wer generell Schwierigkeiten hat, sich die Nummern zu merken, kann die Computer beispielsweise nach Alter nummerieren – in der Regel weiß man genau, welchen PC man zuerst gekauft hat.

Der Vollständigkeit halber sei hier auch das sogenannte Gateway erwähnt. Innerhalb des Heimnetzwerks können sämtliche Geräte direkt miteinander kommunizieren und Daten austauschen. Soll hingegen eine Verbindung zu einem Gerät aufgebaut werden, das sich nicht innerhalb des adressierbaren Adressbereichs befindet, müssen diese Heimnetze miteinander verbunden werden. Diese Aufgabe übernimmt das Gateway bzw. der Router, der quasi sämtliche verfügbaren Netzwerke kennt und die Pakete bzw. Anforderungen entsprechend weiterleitet und empfängt. Im Internet sind daher einige Router in Betrieb, da es technisch nahezu unmöglich ist, dass ein einzelner Router alle verfügbaren Netze kennt und direkt adressieren kann.

In der Regel hat der Router auch einen DHCP-Server eingebaut, der für die Vergabe der IP-Adressen im Heimnetz zuständig ist. Sind Daten für eine IP-Adresse außerhalb des Heimnetzes bestimmt, werden sie automatisch an das konfigurierte Standard-Gateway, also den Router, weitergeleitet. Verbindet sich der heimische DSL-WLAN-Router mit dem Internet, versteckt er das private Netz hinter der öffentlichen IP-Adresse, die der DSL-WLAN-Router beim Verbindungsaufbau vom Internetprovider erhalten hat. Dieser Mechanismus der Adressumsetzung, NAT (**N**etwork **A**ddress **T**ranslation) genannt, sorgt dafür, dass die Datenpakete vom Heimnetz in das Internet (und wieder zurück) gelangen.

1.1.3 IP-Adressen im Internet übermitteln

Alle Server im Internet sind ebenfalls über eine IP-Adresse ansprechbar, aber das könnte sich keiner merken. Wer weiß schon, dass sich hinter *217.64.171.171 www.franzis.de* verbirgt? Deshalb gibt es im Internet zentrale Server, deren einzige Aufgabe darin besteht, für die von Ihnen eingegebene Internetadresse (URL) den richtigen Zahlencode bereitzustellen.

Nichts anderes passiert nämlich bei der Eingabe der URL: Der Rechner übermittelt seine Anfrage im Klartext an den sogenannten **D**omain **N**ame **S**erver (DNS). Ein DNS-Server führt eine Liste mit Domainnamen und den IP-Adressen, die jedem Namen zugeordnet sind.

Wenn ein Computer die IP-Adresse zu einem bestimmten Namen benötigt, sendet er eine Nachricht an den DNS-Server. Dieser sucht die IP-Adresse heraus und sendet sie an den PC zurück. Kann der DNS-Server die IP-Adresse lokal nicht ausfindig machen, fragt er einfach andere DNS-Server im Internet, bis die IP-Adresse gefunden ist. Damit die Daten, die Sie angefordert haben – und im Internet wird jede Seite aus übermittelten Daten aufgebaut –, auch wieder zu Ihnen bzw. zu Ihrem Rechner zurückgelangen, braucht der Server Ihre IP-Adresse. Nun wird nicht jedem Internetteilnehmer kurzerhand eine IP-Adresse verliehen – dafür gibt es einfach nicht genug Adressen. Statt-

dessen hat jeder Provider einen Pool mit IP-Adressen, die jeweils nach Bedarf vergeben werden.

Diese Technik ist quasi nichts anderes als die eines DHCP-Servers (**D**ynamic **H**ost **C**onfiguration **P**rotocol). Damit bekommen alle an ein Netzwerk angeschlossenen Computer, egal ob WLAN oder nicht, automatisch die TCP/IP-Konfiguration zugewiesen. Zusammen mit Ihrer Anfrage bei einer URL wird also Ihre eigene dynamische Adresse übermittelt, damit Sie auch eine Antwort bekommen.

1.1.4 Aus dem Internet ist nur der Router sichtbar

Wenn Sie Ihr Netzwerk mit einem Router für den Internetzugang ausstatten, übernimmt Ihr Router künftig einen Teil der Aufgaben rund um die Adressierung. Das macht Ihnen das Leben nicht nur etwas leichter, sondern vor allem viel sicherer, denn nach außen tritt lediglich der Router in Erscheinung, Ihren PC bekommt das Internet nicht so leicht zu sehen. Das beginnt schon damit, dass von außen nicht mehr die zugewiesene Adresse des Rechners zu sehen und zu verwenden ist, sondern die des Routers. Alle Anfragen stellt der Router, alle Antworten nimmt er entgegen und leitet sie netzwerktechnisch betrachtet als Switch innerhalb des heimischen Netzes an den passenden Rechner weiter.

Für den Router gibt es also intern den Nummernkreis *192.168.X.X* und nach außen alle anderen. Der einzelne Rechner ist nicht mehr direkt ansprechbar, sondern die Adresse ist immer die des Routers. Das ist ein erster Schritt in Richtung mehr Sicherheit im Internet, denn nun kann nicht mehr direkt auf möglicherweise offene Ports Ihres Rechners oder eines anderen im Netz zugegriffen werden. Noch mehr Sicherheit bietet eine im Router aktivierte Firewall, deren Ziel es ist, nur zulässige und ungefährliche Pakete durchzulassen und bestimmte Pakete kurzerhand abzulehnen. Sie nehmen ja an der Haustür auch nicht jede Nachnahme an.

1.1.5 Dynamische DNS-Lösung für Internet-Zugriffe konfigurieren

Möchten Sie Ihre Heimnetz-Steuerzentrale auf dem Raspberry Pi auch über das Internet erreichen, etwa weil Sie vom Büro aus die Heiztemperatur im heimischen Wohnzimmer regeln möchten, dann benötigen Sie für Ihren DSL/WLAN-Router zu Hause eine dynamische DNS-Lösung. Mithilfe einer dynamischen IP-Adresse machen Sie den WLAN/DSL-Router im Internet bekannt, und mit einem Raspberry Pi stellen Sie die Steuerung für die Gerätschaften im Heimnetz oder im Internet zu Verfügung. Jedes Mal, wenn Sie sich in das Internet einloggen, bekommt Ihr DSL-WLAN-Router automatisch vom Provider eine IP-Adresse zugeteilt. TCP und IP sind die wichtigsten Protokolle, die für die Kommunikation zwischen Rechnern möglich sind. Es gibt jedoch weitere Protokolle und Techniken wie beispielsweise SSH, mit denen Sie beim Lesen dieses Buchs in Berührung kommen. TCP/IP kommt in einem Netzwerk zum Einsatz, und jeder Computer, der in einem Netzwerk TCP/IP nutzen möchte, braucht eine IP-

Adresse. Diese IP-Adresse lautet bei jeder Einwahl anders – sie stammt aus einem IP-Adressenpool, den der Provider reserviert hat.

DNS: Namen statt Zahlen

Der Vorteil von DNS ist, dass Sie den Computer auch über seinen Namen ansprechen können. Es ist einfacher, statt einer IP-Adresse wie *http://192.168.123.1* die Adresse *http://IHRDOMAINNAME.DNSSERVICEANBIETER.NET* einzutippen, Namen lassen sich leichter merken als Zahlen bzw. IP-Adressen. Für das dynamische DNS gibt es verschiedene Anbieter, die ihre Dienste zum Teil kostenlos anbieten.

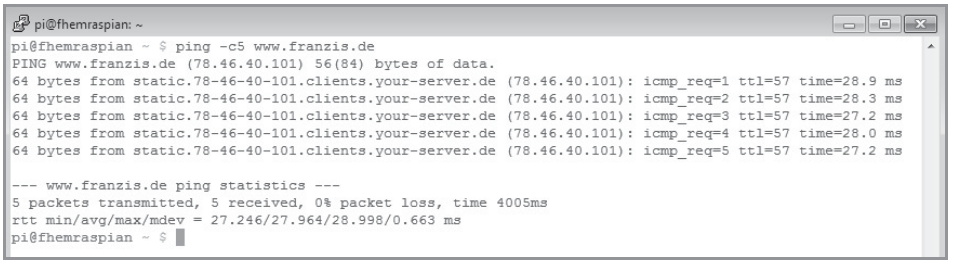

Bild 1.2: Mit dem Befehl *ping DNS-Name* finden Sie die IP-Adresse eines DNS-Namens heraus. In diesem Beispiel lautet die IP-Adresse für *www.franzis.de 78.46.40.101*.

Geben Sie beispielsweise *http://IHRDOMAINNAME.DNSSERVICEANBIETER.NET* in die Adressleiste des Webbrowsers ein, erkennt dieser am *http*-Kürzel, dass er das HTTP-Protokoll verwenden muss. Der doppelte Schrägstrich *//* bedeutet, dass es sich um eine absolute URL handelt. Mit der URL *IHRDOMAINNAME.DNSSERVICEANBIETER.NET* wird ein Kontakt zu dem DNS-Server Ihres ISP (**I**nternet **S**ervice **P**rovider) hergestellt. Damit wird dieser DNS-Name in eine IP-Adresse umgewandelt.

Für dynamisches DNS gibt es verschiedene Anbieter, die eine solche Funktionalität kostenlos zur Verfügung stellen. Die bekanntesten Anbieter sind in der folgenden Tabelle aufgeführt. Die Vorgehensweise bei der Anmeldung ist im Prinzip immer die gleiche, für welche Sie sich entscheiden, bleibt Ihnen überlassen.

Anbieter (kostenlos)	
no-ip.com	www.no-ip.com
DNSExit	www.dnsexit.com/
AVM (nur für Fritzboxen mit FRITZ!OS ab Version 5.20)	https://www.myfritz.net/login.xhtml
FreeDNS	freedns.afraid.org
Securepoint spdyn	www.spydn.de

Egal für welchen Anbieter Sie sich entscheiden, die Prozedur des Registrierens und Einrichtens sowie die Konfiguration des Clients bleiben Ihnen nicht erspart. Im nächsten

Schritt richten Sie den DSL-WLAN-Router so ein, dass Sie aus dem Internet Zugriff auf den Raspberry Pi bekommen – am besten über einen Port, der nur Ihnen bekannt ist.

DynDNS Domain bei spdyn.de einrichten

Made in Germany – wer in Sachen dynamisches DNS zum Nulltarif einen deutschen Anbieter auswählt, macht nicht viel verkehrt. Generell ist das Vorgehen bei sämtlichen Anbietern dasselbe: Erst nach dem Registriervorgang und der Bestätigung des E-Mail-Kontos haben Sie die Möglichkeit, einen kostenlosen dynamischen DNS-Anschluss zu reservieren.

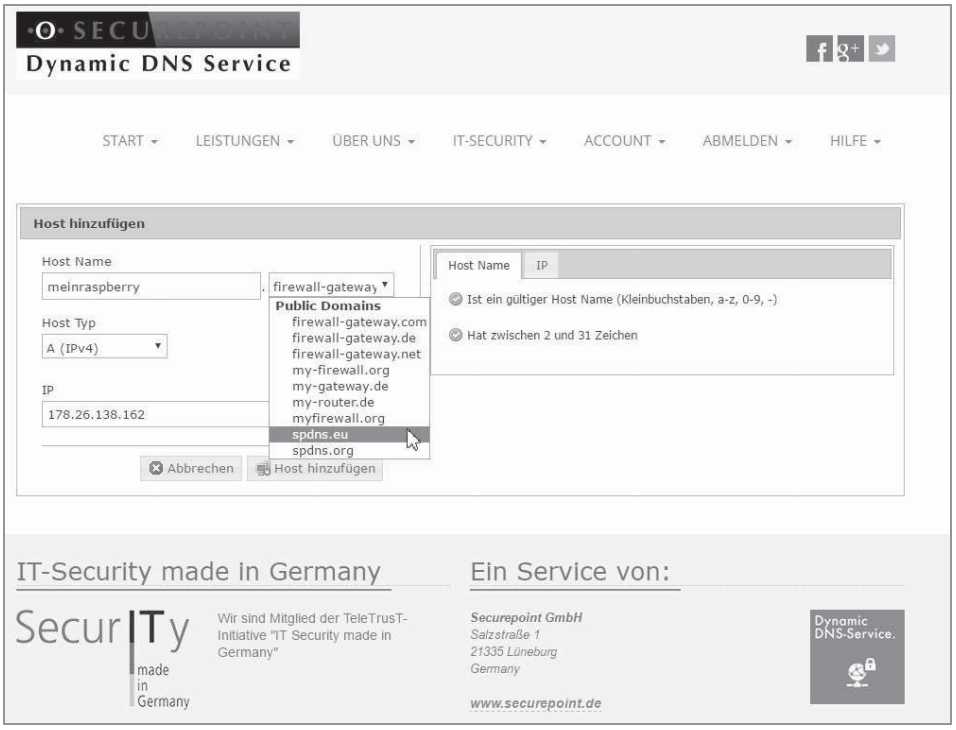

Bild 1.3: Zunächst tragen Sie den persönlichen Hostnamen ein und wählen im zweiten Schritt im Dropdown-Menü die gewünschte Domäne aus.

Nach der Registrierung bei dem Lüneburger Anbieter Securepoint (spdyn.de) fügen Sie dem angelegten Profil bei *Hosts* einen IPv4 Host hinzu und richten auf dem Raspberry Pi den spdns Dynamic DNS Update-Client ein. Alternativ konfigurieren Sie den heimischen Router so um, dass er den Securepoint DNS-Service kennt.

spdns Dynamic DNS mit der FRITZ!BOX

In diesem Abschnitt wird exemplarisch die weitverbreitete FRITZ!BOX verwendet, doch auch die Router-Konkurrenz im DSL/Kabel/Glasfasermarkt bietet die Einrichtung eines

Anbieters für die dynamische DNS-Adresse ein. Bei der FRITZ!BOX finden Sie die DynDNS-Einstellungen im Menü *Erweiterte Einstellungen -> Internet -> Freigaben -> Dynamic DNS*. Da nur die bekanntesten Anbieter von AVM in dem Einrichtungsassistenten gelistet werden, wählen Sie für die Verwendung des Securepoint Dynamic DNS Service als Dynamic DNS-Anbieter den Eintrag *Benutzerdefiniert* aus und tragen dort nachstehenden URL als Update-URL ein:

```
update.spdyn.de/nic/update?hostname=<domain>&myip=<ipaddr>
```

Beachten Sie, dass in diesem Beispiel die spitzen Klammern keine Platzhalter darstellen – geben Sie die Zeile eins zu eins in den FRITZ!BOX-Dialog ein. Unterstützt die FRITZ!BOX das DynDNS-Update auch über SSL, verwenden Sie dort stattdessen die nachstehende Update-URL:

```
https://update.spdyn.de/nic/update?hostname=<domain>&myip=<ipaddr>
```

Viele Internetanbieter bieten parallell zu einer IPv6-Hostadresse eine IPv4-Host an, und früher war es üblich, ausschließlich Adressen aus dem IPv4-Host-adressenbereich zu verwenden. Kabelanbieter wie Kabel Deutschland / Vodafone stellen aus Kompatibilitätsgründen den Internetanschluss sowohl mit einer IPv6-Adresse als auch mit einer IPv4-Hostadresse im Parallelbetrieb zur Verfügung. In Sachen dynamisches DNS bedeutet das, dass die FRITZ!BOX neben dem IPv4-Host auch den IPv6-Host aktualisieren kann. Soll dies über spdyn.de erfolgen, verwenden Sie für beide Adressen idealerweise denselben Hostnamen und tragen im Feld *Update-URL* die nachfolgenden Zeilen ein:

```
https://update.spdyn.de/nic/update?hostname=<domain>&myip=<ipaddr>
https://update.spdyn.de/nic/update?hostname=<domain>&myip=<ip6addr>
```

Das Feld *Update-URL* ist im Menü *Erweiterte Einstellungen -> Internet -> Freigaben -> Dynamic DNS* zu finden.

Bild 1.4: Tragen Sie in das Feld *Domainname* den persönlichen spdyn-Hostnamen ein, in das Feld *Benutzername* die Bezeichnung des spdyn-Accounts samt dazugehörigem Kennwort. Nach dem Klick auf die Schaltfläche *Übernehmen* kümmert sich die FRITZ!BOX um die automatische Aktualisierung der WAN-IP-Adresse für das dynamische DNS.

Soll der Raspberry Pi die Aktualisierung der WAN-IP-Adresse für die dynamische DNS-Adresse übernehmen - beispielsweise weil der eingesetzte Router keine Einstellungen für dynamisches DNS bietet, gehen Sie wie im nächsten Abschnitt beschrieben vor.

DNS-Update-Client für den Raspberry Pi

Das nachfolgende Verfahren lässt sich auch mit alternativen DynDNS-Anbietern anwenden - in diesem Beispiel wird der spdns-Dynamic-DNS-Update-Client für den oben angelegten spdyn.de-Account installiert und konfiguriert. Er steht kostenfrei auf spdns.de zur Verfügung und lässt sich bequem per `wget`-Kommando in das Home-Verzeichnis des Benutzers `pi` laden, vorausgesetzt der Raspberry Pi ist mit dem Internet verbunden.

```
cd ~
wget http://my5cent.spdns.de/wp-
content/uploads/2014/12/spdnsUpdater_bin.tar.gz
```

Nach dem Download entpacken Sie die Archivdatei per `tar`-Kommando:

```
tar -zxvf spdnsUpdater_bin.tar.gz
```

Die Archivdatei besteht aus zwei Dateien. Die Konfiguartionsdatei wird mit sudo-Berechtigungen in das */etc*-Verzeichnis verschoben, die Binärdatei für das eigentliche Update wird in ein eigenes Verzeichnis - hier *updater* - verschoben, die nötigen Berechtigungen werden angepasst und die Eigentümerschaft gesetzt. Zu guter Letzt bearbeiten Sie mit dem nano-Editor die Konfigurationsdatei *spdnsu.conf* im */etc*-Verzeichnis und tragen dort den dynamischen Host, den Benutzernamen samt dazugehörigem Kennwort, ein.

```
sudo mv spdnsu.conf /etc/
sudo mkdir -p updater
sudo mv spdnsu updater/
sudo chmod u+x updater/spdnsu
sudo chown -R pi:pi /home/pi/updater/
sudo nano /etc/spdnsu.conf
```

Nach dem Speichern der Konfigurationsdatei können Sie den Updater manuell ausführen und testen, ob er die WAN-IP-Adresse des genutzten Internetanschlusses zurückliefert:

```
./updater/spdnsu
```

Die WAN-IP-Adresse des genutzten Internetanschlusses wird in diesem Beispiel in die Datei */tmp/spdnsuIP.cnf* geschrieben, deren Inhalt sich per `cat`-Kommando auf der Kommandozeile anzeigen lässt.

```
cat /tmp/spdnsuIP.cnf
```

```
pi@raspi3cam:~ $ cd ~
pi@raspi3cam:~ $ wget http://my5cent.spdns.de/wp-content/uploads/2014/12/spdnsUpdater_bin.tar.gz
--2017-07-04 23:24:49--  http://my5cent.spdns.de/wp-content/uploads/2014/12/spdnsUpdater_bin.tar.gz
Auflösen des Hostnamen »my5cent.spdns.de (my5cent.spdns.de)«... 78.53.15.99
Verbindungsaufbau zu my5cent.spdns.de (my5cent.spdns.de)|78.53.15.99|:80... verbunden.
HTTP-Anforderung gesendet, warte auf Antwort... 200 OK
Länge: 7950 (7,8K) [application/octet-stream]
In »»spdnsUpdater_bin.tar.gz«« speichern.

spdnsUpdater_bin.tar.gz       100%[=============================================>]   7,76K  --.-KB/s   in 0,05s

2016-07-04 23:24:49 (149 KB/s) - »»spdnsUpdater_bin.tar.gz«« gespeichert [7950/7950]

pi@raspi3cam:~ $ tar -zxvf spdnsUpdater_bin.tar.gz
spdnsu.conf
spdnsu
pi@raspi3cam:~ $ sudo mv spdnsu.conf /etc/
pi@raspi3cam:~ $ sudo mkdir -p updater
pi@raspi3cam:~ $ sudo mv spdnsu updater/
pi@raspi3cam:~ $ sudo chmod u+x updater/spdnsu
pi@raspi3cam:~ $ sudo chown -R pi:pi /home/pi/updater/
pi@raspi3cam:~ $ sudo nano /etc/spdnsu.conf
pi@raspi3cam:~ $ ./updater/spdnsu
pi@raspi3cam:~ $ cat /tmp/spdnsuIP.cnf
currentIP=178.16.138.49
pi@raspi3cam:~ $
```

Bild 1.5: Wird der Updater nach der Konfiguration manuell gestartet, wird die aktuelle IP-Adresse der WAN-Verbindung in die Datei */tmp/spdnsuIP.cnf* geschrieben.

Wer diese Aktualisierung automatisch vom Raspberry Pi erledigen lassen möchte, der nutzt dort die *crontab*-Datei. Dafür starten Sie diese im Bearbeitungsmodus

```
sudo crontab -e
```

und tragen dort die nachstehende Zeile ein

```
*/10 * * * * /home/pi/updater/spdnsu
```

Speichern Sie die Datei und beenden Sie den Nano-Editor – nach einen Neustart des Raspberry Pi sollten Sie testen, ob alles wie gewünscht funktioniert, damit der Raspberry Pi die aktuelle WAN-Adresse automatisch an den DynDNS-Anbieter übermittelt.

1.1.6 Portfreigabe: Raspberry Pi in der Router-Software konfigurieren

Die TCP- und UDP-Ports (**U**ser **D**atagram **P**rotocol) sorgen für die Kommunikation auf Netzwerk- bzw. Anwendungsebene. Grundsätzlich gilt auch hier: Weniger ist mehr. Je weniger Ports geöffnet und Dienste verfügbar sind, desto weniger Angriffsfläche stellt der DSL-Router nach außen dar. So können Sie die Nutzung bestimmter Internetdienste wie das Surfen im WWW (HTTP), das **F**ile **T**ransfer **P**rotokoll (FTP) und viele andere für alle oder einige Benutzer in Ihrem Netzwerk blockieren.

Doch Vorsicht: Wird der Router zu sicher eingestellt, leidet die Funktionalität, weil bestimmte Programme nicht mehr richtig funktionieren. Wer beispielsweise einen Webserver (HTTP-Protokoll mit Port 80) hinter einem Router betreiben möchte, der muss den DSL-Router so einstellen, dass die Anfragen aus dem Internet auch bis zum Raspberry-Pi-Webserver kommen können. Erst dann kann dieser reagieren und die Anfragen beantworten. Welchen Port Sie öffnen, hängt von dem eingesetzten Server-

programm und vor allem von Ihren persönlichen Ansprüchen und Sicherheitsbedürfnissen ab.

Der Router kann auch so eingestellt werden, dass bestimmte Ports am Router offen sind, die Daten, die dort ankommen, aber nur an einen bestimmten Rechner bzw. eine bestimmte IP-Adresse weitergeleitet werden. Diese Technik läuft unter Portweiterleitung bzw. Port-Triggering. Die Porteinstellungen des WLAN-Routers nehmen Sie über die Weboberfläche vor. Im Falle einer FRITZ!Box ist das der Dialog *Internet/Portfreigabe* auf der Weboberfläche.

1.1.7 Portfreigaben einrichten und konfigurieren

Achten Sie darauf, dass bei der Konfiguration einer Portfreigabe die Zieladresse immer gleich bleibt. Hier ist es möglicherweise besser, für den Zielrechner im heimischen Netz eine feste IP-Adresse einzurichten. Verwenden Sie im Zweifelsfall statt einer DHCP-Adresse für den Computer eine statische IP-Adresse.

Mithilfe der FRITZ!Box-Portfreigabe lassen sich Dienste und verwendete Ports explizit bestimmten Rechnern im Heimnetz, in diesem Fall dem Raspberry Pi, zuordnen. Abhängig vom DSL-Router-Modell ist auch der umgekehrte Fall möglich, und es lassen sich ebenfalls bestimmte Dienste und Ports für bestimmte Rechner blockieren. Bei Netgear-Modellen ist dafür der Schalter *Dienste sperren/Block Services* zuständig, mit dem Sie den Internetzugang bestimmter Benutzer in Ihrem lokalen Netzwerk basierend auf deren IP-Adressen sperren können.

Bild 1.6: *Dienstetabelle* listet bei Routern aus dem Hause Netgear alle Dienste auf, die gegenwärtig gesperrt werden. Sie können Dienste dieser Tabelle hinzufügen oder sie auch daraus löschen.

Zusätzlich können Sie die Dienstsperrung bei manchen Routern auch von der Zeitplanung abhängig machen.

1.1.8 Benutzerkonten für unbefugte Zugriffe absichern

Spätestens jetzt, wenn der Raspberry Pi über das Internet erreichbar ist, ist es auch Zeit, ihn bzw. die entsprechenden Userkonten abzusichern, um möglichen Einbrechern wenig Zerstörungsspielraum zu geben. Grundsätzlich sollten Sie das Standard-Benutzerkonto `pi` bereits angepasst und das Standard-Kennwort `raspberry` auf ein sicheres Kennwort Ihrer Wahl geändert haben. Dies erledigen Sie mit dem Kommando:

```
sudo passwd pi
```

Sie geben das neue Kennwort ein und bestätigen dies im zweiten Schritt. Mit der Benutzerkennung `pi` können Sie sich auch die administrativen root-Rechte mittels `sudo`-Kommando holen.

Wer für das root-Konto auf dem Raspberry Pi ebenfalls ein persönliches Konto setzen möchte, der erledigt dies mit den Befehlen:

```
sudo -i
passwd
```

Hier tragen Sie zunächst das neue Kennwort und anschließend die Kennwortbestätigung ein, um das root-Konto mit einem persönlichen Kennwort abzusichern.

1.2 Raspberry Pi: Standards und Anschlüsse

Grundsätzlich benötigen Sie einen passenden Adapter, der das eingesetzte Funkprotokoll wie beispielsweise LON, BACnet, KNX, EnOcean, FS20 oder HomeMatic unterstützt. Bei den Platzhirschen wie FS20 oder HomeMatic stehen beispielsweise mit dem FHZ1000-Modul (FS20) oder dem LAN-Adapter (HomeMatic) quasi Herstellerschnittstellen auch für den Raspberry Pi über USB zur Verfügung. Diese sind etwas teurer als die Alternativen, die es beispielsweise von Drittanbietern gibt, wie *busware.de* mit dem CUL-Stick (hier der CC1101) oder dem COC-Modul, die beide in der Lage sind, Steuersignale von Protokollen im 868-MHz-Frequenzbereich zu empfangen und zu senden.

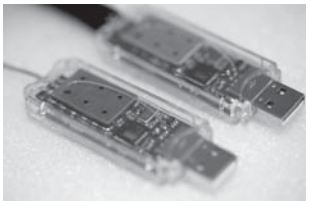

Bild 1.7: Doppelt gesteckt hält besser: Wer FS20 und HomeMatic gleichzeitig in einem Funknetz betreiben möchte, benötigt zwei unterschiedlich konfigurierte CUL-Module.

Damit ist die Alternative grundsätzlich fähig, sowohl mit FS20- als auch mit HomeMatic-Geräten im Funknetz zu kommunizieren – beide Standards sind jedoch miteinander inkompatibel. Möchten Sie beide in einem Funknetz betreiben, dann

benötigen Sie zwei entsprechende 868-MHz-Transceiver-Dongles – einen für FS20, den anderen für HomeMatic.

Gerät	Abkürzung für ...	CPU-Typ	RAM	Speicher
CUL	CC1101 USB Light	V1: at90usb162 V2: at90usb162 V3: atmega32U4	V1: 0.5 KB V2: 0.5 KB V3: 2.5 KB	V1: 16 KB V2: 16 KB V3: 32 KB
CUN	CC1101 USB Network	at90usb64	2.0 KB	64 KB
CUNO	CC1101 USB Network & OneWire	atmega644p	2.0 KB	64 KB
CUR	CC1101 USB Remote	CUR V1: at90usb128 CUR V2: at90usb64	CUR V1: 4.0 KB CUR V2: 2.0 KB	V1: 128 KB V2: 64 KB
COC	CC1101 OneWire Card	atmega644	2.0 KB	64 KB

So ist die eigentliche Funkübertragung das eine, das unterstützte Protokoll das andere. Sprechen Sender und Empfänger die gleiche Sprache, nutzen sie also das gleiche Protokoll, dann ist der Austausch von Informationen und Daten möglich. Die Gegenstellen mit Funkanschluss – also die zu steuernden Komponenten wie Sensoren, Aktoren und Empfänger – kommen mit einem eigenen Protokoll zur Datenübertragung. In diesem Buch stellen wir die beiden wichtigsten vor, jedoch ist die grundsätzliche Herangehensweise auch bei anderen Protokollen und Geräten in etwa dieselbe.

1.2.1 Durchblick im FS20- und HomeMatic-Protokoll

Sowohl das FS20- als auch das HomeMatic-Protokoll funken im 868-Mhz-Bereich, kommen vom gleichen Hersteller (EQ3, *www.eq-3.de*) und werden durch zahlreiche Vertriebspartner, wie ELV und Fachhandelsketten wie Conrad Electronic, vermarktet. EQ3 ist ein Tochterunternehmen von ELV und erweitert stetig das Produktprogramm im Einsteigersegment FS20, sodass es hier auch die meisten unterstützten Devices gibt. Aufgrund der vergleichsweise günstigen Komponenten ist es bei vielen Anwendern beliebt. HomeMatic-Komponenten sind bei gleicher grundsätzlicher Funktion in Sachen Anschaffungskosten höher angesiedelt.

Die Datenpakete werden mit den Funkprotokollen FHT und HMS übertragen. Während FHT-Geräte bei der Heizungssteuerung zum Einsatz kommen, sind HMS-Geräte eher für Sicherheits- und Überwachungsaufgaben geeignet.

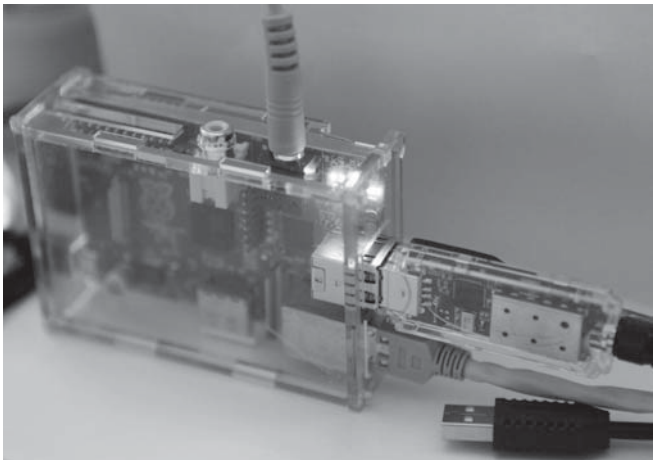

Bild 1.8: Raspberry Pi mit USB-CUL-Modul: Bei der Konfiguration müssen Sie sich für ein zu verwendendes Protokoll, FS20 oder HomeMatic, entscheiden.

Der große Nachteil der FS20-Technik ist die Kommunikation ohne Bestätigung. Wird beispielsweise ein Signal von der Steuereinheit zum Schalter geschickt, dann erhält diese keine direkte Rückmeldung, ob der Schaltbefehl erfolgreich ausgeführt wurde oder nicht. Dies ist bei HomeMatic-Geräten anders: Hier sorgt eine verschlüsselte Kommunikation dafür, dass die Sendeeinheit eine Bestätigung des Schaltvorgangs erhält. Ob Sie diese Sicherheit wirklich benötigen, steht auf einem anderen Blatt, und auch die etwas bessere Verarbeitung der HomeMatic-Komponenten lässt sich EQ-3 bzw. der Versender ELV gern bezahlen. Die Einrichtung der Komponenten beim Raspberry Pi ist jedoch nahezu identisch.

1.2.2 Angepasstes Funkmodul für den GPIO-Einsatz

Für den Raspberry Pi gibt es eigens ein passendes Funkmodul, das auf die GPIO-Pins gesteckt werden kann. Doch leider passt es auf Anhieb nur auf die »alten« Raspberry-Pi-1-Boards mit der Revision 1 (256 MB), der Raspberry-Pi-1-Nachfolger mit 512 MB sowie die Modelle Raspberry Pi 2, 3 und Zero mit 40 Pins passen zunächst nicht, da eine Steckverbindung auf dem Raspberry Pi um Zehntelmillimeter im Weg steht. Behelfen Sie sich mit einem Hebelwerkzeug oder einer Zange, zum Beispiel mit der Zange eines Leatherman-Tools, um den Plastikschutz des DSI-Anschlusses zu entfernen.

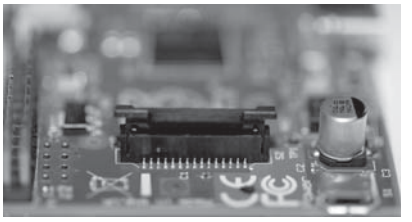

Bild 1.9: die GPIO-Reihe, ganz rechts unten der USB-Strom-Anschluss des Raspberry Pi: Der schwarze Schutzdeckel des DSI-Anschlusses (für TFT-Touchscreens) wurde bereits gelockert und leicht angehoben. Ist er komplett abgezogen, lässt sich das COC-Erweiterungsboard problemlos auf die neuere Revision mit 512 MB RAM setzen.

Ist die hardwareseitige Voraussetzung erfüllt und das Erweiterungsboard mit einem spürbaren Klick auf der GPIO-Pfostenleiste eingerastet, dann können Sie den Raspberry Pi wieder in Betrieb nehmen, also Stromversorgung und Netzwerkkabel anschließen und die Antenne am Antennenausgang des COC-Moduls anbringen. Hier sind diejenigen mit einem Bastelgehäuse leicht im Vorteil, die die Seitenwand für die Antenne einfach nicht verwenden und somit den Raspberry Pi trotzdem in ein Gehäuse packen können.

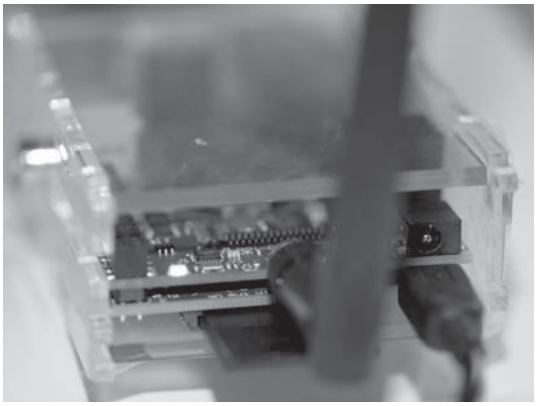

Bild 1.10: Kein Schönheitspreis, aber zweckmäßig: In diesem Fall wurde einfach das Seitenteil des Gehäuses nicht genutzt, um den Raspberry Pi samt COC-Erweiterung und verbauter Antenne in ein Gehäuse zu zwängen.

Alternativ nutzen Sie eine Bohrmaschine und bohren mit einem Holzbohrer an der entsprechenden Stelle vorsichtig eine passende Öffnung für die Antenne des COC-Moduls, um das Gehäuse weiter in Betrieb nehmen zu können. Damit ist die Raspberry-Pi-Hardware nun einsatzbereit, um die Konfiguration der Software und der Treiber vorzunehmen.

Grundsätzlich entfernen Sie oder besser kommentieren Sie in der Datei */etc/inittab* sämtliche Zeilen aus, die auf einen Eintrag mit der Bezeichnung *ttyAMA0* verweisen. Auch in der Kernel-Bootdatei */boot/cmdline.txt* entfernen Sie die Einträge *console=ttyAMA0,115200* und *kgdboc=ttyAMA0,115200*. Sicherheitsbewusste sichern zuvor die Datei mit dem Kommando

```
sudo cp /boot/cmdline.txt /boot/cmdline.txt.original
```

und starten nach der Änderung den Raspberry Pi neu. Wer anstelle eines COC die USB-Stick-Variante CUL im Einsatz hat, der kann diese nicht nur an der FRITZ!Box oder an einem Linux-Computer, sondern auch am USB-Anschluss des Raspberry Pi betreiben.

1.2.3 USB-Adapter als Alternative für den Raspberry Pi

Setzen Sie statt eines COC-Moduls einen USB-Adapter (CUL, CC1101 USB Light) für den Raspberry Pi ein, dann ist die Hardware-Installation schnell erledigt: Hier brauchen Sie nur die Antenne an den USB-Stick zu schrauben und in den USB-Anschluss

des Raspberry Pi zu stecken. Falls der Standort des Raspberry Pi nicht so ideal ist und die Antenne an einem sinnvolleren Ort platziert werden soll, ist gegebenenfalls eine USB-Verlängerung sinnvoll.

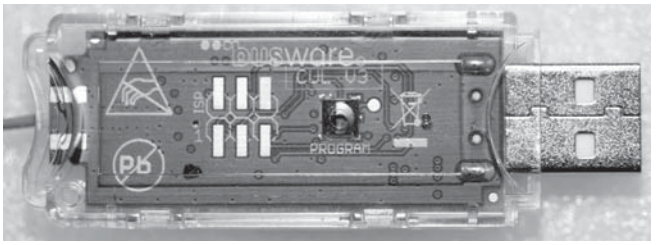

Bild 1.11: Zum Verwechseln ähnlich: Ein CUL-Stick von *busware.de* sieht ähnlich wie ein USB-Speicherkarten-adapter aus.

Ist das neue Gerät einmal am USB-Anschluss des Raspberry Pi eingesteckt, prüfen Sie per `lsusb` und `dmesg`, ob es ordnungsgemäß erkannt wurde oder nicht.

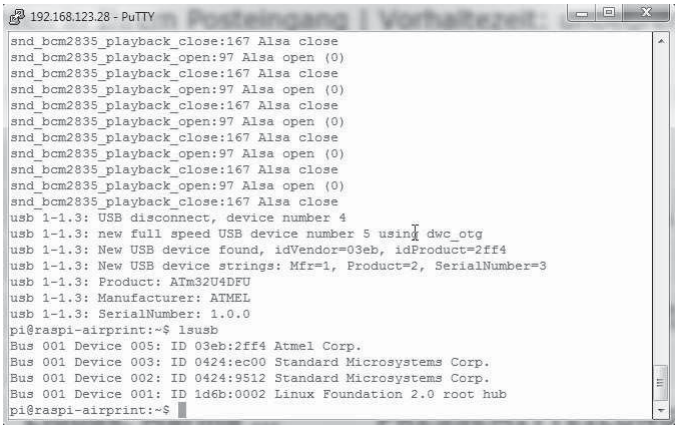

Bild 1.12: ATMEL mit Geräte-ID `03eb:2ff4`. Der USB-CUL wird auf Anhieb per `lsusb` auf dem USB-Bus erkannt.

Eine ausführliche, geordnete Übersicht erhalten Sie mit dem Kommando:

```
sudo lsusb -v | grep -E '\<(Bus|iProduct|bDeviceClass|bDeviceProtocol)'
2>/dev/null
```

Für Funksteckdosen, Schalter, Aktoren (Wandler) und Sensoren existieren zig unterschiedliche Standards auf dem Markt. Das COC bzw. CUL deckt die wichtigsten mit 433 MHz und 868 MHz ab. Hier müssen Sie sich bei der späteren FHEM-Konfiguration für jedes Device für einen Funkstandard entscheiden. Haben Sie mehrere unterschiedliche Technologien zu Hause im Einsatz, dann benötigen Sie für jeden Standard den passenden Controller.

```
[39446.520161] usb 1-1.3: new full-speed USB device number 5 using dwc_otg
[39446.623418] usb 1-1.3: New USB device found, idVendor=03eb, idProduct=204b
[39446.623448] usb 1-1.3: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[39446.623465] usb 1-1.3: Product: CUL868
[39446.623476] usb 1-1.3: Manufacturer: busware.de
[39446.708665] cdc_acm 1-1.3:1.0: ttyACM0: USB ACM device
[39446.713003] usbcore: registered new interface driver cdc_acm
[39446.713033] cdc_acm: USB Abstract Control Model driver for USB modems and ISDN adapters
pi@fhemraspian /usr/share/fhem/FHEM $ ls /dev/ttyA
ttyACM0  ttyAMA0
pi@fhemraspian /usr/share/fhem/FHEM $ ls /dev/ttyA
ttyACM0  ttyAMA0
pi@fhemraspian /usr/share/fhem/FHEM $ ls /dev/ttyAMA0
/dev/ttyAMA0
pi@fhemraspian /usr/share/fhem/FHEM $ ^C
pi@fhemraspian /usr/share/fhem/FHEM $
```

Bild 1.13: Der `dmesg`-Befehl sorgt für Klärung: Das ATMEL-Device wurde ordnungsgemäß vom Raspberry Pi erkannt und beim USB-Strang eingehängt. Der Eintrag `ttyAMA0` bei `dmesg` gibt den ersten Hinweis, dass der COC im Verzeichnisbaum unter `/dev/ttyAMA0` eingehängt wurde und sich später in `fhem.cfg` für den COC nutzen lässt.

In diesem Fall haben wir uns auf das FS20-System sowie den verbreiteten HomeMatic-Standard beschränkt: Das am GPIO-Port angeschlossene COC kümmert sich um die FS20-Technologie in dem Funknetz, das per USB angeschlossene CUL versorgt die HomeMatic-Komponenten in einem zweiten Funknetz. Wie Sie diese Funkschnittstellen mit FHEM in Betrieb nehmen, lesen Sie im Abschnitt »Anpassen der FHEM« auf Seite 45.

```
2013.01.02 01:04:08 3: Opening COC device /dev/ttyAMA0
2013.01.02 01:04:08 3: Setting COC baudrate to 38400
2013.01.02 01:04:08 3: COC device opened
2013.01.02 01:04:08 3: COC: Possible commands: mCFiAZOGMRTVWXefltux
2013.01.02 01:04:08 3: Opening CUL_0 device /dev/ttyACM0
2013.01.02 01:04:08 3: Setting CUL_0 baudrate to 9600
2013.01.02 01:04:08 3: CUL_0 device opened
2013.01.02 01:04:08 3: CUL_0: Possible commands: BCFiAZEGMRTVWXefmltux
2013.01.02 01:04:08 2: Switched CUL_0 rfmode to HomeMatic
2013.01.02 01:04:08 3: telnetPort: port 7072 opened
2013.01.02 01:04:09 1: Including /var/log/fhem/fhem.save
2013.01.02 01:04:09 4: /fhem?save=Save+fhem.cfg&saveName=fhem.cfg&cmd=style+save+fhem.cf
```

Bild 1.14: Logfile von FHEM klärt auf: Beide Adapter in einer FHEM-Konfiguration im Betrieb.

Um keinen überflüssigen Funksmog im Wohnbereich zu erzeugen, ist es sinnvoll, die Anzahl der Funkstandards so gering wie möglich zu halten – nicht zuletzt wegen des erhöhten Integrationsaufwands. Mit FHEM lassen sich verschiedene Technologien zusammenführen und auf einer einheitlichen Basis gemeinsam betreiben. Je nach verwendeter Funktechnologie ist die FHEM-Konfiguration entsprechend anzupassen. Oft ist es notwendig, ein zweites CUL oder zusätzlich die Platine COC für den Raspberry Pi in Betrieb zu nehmen.

1.3 Raspberry Pi: Camera Module v1 und v2

Obwohl zeitgleich mit der damaligen Veröffentlichung des Raspberry Pi mit vorgestellt, ist die Raspberry-Pi-Kamera erst seit Mai 2013 bestellbar, verbunden mit langen

Wartezeiten. Liegt die HD-fähige Raspberry-Pi-Kamera endlich im Briefkasten und ist ausgepackt, kann über die Kompaktheit und das mit wenigen Gramm geringe Gewicht des Kameramoduls nur gestaunt werden: In der Größe ist die Platine in etwa mit einer SD-Karte vergleichbar, die Bauhöhe des Linsenobjektivs samt Platine entspricht mit 9 mm drei aufeinanderliegenden Euro-Münzen. Mittlerweile ist das Kameramodul in Version 2 zum selben Preis wie sein Vorgänger erhältlich, auch in einer speziellen NOIR-Version, die ohne Infrarot-Filter kommt.

1.3.1 Kameramodul mit dem Raspberry Pi koppeln

Für den Anschluss an den Raspberry Pi ist an der Platine des Kameramoduls ein rund 15 cm langes Flachbandkabel angebracht, das für den CSI-Anschluss auf dem Raspberry Pi vorgesehen ist.

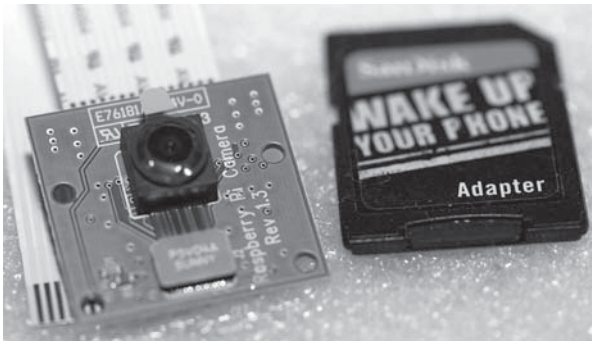

Bild 1.15: Das im Juni 2013 ausgelieferte Raspberry-Pi-Kameramodul trägt die Revision V1.3 und ist ungefähr so groß wie eine SD-Karte.

Die technischen Werte des Kamerasensors der Raspberry-Pi-Kamera sind in etwa mit denen eines Smartphones vergleichbar, mit dem 5-Megapixel-Sensor sind zudem Videos im HD-Format 1080p oder 720p sowie im betagten VGA-Format 640 × 480 möglich.

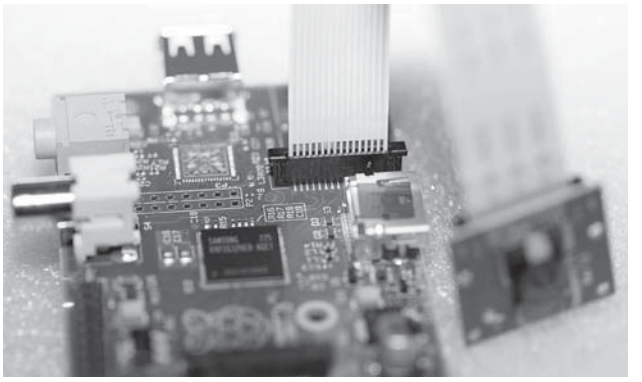

Bild 1.16: Beim Einbau in den CSI-Pfosten zeigen die Anschlüsse des Flachbandkabels in Richtung HDMI-Anschluss auf der Raspberry-Pi-Platine.

Nach dem Einstecken des Flachbandkabels installieren Sie die notwendige Software und nehmen die Kamera in Betrieb.

1.3.2 Betriebssystem und Firmware auffrischen

Ist die Kamera mit dem Raspberry Pi verbunden, müssen das Betriebssystem und die Firmware des Raspberry Pi auf den aktuellen Stand gebracht werden, sofern das noch nicht geschehen ist. Für das weitverbreitete Raspbian nutzen Sie folgende Kommandos – nicht nur, um das Betriebssystem aufzufrischen, sondern auch, um die Kamera in Betrieb zu nehmen.

```
sudo -s
apt-get update
apt-get upgrade -y
apt-get install git-core -y
wget https://raw.github.com/Hexxeh/rpi-update/master/rpi-update -O
/usr/bin/rpi-update
chmod +x /usr/bin/rpi-update
rpi-update
```

Das Auffrischen des Betriebssystems kann abhängig von der Anzahl der bereits installierten Pakete sowie der zur Verfügung stehenden Internetbandbreite eine Weile dauern. Zu guter Letzt wird die Firmware auf den aktuellen Stand gebracht.

Sind die installierten Pakete sowie die Firmware für den Raspberry Pi aktualisiert und die notwendigen Treiber für die Kamera installiert, prüfen Sie das nach dem Neustart per Kommando

```
reboot
```

und anschließend den Versionsstand des Raspberry Pi:

```
uname -a
```

Im nächsten Schritt richten Sie die Kamera mit dem bewährten Konfigurationswerkzeug `raspi-config` ein.

1.3.3 Camera Module in Betrieb nehmen

Erst mit dem Einspielen des Raspbian-Updates steht auch im Konfigurationswerkzeug `raspi-config` ein neuer Menüpunkt `Enable Camera` zum Einschalten einer angeschlossenen Kamera zur Verfügung.

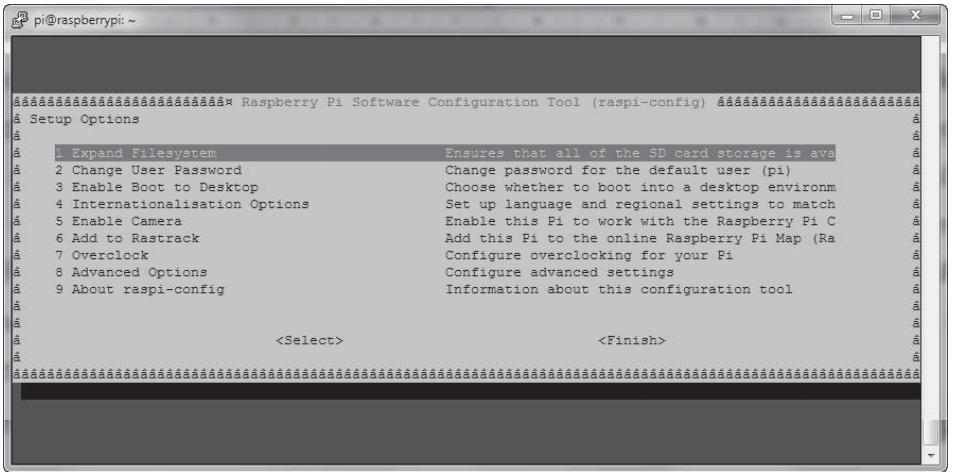

Bild 1.17: Nach dem Neustart des Raspberry Pi starten Sie per Kommandozeile das Konfigurationswerkzeug `raspi-config`.

Navigieren Sie mit den Pfeiltasten zum Punkt `Enable Camera` und drücken Sie die Enter-Taste. Anschließend erscheint ein Dialog, in dem Sie die Auswahl per `Enable` nochmals bestätigen. Analog gehen Sie vor, wenn Sie die Kamera später wieder abschalten wollen – in diesem Fall wählen Sie dann aber den Eintrag `Disable` aus.

Im nächsten Schritt legen Sie im Menü bei `Advanced Options` und dem Untermenü `Memory Split` fest, wie viel Grafikspeicher für die GPU zur Verfügung stehen soll. Hier lautet die Empfehlung, je nach Raspberry Pi die Hälfte des vorhandenen Arbeitsspeichers für die GPU zu nutzen. In diesem Beispiel – im Einsatz ist ein mit 256 MB ausgestatteter Raspberry Pi, Modell A – bekommt die GPU demnach mit dem Wert `128` entsprechend viel RAM-Speicher zugeordnet.

Das Einrichten der Kamera ist damit abgeschlossen. `raspi-config` möchte nun einen Neustart initiieren, damit beispielsweise die Speicherzuordnung für die GPU aktiv wird. Bestätigen Sie den Neustart per OK-Auswahlmenü. Anschließend kann die Kamera umgehend genutzt werden.

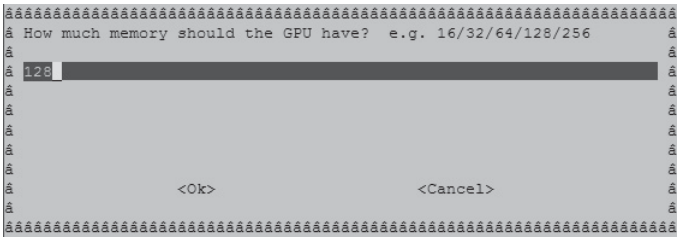

Bild 1.18: Nach der Zuordnung der Speichermenge von 128 MB für die GPU navigieren Sie per Tab-Taste zur `Ok`-Schaltfläche und schließen den Vorgang ab.

1.3.4 Fotografieren mit Kommandozeilenbefehl

Nach dem Neustart des Raspberry Pi stehen, neben der Kamera, auch über das Betriebssystem Raspbian neue Tools im der Kommandozeile zur Verfügung, die zusätzliche Tools wie `fswebcam` überflüssig machen sollen. Sie können `raspistill` zur Aufnahme einzelner Fotoaufnahmen nutzen, während das Werkzeug `raspivid` für die Aufnahme von Videos vorgesehen ist.

In Sachen Dateiausgabe unterstützt `raspistill` die gängigsten Formate wie JPG, BMP, GIF und PNG, das Tool `raspivid` nimmt in H.264-Codierung auf. Zusätzlich können mit `raspistill` im der Kommandozeile Parameter wie Belichtungsmodi, Farbeffekte und Kontrasteinstellungen verändert werden. Das Tool bietet umfangreiche Möglichkeiten, beim Fotografieren und Verarbeiten der Aufnahmen diverse Parameter zu setzen. So lässt sich beispielsweise mit der Option `-t` die Verzögerung in Millisekunden einstellen, falls am Raspberry Pi ein Bildschirm angeschlossen ist, um eine Vorschau des Kamerabilds anzuzeigen:

```
raspistill -t 500 -o aufnahme.jpg
```

Wenn die Kamera beispielsweise an einem Türspion eingesetzt wird und seitenverkehrte Aufnahmen erzeugt, können Sie die Ansicht des Bilds mit der Option `-hf` anpassen:

```
raspistill -hf -o aufnahme.jpg
```

Über die zahlreichen weiteren Optionen und Parameter des `raspistill`-Werkzeugs informiert die Hilfeseite, die Sie mit der einfachen Eingabe von `raspistill| less` auf der Konsole aufrufen. Mit den Pfeiltasten navigieren Sie in den Hilfeseiten, mit der [Q]-Taste (*Quit*) verlassen Sie die Hilfe. Viel Spaß beim Fotografieren mit dem Raspberry Pi.

1.3.5 LED abschalten und heimlich fotografieren

Wie die meisten Kameras bietet auch das Raspberry-Pi-Kameramodul eine optische Benachrichtigung in Form einer LED, die standardmäßig beim Anfertigen einer Aufnahme leuchtet. Es kann jedoch Einsatzzwecke geben, in denen diese rote LED besser abgeschaltet werden sollte, etwa beim heimlichen Fotografieren durch den Türspion bei einem Klingelsignal oder in einem Vogelhaus, in dem das Motiv natürlich nicht bemerken soll, dass es fotografiert wird. Um nun die angesprochene LED auf dem Raspberry-Pi-Kameraboard abzuschalten, reicht ein zusätzlicher Eintrag in der `config.txt`-Datei des Raspberry Pi aus:

```
disable_camera_led=1
```

Um die `config.txt` bearbeiten zu können, benötigen Sie neben `root`-Berechtigungen auch einen Editor – hier kommt nano zum Einsatz. Mit dem Kommando

```
sudo nano /boot/config.txt
```

öffnen Sie die Datei. Nutzen Sie dann die Pfeiltasten, um zum Dateiende zu gelangen. Dort tragen Sie nun den obigen Eintrag `disable_camera_led=1` ein und drücken anschließend die Tastenkombination Strg + X, um den nano-Editor zu beenden. Dies bestätigen Sie mit Drücken der Y-Taste, gefolgt von Enter. Nun sollte beim Anfertigen einer Aufnahme beispielsweise durch `raspistill` die optische Benachrichtigung ausgeschaltet sein. Die Änderung wird erst nach einem Neustart des Raspberry Pi aktiv.

1.3.6 Programmierung der Raspberry-Pi-Kamera

Wie auf dem Raspberry Pi üblich, können die mit dem Computer verbundenen Geräte und angeschlossenen Gadgets über die Kommandozeile mit den Standardwerkzeugen des Betriebssystems angesprochen und genutzt werden. Dies gilt gleichermaßen für eine am USB-Anschluss angeschlossene Webcam wie auch für eine im CSI-Anschluss angesteckte Raspberry-Pi-Kamera, die sich wie der Raspberry Pi standardmäßig ohne Gehäuse im einfachen Platinenlook zeigt und mithilfe des 15-poligen Flachbandkabels mit dem Raspberry Pi verbunden ist.

Ist die Kamera ordnungsgemäß am Betriebssystem angemeldet, können Sie prinzipiell jede Skript- und Programmiersprache nutzen. Gerade für den Einsteiger empfiehlt es sich, zunächst auf Bewährtes zurückzugreifen und möglichst verfügbare Bibliotheken und APIs für die eigenen Programme und Skripte zu verwenden. So stehen für nahezu sämtliche Anwendungszwecke solche Bibliotheken und APIs zur Verfügung. Manche Perlen müssen Sie wirklich suchen, denn bei der Vielzahl an Möglichkeiten – Repository-Verwaltung der Marktführer *github* und *sourceforge*, aber auch zig unterschiedliche Foren und Blogs – geht manchmal der Überblick etwas verloren.

Auf der sicheren Seite sind Sie, wenn Sie sich zunächst auf github und sourceforge umsehen, von dort die eine oder andere API auf den Raspberry Pi herunterladen und einfach mal ausprobieren, ob Sie mit dem, was der Entwickler für die Open-Source-Gemeinde zur Verfügung gestellt hat, überhaupt etwas anfangen können. Für die Raspberry-Pi-Kamera gibt es Module und Erweiterungen wie Sand am Meer, aber – weniger ist mehr: Die meisten APIs sind redundant oder teilweise auch schlicht und ergreifend nutzlos, da mittlerweile bereits verbesserte Versionen wie `raspistill` und `raspivid` zur Verfügung stehen. Es gibt aber dennoch gerade für Entwickler Praktisches zu entdecken, das nicht nur in Sachen Codeoptimierung und Lesbarkeit, sondern auch beim Programmieraufwand als solchem wertvolle Hilfe leisten kann. Für die Raspberry-Pi-Kamera lohnt sich für den Python-Entwickler beispielsweise das `picam`-Modul, das auf github (*https://github.com/ashtons/picam*) zur Verfügung steht.

```
mkdir picam
cd picam
wget https://github.com/ashtons/picam/archive/master.zip
mv master.zip picam-master.zip
unzip picam-master.zip
cd picam-master/
sudo python setup.py install
```

Das `python setup.py install`-Kommando sorgt dafür, dass das Python-Modul auf dem lokalen Computer zur Verfügung steht und bei der Entwicklung eines Python-Programms einfach per `import modulname` eingebunden werden kann, um die Funktionen des Moduls nutzen zu können. In diesem konkreten Fall reicht hier die Zeile

```
import picam
```

aus, um das `picam`-Modul in das eigene Python-Skript einzubinden. Um nun per Python-Aufruf eine Aufnahme anzufertigen, ist folgender Code ausreichend:

```
#!/bin/python
import picam
pic = picam.takePhoto()
pic.save('/home/pi/picam/bild.jpg')
```

Speichern Sie die Datei und führen Sie sie aus.

Erscheinen Fehlermeldungen wie beispielsweise `ImportError: No module named PIL`, ist das in der Regel auf fehlende Pakete auf dem Raspberry Pi zurückzuführen.

Im obigen Fall ist die Installation von PIL mit dem Kommando

```
sudo apt-get install python-imaging-tk
```

notwendig, um das Skript erfolgreich zu starten. Augenscheinlich ist der Aufwand etwas höher, um eine gewöhnliche Aufnahme zu erzeugen, der Vorteil der `picam`-Library liegt aber vor allem darin, bei automatisierten Aufnahmen die Kamera optimal nach bestimmten Parametern automatisch steuern und konfigurieren zu können, bevor der Auslösevorgang erfolgt.

```
picam.config.imageFX = picam.MMAL_PARAM_IMAGEFX_WATERCOLOUR
picam.config.exposure = picam.MMAL_PARAM_EXPOSUREMODE_AUTO
picam.config.meterMode = picam.MMAL_PARAM_EXPOSUREMETERINGMODE_AVERAGE
picam.config.awbMode = picam.MMAL_PARAM_AWBMODE_SHADE
picam.config.ISO = 0 #auto
picam.config.ISO = 400
picam.config.ISO = 800
picam.config.sharpness = 0              # -100 bis 100
picam.config.contrast = 0               # -100 bis 100
picam.config.brightness = 50            #  0 bis 100
picam.config.saturation = 0             #  -100 bis 100
picam.config.videoStabilisation = 0     # 0 or 1 (false oder true)
picam.config.exposureCompensation  = 0  # -10 to +10 ?
picam.config.rotation = 90              # 0-359
picam.config.hflip = 1                  # 0 or 1
picam.config.vflip = 0                  # 0 or 1
picam.config.shutterSpeed = 20000       # 0 = auto, otherwise the shutter
speed in ms
```

So lassen sich im Python-Skript Parameter wie ISO-Werte, Schärfe, Kontrast, Helligkeit und vieles mehr einstellen – binden Sie beispielsweise Sensoren und LEDs (Helligkeitssensoren, IR-LEDs etc.) mit in die Programmlogik ein, lassen sich wunderbare Automatismen schaffen.

1.3.7 Infrarotfotografie mit dem Pi-NoIR-Modul

Das Pi-NoIR-Infrarotkameramodul ist eine Variante des Raspberry-Pi-Kameramoduls für sichtbares Licht, bei dem im Gegensatz zur normalen Raspberry-Pi-Kamera kein IR-Infrarotfilter auf dem Sensor vorhanden ist. Beide Raspberry-Pi-Kameramodule haben den gleichen Bildsensor mit einer Auflösung von 5 Megapixeln – der Sony-Sensor der Version V2 des Kameramoduls liefert hingegen 8 Megapixel. Durch das Weglassen des Filters für sichtbares Licht kann die NoIR-Kamera Infrarotstrahlung fotografieren und filmen.

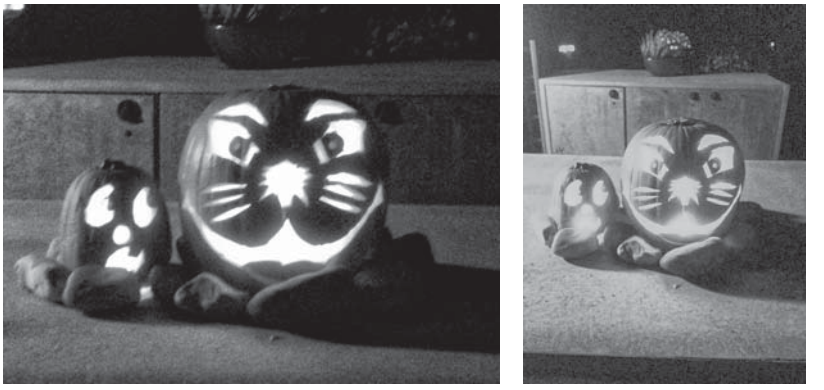

Bild 1.19: Graustufenmodell im Vergleich: links die Aufnahme mit dem normalen Kameramodul und externer Belichtung, rechts die Aufnahme bei schlechten Lichtverhältnissen mit dem NoIR-Kameramodul.

Bei Tageslicht und normalen Lichtverhältnissen nehmen sich die beiden Module nichts: Der Anwendungsbereich für das Raspberry-Pi-NoIR-Modul umfasst demnach die Alarmanlage zu Hause und die Objektüberwachung, aber auch die automatisierte Beobachtung von Tieren mithilfe eines Bewegungsmelders und dergleichen.

1.4 GPIO-Schnittstelle: Pin-Belegung und Zugriff

Neben der USB- und Netzwerkschnittstelle lädt vor allem die sogenannte GPIO-Schnittstelle (**G**eneral **P**urpose **I**nput/**O**utput) des Raspberry Pi zum Basteln und Ausprobieren ein. So bringen Sie nicht nur innerhalb kurzer Zeit eine LED auf einem Steckboard zum Leuchten, sondern realisieren ganze Schaltungen und Fernbedienungen mit dem Raspberry Pi.

Entweder Sie setzen auf Lösungen der Marke Eigenbau oder auf durchaus hilfreiche Unterstützung in Form zusätzlich zu erwerbender Steckboards wie das PiFace-Board (ca. 34 Euro inklusive Versand) oder das umfangreich bestückte Erweiterungsboard Gertboard (ca. 42 Euro inklusive Versand), das nach seinem Schöpfer Gert van Loo benannt ist. Dieses Platine ist über den britischen Elektronikdistributor *Farnell/element14* (*http://uk.farnell.com*) jetzt auch komplett erhältlich. Damit lassen sich Motoren und Roboter steuern, Türen öffnen, Geräte und Licht ein- und ausschalten und vieles mehr.

Im Rahmen der Heimautomation ist solch ein Erweiterungsboard meist zu sperrig, somit ist eine Lösung über 1-Wire oder den Eigenbau die bessere Alternative. Grundvoraussetzung ist der Zugriff per Software auf die Schnittstelle bzw. die Funktionen der einzelnen GPIO-Pins. Dafür stehen zahlreiche Möglichkeiten zur Verfügung, die in den nachfolgenden Projekten anhand praktischer Beispiele erklärt werden.

1.4.1 Aufklärung über die GPIO-Pin-Belegung

Abhängig von der Raspberry-Pi-Version sind die Pin-GPIO-Bezeichnungen leicht unterschiedlich. Um nach Kauf und Lieferung zu kontrollieren, welche Version des Raspberry Pi geliefert wurde, nutzen Sie die Kommandozeile. Mit dem Befehl

```
cat /proc/cpuinfo
```

lassen Sie sich die Hardwareinformationen – in diesem Beispiel die CPU-Prozessorinformationen – ausgeben. In der tabellarischen Ausgabe suchen Sie nach dem Eintrag `Revision` – hier steht für den Code 1 das Modell A.

```
pi@raspberrypi ~ $ cat /proc/cpuinfo
Processor       : ARMv6-compatible processor rev 7 (v6l)
BogoMIPS        : 697.95
Features        : swp half thumb fastmult vfp edsp java tls
CPU implementer : 0x41
CPU architecture: 7
CPU variant     : 0x0
CPU part        : 0xb76
CPU revision    : 7

Hardware        : BCM2708
Revision        : 0003
Serial          : 00000000        e0
pi@raspberrypi ~ $
```

Bild 1.20: Für den Raspberry-Pi-1-B-Nachfolger bzw. eine weitere, unwesentlich geänderte Revision 3 wird der Code 2 genutzt. Für das Raspberry Pi 1 Modell B Revision 2 werden die Codes 4, 5 und 6 genutzt.

Für die Nummerierung der Pins auf der Platine ist es egal, welche Revision der Raspberry Pi hat – die Zählrichtung ausgehend von Pin 1 ist immer dieselbe.

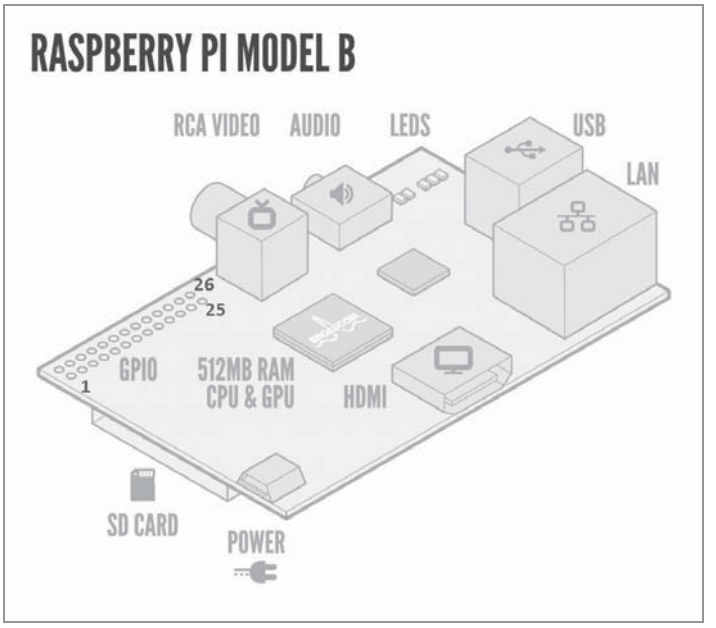

Bild 1.21: Der grundsätzliche Aufbau des Raspberry Pi und der GPIO-Pin-Leiste (Grafik: *raspberrypi.org*)

In Sachen C-Programmierung und Shell-Zugriff lohnt es sich, die kostenlose WiringPi-API näher zu betrachten. Ähnlich wie die oben genannte Python-Bibliothek bietet sie einfacheren Zugriff auf die GPIO-Pins des Raspberry Pi.

1.4.2 Direkter GPIO-Zugriff mit WiringPi

Warum das Rad neu erfinden, wenn es an nahezu jeder Ecke einen Reifenhändler gibt? Gerade beim Erstellen von Shell-Skripten in C oder Python ist der Umgang mit den GPIO-Anschlüssen relativ einfach gelöst. Für mehr Möglichkeiten beim Programmieren und vor allem mehr Übersicht sorgt die Auslagerung von Funktionen in eine API-Schnittstelle (*Advanced Programming Interface*). Im Rahmen dieses Buchs greifen wir auf die äußerst praktische WiringPi-API des Entwicklers Gordon Drogon (*https://projects.drogon.net/raspberry-pi/wiringpi/download-and-install/*) zurück.

Doch bevor Sie diese API installieren, achten Sie darauf, das System auf dem Raspberry Pi auf den aktuellen Stand zu bringen. Wie gewohnt, nutzen Sie dafür das entsprechende `update`- bzw. `upgrade`-Programm von Raspian:

```
sudo apt-get update
sudo apt-get upgrade
```

Haben Sie die GIT-Versionsverwaltung auf dem Raspberry Pi installiert, dann klonen Sie das WiringPi-Paket auf Ihren Computer. Dies erledigen Sie mit dem Kommando

```
git clone git://git.drogon.net/wiringPi
```