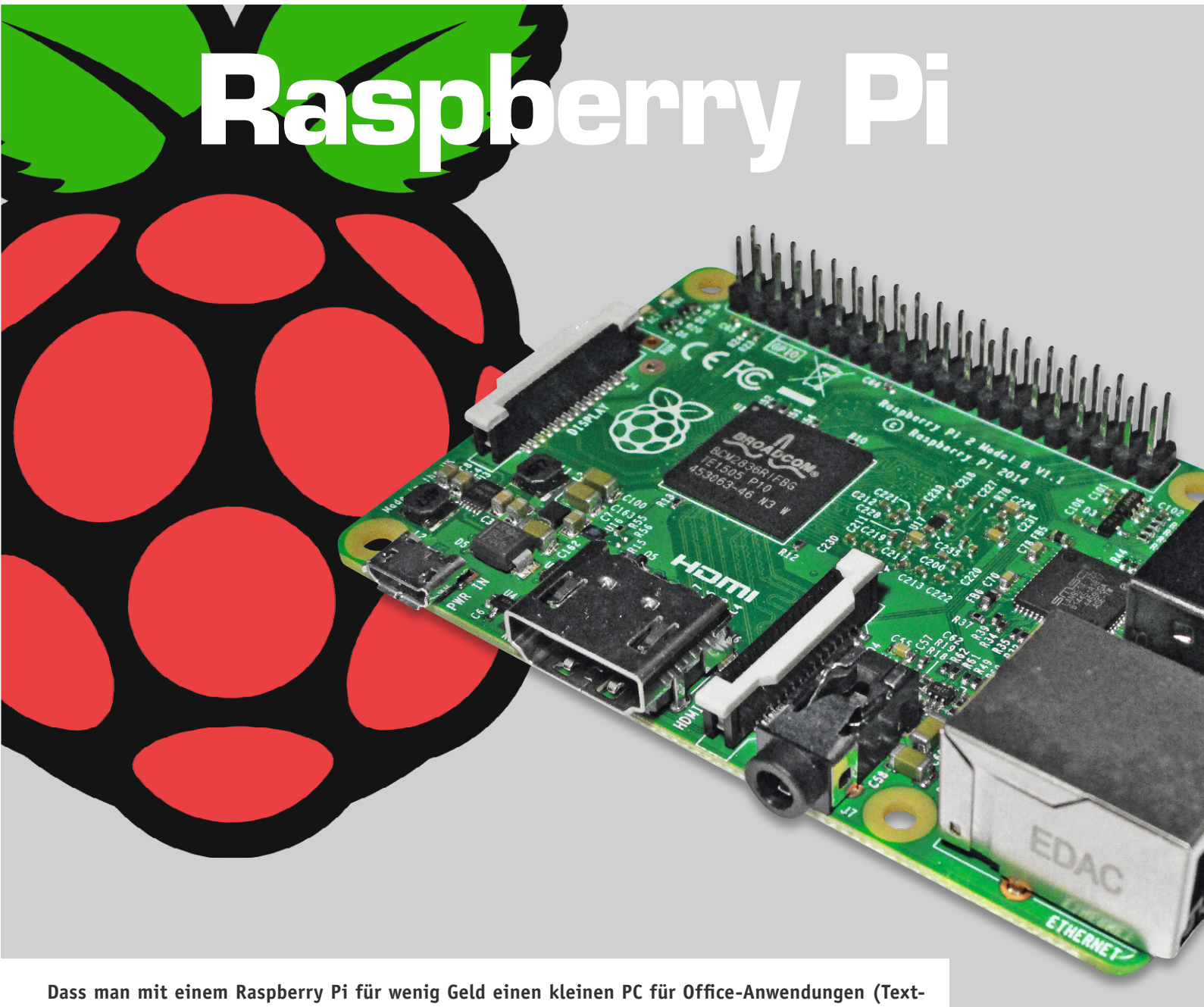
Vom Start bis zur Home-Automation

Raspberry Pi



Dass man mit einem Raspberry Pi für wenig Geld einen kleinen PC für Office-Anwendungen (Textverarbeitung, Tabellenkalkulation, Websurfen ...), Multimedia, Kioskbetrieb oder für elektronische Projekte bekommt, macht ihn sehr attraktiv. Dass man mit geringen Vorkenntnissen einen eigenen preiswerten, kleinen, stromsparenden und lautlosen Webserver bzw. Home-Automation-Server erstellen kann, macht den Raspberry noch attraktiver und wird in diesem Artikel dargestellt.

Teil 4: Netzzugriff

Im vorliegenden Artikel wird der Weg zur Home-Automation in drei Blöcken beschrieben:
- Zugriff auf den Raspberry Pi über SSH
- Zugriff mit Browser
- Zugriff aus dem Internet

Die IP-Adresse des Raspberry Pi

Um den Raspberry Pi über das LAN/WLAN oder über das Internet anzusprechen, wird dessen IP-Adresse benötigt. Die schnellste Möglichkeit, die IP-Adresse herauszufinden, ist die Eingabe des Befehls *ifconfig* in einem Befehlszeilenfenster. In der Ausgabe des Befehls (Bild 1) sieht man unter dem entsprechenden Adapter – in Bild 1 für einen Raspberry Pi, der per WLAN-Dongle im Netz ist – die IP-Adresse des Raspberry Pi. Im Beispiel: 192.168.2.59.

Falls man keinen direkten Zugang zum Raspberry hat – zum Beispiel, weil dieser ohne Tastatur und Bildschirm („headless") betrieben wird oder sich fest eingebaut oder hoch platziert befindet –, kann man über das Menü des Routers den Raspberry mit seiner IP-Adresse identifizieren.

Wenn man auch keinen Zugriff auf den Router hat, dann kann man das Programm „Angry IP Scanner" [1] verwenden, um alle vergebenen IP-Adressen im eigenen Netzwerk anzuzeigen. Dazu sollte man möglichst viele andere Geräte ausschalten, um den Raspberry Pi sicher erkennen zu können.

SSH einschalten in Raspi-Config

Durch Eingabe von *sudo raspi-config* in einem Befehlszeilenfenster gelangt man zum Konfigurationsmenü des Raspberry Pi.

Unter Punkt 8 (Advanced Options) wird der SSH-Netzwerkzugang zum Raspberry Pi aktiviert:
sudo raspi-config
8 – Advanced Options
A4 – SSH
<Enable> (siehe Bild 2)

```
pi@raspberrypi ~ $ ifconfig
eth0      Link encap:Ethernet  Hardware Adresse b8:27:eb:57:96:32
          UP BROADCAST MULTICAST  MTU:1500  Metrik:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          Kollisionen:0 Sendewarteschlangenlänge:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo        Link encap:Lokale Schleife
          inet Adresse:127.0.0.1  Maske:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metrik:1
          RX packets:12 errors:0 dropped:0 overruns:0 frame:0
          TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
          Kollisionen:0 Sendewarteschlangenlänge:0
          RX bytes:1304 (1.2 KiB)  TX bytes:1304 (1.2 KiB)

wlan0     Link encap:Ethernet  Hardware Adresse ac:a2:13:22:82:37
          inet Adresse:192.168.2.59  Bcast:192.168.2.255  Maske:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metrik:1
          RX packets:2285148 errors:0 dropped:0 overruns:0 frame:0
          TX packets:17909 errors:0 dropped:0 overruns:0 carrier:0
          Kollisionen:0 Sendewarteschlangenlänge:1000
          RX bytes:545363629 (520.0 MiB)  TX bytes:1967764 (1.8 MiB)
```

Bild 1: Ermittlung der IP-Adresse mit „ifconfig"

Bild 2: Einschalten von SSH im Konfigurationsmenü Raspi-Config

Vorbereitungen

In Bild 3 ist die für diesen Artikel verwendete Demo-Schaltung zu sehen. An Pin 25 des Raspberry befindet sich zu Testzwecken eine LED. Für eine Home-Automation-Umgebung würde man statt der LED einen Transistor/MOSFET ansteuern, mit dem wiederum ein Relais zum Schalten einer Lampe oder einer Pumpe o. Ä. angesteuert werden könnte. Statt eines mechanischen Relais könnte man auch ein Solid-State-Relais verwenden. Besonders interessant wird es, wenn man an die Pins des Raspberry Pi Komponenten aus der HomeMatic-Serie oder der FS20-Serie anschließt. Es bietet sich hier das HomeMatic 8-Kanal-Sendemodul (CF–13 29 39) oder ein FS20-Sendemodul (CF-06 68 17 oder CF-09 22 04) an. So lassen sich mit den in diesem Artikel beschriebenen Techniken die Pins des Raspberry Pi ansteuern, und über das

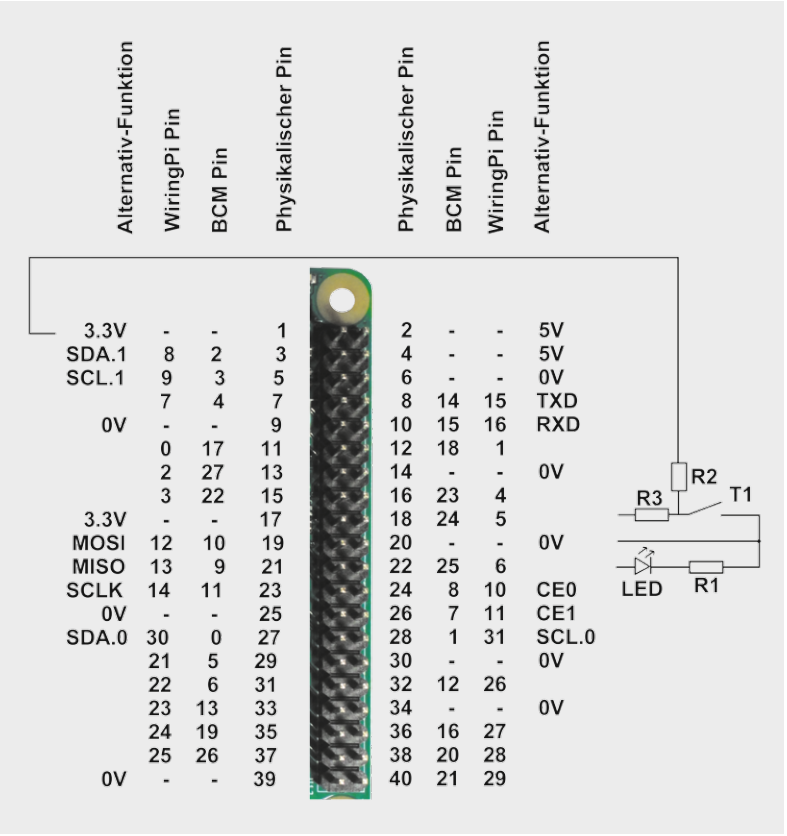


Bild 3: GPIO-Header des Raspberry Pi 2 mit Beispielbeschaltung

HomeMatic/FS20-Sendemodul werden die entsprechenden Pinzustände zum Schalten von Verbrauchern verwendet, die im Haus verteilt stehen und per Funk angesteuert werden.

```
#!/bin/bash
# Aufruf mit    bash lampe_an.sh
# GPIO Ausgabe
# LED anschalten an GPIO25

if [ -d /sys/class/gpio/gpio25 ]                    # Wenn das Verzeichnis existiert
then
  echo "alles ok"
else                                                # sonst
  echo "25" > /sys/class/gpio/export                # GPIO25 anlegen
  sudo chmod 666 /sys/class/gpio/gpio25/direction   # Schreibrechte vergeben
  sudo chmod 666 /sys/class/gpio/gpio25/value       # Schreibrechte vergeben
  echo "out" > /sys/class/gpio/gpio25/direction     # Richtung: "out" = Ausgabe   "in" = Eingabe
fi

echo „Lampe anschalten"                             # Text zur Kontrolle ausgeben
echo "1" > /sys/class/gpio/gpio25/value             # "1" = Ausgabe von 3,3 Volt
```

Bild 4: Skript lampe_an.sh

```
#!/bin/bash
# Aufruf mit    bash lampe_aus.sh
# GPIO Ausgabe
# LED ausschalten an GPIO25

if [ -d /sys/class/gpio/gpio25 ]                    # Wenn Verzeichnis existiert
then
  echo "alles ok"                                   # sonst
else
  echo "25" > /sys/class/gpio/export                # GPIO25 anlegen
  sudo chmod 666 /sys/class/gpio/gpio25/direction   # Schreibrechte vergeben
  sudo chmod 666 /sys/class/gpio/gpio25/value       # Schreibrechte vergeben
  echo "out" > /sys/class/gpio/gpio25/direction     # Richtung: "out" = Ausgabe   "in" = Eingabe
fi

echo "Lampe ausschalten"                            # Text zur Kontrolle ausgeben
echo "0" > /sys/class/gpio/gpio25/value             # "0" = Ausgabe von 0 Volt
```

Bild 5: Skript lampe_aus.sh

```
#!/bin/bash
# Aufruf mit     bash pinabfrage.sh
# GPIO Eingabe
# Taster/Schalter/Jumper an GPIO24

echo "24" > /sys/class/gpio/export                        # GPIO24 anlegen
sudo chmod 666 /sys/class/gpio/gpio24/direction           # Schreibrechte vergeben
sudo chmod 666 /sys/class/gpio/gpio24/value               # Schreibrechte vergeben
echo "in" > /sys/class/gpio/gpio24/direction              # Richtung: "out" = Ausgabe   "in" = Eingabe

echo "Pinabfrage: "                                       #  Textausgabe zur Kontrolle
# Nur Anzeige des Pin-Zustandes:
# cat /sys/class/gpio/gpio24/value                        # "1" = 3,3 Volt liegt an GPIO-Pin GPIO24
                                                          # "0" = 0 Volt liegt an GPIO-Pin GPIO24
# Mit Auswertung:
wert_am_pin=$(cat /sys/class/gpio/gpio24/value)           # Wert am Pin einer Variablen zuweisen
if [ $wert_am_pin -eq 1 ]                                 # Wenn "1" am Pin anliegt ...
then                                                      # dann
  echo "Kontakt offen"                                    # "Kontakt offen" ausgeben
else                                                      # sonst
  echo "Kontakt geschlossen"                              # "Kontakt geschlossen" ausgeben
fi

echo "24" > /sys/class/gpio/unexport                      # Bereinigung
```

Bild 6: Skript pinabfrage.sh

An Pin 24 ist außerdem beispielhaft ein Kontakt zu sehen (Bild 3). Hier kann im Praxiseinsatz ein Tür-/Fensterkontakt oder etwas Ähnliches angeschlossen werden.

Die Bilder 4, 5 und 6 zeigen kleine (nichtoptimierte) Beispielskripte für das An- bzw. Ausschalten von Geräten und für das Abfragen eines Pins.

SSH-Zugriff von Windows

Nachdem SSH auf dem Raspberry aktiviert wurde und die IP-Adresse des Raspberry bekannt ist, kann man von innerhalb des heimischen (W)LANs per SSH auf den Raspberry zugreifen.

Dazu benötigt man einen sogenannten SSH-Client – also ein Programm, mit dessen Hilfe man von einem Client per SSH auf den Server (Raspberry) zugreifen kann.

Für Windows-Systeme hat sich ein SSH-Client namens PuTTY bewährt, den man unter [2] herunterladen kann.

Nach der Installation und dem Aufruf von *putty.exe* werden die Verbindungsdaten eingegeben (Bild 7). Im Wesentlichen ist hier die zuvor ermittelte IP-Adresse des Raspberry Pi und als Port 22 (für SSH-Zugriff) einzugeben.

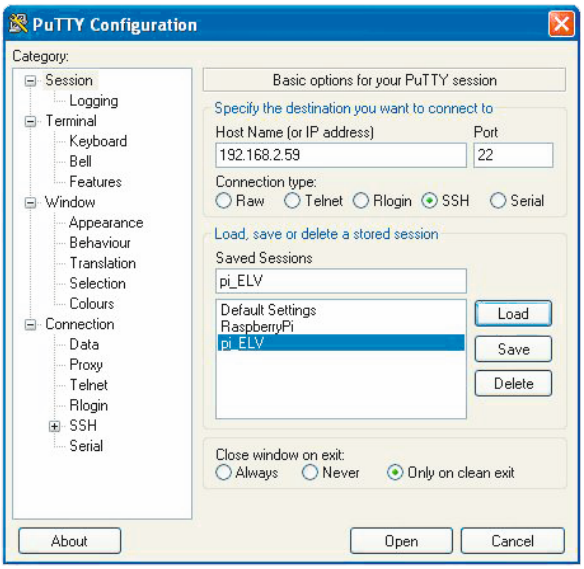


Bild 7: PuTTY-Konfiguration

Im Konfigurationsfenster von PuTTY lassen sich Konfigurationen für spätere Verwendung speichern. Das ist sehr nützlich, wenn auf verschiedene SSH-Server zugegriffen werden soll.

Nach Anklicken von *Open* erhält man einen PuTTY-Bildschirm, der einem Befehlszeilenfenster im Raspbian ähnelt. Dort hat man nach Eingabe von Userid und Kennwort ein Terminalfenster mit einem Prompt (Bild 8).

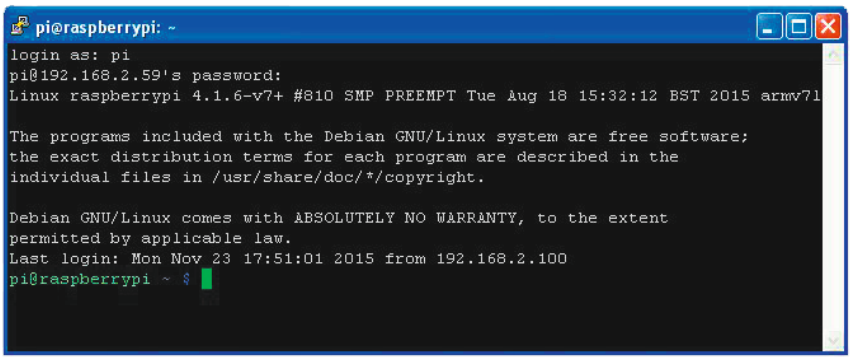


Bild 8: PuTTY-Fenster

In diesem PuTTY-Befehlszeilenfenster kann man auf dem Raspberry arbeiten, als säße man direkt davor. Man kann beispielsweise einen GPIO-Pin des Raspberry ein- oder ausschalten oder einen Pin abfragen (Bild 9).

Auf diese Weise kann man innerhalb des Heimnetzwerks Lampen ein- oder ausschalten oder Kontakte abfragen.

```
pi@raspberrypi ~ $ bash lampe_an.sh
alles ok
Lampe anschalten
pi@raspberrypi ~ $ bash lampe_aus.sh
alles ok
Lampe ausschalten
pi@raspberrypi ~ $ bash pinabfrage.sh
Pinabfrage:
Kontakt geschlossen
pi@raspberrypi ~ $
```

Bild 9: „Fernbedienung" des Raspberry Pi über einen Windows-PC mittels PuTTY

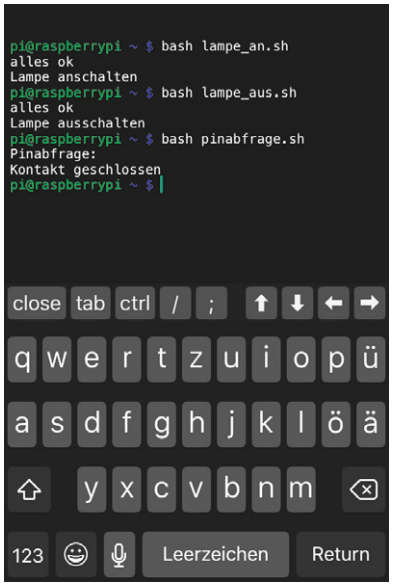


Bild 10: SimpleSSH auf iOS – Terminalfenster

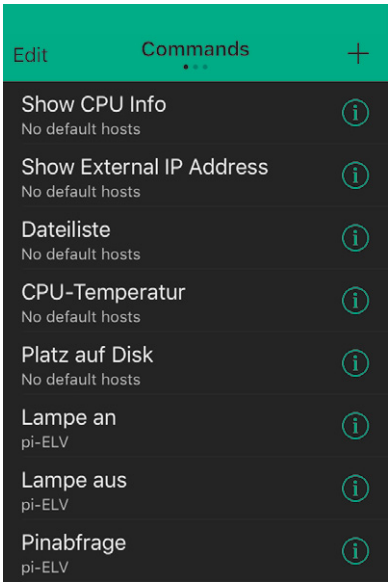


Bild 11: SimpleSSH auf iOS – Kommandos

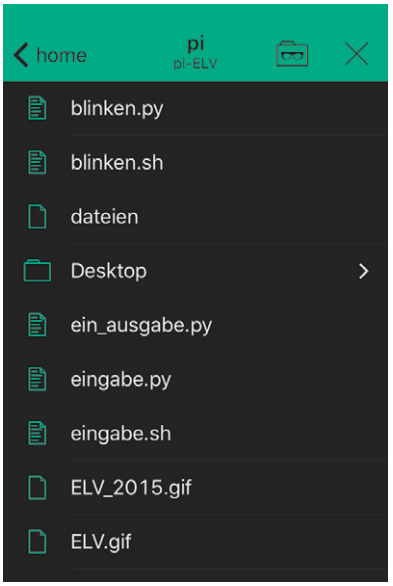


Bild 12: SimpleSSH auf iOS – Dateimanager

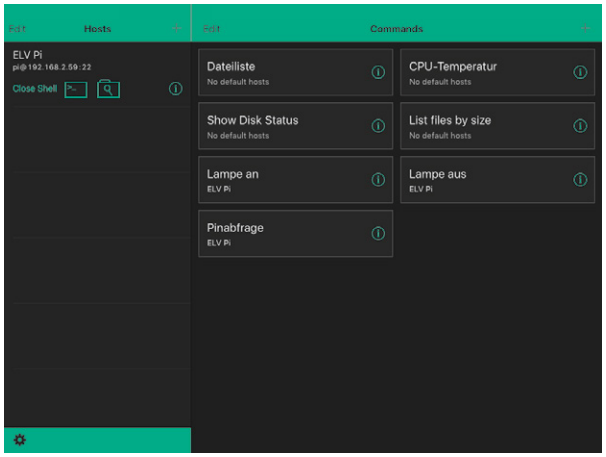


Bild 13: SimpleSSH auf iPad

SSH-Zugriff von OS X

Auf OS-X-Geräten kann man das integrierte Terminalprogramm über *Finder – Programme – Dienstprogramme – Terminal* aufrufen. Dann kann man über das Menü *Shell – Neue entfernte Verbindung* (Shift-Befehlstaste-K) zu einem Konfigurationsfenster gelangen, in dem man *Sichere Shell (ssh)* als Dienst anwählt, einen Server (IP-Adresse) mit + hinzufügt und im unteren Bereich des Konfigurationsfensters den Benutzernamen (und ggf. die IP-Adresse des SSH-Servers) eingibt. Nach dem Verbinden mit dem SSH-Server muss nur noch das Kennwort eingegeben werden (der Cursor bewegt sich dabei aus Sicherheitsgründen nicht), und man kann Befehle auf dem Raspberry eingeben.

Alternativ könnte man im normalen OS-X-Terminalfenster eingeben: *ssh benutzername@ipadresse*.

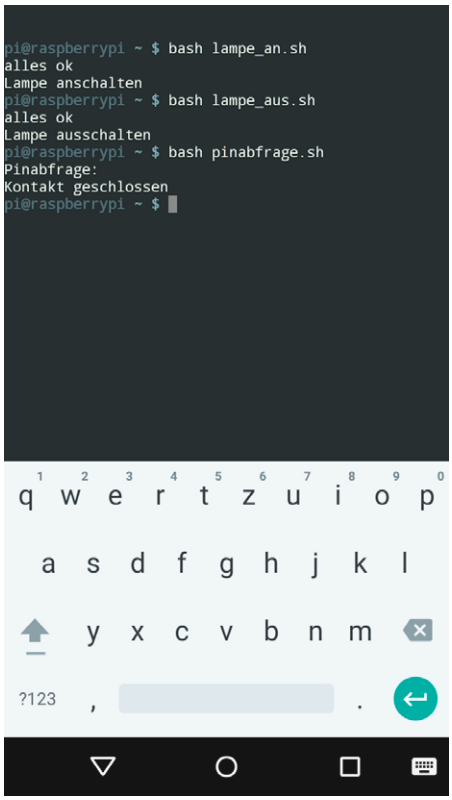


Bild 14: SSH-Client JuiceSSH auf Android

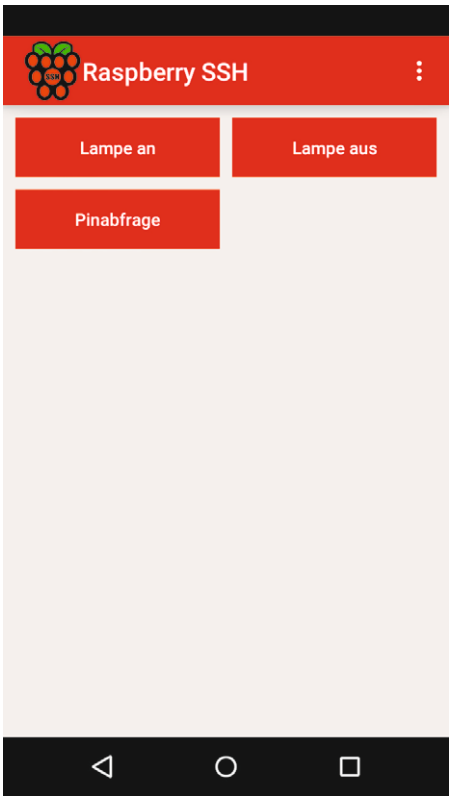


Bild 15: Raspberry SSH auf Android

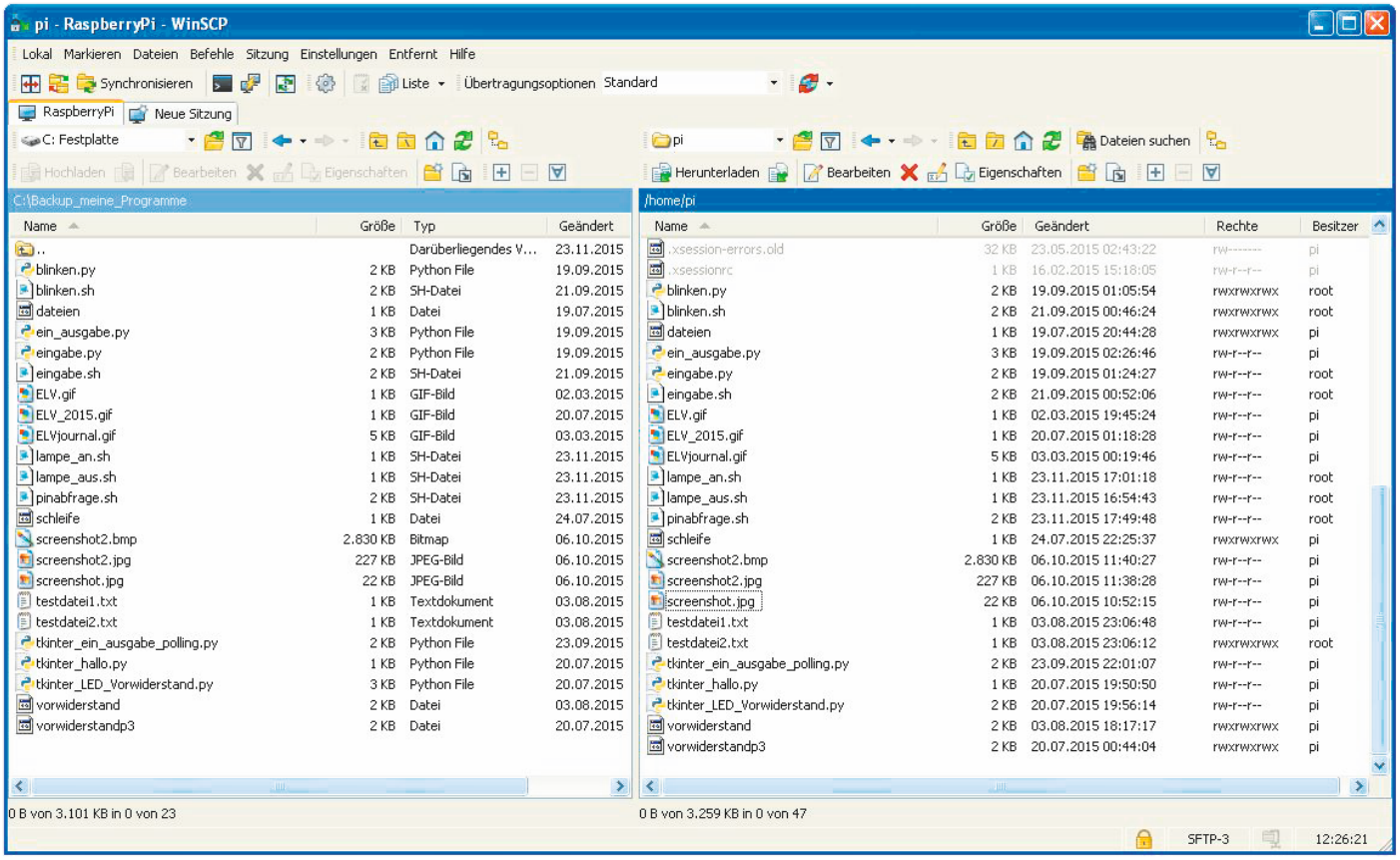


Bild 16: Dateimanager WinSCP

SSH-Zugriff von iOS-Geräten

Für den bequemen Zugriff auf den Raspberry Pi von einem iPhone oder iPad aus eignet sich sehr gut die App „SimpleSSH“ [3]. Mit dieser App ist es sowohl möglich, auf dem iOS-Gerät Linux-Befehle einzugeben wie in PuTTY unter Windows (Bild 10) als auch Befehle unter einem sprechenden Namen abzuspeichern (Bild 11). Außerdem gibt es einen Dateimanager (Bild 12). Die App lässt sich aus dem App Store laden. Insbesondere durch die Möglichkeit, Befehle zu speichern (Bild 11), kann man mit dieser App auf elegante Weise und bequem Verbraucher schalten oder Parameter (Kontaktstatus, Temperatur ...) abfragen. Bild 13 zeigt die Oberfläche der App auf einem iPad.

SSH-Zugriff von Android

Selbstverständlich gibt es auch SSH-Clients für Android, zum Beispiel JuiceSSH [4] für Befehlseingaben (Bild 14) oder die sehr schöne App „Raspberry SSH“ (Bild 15). Beide und natürlich viele weitere SSH-Apps lassen sich aus dem Play Store herunterladen.

SSH-Dateimanager WinSCP

Ein sehr nützliches Werkzeug ist der Dateimanager WinSCP (Bild 16), mit dem sich Dateibäume von PC und Raspberry Pi gleichzeitig anzeigen und Dateien sehr schön hin und her kopieren lassen. Man kann mit WinSCP zum Beispiel sehr einfach Dateien zu Backup-Zwecken oder zum Weiterbearbeiten vom Raspberry auf den PC kopieren. Durch Doppelklick auf eine Datei sieht man den Inhalt der Datei im Editor. Mit Rechtsklick kann man unter „Eigenschaften“ die Zugriffsrechte einer Datei verändern. WinSCP lässt sich unter [5] kostenfrei für Windows herunterladen. Bei der Anmeldung muss man lediglich wieder als Rechnernamen die IP-Adresse des Raspberry Pi, die Portnummer 22 (für SSH), Übertragungsprotokoll SFTP sowie Benutzernamen und Kennwort eingeben.

Apache-Server und PHP

Wer sich ein wenig mit HTML-Seitengestaltung beschäftigt hat, weiß, dass man keinen Webserver braucht, um mit HTML-Seitengestaltung anzufangen. Es reicht, HTML-Dateien lokal in einer Datei zu erstellen und diese dann mit einem Webbrowser aufzurufen. Das ist für erste, einfache Tests schon ganz nützlich, aber um die Seiten im Netz zugänglich zu machen, benötigt man einen Webserver (HTTP-Server) wie zum Beispiel den verbreiteten Apache-Server.

Die Installation von Apache auf dem Raspberry Pi erfolgt folgendermaßen:

pi@raspberrypi ~ $ sudo apt-get update
pi@raspberrypi ~ $ sudo apt-get install apache2 –y

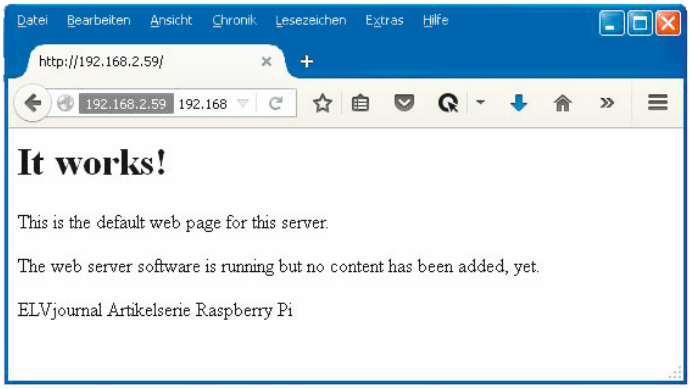


Bild 17: Welcome-Seite Apache

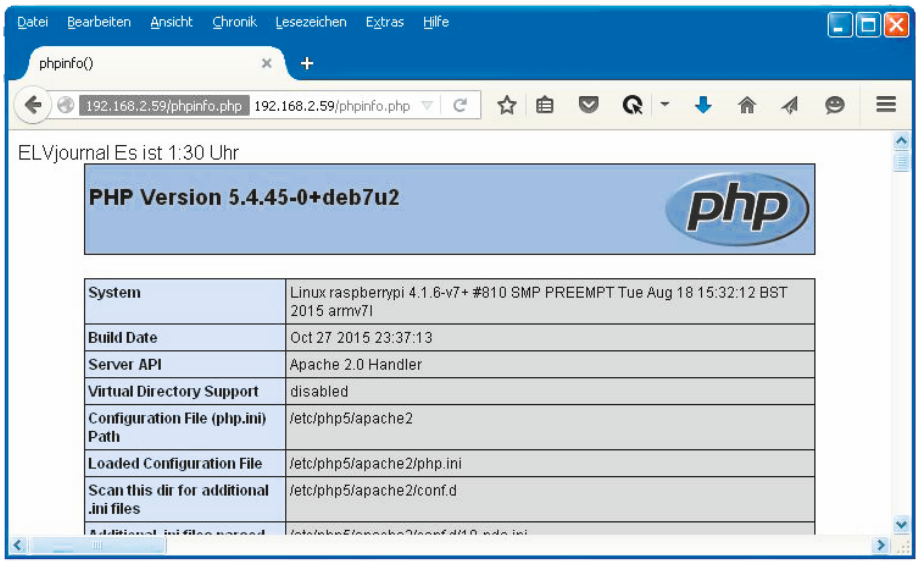


Bild 18: Welcome-Seite PHP

Nach der Installation kann man testen, welche Version von Apache installliert ist:

pi@raspberrypi ~ $ sudo apachectl –v

Nachdem der Apache-Server installiert ist, kann man im Browser auf dem Raspberry als URL eingeben: *localhost* oder *127.0.0.1*

Von einem anderen Computer im selben Netzwerk kann man die Einstiegsseite des Apache-Servers durch Eingabe der IP-Adresse des Raspberry im Browser (zum Beispiel 192.168.2.59) erreichen.

In beiden Fällen erhält man eine Anzeige ähnlich wie in Bild 17.

```
<html>
<head>
<meta name="viewport" content="width=device-width"/>
<title>ELVjournal GPIO &uumlber PHP und wiringPi schalten</title>
</head>
<body>
GPIOs schalten und abfragen

<form method="get" action="php_out_in.php">
<input type="submit" value="LEDan" name="Befehl">
<input type ="submit" value="LEDaus" name="Befehl">
<input type ="submit" value="Pintest" name="Befehl">


<?php
# Pin 25 als Ausgang und Pin 24 als Eingang definieren:
$modeon25 = trim(@shell_exec("/usr/local/bin/gpio -g mode 25 out"));
$modeon24 = trim(@shell_exec("/usr/local/bin/gpio -g mode 24 in"));

# Testen, ob ein Befehl mit URL mitgegeben wurde:
# Wenn ja: Befehl anzeigen
if (isset($_GET["Befehl"]))  {
  echo "<br><br>  Befehl vorhanden  <br>";
  echo "  Befehl: ". htmlspecialchars($_GET["Befehl"])."<br>";
};


# Wenn Befehl mit URL mitgegeben dann Befehl auswerten:
if (isset($_GET["Befehl"])){
  if ($_GET["Befehl"] === "LEDan") {
    $val = trim(@shell_exec("/usr/local/bin/gpio -g write 25 1"));
    echo "LED 25 an";
  }
  else if ($_GET["Befehl"] === "LEDaus") {
    $val = trim(@shell_exec("/usr/local/bin/gpio -g write 25 0"));
    echo "LED 25 aus";
  }
  else if ($_GET["Befehl"] === "Pintest") {
    $val = trim(@shell_exec("/usr/local/bin/gpio -g read 24"));
    echo "Pinstatus Pin 24: ".$val;
  }
}
?>

</body>
</html>
```

Bild 19: PHP-Programm zum Steuern und Abfragen von Portpins

Die HTML-Dateien befinden sich standardmäßig im Verzeichnis /var/www. Dort findet man auch die Datei /var/www/index.html, die als Begrüßungsseite aufgerufen wird, wenn im Browser nur die IP-Adresse des Raspberry angegeben wird. (Beim neuen Raspbian Jessie: /var/www/html/index.html.) In diesem Verzeichnis kann man mit *sudo nano index.html* die Apache-Begrüßungsseite individuell verändern.

Für Bild 17 wurde die letzte Zeile in der HTML-Datei angefügt. Der Raspberry Pi ist nun ein Server im lokalen Netz und es können beispielsweise Informationsseiten, Fotos o. Ä. für alle Teilnehmer des lokalen Netzes zur Verfügung gestellt werden.

Da die reine HTML-Seitengestaltung oft nicht genügend Möglichkeiten für Interaktivität bzw. Rechnen bietet, wird darüber hinaus noch PHP installiert:

pi@raspberrypi ~ $ sudo apt-get install php5 libapache2-mod-php5 –y

Nach erfolgreicher PHP-Installation kann man im Verzeichnis /var/www mit *sudo nano phpinfo.php* eine einfache erste PHP-Seite anlegen mit folgendem Inhalt:

```
<?php
echo "  ELVjournal  ";
echo "  Es ist ".date("G:i")." Uhr";
phpinfo();
?>
```

Wird der Name der PHP-Datei als letzter Teil der URL in einem Webbrowser im selben Netzwerk eingegeben (zum Beispiel *192.168.2.59/phpinfo.php*), holt sich der Browser die entsprechenden Informationen vom Raspberry Pi (Bild 18).

Nachdem nun Apache und PHP installiert sind und funktionieren, kann man eine PHP-Anwendung zum Ansteuern und Abfragen von Portpins erstellen.

In Bild 19 ist ein beispielhaftes PHP-Programm abgebildet, mit dem ein Pin des Raspberry ein- bzw. ausgeschaltet und der Zustand eines anderen Pins abgefragt werden kann. Die Eingabe des PHP-Programms erfolgte im Nano-Editor mit *sudo nano php_out_in.php*.

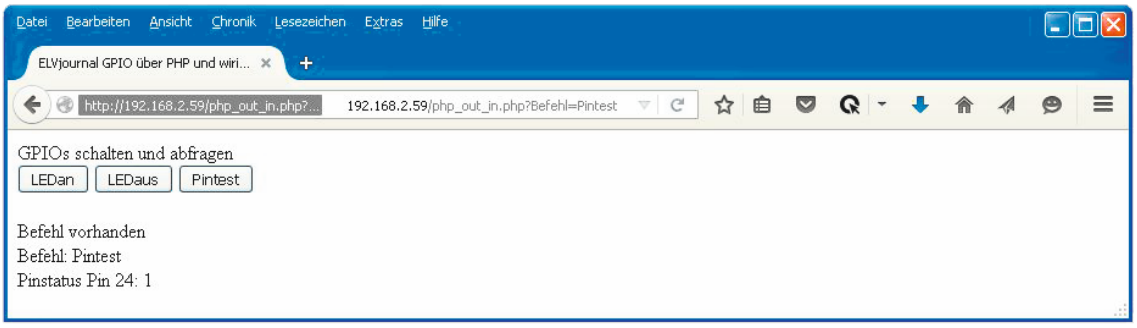


Bild 20: Im Webbrowser

Erläuterungen:
Das PHP-Programm wird aufgerufen, indem in einem Browser im selben lokalen Netzwerk als URL eingegeben wird: *192.168.2.59/php_out_in.php*. Die URL setzt sich wieder aus der IP-Adresse des Raspberry und dem Dateinamen der PHP-Datei zusammen. Bild 20 zeigt das Ergebnis des Aufrufs.

Wie in Bild 19 zu verfolgen ist, werden zunächst drei Pushbuttons definiert sowie beschrieben, was passieren soll, wenn der jeweilige Pushbutton gedrückt wird. Der „Trick" ist hier, dass beim Klicken eines Pushbuttons die Datei wieder aufgerufen und dabei der jeweilige Befehl mitgegeben wird (in Bild 20 in der URL-Zeile zu sehen).

Im PHP-Teil werden die Pins auf Ausgabe bzw. Eingabe gesetzt. Dafür wird WiringPi verwendet (vgl. ELVjournal 6/2015). Zu Kontrollzwecken wird mit *isset* abgefragt, ob die PHP-Datei mit einem Parameter/Befehl aufgerufen wurde. Falls dies der Fall ist, wird der Befehl angezeigt. (Diesen Teil kann man natürlich nach dem Testen weglassen bzw. auskommentieren.)

Die eigentliche Auswertung besteht darin, dass – falls ein Befehl mitgegeben wurde – geprüft wird, welcher Befehl (LEDan, LEDaus, Pintest) mit der URL mitgegeben wurde und die dementsprechende Aktion (Pin einschalten, Pin ausschalten, Pin abfragen) per WiringPi ausgeführt wird.

Durch individuelle Erweiterung des PHP-Programms (mehr Pushbuttons, andere Beschriftungen usw.) kann man per Webbrowser innerhalb des Heimnetzwerks auf die Pins des GPIO-Ports (und andere Ressourcen) des Raspberry Pi zugreifen.

Statische IP-Adresse für den Raspberry Pi

Im Normalfall bekommt jedes Gerät – ob PC, Raspberry Pi, Smartphone oder was auch immer – beim Einschalten eine IP-Adresse vom Router zugewiesen. Nun kann es vorkommen, dass ein Gerät beim nächsten Einschalten im Netz bzw. einem Neustart des Routers eine andere IP-Adresse zugewiesen bekommt. Meistens ist das kein Problem, weil man auf die wenigsten Geräte per IP-Adresse zugreifen muss, wie in diesem Artikel beschrieben. Wenn eine andere IP-Adresse vergeben wurde, könnte man nicht mehr unter derselben IP-Adresse auf den Raspberry zugreifen wie beim Mal davor. Um nun nicht jedes Mal von Neuem die IP-Adresse des Raspberry Pi nachsehen und neu in Programme oder Smartphone-Apps eingeben zu müssen, sollte man dafür sorgen, dass der Raspberry jedes Mal dieselbe IP-Adresse bekommt. Man spricht von einer „statischen" IP-Adresse.

Zum Einrichten einer statischen IP-Adresse für den Raspberry gibt es zwei Möglichkeiten:
- Einstellen auf dem Raspberry
- Einstellen im Konfigurationsmenü des Routers

Das Festlegen einer statischen IP-Adresse auf dem Raspberry geschieht in vier Schritten:

1. **Klarheit über die Adressen verschaffen**
   Mit *ifconfig* (siehe Bild 1) und ggf. im Router-Menü nachsehen, welche IP-Adresse der Raspberry bisher hat (z. B. 192.168.2.59) und welche Adressen der Router für DHCP (= automatische IP-Adressenvergabe durch den Router) eingestellt hat (zum Beispiel 192.168.2.**50** bis 192.168.2.**199**). Man kann auch das Programm „Angry IP Scanner" [1] benutzen, um am PC die vergebenen IP-Adressen nachzusehen. Da für den Raspberry Pi eine statische IP-Adresse außerhalb des DHCP-Adressbereiches vergeben werden soll und der Router selbst oftmals die Adresse xxx.xxx.xxx.1 hat (z. B. 192.168.2.1), kann man zum Beispiel die Adresse 192.168.2.2 für den Raspberry Pi wählen.

2. **Die Datei wpa_supplicant.conf editieren**
   pi@raspberrypi ~ $ *sudo nano /etc/wpa_supplicant/wpa_supplicant.conf*
   Die Datei sieht ähnlich aus wie in Bild 21. Hinter „ssid" muss der Name des WLAN-Netzes stehen und hinter „psk" steht das Kennwort für das WLAN. Das Wichtigste ist nun, eine Zeile mit dem Inhalt *id_str="elv"* hinzuzufügen (Bild 21).
   Statt elv kann selbstverständlich auch ein anderer Name gewählt werden.

3. **Die Datei interfaces editieren**
   pi@raspberrypi ~ $ sudo nano /etc/network/interfaces
   In der Datei interfaces müssen nun die bisher vorhandenen Einträge ggf. mit einem Doppelkreuz (#) auskommentiert oder gelöscht und neue Zeilen eingefügt werden, so dass die Datei ähnlich wie in Bild 22 aussieht.
   Nach „address" muss jeweils die gewünschte statische IP-Adresse eingetragen werden. Nach „gateway" wird die IP-Adresse des Routers im LAN eingetragen.

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
        ssid="meinWLAN"
        psk="meinKennwort"
        proto=RSN
        key_mgmt=WPA-PSK
        pairwise=TKIP
        auth_alg=OPEN
        id_str="elv"
}
```

Bild 21: Die Datei wpa_supplicant.conf

```
auto lo
iface lo inet loopback
iface eth0 inet static
address 192.168.2.2
netmask 255.255.255.0
gateway 192.168.2.1
allow-hotplug wlan0
iface wlan0 inet manual
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
iface elv inet static
address 192.168.2.2
netmask 255.255.255.0
gateway 192.168.2.1
iface default inet dhcp
```

Bild 22: Die Datei interfaces für die statische IP-Adresse

4. **Die Änderungen werden nach einem Neustart des Raspberry wirksam:**
   pi@raspberrypi ~ $ sudo shutdown –r now

Nun bekommt der Raspberry immer dieselbe IP-Adresse. Achtung: Natürlich muss nun die neue statische IP-Adresse (hier 192.168.2.2) für SSH-Verbindungen oder Browserzugriffe verwendet werden! Auch in Apps muss die IP-Adresse entsprechend aktualisiert werden!

Je nach eingesetzter Betriebssystem-Version auf dem Raspberry kann die Vorgehensweise abweichend sein. Das Einstellen einer statischen IP-Adresse im Router-Menü kann einfacher sein – sofern der eingesetzte Router dies unterstützt.

Zugriff auf den Raspberry über das Internet

Um aus dem Internet auf den Raspberry Pi zuzugreifen, sind drei Schritte erforderlich:

1. Die öffentliche IP-Adresse, unter der der Router aus dem Internet erreichbar ist, im Router-Menü oder unter [6] herausfinden
2. Port-Forwarding im Router einstellen:
   Im Router wird eine Portweiterleitung für Port 80 (http) zum Raspberry Pi eingeschaltet. Dadurch werden alle Anfragen aus dem Internet, die mit einem Browser über Port 80 auf die öffentliche IP-Adresse des Routers kommen, zum Raspberry Pi weitergeleitet.
3. Router unter seiner öffentlichen Adresse ansprechen:
   Nachdem man die öffentliche Adresse des eigenen Routers kennt und die Portweiterleitung für Port 80 im Router-Menü eingestellt hat, kann man im Browser im Internet (!) die öffentliche IP-Adresse des Routers eingeben. Die Anfrage wird an den Raspberry weitergeleitet und die entsprechende Seite vom Raspberry wird angezeigt! Wenn nur die öffentliche Router-IP-Adresse gemäß [6] eingegeben wird, dann wird die Standardseite index.html (siehe Bild 17) angezeigt. Wenn man hinter der öffentlichen Router-IP-Adresse noch eine bestimmte Seite anfordert – zum Beispiel php_out_in.php –, wird die entsprechende Seite angezeigt (Bild 20).

   Achtung: Je nach Router kann man im WLAN den Raspberry nur unter seiner lokalen IP-Adresse (zum Beispiel 192.168.2.2) erreichen und die öffentliche IP-Adresse des Routers nur von außerhalb des eigenen WLANs benutzen! (Stichwort NAT = Network Address Translation, das von einigen Routern unterstützt wird und von manchen Routern – z. B. Speedport 724 – nicht.)
   Achtung: Das Netz ist nun offener für Angreifer aus dem Internet. Man muss die Risiken abwägen und ggf. weitere Schutzmaßnahmen ergreifen.

Router trotz wechselnder IP-Adresse ansprechen (Dynamische-DNS-Weiterleitung)

In den meisten Fällen wird in regelmäßigen Intervallen (z. B. einmal in der Nacht) die Verbindung zwischen Router und Internet getrennt und wieder aufgebaut. Danach bekommt der Router eine neue öffentliche IP-Adresse zugewiesen, was wiederum zur Folge hat, dass man bei jedem Zugriff aus dem Internet neu ermitteln müsste, welche öffentliche IP-Adresse der Router hat. Da das natürlich nicht praktikabel ist, bieten sogenannte Dynamische-DNS-Dienste kostenlos oder gegen geringe Gebühren an, dass man eine feste URL des Dienstanbieters benutzen kann und dieser dann die Umsetzung in die aktuelle IP-Adresse des Routers durchführt. Ein derartiger Dienstanbieter ist selfhost.de. Im Router wird unter „Dynamisches DNS" der Anbieter eingetragen.

Der Router meldet sich bei selfhost.de, sobald er eine (neue) öffentliche IP-Adresse hat. Dadurch ist auf dem selfhost.de-Server jederzeit die aktuelle öffentliche IP-Adresse hinterlegt und man kann sich über selfhost an den Router wenden. Man bekommt eine Adresse wie hansmueller.selfhost.eu, an die man nur noch die gewünschte Seite auf dem Raspberry Pi anhängt: *hansmueller.selfhost.eu/php_out_in.php*

Nun kann man jederzeit unter einer festen URL, die man sich „bookmarken" kann, auf die auf dem Raspberry hinterlegten Seiten zugreifen.

Der DNS-Dienst von selfhost ist kostenfrei, aber man muss alle 30 Tage seine Identität per Klick bestätigen. Für einmalig 5 Euro kann man eine sogenannte Brief-Identifikation durchführen und kann dann den Dienst immer kostenfrei nutzen.

„Schöne" Internetadresse

Damit man nicht jedes Mal die komplette URL (xx.xxx.xxx.xxx/php_out_in.php oder hansmueller.selfhost.eu/php_out_in.php) eingeben muss, kann man sich über Provider wie Strato, 1&1, selfhost o. Ä. eine „schöne" URL geben lassen (zum Beispiel www.hans_mueller_home.de oder www.mein_zuhause.de).

Fazit

Wie in diesem Artikel gezeigt wurde, ist es mit einem Raspberry Pi und sehr geringen Vorkenntnissen leicht möglich, zunächst einmal innerhalb des WLANs die Pins am Raspberry Pi zu schalten bzw. abzufragen. Dadurch kann man mit angeschlossenen Relaisstufen oder HomeMatic/FS20-Sendern von innerhalb des WLANs Verbraucher schalten und Kontakte abfragen.

Mit ein bisschen weiterem Aufwand lassen sich die Möglichkeiten kostenfrei und schnell auch derart erweitern, dass die Steuerungen/Abfragen von irgendwo auf der Erde über das Internet erfolgen können. Von unterwegs die Heizung schon einmal einschalten oder abfragen, ob das Fenster geschlossen ist, ist dadurch sehr kostengünstig möglich. Eine Erweiterung des Hausautomationssystems um Temperaturabfragen, Servosteuerungen, Sonnenaufgangs-/Sonnenuntergangsberechnungen usw. ist Schritt für Schritt leicht möglich. ELV

Weitere Infos:

- Raspberry Pi: www.raspberrypi.org/about
- Raspbian: www.raspbian.org
- Deutsches Raspberry-Pi-Forum: www.forum-raspberrypi.de
- Englischsprachiges Raspberry-Pi-Forum: www.raspberrypi.org/forums

[1] IP-Adressen-Scanner: http://angryip.org
[2] SSH-Client für Windows: www.putty.org
[3] SSH-Client für iOS: http://simplessh.hirmer.me/
[4] SSH-Client für Android: https://juicessh.com
[5] SSH-Dateimanager WinSCP: https://winscp.net
[6] Öffentliche IP-Adresse herausfinden: www.wieistmeineip.de oder http://meineipadresse.de
[7] DynDNS-Dienst: www.selfhost.de
[8] Webhoster: Zum Beispiel strato.de oder 1und1.de

Empfohlene Produkte/Bauteile:	Best.-Nr.	Preis
Raspberry Pi 2 B, Starter-Set	CF-11 93 80	€ 89,95
Raspberry Pi 2 B, 1 GB	CF-11 93 85	€ 37,95
Raspberry Pi Zero	CF-12 26 20	€ 14,95
EDIMAX USB-WLAN-Adapter	CF-11 18 84	€ 8,95
NOOBS-Betriebssystem für Raspberry Pi 2, auf microSD-Karte	CF-12 01 50	€ 14,95
Netzteil (5 V/2 A)	CF-11 78 49	€ 6,25
Hama-Multikartenleser Basic	CF-12 17 52	€ 7,95
Funk-Tastatur mit Touchpad	CF-11 66 75	€ 36,95
Verbindungskabel-Set 25 cm	CF-11 78 53	€ 6,95
Verbindungskabel-Set 50 cm	CF-11 78 54	€ 7,95
Verbindungskabel für Raspberry Pi 2x 20-polig, 30 cm	CF-11 78 52	€ 5,95
Verbindungskabel für Raspberry Pi 2x 20-polig female, 25 cm	CF-11 97 41	€ 5,95
HomeMatic 8-Kanal-Sendemodul	CF-13 29 39	€ 19,95
FS20 S4M 2-/4-Kanal-Sendemodul	CF-06 68 17	€ 17,95
FS20 S8M 4-/8-Kanal-Sendemodul	CF-09 22 04	€ 24,95
Buch: Raspberry Pi: Mach's einfach	CF-11 84 57	€ 30,–
Buch: Raspberry Pi Unchained	CF-12 16 10	€ 34,95
Buch: Raspberry Pi	CF-11 57 70	€ 19,95
Buch: Hausautomation mit Raspberry Pi	CF-11 54 44	€ 30,–
Buch: Spannende Projekte mit dem Raspberry Pi	CF-11 57 71	€ 29,99
Buch: Sensoren am Raspberry Pi	CF-11 84 53	€ 30,–

Infos zu den Produkten/Bauteilen finden Sie im Web-Shop. Preisstellung Dezember 2015 – aktuelle Preise im Web-Shop