

Voraussetzungen

Man benötigt einen **Raspberry Pi 2** mit Netzteil, SD-Karte mit min. 4GB (mit installiertem Raspian Debian Wheezy Version 3.18 (Release Date 31.01.2015) und ein LAN-Kabel. Außerdem benötigen man einen WLAN-Dongle, z.B Edimax EW-7811 Un. Info's unter: http://www.edimax.com/edimax/download/download/data/edimax/global/download/for_home/wireless_adapters/wireless_adapters_n150/ew-7811un

Installation und Konfiguration

Nachdem der Raspberrry Pi 2 angeschlossen ist, verbinde mittels SSH (z.B „Putty“) und aktualisiere zuerst alle Pakete und den Kernel, damit die Unterstützung für möglichst viele Realtek basierten WLAN-Dongle (**Realtek RTL8188CUS**) vorhanden sind:

```
sudo apt-get update
sudo apt-get dist-upgrade
sudo rpi-update
sudo apt-get install hostapd dnsmasq
sudo apt-get install bridge-utils hostapd
sudo apt-get install isc-dhcp-server
wget http://www.daveconroy.com/wp3/wp-content/uploads/2013/07/hostapd.zip
unzip hostapd.zip
sudo mv /usr/sbin/hostapd /usr/sbin/hostapd.bak
sudo mv hostapd /usr/sbin/hostapd.edimax
sudo ln -sf /usr/sbin/hostapd.edimax /usr/sbin/hostapd
sudo chown root.root /usr/sbin/hostapd
sudo chmod 755 /usr/sbin/hostapd
```

```
sudo reboot
```

Prüfe ob der WLAN-Stick erkannt wird, indem Du dazu **lsusb** auf dem Terminal eingibst. Die Ausgabe der letzten Zeile sollte dann folgendermaßen (oder ähnlich) aussehen:

```
lsusb
```

```
pi@SmartAutomation ~ $ 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 7392:7811 Edimax Technology Co., Ltd EW-7811Un 802.11n Wireless Adapter [Realtek RTL8188CUS]
Bus 001 Device 005: ID 0781:5571 SanDisk Corp.
pi@SmartAutomation ~ $
```

Danach prüfe ob der Kernel-Treiber in meinem Fall für den EW-7811 Un geladen wurde:

Lsmod

```
pi@SmartAutomation ~ $ lsmod
Module                  Size  Used by
snd_bcm2835             18850  0
snd_pcm                 75388  1 snd_bcm2835
snd_seq                 53078  0
snd_seq_device          5628  1 snd_seq
snd_timer               17784  2 snd_pcm,snd_seq
snd                     51667  5 snd_bcm2835,snd_timer,snd_pcm,snd_seq,snd_seq_device
sg                      18207  0
8192cu                  528381 0
uio_pdrv_genirq         2958  0
uas                     13203  0
uio                      8119  1 uio_pdrv_genirq
pi@SmartAutomation ~ $
```

Als letzter Test untersuche ob der EW-7811 Un auch betriebsbereit ist

Iwconfig

```
pi@SmartAutomation ~ $ iwconfig
wlan0      IEEE 802.11bg  ESSID:"smart@automation"  Nickname:"<WIFI@REALTEK>"
Mode:Master  Frequency:2.437 GHz  Access Point: 74:DA:38:05:04:B1
Bit Rate:54 Mb/s   Sensitivity:0/0
Retry:off   RTS thr:off   Fragment thr:off
Power Management:off
Link Quality=2/100  Signal level=2/100  Noise level=0/100
Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
Tx excessive retries:0 Invalid misc:0  Missed beacon:0

lo         no wireless extensions.

eth0       no wireless extensions.

pi@SmartAutomation ~ $
```

Anschließend können die nötigen Pakete **hostapd** und **dnsmasq** installiert werden:

```
sudo apt-get install hostapd dnsmasq
```

Access Point einrichten

Zuerst muss dem WLAN-Stick eine feste IP-Adresse zugewiesen werden. Editiere dazu die Datei `/etc/network/interfaces`, sodass der Inhalt nun folgendermaßen aussieht:

```
sudo nano /etc/network/interfaces
```

```
auto lo

iface lo inet loopback
iface eth0 inet dhcp

allow-hotplug wlan0
iface wlan0 inet static
address 192.168.42.1
netmask 255.255.255.0

#iface wlan0 inet manual
#wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
#iface default inet dhcp
```

Wichtig ist, dass **wlan0** nur einmalig in dieser Datei genannt wird

Nun muss hostapd eingerichtet werden. Dazu editiere zunächst die 10. Zeile in der Datei `/etc/default/hostapd`:

```
sudo nano /etc/default/hostapd
```

```
# Defaults for hostapd initscript
#
# See /usr/share/doc/hostapd/README.Debian for information about alternative
# methods of managing hostapd.
#
# Uncomment and set DAEMON_CONF to the absolute path of a hostapd configuration
# file and hostapd will be started during system boot. An example configuration
# file can be found at /usr/share/doc/hostapd/examples/hostapd.conf.gz
#
DAEMON_CONF="/etc/hostapd/hostapd.conf"

# Additional daemon options to be appended to hostapd command:-
#   -d show more debug messages (-dd for even more)
#   -K include key data in debug messages
#   -t include timestamps in some debug messages
#
# Note that -B (daemon mode) and -P (pidfile) options are automatically
# configured by the init.d script and must not be added to DAEMON_OPTS.
#
#DAEMON_OPTS=""
```

Schließlich muss die dort angegebene Daemon-Konfiguration erstellt und mit den gewünschten Einstellungen gefüllt werden. Dazu kopiere einfach diese Konfiguration (mit eventuellen Anpassungen) in die `/etc/hostapd/hostapd.conf`-Datei.

sudo nano /etc/default/hostapd

```
# Genutztes Interface, muss bei Bedarf geändert werden (siehe "ifconfig"-Ausgabe)
interface=wlan0
# Realtek-Treiber, muss bei anderem Hersteller angepasst werden
driver=rtl871xdrv

# WLAN-Konfiguration
ssid=smart@automation
hw_mode=g
channel=6
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0

# WLAN-Sicherheit (Passwort anpassen!)
wpa=2
wpa_passphrase=passwort123456
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

Zum Abschluss muss ein Client noch eine IP-Adresse erhalten, wenn er sich beim Access Point anmeldet. Dazu editieren Sie die dnsmasq-Konfiguration unter `/etc/dnsmasq.conf` folgendermaßen:

```
# Configuration file for dnsmasq.

# ...

# If you want dnsmasq to listen for DHCP and DNS requests only on
# specified interfaces (and the loopback) give the name of the
# interface (eg eth0) here.
# Repeat the line for more than one interface.
interface=wlan0

# ...

# Uncomment this to enable the integrated DHCP server, you need
# to supply the range of addresses available for lease and optionally
# a lease time. If you have more than one network, you will need to
# repeat this for each network on which you want to supply DHCP
# service.
dhcp-range=192.168.42.2,192.168.42.100,255.255.255.0,12h

# ...
```

Somit vergibt nun der Raspberry Pi2 IP-Adressen im Bereich von 192.168.42.2 bis 192.168.42.100 für jeweils 12 Stunden. Starte nun die beiden Services neu und verbinde Dich mit dem neuen WLAN "smart@automation". Zwar haben wir noch keine Internet-Verbindung, jedoch könnte unter 192.168.42.1 ein Webserver erreicht werden.

`sudo service hostapd restart`

```
pi@SmartAutomation ~ $ sudo service hostapd restart
[ ok ] Stopping advanced IEEE 802.11 management: hostapd.
[....] Starting advanced IEEE 802.11 management: hostapdioctl[RTL_IOCTL_HOSTAPD]: Invalid argument
. ok
pi@SmartAutomation ~ $
```

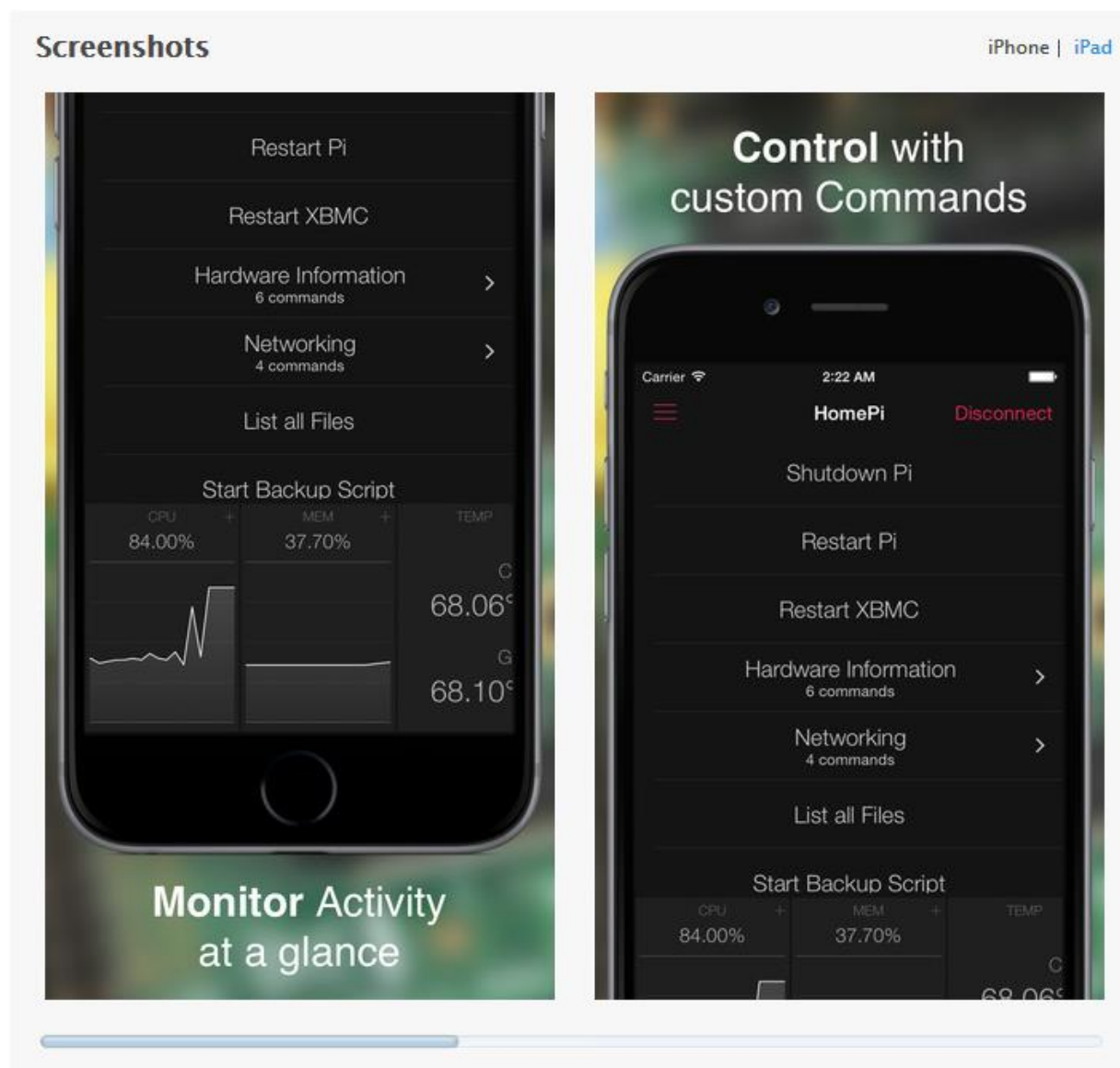
`sudo service dnsmasq restart`

```
pi@SmartAutomation ~ $ sudo service dnsmasq restart
[ ok ] Restarting DNS forwarder and DHCP server: dnsmasq.
pi@SmartAutomation ~ $
```

Das wars ... und viel Spass

Einfacher Raspberry Pi Monitor zum testen

<https://itunes.apple.com/DE/app/id882965594>



Option: Internet-Verbindung weiterleiten

Damit nun auch die WLAN-Clients eine Verbindung zum Internet erhalten, müssen ein paar weitere Einstellungen vorgenommen werden: Entferne zunächst die Raute vor dieser Zeile in der Datei `/etc/sysctl.conf`:

```
sudo nano /etc/sysctl.conf
```

```
# Uncomment the next line to enable packet forwarding for IPv4
#net.ipv4.ip_forward=1
```

Und lese die Datei anschließend erneut ein:

```
sudo sysctl -p
```

Nun weise iptables an, den Internet-Traffic von wlan0 auf eth0 weiterzuleiten, indem Sie folgendes Skript bei jedem Neustart des Raspberry Pis ausführen lassen. Legen Sie dazu eine Datei unter `/etc/network/if-up.d/accesspoint` an und kopieren Sie folgenden Inhalt dort hinein:

```
sudo nano /etc/network/if-up.d/accesspoint
```

```
#!/bin/sh
iptables --table nat --append POSTROUTING --out-interface eth0 -j MASQUERADE
iptables --append FORWARD --in-interface wlan0 -j ACCEPT
```

Schließlich diese Datei noch ausführbar machen und den Raspberry Pi2 neu starten:

```
sudo chmod +x /etc/network/if-up.d/accesspoint
sudo reboot
```