Erstes Semester Hardwareaufgaben

Farin Lippmann

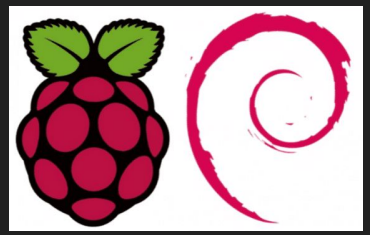
Raspberry Pi - Welches OS?

Ansprüche:

- Vielseitige Einsetzbarkeit
- Einfachheit / Ease of Use
- Verfügbarkeit
- Verbreitung / Community

Einige Kandidaten:

- Raspbian
- Pidora
- Ubuntu Mate
- Archlinux
- Windows 10 IoT Version



Entscheidung

Ubuntu Mate: + relativ weit verbreitet
- macht GPIO schwierig

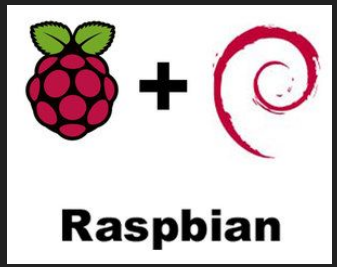
Archlinux: + großer Spielraum
+ totaler Allrounder
- überwältigend komplex

Pidora: + vielseitig nutzbar
+ relativ simpel
- etwas instabil
- weniger stark verbreitet

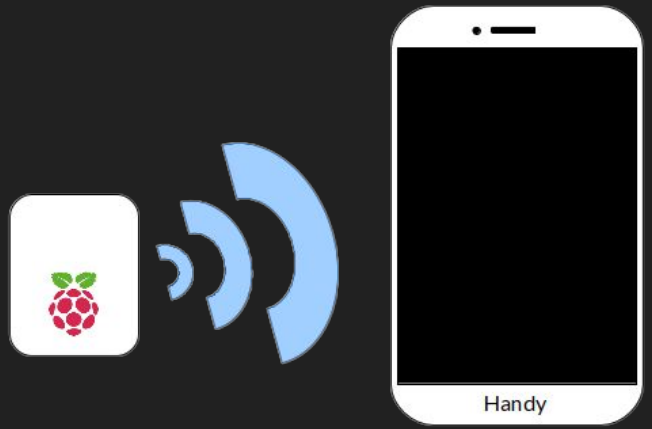
Windows 10 IoT + z.T. bekannte Oberfläche
- nur auf IoT zugeschnitten

Raspbian: + riesige Community
+ vielseitig genug für mich
+ simpel
+ viele Tutorials u. Guides
+ 100% für den Pi optimiert

-> Sieger:



Access Point



1. apt-get install dnsmasq hostapd

2. nano /etc/dhcpcd.conf

```
interface wlan0
static ip_address=192.168.4.2
nohook wpa_supplicant
```

3. nano /etc/dnsmasq.conf

```
interface=wlan0
dhcp-range:192.168.4.10,
           192.168.4.20,
           255.255.255.0,
           24h
```

Access Point

4. nano /etc/hostapd/hostapd.conf

```
interface=wlan0
driver=nl80211
ssid=NAME
hw_mode=g
channel=7
wmm_enabled=0
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=PASSWORT >7 Zeichen
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

5. nano /etc/default/hostapd

```
DAEMON_CONF:”/etc/hostapd/hostapd.conf”
```

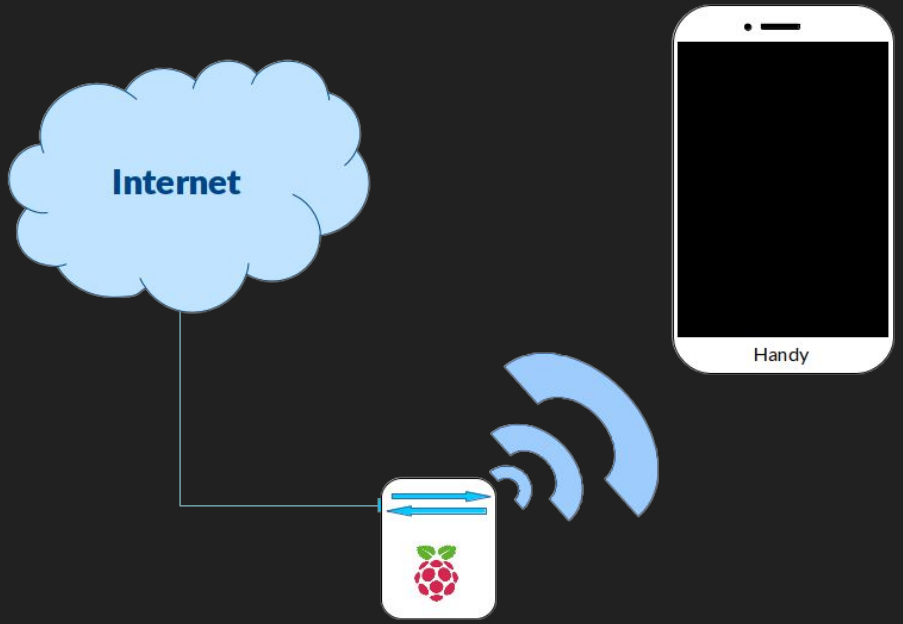
6. iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

7. sh -c “iptables-save > /etc/ iptables.ipv4.nat”

8. nano /etc/rc.local

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

Bridge



1. apt-get install hostapd bridge-utils
   systemctl stop hostapd

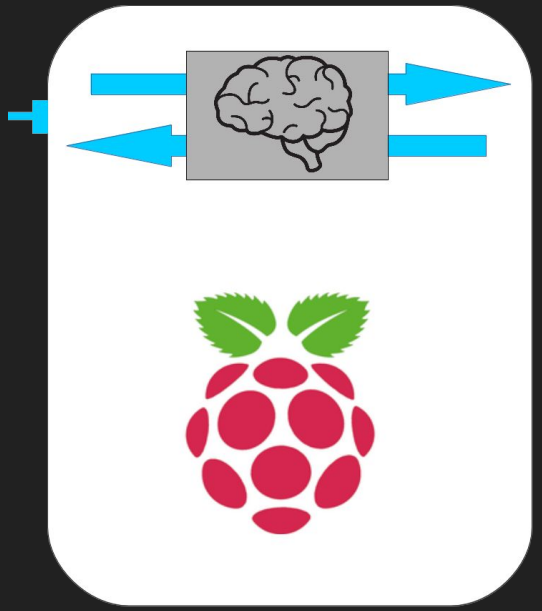
3. nano /etc/dhcpcd.conf
   denyinterfaces wlan0
   denyinterfaces eth0

4. brctl addbr br0
   brctl addif br0 eth0

5. nano /etc/network/interfaces
   auto br0
   iface br0 inet manual
   bridge_ports eth0 wlan0

6. nano /etc/hostapd/hostapd.conf
   bridge=br0
   #driver=nl80211

Router



1. Bridge entfernen

2. nano hostapd.conf
   ht_capab=[HT=40][SHORT-GI-40][DSSS_CCK-40]

2. nano /etc/dhcpcd.conf
   interface wlan0
   static ip_address=192.168.4.2
   static routers=192.168.4.2
   static domain_name_servers=192.168.13.1

3. nano /etc/dnsmasq.conf
   interface wlan0
   domain-needed
   bogus-priv
   dhcp-range:...

4. -A POSTROUTING -o eth0 -j MASQUERADE
   -A FORWARD -i eth0 -o wlan0 -m state --state
   RELATED,ESTABLISHED -j ACCEPT
   -A FORWARD -i wlan0 -o eth0 -j ACCEPT

Bridge vs. Router

- OSI: Data-Link Layer (2. Schicht)

- verbindet nur homogene Netzwerke

- leitet Daten nach MAC-Adr. weiter

- kann Netzwerke nicht unterscheiden

- transparente Bridges erstellen eine Weiterleitungstabelle, ansonsten werden Daten gebroadcastet

- nur nützlich wenn Subnetting nicht möglich ist

- OSI: Network Layer (3. Schicht)

- verbindet auch heterogene Netzwerke

- leitet Daten nach IP-Adressen weiter

- unterscheidet Netzwerke mit Netzmaske

- erstellen eine Routingtabelle

- kann Daten filtern

- meist effizienter, nützlicher als Bridge

Zeitliche Verfügbarkeit des AP

1. crontab -e

```
0 6 * * * sudo service dhcpcd start
0 17 * * * sudo service dhcpcd stop
@reboot sh /home/pi/Documents/script.sh
```

2. nano /home/pi/Documents/script.sh

```
#!/bin/sh
Stunde=$(date +%H)

if [ $Stunde -gt 16 ]
then
  sudo service dhcpcd stop
fi

if [ $Stunde -lt 7 ]
then
  sudo service dhcpcd stop
fi
```

3. chmod +x /home/pi/Documents/script.sh

IP-Adressen

- Adresse zum Transport von Daten in Computernetzen

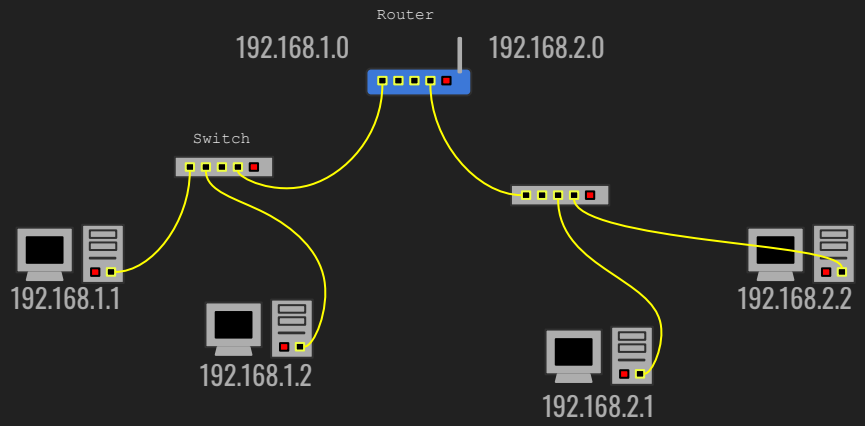
- OSI: Network Layer (3. Schicht)

- Aufbau (IPv4): xxx.xxx.xxx.xxx

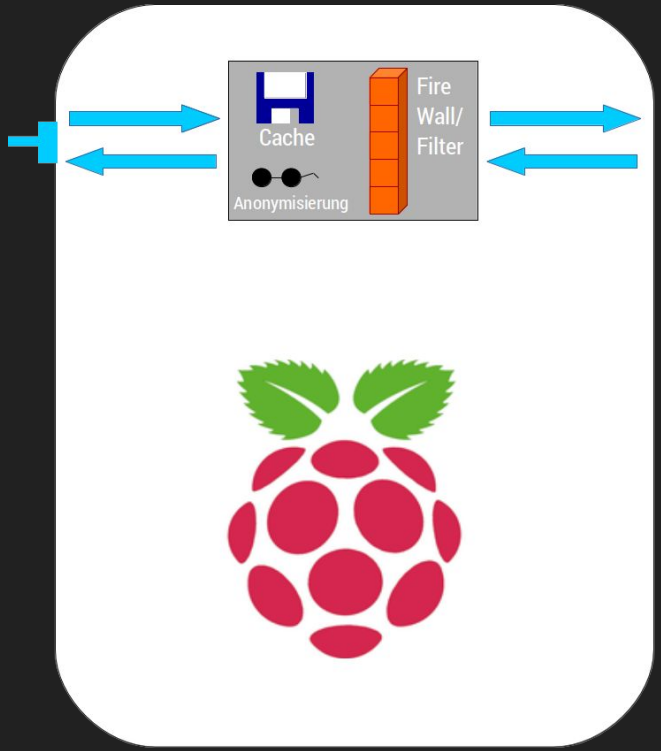
- jedes verschickte Datenpaket hat Quelladresse und Zieladresse

- Subnetting: Zerlegen von Netzwerken durch Teilung der IP-Adresse in Netzwerkteil und Hostteil

Beispiel mit Netmask 255.255.255.0:



Proxy



- OSI: Application Layer (7. Schicht)

- ein Server, der statt dem eigentlichen Ziel angefragt wird, und dann dieses selbst anfragt

- kann Performance verbessern, durch Caching von
  oft genutzten http Seiten

- Ziel sieht nur Anfrage des Proxy -> Anonymisiert

- Firewall / Filter kann unerwünschte / gefährliche Seiten blockieren -> Sicherheit

Proxy

1. apt-get install squid

2. apt-get install apache2-utils

3. sudo nano /etc/dhcpcd.conf
   interface eth0
   static ip_address=192.168.13.38
   static routers=192.168.13.1
   static domain_name_servers=192.168.13.1

4. cp /etc/squid/squid.conf
   /etc/squid/squidoriginal.conf.bak

5. nano /etc/squid/squid.conf
   ([STRG] + [W] zum Suchen)

6. (Alle lokalen durchlassen)
   acl localnet src 192.168.4.0/24
   http_access allow localnet

7. (Bestimmte URLs blocken)
   acl bad_url dstdomain .google.com
   http_access deny bad_url

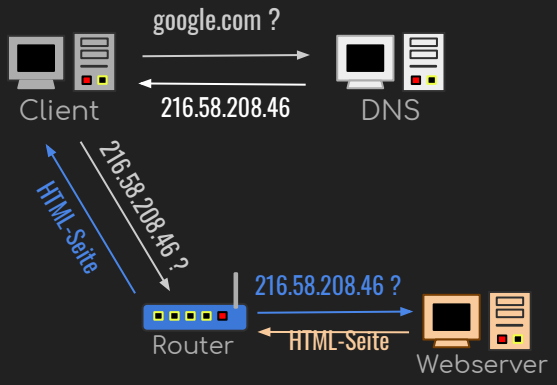
8. (Basic Authentifizierung)
   auth_param basic program
   /usr/lib/squid3/basic_ncsa_auth
   /etc/squid/passwords
   auth_param basic realm proxy acl
   authenticated proxy_auth REQUIRED
   http_access allow authenticated

9. (Im Browser oder Handy Proxy einstellen)
   Proxy: 192.168.13.38
   Port: 3128

DNS

- Server, der Liste mit Domainnamen und IP-Adressen verwaltet

- Beispiel:



- DNS Lookups funktionieren auch andersherum

1. `apt-get install dnsmasq`

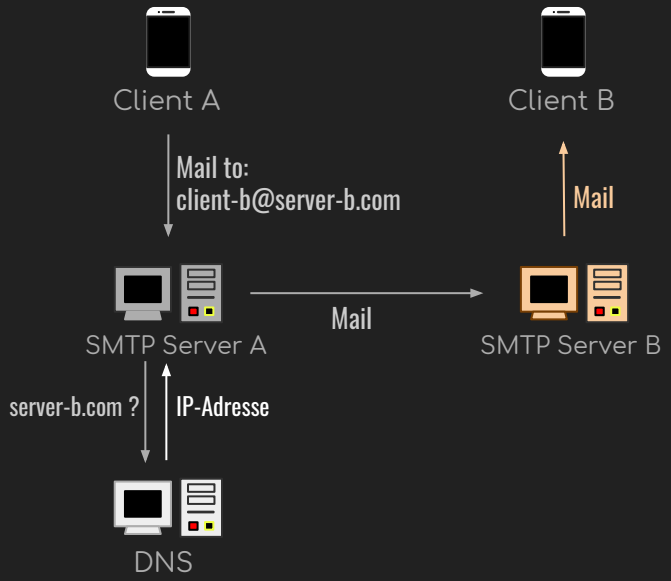
2. `nano /etc/dnsmasq.conf`
   `interface=wlan0`

3. `nano /etc/hosts`
   `(Bsp:) 8.8.8.8 dns.google.com`

4. `nano /etc/resolv.conf`
   `nameserver 127.0.0.1`

5. `service dnsmasq restart`

Mail Server



Ausführliches Tutorial mit Dovecot und IMAP:

https://samhobbs.co.uk/raspberry-pi-email-server

```
1. apt-get install postfix
     (wähle "Internet Site")
     (setze Domain Name (Bsp.: raspi.de)

2. cp main.cf main.cf.BAK
   cp master.cf master.cf.BAK

3. nano /etc/postfix/main.cf
     home_mailbox = Maildir/

4. apt-get install telnet

5. telnet localhost 25
   ehlo pi
   mail from: pi
   rcpt to: (z.B.) farin.lippmann@inverso.de
   data
   Testmail
   quit
```