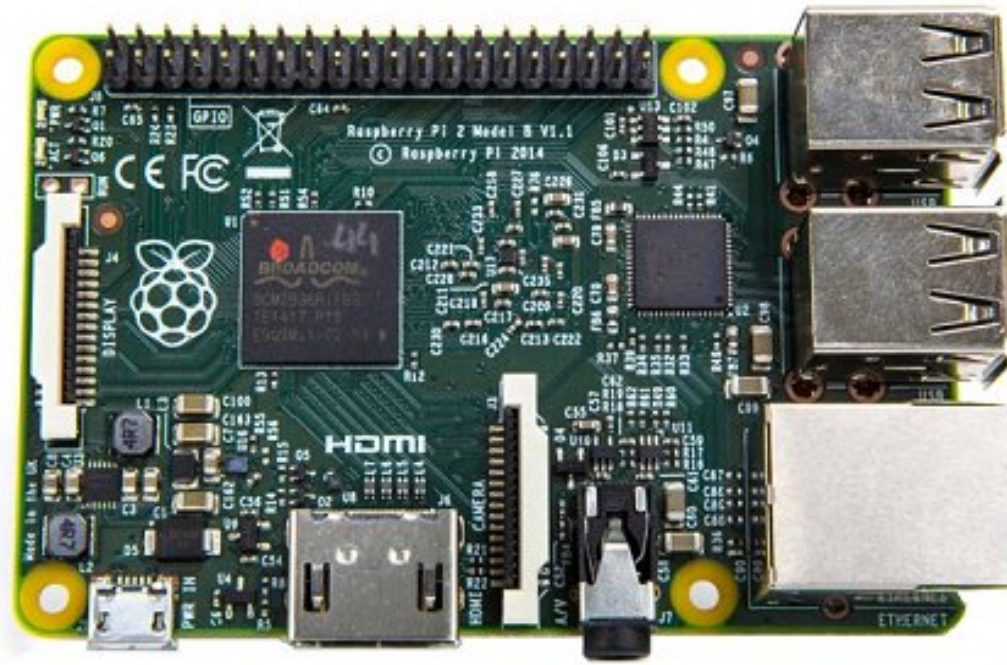


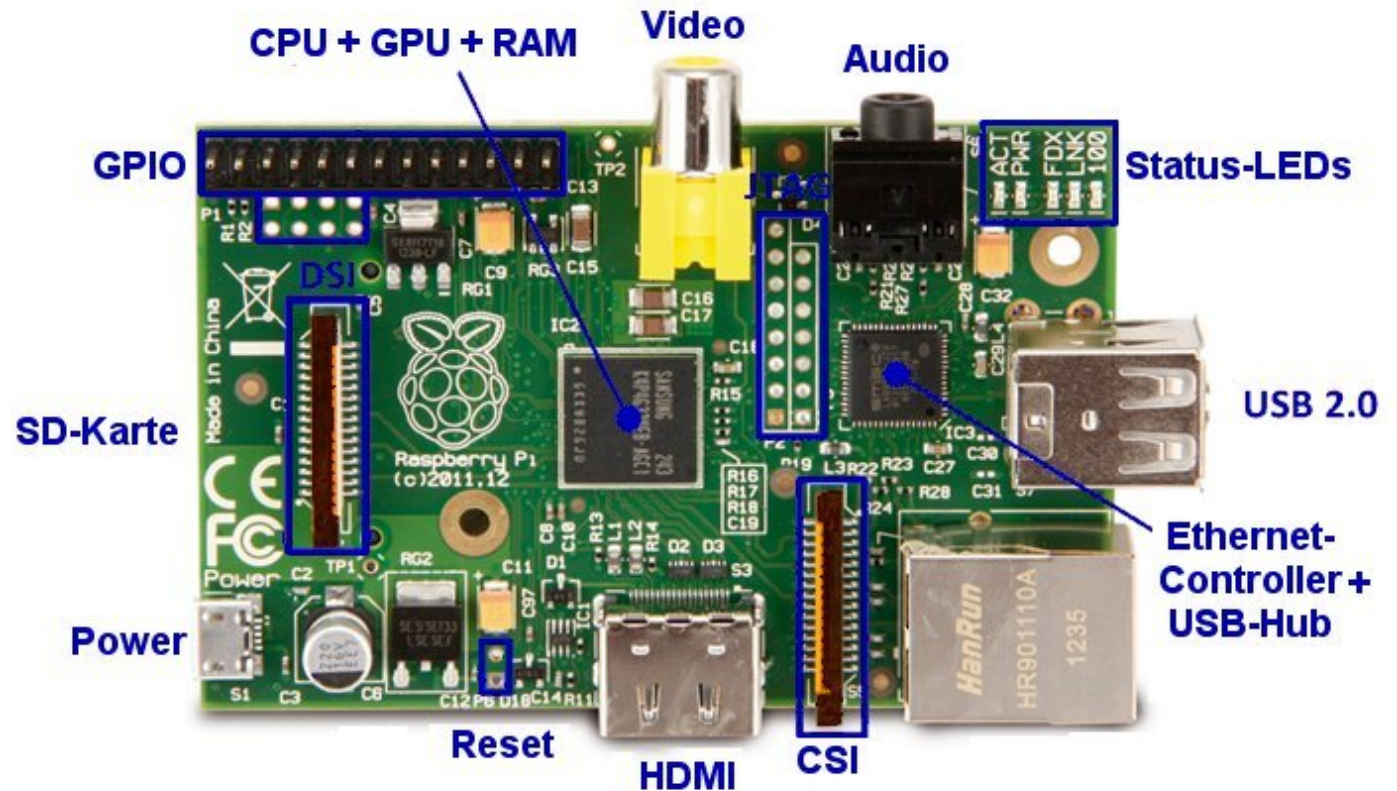
# ***Raspberry Pi Einführung***

**Prof. Dipl.-Ing. Klaus Knopper**



# Raspberry Pi Modell B

- BCM2835 SoC
- 512 MB RAM
- ARM11 (armv6) Prozessor
- VideoCore-IV-GPU
- 2x USB-2.0
- HDMI-Ausgang (Video + Audio)
- Composite-Video
- Audio-Ausgang (3,5mm Klinke)
- D-/MMC-Karten-Slot
- Ethernet-Port (10/100 MBit)
- 21 GPIO-Pins (z.B. UART, SPI und I2C)
- DSI (Display Serial Interface) über 15-poligen Flat Flex Connector
- Kamera-Interface, CSI (Camera Serial Interface) über 15-poligen Flat Flex Connector
- Altern. Stromversorgung per  $\mu$ -USB



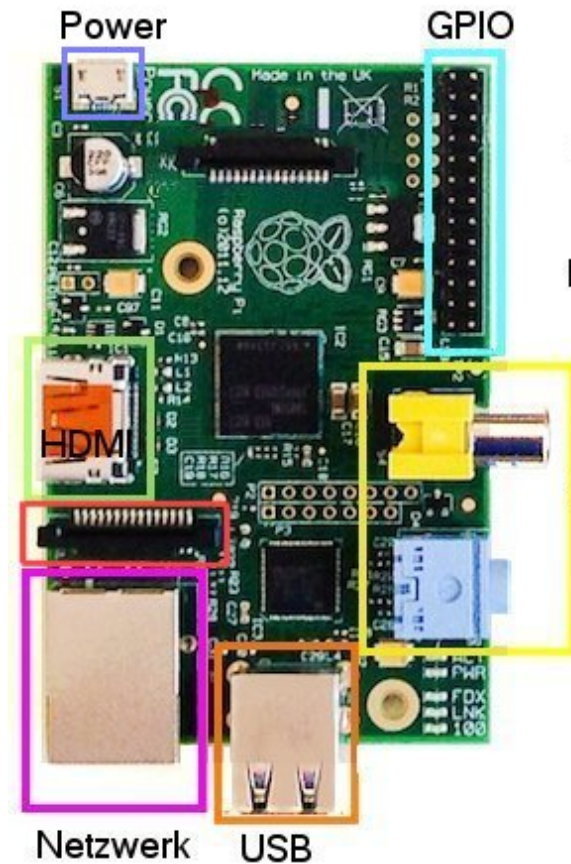
Quelle: [netzmafia.de](http://netzmafia.de)



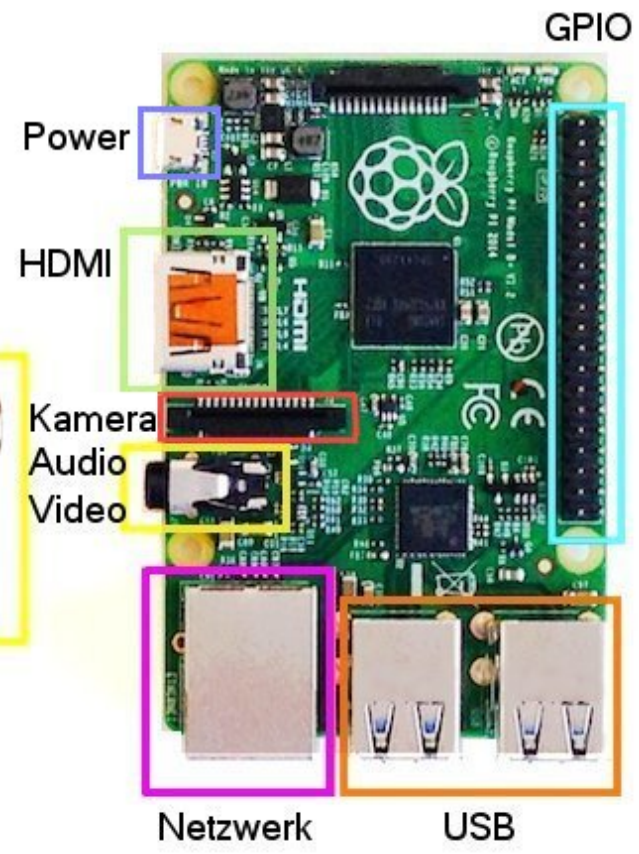
# Unterschiede Modell B und B+

- + zusätzliche Pins an der GPIO-Steckerleiste (I<sup>2</sup>C Bus)
- | Audio und Composite-Video in einem Stecker
- + 2 zusätzliche USB 2.0 Ports
- | Geringerer Stromverbrauch
- + stabiler beim Umstecken
- |  $\mu$ -SD Kartenslot
- | Qualität der PWN Audio-Ausgabe durch geänderten Stromlaufplan besser

Modell B



Modell B+



Quelle: [netzmafia.de](http://netzmafia.de)

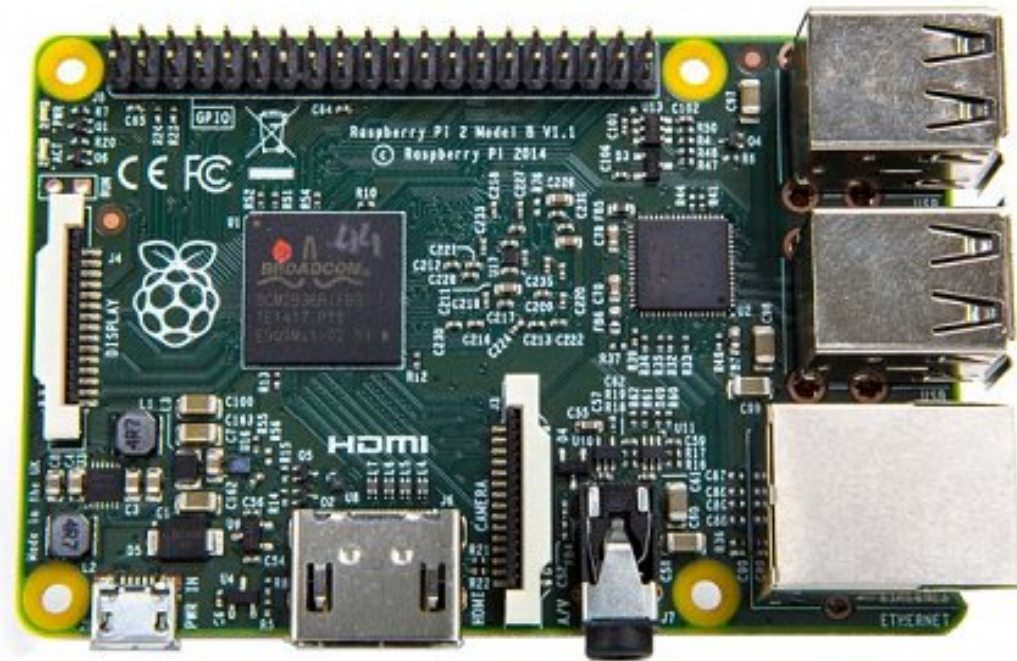




# Raspberry Pi 2

- + Vier Cortex-A7-Kerne, die mit 900 MHz getaktet
- + neues SoC BCM2836 (2835+1)
- + RAM 1GB

Sonst wie Raspi B+ inkl. Anschlussbelegung.



Quelle: [netzmafia.de](http://netzmafia.de)

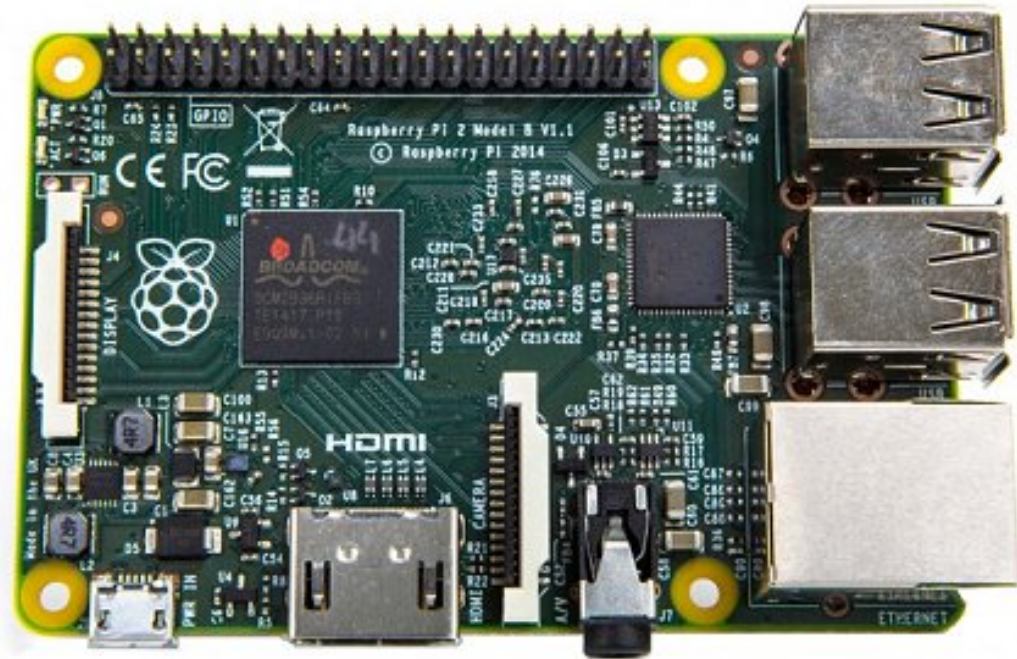


# Raspberry Pi 3

- + SoC BCM2837 von Broadcom
- + ARM Cortex-A53 Prozessor, **64-Bit-Quadcore** mit **1,2 GHz** („doppelt so schnell wie der Raspberry Pi 2“)
- + WLAN BCM43438 integriert (802.11b/g/n)+ unterstützt Bluetooth 4.1 (Classic und Low Energy).
- \* Kompatibilitäts-Problem: PL011-UART jetzt für Bluetooth zuständig, um serielle Konsole per festem Takt wieder nutzbar zu machen, Bootoption `enable_uart=1` notwendig.

Sonst wie Raspi 2 inkl. Anschlussbelegung.

Quelle: [netzmafia.de](http://netzmafia.de)

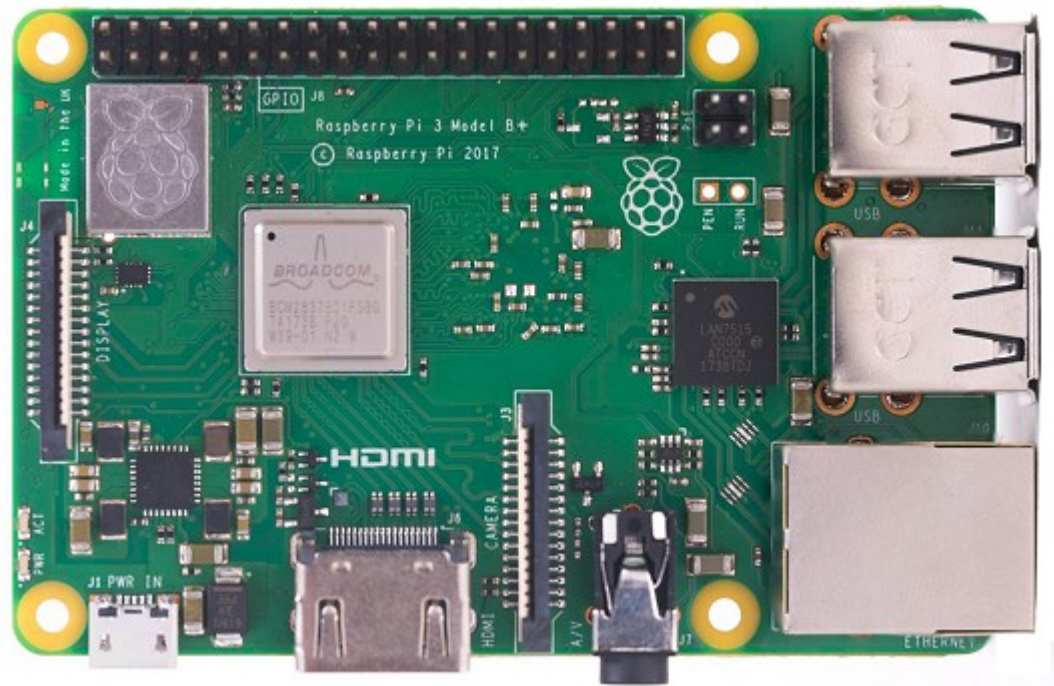


# Raspberry Pi 3 B+

- + SoC **BCM2837B0** von Broadcom
- + ARM Cortex-A53 Prozessor, **64-Bit-Quadcore** mit **1,4 GHz** („schneller als Raspberry Pi 3“)
- + WLAN **BCM43455** integriert (802.11b/g/n **inkl 5GHz Band bis 102 MBit/s**)
- + Bluetooth **4.2** (Classic und Low Energy).
- + Microchip **Gigabit LAN7515** mit **PoE**
- \* Weiterhin USB 2.0, aber mit max 300MBit/s statt 95.

Höhere Leistungsaufnahme (bis 7W), daher der Heatspreader auf der CPU.

Quelle: [netzmafia.de](http://netzmafia.de)



# Raspberry Pi Zero W

CPU: BCM2835 - ARM1176JZ-F v6 32Bit  
Single Core mit mathematischem  
Koprozessor (VPU) und DSP, 1 GHz  
GPU: Videocore IV, Dual Core, 128 KB L2-  
Cache, 250 MHz mit Unterstützung von  
OpenGL ES 2.0 und OpenVG 1.1  
RAM: 512 MB SDRAM @ 400 MHz  
Maße: 65 mm x 30 mm x 5 mm  
Anschlüsse: 1x USB 2.0, Micro-HDMI,  
Composite Video (unbestückt), microSD-  
Karte  
GPIO-Pins: 40 (unbestückt)  
WLAN+BT: Cypress CYW43438 Wireless-  
Chip für 802.11b/g/n-WLAN sowie  
Bluetooth 4.0



Quelle: [netzmafia.de](http://netzmafia.de)





# ***Betriebssysteme (Images)***

<http://www.raspberrypi.org/downloads/>





# *Installation / Konfiguration*

Grundsätzlich:

- ▶ Image auf SD-Karte entpacken  
(z.B. `dd if=raspbian.img of=/dev/sdb`, oder unter Windows mit **etcher**, mehrere Partitionen werden dabei angelegt)
- ▶ Booten (dabei passiert auto-Resizing auf physikalische Größe mit Neustart), nachher Konfiguration per (Text-) GUI `raspi-config`
- ▶ Neue Pakete installieren (Plugins bei XBMC oder Pakete bei Raspbian)



# ***Bootoptionen für Raspbian***

- ▶ Die Bootdateien liegen bei Raspbian auf der ersten Partition (Kernel, Initialsystem)
- ▶ Konfigurationsdatei: `config.txt`
  - ▶ `enable_uart=1` für Raspberry Pi 3
- ▶ Autostart von SSH: Leere Datei „ssh“ auf 1. Partition anlegen!



# *Zugang zum Raspberry Pi*

- ▶ „Traditionell“: Monitor (HDMI, Composite, Adapter), USB-Keyboard+Maus
  - ▶ Zeitgemäß:
    - ▶ Login per SSH verschlüsselt via LAN oder WLAN
    - ▶ Login per Remote-Desktop (VNC, rdesktop)
- Problem: Noch keine IP-Adresse konfiguriert oder bekannt!
- ▶ Professionell (Embedded Programmierung): Serielle Konsole über GPIO / (D)UART (direktes „Andocken“ ans System)



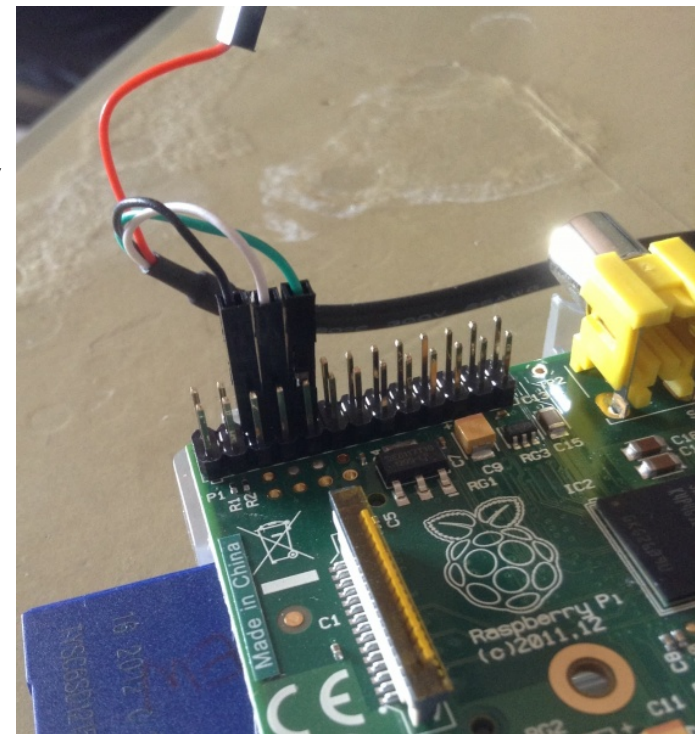
# Serielle Konsole (Hardware)

## USB → GPIO/UART-Kabel

GROUND = Schwarz = 3. Pin  
TxD = Weiß = 4. Pin  
RxD = Grün = 5. Pin

(angegebene Pinbelegung ist hier „von oben links“ gezählt, das ist aber nicht die offizielle Zählweise lt. Datenblatt!)

5V = Rot = Pin 1 *kann* zur Stromversorgung angeschlossen werden, *sollte* *muss* aber weggelassen werden, falls schon anderweitig Stromversorgung (z.B. Netzteil) angeschlossen ist!



(Quelle: [http://elinux.org/RPi\\_Serial\\_Connection](http://elinux.org/RPi_Serial_Connection))





# ***Serielle Konsole (Software)***

- ▶ Unter Raspbian ist die serielle Konsole standardmäßig aktiv.
- ▶ Auf der anderen Seite des Kabels muss ein Terminalprogramm gestartet werden, z.B. **minicom**, **screen** oder **putty**  
Einstellung: Device /dev/ttyUSB0 Speed 115200  
  
`screen /dev/ttyUSB0 115200`
- ▶ Einloggen mit Login: pi Passwort: raspberry



# Übung:

## *Raspi in Betrieb nehmen*

- ▶ Achtung: Während der  $\mu$ -SD Karten-Leser beim Raspberry Pi 2 „klickt“, wird ab Raspberry Pi 3 die Karte nur vorsichtig und ohne Druck bis zum Widerstand eingeschoben. Kein Einrasten!
- ▶ Anmelden: pi raspberry
- ▶ WLAN konfigurieren: pi raspberry
- ▶ Sprach- und Desktop-Einstellungen
- ▶ Software nachinstallieren
- ▶ Software Tests...



# ***Tipp für Zugang per serielllem Kabel: Netzwerk-Einstellungen in der Konsole***

- ▶ Am einfachsten: Editieren von `/etc/network/interfaces`

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

```
auto wlan0  
iface wlan0 inet dhcp  
    wpa-ssid pi  
    wpa-psk password
```

- ▶ Nach dem Speichern:  

```
sudo ifdown --force wlan0  
sudo ifup wlan0
```



# ***SSH-Server aktivieren***

- ▶ Am einfachsten: Anlegen einer Datei namens „ssh“ (leer) auf der 1. Partition der SD-Karte → Sorgt dafür, dass der SSH-Server automatisch gestartet wird!





# Netzwerk-Konfiguration

- ▶ Während sich `/etc/network/interfaces` um die Konnektierung kümmert, wird der Nameserver (vom DHCP-Client) in `/etc/resolv.conf` eingetragen:

```
nameserver 10.0.0.1
```

- ▶ `ifconfig device [ip-adresse] [netmask netzmaske]`
- ▶ `route [-n] ...`
- ▶ Sobald das Netz erreichbar ist, kann mit der Installation neuer Pakete begonnen werden, z.B. vnc oder rdp Server für graphischen Remote-Zugriff.



# Software-Auswahl und Konfiguration anpassen (Kommandozeilen-Variante)

- ▶ Raspbian: Menügeführte Konfiguration mit `sudo raspi-config` (Übung)
- ▶ Softwarepakete aus Debian/Raspian-Repository nachinstallieren (auch Ubuntu):

Kommando	Wirkung
<code>apt update</code>	SW-Datenbank aktualisieren
<code>apt upgrade</code>	(VORSICHT!) Komplettes System aktualisieren
<code>apt search</code> <b>Stichwort</b>	Software suchen
<code>apt show</code> <b>paketname</b>	Details anzeigen
<code>apt install</code> ( <i>oder</i> <code>remove</code> ) <b>paketname</b>	Softwarepaket installieren/aktualisieren oder entfernen



# ***Kommandozeile vs. Graphische Oberfläche***

- ▶ Sofern das Gerät schon im Netz ist, kann mit dem Kommando „ifconfig“ die IP-Adresse herausgefunden werden!
- ▶ `ssh raspip-ip-adresse` startet zunächst nach erfolgreichem Login eine Shell für den Benutzer „pi“.
- ▶ SSH-Client unter Linux zum Raspberry Pi erlaubt auch den Direktstart graphischer Programme, die auf die eigene Desktop-Oberfläche „getunnelt“ werden.
- ▶ Desktop-Projektion bzw. Starten eines virtuellen Desktop per `tightvncserver` ist möglich (VNC). Für (Windows-) RDP-Clients kann entsprechend ein RDP-Server installiert werden.  
`sudo apt install tightvncserver xrdp`



# ***Viel Spaß beim „Basteln“!***

