

Windischgarsten

2020.9

**Raspberry Pi meets Industrie 4.0 & IoT
Agile Softwareentwicklung & KI**



**Prof. Dipl.-Ing. Klaus Knopper
<knoppix@knopper.net>**



Netzwerk Schulungsraum

Wireless LAN SSID: **seminar.lan (Passwort s. Tafel)**

Netzwerk LAN+WLAN: **10.0.0.X** (per DHCP)

Netzmaske: **255.255.255.0**

Gateway+DNS: **10.0.0.1**

WWW (Referent): **http://10.0.0.10**

(noch nicht fertig) SAMBA/Windows:

NAS Storage: **net use K: \\10.0.0.10\www
\\10.0.0.2**



Organisatorisches

- ▶ Beginn Mo 9:30 – 18:00, Di 8:30-18:00 ggf. nachmittags Exkursion, Mi 8:30 - ca 16:00 Uhr (nachmittags „freie Projekte“)
- ▶ Mittagspause ca. 13:00 und Abendessen nach Bedarf (ggf. solange das Essen noch warm ist bis 18:30 Uhr), nach dem Abendessen weiterhin noch 2+ Stunden „freie Projekte“ möglich
- ▶ Diesmal: Parallel laufendes Projekt mit Agiler Software-Entwicklung (erst kurz Theorie zu SCRUM, danach Mini-SCRUM Projekt)
- ▶ Offenes Programm mit hohem praktischen Anteil



Betriebssysteme (Images)

- ▶ <http://www.raspberrypi.org/downloads/>
- ▶ (und auf Webserver im Ordner „Raspimages“)



Programm

- ▶ Alte und neue Pi-Modelle
- ▶ SCRUM in 5 Minuten / Projekt
- ▶ Kurzeinführung Linux allgemein
- ▶ Anbindung Pi ↔ Arduino, z.B. Octoprint, und 3D-Druck-Projekte rund um Raspberry Pi
- ▶ Heimautomatisierung (cloudbusters.knopper.net ff.)
- ▶ KI-Programmierung mit TensorFlow auf Pi

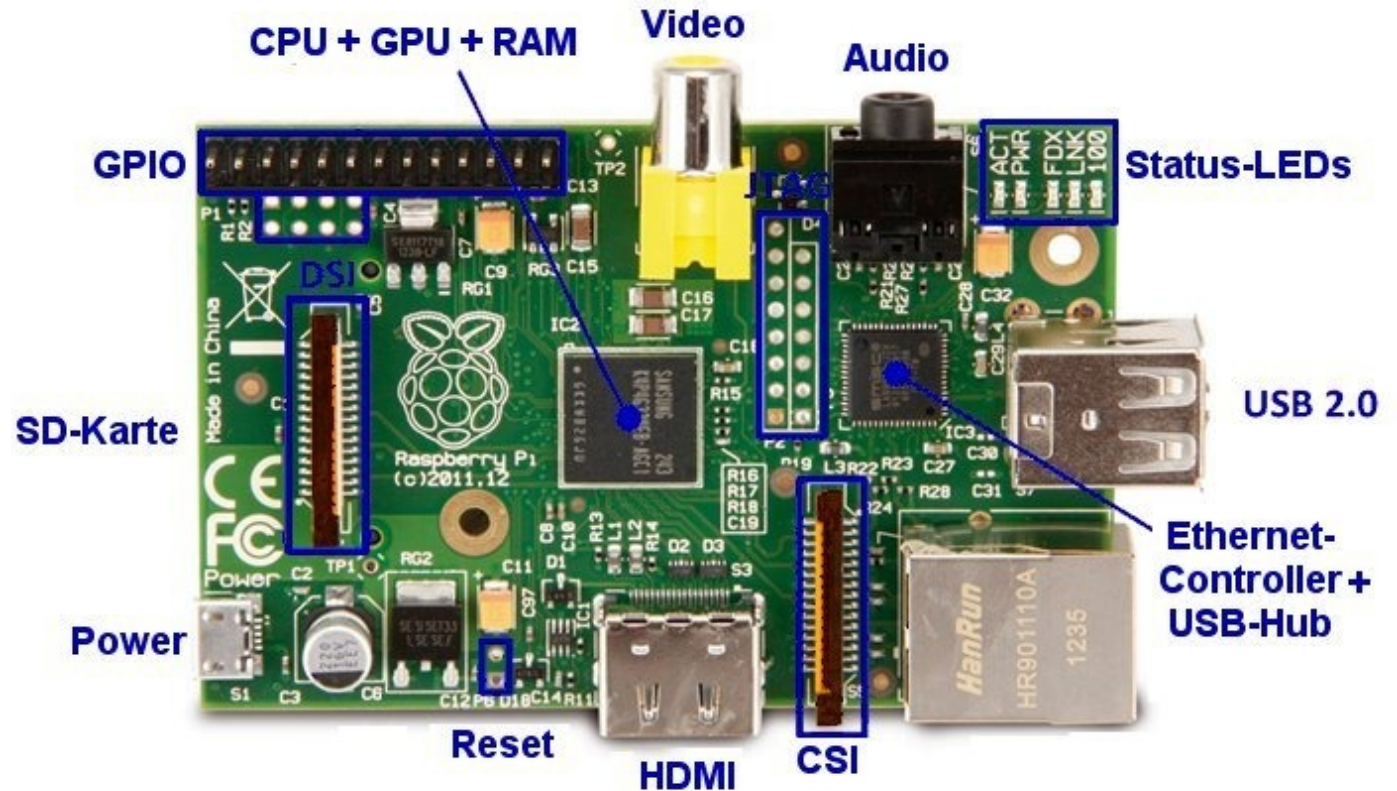


Alle Raspberry Pi Modelle (Einführung)



Raspberry Pi Modell B (alt)

- 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 μ -USB

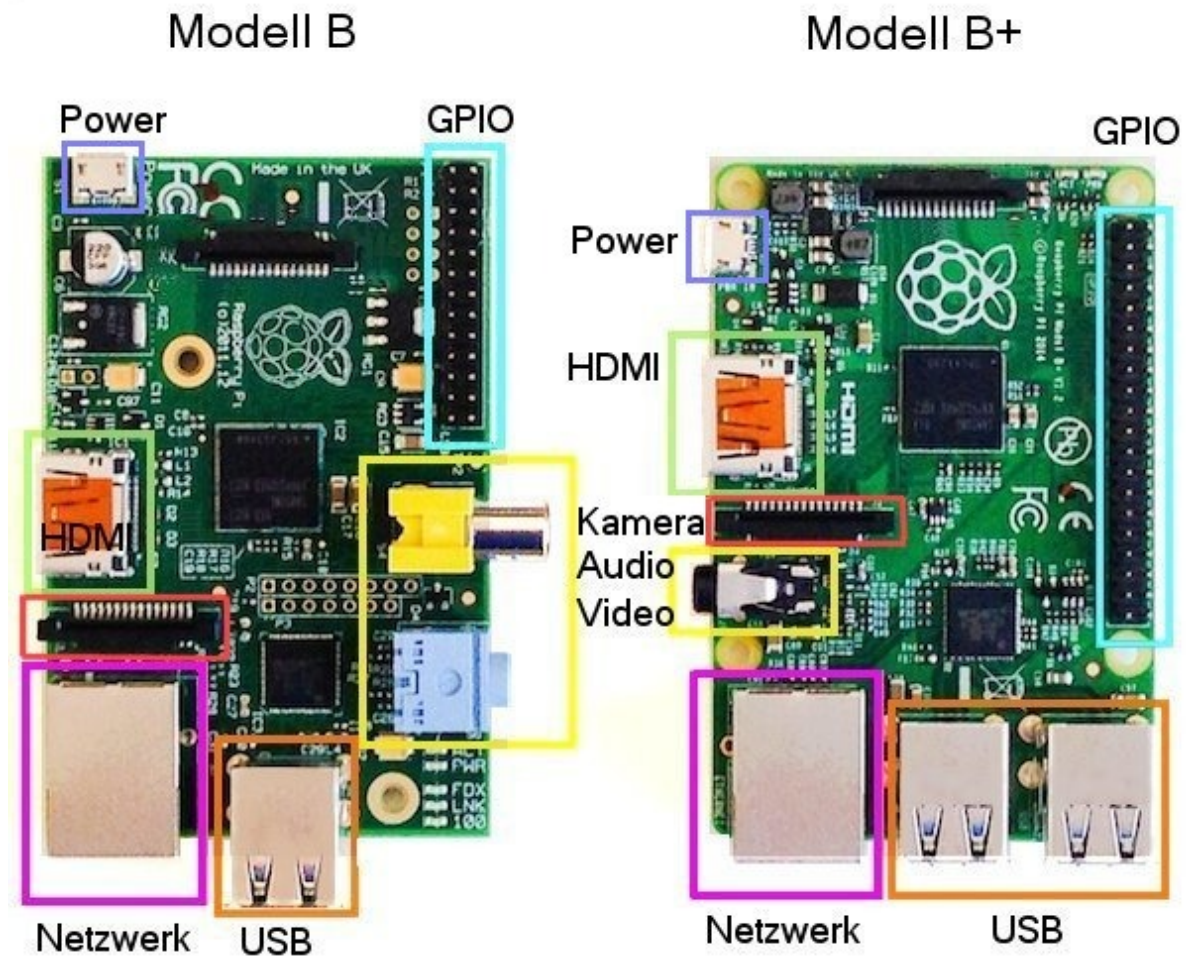


Quelle: netzmafia.de



Unterschiede Modell B und B+ (alt)

- + zusätzliche Pins an der GPIO-Steckerleiste (I²C Bus)
- | Audio und Composite-Video in einem Stecker
- + 2 zusätzliche USB 2.0 Ports
- | Geringerer Stromverbrauch
- + stabiler beim Umstecken
- | μ -SD Kartenslot
- | Qualität der PWN Audio-Ausgabe durch geänderten Stromlaufplan besser



Quelle: netzmafia.de



Raspberry Pi 2 (alt)

- + 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

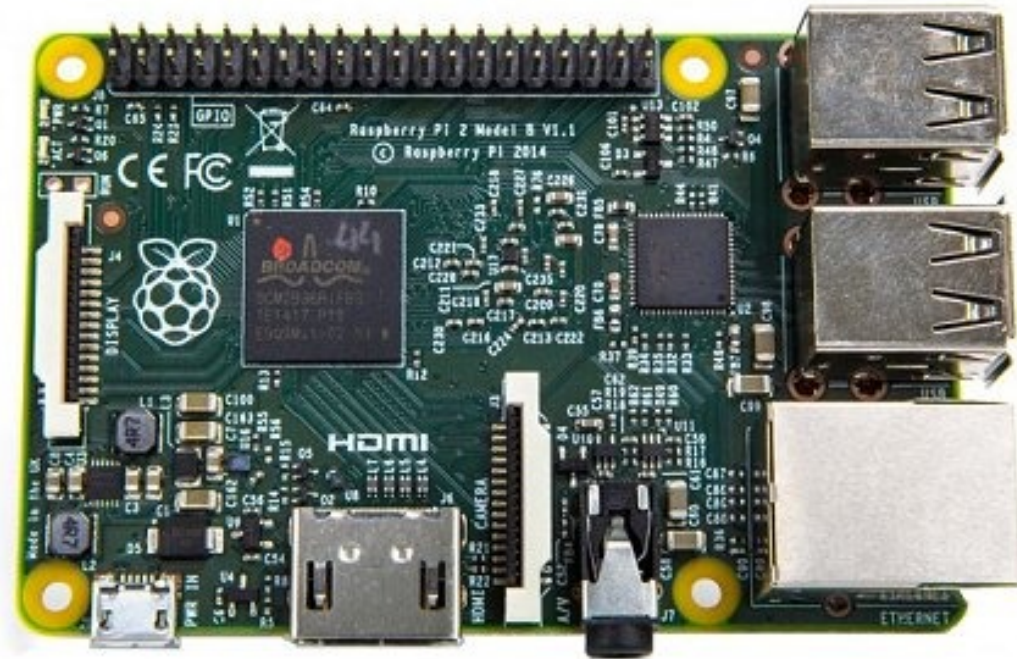


Raspberry Pi 3 (nicht so alt)

- + 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.

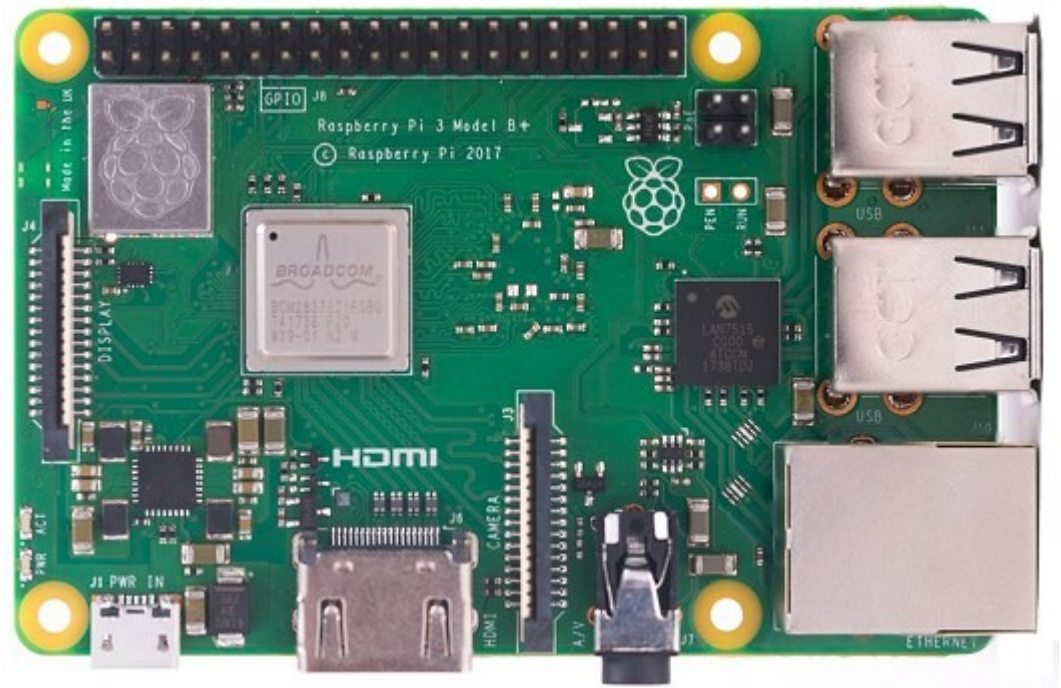
Quelle: netzmafia.de



Raspberry Pi 3B+

- + SoC **BCM2837B0** von Broadcom
- + ARM Cortex-A53 Prozessor, 64-Bit-Quadcore mit **1,4 GHz**
- + WLAN **BCM43455** integriert (802.11b/g/n **Dualband 5GHz**) + unterstützt Bluetooth **4.2** (Classic und Low Energy).
- + LAN7515 **Gigabit** LAN (gedrosselt durch USB 2.0 Bus), **PoE-fähig mit Step-Down-Wandler**
- * Metallgehäuse für Prozessor durch höhere Leistungsaufnahme (bis 7 Watt)

Sonst wie Raspi 3 inkl. Anschlussbelegung.



Quelle: 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



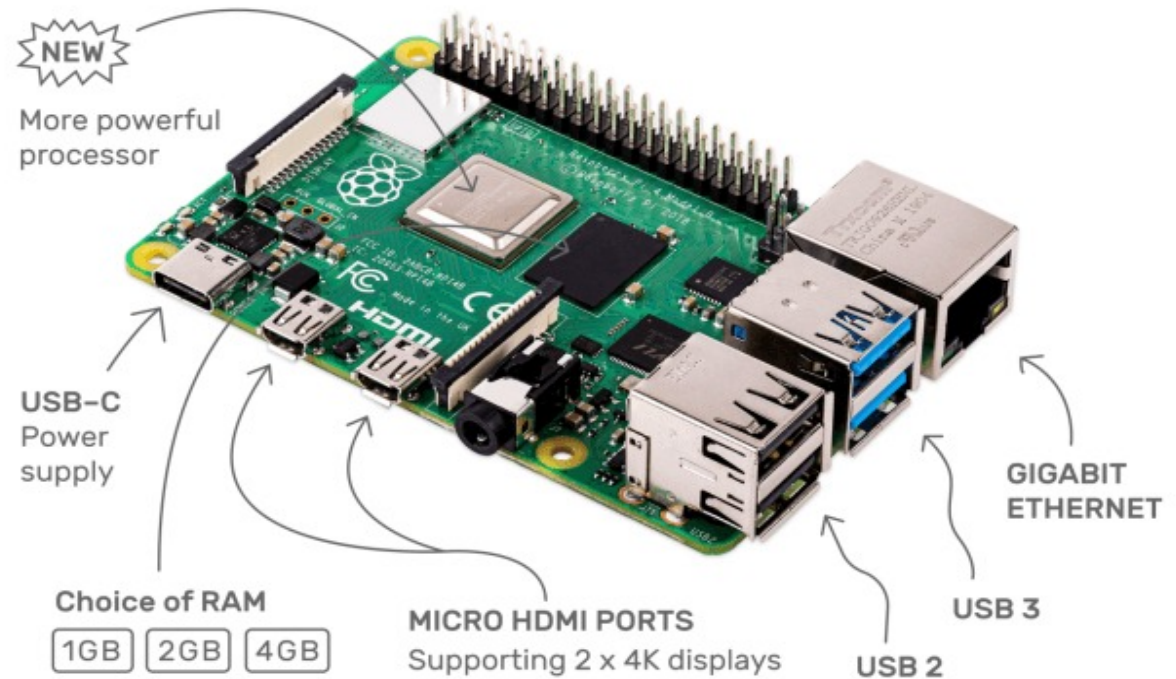
Mit einer **zusätzlichen** RTC- Schaltung
kann „Deep Sleep“ per Halt / Reset
erzeugt werden (Raspi wacht in
regelmäßigen Intervallen auf) → Läuft mit
Akku sehr lange

Quelle: netzmafia.de



Raspberry Pi 4 (neustes Modell 2020)

- + SoC **BCM2711** von Broadcom
- + ARM Cortex-A72 (ARM v8) Prozessor, 64-Bit-Quadcore mit **1,5 GHz**
- + wahlweise **1GB, 2GB, 4GB** und (ab Mai 2020) **8GB** SDRAM
- + WLAN **BCM43455** integriert (802.11ac **Dualband 5GHz**) + unterstützt Bluetooth **4.2** (Classic und **Low Energy**).
- + **Gigabit LAN**
- + 2 USB2 und 2 **USB3-Ports**
- + **2x Micro HDMI**, H.265 (4kp60 decode), H264 (1080p60 decode, 1080p30 encode)
- Höhere Leistungsaufnahme (**3A** bei 5V!)



Die ersten Modelle hatten einen [USB-C Detektionsfehler](#) und wurde daher von „intelligenten“ Ladegeräten nicht erkannt (inzwischen → behoben).

Achtung:

- Höhere Leistung/Hitzeentwicklung, möglichst Kühlkörper aufkleben
- Strombedarf höher als beim 3 (15W Netzteil empfohlen) (5V per TTL-Adapter über GPIO-Leiste reicht nicht mehr)

Quelle: [RaspberryPi.org](https://www.raspberrypi.org)



Installation / Konfiguration

Grundsätzlich:

- ▶ Image auf SD-Karte entpacken
(z.B. unter Linux:
`dd if=raspbian.img of=/dev/sdb bs=1M`
mehrere Partitionen werden dabei angelegt)
Windows: `etcher.io`, oder den neuen Imager von
`raspberrypi.org`
- ▶ Booten (dabei 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 Raspbian/Raspberry Pi OS

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



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) → S. Handout „Zugang über serielle Schnittstelle“



Serielle Konsole (Hardware)

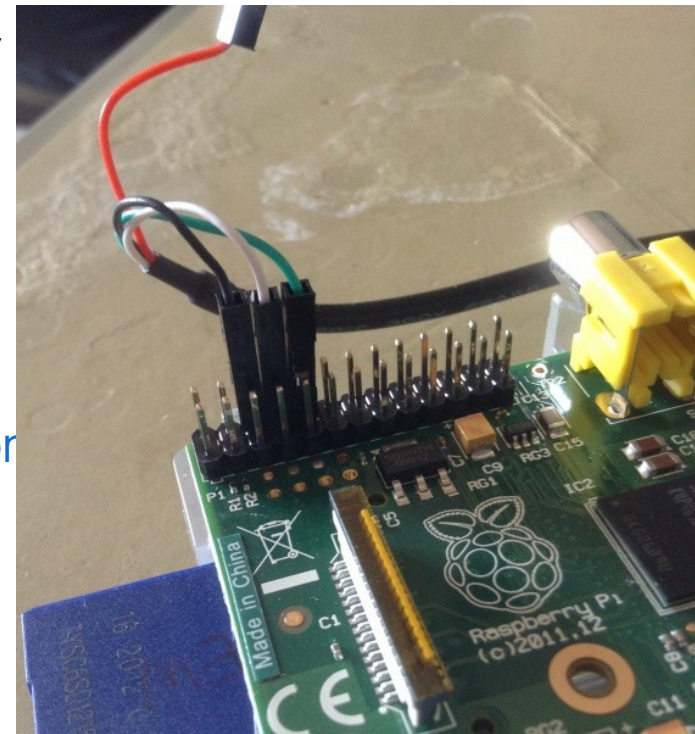
USB → GPIO/UART-Kabel

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)



Serielle Konsole (Software)

- ▶ Unter Raspbian ist die serielle Konsole standardmäßig per `/etc/inittab`-Eintrag aktiv:

```
#Spawn a getty on Raspberry Pi serial line  
T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100
```

- ▶ 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`



Serielle Konsole (putty)

Category: Basic options for your PuTTY session

Specify the destination you want to connect to

Serial line: Speed:

Connection type:
 Raw Telnet Rlogin SSH Serial

Load, save or delete a stored session

Saved Sessions

Session Name
pi
Default Settings
pi

Close window on exit:
 Always Never Only on clean exit

```
raspberrypi3 login:
Poky (Yocto Project Reference Distro) 3.1.2 raspberrypi3 /dev/ttyS0

raspberrypi3 login:
Poky (Yocto Project Reference Distro) 3.1.2 raspberrypi3 /dev/ttyS0

raspberrypi3 login:
Poky (Yocto Project Reference Distro) 3.1.2 raspberrypi3 /dev/ttyS0

raspberrypi3 login:
Poky (Yocto Project Reference Distro) 3.1.2 raspberrypi3 /dev/ttyS0

raspberrypi3 login: root
root@raspberrypi3:~# free
total      used         free      shared  buff/cache   available
Mem:      946608      32016      889108        244       25484      895360
Swap:          0           0           0

root@raspberrypi3:~# df
Filesystem      1K-blocks      Used Available Use% Mounted on
/dev/root        225973      173097    40998   81% /
devtmpfs         341720         0    341720   0% /dev
tmpfs            473304         180    473124   0% /run
tmpfs            473304         64    473240   0% /var/volatile

root@raspberrypi3:~# █
```



Netzwerk-Einstellungen in der Konsole

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

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

```
auto wlan0
iface wlan0 inet dhcp
    wpa-proto WPA2 WPA
    wpa-key-mgmt WPA-PSK
    wpa-ssid seminar.lan
    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

- ▶ Raspbian: Menügeführte Konfiguration mit `sudo raspi-config` (s. Übung 1)
- ▶ Softwarepakete aus Debian/Raspian-Repository nachinstallieren:

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



Kommandozeile vs. Graphische Oberfläche

- ▶ `ssh raspi-ip-adresse` startet zunächst nach erfolgreichem Login eine Shell für den Benutzer „pi“.
- ▶ SSH unter Linux 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 RDP-Clients kann entsprechend ein RDP-Server installiert werden:
`sudo apt-get install xrdp`



Workshops

...zu den Wunschthemen:

- ▶ Handouts
- ▶ Übungen
- ▶ Beispiele (Skripte, Webseiten)

Materialien werden nach dem Kurs im Web zur Verfügung gestellt.

<http://knopper.net/Windischgarsten/Kursdatum>



Viel Spaß beim „Basteln“!

