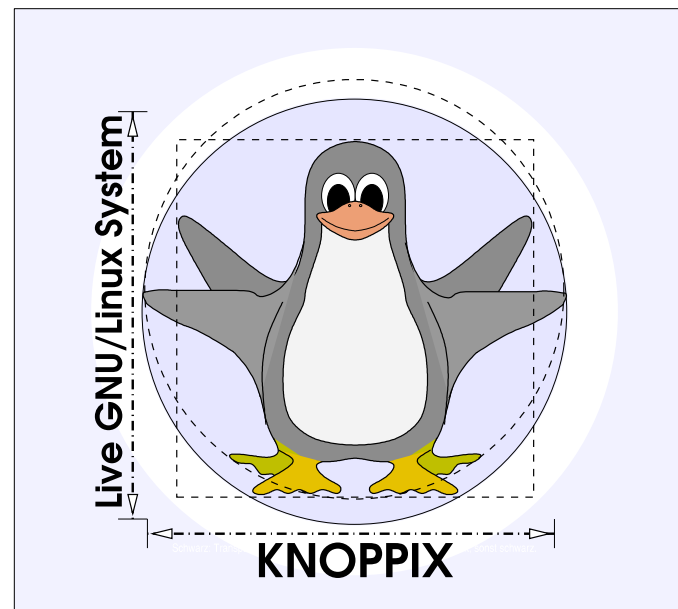


Windischgarsten **2017**

Raspberry Pi, Mobile Networking & More



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



Netzwerk Schulungsraum

Wireless LAN SSID: **pi2** (für RasPis, NICHT pi)

Wireless LAN Passwort (PSK): **sperlhof**

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

Netzmaske: **255.255.255.0**

Router: **10.0.0.1**

DNS: **10.0.0.1**

Referent: **10.0.0.10**



Organisatorisches

- ▶ Beginn Do 9:30 – 12:00, nachmittags Exkursion „fliegender embedded Linux-Computer mit First-Person-View (FPV)“, Fr+Sa 9:00, Ende Freitag 13:00 Uhr (nachmittags „freie Projekte“)
- ▶ Mittagspause und Abendessen nach Bedarf, vor dem Abendessen noch 2+ Stunden „freie Projekte“
- ▶ Sowohl „Anwendung“ als auch „Systemprogrammierung“ (teilweise parallel möglich, s. [Übungen](#))
- ▶ Offenes Programm mit hohem praktischen Anteil, Teilnehmerwünsche:



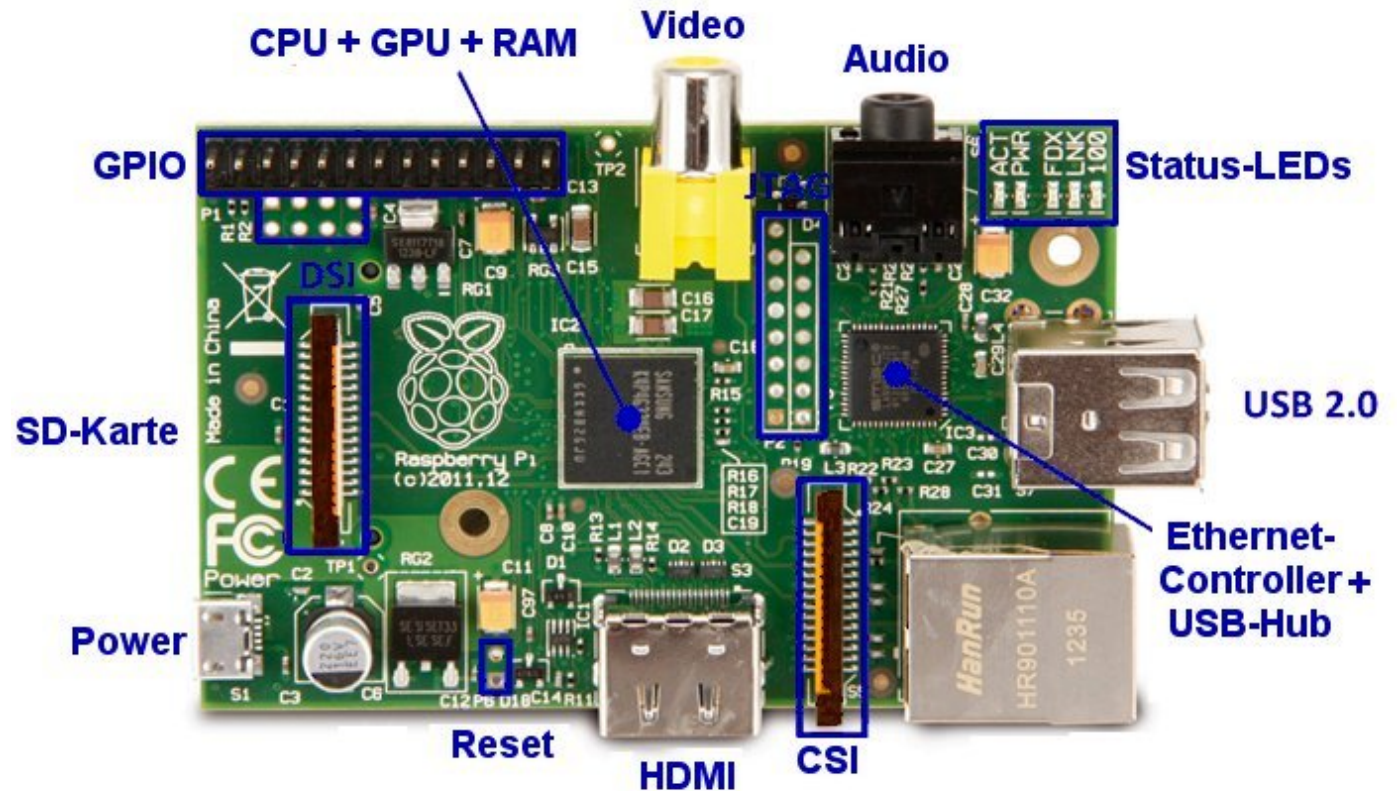
Teilnehmerwünsche

- ▶ Raspberry Pi Zero W in Betrieb nehmen, vorher GPIOs wg. serieller Schnittstelle einlöten.
- ▶ Kernel mit Cross-Compiler selbst compilieren, alternativ: RT-Kernel installieren
- ▶ Yocto mal wieder probieren? (ca. 40GB freier Plattenplatz erforderlich)
- ▶ Arduino?
- ▶ 3D-Druck-Projekte rund um Raspberry Pi
- ▶ ...



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)
- 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



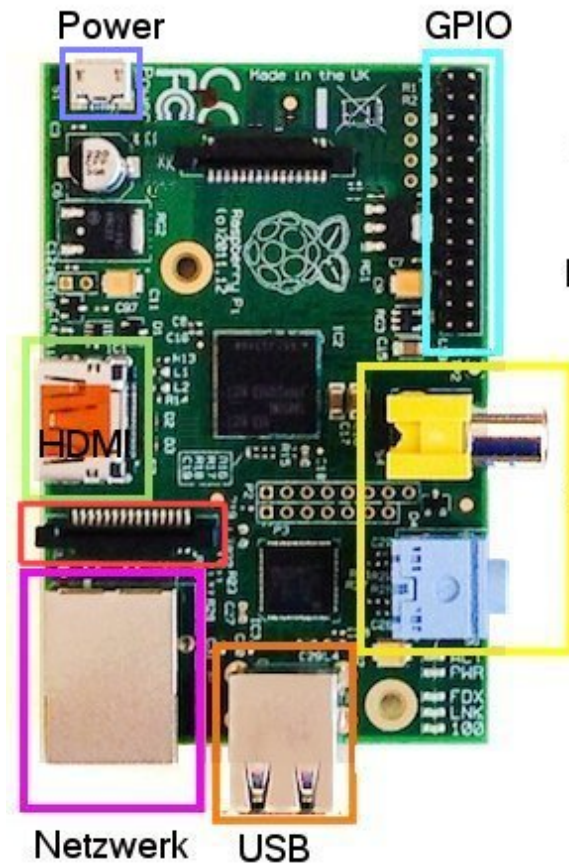
Quelle: netzmafia.de



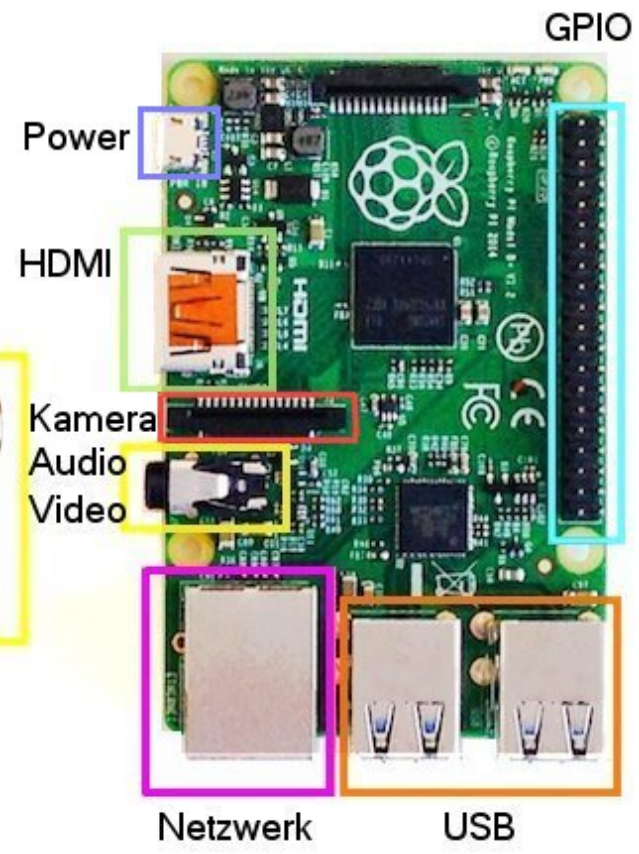
Unterschiede Modell B und B+

- + 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

Modell B



Modell B+



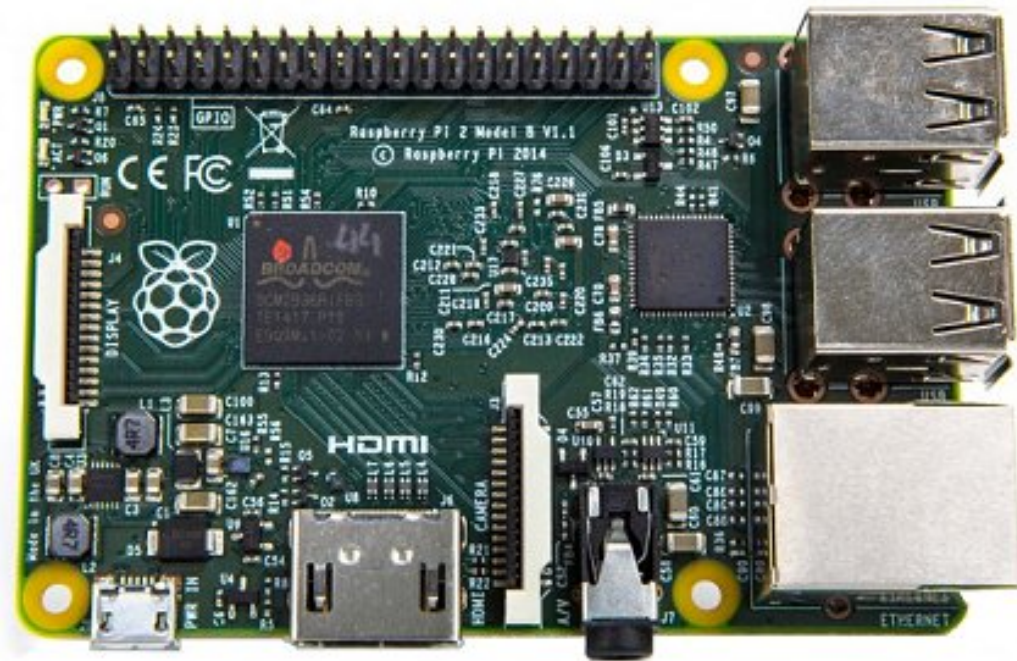
Quelle: netzmafia.de



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

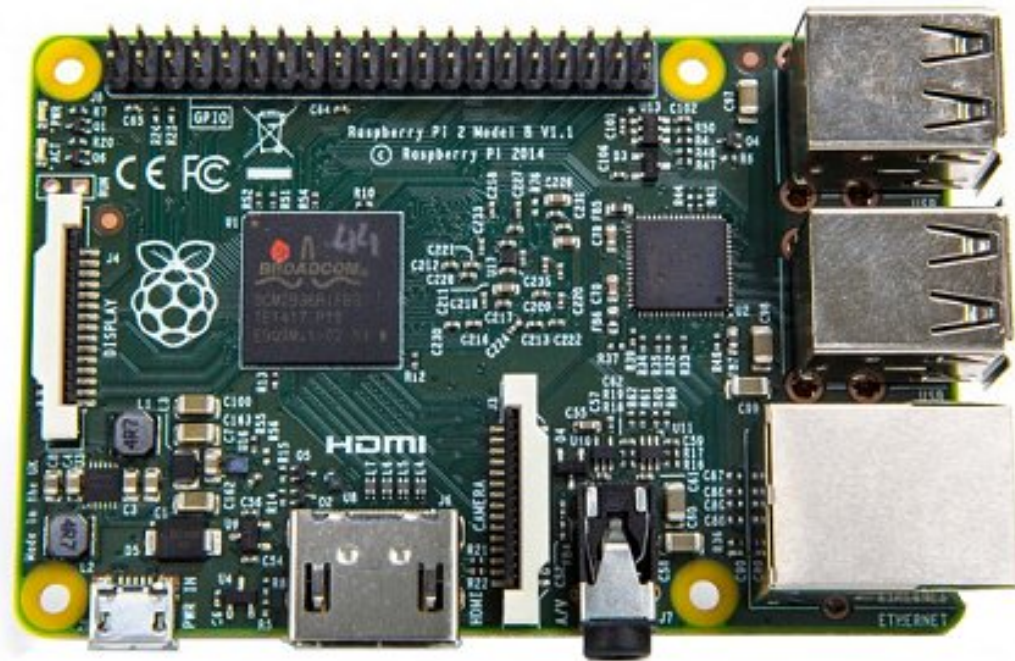


Raspberry Pi 3

- + 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



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



Betriebssysteme (Images)

- ▶ <http://www.raspberrypi.org/downloads/>
- ▶ Auswahl beliebiger SD-Karten-Images im „RasPi-Images“-Ordner auf dem USB-Stick (1. Partition FAT32), zusammen mit Knoppix 8.1 + Arduino-IDE.



Installation / Konfiguration

Grundsätzlich:

- ▶ Image auf SD-Karte entpacken
(z.B. `dd if=raspbian.img of=/dev/sdb`, mehrere Partitionen werden dabei angelegt)
- ▶ 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 Raspian

- ▶ Die Bootdateien liegen bei Raspian 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 3
- ▶



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)

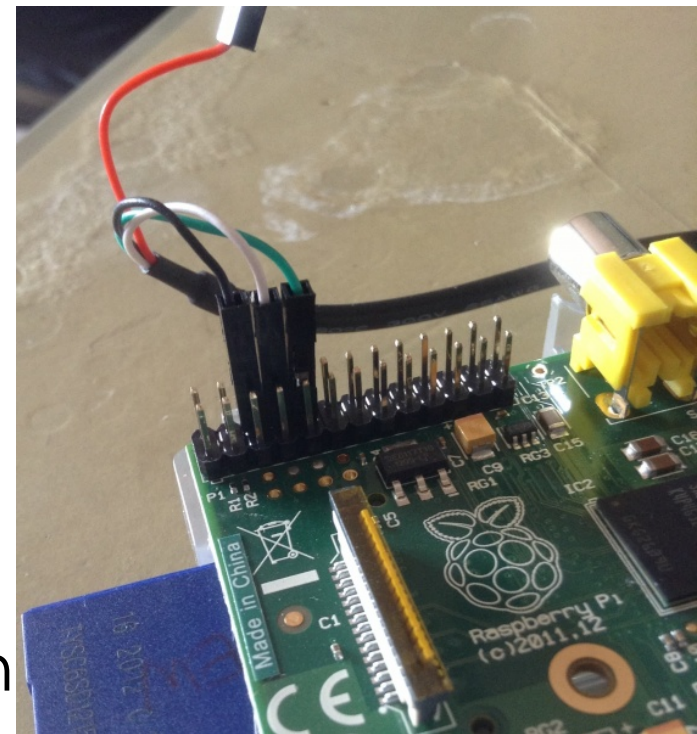
- ▶ **USB → GPIO/UART-Kabel** (Pinbelegung beachten!)

GROUND = Schwarz = Pin 6

TxD = Weiß = Pin 8

RxD = Grün = Pin 10

5V = Rot = Pin 2 *kann* zur
Kopplung der Stromversorgung
angeschlossen werden, ~~sollte~~
muss aber weggelassen werden
falls schon anderweitig Power!



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`



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 pi2
    wpa-psk sperlhof
```

- ▶ Nach dem Speichern:

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



Netzwerk-Konfiguration

- ▶ Während sich `/etc/network/interfaces` um die Konnektierung kümmert, wird der Nameserver 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

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



Apt Proxy

- ▶ Durch Setzen einer Umgebungsvariablen
export http_proxy=http://10.0.0.100:9999
bzw.
export ftp_proxy=http://10.0.0.100:9999
kann apt-get in der aktuellen Shell angewiesen werden, den Cache auf dem Referenten-Rechner zu verwenden (das sollte die Downloads beschleunigen!).



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/>



Viel Spaß beim „Basteln“!

