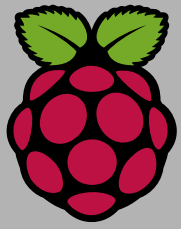


Der Raspberry Pi



KaLUG-Treffen 29.4.2013

Sven Geggus <sven@gegg.us>

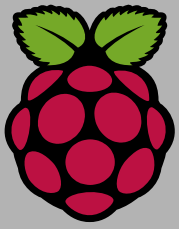


Raspberry Pi: Einleitung



Raspberry Pi ['ʁɑːzbʁi paɪ]

Der wohl preisgünstigste
(vollwertige) Linux PC

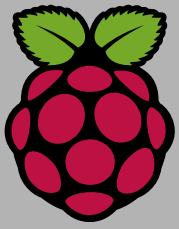


Raspberry Pi: Einleitung



Vorteile - 1(2)

- preisgünstig
- Leichter Zugang zu Peripherieschnittstellen (SPI, I2C, 1-wire, PWM, GPIO, ...)
- Normale PC Peripherie verwendbar
- Standard Linuxdistribution lauffähig (Raspbian ist Debian GNU/Linux)

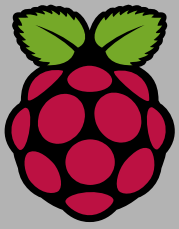


Raspberry Pi: Einleitung



Vorteile - 2(2)

- Ansteuerung der Peripherie über normale Linux Kernel API
- 5MP Kameramodul verfügbar (demnächst)
- Guter community support (Im Vergleich zu Pandaboard, BeagleBoard, etc.)
- Hardware Floating Point Unit

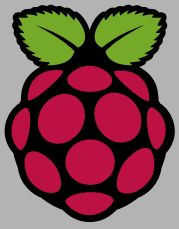


Raspberry Pi: Einleitung



Nachteile (besser geht immer)

- Nur eine LAN Schnittstelle
- Kein WLAN
(Nachrüstbar per USB ca. 10€)
- Keine SATA Schnittstelle
- Keine quelloffene Software für Videocore IV GPU
- Nur ein PWM Interface
(workaround via I2C oder SPI möglich)
- Kein On-Board Flash
- „Low-Level-peripherals“ (3,3V) nicht 5V tolerant

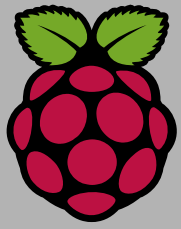


Raspberry Pi: Einleitung



Meine Wishlist (für künftige Software releases)

- Quelloffener Treiber für GPU
- Anschlußmöglichkeit für Smartphone Display an DSI (siehe auch Quelloffener Treiber für GPU...)
- Mainline Kernel Support
- /dev/input/eventX support für GPIO
- Unterstützung für Dateisystem F2FS



Raspberry Pi: Einleitung



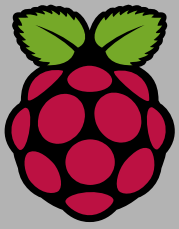
Fazit Vor- und Nachteile

Gut geeignet für:

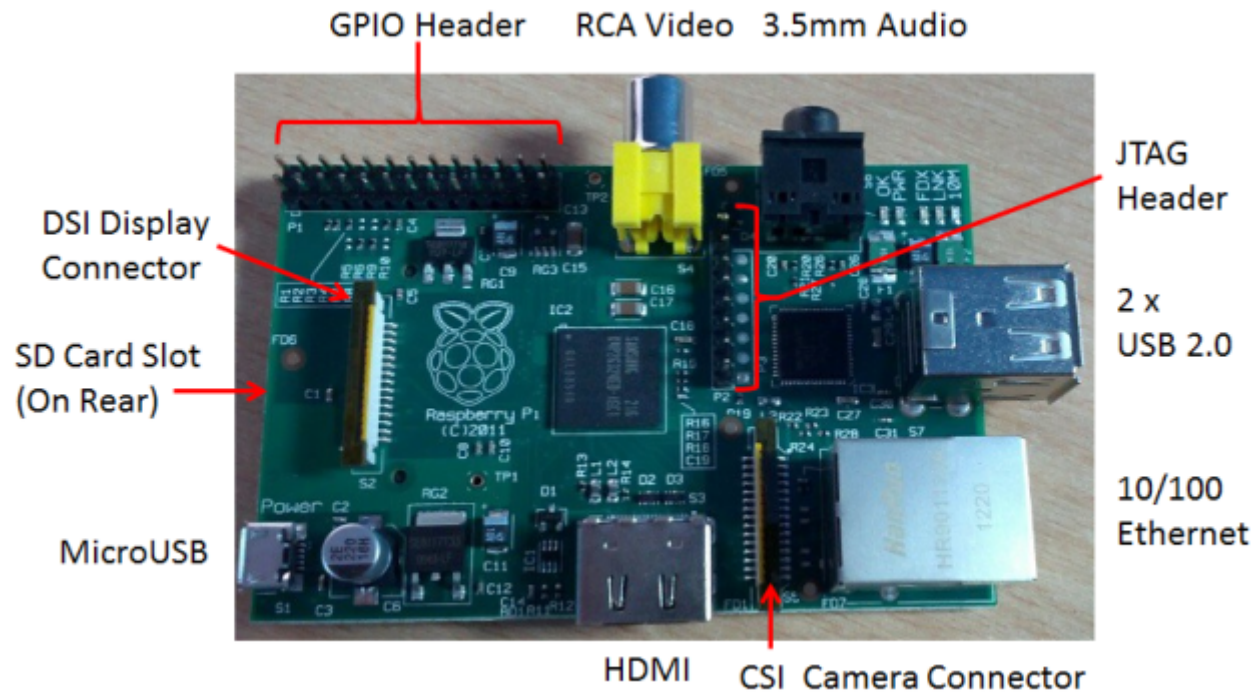
- Hardwarebasteleien
- Media Center

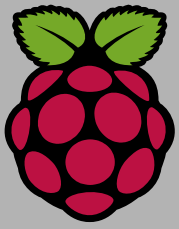
Eher schlecht geeignet für:

- Router
- NAS

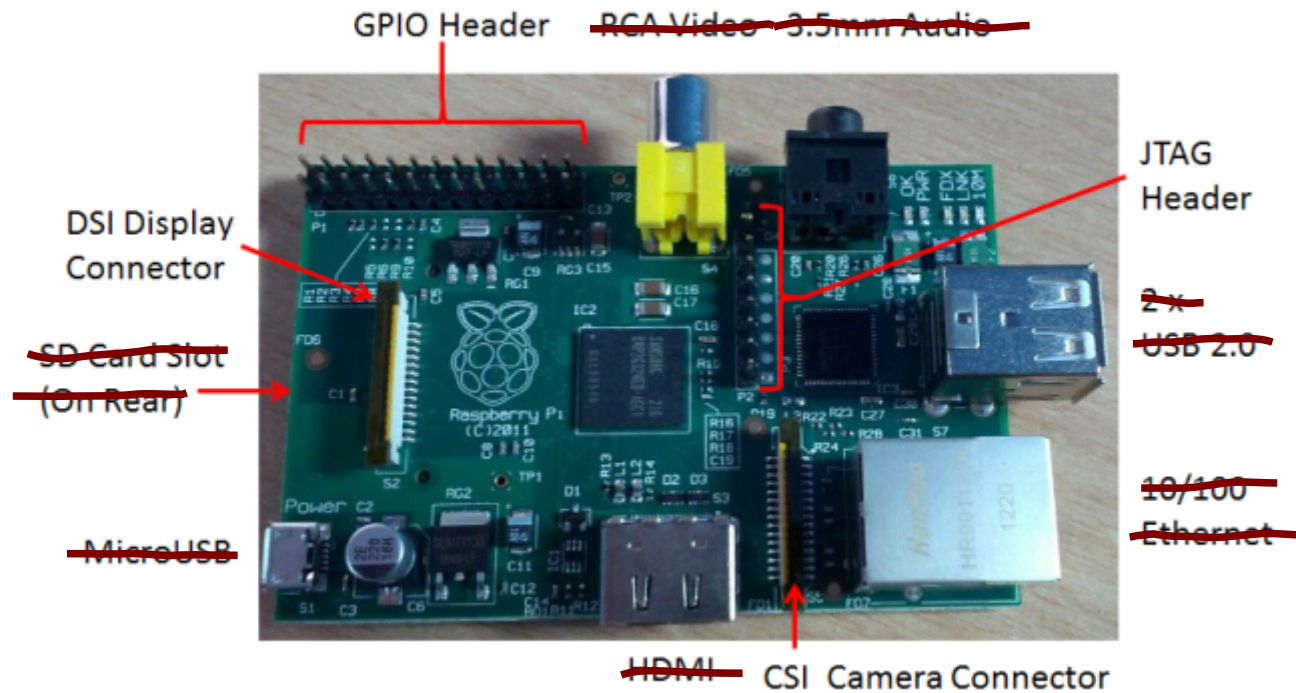


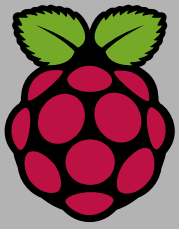
Raspberry Pi: Anschlüsse



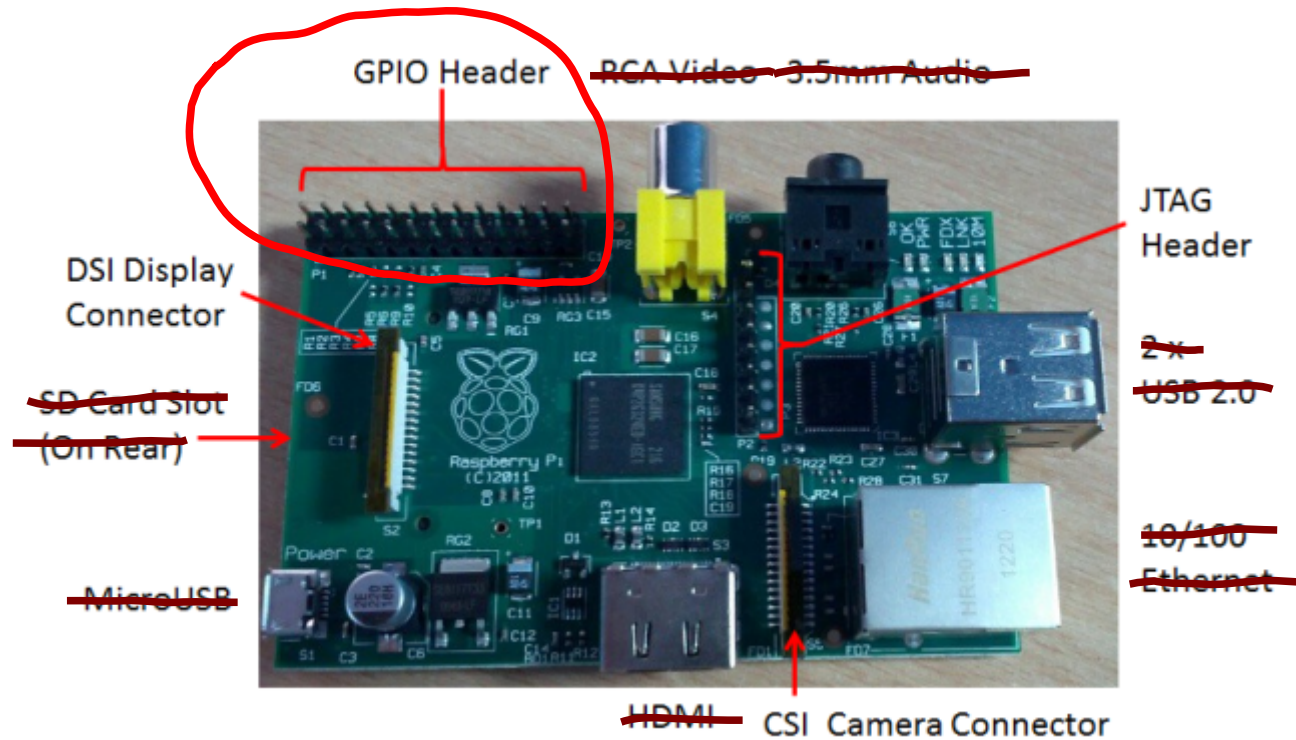


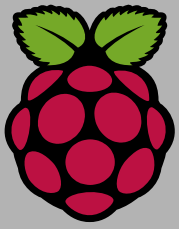
Raspberry Pi: Anschlüsse



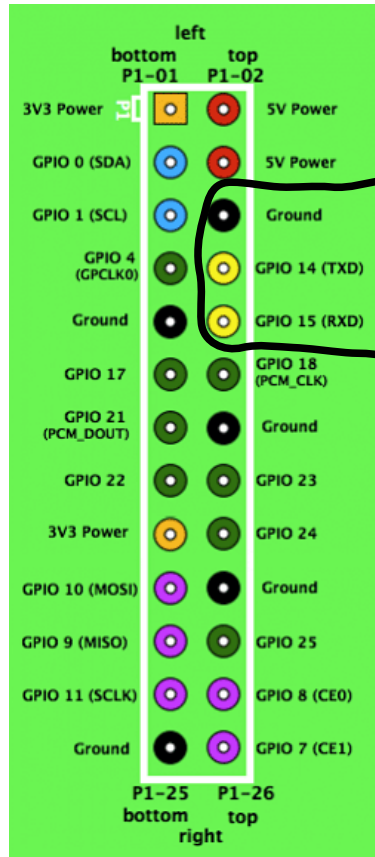


Raspberry Pi: Anschlüsse



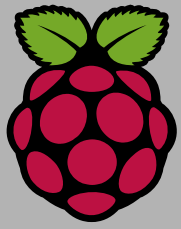


Raspberry Pi: „Low level peripherals“



Die serielle Schnittstelle (RS-232)

- 3,3V mit serieller Konsole
- Umsetzung auf Normpegel und/oder Anschluß an PC über altes Handy Datenkabel (USB oder RS-232) oder MAX2323
- Nach Abschaltung der seriellen Konsole ist der Port als normale RS-232 ohne Handshake-Signale (RTS,CTS, etc.) nutzbar
- Linux API: /dev/ttyS0
- Alternativ als GPIO verwendbar

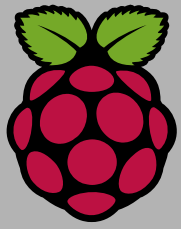


Raspberry Pi: „Low level peripherals“



Das „Serial Peripheral Interface“ (SPI)

- Busspannung 3,3V
- Ein Hardwareinterface zum Anschluss schnellerer Peripherie wie AD-Wandler CAN-Bus Controller etc.
- Linux API: /dev/spidev0.0 und /dev/spidev0.1 oder eigener Kernaltreiber
- Alternativ als GPIO verwendbar

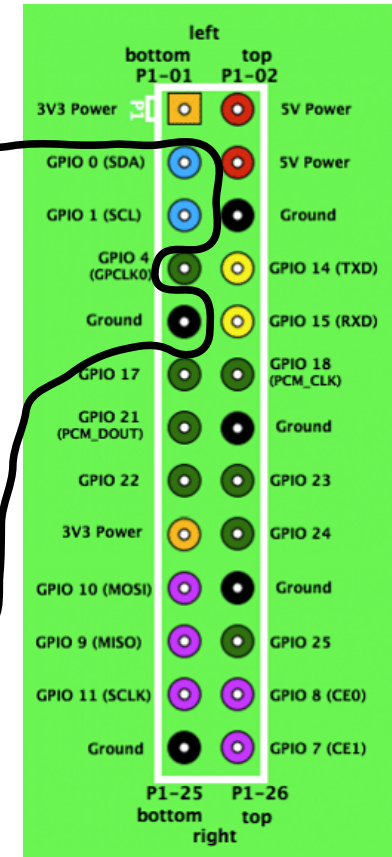


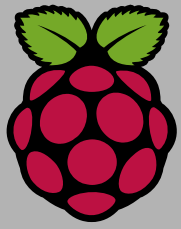
Raspberry Pi: „Low level peripherals“



Der I²C Bus

- Busspannung 3,3V
- Linux API: /dev/i2c-0 bzw. /dev/i2c-1 (je nach Boardversion) oder eigener Kernaltreiber
- Alternativ als GPIO verwendbar



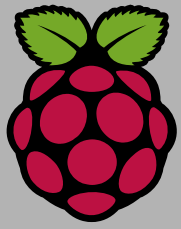


Raspberry Pi: „Low level peripherals“



Die „general purpose IO“ ports (GPIO)

- 3,3V Pegel (Nicht 5 Volt tolerant!)
- Ansteuerung über Linux GPIO API (/sys/class/gpio/)
- Derzeit leider kein Treiber für /dev/input/eventX
- Die SPI, I²C und RS-232 Ports sind alternativ als GPIO verwendbar



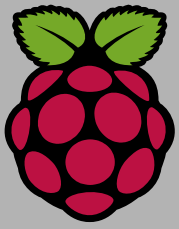
Raspberry Pi: „Low level peripherals“



Der 1-wire Bus

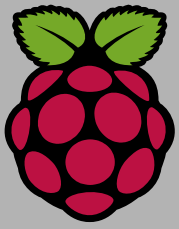
- Proprietärer Bus von Maxim/Dallas
- Software-only über w1-gpio.ko (Beliebiger GPIO Port, default=4)
- Kernaltreiber defekt seit Kernel 2.6.39, repariert in aktuellen Kernels (ggf. patch einspielen, siehe <http://blog.geggus.us/>)
- Besser alternativ über I²C Hardware Busmaster ansteuern





Ansteuerung von I2C Bausteinen

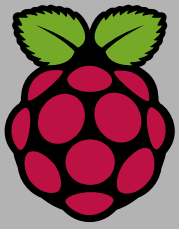
- Aktivierung des I2C Treibers durch Entfernen des Eintrags
`blacklist i2c-bcm2708`
in der Datei
`/etc/modprobe.d/raspi-blacklist.conf`
- Anzeige der Bausteine über Shell-Befehl:
`i2cdetect -y <Busnummer>`
Busnummer ist 0 bei älteren Modellen, 1 bei neueren
Modellen (`grep Revision /proc/cpuinfo`)
- Ansteuerung der Register von I2C Bausteinen über Shell-Befehl `i2cset`
- Details gleich in der Live Demo



Ansteuerung der GPIO über sysfs

- Am Besten sysfsutils installieren
(`apt-get install sysfsutils`)
und danach dann die weitere Konfiguration über
`/etc/sysfs.conf`
- Beispiel `/etc/sysfs.conf` für Taste gegen Masse an
GPIO21 mit Interrupt-Auslösung bei fallender Flanke:

```
class/gpio/export = 21
class/gpio/gpio21/direction = in
class/gpio/gpio21/edge = falling
mode class/gpio/gpio21/value = 666
```



Ansteuerung der GPIO über sysfs

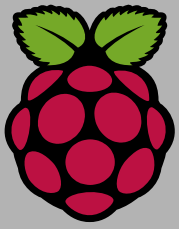
- **Beispiel** `/etc/sysfs.conf` für Relais an GPIO22-25:

```
class/gpio/export = 22  
class/gpio/gpio22/direction = out  
mode class/gpio/gpio22/value = 666
```

```
class/gpio/export = 23  
class/gpio/gpio23/direction = out  
mode class/gpio/gpio23/value = 666
```

```
class/gpio/export = 24  
class/gpio/gpio24/direction = out  
mode class/gpio/gpio24/value = 666
```

```
class/gpio/export = 25  
class/gpio/gpio25/direction = out  
mode class/gpio/gpio25/value = 666
```



Ansteuerung der GPIO über sysfs

- Beispiel GPIO25 über Shellbefehl „high“ schalten

```
echo 1 >/sys/class/gpio/gpio25/value
```

- Beispiel GPIO25 über Shellbefehl „low“ schalten

```
echo 0 >/sys/class/gpio/gpio25/value
```

- Weitere Details gleich in der Live Demo