

# Programm Raspberry-Pi Seminar

## Windischgarsten 2016

(WLAN-Zugang: pi, raspberry, 10.0.0.\*, LAN hat die gleichen Netzwerk-Adressen!)

Referenten-Rechner mit Webserver: **10.0.0.241**

### Ergänzungen (Kurs von 2015):

1. Nach dem Seminar befinden sich alle im Seminar erstellten Unterlagen und gesammelten Erkenntnisse unter  
<http://knopper.net/Windischgarsten>
2. Druckvorlagen und Projekte für den 3D-Druck/3D-Scan usw:  
<http://thingiverse.com>
3. „[Google Cardboard](#)“ heißt die 3D-Brille, mit der man Photosphere-Aufnahmen räumlich anschauen kann. Conrad hat die billigste (3€?), auf Amazon gibt es verschiedene Modelle („Magic Cardboard“, ca. 11 EUR, mit Magnetschalter)
4. Der im Kurs verwendete 3D-Drucker ist der [TeeBot](#), ein Raspberry Pi mit [Octoprint](#) wird als Webschnittstelle verwendet, um ihn übers Netz zu bedienen, zu beobachten, oder auch Ausdrücke abbuchen zu können. Zum 3D-Design und Generieren von Druckdateien werden [openscad](#) bzw. [slic3r](#) verwendet (gibt es für Linux und Windows).

### Vorgeschlagene Projekte:

- **OpenGL in Raspbian** einrichten für 3D-Unterstützung (Openscad, Blender, 3D-Spiele)
- **Kiosk-Modus** beim Einschalten: Browser startet im fullscreen mit einer bestimmten Webseite (Bildschirmschoner aus, Bedienelemente reduziert, Browser soll bei Seitenfehler oder wenn WWW-Server nicht erreichbar ist, automatisch neu starten).
- RaspiCam einbinden, ggf. streamen in Webseite oder per URL
- **Medienkonvertierung**: Wie mache ich aus Videoformat xyz (z.B. DVD-Video) ein auf dem Raspberry hardware-dekodierbares MP4 (h264 Video, AAC Audio)?
- Einbinden und Ansprechen / Weiterleiten von **Windows-Druckern per CUPS**
- **C-Programm** nativ auf RasPi übersetzen, oder Cross-Compilieren unter Intel/AMD für ARM
- Raspberry als **Medienserver** per: SAMBA, Upnp, evtl. Multicast/UDP, z.B. direkt von SAT-DVB-S aufnehmen und ins Netz streamen (Codec gleich lassen!)

- Per PHP-Skript die GPIO-Pins ansprechen (Fernsteuerung über Webinterface)

---

# 1 Einführung

## 1.1 Grundlagen (Theorie)

Allgemeines zu den Raspberry Pi Modellen: Hardware, Leistung, Einsatzzweck (Beispielprojekte)

## 1.2 Hands-on Experimente (Praxis)

Ausprobieren der verschiedenen zur Zeit verfügbaren Linux-Versionen für RasPi mit vorbereiteten SD-Karten-Images:

- Raspbian (Standard-Distribution als Basis für den Eigenbau)
- Ubuntu Mate (Desktop-optimierte Distribution)
- OpenELEC (Multimedia-optimierte Distribution mit Kodi/XBMC Media player)

## 1.3 Selbstbau einer Distribution (Theorie)

Vorgehen beim Modifizieren einer vorhandenen Distribution wie Raspbian vs. Selbstbau „from scratch“, hier wird die Meta-Distribution yocto kurz vorgestellt, mit der sich embedded Distributionen von Grund auf programmieren lassen, u.a. mit Vorlagen für Raspberry Pi.

## 1.4 Brainstorming

ggf. Anpassung des Programms nach Vorstellungen, Wünschen und Ideen der Teilnehmer unter Berücksichtigung des vorhandenen Hard- und Software-Materials und Zeitaufwand.

# 2 Multimedia mit Raspberry Pi

## 2.1 Software

- Einsatz als Medienserver, Streamingserver, Home-Server (mit und ohne Display)
- Umgang mit den diversen Multimedia-Programmen unter Linux (Videoplayer etc.)

## 2.2 Zusatz-Hardware + Treiber

- Einsatz von USB-TV-Sticks (RasPI als stromsparender DVBT-Receiver+Videorecorder)
- Einsatz von USB-Webcams
- Einsatz des RasPI-Kameramoduls
- Infrarot-Fernbedienung
- Touchscreen

## 2.3 Remote Control

- Zugriff auf Webcams oder Kameramodul aus Applikationen
- Anwendung als Überwachungsgerät z.B. für 3D-Drucker mit Notabschaltung, fernbedienbare Kamera.

## 3 Erweiterungen

Beispiele: Images oder Inhalte von Images sichern, Images (Partitionen) mit mount -o loop mounten.

### 3.1 Kernel-Erstellung und Modifikationen

Cross-Compiling von Kernel und Softwarepaketen aus Debian

### 3.2 Setups

- Raspi als Mehrzweck-Heim-Router
- transparenter Proxy
- Torrent-Client/Server
- Firewall,
- VPN-Einwahl-Server (OpenVPN oder einfachere Lösungen wie SSH-Tunnel, httptunnel, ...)

### 3.3 Programmieren direkt auf dem Raspberry Pi

Entwicklungsumgebungen und Programmiersprachen, *nach verfügbarer Zeit und Interesse*

- C/C++
- HTML/PHP/Javascript
- Datebanken (MySQL)
- Java
- Python
- Smalltalk (Squeak)