

XBMC Installationsanleitung Ubuntu 10.10 64 Bit Installation



Inhaltsverzeichnis:

- 1 Betriebssystem Installation
 - 1.1 Bootmedium vorbereiten
 - 1.1.1 Ubuntu Mini downloaden
 - 1.1.2 UnetBootin downloaden
 - 1.1.3 Ubuntu Stick vorbereiten
 - 1.2 Ubuntu installieren
 - 1.2.1 vom Stick booten
 - 1.2.2 Installation
- 2 VDR Installation
 - 2.1 Source anbinden
 - 2.2 VDR kompilieren und installieren
 - 2.3 Plugins
 - 2.3.1 --Platzhalter--
 - 2.3.2 VNSI-Plugin
 - 2.3.3 Wirbelscan-Plugin
 - 2.3.4 femon-Plugin
 - 2.3.5 epgsearch-Plugin
- 3 XBMC Installation
 - 3.1 Vorbereitung
 - 3.1.1 Abhängigkeiten installieren
 - 3.1.2 Repositories anbinden
 - 3.1.3 Pakete installieren
 - 3.2 Grafikkartentreiberinstallation
 - 3.2.1 Abhängigkeiten installieren
 - 3.2.2 Treiber Installation
 - 3.2.3 Treiber Test
 - 3.2.4 X11 Berechtigung setzen
 - 3.2.5 VA-API Einrichtung

- 3.2.5.1 Libdrm installieren
- 3.2.5.2 Hacked Libs installieren
- 3.2.5.3 VAAPI testen
- 3.2.6 XBMC bauen
 - 3.2.6.1 LibBluray bauen
 - 3.2.6.2 XBMC Sourcen holen
 - 3.2.6.3 XBMC installieren
 - 3.2.6.4 XBMC Autostart

4 Hardware einrichten

- 4.1 TV Karten Treiber installieren
- 4.2 Alsa updaten
- 4.3 AMD Zacate Soundfix
- 4.4 Lirc installieren
- 4.5 LCDprog installieren
- 4.6 zweite HDD anbinden

5 Konfiguration

- 5.1 VDR
 - 5.1.1 Kanalliste
- 5.2 XBMC
 - 5.2.1 Sprache
 - 5.2.2 PVR Client
 - 5.2.3 LiveTV
 - 5.2.4 Vsync
 - 5.2.5 Skin
- 5.3 Lirc
- 5.4 LCD
- 5.5 Boot Splash XBMC
- 5.6 XBMC User Shutdown Recht

1. Betriebssystem Installation

1.1 Bootmedium vorbereiten

1.1.1 Ubuntu Mini downloaden

Zunächst lädt man die Minimal installations iso von Ubuntu herunter. Für dieses How-To verwende ich die 10.10 x64 minimal CD.

Maverick ISO:

<http://archive.ubuntu.com/ubuntu/dists/maverick/main/installer-amd64/current/images/netboot/mini.iso>

Natty ISO: <http://archive.ubuntu.com/ubuntu/dists/natty/main/installer-amd64/current/images/netboot/mini.iso>

1.1.2 UnetBootin downloaden

Ich bevorzuge die Installtion vom USB Stick, dazu brauche ich das kleine Tool UnetBootin!

<http://www.computerbase.de/downloads/system/universal-netboot-installer/>

Ein kleiner 128MB Stick reicht vollkommen, einfach anstecken, Programm öffnen, unten die mini.iso auswählen und den InstallationsStick erstellen.

1.1.3 Ubuntu Stick vorbereiten



1.2 Ubuntu installieren

1.2.1 vom Stick booten

Im weiteren bootet man den HTPC von diesem Stick, oft kann man dies via Einmalboot vornehmen indem man beim BIOS Screen auf F12 drückt. Ist aber abh. vom Boardhersteller.

Das Bootmenü erscheint, man wähle "Install" und wartet bis die eigentliche Ubuntu-Installation losgeht.

1.2.2 Installation

- Sprache German auswählen
- Land Deutschland auswählen
- Tastaturmodell erkennen „Ja“
- DHCP abbrechen
- Statische IP-Adresse vergeben
- USB Stick abziehen!!!
- Rechnernamen eingeben
- Spiegelserver Deutschland
- Festplatten partitionieren: Geführt - verwende vollständige Festplatte
- Benutzer xbmc einrichten
- keine automatische Aktualisierung
- SSH installieren
- Grub-Bootloader mit Ja installieren

2. VDR Installation

2.1 Source anbinden

Eintrag in die sources.list einfügen

```
nano /etc/apt/sources.list
```

```
deb-src http://e-tobi.net/vdr-experimental squeeze base backports addons  
vdr-multipatch
```

```
wget http://e-tobi.net/vdr-experimental/pool-squeeze/binary/base/e-tobi-  
keyring_2008.03.08_all.deb
```

```
dpkg -i e-tobi-keyring_2008.03.08_all.deb
```

2.2 VDR kompilieren und installieren

Abhängigkeiten installieren

```
apt-get install git-core make g++ gcc gawk pmount libtool nasm yasm
automake cmake gperf zip unzip bison libSDL-dev libSDL-image1.2-dev
libSDL-gfx1.2-dev libSDL-mixer1.2-dev libfribidi-dev liblzo2-dev
libfreetype6-dev libsqlite3-dev libogg-dev libasound-dev python-sqlite
libglew-dev libcurl3 libcurl4-gnutls-dev libxrandr-dev libxrender-dev
libmad0-dev libogg-dev libvorbisenc2 libsmbclient-dev libmysqlclient-dev
libpcre3-dev libdbus-1-dev libhal-dev libhal-storage-dev libjasper-dev
libfontconfig-dev libbz2-dev libboost-dev libenca-dev libxt-dev libxtst-
dev libxmu-dev libpng-dev libjpeg-dev libpulse-dev mesa-utils libcdio-dev
libsamplerate-dev libmms-dev libmpeg3-dev libfaad-dev libflac-dev
libiso9660-dev libass-dev libssl-dev fp-compiler gdc libwavpack-dev
libmpeg2-4-dev libmicrohttpd-dev libmodplug-dev libssh-dev gettext cvs
autopoint dh-autoreconf ccache python-all-dev sendmail dialog
libncurses5-dev
```

```
apt-get install subversion make g++ gcc gawk pmount libtool nasm automake
cmake gperf unzip bison libSDL-dev libSDL-image1.2-dev libSDL-gfx1.2-dev
libSDL-mixer1.2-dev libSDL-sound1.2 checkinstall libSDL-dev libSDL-
image1.2-dev libSDL-gfx1.2-dev libSDL-mixer1.2-dev libfribidi-dev liblzo2-
dev libfreetype6-dev libsqlite3-dev libogg-dev libasound2-dev python-
sqlite libglew-dev libcurl3 libcurl4-openssl-dev x11proto-xinerama-dev
libxinerama-dev libxrandr-dev libxrender-dev libmad0-dev libogg-dev
libvorbis-dev libsmbclient-dev libmysqlclient-dev libpcre3-dev libdbus-1-
dev libhal-dev libhal-storage-dev libjasper-dev libfontconfig-dev libbz2-
dev libboost-dev libfaac-dev libenca-dev libxt-dev libxtst-dev libxmu-dev
libpng-dev libjpeg-dev libpulse-dev mesa-utils libcdio-dev libsamplerate-
dev libmms-dev libmpeg3-dev libfaad-dev libflac-dev libiso9660-dev libass-
dev libssl-dev libwavpack-dev libmpeg2-4 libmpeg2-4-dev libmicrohttpd-dev
libmicrohttpd5 libmodplug-dev yasm python2.7-dev libvdpau-dev autopoint
```

```
apt-get update
```

```
apt-get build-dep vdr-dev
```

VDR kompilieren und installieren

```
cd /usr/src
```

```
apt-get source vdr
```

```
cd /usr/src/vdr-1.7.18/
```

```
dpkg-buildpackage -rfakeroot -us -uc -b
```

Nun findet man das fertige VDR-Paket in /usr/src . Da wir es für das Kompilieren der Plugins benötigen, installieren wir auch das vdr-dev-Paket:

```
cd /usr/src
```

```
dpkg -i vdr-dev_1.7.18-2~ctvdr1_all.deb vdr_1.7.18-2~ctvdr1_amd64.deb
```

2.3 Plugins

2.3.1 --Platzhalter--

2.3.2 VNSI-Plugin

Das VNSI-Plugin ist die Schnittstelle von VDR zu XBMC.

```
# wenn git noch nicht installiert ist, folgendes ausführen
$ sudo apt-get install git-core
$ cd /usr/src/vdr-1.7.18/PLUGINS/src
# Bei einem Update statt dem nächsten Schritt "git pull" im Verzeichnis
/usr/src/vdr-1.7.18/PLUGINS/src/vdr-plugin-vnsiserver ausführen!
$ git clone https://github.com/pipelka/vdr-plugin-vnsiserver.git
$ cd /usr/src/vdr-1.7.18/PLUGINS/src/vdr-plugin-vnsiserver
$ make
$ cp /usr/src/vdr-1.7.18/PLUGINS/lib/libvdr-vnsiserver.so.1.7.18
/usr/lib/vdr/plugins/
```

Config-Files kopieren

```
$ cp -R /usr/src/vdr-1.7.18/PLUGINS/src/vdr-plugin-vnsiserver/vnsiserver/
/var/lib/vdr/plugins/
$ ln -s /var/lib/vdr/plugins/vnsiserver /var/lib/vdr/plugins/vnsi-server
$ chown vdr:vdr /var/lib/vdr/plugins/vnsiserver/allowed_hosts.conf
```

Die Datei /var/lib/vdr/plugins/vnsi-server/allowed_hosts.conf sollte dann auf das interne Netz angepasst werden:

Anschließend muss VDR neu gestartet werden, um das neue Plugin zu laden:

```
$ /etc/init.d/vdr restart
```

2.3.3 Wirbelscan-Plugin

Das Wirbelscan-Plugin wird zur Kanalsuche verwendet. Entsprechende Optionen sind in XBMC enthalten.

```
$ cd /usr/src/vdr-1.7.18/PLUGINS/src
$ apt-get build-dep vdr-plugin-wirbelscan
$ apt-get source vdr-plugin-wirbelscan
$ cd /usr/src/vdr-1.7.18/PLUGINS/src/vdr-plugin-wirbelscan-0.0.7
$ make
$ cp /usr/src/vdr-1.7.18/PLUGINS/lib/libvdr-wirbelscan.so.1.7.18
/usr/lib/vdr/plugins/
```

2.3.4 femon-Plugin

```
$ cd /usr/src/vdr-1.7.18/PLUGINS/src/
$ apt-get build-dep vdr-plugin-femon
$ apt-get source vdr-plugin-femon
$ cd /usr/src/vdr-1.7.18/PLUGINS/src/vdr-plugin-femon-1.7.10/
$ dpkg-buildpackage -rfakeroot -us -uc -b
$ cd /usr/src/vdr-1.7.18/PLUGINS/src/
$ sudo dpkg -i vdr-plugin-femon_1.7.10-1_amd64.deb
```

2.3.5 epqsearch-Plugin

```
$ cd /usr/src/vdr-1.7.18/PLUGINS/src/
$ apt-get build-dep vdr-plugin-epgsearch
$ apt-get source vdr-plugin-epgsearch
$ cd /usr/src/vdr-1.7.18/PLUGINS/src/vdr-plugin-epgsearch-0.9.25~beta21/
$ dpkg-buildpackage -rfakeroot -us -uc -b
$ cd /usr/src/vdr-1.7.18/PLUGINS/src/
$ sudo dpkg -i vdr-plugin-epgsearch_0.9.25~beta21-2_amd64.deb
```

3. XBMC Installation

3.1 Vorbereitung

3.1.1 Abhängigkeiten installieren

```
$ sudo apt-get install ssh python-software-properties
$ sudo apt-get install xinit x11-xserver-utils upower udisks pm-utils
$ sudo apt-get install subversion make g++ gcc gawk pmount libtool nasm
yasm automake cmake gperf zip unzip bison libSDL-dev libSDL-image1.2-dev
libSDL-gfx1.2-dev libSDL-mixer1.2-dev libfribidi-dev liblzo2-dev
libfreetype6-dev libsqlite3-dev libogg-dev libasound-dev python-sqlite
libglew-dev libcurl3 libcurl4-gnutls-dev libxrandr-dev libxrender-dev
libmad0-dev libogg-dev libvorbisenc2 libsmbclient-dev libmysqlclient-dev
libpcre3-dev libdbus-1-dev libhal-dev libhal-storage-dev libjasper-dev
libfontconfig-dev libbz2-dev libboost-dev libenca-dev libxt-dev libxmu-dev
libpng-dev libjpeg-dev libpulse-dev mesa-utils libcdio-dev libsamplerate-
dev libmpeg3-dev libflac-dev libiso9660-dev libass-dev libssl-dev fp-
compiler gdc libmpeg2-4-dev libmicrohttpd-dev libmodplug-dev libssh-dev
gettext cvs python-dev libyajl-dev
```

3.1.2 Repositorys anbinden

```
$ sudo su
$ add-apt-repository ppa:ubuntu-x-swat/x-updates
$ add-apt-repository ppa:alexandr-surkov/vdr-maverick
$ add-apt-repository ppa:lars-opdenkamp/xbmc-pvr
$ add-apt-repository ppa:xorg-edgers/ppa
$ apt-get update
```

3.1.3 Pakete installieren

```
$ sudo apt-get install xorg alsa-utils mesa-utils git-core fontconfig
liblcms1 libmng1 libqt3-mt librtmp0 libvdpaul python-qt3 python-sip ttf-
liberation
```

3.2 Grafikkartentreiberinstallation

3.2.1 Abhängigkeiten installieren

```
$ sudo apt-get install build-essential cdbx fakeroot dh-make debhelper  
debconf libstdc++6 dkms libqtgui4 wget execstack libelfg0
```

```
$ sudo apt-get install ia32-libs
```

3.2.2 Treiber Installieren

```
$ cd /  
$ mkdir catalyst  
$ cd catalyst  
$ wget http://www2.ati.com/drivers/linux/ati-driver-installer-11-6-  
x86.x86_64.run  
$ chmod +x ati-driver-installer-11-6-x86.x86_64.run  
$ sudo apt-get remove --purge xserver-xorg-video-radeon  
$ sudo apt-get remove --purge fglrx fglrx_* fglrx-amdcccle* fglrx-dev*  
xorg-driver-fglrx  
$ sh ati-driver-installer-11-6-x86.x86_64.run --buildpkg Ubuntu/maverick
```

Erstellte .deb Pakete installieren

```
$ sudo dpkg -i fglrx*.deb  
$ sudo aticonfig --initial -f  
$ sudo aticonfig --input=/etc/X11/xorg.conf --tls=1  
$ reboot
```

3.2.3 Treiber Test

Nach dem Reboot folgendes eingeben:

```
$ startx
```

(Achtung Monitor muss mehr als 1024x768 können sonst sieht man nur schwarz) oben links erscheint ein kleiner Cursor, dort

```
$ fglrxinfo
```

eingeben, ATI Treiber muss angezeigt werden mit Grafikkartentyp dann ok zum Testen kann man auch noch

```
$ fgl_glxgears
```

eingeben, dann sollte ein 3D Würfel erscheinen, mit ESC beenden

3.2.4 X11 Berechtigung setzen

```
$ adduser xbmc video  
$ adduser xbmc audio  
$ reboot  
$ sudo su  
$ sh -c 'echo "allowed_users=anybody" > /etc/X11/Xwrapper.config'  
$ reboot
```

3.2.5 VAAPI Einrichtung

3.2.5.1 Libdrm installieren

```
$ cd /catalyst
$ wget http://dri.freedesktop.org/libdrm/libdrm-2.4.26.tar.gz
$ tar -zxpvf libdrm-2.4.26.tar.gz
$ sudo apt-get install pkg-config
$ sudo apt-get install libpthread-stubs0
$ sudo apt-get install libpthread-stubs0-dev
$ sudo apt-get install libpciaccess-dev
$ cd libdrm-2.4.26
$ ./configure
$ make
$ make install
$ apt-get install libdrm-dev
```

3.2.5.2 Hacked Libs installieren

```
$ cd /catalyst
$ wget http://www.splitted-desktop.com/~gbeauchesne/libva/pkgs/amd64/libva1\_0.32.0-1+sds2\_amd64.deb
$ wget http://www.splitted-desktop.com/~gbeauchesne/libva/pkgs/amd64/libva-dev\_0.32.0-1+sds2\_amd64.deb
$ wget http://www.splitted-desktop.com/~gbeauchesne/xvba-video/xvba-video\_0.7.8-1\_amd64.deb
$ sudo dpkg -i libva1_0.32.0-1+sds2_amd64.deb
$ sudo dpkg -i libva-dev_0.32.0-1+sds2_amd64.deb
$ sudo dpkg -i xvba-video_0.7.8-1_amd64.deb
$ sudo apt-get upgrade
$ reboot
```

3.2.5.3 VAAPI testen

```
$ startx
```

warten bis kleiner Cursor oben links vorhanden

```
$ vainfo
```

3.2.6 XBMC bauen

3.2.6.1 LibBluray bauen

```
$ cd /usr/src
$ git clone git://git.videolan.org/libbluray.git
$ cd libbluray
$ ./bootstrap
$ ./configure --prefix=/usr
```

```
$ make
$ make install
```

3.2.6.2 XBMC Sourcen holen

```
$ cd /usr/src
$ git clone https://github.com/opdenkamp/xbmc.git
```

3.2.6.4 XBMC installieren

```
$ cd /usr/src/xbmc
$ ./bootstrap
$ ./configure --prefix=/usr --enable-vaapi --enable-libbluray
$ make -j2
$ make install
```

3.2.6.5 XBMC Autostart

```
$ wget http://paste.kde.org/7158/raw/ -O /etc/init/xbmc.conf
$ ln -s /lib/init/upstart-job /etc/init.d/xbmc
$ wget http://paste.kde.org/7022/raw/ -O /usr/bin/runXBMC
$ chmod a+x /usr/bin/runXBMC
$ reboot
```

4. Hardware einrichten

4.1 TV Karten Treiber installieren

```
$ apt-get install s2-liplianin-dkms
```

oder

```
$ cd /usr/src
$ git clone git://linuxtv.org/media_build.git
$ cd media_build
$ apt-get install libproc-processtable-perl patchutils
$ ./build.sh
$ make install
$ reboot
```

4.2 Alsa updaten

```
$ sudo su
$ cd /usr/src
$ wget http://dl.dropbox.com/u/21136636/AlsaUpgrade-1.0.24-2.sh
$ chmod 777 /usr/src/AlsaUpgrade-1.0.24-2.sh
$ ./AlsaUpgrade-1.0.24-2.sh -d
```

```
$ ./AlsaUpgrade-1.0.24-2.sh -c
$ ./AlsaUpgrade-1.0.24-2.sh -i
$ reboot
```

4.3 AMD Zacate Soundfix

```
$ sudo nano /etc/modprobe.d/alsa-base.conf
letzte Zeile einfügen: options snd_hda_intel index=1
$ reboot
$ alsamixer
mit "m" den spdif unmuten (darauf achten mehrere /PDIF) und damit laut
schalten!
$ sudo alsactl store
```

4.4 Lirc installieren

```
$ apt-get install lirc
Remote control configuration: Soundgraph iMON Antec Veris
IR transmitter: none
```

4.5 LCDprog installieren

```
$ apt-get install lcdproc
```

4.6 zweite HDD anbinden

Partitionieren

```
$ fdisk /dev/sdb (sdb = zweite Festplatte)
Befehl p (alle Partitionen anzeigen)
Befehl d (Partitionen löschen)
Befehl n (neue Partition anlegen)
Befehl p (Primäre Partition anlegen)
Befehl 1 (Partition 1 anlegen)
Befehl 1 (Erster Zylinder 1 festlegen)
Befehl 243201 (Letzter Zylinder angeben)
Befehl w (Änderungen speichern und Fdisk beenden)
```

Formatieren

```
$ mkfs.ext4 /dev/sdb1
```

Automount

```
$ cd /media
$ mkdir daten
$ chgrp xbmc daten
$ chmod 770 daten
```

HDD Mounten

1. UUID rausfinden `blkid`
2. `/etc/fstab` bearbeiten

```
UUID=41c22818-fbad-4da6-8196-c816df0b7aa8 /media/daten ext4
defaults,errors=remount-ro 0 1
```

Ordner für XBMC auf HDD anlegen

```
$ cd /media/daten
$ mkdir bilder
$ mkdir musik
$ mkdir videos
$ chgrp xbmc bilder
$ chgrp xbmc musik
$ chgrp xbmc videos
$ chown 770 bilder
$ chown 770 musik
$ chown 770 videos
```

5. Konfiguration

5.1 VDR

5.1.1 Kanalliste

```
$ sudo su
$ /etc/init.d/vdr stop
$ cd /var/lib/vdr
$ rm channels.conf
$ wget http://wbreu.htpc-forum.de/downloads/1120311kurz.conf
$ mv 1120311kurz.conf channels.conf
$ /etc/init.d/vdr start
```

5.1.2 Softcam Einrichtung

noch prüfen

5.2 XBMC konfigurieren

5.2.1 Sprache

Unter System -> Configurartion -> Apperance -> International auf Deutsch stellen

5.2.2 PVR Client

Dann unter Einstellungen -> Addons -> installierte Addons -> PVR Klienten -> VSNI "aktivieren"

5.2.3 LiveTV

Unter Einstellungen -> LiveTV -> LiveTV aktivieren

5.2.4 Vsync

Einstellungen -> System -> Video Settings -> VSync immer an!

5.3 Lirc

noch prüfen

5.4 LCD

```
$ nano /etc/LCDD.conf

- Driver=curses ersetzen mit Driver=imonlcd
- Hello und GoodBye auskommentieren
- In der Sektion Soundgraph iMON LCD
  - Protocol=1
  - OnExit=1
  - Contrast=1
  - DiskMode=1
$ /etc/init.d/LCDD restart
```

5.5 Boot Splash XBMC

```
$ aptitude install plymouth-label
$ wget http://excycle.nl/plymouth-theme-xbmc-logo.deb
$ dpkg -i plymouth-theme-xbmc-logo.deb
$ sudo apt-get install v86d
$ sudo nano /etc/default/grub
```

Folgende Zeilen ausfindig machen:

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
#GRUB_GFXMODE=640x480
```

und durch diese ersetzen (Eigene Auflösung angeben!)

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash nomodeset
video=uvesafb:mode_option=1366x768-24,mtrr=3,scroll=ywrap"
GRUB_GFXMODE=1366x768
```

```
$ sudo nano /etc/initramfs-tools/modules
```

Diese Zeile hinzufügen (Wieder eigene Auflösung angeben!):

```
uvesafb mode_option=1366x768-24 mtrr=3 scroll=ywrap
```

```
$ echo FRAMEBUFFER=y | sudo tee /etc/initramfs-tools/conf.d/splash
$ sudo update-grub
$ sudo update-initramfs -u
```

```
$ reboot
```

5.6 XBMC User Shutdown Recht

```
/etc/sudoers
```

```
easyys ALL = NOPASSWD: /sbin/shutdown
```