

Raspberry PI 2B / 3B+ /4 Grundsatz einrichten mit Oscam-Umgebung (Build und Server) mit Raspbian Buster

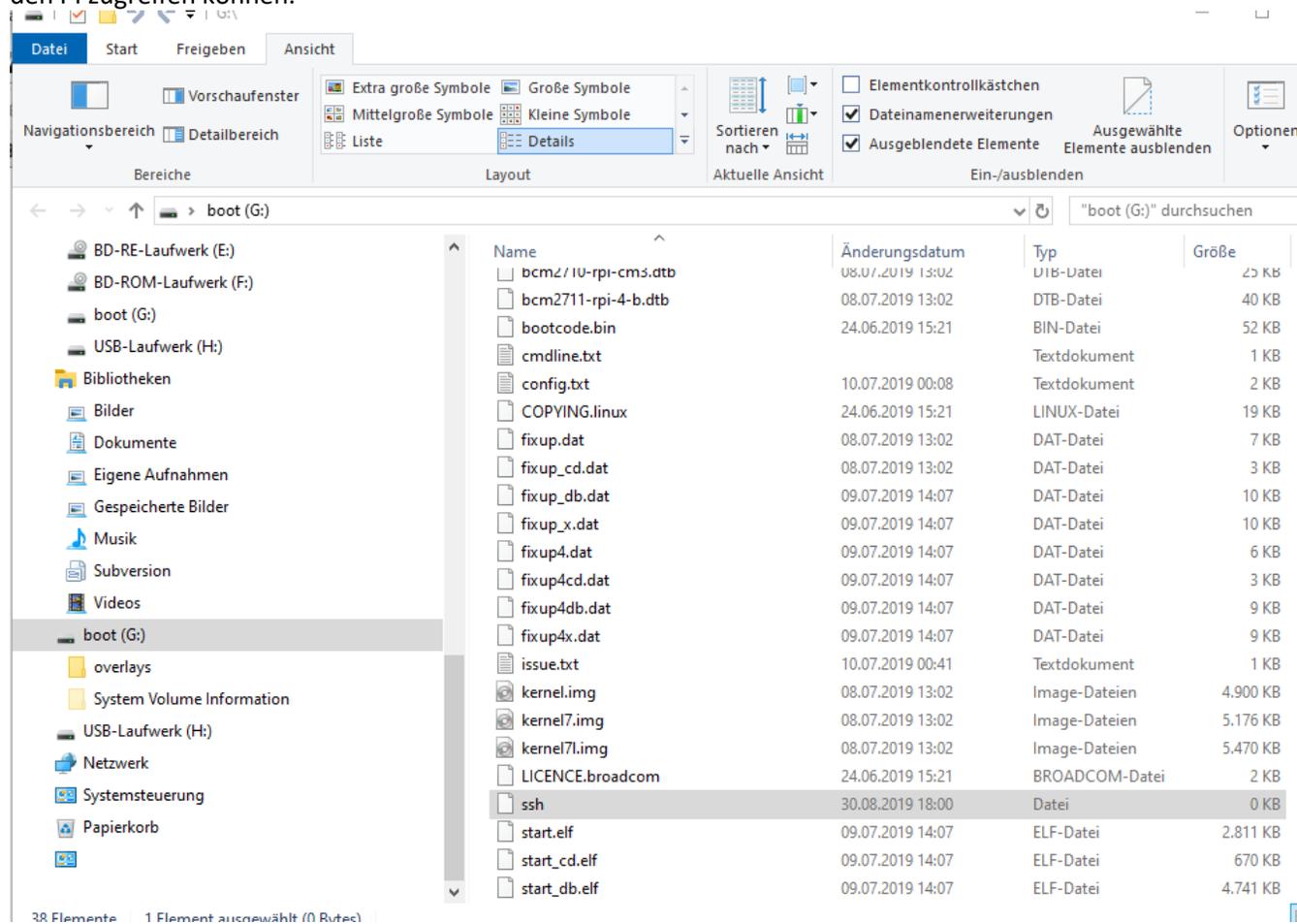
Inhalt

Systeminstallation Raspbian Buster (optimiert).....	2
Oscam installieren inkl. Buildumgebung.....	17
Buildumgebung einrichten und oscam-binary erzeugen	17
Oscam-Server einrichten	26
OSCAM konfigurieren	32
OSCAM updaten	36

Systeminstallation RaspBian Buster (optimiert)

Das aktuelle RASPBIAN Buster full [herunterladen](#) und mit [Win32diskimager](#) auf eine MicroSD Karte schreiben.

Nach erfolgreichem Schreiben eine (leere) Datei „ssh“ auf dem Laufwerk erzeugen, damit wir von Anfang an mit Putty oder einem anderen ssh-Client auf den PI zugreifen können.



Dann legen wir die SD-Card in den PI und booten

Jetzt kommen wir schon mit ssh (putty) auf den PI und loggen uns mit User PI und Passwort raspberry ein.

Zu allererst vergeben wir für root ein passwort

```
sudo passwd
```

```
pi@raspberrypi:~$ sudo passwd
New password:
Retype new password:
passwd: password updated successfully
pi@raspberrypi:~$ █
```

Anschließend ändern wir die Config

```
sudo raspi-config
```

```
pi@raspberrypi:~$ sudo raspi-config
Created symlink /etc/systemd/system/multi-user.target.wants/vncserver-x11-service.service → /usr/lib/systemd/system/vncserver-x11-serviced.service.
█
```

```
Raspberry Pi Software Configuration Tool (raspi-config)

1 Change User Password      Change password for the current u
2 Network Options          Configure network settings
3 Boot Options             Configure options for start-up
4 Localisation Options     Set up language and regional sett
5 Interfacing Options      Configure connections to peripher
6 Overclock               Configure overclocking for your P
7 Advanced Options        Configure advanced settings
8 Update                  Update this tool to the latest ve
9 About raspi-config       Information about this configurat

<Select>                  <Finish>
```

In den Interface-Options enablen wir SSH (wenn noch nicht mit der Datei ssh passiert) und VNC

```
Raspberry Pi Software Configuration Tool (raspi-config)

P1 Camera                 Enable/Disable connection to the
P2 SSH                    Enable/Disable remote command lin
P3 VNC                    Enable/Disable graphical remote a
P4 SPI                    Enable/Disable automatic loading
P5 I2C                    Enable/Disable automatic loading
P6 Serial                 Enable/Disable shell and kernel m
P7 1-Wire                 Enable/Disable one-wire interface
P8 Remote GPIO            Enable/Disable remote access to G

<Select>                  <Back>
```

In den Localization gehen wir auf de-de UTF8 als Default (Change local) und setzen unter Change Timezone (Europe-Berlin[für de])

```
Raspberry Pi Software Configuration Tool (raspi-config)

I1 Change Locale          Set up language and regional sett
I2 Change Timezone       Set up timezone to match your loc
I3 Change Keyboard Layout Set the keyboard layout to match
I4 Change Wi-fi Country  Set the legal channels used in yo

<Select>                <Back>
```

```
Configuring locales

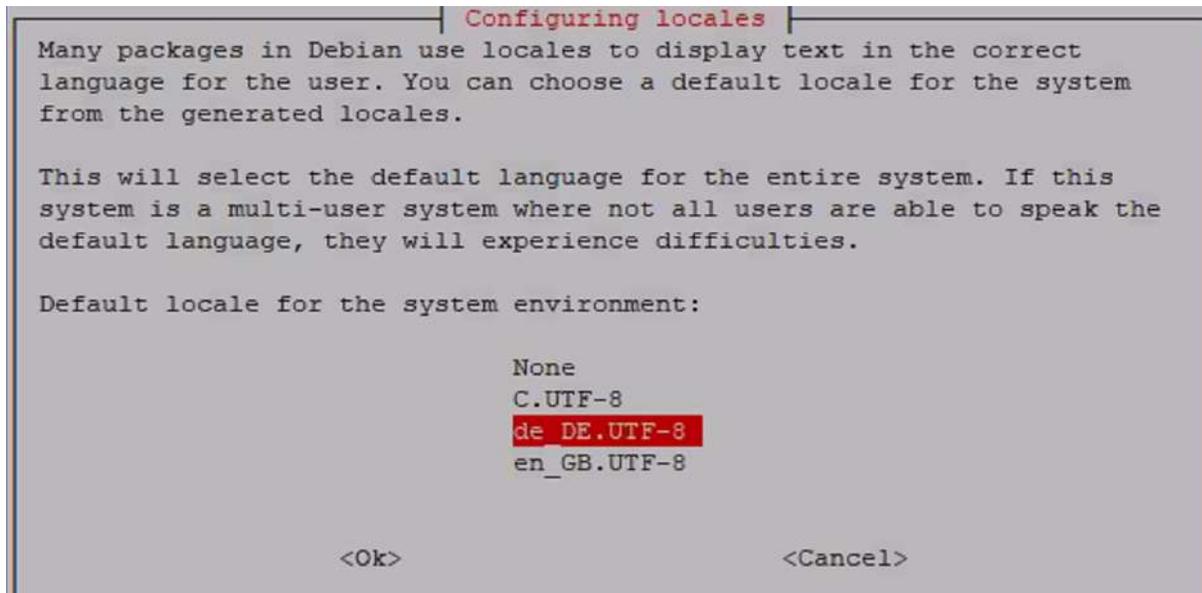
Locales are a framework to switch between multiple languages and allow
users to use their language, country, characters, collation order, etc.

Please choose which locales to generate. UTF-8 locales should be chosen
by default, particularly for new installations. Other character sets may
be useful for backwards compatibility with older systems and software.

Locales to be generated:

[ ] de_BE@euro ISO-8859-1
[ ] de_CH ISO-8859-1
[ ] de_CH.UTF-8 UTF-8
[ ] de_DE ISO-8859-1
[ ] de_DE.UTF-8 UTF-8

<Ok>                    <Cancel>
```



Das dauert dann ein bisschen:

```
pi@raspberrypi:~$ sudo raspi-config
Created symlink /etc/systemd/system/multi-user.target.wants/vncserver-x11-ser
ed.service → /usr/lib/systemd/system/vncserver-x11-serviced.service.
Generating locales (this might take a while)...
de_DE.UTF-8...
```

Dann setzen wir noch die Zeitzone:

Raspberry Pi Software Configuration Tool (raspi-config)

- I1 Change Locale Set up language and regional settings to match y
- I2 Change Timezone Set up timezone to match your location
- I3 Change Keyboard Layout Set the keyboard layout to match your keyboard
- I4 Change Wi-fi Country Set the legal channels used in your country

<Select>

<Back>

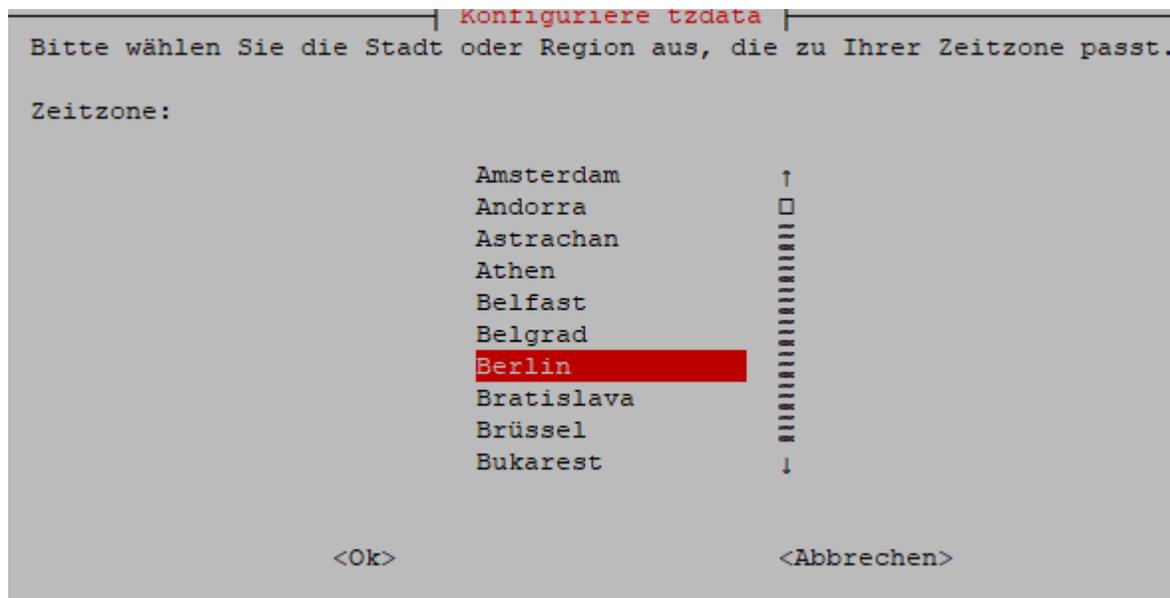
Bitte wählen Sie das geographische Gebiet aus, in dem Sie leben. Die folgenden Fragen werden dies durch eine Auswahl von Städten, die die Zeitzonen repräsentieren, in denen sie liegen, weiter einschränken.

Geographisches Gebiet:

- Afrika ↑
- Amerika ▬
- Antarktis ▬
- Australien ▬
- Arktis □
- Asien ▬
- Atlantik ▬
- Europa ↓

<Ok>

<Abbrechen>



Unter Network-Options können wir dem PI noch einen aussagefähigen Namen vergeben

```
Raspberry Pi Software Configuration Tool (raspi-config)
N1 Hostname          Set the visible name for this Pi on a network
N2 Wi-fi            Enter SSID and passphrase
N3 Network interface names Enable/Disable predictable network interface na

<Select>          <Back>
```

Jetzt rebooten wir erstmal

```
sudo reboot
```

nach erfolgtem Reboot loggen wir uns erneut mit ssh (putty) ein
mit

```
sudo nano /etc/ssh/sshd_config
```

fügen wir am Schluss

```
PermitRootLogin = yes
```

ein oder ändern ein bestehendes PermitRootLogin ab.

```
GNU nano 3.2 /etc/ssh/sshd config Verändere
AcceptEnv LANG LC_*

# override default of no subsystems
Subsystem sftp /usr/lib/openssh/sftp-server

# Example of overriding settings on a per-user basis
#Match User anoncvs
#   X11Forwarding no
#   AllowTcpForwarding no
#   PermitTTY no
#   ForceCommand cvs server

PermitRootLogin = yes

I

^G Hilfe ^O Speichern ^W Wo ist ^K Ausschneid ^J Ausrichten ^C Cursor
^X Beenden ^R Datei öffn ^\ Ersetzen ^U Ausschn. r ^T Rechtschr. ^ Zu Zeile
```

Mit CTRL-O speichern und dann mit CTRL-X beenden

Zum übernehmen restarten wir den Service

```
sudo /etc/init.d/ssh restart
```

```
pi@raspi2b:~$ sudo /etc/init.d/ssh restart
[ ok ] Restarting ssh (via systemctl): ssh.service.
```

Zum Prüfen können wir jederzeit

```
sudo /etc/init.d/ssh status
```

ausführen

```
pi@raspberry2b:~$ sudo /etc/init.d/ssh status
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: en
   Active: active (running) since Wed 2019-09-04 18:41:23 BST; 6s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Process: 810 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
  Main PID: 811 (sshd)
    Memory: 724.0K
    CGroup: /system.slice/ssh.service
            └─811 /usr/sbin/sshd -D

Sep 04 18:41:22 raspberry2b systemd[1]: Starting OpenBSD Secure Shell server...
Sep 04 18:41:23 raspberry2b sshd[811]: Server listening on 0.0.0.0 port 22.
Sep 04 18:41:23 raspberry2b sshd[811]: Server listening on :: port 22.
Sep 04 18:41:23 raspberry2b systemd[1]: Started OpenBSD Secure Shell server.
pi@raspberry2b:~$
```

Wir führen alle Updates jetzt erstmal durch

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

Das dauert schon mal ne Weile

dann installieren wir den Midnightcommander,

```
sudo apt-get install -y mc
```

RemoteDesktopunterstützung,

Ab hier sollte der RemoteDesktop (RDP) Zugriff über Windows funktionieren

```
sudo apt-get install -y xrdp
```

Samba

```
sudo apt-get install -y samba
```

und svn

```
sudo apt-get install -y subversion
```

Jetzt müssen wir Samba Passwörter für die User root und PI angelegen

```
sudo smbpasswd -a pi
```

```
sudo smbpasswd -a root
```

Im Anschluss müssen wir die smb.conf noch bearbeiten:

```
sudo nano /etc/samba/smb.conf
```

dort die workgroup durch den Namen der eigenen Workgroup ersetzen

```
GNU nano 3.2 /etc/samba/smb.conf
# commented-out examples in this file.
# - When such options are commented with ";", the proposed setting
#   differs from the default Samba behaviour
# - When commented with "#", the proposed setting is the default
#   behaviour of Samba but the option is considered important
#   enough to be mentioned here
#
# NOTE: Whenever you modify this file you should run the command
# "testparm" to check that you have not made any basic syntactic
# errors.
#
#=====  

[global]
## Browsing/Identification ###
# Change this to the workgroup/NT-domain name your Samba server will part of
workgroup = WORKGROUP
[ 236 Zeilen gelesen ]
^G Hilfe      ^O Speichern ^W Wo ist      ^X Ausschneid ^U Ausrichten ^C Cursor
^X Beenden    ^R Datei öffn ^\ Ersetzen   ^U Ausschn. r ^M Rechtschr. ^_ Zu Zeile
```

Irgendwo unter [global] hinzufügen, damit wir symlinks folgen können

```
follow symlinks = yes
unix extensions = no
wide links = yes
```

```
GNU nano 3.2 /etc/samba/smb.conf Verändert

# RPC pipe.
; add group script = /usr/sbin/addgroup --force-badname %g

##### Misc #####
follow symlinks = yes
unix extensions = no
wide links = yes

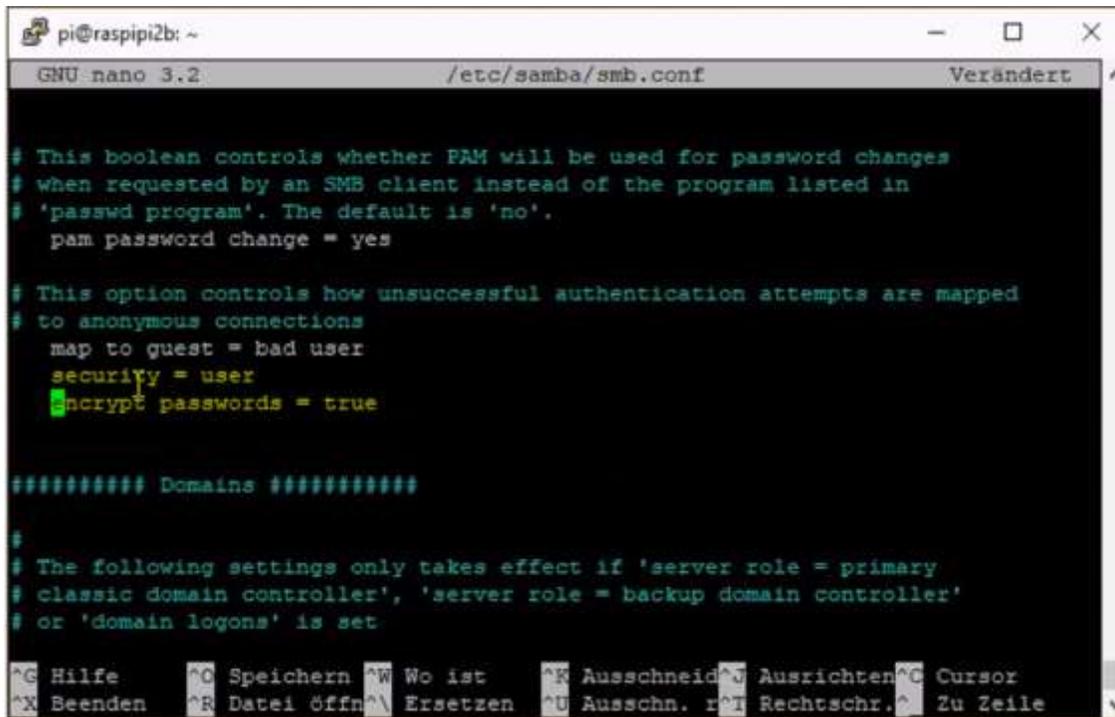
# Using the following line enables you to customise your configuration
# on a per machine basis. The %m gets replaced with the netbios name
# of the machine that is connecting
; include = /home/samba/etc/smb.conf.%m

# Some defaults for winbind (make sure you're not using the ranges
# for something else.)
; idmap config * : backend = tdb
; idmap config * : range = 3000-7999

^G Hilfe ^G Speichern ^W Wo ist ^X Ausschneid ^U Ausrichten ^C Cursor
^X Beenden ^R Datei öffn ^\ Ersetzen ^U Ausschn. r ^T Rechtschr. ^ Zu Zeile
```

Unter ##### Authentication #####

```
security = user
encrypt passwords = true
```



```
pi@raspberry2b: ~
GNU nano 3.2 /etc/samba/smb.conf  Verändert
# This boolean controls whether PAM will be used for password changes
# when requested by an SMB client instead of the program listed in
# 'passwd program'. The default is 'no'.
  pam password change = yes

# This option controls how unsuccessful authentication attempts are mapped
# to anonymous connections
  map to guest = bad user
  security = user
  encrypt passwords = true

##### Domains #####

#
# The following settings only takes effect if 'server role = primary
# classic domain controller', 'server role = backup domain controller'
# or 'domain logons' is set

^G Hilfe      ^O Speichern ^W Wo ist    ^K Ausschneid ^J Ausrichten ^C Cursor
^X Beenden    ^R Datei Öffn ^\ Ersetzen  ^U Ausschn. f ^T Rechtschr. ^_ Zu Zeile
```

hinzufügen und am Schluss der Datei

```
[rootfs]
path=/
browseable=YES
writeable=YES
valid users= pi, root
only guest=no
create mask=0777
directory mask=07777
public=no
force user = root
```

```
GNU nano 3.2 /etc/samba/smb.conf Verändere
# You may need to replace 'lpadmin' with the name of the group your
# admin users are members of.
# Please note that you also need to set appropriate Unix permissions
# to the drivers directory for these users to have write rights in it.
; write list = root, @lpadmin

[rootfs]
path=/
browseable=YES
writeable=YES
valid users= pi, root
only guest=no
create mask=0777
directory mask=0777
public=no
force user = root
█

^G Hilfe ^O Speichern ^W Wo ist ^K Ausschneid ^J Ausrichten ^C Cursor
^X Beenden ^R Datei öffn ^\ Ersetzen ^U Ausschn. r ^H Rechtschr. ^_ Zu Zeile
```

Dann speichern (CTRL-O) und beenden (CGTRL-X)

im Anschluß müssen wir die Änderungen übernehmen

```
sudo service smb restart
```

```
pi@raspberry2b:~$ sudo service smb restart
pi@raspberry2b:~$ █
```

mit

```
sudo smbstatus
```

können wir dann prüfen, ob samba läuft

```
pi@raspipi2b:~$ sudo smbstatus
Samba version 4.9.5-Debian
PID      Username      Group          Machine        Pr
ocol Version  Encryption    Machine
Signing
-----
Service  pid          Machine       Connected at   Encryption
Signing
-----
No locked files
pi@raspipi2b:~$
```

Der PI sollte im Explorer sichtbar sein und beim Klick auf rootfs eine Benutzerabfrage erfolgen.

Damit alles bequem per Explorer bearbeitet werden kann, den User root (und Notepad++ !!!) benutzen.

Oscam installieren inkl. Buildumgebung

Buildumgebung einrichten und oscam-binary erzeugen

Zuallererst installieren wir die „Entwicklungsumgebung“.

Hierfür nutzen wir simplebuild 3 RC6 von Gorgone

Wir gehen wieder per ssh auf den PI

```
cd /opt
```

```
sudo svn co https://github.com/gorgone/s3_releases.git/trunk/ s3
```

oder

```
sudo git clone https://github.com/gorgone/s3_releases.git s3
```

```
pi@raspbpi2b:~$ sudo svn co https://github.com/gorgone/s3_releases.git/trunk
/ s3
A    s3/LICENSE
A    s3/README.md
A    s3/s3
A    s3/support
A    s3/support/configs
A    s3/support/configs/compiler_option
A    s3/support/configs/compiler_option.txt
A    s3/support/configs/config.sh.master
A    s3/support/configs/max_cpus
```

```
cd s3
```

für den ersten Lauf mit Dependency Updates:

```
sudo ./s3
```

Hier wird nun das System gecheckt, ob alles vorhanden ist

```
CHECK for binaries
-----
need  dialog          (not found)
have  grep             /bin/grep
need  gawk             (not found)
have  wget            /usr/bin/wget
have  tar             /bin/tar
have  bzip2           /bin/bzip2
have  svn             /usr/bin/svn
have  xz              /usr/bin/xz
need  upx             (not found)
have  patch           /usr/bin/patch
have  gcc             /usr/bin/gcc
```

Und fehlende Packages nachinstalliert

```
INSTALLATION
-----
update package list...

OK:1 http://raspbian.raspberrypi.org/raspbian buster InRelease
OK:2 http://archive.raspberrypi.org/debian buster InRelease
Paketlisten werden gelesen...

install packages bitte warten...
```

Nach Abschluss der Installation Enter drücken:

```

-----
beginnen mit ./s3 menu
Kommandozeilen Parameter sind
-----

toolchains :
-----
arm_dream cool cool2 dockstar dreambox_fpu
dream_one fritz73xxOS65 fritz74xxOS65
hd51 ineos mca mipsel mipsel_s3_ssl098
mipsel_s3_ssl100 mipsel_s3_ssl102 mipsoe20
native openpli40 openwrt1505cc osmio4k
ppcold rasb_hard samsungtv sh_4 sh4
solo4k syn_atom
simplebuild options :
-----
help credit clean enable_emu fix_me
tccheck tcrepair upload update_me
update_emu lang_select loadonly checkout
ssh_profiles syscheck sysinfo svnup
menu get_patch tedit svnpatch profiles
svnrestore update
Taste [ENTER] zum fortsetzen...

```

Patch einbauen (es reicht das kopieren in das patchverzeichnis -> wird automatisch von Simplebuild angewandt):

```
cd /opt/s3/patches
```

Zur Sicherheit für später

```
sudo rm *.patch
```

```
sudo wget <https:// .....>
```

```
cd /opt/s3
```

anschließend zum konfigurieren und bauen:

```
sudo ./s3 menu
```

```
SIMPLEBUILD_V(3.0 RC6)
```

```
-[ Hauptmenü ]-
```

```
Simplebuild V3.0 RC6 by Gorgone
```

```
Continue weiter mit lokalem SVN r11546  
CHANGE anderen SVN Stand (7000+) laden  
CEDIT Konfigurations Editor starten  
LANG Systemsprache manuell auswählen  
EXIT simplebuild sofort beenden
```

```
< OK >
```

ACHTUNG: nach dem erstmaligen Download muss man in Zukunft über Change gehen, um die aktuellste Version zu bekommen !!

```
-[ Toolchain Menü ]-  
  
Die aktuelle SVN REVISION ist : r11546  
  
BACK      zurück zum Hauptmenü  
native    Nativ für raspip12b-armv6l  
EXIT      simplebuild sofort beenden  
  
< OK >      < INFO >
```

```
-[ Baumenü ]-  
  
-----  
Benutzername      = root  
Toolchain          = native  
Compiler           = arm-linux-gnueabi-hf-gcc  
Debug Info        = CPU-Threads(1) SVN(11546) SCRIPT(3.0 RC6)  
USE_Variablen     = UTF8 LIBCRYPTO  
  
-----  
BUILD             oscam jetzt bauen  
ONFIGURE          oscam/Build Konfiguration  
LOAD_PROFILE      vorhandenes Profil laden  
SAVE_PROFILE      Profil speichern  
SHOW_BUILDLOG     zeige lastbuild.log  
EDIT_CONF_DIR     Oscam config PATH  
BACK              zurück zur Toolchainauswahl  
EXIT              simplebuild sofort beenden  
  
< OK >
```

```

-[ Konfiguration Menü ]-
ADDONS :
WEBIF LIVELOG JQuery DVBAPI READ_SDT IRDETO_GUESSING DEBUG MONITOR LB

PROTOCOLS :
CAMD35 CAMD35_TCP NEWCAMD CCCAM CCCSHARE GBOX SCAM

READERS :
NAGRA NAGRA_MERLIN IRDETO CONAX CRYPTOWORKS SECA VIACCESS VIDEOGUARD DRE
TONGFANG BULCRYPT GRIFFIN DGCRYPT

CARD_READERS :
PHOENIX INTERNAL STINGER

USE VARIABLES :
UTF8 LIBCRYPTO

BACK          zurück zum Baumenü
OSCAM_MODULE oscam Module auswählen
OSCAM_EXTRA  oscam Extras USB/PCSC auswählen
BUILD_EXTRA  build Extras compress/patch auswählen
SELECT_STAPI select stapi
RESET        Konfiguration zurück setzen

100%
< Ok >
```

Hier dann die Addons wie SSL etc. aus- oder abwählen

Unter Build extra

```

BACK          zurück zum Baumenü
OSCAM_MODULE oscam Module auswählen
OSCAM_EXTRA  oscam Extras USB/PCSC auswählen
BUILD_EXTRA  build Extras compress/patch auswählen
SELECT_STAPI select stapi
RESET        Konfiguration zurück setzen
```

Noch auswählen, daß der Patch auch genutzt wird.

```
-[ Build EXTRA ]-
OPTIONS
[ ] USE_COMPRESS  USE_COMPRESS=1
[ ] USE_TARGZ      USE_TARGZ=1
[*] USE_PATCH      USE_PATCH=1

< OK >
```

Er ist dann noch nicht unter addons sichtbar aber er wird beim Build angewandt.

Nach dem ersten Build ist er auch unter Addons am Schluß zu sehen.

Geht man dann zurück müssen wir unbedingt das Config-verzeichnis setzen

```
-[ Baumenü ]-
-----
Benutzername   = root
Toolchain      = native
Compiler       = arm-linux-gnueabi-hf-gcc
Debug Info     = CPU-Threads(4) SVN(11539) SCRIPT(3.0 RC6)
USE_Variablen  = LIBUSB UTF8 LIBCRYPTO SSL PATCH CONFDIR
-----
BUILD          oscam jetzt bauen
CONFIGURE      oscam/Build Konfiguration
LOAD_PROFILE   vorhandenes Profil laden
SAVE_PROFILE   Profil speichern
SHOW_BUILDLOG  zeige lastbuild.log
EDIT_CONF_DIR  Oscam config PATH
BACK           zurück zur Toolchainauswahl
EXIT          simplebuild sofort beenden

< OK >
```

```
-[ Toolchain ]-  
  
CONF_DIR=/usr/local/etc 'default'  
[ESC] = default  
  
/usr/local/etc  
  
< OK > <Abbrechen>
```

Dann können wir jetzt oscam bauen

```
-[ Baumenü ]-  
  
Benutzername = root  
Toolchain = native  
Compiler = arm-linux-gnueabi-gcc  
Debug Info = CPU-Threads(4) SVN(11539) SCRIPT(3.0 RC6)  
USE_Variablen = LIBUSB UTF8 LIBCRYPTO SSL PATCH CONFDIR  
  
BUILD ocam jetzt bauen  
CONFIGURE ocam/Build Konfiguration  
LOAD_PROFILE vorhandenes Profil laden  
SAVE_PROFILE Profil speichern  
SHOW_BUILDLOG zeige lastbuild.log  
EDIT_CONF_DIR Oscam config PATH  
BACK zurück zur Toolchainauswahl  
EXIT simplebuild sofort beenden  
  
< OK >
```

die Binaries liegen dann unter /opt/s3/binaries

```
ls /opt/s3/binaries/
```

Oscam-Server einrichten

Die Oscam-Binary nach /usr/local/bin kopieren (und dabei gleich nach oscam umbenennen):

```
sudo cp /opt/s3/binaries/oscam-svnXXXXXX-ZZZZ /usr/local/bin/oscam
```

Kontrolle:

```
ls /usr/local/bin/
```

```
pi@raspbpi2b:~/opt/s3 $ ls /usr/local/bin/
oscam
pi@raspbpi2b:~/opt/s3 $ █
```

Damit sie auch beim Start geladen wird

```
sudo nano /etc/systemd/system/oscam.service
```

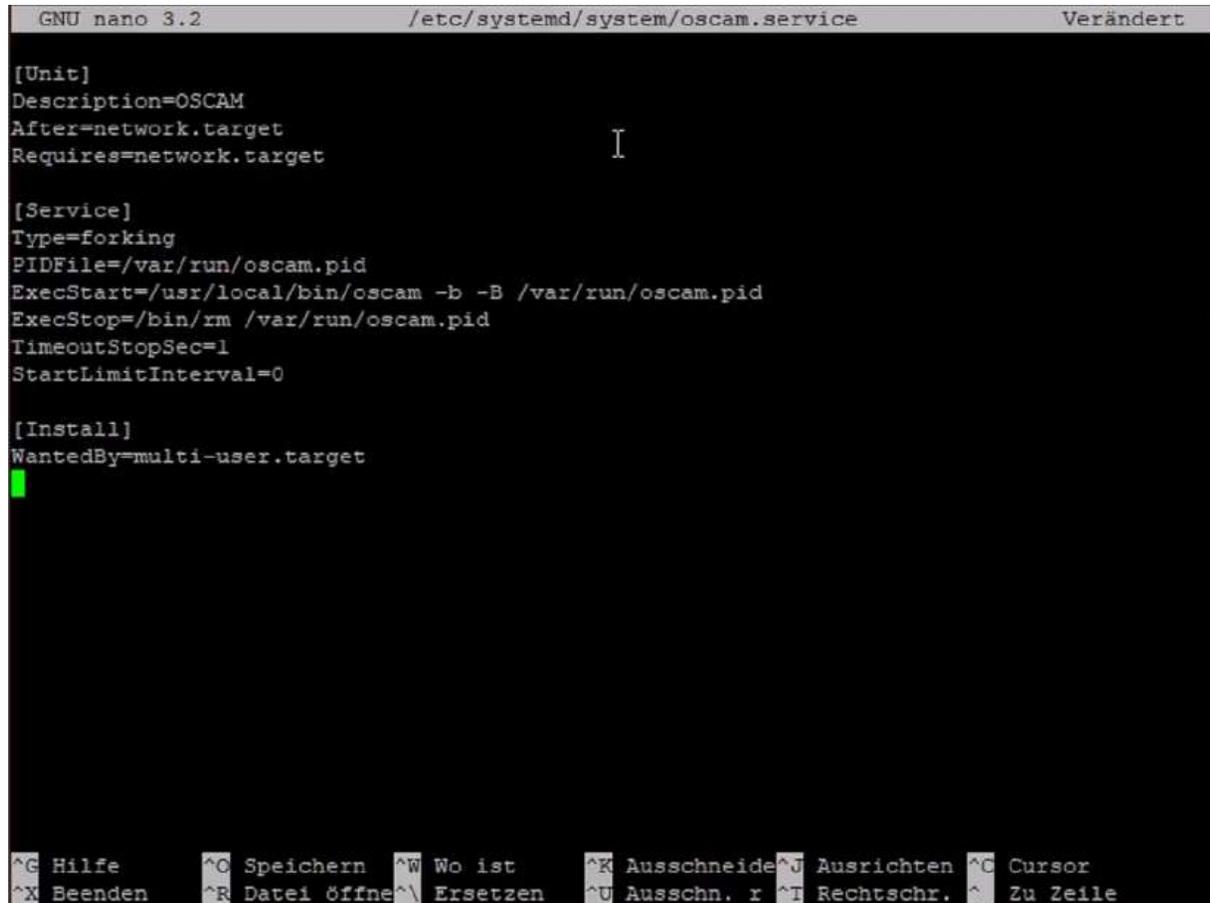
das hier einfügen

```
[Unit]
Description=OSCAM
After=network.target
Requires=network.target

[Service]
Type=forking
PIDFile=/var/run/oscam.pid
```

```
ExecStart=/usr/local/bin/oscam -b -B /var/run/oscam.pid
ExecStop=/bin/rm /var/run/oscam.pid
TimeoutStopSec=1
StartLimitInterval=0

[Install]
WantedBy=multi-user.target
```



```
GNU nano 3.2 /etc/systemd/system/oscam.service Verändert

[Unit]
Description=OSCAM
After=network.target
Requires=network.target

[Service]
Type=forking
PIDFile=/var/run/oscam.pid
ExecStart=/usr/local/bin/oscam -b -B /var/run/oscam.pid
ExecStop=/bin/rm /var/run/oscam.pid
TimeoutStopSec=1
StartLimitInterval=0

[Install]
WantedBy=multi-user.target

^G Hilfe      ^O Speichern  ^W Wo ist    ^K Ausschneide ^J Ausrichten  ^C Cursor
^X Beenden    ^R Datei öffne ^\ Ersetzen   ^U Ausschn. r  ^I Rechtschr. ^_ Zu Zeile
```

CTRL-O zum speichern und mit CTRL-X beenden

Mit

```
systemctl enable oscam.service
```

dem System bekannt machen.

Die Legitimierungen beantworten wir mit 1 (user PI)

```
pi@raspbpi2b:/opt/pi $ systemctl enable oscam.service
===== AUTHENTICATING FOR org.freedesktop.systemd1.manage-unit-files =====
Legitimierung ist notwendig für die Verwaltung von Systemdiensten und Einheitendateien.
Multiple identities can be used for authentication:
 1. ,, (pi)
 2. root
Choose identity to authenticate as (1-2): 1
Password:
===== AUTHENTICATION COMPLETE =====
Created symlink /etc/systemd/system/multi-user.target.wants/oscam.service -> /etc/systemd/system/oscam.service.
===== AUTHENTICATING FOR org.freedesktop.systemd1.reload-daemon =====
Legitimierung ist zum erneuten Laden des systemd-Zustands notwendig.
Multiple identities can be used for authentication:
 1. ,, (pi)
 2. root
Choose identity to authenticate as (1-2): 1
Password:
===== AUTHENTICATION COMPLETE =====
pi@raspbpi2b:/opt/pi $ █
```

Starten mit

```
systemctl start oscam.service
```

```
pi@raspberry2b:/opt/v3 $ systemctl start oscam.service
===== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units =====
Legitimierung ist zum Starten von »oscam.service« notwendig.
Multiple identities can be used for authentication:
 1. ,, (pi)
 2. root
Choose identity to authenticate as (1-2): 1
Password:
===== AUTHENTICATION COMPLETE =====
pi@raspberry2b:/opt/v3 $
```

Prüfen mit

```
systemctl status oscam.service
```

```
pi@raspberry2b:/opt/v3 $ systemctl status oscam.service
● oscam.service - OSCAM
   Loaded: loaded (/etc/systemd/system/oscam.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2019-09-04 20:52:17 CEST; 8s ago
     Process: 18137 ExecStart=/usr/local/bin/oscam -b -B /var/run/oscam.pid (code=exited, s
   Main PID: 18139 (oscam)
     Memory: 1.8M
     CGroup: /system.slice/oscam.service
            └─18138 /usr/local/bin/oscam -b -B /var/run/oscam.pid
              └─18139 /usr/local/bin/oscam -b -B /var/run/oscam.pid

Sep 04 20:52:17 raspberrypi2b systemd[1]: Starting OSCAM...
Sep 04 20:52:17 raspberrypi2b systemd[1]: oscam.service: Can't open PID file /run/oscam.pid
Sep 04 20:52:17 raspberrypi2b systemd[1]: oscam.service: Supervising process 18139 which is
Sep 04 20:52:17 raspberrypi2b systemd[1]: Started OSCAM.
lines 1-14/14 (END)
pi@raspberry2b:/opt/v3 $
```

Mit CTRL-C abbrechen

wenn jetzt alles passt kann man das OSCAM-Web IF schon aufrufen (Port 8888)

Dann starten wir mal den PI neu

```
sudo reboot
```

Der OSCAM-WEBIF PORT sollte jetzt auf Port **8888** zu erreichen sein (Standard Oscam)

Nicht nötig, aber für den einfachen Zugriff können wir noch die Verzeichnisse der OSCAM Installation uns im Explorer einfach anzeigen lassen:

```
sudo nano /etc/samba/smb.conf
```

Am Schluss

```
[OSCAM_CONF]
path=/usr/local/etc
browseable=YES
writeable=YES
valid users= pi, root
only guest=no
create mask=0777
directory mask=07777
public=no
force user = root

[OSCAM_BIN]
path=/usr/local/bin
browseable=YES
writeable=YES
valid users= pi, root
only guest=no
create mask=0777
directory mask=07777
public=no
force user = root
```

einfügen

```
GNU nano 3.2 /etc/samba/smb.conf Verändert
create mask=0777
directory mask=0777
public=no
force user = root

[OSCAM_CONF]
path=/usr/local/etc
browseable=YES
writeable=YES
valid users= pi, root
only guest=no
create mask=0777
directory mask=0777
public=no
force user = root

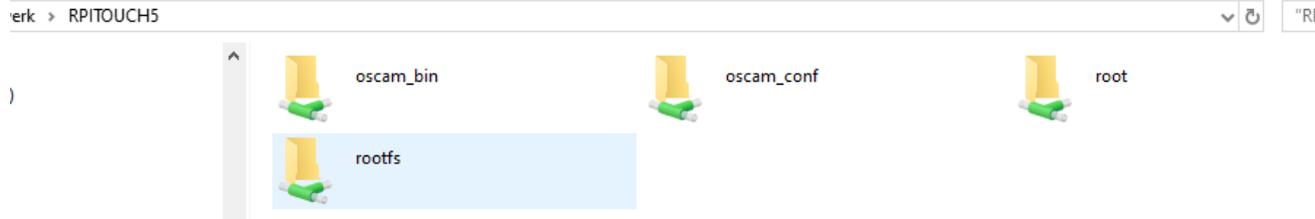
[OSCAM_BIN]
path=/usr/local/bin
browseable=YES
writeable=YES
valid users= pi, root
only guest=no
create mask=0777
directory mask=0777
public=no
force user = root
█

^G Hilfe      ^O Speichern  ^W Wo ist    ^K Aussch.  ^J Ausricht.  ^C Cursor
^X Beenden    ^R Datei öfne ^\ Ersetzen  ^U Ausschn. r ^I Rechtschr. ^_ Zu Zeile
```

Mit CTRL-O speichern und CTRL-X beenden

```
sudo service smbd restart
```

jetzt haben wir Schnellzugriff auf die Config via Windows Explorer :



OSCAM konfigurieren

Wenn die Easymouse oder ein anderer Smartreader per USB verbunden wurde, schauen wir uns, an welchen Port.

Das geht mit:

um eine eindeutige dauerhafte Zuordnung zu dem angeschlossenen Smartreader zu bekommen (gerade wenn mehrere angeschlossen werden), benutzen wir die UDEV-Regeln :

```
ls -l /dev/ttyUSB*
```

dann sollte als Ausgabe sowas kommen:

```
pi@raspberry2b:~$ ls -l /dev/ttyUSB*
crw-rw---- 1 root dialout 188, 0 Sep  4 20:53 /dev/ttyUSB0
pi@raspberry2b:~$
```

dann folgendes eingeben:

```
sudo udevadm info --name=/dev/ttyUSB0 --attribute-walk
```

in der folgenden Ausgabe sucht ihr nach den rot markierten Werten (Auszug von meiner Easymouse) und schreibt sie auf oder kopiert sie:

```
looking at device '/devices/platform/soc/3f980000.usb/usb1/1-1/1-1.1/1-1.1.3/1-1.1.3:1.0/ttyUSB0/tty/ttyUSB0':
KERNEL=="ttyUSB0"
SUBSYSTEM=="tty"
DRIVER=="

[....]

looking at parent device '/devices/platform/soc/3f980000.usb/usb1/1-1/1-1.1/1-1.1.3':
KERNELS=="1-1.1.3"
SUBSYSTEMS=="usb"
DRIVERS=="usb"
ATTRS{bMaxPower}=="90mA"
ATTRS{manufacturer}=="FTDI"
ATTRS{bDeviceProtocol}=="00"
ATTRS{version}==" 2.00"
ATTRS{idVendor}=="0403"
ATTRS{tx_lanes}=="1"
ATTRS{bConfigurationValue}=="1"
ATTRS{bDeviceSubClass}=="00"
ATTRS{avoid_reset_quirk}=="0"
ATTRS{busnum}=="1"
ATTRS{devspec}==" (null) "
ATTRS{serial}=="AD02E90Y"
ATTRS{bmAttributes}=="a0"
ATTRS{ltm_capable}=="no"
ATTRS{speed}=="12"
ATTRS{removable}=="removable"
ATTRS{maxchild}=="0"
ATTRS{devpath}=="1.1.3"
ATTRS{bcdDevice}=="0600"
ATTRS{idProduct}=="6001"
ATTRS{bMaxPacketSize0}=="8"
```

Nun die Regel festlegen

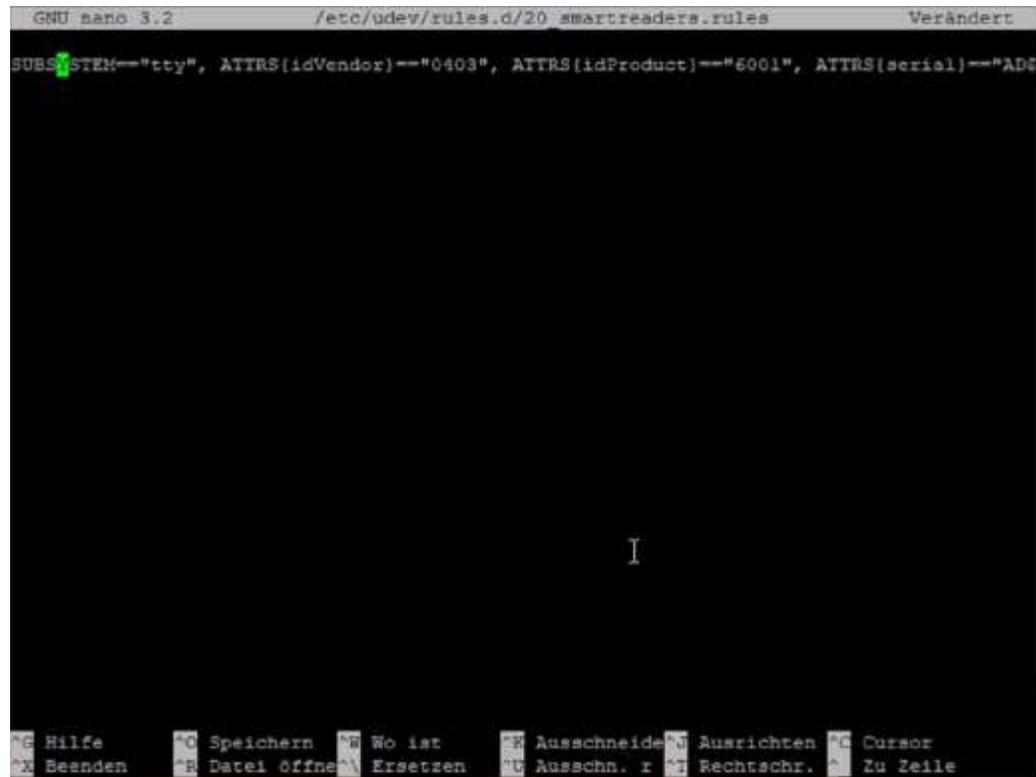
```
sudo nano /etc/udev/rules.d/20_smartreaders.rules
```

nun tragt ihr dort folgendes ein (das rote entsprechend durch eure Werte ersetzen) :

```
SUBSYSTEM=="tty", ATTRS{idVendor}=="<SUBSYSTEM >", ATTRS{idProduct}=="<ATTRS{idProduct}>",  
ATTRS{serial}=="<ATTRS{serial}>", SYMLINK+="<DEINEABKÜRZUNG>"
```

Beispiel bei mir (als Symlink nutze ich den Dienst, bzw. Karte die ich da benutze) :

```
SUBSYSTEM=="tty", ATTRS{idVendor}=="0403", ATTRS{idProduct}=="6001", ATTRS{serial}=="AD02E90Y",  
SYMLINK+="ttyUSB_HD02"
```



The screenshot shows the GNU nano 3.2 text editor editing the file /etc/udev/rules.d/20_smartreaders.rules. The content of the file is: SUBSYSTEM=="tty", ATTRS{idVendor}=="0403", ATTRS{idProduct}=="6001", ATTRS{serial}=="AD02E90Y", SYMLINK+="ttyUSB_HD02". The editor interface includes a status bar at the bottom with various keyboard shortcuts for actions like 'Hilfe', 'Speichern', 'Wo ist', 'Ausschneiden', 'Ausrichten', 'Cursor', 'Beenden', 'Datei öffnen', 'Ersetzen', 'Ausschn. r', 'Rechtschr.', and 'Zu Zeile'.

Das ganze für alle anderen Reader die ihr angeschlossen habt.

Am ende startet Ihr dann noch euren Raspberry Pi ein mal neu und dann sollte das ganze schon fertig sein.

```
sudo reboot
```

mit einem

```
ls -l /dev/ttyUSB*
```

überprüfen, sieht dann so aus:

```
pi@raspberry2b:~$ ls -l /dev/ttyUSB*
crw-rw---- 1 root dialout 188, 0 Sep  4 20:58 /dev/ttyUSB0
lrwxrwxrwx 1 root root      7 Sep  4 20:58 /dev/ttyUSB_HD02 -> ttyUSB0
pi@raspberry2b:~$
```

Wenn Ihr jetzt den Reader in euer OScam einbinden möchtet, so muss dies in der OScam-Server so aussehen (Beispiel HD02)

```
[reader]
label                = HD02
description           = Fuer HD+
protocol              = mouse
device               = /dev/ttyUSB_HD02
#device              = /dev/ttyUSB0
caid                  = 1830,1843
boxkey                = A7...7C
rsakey                = BF..D5
# Keys selbst suchen
detect                = cd
group                 = 1
emmcache              = 1,3,2,0
blockemm-unknown     = 1
blockemm-g            = 1
auprovid              = 003411
```

OSCAM updaten

Wenn bereits ein lokale SVN-Kopie von OSCAM existiert muss das erste upgedated werden:

```
cd /opt/s3/patches
```

```
sudo rm ,.patch
```

```
sudo wget <patchlink>
```

```
cd /opt/s3
```

```
sudo ./s3 menu
```

und dann dort

CHANGE anderen SVN-Stand benutzen auswählen (sollte dann die aktuellste angezeigt werden)

und dann erst im Menü continue -> native -> Build

Anschließend OSCAM stoppn

```
sudo systemctl stop oscam.service
```

Die Oscam-Binary nach /usr/local/bin kopieren:

```
sudo cp /opt/s3/binaries/oscam-svnXXXXXX-ZZZZ /usr/local/bin/oscam
```

und OSCAM wieder starten

```
sudo systemctl start oscam.service
```