

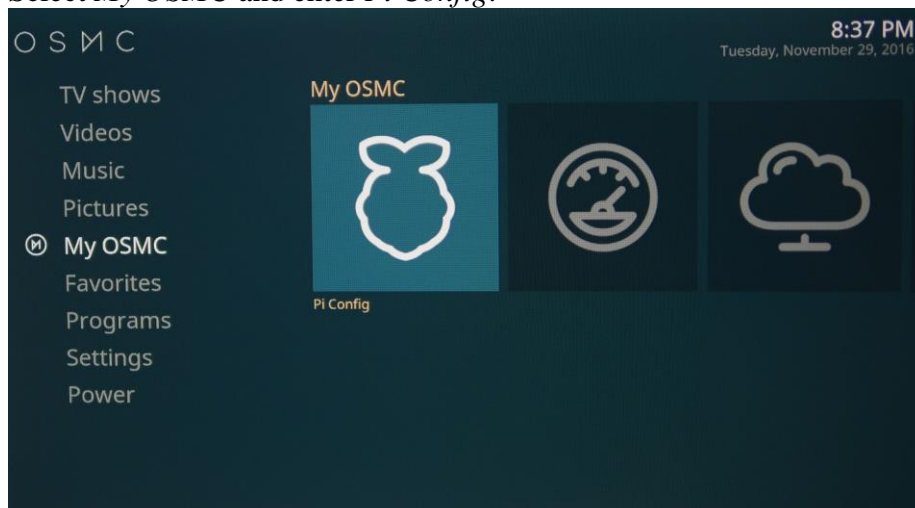
Jukebox Installation

1. Installation Betriebssystem OSMC
2. Image aus Ordner mit Win32Diskimager auf die SD schreiben
3. SD Karte in Pi stecken und Maus Tastatur Bildschirm und Netzteil anschließen.
4. Beim ersten Starten kommt danach automatisch eine Installation und danach wird man durch ein Einstellungsmenü geführt dies nun auf sich anpassen. Wenn gefragt wird ob SSH aktiviert werden soll hier das Kästchen aktivieren also hell machen.
5. Unter Einstellungen unten von Standard auf Expert einstellen, dann auf Systeminformationen die IP-Adresse ablesen, da wir diese für Putty brauchen.
6. Nun mit Benutzer osmc und Passwort osmc einloggen.
7. Einmal updaten `sudo apt-get update`
8. Jetzt HiFiBerry miniAMP installieren

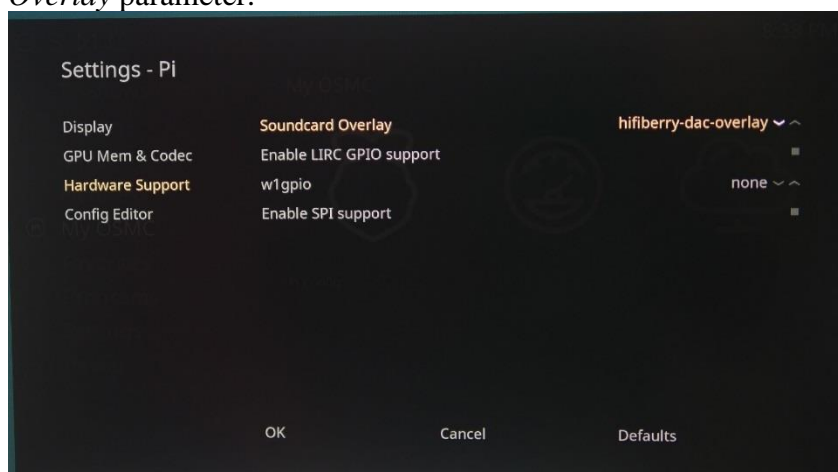
Enabling HiFiBerry sound card in OSMC

OSMC supports HiFiBerry boards out of the box. All it takes is a few tweaks to enable it.

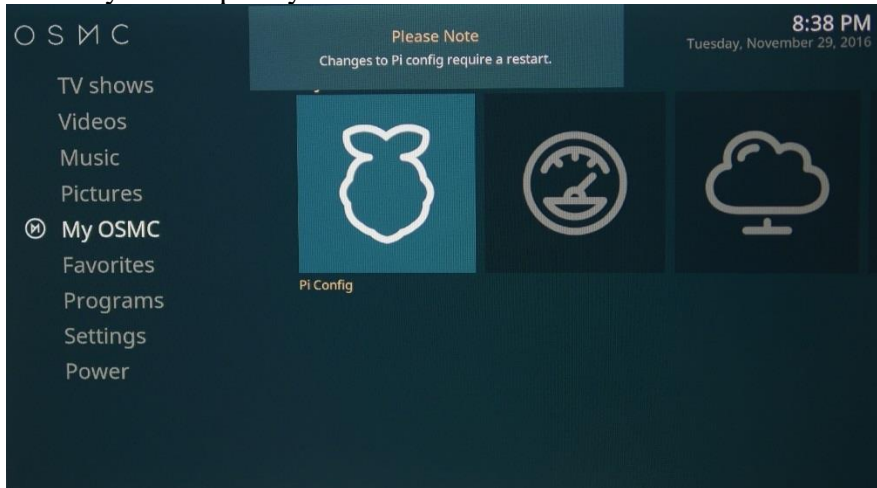
1. Select *My OSMC* and enter *Pi Config*.



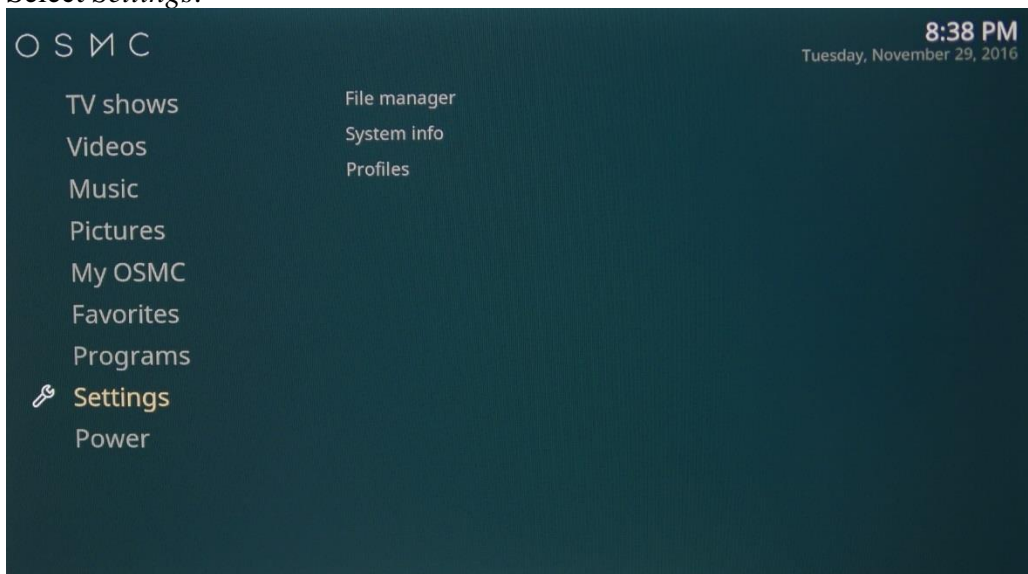
2. Select *Hardware Support* and choose your HiFiBerry in the *Soundcard Overlay* parameter.



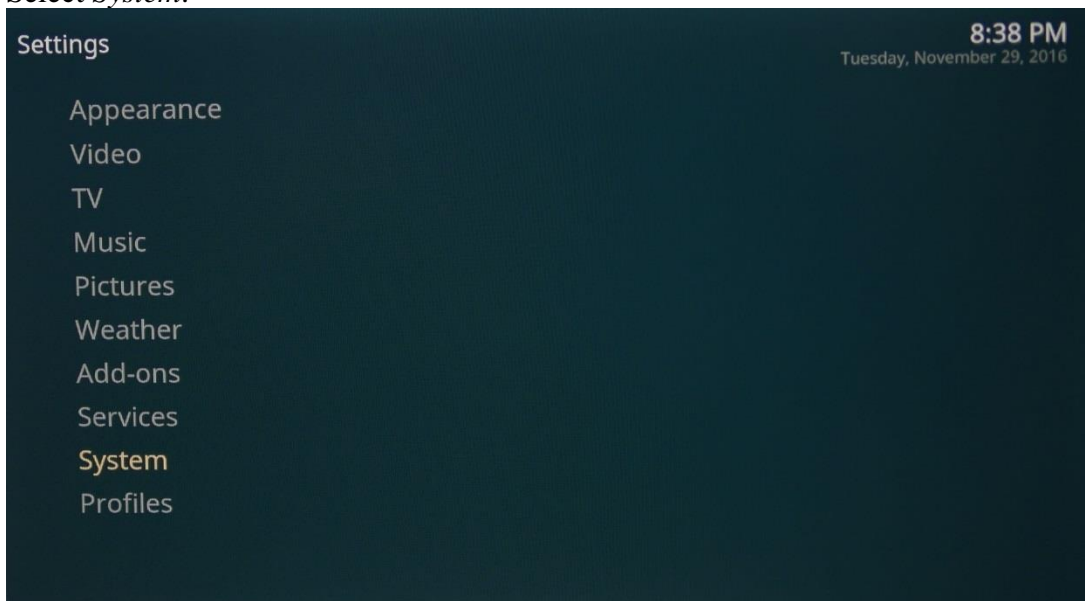
3. Restart your Raspberry Pi.



4. Select *Settings*.

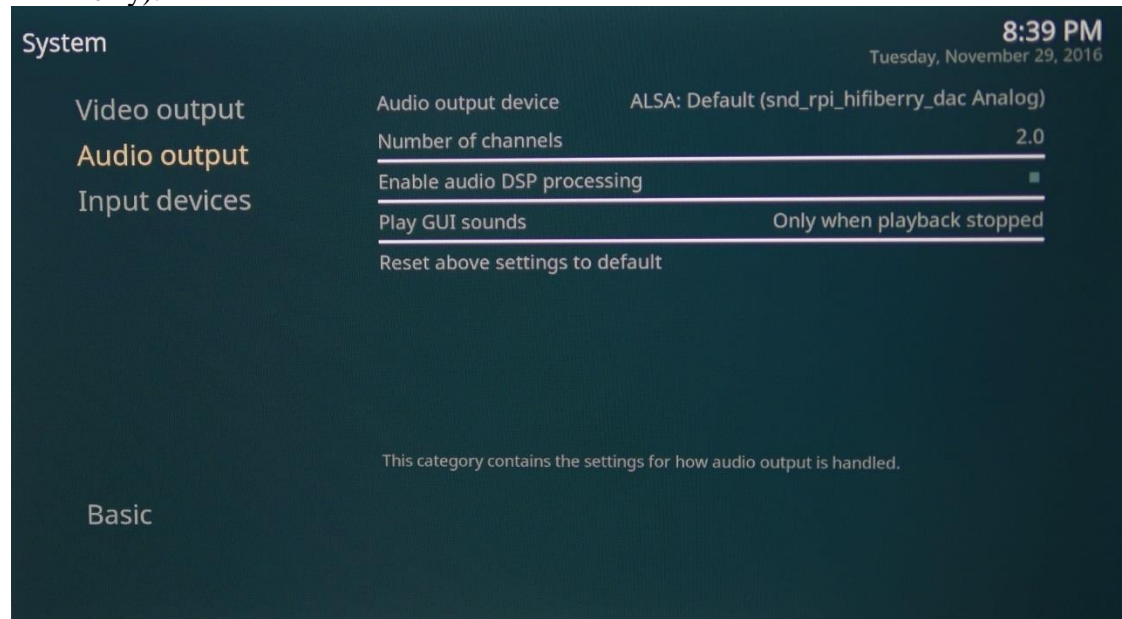


5. Select *System*.



6. Select *Audio output* and set the Audio output device parameter to ALSA: Default (snd_rpi_hifiberry_dac Analog) (or something similar, depending on your

HiFiBerry).



7. Enjoy!

Kopiert von hier <https://www.hifiberry.com/build/guides/enabling-hifiberries-in-osmc/>

Weiter folgende eingaben über putty tätigen.

1. Standard Zeile löschen

```
sudo nano /boot/config.txt
```

Jetzt nach dtparam=audio=on suchen und löschen. Ist ganz unten.

Nun folgende Zeile hinzufügen:

```
dtoverlay=hifiberry-dac
```

Jetzt nano mit STRG+X beenden. Datei aber speichern.

2. ALSA Konfigurieren

```
sudo nano /etc/asound.conf
```

Folgendes einfügen

```
pcm.!default {
    type hw card 0
}
ctl.!default {
    type hw card 0
}
```

Und dies im putty ausführen

```
sudo apt-get install -y curl wget binutils alsa-utils
```

3. Jetzt neustarten

```
Sudo reboot
```

9. USB Installieren

```
sudo apt-get install usbutils
```

10. Python installieren

```
sudo apt-get install python-dev
```

```
sudo apt-get install python-rpi.gpio
```

Ich habe diesen nochmal hinterher geschickt, da der zweite einen Fehler ausspuckte

```
sudo apt-get install python-dev python-pip gcc
```

```
sudo apt-get install gcc
```

```
sudo apt-get install python-pip
```

```
sudo pip install RPi.GPIO
```

11. Python dev Tools

```
sudo sed -i '/_hashlib.so/d' /var/lib/dpkg/info/python2.7.list
```

```
sudo sed -i '/_ssl.so/d' /var/lib/dpkg/info/python2.7.list
```

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

```
sudo apt-get install python-dev python-setuptools python-pip build-essential evtest
```

```
sudo pip install evdev
```

12. Github Daten von Fuzzwah kopieren

Erste Vezeichnis wechseln

```
cd /home/osmc/
```

```
wget https://github.com/Fuzzwah/xbmc-rfid-music/archive/master.zip
```

und entpacken

```
unzip master
```

13. Jetzt habe wir folgende Verzeichnisstruktur, welche wir auch unter Samba freigeben werden.

```
/home/osmc/xbmc-rfid-music/
```

Wir geben aber einfach das ganze "home" Verzeichnis frei.

```
sudo chmod 770 /home/
```

Jetzt tragen wir die neu angelegten Verzeichnisse ein. Dazu öffnen wir mit nano die smb.conf Datei

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

```
[global]
```

```
workgroup = WORKGROUP
```

```
security = user
```

```
encrypt passwords = yes
```

```
[jukebox]
```

```
comment= Jukebox
```

```
path=/home/
```

```
browseable=Yes
```

```
writable=Yes
```

```
only guest=no
```

```
create mask=0777
```

```
directory mask=0777
```

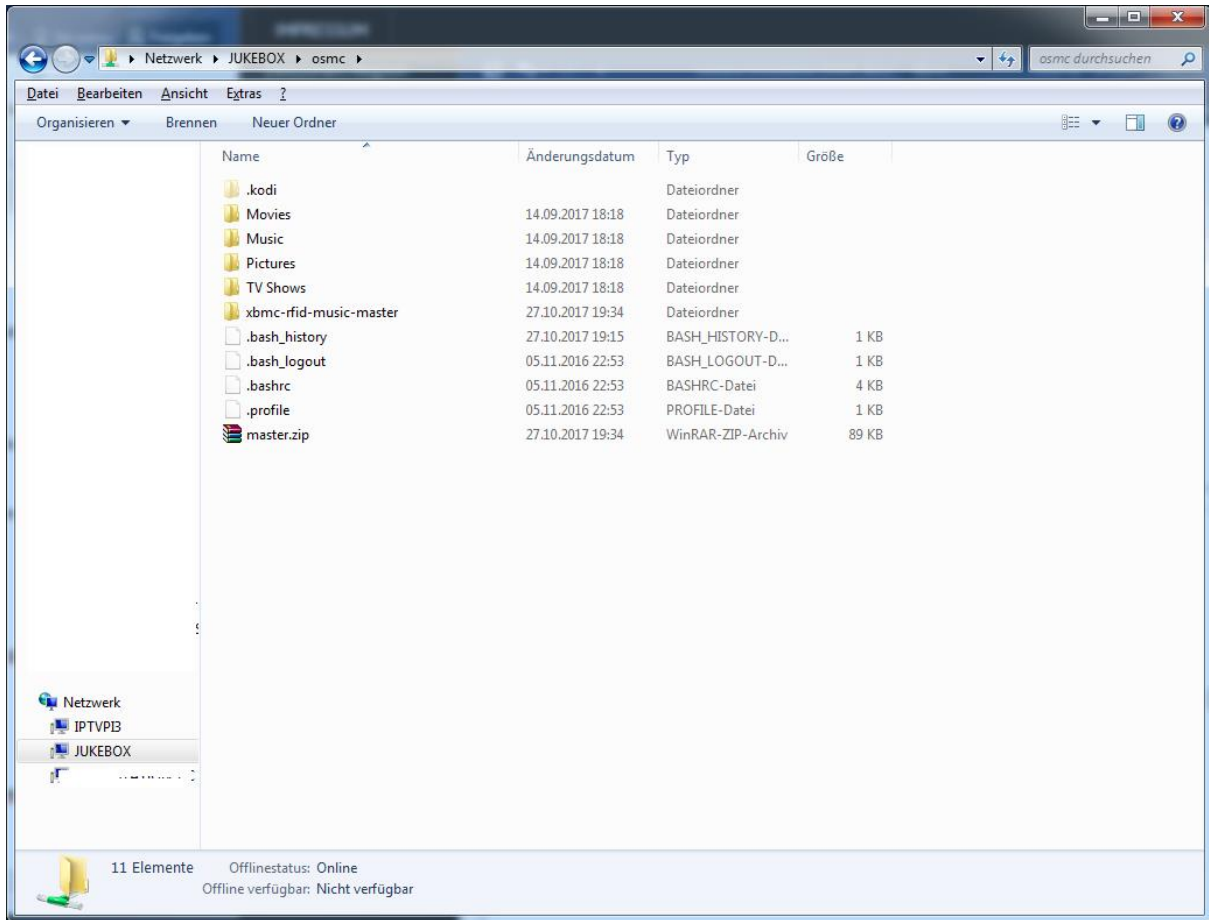
```
public=no
```

14. Installieren von SAMBA, damit man mit PC auf den Pi zugreifen kann.

Hierzu klicken wir auf dem Bildschirm mit der Maus im Startmenü auf „My OSMC“, Einkaufswagen, dann Samba Server wählen und installieren. Danach unten rechts auf apply. Danach wird installiert. Dann neustarten, dies kann auch über das Bedienmenü am Bildschirm gemacht werden.

Jetzt tauch der Pi auch am Windows PC unter Netzwerkumgebung auf oder ihr tippt in der Adresszeile bei Arbeitsplatz/Computer

\\IP-Adresse des Pis



15. Installation RFID Leser

Gerät per USB anschließen. Es sollte dann ein „beep“ kommen. Dieser wird immer wieder kommen beim Einstecken und auch wenn ein Chip über das Gerät gezogen wird.

Jetzt muss die LED Lampe am Reader aufleuchten und wenn ein Chip oder eine Karte davor gehalten wird muss es „beep“ machen.

Ein paar Infos über Geräte und wie diese heißen welche am Pi sind gibt uns der Befehl

```
ls -la /dev/input/by-id/
```

```
drwxr-xr-x 2 root root 140 Oct 27 19:42 .
drwxr-xr-x 4 root root 200 Oct 27 19:42 ..
lrwxrwxrwx 1 root root 9 Oct 27 19:42 usb-RXGCoLtd_HIDKeys-event-kbd -> ../event3
lrwxrwxrwx 1 root root 9 Oct 27 19:42 usb-Logitech_USB_Keyboard-event-if01 -> ../event1
lrwxrwxrwx 1 root root 9 Oct 27 19:42 usb-Logitech_USB_Keyboard-event-kbd -> ../event0
lrwxrwxrwx 1 root root 9 Oct 27 19:42 usb-Logitech_USB_Optical_Mouse-event-mouse -> ../event2
lrwxrwxrwx 1 root root 9 Oct 27 19:42 usb-Logitech_USB_Optical_Mouse-mouse -> ../mouse0
osmc@jukebox:~$
```

Dann installieren wir noch folgendes

```
sudo apt-get install lusb
```

```
sudo apt-get update
```

```
sudo apt-get install usbutils
```

```
sudo apt-get install usbutils
```

Jetzt diesen Befehl eingeben, damit wir dem Reader einen User zuschreiben.

```
sudo lsusb
```

```
osmc@jukebox:~$ sudo lsusb
Bus 001 Device 006: ID 16c0:27db Van Ooijen Technische Informatica Keyboard
Bus 001 Device 005: ID 046d:c077 Logitech, Inc.
Bus 001 Device 004: ID 046d:c328 Logitech, Inc.
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp. SMSC9512/9514 Fast Ethernet Adapter
Bus 001 Device 002: ID 0424:9514 Standard Microsystems Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

1-3 sind USB Ports Hub und Netzwerkkarte+Wifi vom Pi

4-5 sind Maus und Tastatur

6 müsste mein RFID Reader sein.

Wir brauchen diese ID von dem Gerät **16c0:27db** gleich.

```
sudo nano /etc/udev/rules.d/80-rfid.rules
```

folgendes eintragen

```
SUBSYSTEMS=="usb" ATTRS{idVendor}=="16c0" ATTRS{idProduct}=="27db" MODE:="0600"
SYMLINK+="RFID" OWNER="osmc"
```

Und wieder mit STRG+X und Y verlassen.

16. Einrichten

Jetzt müssen wir die Datei conf.py prüfen. Dies machen wir indem wir uns am PC auf den Pi einloggen und unter dem Ordner für die Skripte die Datei raussuchen und mit proton (einen Editor) öffnen.

Verzeichnisstruktur

```
\\192.168.178.65eurepi IP\osmc\xbmc-rfid-music-master
```

Bei mir sieht die Datei so aus: Achtung die Slashes sind anders herum im Windows!

```
# which device is your RFID reader?
```

```
rfidreader = "/dev/input/event3"
```

```
# path to playlist files, ensure you have the trailing slash
```

```
playlistpath = "/home/osmc/.kodi/userdata/playlists/music"
```

```

# base path of where your music is stored
# (this is only used by the playlist creator m3u.py)
musicbasepath = "/home/osmc/Music"

# auto shuffling options
# you can set either of these variable to a string of character(s)
# if both settings are blank, nothing is ever shuffled
# if both are set, shuffle string over rules
# if a playlist contains this string it will be shuffled (randomized):
shuffle_string = None

# alternatively, all playlists will be shuffled unless the name contains this string:
no_shuffle_string = "-"

# I personally use no_shuffle_string = "-" and name playlists for albums like this:
# artist-album_title.m3u

# For genre playlists or playlists containing many albums for an artist I name them:
# jazz.m3u
# artist.m3u

```

An welchem event der Reader angeschlossen ist könnt ihr im Abschnitt 15 sehen.

Nun Datei schließen und speichern.

17. Playlist einfügen

```
python addcard.py ~/home/osmc/.kodi/userdata/playlists/music/test.m3u
```

Jetzt wechseln wir in das Verzeichnis der skripte im Putty.

```
cd /home/osmc/xbmc-rfid-music-master
```

jetzt fügen wir eine testplaylist ein.

```
sudo python addcard.py test.m3u
```

```
sudo python rfid-music.py
```


18. Das System als Service integrieren

```
sudo chmod 755 rfid-music.py
```

```
sudo cp xbmc-rfid-music /etc/init.d/
```

```
sudo chmod 755 /etc/init.d/xbmc-rfid-music
```

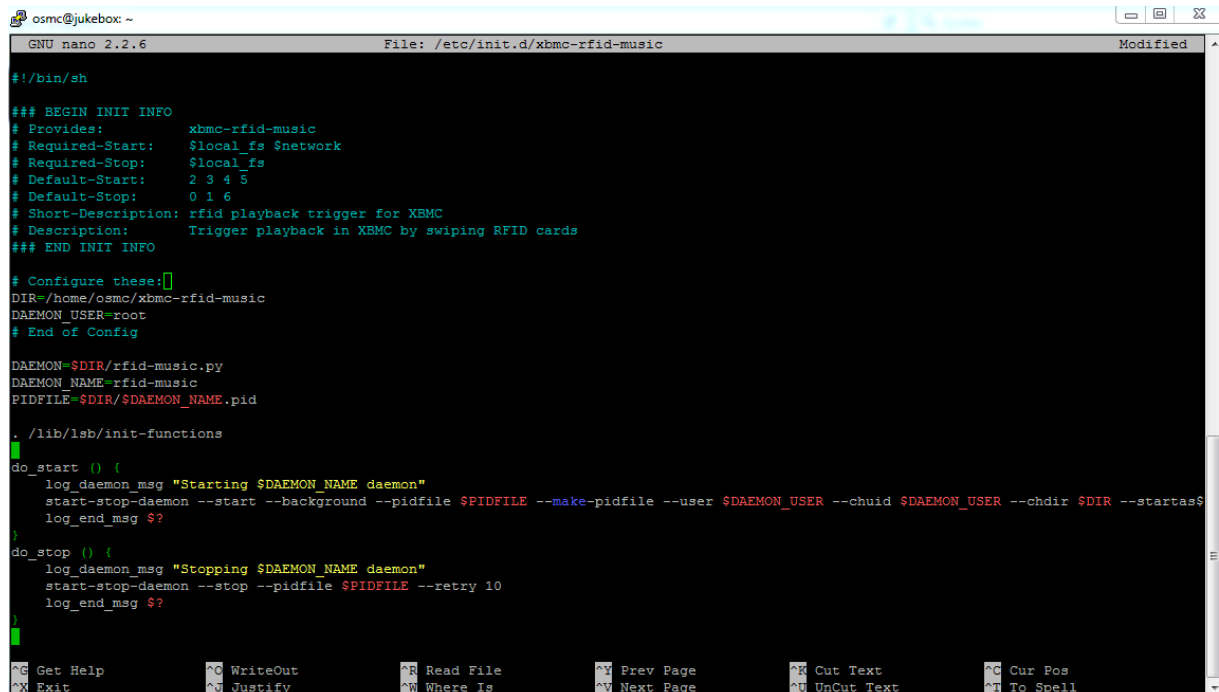
```
sudo update-rc.d xbmc-rfid-music defaults
```

Anpassen ist auch

```
Sudo nano /etc/init.d/xbmc-rfid-music
```

Hier muss unten konfigurieren diese

```
DIR=/home/osmc/xbmc-rfid-music
```



```
osmc@jukebox: ~
GNU nano 2.2.6 File: /etc/init.d/xbmc-rfid-music Modified
#!/bin/sh

### BEGIN INIT INFO
# Provides:          xbmc-rfid-music
# Required-Start:    $local_fs $network
# Required-Stop:     $local_fs
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: rfid playback trigger for XBMC
# Description:       Trigger playback in XBMC by swiping RFID cards
### END INIT INFO

# Configure these:[]
DIR=/home/osmc/xbmc-rfid-music
DAEMON_USER=root
# End of Config

DAEMON=$DIR/rfid-music.py
DAEMON_NAME=rfid-music
PIDFILE=$DIR/$DAEMON_NAME.pid

. /lib/lsb/init-functions

do_start () {
    log_daemon_msg "Starting $DAEMON_NAME daemon"
    start-stop-daemon --start --background --pidfile $PIDFILE --make-pidfile --user $DAEMON_USER --chuid $DAEMON_USER --chdir $DIR --startas$
    log_end_msg $?
}

do_stop () {
    log_daemon_msg "Stopping $DAEMON_NAME daemon"
    start-stop-daemon --stop --pidfile $PIDFILE --retry 10
    log_end_msg $?
}

^G Get Help      ^C WriteOut      ^R Read File     ^Y Prev Page     ^K Cut Text      ^G Cur Pos
^X Exit          ^U Justify       ^W Where Is      ^V Next Page     ^U UnCut Text   ^T To Spell
```

Um zu prüfen ob der Service jetzt automatisch startet einmal `sudo reboot`

Und dann folgendes eingeben

```
sudo service xbmc-rfid-music status
```

dann steht dort in grün activ

Aktueller Fehler

```
osmc@jukebox: ~
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Oct 27 21:35:12 2017 from 192.168.178.42
osmc@jukebox:~$ sudo service xbmc-rfid-music status
* xbmc-rfid-music.service - LSB: rfid playback trigger for XBMC
   Loaded: loaded (/etc/init.d/xbmc-rfid-music)
   Active: active (exited) since Fri 2017-10-27 21:43:30 CEST; 1min 21s ago
   Process: 321 ExecStart=/etc/init.d/xbmc-rfid-music start (code=exited, status=
0/SUCCESS)

Oct 27 21:43:29 jukebox systemd[1]: Starting LSB: rfid playback trigger for....
Oct 27 21:43:30 jukebox xbmc-rfid-music[321]: Starting rfid-music daemon:sta...
Oct 27 21:43:30 jukebox xbmc-rfid-music[321]: start-stop-daemon: child retur...
Oct 27 21:43:30 jukebox xbmc-rfid-music[321]: start-stop-daemon: unable to c...
Oct 27 21:43:30 jukebox xbmc-rfid-music[321]: failed!
Oct 27 21:43:30 jukebox systemd[1]: Started LSB: rfid playback trigger for XBMC.
Hint: Some lines were ellipsized, use -l to show in full.
osmc@jukebox:~$ sudo service xbmc-rfid-music status -l
* xbmc-rfid-music.service - LSB: rfid playback trigger for XBMC
   Loaded: loaded (/etc/init.d/xbmc-rfid-music)
   Active: active (exited) since Fri 2017-10-27 21:43:30 CEST; 1min 48s ago
   Process: 321 ExecStart=/etc/init.d/xbmc-rfid-music start (code=exited, status=0/SUCCESS)

Oct 27 21:43:29 jukebox systemd[1]: Starting LSB: rfid playback trigger for XBMC..
Oct 27 21:43:30 jukebox xbmc-rfid-music[321]: Starting rfid-music daemon:start-stop-daemon: unable to open pidfile ...tory)
Oct 27 21:43:30 jukebox xbmc-rfid-music[321]: start-stop-daemon: child returned error exit status 2 (No such file o...tory)
Oct 27 21:43:30 jukebox xbmc-rfid-music[321]: start-stop-daemon: unable to chdir() to /home/osmc/xbmc-rfid-music (N...tory)
Oct 27 21:43:30 jukebox xbmc-rfid-music[321]: failed!
Oct 27 21:43:30 jukebox systemd[1]: Started LSB: rfid playback trigger for XBMC.
Hint: Some lines were ellipsized, use -l to show in full.
osmc@jukebox:~$ sudo service xbmc-rfid-music status -l
Usage: /etc/init.d/xbmc-rfid-music {start|stop|restart|status}
osmc@jukebox:~$ status
-bash: status: command not found
osmc@jukebox:~$
```