Convergence

multimedia and home automation (at a very low cost)

A Raspberry PI wide application

Alex Pojer

Convergence

multimedia and home automation (at a very low cost)

A Raspberry PI wide application

by Alex Pojer

Preface

In 2014 I heard about the Raspberry Pi board, so I bought one in the middle of that year to test its potential. The results were exciting, because they allowed me to approach the much-proclaimed *technological convergence* [Ref.1].

I had defined some initial objectives, which were not restricted to a simple *media-center*, but which included some *home-automation* concepts, then still little known to the general public (*Amazon Alexa* was only released, as a limited edition, in late 2014).

What until recently seemed inconceivable, was at that moment achievable by means of a small, cheap and harmless device.

While I was approaching those goals, I developed the procedures that accomplished them and took note, as if I was connecting the dots of a still partially unknown drawing.

These are the notes of that project.

The board used for the test was a Raspberry Pi Model B+, but the concepts may be easily transferred to other more powerful modern boards.

1.Aim of this book

The mission of this project is to define an optimal hw/sw configuration to implement a comprehensive home controller.

1.1.Project specifications

Hereafter, you can find some of the specifications this system should realize:

- TV program digital receiver (terrestrial or Sat)
- Personal Video Recorder (PVR)
- IP-TV
- Mediaplayer
- Upnp/SMB/VPN server
- phone line connection (PSTN) to forward VOIP in/out-coming calls
- voice control commands (home light control, knowledge easy access, …)
- webradio/media with external audio amplifier connection
- anti-theft home security system [Ref.2]
- access to the system via remote login connection [Ref.3]
- infrared remote receiver
- remote control via smartphone app [Ref.4]
- personal cloud

Not all of these goals have been fully developed here, but the road is drawn.

2.Project implementation with Raspberry Pi (modelB+)

To get enough flexibility, performances and a reduced power consumption, *Raspberry Pi* is a good compromise. It is a Single-Board Computer (SBC) with ARM processors, memory, Inputs/Outputs (I/O) and graphic subsystem. It is developed in the UK by the Raspberry Pi Foundation as a “way of approaching computer science”.

2.1.Choosing the Operating System

A lot of Operating Systems are now available for ARM processors. Hereafter, you can find some of them, available for free:

1. Raspberry Pi OS (previously called Raspbian)
   This distribution is derived from Debian, and it has been created specifically for Raspberry Pi systems.
2. OSMC (Open Source Media Center, formerly Raspbmc)
   Even this distribution is derived from Debian, but built around the KODI mediacenter (formerly known as XBMC). It can handle more components in an easier way (the *OSMC Settings* internal module is already enabled to manage external USB Audio devices, firewall, ssh, remote control, etc.) [Ref.5]
3. LibreELEC
   Light Linux distribution built around Kodi mediacenter
4. O.S images
   From the Raspberry Pi website you can get officially supported and third party operating systems images.

How to write down a downloaded O.S. image

To install an O.S. image on a SD card, you can follow the Official Raspberry Pi organization online guide or look at the Addendum ("How to write down an image").

2.2.Needed components or services

1. Raspberry Pi platform board [Ref.6]
2. Operating System (O.S.)
3. Power supplier
4. Plastic box (minimum size 60x90x25mm)
5. SD card (suggested 8GB)
6. External HDD/USB key (self-powered or connected to self-powered usb hub)
7. Self-powered USB hub (if needed, to enhance driving capability of the system)
8. Analog modem (only for pstn interface purposes)
9. USB2Serial converter (only to be used to interact with analog line)
10. DTT USB receiver
11. SAT USB receiver
12. Wireless keyboard with touchpad
13. Video Component (or HDMI) to VGA converter (not needed if a modern TV-set is available or using a SSH remote connection)
14. USB external audio DAC
15. Ethernet/wireless internet connection
16. Heatsinks kit for overclocking
17. Expansion Shield with IN/OUT ports
18. External audio amplifier and speakers

19. Static IP address or equivalent

3.Which Operating System

From the Raspberry Pi website Downloads page [Ref.7], you can get some O.S. images to be flashed in your SD memory card.
To flash the SD, if you are using a GNU/Linux PC you can follow the procedure in the section How to write down an image to SD card in the *Addendum*. For others PCs, you can read the reference at link [Ref. 8]
Afterwards, only two O.S. will be evaluated: **RaspberryPi OS** and **OSMC**. They are quite different, because the first one is a typical Linux distribution; the second one is a Linux distribution with a mediacenter user interface.

3.1.RaspberryPi OS/Raspbian

"Raspbian is a free operating system based on Debian, optimized for the Raspberry Pi hardware. An operating system is the set of basic programs and utilities that makes your Raspberry Pi to run. However, Raspbian provides more than a pure OS: it comes with over 35,000 packages, pre-compiled software bundled in a nice format for easy installation on your Raspberry Pi." [Ref.9]

3.1.1.Setting up the system

Once you'll turn on the system, you'll get a prompt. Then you can set up the keyboard, running from the shell this command:

```
Raspi-config
```

At this point, you can launch the Graphic User Interface (GUI):

```
startx
```

3.1.2.Programs to install

To simplify the search and installation of a software from the online repositories, you can start installing **synaptic**, running from the shell this command:

```
sudo apt-get install synaptic
```

To get a mediacenter, install **kodi**, available on the default Raspbian repositories [Ref.10]:

```
sudo apt-get install kodi
```

3.2.OSMC

OSMC (*Open Source Media Center*) ([Ref.11]) is a linux-based Operating System distribution with a Kodi media-center interface and it was born for Raspberry Pi. It is maintained by Sam Nazarko and a team of volunteers. It was formerly known as Raspbmc.

After installation, if needed, you can reconfigure your keyboard layout [Ref.12]:

```
sudo dpkg-reconfigure keyboard-configuration
```

or reconfigure locales:

```
sudo dpkg-reconfigure locales
```

3.2.1.Configuration menu from KODI/OSMC

OSMC has a configuration program to set some specific features.
These ones are:

- Updates/backups settings
- Services (ssh/samba server)
- Overclock settings
- App Store
- Network settings
- Remotes
- Pi Configuration (hdmi output, MPG2 license, memory allocation, GPIO configuration)

To enter settings menu, from the KODI interface, select:

Program -> OSMC Settings

3.2.2.Setting a USB Audio device

To enable a USB audio device and setting *dvdplayer* to enable audio in video playback, you can read Audio problems using external usb peripheral in the chapter "Tuning the system: issues and surroundings".

4.Choosing applications

Once you have chosen the preferred O.S., you could even choose the applications to install on it. There is a convenient way to proceed with the installation, using a computer remotely connected via a *ssh* terminal to the IP of your system in the local subnetwork.
Hereafter, you can find a collection of useful applications.

4.1.KODI

"**Kodi** (formerly XBMC) is a free and open-source media player software application developed by the *XBMC Foundation*, a non-profit technology consortium. Kodi is available for multiple operating systems and hardware platforms" (Wikipedia).
Kodi has become very popular for its easy-to-use interface and for its *Add-ons* that give it great flexibility, allowing to integrate images, music, video and other kind of entertainments.
If you choose a distribution without KODI "embedded" (e.g. RaspberryPiOS), you can install prebuilt packages, looking for them in the Internet.

Hereafter, the related Raspbian repository:

http://www.raspbian.org/RaspbianRepository

After installation you can run Kodi:

kodi --standalone

4.2.CODECs and Audio/Video conversions

If you have chosen RasberryPiOS/Raspbian as O.S., to perform A/V conversions, you may install one of the following tools:

- FFmpeg [Ref.13]
- MEncoder
- handbrake [Ref.14]
- Gstreamer

The first three ones are specific transcoding tools. The last one is a pipeline based command, linking together other commands in a complex workflow.

To make the transcoding operation more efficient, it's better to use hardware accelerated audio/video conversions. In this case, the converter should use the Broadcom's OpenMAX libraries. Hereafter, I'll introduce using Gstreamer in addiction to these libraries.

4.2.1.Installing Gstreamer with OpenMAX libraries

As usual, you may choose to install the tool from source code compilation or installing already compiled software.

4.2.1.1.Source code compilation

You can install Gstreamer compiling its source code. According to [Ref.15], you need the OSMC development packages:

```
sudo apt-get update
sudo apt-get -y install rbp-userland-dev-osmc
```

Then you have to compile gstreamer with an option such as *--enable-openmax* when you run *./configure*

4.2.1.2.Executable release

You can install the already compiled version from the *Raspian Repos*, or adding to /etc/apt/sources.list the source (now deprecated):

```
deb http://vontaene.de/raspbian-updates/ . main
```

Then, to go further with the installation [Ref.16]:

```
sudo apt-get update

sudo apt-get install libgstreamer1.0-0 libgstreamer1.0-0-dbg libgstreamer1.0-dev liborc-0.4-0 \
liborc-0.4-0-dbg liborc-0.4-dev liborc-0.4-doc gir1.2-gst-plugins-base-1.0 gir1.2-gstreamer-1.0 \
gstreamer1.0-alsa gstreamer1.0-doc gstreamer1.0-omx gstreamer1.0-plugins-bad \
gstreamer1.0-plugins-bad-dbg gstreamer1.0-plugins-bad-doc gstreamer1.0-plugins-base \
gstreamer1.0-plugins-base-apps gstreamer1.0-plugins-base-dbg gstreamer1.0-plugins-base-doc \
gstreamer1.0-plugins-good gstreamer1.0-plugins-good-dbg gstreamer1.0-plugins-good-doc \
gstreamer1.0-plugins-ugly gstreamer1.0-plugins-ugly-dbg gstreamer1.0-plugins-ugly-doc \
gstreamer1.0-pulseaudio gstreamer1.0-tools gstreamer1.0-x libgstreamer-plugins-bad1.0-0 \
libgstreamer-plugins-bad1.0-dev libgstreamer-plugins-base1.0-0 libgstreamer-plugins-base1.0-dev
```

To check the Gstreamer plugins are installed, you can use the *gst-inspect-1.0* command, then to learn more about a specific plugin, you need to pass its name to the command line.

To check if the OpenMAX extensions have been correctly acquired, please get the results for:

```
gst-inspect-1.0 | grep omx
```

Be aware that, in order to decode mpeg2 via hardware (omxmpeg2-hardware), you should have the mpeg2 license enabled on your RaspberryPi.

4.2.2.Using Gstreamer for realtime conversion

Hereafter, you can find the command to hardware transcode an mkv/ts/mpeg2 AV flow into h264 format, with AAC audio encoding:

```
gst-launch-1.0 -e filesrc location="test.mkv" \
! decodebin name=demux demux. \
! queue \
! audioresample \
! audioconvert dithering=0 \
! "audio/x-raw,channels=2" \
! voaacenc bitrate=128000 \
! mux. mp4mux name=mux \
! filesink location="test.mp4" demux. \
! queue \
! videoconvert \
! omxh264enc target-bitrate=1500000 control-rate=variable inline-header=true periodicty-idr=250 interval-intraframes=250 \
! "video/x-h264,stream-format=byte-stream,profile=high" \
! h264parse config-interval=2 \
! mux.
```

4.3.Asterisk

Asterisk is an open source project capable of providing Private Branch Exchange (PBX), a VoIP Gateway/Server or other services for an IP/PSTN communication system.

Asterisk and the Apache web server need a huge amount of RAM that might slow down the system when running X Windows at the same time. To fix this, you can either shut down X Windows, or use the **FreePBX** interface, a web-based Graphic User Interface (GUI), from another computer on the network.

For installation purposes, you can follow the procedure in the par. Steps to install Asterisk and FreePBX in the *Addendum*.

If you want to work with XWindows graphical environment, shut down Apache and Asterisk from a root shell:

```
service apache2 stop
amportal stop
```

To start them again run:

```
service apache2 start
amportal start
```

4.4.Other applications

Hereafter, there's a list of recommended software, very helpful in setting and using your system as a PC. To manage the installation you can use Synaptic.

Tightvncserver *(vnc server for remote connections)*
php5
php-pear

You may install the following software as a starting set up of your system:

vlc	*(multimedia player, with a wide variety of CODECs embedded)*
synaptic	*(graphic interface to manage software installation)*
csh	*(advanced terminal command line)*
lxde	*(desktop environment)* [Ref.17]
nedit	*(advanced intuitive text editor)*
gufw	*(graphic interface for firewall set-up)*
motion	*(application to record/catch images from a web cam)* [Ref.18]
lame	*(mp3 audio Encoder)*
screen	*(sect.4 in the "Addendum", "How to install and use Screen")*

5.Voice recognition, control and synthesis

To get voice recognition and vocal synthesis, you can install local or remote services.
There are a lot of projects to obtain this service (*festival*, *mbrola*, *sphinix*, *espeak*), but I chose *Voicecommand* from *Steven Hickson*'s project. [Ref.19]
This project was using *Google APIs* to manage voice recognition and synthesis, and it was able to control Raspberry PI I/O pins according to voice commands.
In this way you can configure your system to behave as a modern *Google Home Assistant*, acquiring information in the Internet, programming some guided responses to your questions, controlling your home appliances, and so on.

As an example, I controlled the switch on/off of a lamp and the color of its light just using vocal commands ([Ref.20]) via infrared pulses, with the circuit specified at par. 8.2.

5.1.Installation of Voicecommand

To get installed *Voicecommand* [Ref.21], connect to your system via a *ssh* terminal and enter:

```
sudo apt-get install git-core
git clone git://github.com/StevenHickson/PiAUISuite.git
cd VoiceCommand
```

Change the string given to Wolfram-alpha server in the voicecommand.cpp:

```
link += "&reinterpret=true&translation=true&format=plaintext&input=";
cd PiAUISuite/Install/
./InstallAUISuite.sh
```

5.2.Voicecommand setup instructions

To get information about options and parameters you can use in *Voicecommand* application, type:

```
voicecommand -s
```

5.3.Update

You can update the application running the following commands:

```
cd PiAUISuite
git pull
cd Install
sudo ./UpdateAUISuite.sh
```

Hereafter the dependencies required to run and build:

```
sudo apt-get install libboost1.50-dev libboost-regex1.50-dev youtube-dl axel curl \
   xterm libcurl4-gnutls-dev mpg123 flac sox
```

5.4.Setting up the configuration file

The file *.command.conf* is needed to configure the application. It contains the language definition, the system response, the commands to be interpreted and so on.

You can find an example of this file in the section 5, Example of .command.conf file, of the "*Addendum*".

5.5.How to run voicecommand in background

Voicecommand will work only if it's running. For this reason, if you don't put it in the start up application group, you need to launch it in background. To do so:

1) Connect via remote SSH session
2) run:

```
voicecommand &>/dev/null
CTRL-z
bg
```

Sometimes the application would crash, so, as a future improvement, we could think of a task to check if it's still running and run it again in case of crash.

In the Internet you can find tons of other projects of voice recognition ([Ref.22], [Ref.23]) and *text2speech* systems [Ref.24].

6.Security

The security surveillance mode can be performed via video/audio devices and adding presence detector devices:

1) webcam
2) microphone
3) occupancy sensors

Hereafter, I'm going to consider only the first kind of devices, leaving microphone and occupancy sensors to future implementations.

6.1.Webcam

To manage web cameras you can install **motion**:

```
sudo apt-get install motion
```

If the motion daemon is running, the system is able to detect any movement in the field of view of the camera. After that, you can either save videos or pictures.
Be sure to give the needed writing rights to your saving folders.
Furthermore, you can remotely observe the environment under surveillance, because an integrated web server is also available.
To set the program up, you have to edit the *motion.conf* file, under */etc/motion/*

You can *start/stop/restart* the daemon accordingly:

```
sudo service motion start/stop/restart
```

There are some projects to get the DropBox connection, to store web cameras results in that cloud.

It could be interesting to approach the study of a more complex system, built by *RaspberryPI* (as a system controller) and *Arduino* devices (as peripheral leaf-devices).

7.Remote login

You can access your system via remote login from either inside the same subnetwork or externally (worldwide).
To ensure an adequate level of security, my suggestion is to activate a secure *ssh* protocol, to sustain a cryptographic network protocol, capable of connections over even unsecured networks.

7.1.Getting your public IP address

To reach easily the machine from inside the same local subnetwork, it's better to assign it a fixed IP address (usually something like 192.168.x.x), according to the MAC address of the device, configuring your local router.
If your Internet Service Provider (ISP) provides you with a public static IP address, you can access the machine even worldwide. This is the simplest case.
If your ISP does not, you can use external services that can provide remote access with dynamic DNS [Ref.25]. You can get this facility for free or paid.
I tried the free service from noip.com that, after subscription, let you register till 3 hostnames, with the need of confirming them once every 30 days to keep them active.
Otherwise, you may get your own Dinamic DNS service. You can read the *Addendum* ("How to get external IP address without any service provider") to have an idea about it.

7.2.Setting up Dynamic DNS service updates

Whether you have subscribed either the free or the paid service, you need to update periodically your current IP address.
In order to do that, you could do one of the following:

1) compile & install a specific daemon from noip.com;
2) install *ddclient* from the standard repository;
3) set up your modem if noip is available in its settings.

For the first and third options you can follow the instructions on *noip* website or your modem manual respectively.
As an alternative, to install *ddclient* [Ref.26]:

1) ***sudo apt-get install ddclient***
2) answer to the questions during installation (username, pwd, protocol dyndns2, etc.)
3) then ***sudo nano /etc/ddclient.conf***
4) protocol=dyndns2
5) use=web, web=checkip.dyndns.com/, web-skip='IP Address'
6) server=dynupdate.no-ip.com
7) login=yourusername
8) password=yourpassword
9) yourhostname.no-ip.org

10) to set the number of seconds between updates, add: ***daemon=600***
11) to use ssl, add: ***ssl=yes***
12) restart the client: ***sudo /etc/init.d/ddclient restart***
13) to ensure the configuration is working: ***sudo ddclient -daemon=0 -debug -verbose -noquiet***

After getting your public IP address updated, to get your Raspy reached worldwide you'll have to set up your modem/router's port forwarding, pointing to the Raspy subnetwork IP address (e.g. 192.168.1.xxx) with TCP port:22.

7.3.Virtual Network Computing (Vnc)

To get remotely a full graphic interface for your system, you can log-in and install a Virtual Network Computing (VNC) software [Ref.27], with the following commands:

```
sudo apt-get update
sudo apt-get install vnc-server
```

After *vncserver* installation, to set a password you can type:

```
vncserver
```

To connect with a PC, you can run the VNC **client** using the chosen password for the server on port number **5901**.
For test purposes I used Remmina Remote Desktop Client under Ubuntu-GNU/Linux distribution, without any user name, but only the passwd.

To stop the vnc server on the Raspberry:

```
vncserver -kill :x
```

where x is the display number.

To make vnc server always available after system start up, use the *cron* command to schedule its launch:

```
crontab -e
```

and add the line:

```
@reboot /usr/bin/vncserver
```

8.IR Remote Control/ed

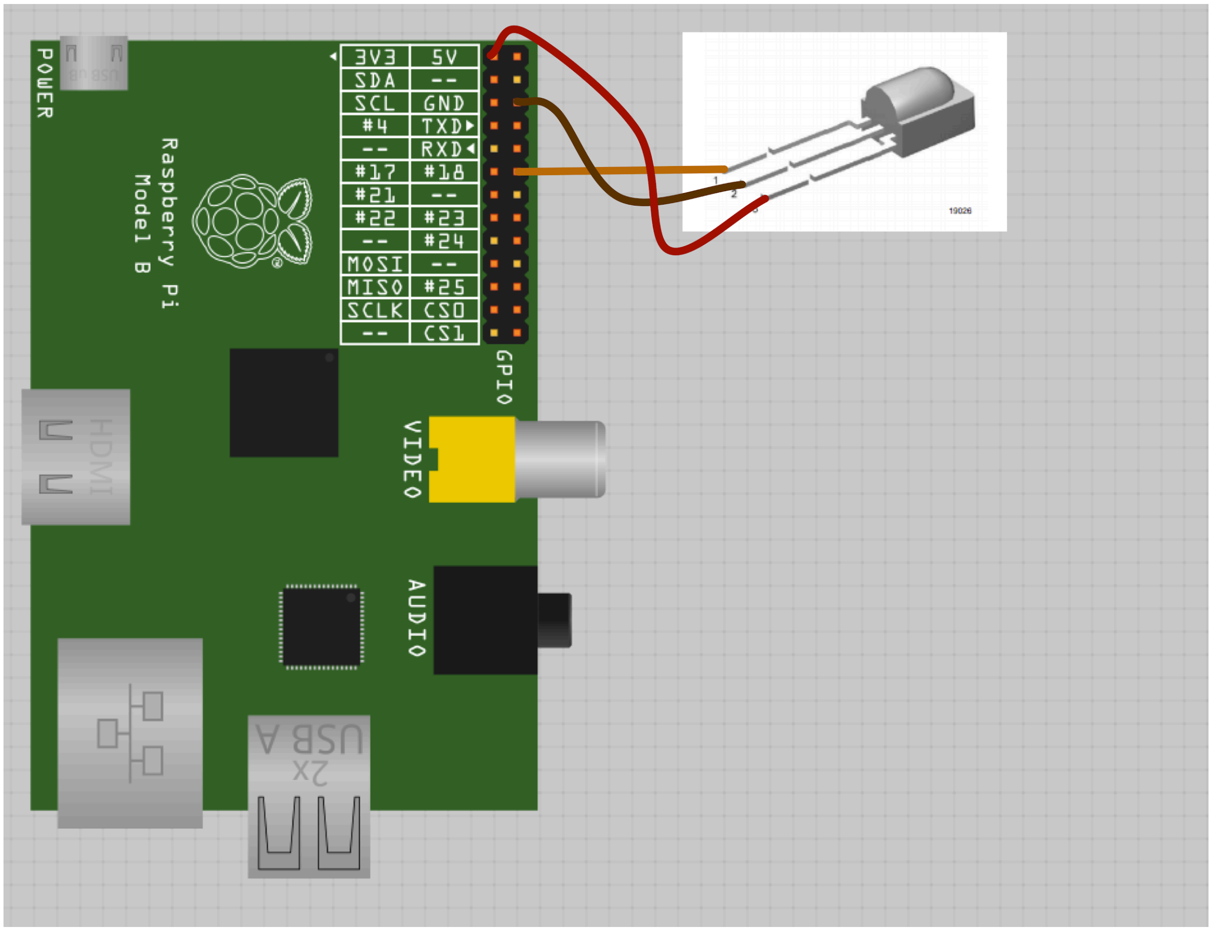
If your Raspberry Pi is connected via HDMI to a TV-set, you can use its remote control to manage the KODI media center too, otherwise, you should have a dedicated infra-red receiver connected to the Raspy.
If your system already includes a DTT USB receiver, likely you already have an infrared (IR) receiver integrated in it.

The advantage of having an IR receiver is that you can control the system by a standard remote control even when the television is switched off.
Below I will describe the steps to create a simple IR receiver/transmitter module to add to your **Raspberry Pi Model B** (**only for the OSMC distribution**).

8.1.Receiver

According to projects at [Ref.28] and [Ref.29] you can build a IR receiver. I did that for the OSMC distribution.
You need a few components:

- 1 IR receiver sensor (TSOP38238);
- Some female to female leads, such as wiring connectors
- A remote control (i.e. your TV-set RC)



Schematic of the IR receiver, for the Raspberry Pi Model B, by Simon Monk

For your convenience, I have listed here the steps to follow, using a *Raspberry Pi Model B+*:

a) connect the IR receiver photo-sensor according to the schematics above (using pins 3.3V, GND and GPIO18)

b) open the “*OSMC Settings*” (in the OSMC distribution) and be sure the “*Enable GPIO TSOP IR Receiver*” option is not set. Otherwise, unset it and restart KODI
c) connect to the Raspberry by a PC via SSH
d) sudo modprobe lirc_rpi
e) sudo kill $(pidof lircd)
f) mode2 -d /dev/lirc0
g) pressing buttons on the remote will display pulse/space codes on the monitor
h) to know the allowed key names for the following process, type the command:
irrecord –list-namespace
i) to generate the *config file*, type the command:
irrecord -d /dev/lirc0 ~/lircd.conf
j) open “*OSMC Settings*” and enable the IR receiver (“*Enable GPIO TSOP IR Receiver*”)
k) set the “*GPIO Remote Profile*” to “*Custom (lircd.conf)*”
l) copy the *lircd.conf* file in */etc/lirc*
m) in the OSMC menù, select *Program -> OSMC Settings*
n) select *Remote* and choose your *lircd.conf* file

8.2.Transmitter

You can even add to the Raspberry board some circuits to control home appliances.
If you prefer to keep your system electrically separated from other external devices, starting from projects [Ref.30] and [Ref.31], you can create an infrared transmitter extension, with the possibility of cloning the remote controls of the devices you need to drive.
To do that, you need few components:

- 2 IR emitting diodes (2N3906);
- 1 transistor P2N2222AG
- 1 10 KΩ resistor
- Some female to female leads, such as wiring connectors

Hereafter, I’m going to write the simple steps to build the IR transmitter:

a) connect the components according to the schematic below, for the **Raspberry Pi Model B** (using of: pins 5V, GND, GPIO22)
b) Click on the “*OSMC Settings*”, in the OSMC distribution, and be sure the “*Enable GPIO TSOP IR Receiver*” option is not set, otherwise, unset it and restart KODI
c) Open a SSH connection from a PC to the Raspberry and add these lines to your */etc/modules* file:
lirc_dev
lirc_rpi gpio_out_pin=22
d) Change your */etc/lirc/hardware.conf* file to:
LIRCD_ARGS="--uinput"
DRIVER="default"
DEVICE="/dev/lirc0"

```
MODULES="lirc_rpi
```

e) Having previously built the receiver, now you can clone any remote control, so, write in the terminal*:*

```
sudo modprobe lirc_rpi
sudo kill $(pidof lircd)
mode2 -d /dev/lirc0
```

f) pressing buttons on the remote **you want to clone**, you will see pulse/space codes on the monitor

g) to know the allowed key names for the following process, via SSH, type the command:

```
irrecord –list-namespace
```

h) to generate the config file with the given names of step g) (even if it is not the final one), type the command

```
irrecord -d /dev/lirc0 ~/lircd.conf2
```

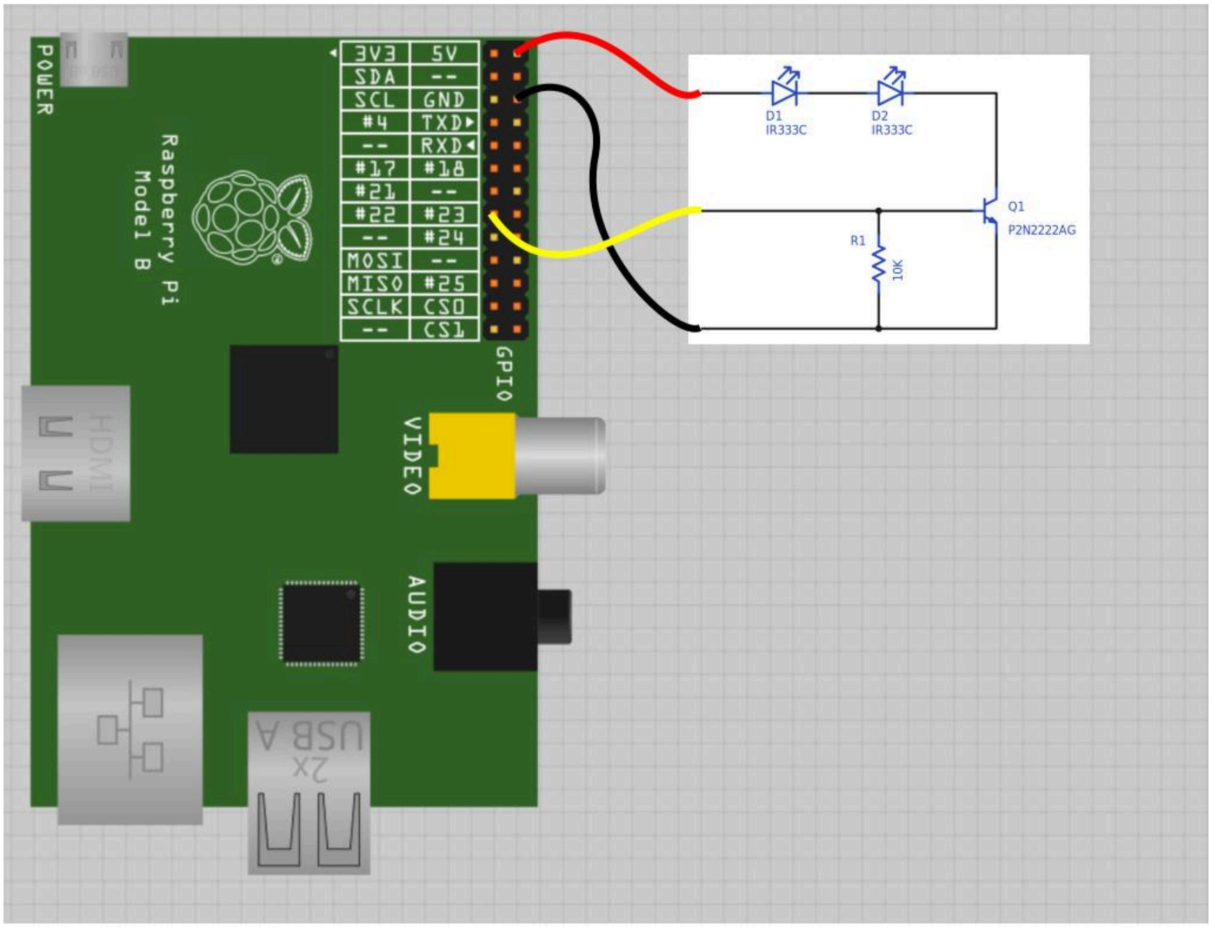
i) copy from **begin remote** to **end remote** from *lircd.conf2* to *lircd.conf* file (already set in IR Receiver setting) and set the remote control name after the keyword **name**, and the final button names

j) Now, restart lircd so it acquires these changes:

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

k) run one of the programmed buttons with:

```
irsend SEND_ONCE Remote_Name PROGRAMMED_BUTTON
```



Schematic of the IR transmitter, for the Raspberry Pi Model B

If you have any issue, look at the section "Issues building a IR transmitter", in the chapter "Tuning the system: issues and surroundings".

9.DVB-T receiver

Using a Raspberry Pi with KODI installed, you can implement a DVB-T (Terrestrial Digital Video Broadcasting) receiver, receiving free digital television programmes.
For this purpose, you need a USB DVB-T receiver and install a TV *back-end* server (such as *TVHeadend*).
At the referenced link [Ref.32],you can find a USB DVB-T device list compatible with Linux kernels. To implement this project, I bought a Realtek RTL2832-chipset based receiver.
To get an easy installation of TVHeadend, you can follow the tutorial in the "*Section 3"* of the *Addendum*, "How to install TVHeadend" (release version 4.1 of the program).

On line you can find lot of tutorials about using TVHeadend [Ref.33].

9.1.EPG

In addition to TV programs, each television broadcaster transmits information related to its daily programs. The *Electronic Programme Guide* (*EPG*) usually contains titles, timings and content plots of programmes to watch.

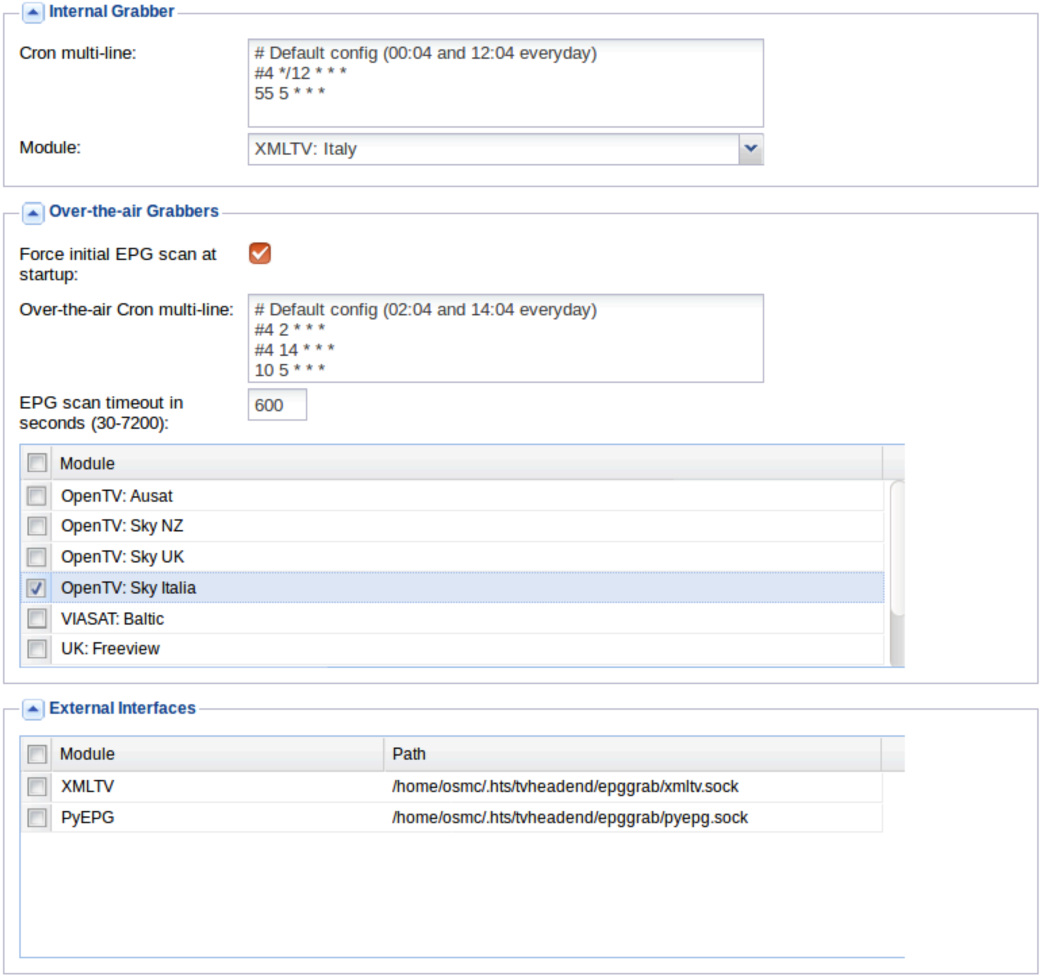
TVHeadend is able to get EPG information in three different ways [Ref.34]:

- Over-The-Air (OTA)
- Internal grabber
- External grabber

The configuration page for these grabbers can be accessed using an Internet browser by navigating the IP address of your device, at **port 9981**, then selecting:

Configuration -> Channel/EPG -> EPG Grabber

See the figure below:



Internal Grabber

Cron multi-line:

```
# Default config (00:04 and 12:04 everyday)
#4 */12 * * *
55 5 * * *
```

Module: XMLTV: Italy

Over-the-air Grabbers

Force initial EPG scan at startup: ☑

Over-the-air Cron multi-line:

```
# Default config (02:04 and 14:04 everyday)
#4 2 * * *
#4 14 * * *
10 5 * * *
```

EPG scan timeout in seconds (30-7200): 600

☐	Module
☐	OpenTV: Ausat
☐	OpenTV: Sky NZ
☐	OpenTV: Sky UK
☑	OpenTV: Sky Italia
☐	VIASAT: Baltic
☐	UK: Freeview

External Interfaces

☐	Module	Path
☐	XMLTV	/home/osmc/.hts/tvheadend/epggrab/xmltv.sock
☐	PyEPG	/home/osmc/.hts/tvheadend/epggrab/pyepg.sock

9.1.1.OTA Grabber

After installing *TVHeadend* back-end server, if you want your system to collect the EPG in the simplest way, you can choose the *Over The Air* (*OTA*) grabber, to get EPG directly over the air, choosing a module from *Over-the-air Grabbers* in the EPG Grabber section.

Please, visit the *github* section of the XMLTV project [Ref.35] for more information and to get your locale *tv_grab_xx* file, then restart TVHeadend.

9.1.2.Internal Grabbers

In this case, the grabber can be initiated from within TVHeadend using a scheduler to set a time to catch EPG data directly from the Internet, by selecting the corresponding "*Module"* in that section.

The scheduler can be programmed in the section *"Cron multi-line"* with five parameters, according to this convention:
[Mins (0-59)] [Hours (0-23)] [Day of the month (0-31)] [Month (1-12)] [Day of the week (0-6)]

A "*", as a parameter, means any value.
As an example, if you want to run the grabber everyday at 5:55 A.M., you can write:

```
55 5 * * *
```

9.1.3.External Interfaces option

To grab the EPG you can install xmltv, as an external grabber:

sudo apt-get install xmltv

Then, run the configuration procedure to choose which channel you want to get information about, generating the xml file with selected channels [Ref.36]:

tv_grab_it --config-file .xmltv/tv_grab_it.conf --output xmltv.xml --days 1

After installing the *socat* command:

sudo apt-get update && sudo apt-get install socat

You can make the EPG available to the unix socket connection:

cat xmltv.xml|socat - UNIX-CONNECT:/home/osmc/.hts/tvheadend/epggrab/xmltv.sock

Enter the web GUI and select *Configuration* -> *Channel/EPG* -> *EPG Grabber*, then in "External Interfaces" choose XMLTV

9.2.Frontend on the smartphone

You can install a front end TvHeadend app to control the PVR functions from your smartphone.
Look for *TVHClient* by *R. Siebert* in your App Store.
It's a nice application and if you wish to support their project, you could buy some extra features.

9.3.Syslog

If you have any issues, you can enable the system log for debugging purposes, adding a line to the configuration file of TVHeadend:

```
sudo nano /etc/default/tvheadend
set TVH_DEBUG = 1
```

Then, from the webGUI:

Configuration -> Debugging -> Debug Log Path: /home/osmc/tvhx.log

9.4.UPnP server

To make the recorded programmes available to your home network (smart TV and so on), you have to:

1) enable UPnP server on Kodi;
2) format an external USB drive (pen/HDD) with **Ext4** Filesystem;
3) connect the USB drive to the RaspberryPi;
4) substitute *video* folder under */home/osmc/.kodi/userdata/playlists* with a symbolic link to the external drive:

   ```
   ln -s /media/<NAME of EXT DEV>/ video
   ```
5) open TVHeadend configuration browser (192.168.1.x:9981) and put the video folder path in:

   Configuration -> Recording -> Recording System Path
6) access the video folder via the smart TV
7) Record the TV programmes in mkv format to allow subtitles and fast forward/rewind (tested on LG smart TV)

10.Upnp/SMB media server

The system can be used as a media server in a local network.

For this purpose you can enable two specific connection protocols:

- Upnp/DLNA
- SMB server.

To enable Upnp, from KODI home, chose *System -> Settings -> Services*, then enable Upnp/DLNA service.

To enable SMB server (in OSMC), from *Program*, open *OSMC settings* and install *Samba Server*, then open a *ssh* remote connection from a PC and define the *smb* user password with the following command:

```
smbpasswd -a osmc
```

11.Personal cloud

You can also implement your own *Personal Cloud*, where to store documents, photos and other files, accessing it using specific apps directly from your smartphone.

For instance, you can use *AndFTP* or *Cx File Explorer* (find them in your app store) to tranfer files from/to your cloud.

Of course, you can also access your remote virtual space via a PC, using *Filezilla* or some specific functionalities, like *“Add network location”* or *“Connect to server”*, if available in the File Manager of your O.S.

12.HDMI vs. USB audio device

If you have HDMI TV set, but you wish to listen to music on your HiFi system, using an external USB audio device, you have to enable HDMI audio output for videos and USB audio DAC for playing music [Ref.37].

To do that:

- Select from the menu, *System -> Settings -> System -> Audio Output*
- Audio Output Device: *ALSA Generic USB Audio Device*
- Select from the menu, *System -> Settings -> Video -> Acceleration*
- keep enabled *OMXPlayer*
- disable *MMAL hw acceleration*

13.Tuning the system: issues and surroundings

Hereafter, you can find some popular issues you may face up and some useful procedures to fix them.

1) No more space left on SD root partition (OSMC) [Ref.38].

From a remote ssh connection terminal, write:

apt-get clean
raspi-config, then select "*Expand Filesystem*" option

2) Audio problems using external usb peripheral

Usually, external usb audio peripherals are working automatically. Anyway, in case of problems, they can be installed, setting up the *dvdplayer* option [Ref.39]

If it’s not already done, create a text file called **advancedsettings.xml** and place it in */home/osmc/kodi/userdata* folder.

The content should be:

```
<advancedsettings>
   <video>
      <defaultplayer>dvdplayer</defaultplayer>
   </video>
</advancedsettings>
```

Run the commands:

```
cat /proc/asound/cards
lsmod | grep "snd"
```

Then:

```
alsamixer
```

and select the right card (F5)

3) To backup/reload your KODI configuration

- Login remotly to the R-Pi (via *ssh*) and, from the */home/pi/* folder, type:

  ```
  tar -czf backup.tar.gz .kodi/
  ```

- Transfer that file to a secondary machine (*ftp*, *sftp*, or *scp* commands are all available options)

When needed (e.g. because of a corruption in your KODI configuration), you can write back the stored backup:

- Move the saved tar-file to the Raspberry-Pi
- Run the following commands:

  ```
  sudo systemctl stop mediacenter; tar -xzf backup.tar.gz; rm backup.tar.gz; sudo systemctl start mediacenter
  ```

4) Setting local date/time

If the procedure to localize your system is not working, you can run the command:

```
tzselect
```

Add the corresponding TZ='xxx/yyy'; export TZ in your .profile file, then log out and log in

5) How to turn down CPU usage

The *weather visualization* in KODI home page or the vnc background server may consume significant CPU power, especially if the Raspberry-Pi is an old model with little performance.
I suggest to disable these two features if not really needed.

6) To **spindown** your external HDD drive

If your external HDD is not spinning down automatically, you can manage to get this working, using some specific commands.
Infact, you can use the *hdparm* command ([Ref.40]) to set up the time the hdd is in idle mode, before entering the stand by mode.
This delay is expressed in terms of 5 seconds-unit, so, in case of setting a 10 minutes delay, you can run the following command from a *ssh* remote shell:

```
sudo hdparm  -S120 /dev/sda
```

To make this working after every reboot of the Raspberry-Pi, you can add to the */etc/hdparm.conf* file:

```
        /dev/sda {
                spindown_time = 120
                }
```

Another way should be using *udisk* command [Ref.41]. Unfortunately, the change of */etc/udisks-glue.conf* is not always working.
Anyway, to get a spindown timeout of 30 minutes (1800 seconds), you need to run:

```
sudo /etc/rc.local
sudo udisks --set-spindown /dev/sda --spindown-timeout 1800
```

7) Back up of your SD card (Linux procedure):

a) [Optional] Shrink down (200 MB) the main partition of the SD card using your disk manager (e.g. Gparted), with respect to the maximum SD card capacity.
b) Get the dev name of the SD card:
```
df -h
```
c) Unmount all the partitions of the SD:
```
umount /dev/mmcblk0p1
```
d) Make the dump of the card:
```
sudo dd bs=4M if=/dev/mmcblk0 | gzip > pi.gz
```

Then…

e) To flash a new card with the saved image:
```
gzip -dc pi.gz | sudo dd bs=4M of=/dev/mmcblk0
```

To check the job in progress, write this command in another terminal window and come back to check the first one:

```
sudo pkill -USR1 -n -x dd
```

8) Fixing a corrupted SD card filesystem

If the filesystem on the SD card is corrupted, you can fix it.

a) Get the SD card and put it in a PC via a card reader
b) Verify the name of the filesystem on the SD card with the command:
```
df -h
```
c) Run the fixing command:
```
sudo fsck.ext4 /dev/mmcblk0p2
```

9) Overclocking issues

If your device is in stuck due to overclock (OC):

a) get the SD card

b) put it in your PC
c) go to the boot partition and edit the *config.txt* file to set back the parameters correctly

10) GPG error during update

During a software package update, you may find an issue like this:

```
>sudo apt-get update
...
Fetched 198 B in 8s (22 B/s)
Reading package lists... Done
W: GPG error: http://vontaene.de . Release: The following signatures couldn't be verified because the public key
is not available: NO_PUBKEY F0DAA5410C667A3E
```

To fix it, please, write in a terminal:

```
sudo apt-get install debian-keyring
```

11) USB microphone settings

Sometimes, the USB microphone device is not automatically recognized. In this case, you can follow the wiki from the Audacity team to make it working [Ref.42]

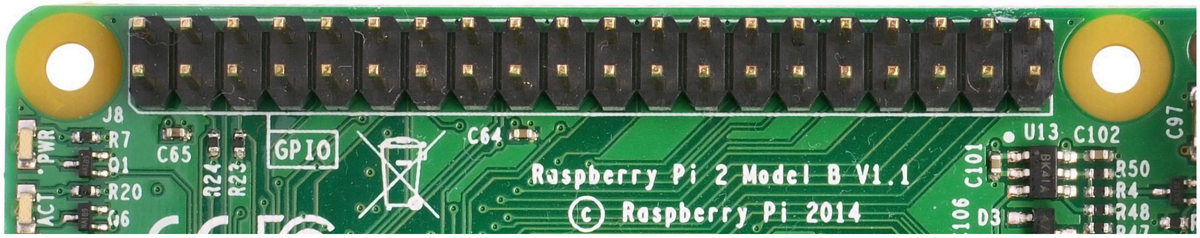
12) Issues building a IR transmitter

You may have trouble getting the infrared transmitter to work at the same time as the KODI receiver, because, as soon as you are going to enable again the OSMC IR Receiver, the IR Transmitter could stop working (at [Ref.43] and [Ref.44], you can find a solution).
After that, restart the LIRC (Linux Infrared Remote Control):

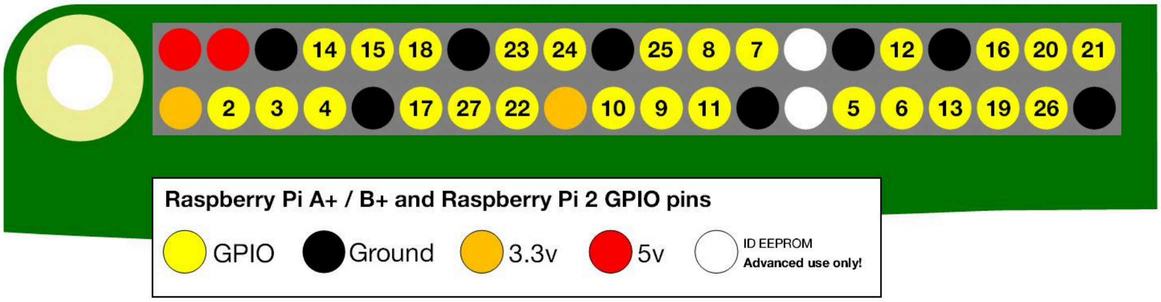
```
sudo /etc/init.d/lirc restart
```

14.Raspberry PI pin diagram

Hereafter is the pinout of the Raspberry Pi 2, Model B as you can get from the Raspberry Pi official site [Ref.45].



Number and functionalities of the pins:



The GPIO pins can be assigned via software as IN or OUT ports.

15.Power consumption

Raspberry Pi has low power consumption.

Using an energy counter, you can measure the energy used by the system. For example, for a RaspberryPI, model B+, with a self powered USB hub, I got approximately an average power of less than 5W.

These results would confirm it’s cheap and convenient to keep the system up 24/7.

16.Addendum

1) How to write down an image on the SD card

Once you’ll get an O.S. image, you have to put it on a SD card, used as a Solid State Disc (SSD) from the Raspberry Pi. Hereafter, some steps to follow if you are using a GNU/Linux O.S. to make this procedure:

a) First of all, you’ll have to know the SD card mountpoint, so, before inserting it into the reader, run the command:
   df -h
b) Now, insert the SD card into the reader and run again:
   df -h
c) A new device should be listed and you should see something like */dev/mmcblk0p1* or */dev/sdX1*, with one or more partitions
d) Unmount all the partitions of the SD card:
   umount /dev/mmcblk0pX
   or
   umount /dev/sdX1,...
   (with X as the partition number)
e) Copy the image to the SD card:
   sudo dd bs=4M if=2017-09-07-raspbian-stretch.img of=/dev/mmcblk0 conv=fsync
f) Or, if the image is zipped:
   unzip -p 2017-09-07-raspbian-stretch.zip | sudo dd of=/dev/mmcblk0 bs=4M \ conv=fsync

g) Before removing the SD card, run this command to make sure the write flash is flushed:
sync

2) Steps to install Asterisk and FreePBX

[Source: http://moishtech.blogspot.fr/2012/06/setup-and-run-asterisk-and-freebpx-on.html
and: http://wiki.freepbx.org]

a) Asterisk and needed software

Install from *synaptic*:

Asterisk
php5
php5-mysql
mysql-server (don't define root password, we'll set it later)

After that, from a terminal:

```
sudo useradd -c "Asterisk PBX" -d /var/lib/asterisk asterisk
sudo chown -R asterisk:asterisk /var/run/asterisk
sudo chown -R asterisk:asterisk /var/log/asterisk
sudo chown -R asterisk:asterisk /var/lib/php5/session/

sudo apt-get install php-pear
sudo pear install db
```

b) Download FreePBX (example with rel. 11.0.25)

```
cd /usr/src/
```

Download the program from http://www.freepbx.org/downloads

```
sudo tar -zxvf freepbx*
```

c) Setting MySQL

```
sudo service mysql start
cd /usr/src/freepbx*

sudo mysqladmin create asterisk
sudo mysqladmin create asteriskcdrdb
sudo mysql asterisk  < SQL/newinstall.sql
sudo mysql asteriskcdrdb  < SQL/cdr_mysql_table.sql
sudo mysql
```

...and continue as it follows, replacing ‘***’ with your own password...

```
GRANT ALL PRIVILEGES ON asteriskcdrdb.* TO asteriskuser@localhost IDENTIFIED BY '***';
GRANT ALL PRIVILEGES ON asterisk.* TO asteriskuser@localhost IDENTIFIED BY '***';

flush privileges;
exit
```

d) Choose the mysql root password

```
sudo mysqladmin -u root password '******'
```

e) Restart Asterisk with correct permissions

```
cd /usr/src/freepbx*
sudo service asterisk stop
sudo ./start_asterisk start
```

f) Check if Asterisk installation was good

```
sudo asterisk -r
```

g) Set the correct Timezone for PHP, opening the PHP Configuration file

```
sudo nano /etc/php5/apache2/php.ini
  date.timezone = Europe/London
```

h) Setup Apache for FreePBX and Asterisk

```
sudo nano /etc/apache2/envvars
  APACHE_RUN_USER www-data
  APACHE_RUN_GROUP www-data
```

Change those to these:

```
  APACHE_RUN_USER asterisk
  APACHE_RUN_GROUP asterisk
```

```
sudo nano /etc/apache2/sites-available/default
```

Under tags "<Directory />" and "<Directory /var/www/>" change the following settings:

```
Options FollowSymLinks
AllowOverride All
```

Finally restart apache using the following command (if you have set the webserver control option in OSMC, you need to set that port differently from 80, because of conflicts with the one used by FreePBX):

```
sudo service apache2 restart
```

i) Install FreePBX

To install FreePBX, you will need to specify the username and password of the asterisk user for the database we setup earlier, starting by typing in the following command (****** is the password you chose):

```
sudo ./install_amp --username=asteriskuser --password=******
```

For each of the prompts, just hit enter except when prompted for

Enter the path to use for your AMP web root: [/var/www/html]

in this case type in the following:

```
/var/www
```

Once this is complete, set the file permissions for FreePBX, to be accessible to apache users:

```
sudo chmod 777 /var/www/* -R
```

j) Restart all services

```
sudo amportal stop
sudo service apache2 stop
sudo service mysql stop

sudo service apache2 start
sudo service mysql start
sudo amportal start
```

k) Enter the FreePBX page

Open a web browser and insert the IP address of the Raspberry Pi:

user: admin
Password: admin (to be changed)

Config file:
/etc/amportal.conf

3) How to install TVHeadend

To build and install TVHeadend you can use the next procedure, based on a discussion available on [Ref.46].
First of all, connect to your system using *ssh.*

Install necessary packages to build TVHeadend:

```
sudo apt-get update -y
sudo apt-get install build-essential pkg-config libssl-dev git -y
sudo apt-get install libavahi-client-dev libavcodec-dev libavfilter-dev \
libavformat-dev libavutil-dev libswscale-dev libavcodec-extra-56 \
liburiparser1 liburiparser-dev debhelper libcurl4-gnutls-dev a56
sudo apt-get install fakeroot -y
```

Create and change a new temp source directory:

```
mkdir -p /tmp/src
cd /tmp/src
```

Download TVHeadhend source from *github*:

```
git clone https://github.com/tvheadend/tvheadend.git
```

Change dir to TVHeadend source dir and change the user that will run TVHeadhend from *hts* to *osmc* before you build the package:

```
cd tvheadend*
sed -i 's/hts/osmc/g' ./debian/tvheadend.default
```

Build the TVHeadend package:

```
AUTOBUILD_CONFIGURE_EXTRA=--disable-libav ./Autobuild.sh -t debian
```

You will end up with two packages (you won't need the dbg package), that you have to copy to your osmc home folder:

```
cp /tmp/src/tvheadend_*deb /home/osmc/
```

Change dir to osmc home dir and install the package:

```
cd /home/osmc/
sudo dpkg -i tvheadend_*.deb
```

If TVHeadend was not yet installed, you'll be asked to enter/choose a user name and password and then you should be able to browse to http://x.x.x.x:9981/

Then, remove the source code:

```
rm -Rvf /tmp/src/tvheadend*
```

To configure TVHeadend:

```
tvheadend -C
```

To restart TVHeadend:

```
sudo service tvheadend restart
```

If the autostart is not working automatically, you can take a look at the tutorial at [Ref.47].

4) How to install and use Screen

Screen is a program, that is very helpful in keeping a terminal session open even if the remote *SSH* session has been temporarely closed. It could be useful when you need to run a long time process (for instance, a video conversion) without having to keep active the corresponding *SSH* remote connection.

To install it:

```
sudo apt-get install screen
```

To open a screen session:

```
screen bash
```

To detach a screen session, close the remote connection

To open a new session:

CTRL + A (capital letter), then press CTRL + D (capital letter)

To list all sessions:

```
screen -list
```

To reconnect:

```
screen -r
```

If you have more than one session open, add the session name:

screen -r <session name>

To terminate a session:

CTRL + D

5) Example of .command.conf file

The following is an example of the configuration file, for the voice recognition program:

```
###########################################
#This is an example of the config file
#These are the special options you can set (remove the #)
!api==L3VH6W-9G64K49RYY
!maxResponse==1
!language==en
#Not to verify if not explicitly
!verify==0
!thresh==0.8
!continuous==1
!response=="OK"
!quiet==0
!ignore==1
!filler==0
!duration==2
!com_dur==1
#!hardware==plughw:1,0
#Here are the commands
show me==/home/pi/AUI/Imaging/test 2
track me==/home/pi/AUI/Imaging/test 1
download==download ...
play $1 season $2 episode $3==playvideo -s $2 -e $3 $1
download $1 season $2 episode $3==download $1 s$2e$3
play==playvideo -r -f ...
multiple==playvideo -r -m -c 5 ...
YouTube==youtube-search ...
#Google==google ...
Google==/home/pi/my_prog/scripts/google ...
#~music==xterm -e pianobar
~weather==/home/pi/AUI/Misc/sayweather.sh
~made you==tts "I was created by Steven Hickson" 2>/dev/null
#~music==xterm -e control-pianobar.sh play
```

```
~terminal==xterm &
#~Internet==midori &
~Internet==lynx --accept_all_cookies &
!hardware==plughw:0,0
#!thresh==2.00794
!keyword==moon
!improper=="Repeat, please"
###################################################
###LIGHTs###########################################
light==/home/pi/my_prog/scripts/lux ...
switchon==/home/pi/my_prog/scripts/switch_on ...
switchoff==/home/pi/my_prog/scripts/switch_off ...
~dark==/home/pi/my_prog/scripts/command_light_off
```

17.References

[Ref.1]: Wikipedia: Technological Convergence
https://en.wikipedia.org/wiki/Technological_convergence

[Ref.2]: Raspberry Pi as low-cost HD surveillance camera
http://www.codeproject.com/Articles/665518/Raspberry-Pi-as-low-cost-HD-surveillance-camera

[Ref.3]: The best free DynDNS providers

https://www.ionos.com/digitalguide/server/tools/free-dynamic-dns-providers-an-overview/

[Ref.4]: XBMC Raspberry Pi remote control using a smart phone

http://asliceofraspberrypi.blogspot.it/2013/07/xbmc-raspberry-pi-remote-control-using.html

[Ref.5]: OSMC
https://osmc.tv/

[Ref.6]: RaspberryPiBoard
https://elinux.org/RPi_Hub

[Ref.7]: Raspberry Pi website downloads page
http://www.raspberrypi.org/downloads

[Ref.8]: Installing operating system images

https://www.raspberrypi.org/documentation/installation/installing-images/

[Ref.9]: Welcome to Raspbian

http://www.raspbian.org/FrontPage

[Ref.10]: Kodi for Raspian

https://kodi.wiki/view/HOW-TO:Install_Kodi_on_Raspberry_Pi#Raspbian

[Ref.11]: About OSMC

https://osmc.tv/about/

[Ref.12]: Re-mapping the keyboard with Debian Squeeze

http://elinux.org/R-Pi_Troubleshooting#Re-mapping_the_keyboard_with_Debian_Squeeze

[Ref.13]: FFmpeg: compilation guide for RaspberryPi

https://trac.ffmpeg.org/wiki/CompilationGuide/RaspberryPi

[Ref.14]: How to encode DVDs using the RaspberryPi

https://raspberrypi.stackexchange.com/questions/9290/how-to-encode-dvds-using-the-raspberry-pi

[Ref.15]: Installing Gstreamer: gst-identify doesn't show any omx codecs

https://discourse.osmc.tv/t/installing-gstreamer-gst-identify-doesnt-show-any-omx-codecs/9561

[Ref.16]: Enabling hardware h264 encoding with gstreamer on the RaspberryPi

http://theiopage.blogspot.it/2013/04/enabling-hardware-h264-encoding-with.html

[Ref.17]: How to install LXDE on raspbmc?

https://raspberrypi.stackexchange.com/questions/9311/how-to-install-lxde-on-raspbmc

[Ref.18]: Raspberry Pi USB Webcam User guide

http://www.elecfreaks.com/6084.html

[Ref.19]: Voice Command v3.0 for the Raspberry PI

http://stevenhickson.blogspot.it/2013/06/voice-command-v30-for-raspberry-pi.html

[Ref.20]: How to Control a Lamp With Voice Commands and a Raspberry Pi

https://computers.tutsplus.com/articles/how-to-control-a-lamp-with-voice-commands-and-a-raspberry-pi--mac-60825

[Ref.21]: Voice Control on the Raspberry PI

http://stevenhickson.blogspot.it/2013/04/voice-control-on-raspberry-pi.html

[Ref.22]: Raspberry Pi Voice Recognition Works Like Siri

https://oscarliang.com/raspberry-pi-voice-recognition-works-like-siri/

[Ref.23]: Building My Own Siri / Jarvis
https://cranklin.wordpress.com/2012/01/13/building-my-own-siri-jarvis/

[Ref.24]: RPi Text to Speech (Speech Synthesis)
https://elinux.org/RPi_Text_to_Speech_(Speech_Synthesis)

[Ref.25]: Dynamic DNS
https://help.ubuntu.com/community/DynamicDNS

[Ref.26]: Configuring ddclient to update your dynamic DNS at noip.com
http://www.andreagrandi.it/2014/09/02/configuring-ddclient-to-update-your-dynamic-dns-at-noip-com/

[Ref.27]: RPi Wheezy VNC
http://elinux.org/RPi_Wheezy_VNC

[Ref.28]: Using an IR Remote with a Raspberry Pi Media Center
https://learn.adafruit.com/using-an-ir-remote-with-a-raspberry-pi-media-center/overview

[Ref.29]: RC GPIO /IR remote working?
https://discourse.osmc.tv/t/rc-gpio-ir-remote-working/1612

[Ref.30]: Setting Up LIRC on the RaspberryPI
http://alexba.in/blog/2013/01/06/setting-up-lirc-on-the-raspberrypi/

[Ref.31]: Raspberry Pi Universal Remote
http://www.instructables.com/id/Raspberry-Pi-Universal-Remote/?ALLSTEPS

[Ref.32]: DVB-T USB Devices
http://www.linuxtv.org/wiki/index.php/DVB-T_USB_Devices#DVB-T_USB_Devices

[Ref.33]: Finally an Easy Way to Watch Live TV in XBMC Media Center
https://mediaexperience.com/finally-an-easy-way-to-watch-live-tv-in-xbmc-media-center/

[Ref.34]: Configuration - Channel/EPG - EPG Grabber
https://docs.tvheadend.org/webui/config_epggrab/

[Ref.35]: XMLTV project
https://github.com/XMLTV/xmltv

[Ref.36]: tv_grab_it - Grab TV listings for Italy

http://manpages.ubuntu.com/manpages/artful/man1/tv_grab_it.1p.html

[Ref.37]: Is it possible to have used DAC output and hdmi audio output simultaneously?

https://discourse.osmc.tv/t/is-it-possible-to-have-used-dac-output-and-hdmi-audio-output-simultaneously/2066

[Ref.38]: Yet one more “no space left on device” issue

https://www.raspberrypi.org/forums/viewtopic.php?t=58849&p=441657

[Ref.39]: Help with USB Audio interface

https://discourse.osmc.tv/t/help-with-usb-audio-interface/35857/4

[Ref.40]: Spinning down HDD

https://www.raspberrypi.org/forums/viewtopic.php?f=63&t=6070

[Ref.41]: External USB HDD is not spinning down

https://discourse.osmc.tv/t/external-usb-hdd-is-not-spinning-down/12171

[Ref.42]: USB mic on Linux

https://wiki.audacityteam.org/wiki/USB_mic_on_Linux

[Ref.43]: The lircd.conf file format

http://www.lirc.org/html/configure.html

[Ref.44]: Lirc has to be manually restarted after boot

https://www.raspberrypi.org/forums/viewtopic.php?f=37&t=51900&sid=f4156404a7f6cfc2cd4f1e1b4b72eef2

[Ref.45]: GPIO

https://www.raspberrypi.org/documentation/usage/gpio/README.md

[Ref.46]: TVHeadend not scanning

https://discourse.osmc.tv/t/tvheadend-not-scanning/1613/33

[Ref.47]: Tutorial: Raspbmc PVR TinyUSB2 DVB-T & TVHeadend All-in-one

https://www.raspberrypi.org/forums/viewtopic.php?t=18090

18.Pictures