|  |
| --- |
| Microsemi |
| Setting Up Raspberry Pi as Kodi Media Centre |
|  |
|  |
| **Verma, Shally** |
| **9/12/2016** |

|  |
| --- |
|  |

Contents

[Introduction 1](#_Toc467774890)

[Purpose 1](#_Toc467774891)

[Reference 1](#_Toc467774892)

[Development Platform 1](#_Toc467774893)

[Setting up Raspberry 1](#_Toc467774894)

[Testing with Kodi 2](#_Toc467774895)

[Testing record and playback on TW1 3](#_Toc467774896)

[Testing 2 TW device at Raspberry SPI CS0 and CS1 3](#_Toc467774897)

[Troubleshooting 4](#_Toc467774898)

# Introduction

Kodi (previously known as xbmc) is an open source Media Centre available for various Operating Systems. Kodi is available as stand-alone installer OR as dedicated Media Centre package like OpenElec, LibreElec.

This instruction guide will start with standard raspbian image using stand-alone kodi installer.

# Purpose

This document is a step-by-step instruction on how to set up Kodi on Raspberry pi using a raspbian image which is not pre-configured for any kodi set up. It will also cover use instructions of Microsemi Voice Control Kodi Add-on to test Microsemi VPROC Devices ASR feature.

# Reference

[1] <https://kodi.tv/>

# Development Platform

|  |  |
| --- | --- |
| Hardware | Raspberry Pi 3 Model B V1.2 |
| Raspbian Image |  |

# Abbreviations

|  |  |
| --- | --- |
| ASR | Automatic Speech Recognition |
| VPROC | Voice Processing |

# Setting up Raspberry

Before you set up raspberry board for kodi, you must have HDMI TV, keyboard and Mouse connected to your raspberry setup.

1. If your raspberry is not pre-configured, get the raspbian image from <https://www.raspberrypi.org/downloads/>. Write image on SD-Card and boot Pi. You may use Win32DiskImager or See instructions here <https://www.raspberrypi.org/documentation/installation/installing-images/README.md>

If asked, enter following login information on boot:

raspberrypi login: pi

Password: raspberry

1. Get Microsemi Vproc SDK sources copy on to your raspberry pi /home/pi directory. Please follow ZLS38100\_SDK\_Build\_Guide\_Raspberry.pdf for native compilation of Microsemi VPROC SDK on pi.
2. A successful build should generate \_sr.so and \_sr.pyc in /home/pi/sdk/libs directory. These are primary import modules from kodi.
3. Kodi add-on is located in /home/pi/sdk/thirdparty/kodi/Kodi-Addon/service.xbmc.voicecontrol
4. Open service.py and make sure sys.path.append points to location where voice control module (\_sr.so and \_sr.pyc) is present. Example, if they are present at /home/pi/sdk/libs directory in SDK directory then following in service.py should be set

sys.path.append('/home/pi/sdk/libs')

**Do not** alter this as Kodi add-on imports voice recognition module \_sr.so from this location.

1. Alternate option is move \_sr.so and \_sr.pyc to standard /usr/lib or to path searched by python package for import modules
2. Now zip this add-on to create service.xbmc.voicecontrol.zip. if you don’t have zip install on pi, then run

sudo apt-get install zip

zip –r service.xbmc.voicecontrol.zip service.xbmc.voicecontrol/

# Testing with Kodi

1. Before you start, make sure on these points:
   1. Make sure Microsemi DAC drivers (HBI and Sound) and overlays are applied and enabled (See section below [Appendix A](#_Appendix_A_Making) And [Enabling 2 TW Devices](#_Testing_2_TW) to see help on this)
   2. By default, our kodi voice command add-on (service.xbmc.voicecontrol.zip) reads voice commands from TW1 device. So, after boot up make sure TW1 has 067 in it
   3. If you wish to change Add-on to read from TW2, change ASR\_DEVICE=2 in Kodi Add-on (detailed steps given in further section).
   4. Current Voice Control Add-on use TW1 as ASR device. If you wish to change, then go inside service.xbmc.voicecontrol directory in this image (available at svn as well) and change following in service.py

ASR\_DEVICE = 2 /\* to use TW2 \*/

ASR\_DEVICE=1 /\* to use TW1 \*/

And zip it using ‘zip’ utility on pi.

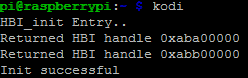
sudo apt-get install zip

zip –r service.xbmc.voicecontrol.tar.gz service.xbmc.voicecontrol/

* 1. Note Raspberry pi IP Address. Kodi has issue when it is exited. Sometimes screen turn blank. In such case, we can use **putty** utility to do ssh login and run kodi from command shell.
  2. If you run into any issue, refer to **Troubleshooting** section for help

1. Run Kodi
   1. If you have Raspberry pi UART console or want to run through command shell. Open Terminal window and type in *Kodi*

A successful run should show you message on console as in snapshot (if 2 devices are used). If you run into error, check **Troubleshooting** section for help.



* 1. Else, Goto Raspberry Desktop Menu->Sound&Video->Kodi Media Centre and click on it (Please note exact path may change as per user raspbian distro).
  2. Go to System->Settings->Add-On->My add-ons-> Services->Voice Control.

You should see ‘Voice Control’ as ‘Enabled’.

1. Start testing with following command “Alexa” ,”Volume up”, “Volume Down”
2. If you want to do some modification to current add-on and re-install it
   1. First Uninstall Add-on
      1. Goto System->Settings->Add-On->My Add-on-> Services->Voice Control
      2. Click to enter it
      3. Click ‘Disable’ followed by ‘Uninstall’
      4. Make sure both devices are closed by checking /proc/hbi directory. It should not have dev\_00 or dev\_01 directory. If it is, then you can also close them using procfs command as mentioned below in Troubleshooting section
   2. If you wish to do some change in Add-on follow this step else skip to next
      1. Go to service.xbmc.voicecontrol directory in this image (you may also get same from svn repo or unzip existing service.xbmc.voicecontrol.zip)
      2. Do your modification
      3. Install zip utility on your local raspberry machine using

sudo apt-get update

sudo apt-get install zip

* + 1. Run zip command

zip -r service.xbmc.voicecontrol.zip service.xbmc.voicecontrol/

* 1. Re-installing Add-on
     1. Exit Kodi
     2. Run kodi again
     3. Goto System->Settings->Add-On->Install from zip file -> Home Folder.

Select service.xbmc.voicecontrol.zip

* + 1. You should see “voice control enabled” message on bottom right corner.
    2. It is installed. Now make sure it is enabled as well.

Goto System->Settings->My Add-on->Services->Voice Control

It should be showing as ‘Enabled’. If disabled, click on it to enter Add-on description and click ‘Enable’.

* 1. Now Go to home screen and test your commands.

# Testing 2 TW device at Raspberry SPI CS0 and CS1

1. Make sure Microsemi Device Overlays are applied and enabled

sudo boot/config.txt

dtparam=spi=on

dtoverlay=microsemi-spi-multi-tw-overlay

microsemi-spi-multi-tw-overlay enables both cs0 and cs1 of raspberry pi to be used by Microsemi DAC

You need to reboot system after doing this change.

1. Also before issuing make hbilnx (to build HBI driver) , following make variables set to

HBI\_MAX\_INSTANCES=2

VPROC\_MAX\_NUM\_DEVS=2

1. Driver hbi.ko is loaded
   1. Manually sudo insmod /home/pi/sdk/libs/lib/modules/ <kernelversion>/extra/hbi.ko
   2. Or boot time See [Appendix A](#_Appendix_A_Making) for boot time enable of Microsemi spi/i2c device
2. Run following HBI procfs commands(Initialize, open, read):
   1. cat /proc/hbi/init\_driver
   2. echo 0:0 > /proc/hbi/open\_device /\* bus num 0, chip select 0 \*/
   3. echo 0:1 > /proc/hbi/open\_device /\* bus num 0, chip select 1\*/
   4. Read registers. Example read 10 bytes from 0x200 register from device 0

echo 200 10 > /proc/hbi/dev\_00/read\_reg

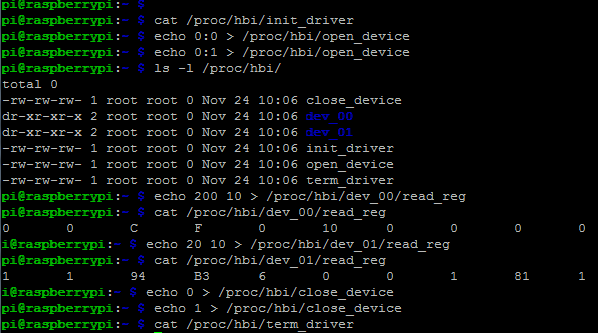
* 1. Check read registers cat /proc/hbi/dev\_00/read\_reg
  2. Close devices 0 and 1 and terminate driver

echo 0 > /proc/hbi/close\_device

echo 1 > /proc/hbi/close\_device

cat /proc/hbi/term\_driver

Example Screenshot



# Testing record and playback on TW1

* 1. Make sure to check device overlays are enabled:

microsemi-dac-overlay.dtbo copied to /boot/overlays and

sudo /boot/config.txt

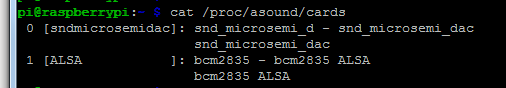
dtparam=i2s=on

dtoverlay=microsemi-dac-overlay

* 1. Make sure TW1 has 2-way communication firmware loaded.
  2. Testing ALSA Playback
     1. Check Microsemi sound card is recognised by kernel. Run command

cat /proc/asound/cards

Should have snd-soc-microsemi-dac listed as one of the card. Here’s is example snapshot:



If TW1 device has firmware supporting 2-way communication, then you may do aplay and arecord example run.

Example commands for case above where sndmicrosemidac is registered as card 0:

arecord -D "hw:0,0" -c 2 -r 16000 -f S16\_LE -t wav test.wav

aplay -D "hw:0,0" test.wav

if its registered as card 1

arecord -D "hw:1,0" -c 2 -r 16000 -f S16\_LE -t wav test.wav

aplay -D "hw:1,0" test.wav

* 1. Testing Kodi Playback

**Kodi uses only uses ALSA Default device** . Thus it is required that sndmicrosemidac is registered as default Card 0 to linux kernel system.

To make sndmicrosemidac as default ALSA soundcard at boot time, refer to [Appendix A](#_Appendix_A_Making)

# Troubleshooting

Please update this section as you come across any issue not covered here.

Q. Blank Screen after kodi exited

1. It is raspberry pi kodi issue. You can use UART console Or run putty to login to pi and run ‘kodi’ from command line.

Q. Getting error “addon structure incorrect”

A. Either zip is created using windows zip utility OR add-on was enable and user disable ,uninstall and trying to re-install add-on.

If you have zipped your add-on using windows utility and copy it over to linux raspberry machine, then it may throw above error. Solution is install zip on your local machine(where kodi is installed) using ‘sudo apt-get install zip’ and then run ‘zip –r <output zip filename> <add-on source directory>

If you follow above process and still see issue then read on..

Kodi has issues with reinstallation of zipped version of add-on. If you have an add-on installed and you did some modification, rezip it and uninstalled previous one so that new changes can be loaded. please exit kodi completely and restart a process of installing add-on from zip

Q. Voice Control Add-on shows “Enabled” still no action on Voice Command

A. Possible reasons:

How to check that ASR device is not opened:

1. cat .kodi/temp/kodi.log

if you see log error “err in sr.init()”, if yes then either device is in use by some other application OR owner permission are not set

1. check owner by running ‘ls –l /dev/hbi’ , it should be pi, pi
2. if owner is correct, run dmesg to check for error messages thrown by driver
   1. example, device already opened
   2. run cat /proc/hbi , if its has dev\_xx directory that is someone is already using it. close that app Or close this device using proc commands like
      1. echo 0 > /proc/hbi/close\_device
      2. cat /proc/hbi/term\_device

# Appendix A Making sndmicrosemidac as default ALSA Device

Please note this procedure been tested on raspbian version

Linux raspberrypi 4.4.19-v7+ #906 SMP Tue Aug 23 15:53:06 BST 2016 armv7l GNU/Linux

For any other raspbian distribution package, basic steps should still be same however there may be change in file paths or names. In case, you don’t see filesnames and paths as in description below, you are please recommended to search raspberry forums on help how to load and enable external modules for your raspbian image.

To enable sndmicrosemidac as default ALSA Device at system boot, user need to copy snd-soc-zl380xx.ko and snd-soc-microsemi-dac.ko in /lib/modules/`uname -r`/kernel/sound/drivers and modify /etc/modules to add following:

snd-soc-zl380xx

snd-soc-microsemi-dac

Here is the detailed procedure below to do that.

1. Make sure Microsemi Device Overlays are applied and enabled

microsemi-dac-overlay.dtbo copied to /boot/overlays and

sudo /boot/config.txt

dtparam=i2s=on

dtoverlay=microsemi-dac-overlay

1. Edit /etc/modules to add the name of the modules to load in it. Type

sudo nano /etc/modules

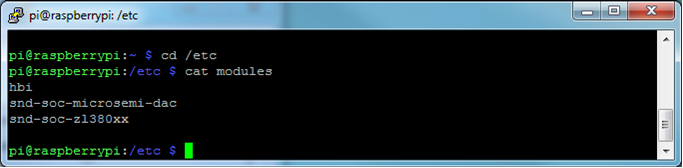
Add the lines below into it and then save it by issuing a [ctrl+x] , then Y enter

hbi

snd-soc-microsemi-dac

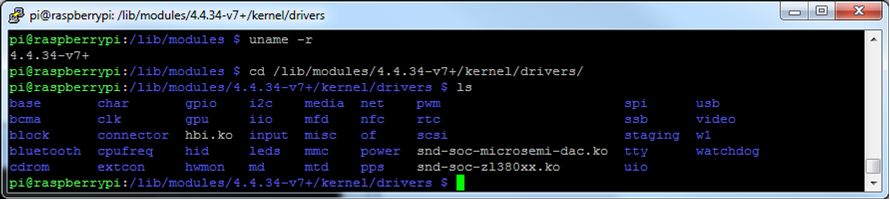
snd-soc-zl380xx

If you do a cat of the file you should see



2- Copy the 3 modules into /lib/modules/`uname –r`/kernel/drivers

Note: `uname –r` is current active kernel version on your platform and variable automatically picks current active linux kernel directory



Note: You can put the modules actually anywhere within /lib/modules/`uname –r`/kernel since modprobe will be looking for the whole /lib/modules/ path to find modules.

1. Copy the /etc/modules config file into /lib/modules/
2. Then issue the command below

sudo depmod -a

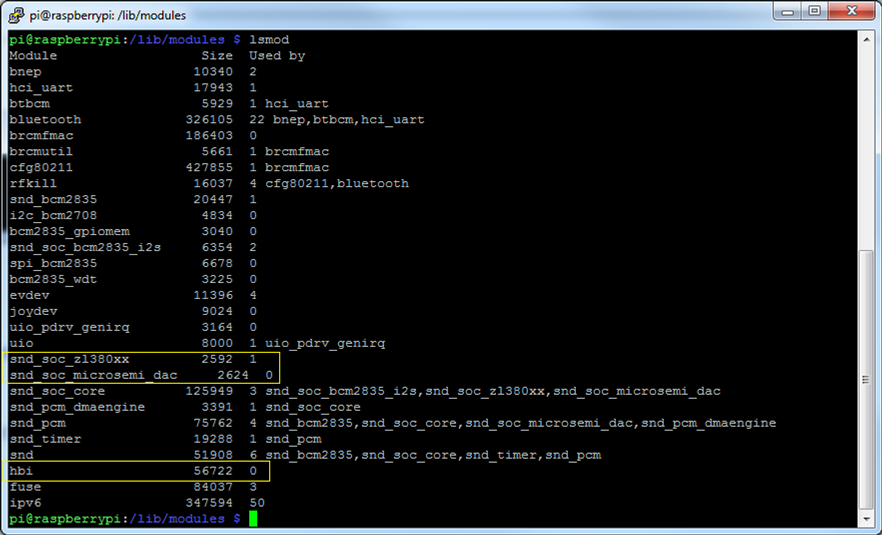
If you do a listing of /lib/modules you should see



5- Reboot

At power up all 3 modules should automatically loaded. You can confirm by issuing an

Lsmod



**Note**: now that the modules are auto loaded, then the Microsemi sound card will appear at card0 [This also require that Device Overlays are applied and enabled]

