**MP3 Player Using Build-root**

To start with this project you should have basic information about:

1- The Linux administration, that you can learn more about from here:

<https://bootlin.com/docs/>

2- Build-root, that you can learn more about from here:

<https://buildroot.org/downloads/manual/manual.html>

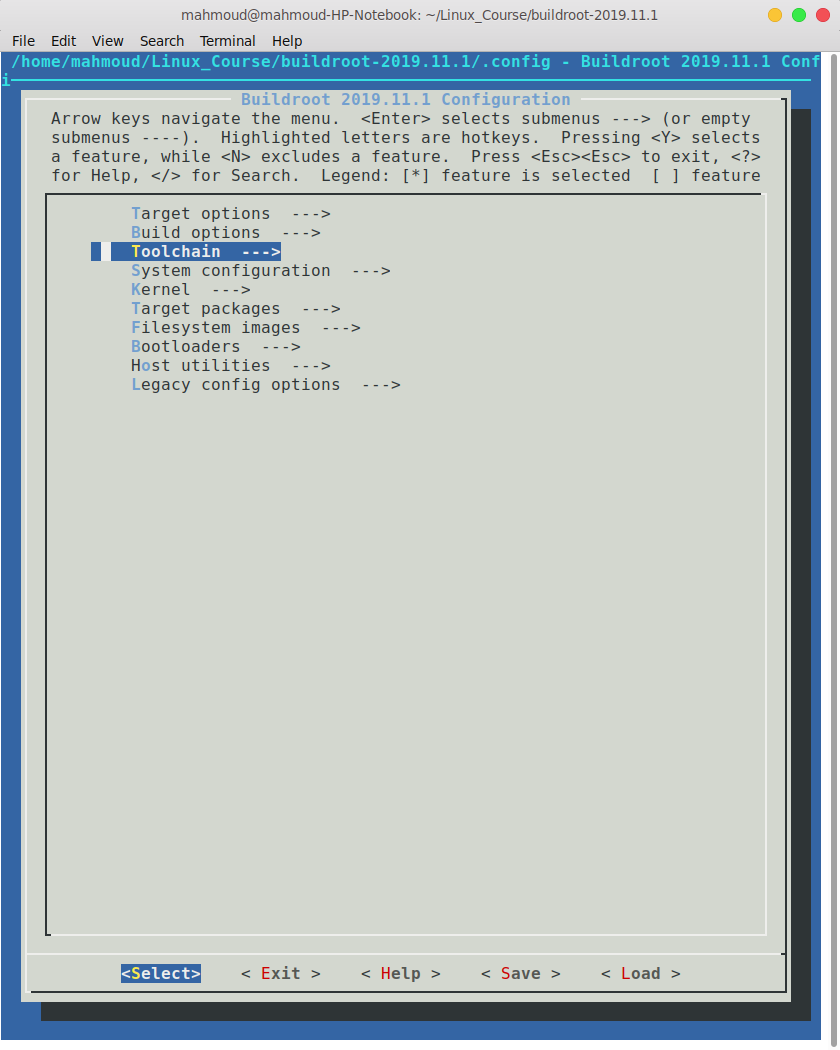
3- Shell scripting, that you can learn more from here:

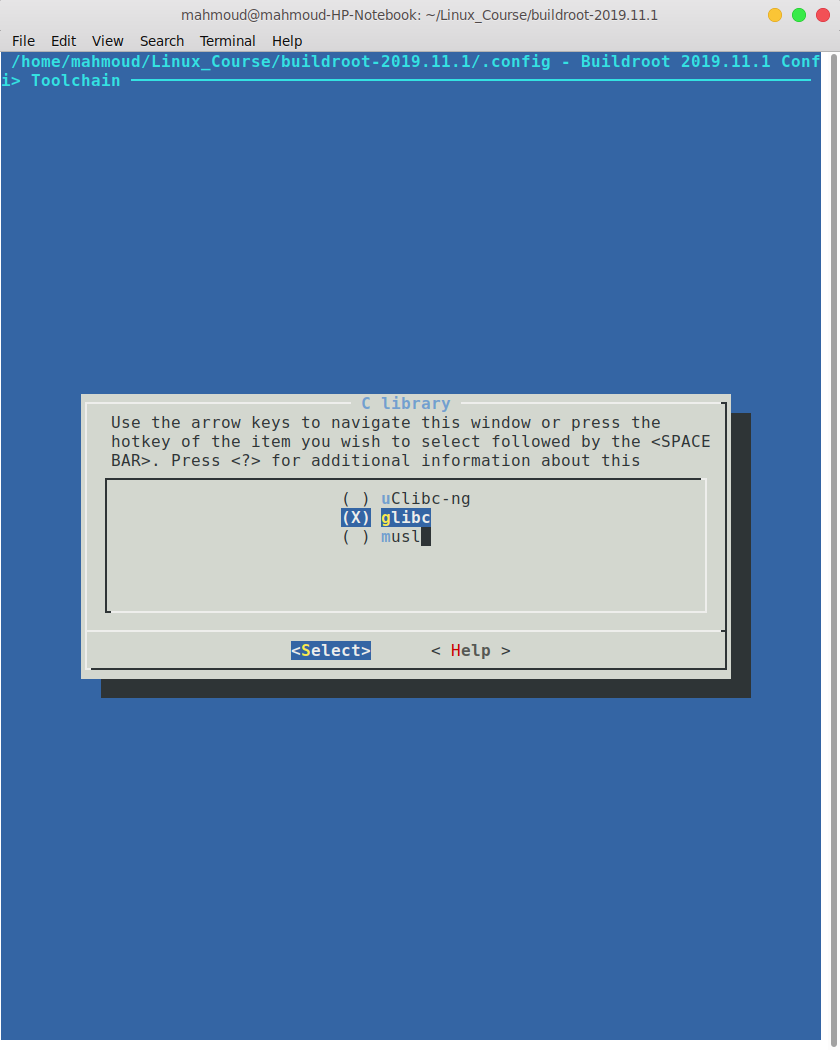
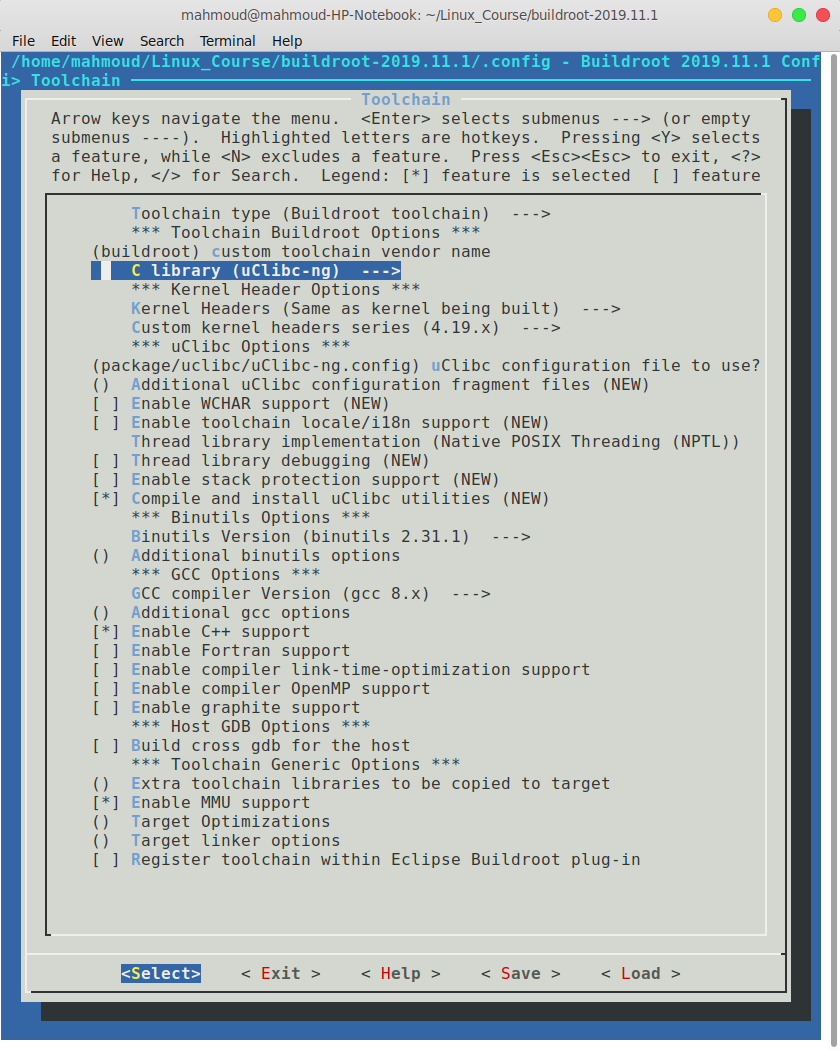
<https://www.youtube.com/watch?v=hwrnmQumtPw>

Now that you know all of these, let us start.

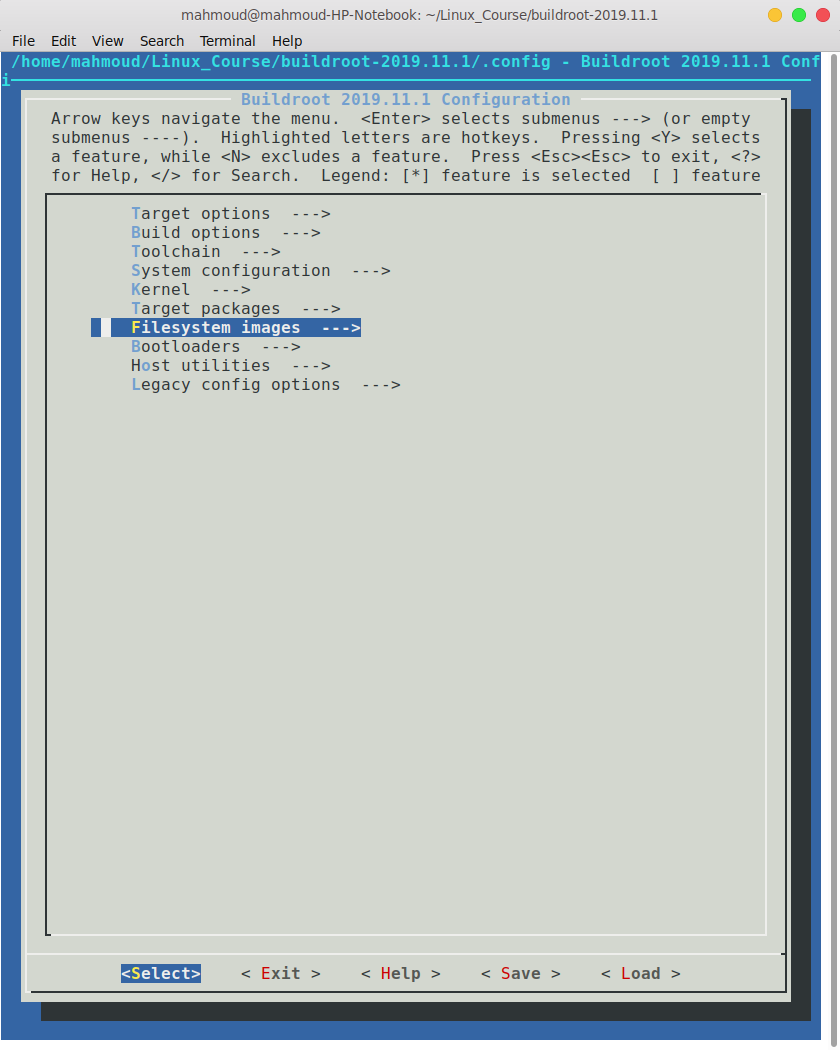
First of all, we need a way to communicate with our raspberry pi, therefore we had to enable the ssh using an overlay to overwrite, the interfaces and sshd\_config, both found in */etc/network, and etc/ssh* respectively *(Check the overlay folder above).*

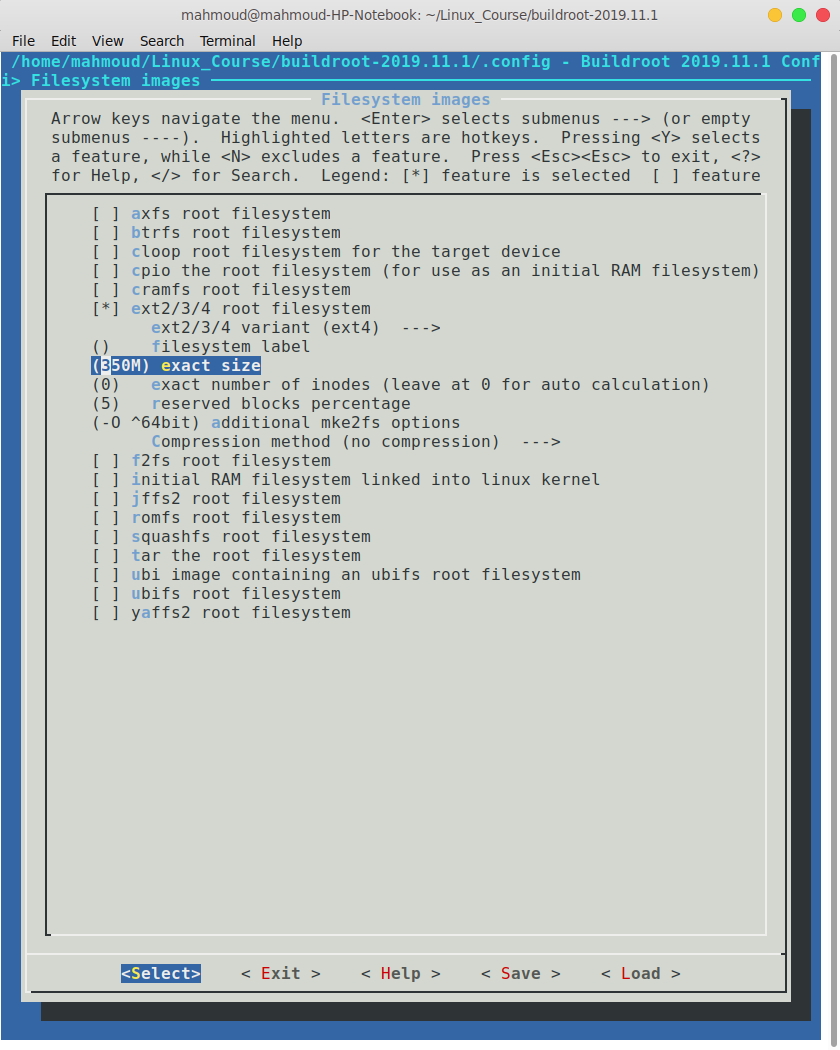
Second of all, you need to change the tool-chain of your build-root from Ulibc to glibc. Open a terminal in your build-root folder and then write the command “make menuconfig”, and follow the upcoming steps:





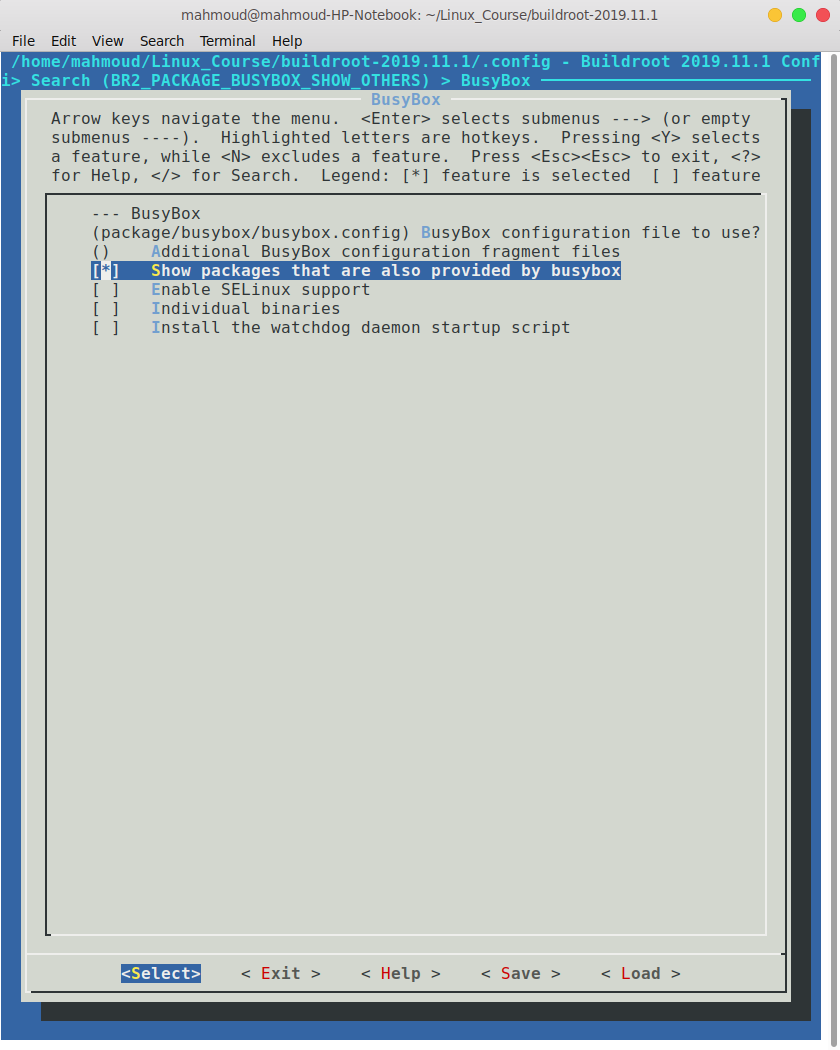
The bigger library is needed to be able to listen to music.

You also need to change the size of the image to 300M:

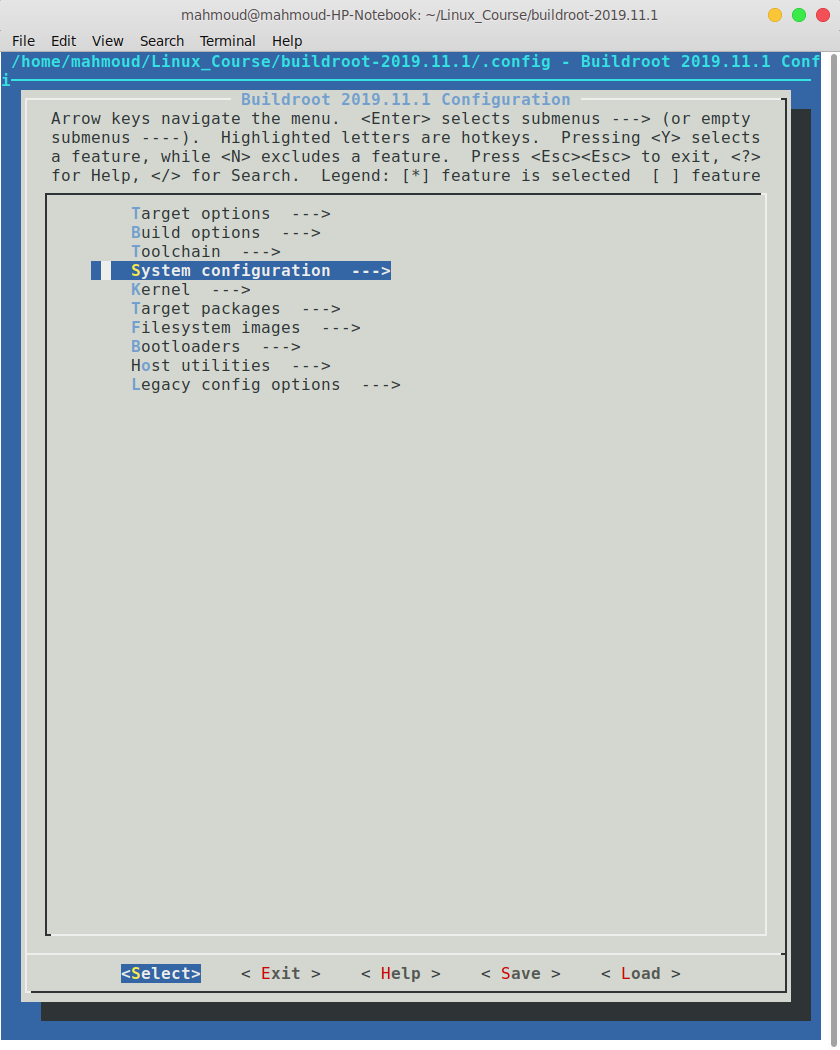


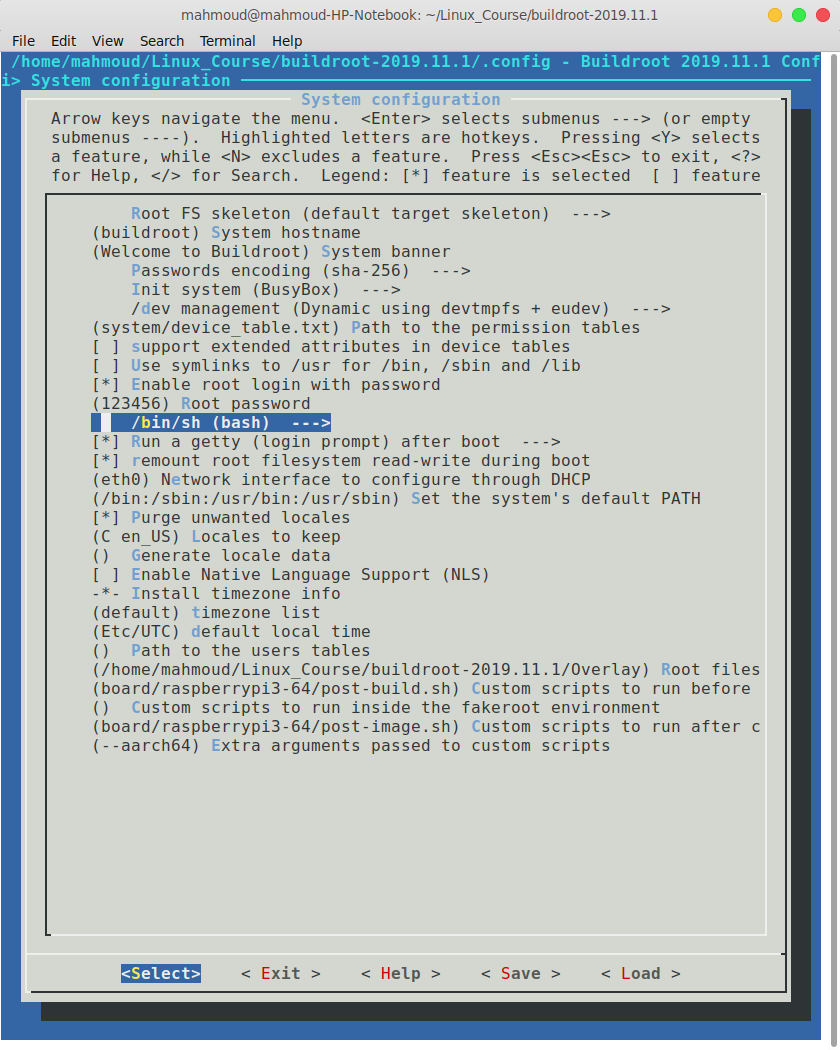
One last thing and your ready to go is to add the bash to your build-root because it is sh by default. Press forward slash “/” and search for “**BR2\_PACKAGE\_BUSYBOX\_SHOW\_OTHERS” (**no spaces, use underscore**),** press 1

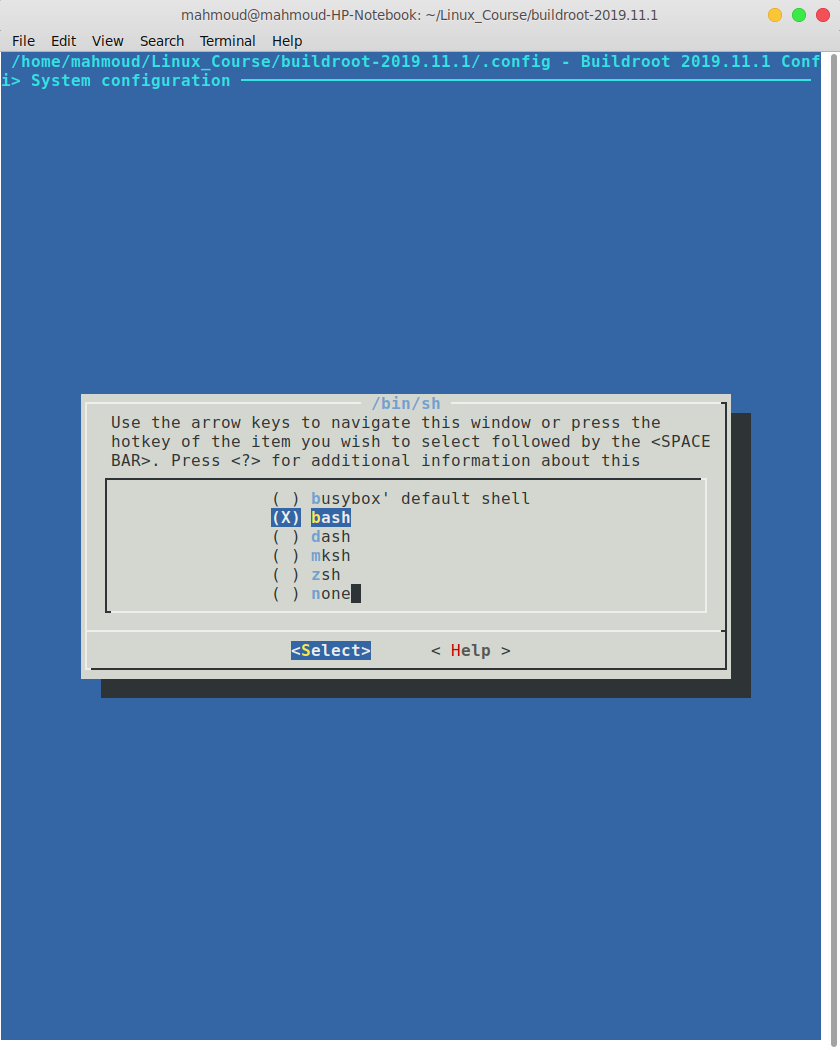
choose the highlights option:



then go back home and follow those steps:

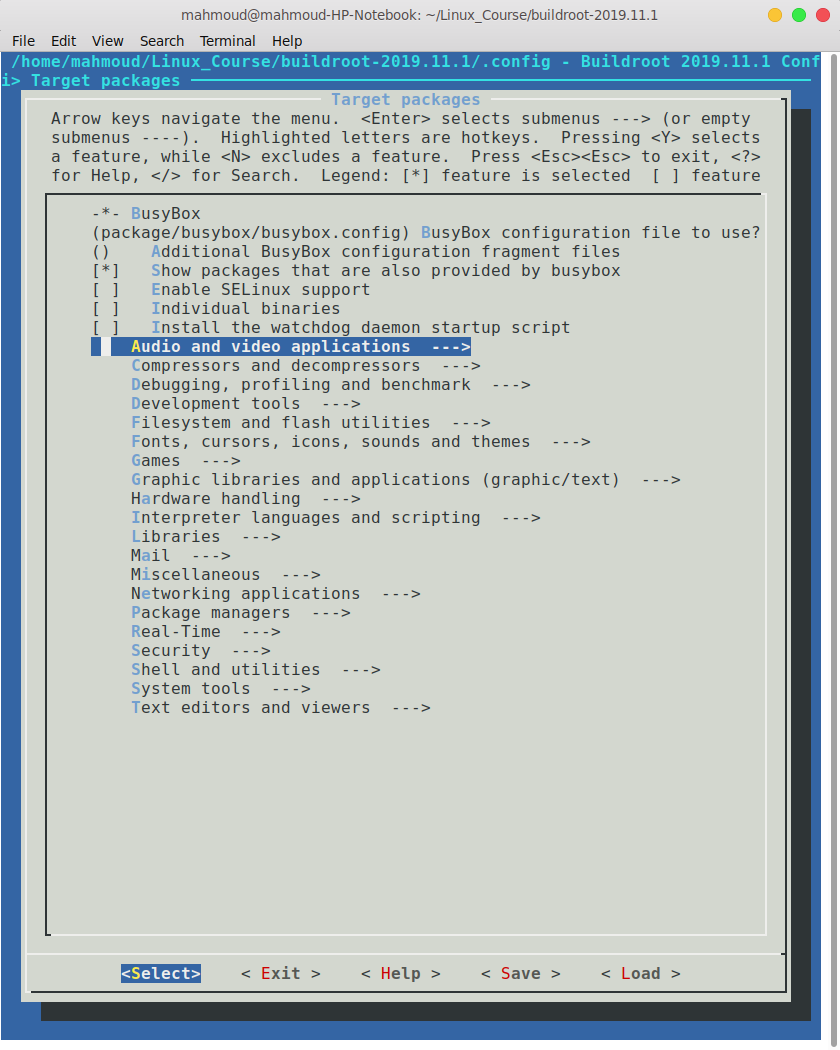
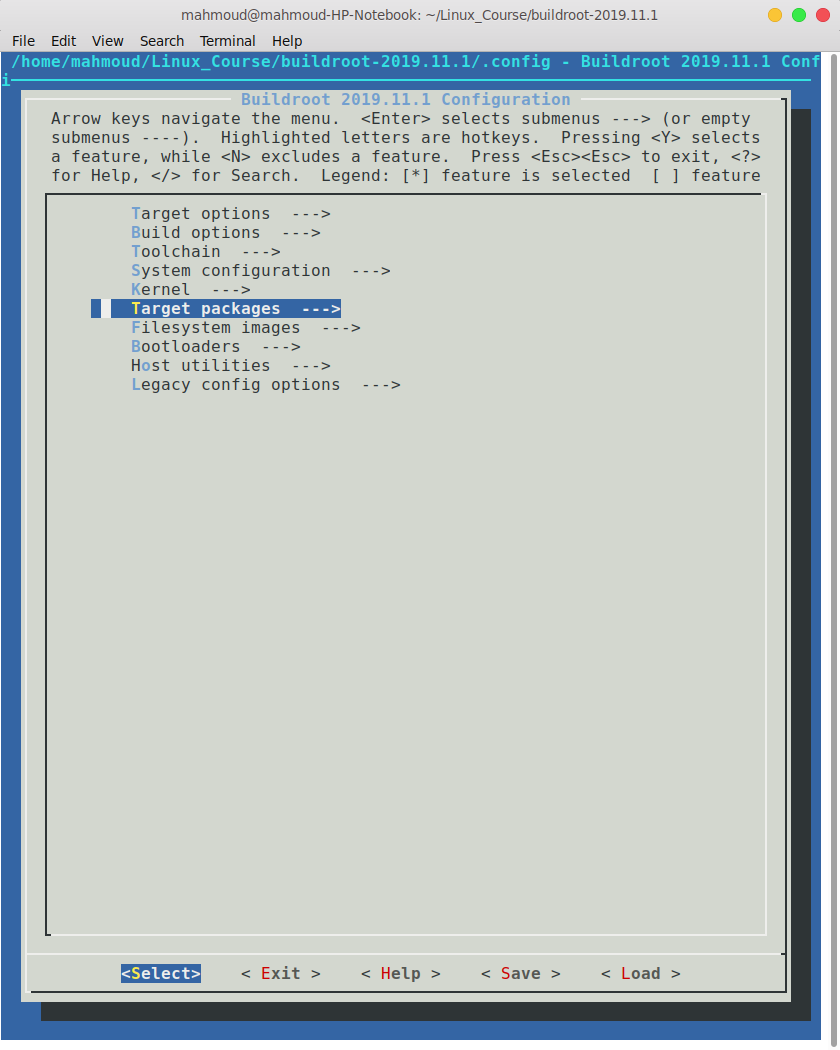


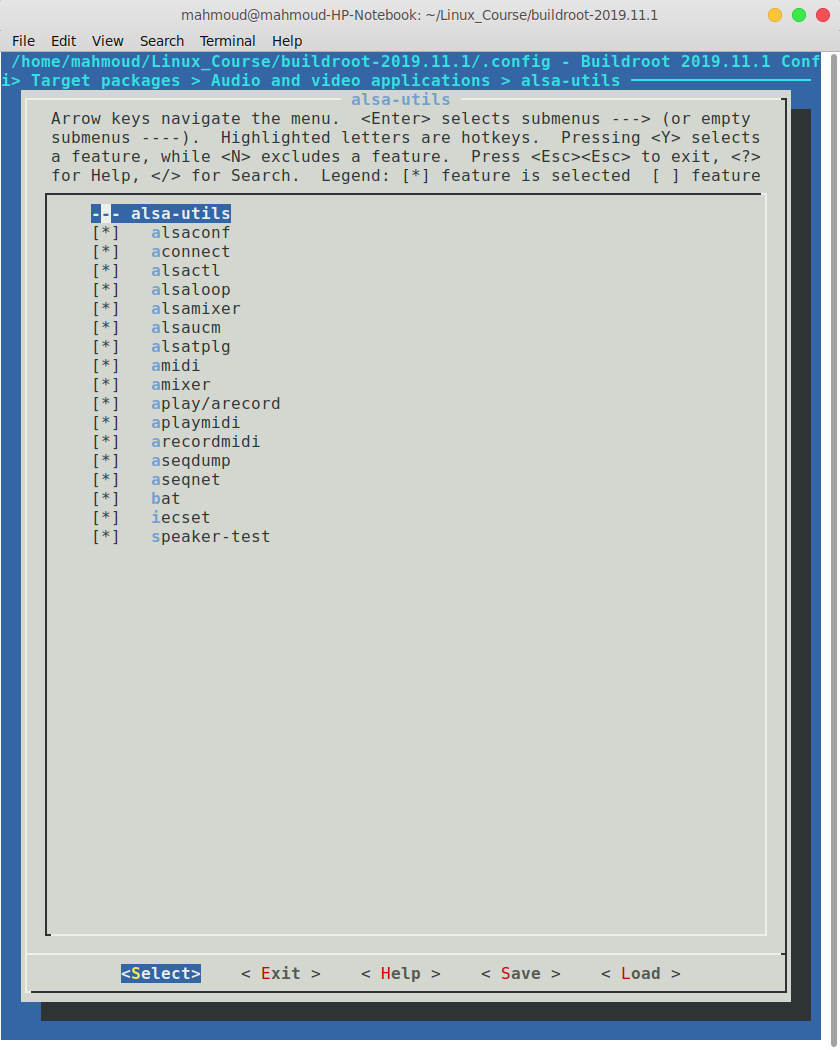
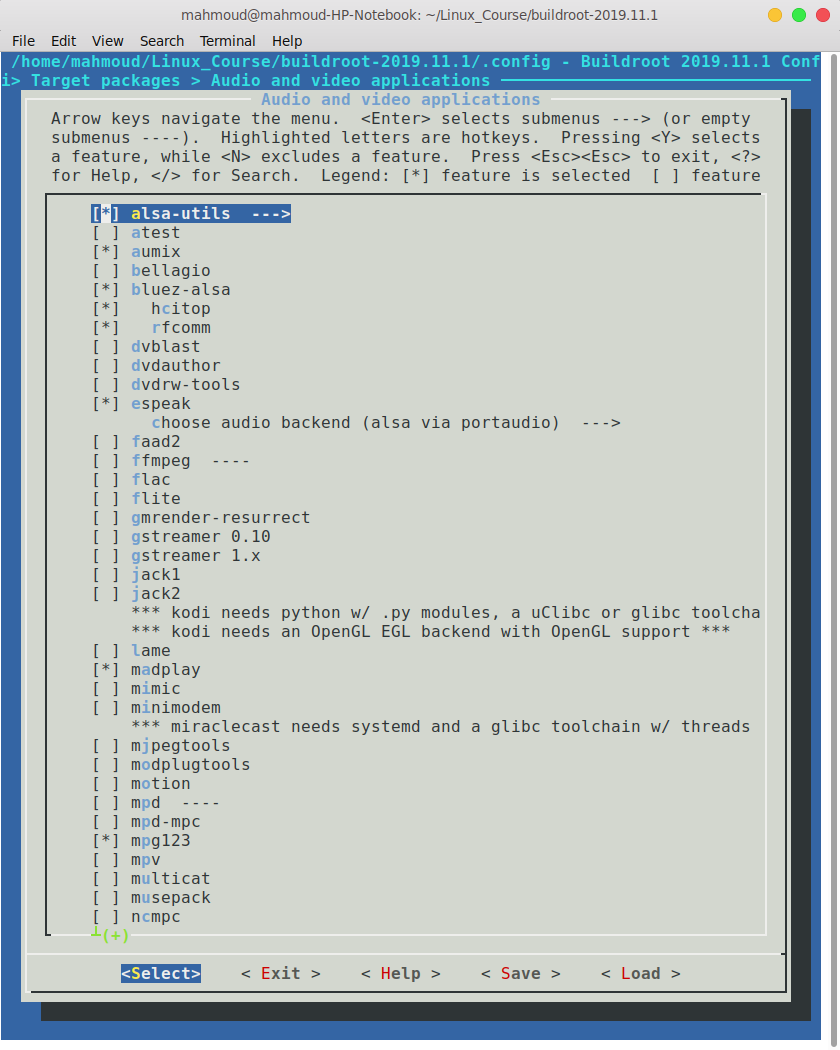


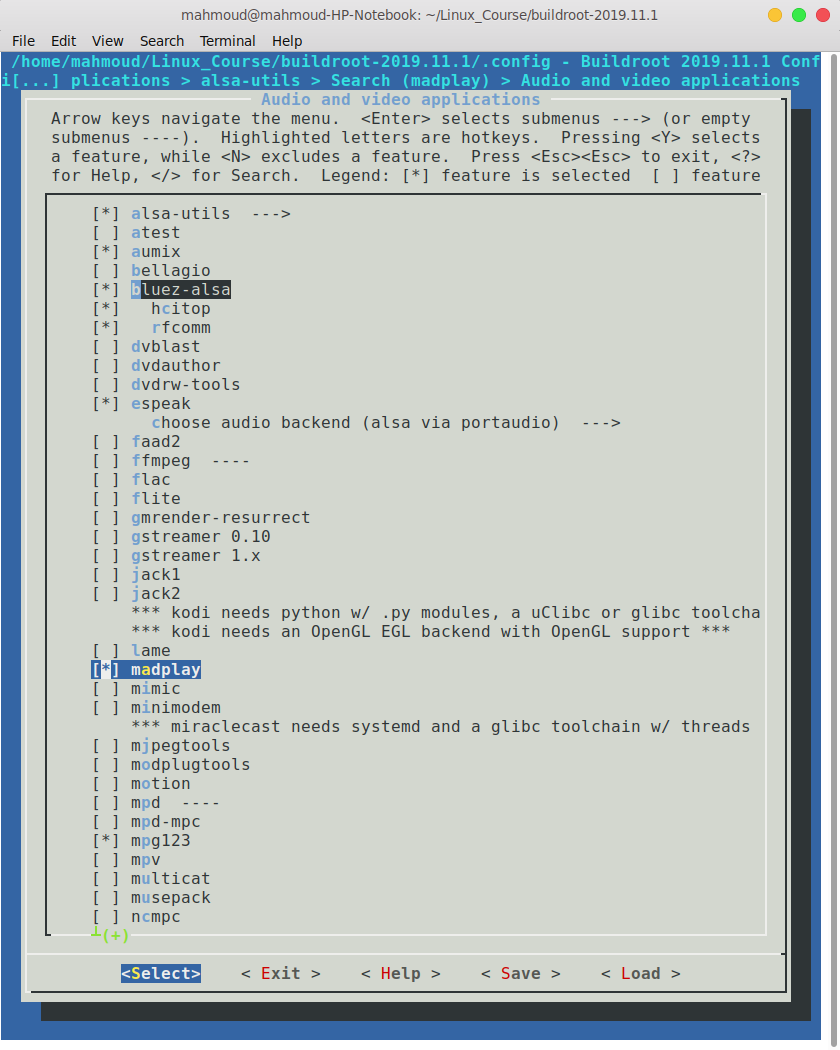


How do you enable the music player you ask? Follow the next configurations:

open menuconfig

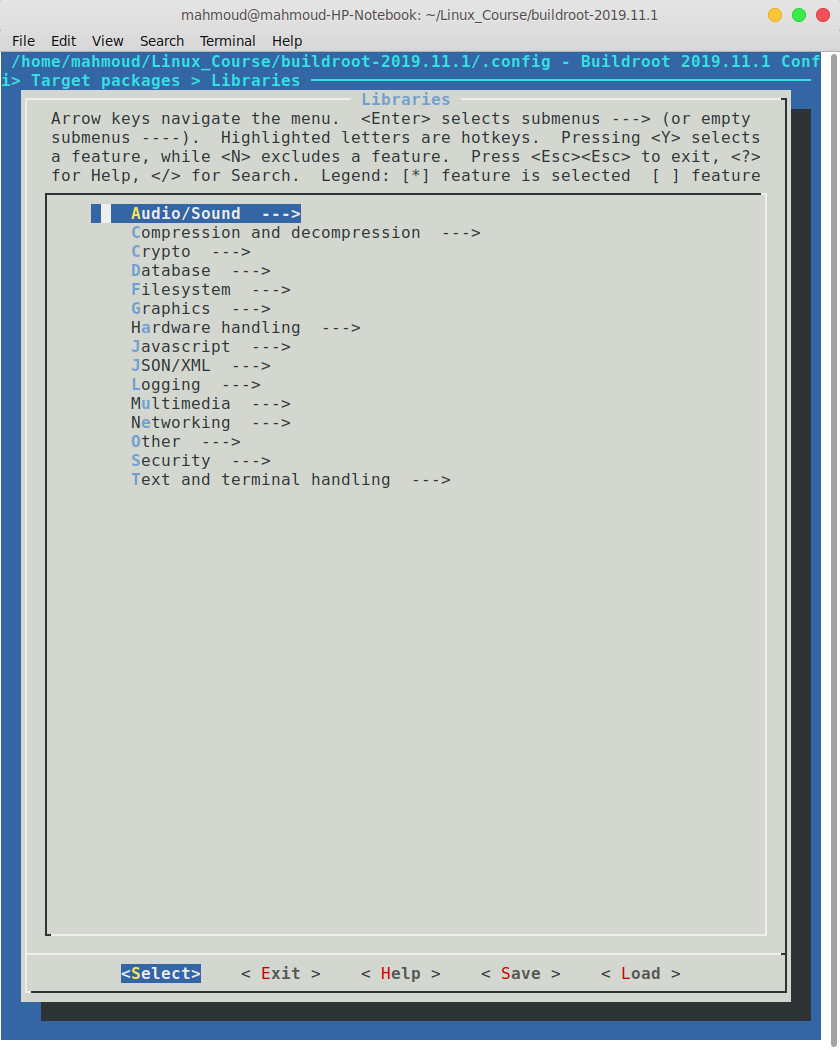
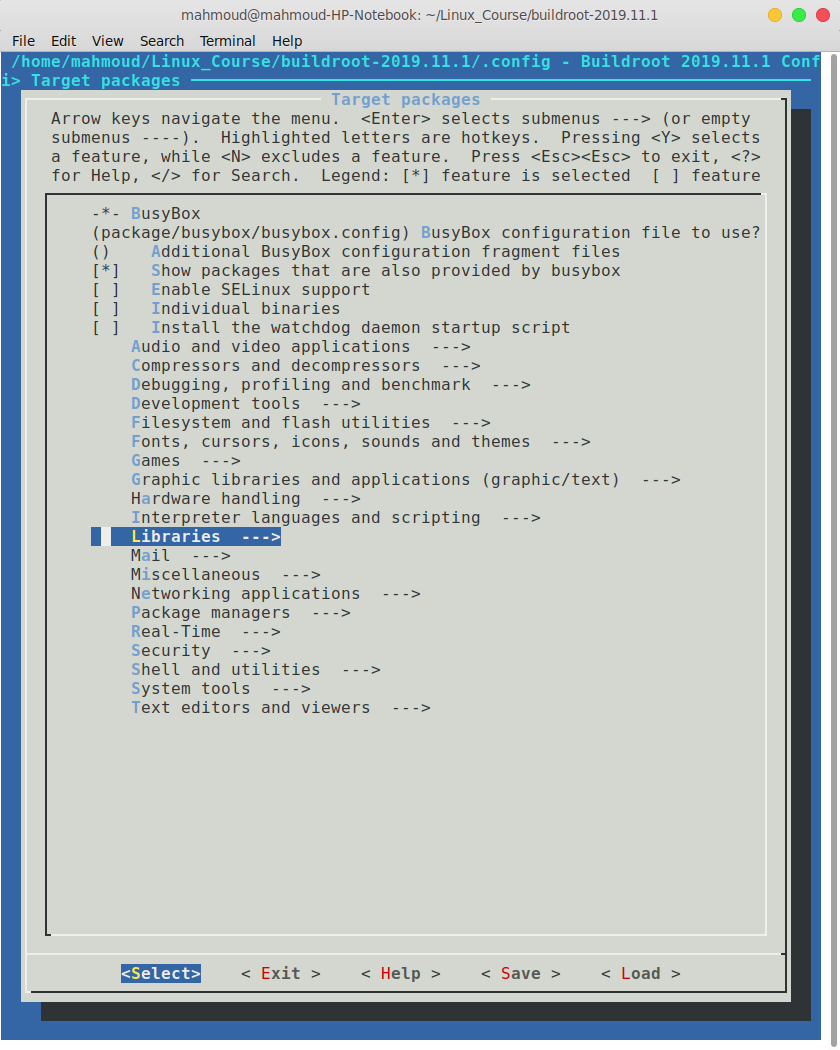


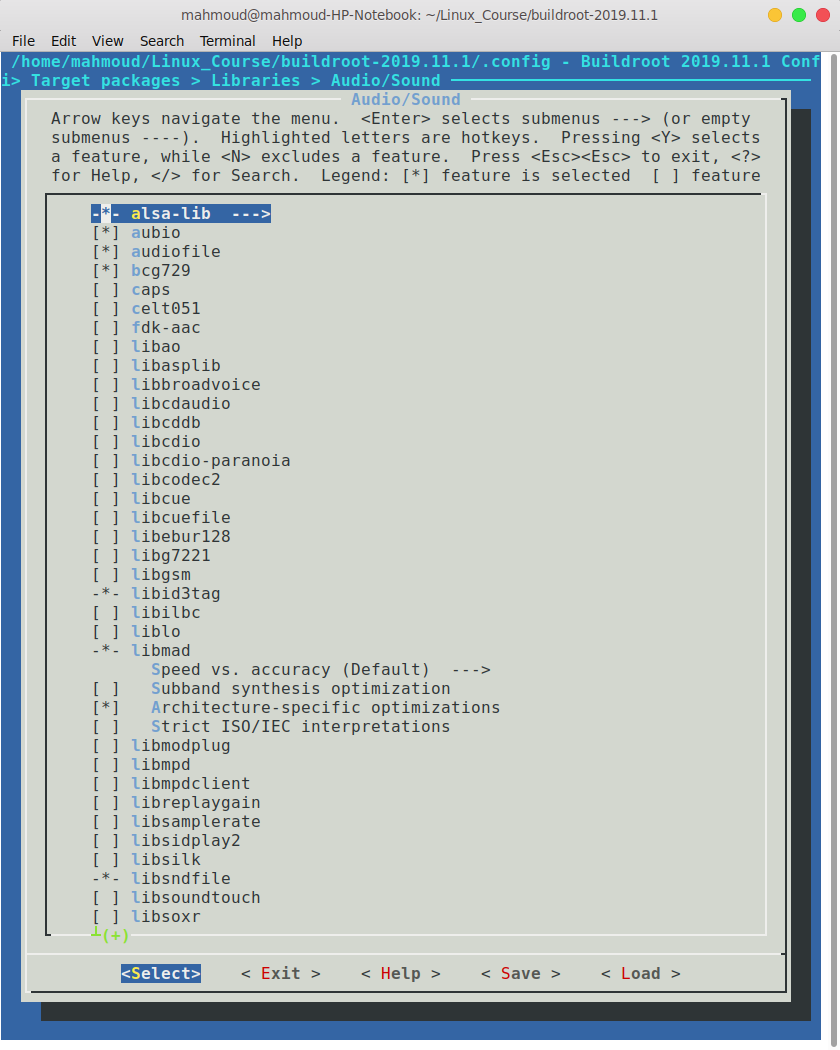


 Those are the needed sound drivers. However, to add music player applications to be able to play music. We are mostly using madplay, but mpg123 is a very good option as well. While you are in the menucofig, press forward slash “/”, write madplay in the search and press enter, press “1”, and you will be taken to there automatically.

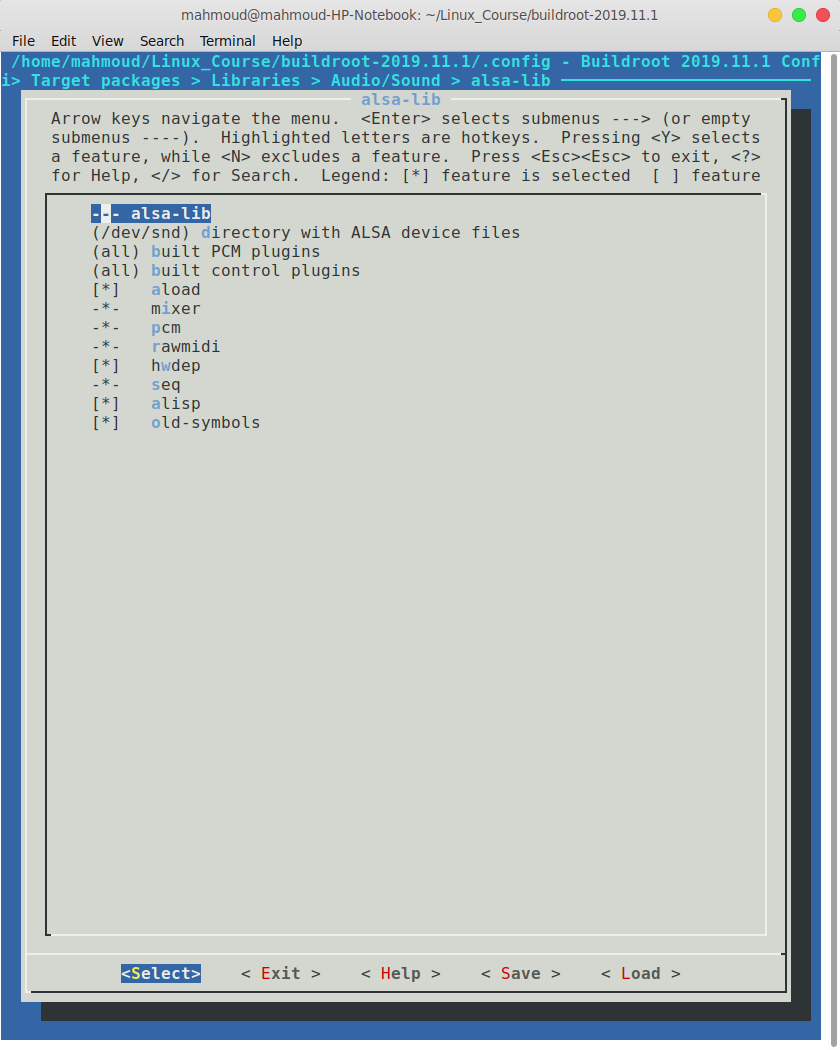
From here select madplay and mpg123. Also select aumix (adjust audio mixer) and pulseaudio (is a general purpose sound server intended to run as a middleware between your applications and your hardware devices), this is the mixer that is needed by the applications.

You still need some libraries for the alsa. In Target packages





here select the first four and then open alsa-lib and select all:



go back to target packages> libraries>Audio/Sound and select the following as well: Libid3tag, libmad, Architecture-specific optimizations, libsndfile, portaudio, alsa support , C++ bindings , speex.

We are now done with the menuconfig for the audio part, and we need to go to another type of configurations. In your build-root folder, open packages and search for “rpr-firmware”. Open config.txt, and add those four lines at the end

dtparam=audio=on

dtparam=audio=on #to be able to use 3.5 mm AUX cable

hdmi\_safe=1 #to be able to use HDMI after the pi is turned on

hdmi\_drive=2

Now a small change in the busybox, for shuffling and sharing. Go to:

package/busybox/busybox.config, and change the CONFIG\_SHUF and the CONFIG\_SHRED to be:

CONFIG\_SHRED=y

CONFIG\_SHUF=y

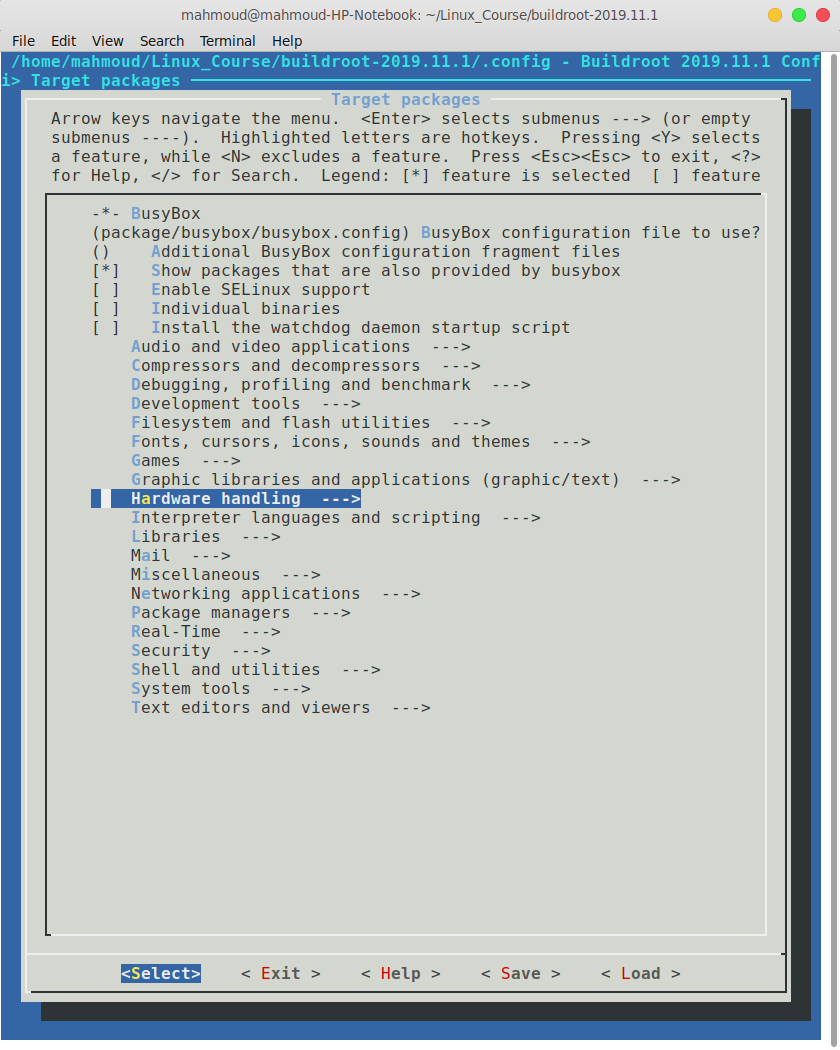
make sure that they are not commented and that there are no spaces.

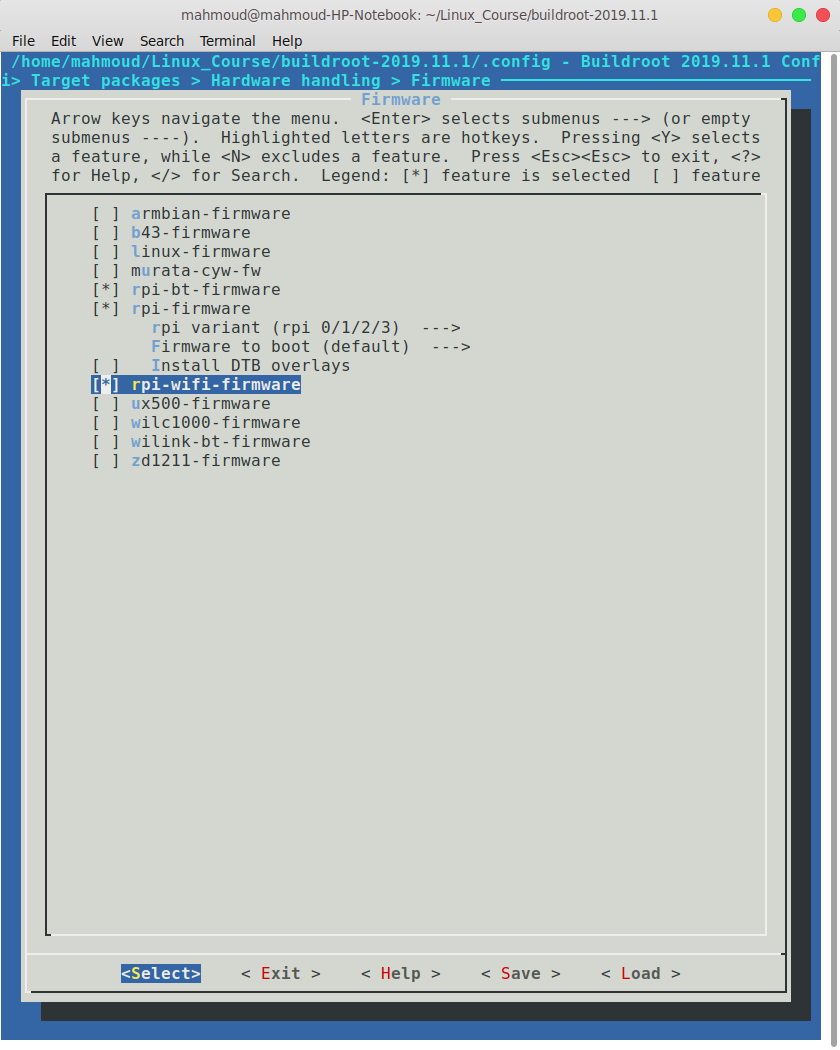
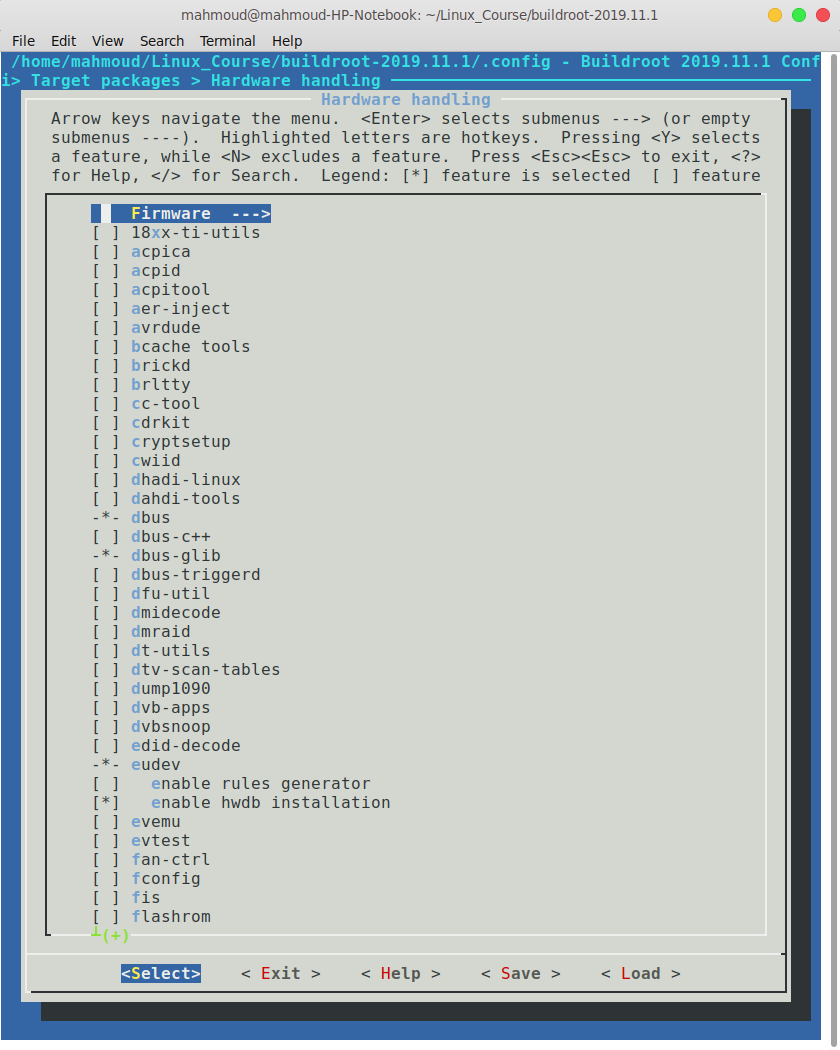
Lastly, in your overlay add the profile file in /*etc* and add the next line:

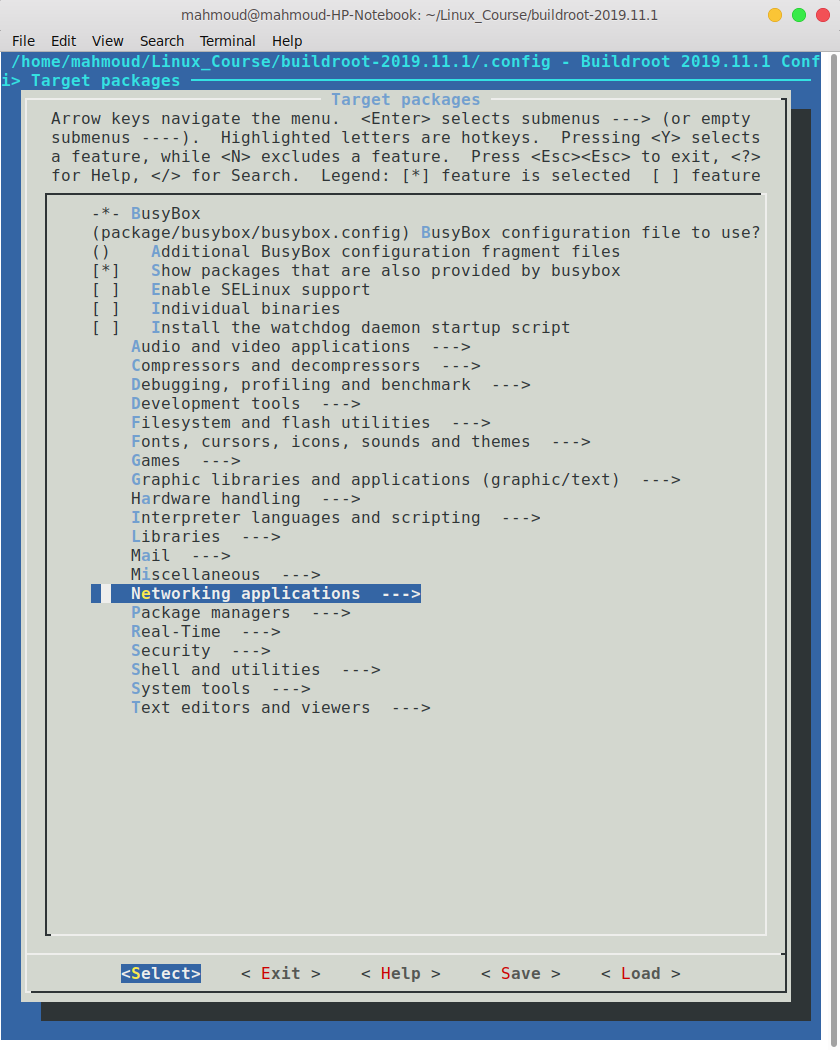
modprobe snd-bcm2835

Note that: I have put this in my init.d folder, both of them will work, the init.d will be explained later on in this paper.

Now you can play any song by connecting your pi over ssh and connecting and aux cable, but what if you want to enjoy this wireless life? For this we are going to need wifi and bluetooth. First let us talk about wifi. This is two simple steps, in target packages>





Go back to Target packages>

in Networking Apps select: ifupdown scripts, openssh and wpa\_supplicant

lastly add wpa\_supplicant.conf to your overlay (Check it out in our overlay)

Note that, we have not been able to send sounds over Bluetooth, however you can follow this link that says how it should work. If you were able to send music over Bluetooth please let us know:

<https://tewarid.github.io/2014/10/29/bluetooth-on-raspberry-pi-with-buildroot.html>

If you want to use buttons, detect the addition or removal of devices like flash drivers, you should make some scripts that do so, and let them run with the init process. To do the latter, you should create a script and place it in the */etc/*init.d and name it SNN-XXXXX.. where N is a number and the Xs should be replaced by the name of this script. Note that the lower the number the sooner your script will run in the process, therefor you should put a high number to make sure that it is running after everything has been initialized. In your project’s post build configurations, chmod the files to be executable, so as to be able to use them. We have our own scripts that you can find in the overlay folder.

This project has been done by:

Mahmoud Gamal Saad

Merna Anis

Sara Abdallah Ahmed

three ITI students from embedded systems track intake 40.