

Debian 8 (or 9) install for MPD (Botic optional)

I advise to read this manual completely before starting any actions.
There are some configuration option which needs to be understand and maybe you need to make some notes for yourself which part is needed or not.

Index:

Introduction	: to Botic or not .. that's the question
Chapter 1	: preparing Debian setup on your Windows computer
Chapter 2	: installing Debian Linux
Chapter 3	: running Debian/Botic from the SD-card or from EMMC flash memory ?
Chapter 4	: updating Debian Linux and installing Music Player Daemon (MPD)
Chapter 5	: getting your music over LAN from a NAS or other computer
Chapter 6	: setting up MPD for USB DAC
Chapter 7	: installing Botic driver (needed for I2S output)
Chapter 8	: overview of Botic settings and wire connections
Chapter 9	: other useful commands

Introduction : to Botic or not .. that's the question

USB DAC:

As most of you know Linux + MPD is a fine platform for playing music to a DAC connected to the USB output of the BeagleBone Black (BBB).

I mention BeagleBone Black (with wired LAN) as this platform is probably the safest choice as there might be issues with other BeagleBone products due to other requirements in software which are not included so you need to fix it yourself.

If you are not a Linux expert I guess you will have a hard time fixing it.

So if you are using BBB → USB out → USB DAC **follow chapters 1 - 2 - 3 - 4 - 5 - 6.**

I2S DAC:

Since Miero developed a special driver enabling the possibility to get the I2S signal out directly from the header pins we get other options handling the audio signal.

This special driver is called "Botic" and is more complicated to handle compared to using USB out to a USB DAC.

Please realise that the standard Debian Linux does not provide I2S out.

So if you are using BBB → I2S out → Isolator*/reclocker*/DAC **follow chapters 1 - 2 - 3 - 4 - 5 - 7 - 8.**

* The isolator and/or reclocker are optional, see Twisted Pear Audio (and others) for options.

For more information on Botic see:

<http://bbb.ieero.com/>

<https://www.diyaudio.com/forums/twisted-pear/258254-support-botic-linux-driver.html>

<https://github.com/miero/linux-beagleboard-botic>

<http://www.twistedpearaudio.com>

When using I2S out you can use the USB port for something else like a thumb-drive with music or an external harddisk with even more music.

This external harddisk can contain all your music so you do not need to do the "music over LAN network" related steps from chapter 5.

See the article about "how to mount USB storage" on hifiduino how to mount your direct connected storage devices:

<https://hifiduino.wordpress.com/2014/03/19/beaglebone-black-accessing-usd-and-usb-storage/>

=====

Chapter 1 : preparing Debian setup on your Windows computer

Use your Windows PC for first steps:

Visit <https://beagleboard.org/latest-images> and select the required image file.

LXQT version is needed when using Graphical Desktop and IoT version is when using the BBB headless and log-in via SSH Terminal (PuTTY)

This manual is written for Debian 9.9 so there might be some other instructions when other versions are used.

IoT version is preferred.

Please note that the image file is compressed (like a ZIP file) and you need to unzip it first with a program like "7ZIP" (free download from internet)

The compressed size is approx. 800 Mb and uncompressed 3.5 Gb.

Use program "Win32Diskimager" (free download from internet) to select the 3.5 Gb image file and copy it to and micro-SD-card (select correct Drive-letter for card) and press the button "Write"

When Win32Diskimager is finished, close the program and remove SD-card.

=====

Chapter 2 : installing Debian Linux

Insert SD card in BeagleBone, hold "boot button" above the SD-card and power up.



At start all 4 LED's on BBB will light and then you can release the boot button.
The boot button forces the BBB to boot from SD-card even when functioning OS is on EMMC.



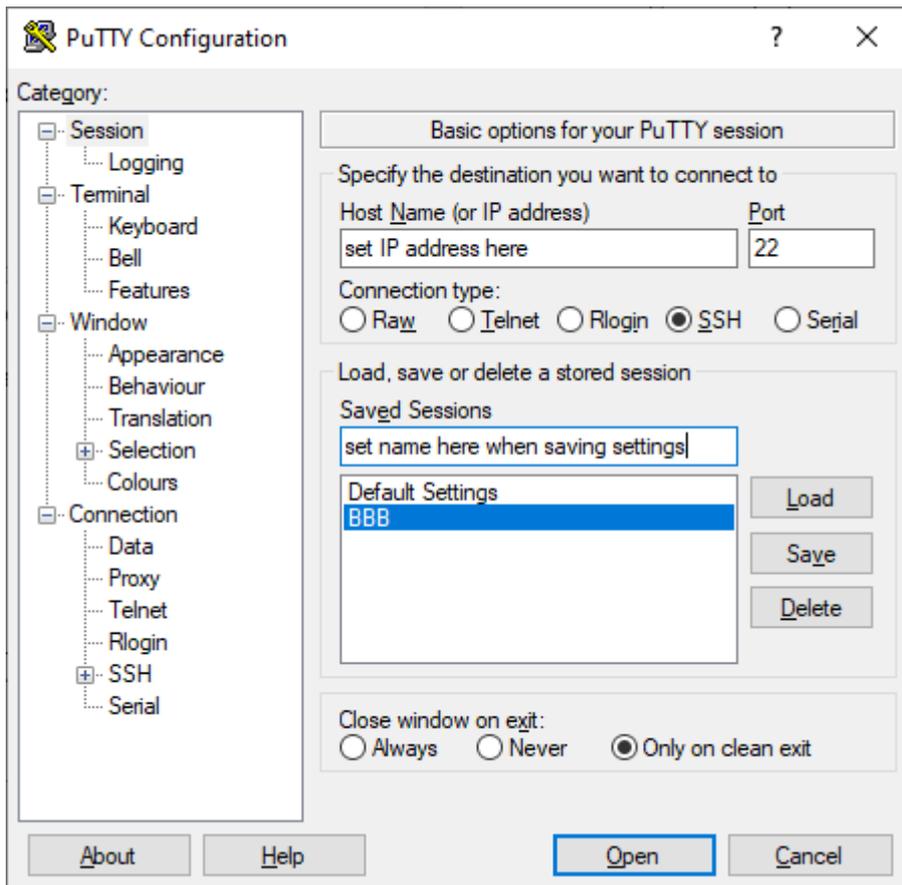
At first time booting from SD-card after 1-2 minutes the BBB should be running and connected with your LAN.

Find the IP address of the BBB:

Use an IP scanner like "Advanced IP Scanner" (free download from internet) and look for device "arm" or "Texas Instruments"

Connect to the BBB using "PuTTY" (free download from internet)
Set the IP address for BBB and keep port 22 and SSH as setting.

Smart would be to save the session so you can select it easier the next time.



Press "Open" and working with Linux starts now !!

PuTTY will open a terminal screen (similar like Microsoft DOS before Windows was standard)

All text in red is what you need to type yourself as command and press "Enter"

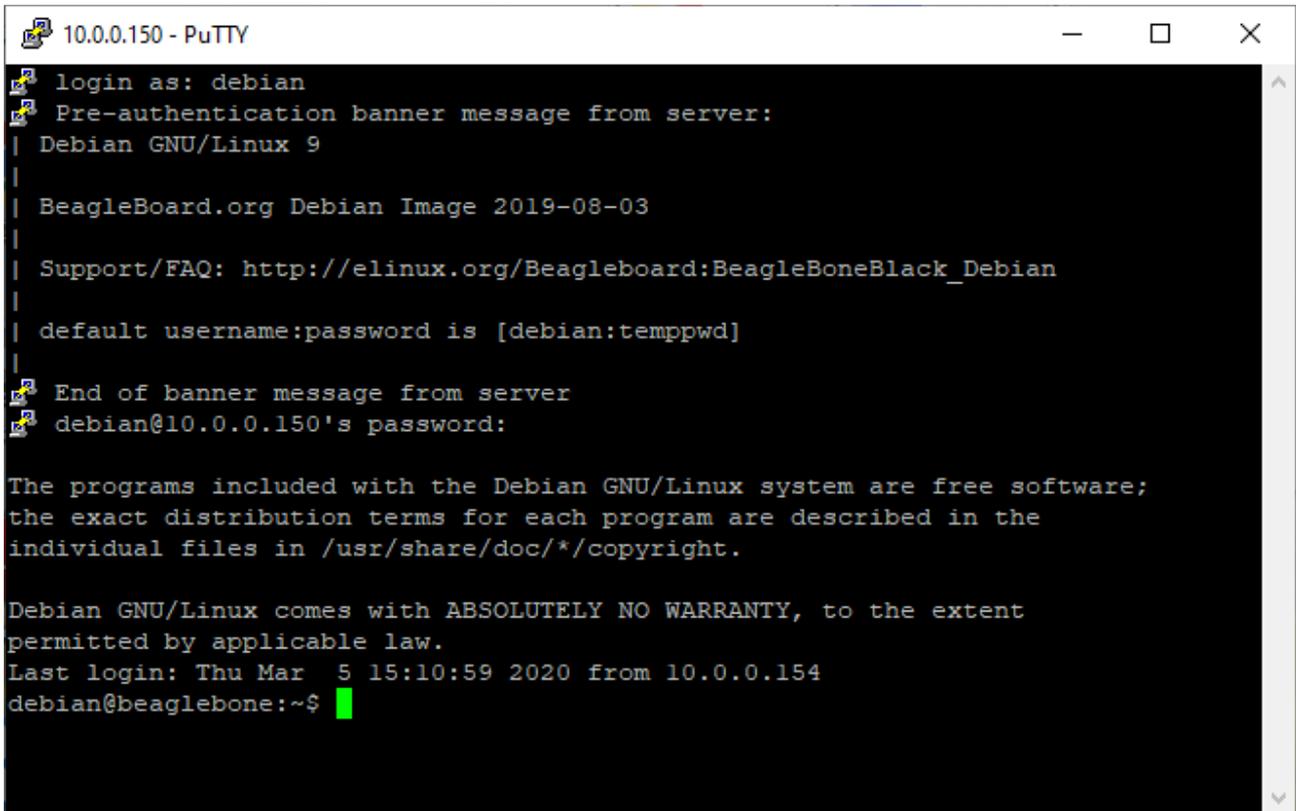
If you are not confident with re-typing long and strange commands copy / paste is done like this:

Copy a command from this document is standard Windows (select with mouse and press "Ctrl+C") but to paste in Linux you only need to right-click with your mouse on the green cursor to paste the text/command.

The other way around is selecting some text in Linux with your mouse is already enough to copy it and you can paste it in Windows as usual with "Ctrl+V" or right-click again in Linux if you want to copy text from one Linux file to another.

Login: **debian**
Password: **temppwd**

After successful login you are here:



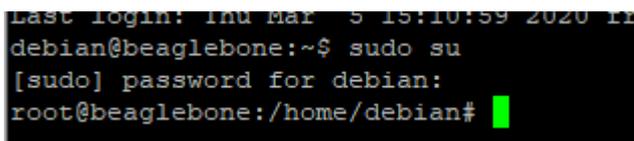
```
10.0.0.150 - PuTTY
login as: debian
Pre-authentication banner message from server:
| Debian GNU/Linux 9
|
| BeagleBoard.org Debian Image 2019-08-03
|
| Support/FAQ: http://elinux.org/Beagleboard:BeagleBoneBlack_Debian
|
| default username:password is [debian:temppwd]
|
End of banner message from server
debian@10.0.0.150's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Mar  5 15:10:59 2020 from 10.0.0.154
debian@beaglebone:~$ █
```

When connected you have limited access or rights.
To get administrator rights please type: **sudo su**

And confirm again when prompted with password **temppwd**



```
Last login: Thu Mar  5 15:10:59 2020 from 10.0.0.154
debian@beaglebone:~$ sudo su
[sudo] password for debian:
root@beaglebone:~/home/debian# █
```

You can see the change from **debian@beaglebone** to **root@beaglebone**.
Root is the standard administrator (super-user) account.

When you start a new PuTTY session you always need to activate root privileges again with **sudo su !!!**

=====

Chapter 3 : running Debian/Botic from the SD-card or from EMMc flash memory ?

Please realise that you are running Linux from the SD card now, which is relatively slow, and less expected lifetime of the SD-card compared of running of the onboard flash memory EMMc.

EMMc will give faster boot and file handling.

Please note that only BBB rev. C will have enough memory on EMMc (4 GB) and older BBB up to rev. B have only 2GB EMMc and there the 3.6GB image from the SD-card will not fit.

You better keep running from SD-card for now anyway until we completely finish the configuration.

When completed we can still copy the SD-card to EMMc (BBB rev. C !!) and keep the SD-card as backup (do not change the files and keep it safe) when you need to re-flash EMMc if you by accident removed some files or made other configuration mistakes in EMMc.

To copy the SD-card now to EMMC and continue configuration in EMMc can be done too but when you need to re-flash in the future you also need to repeat all following steps again ... your choice.

My preference is to create an SD-card with the final configuration and keep that one safe.

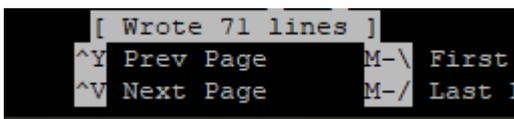
A benefit to run Linux from the SD-card is when you need to change hardware just put the card in the new BBB and it will work immediately like usual.

How to edit files in Linux:

nano = edit tool for any file

After editing a file **always** confirm with **Ctrl+O** (save) + "Enter" and **Ctrl-X** (close file)

You will see after "Ctrl+O" and "Enter" at the bottom confirmation that the file was saved by mentioning "Wrote xx lines" :



```
[ Wrote 71 lines ]
^Y Prev Page      M-\  First
^V Next Page      M-/  Last I
```

If it shows "access denied" you probably forgot to activate the administrator rights earlier by **sudo su**

To copy SD-card configuration to EMMc: → only for BBB rev. C with 4GB EMMc !!

type: **nano /boot/uEnv.txt**

remove "#" at the beginning of this line: (at the bottom of the file) to activate the EMMc flasher:

```
#cmdline=init=/opt/scripts/tools/eMMC/init-eMMC-flasher-v3.sh
```

reboot

hold "boot button" again at start-up until all 4 led's light up

It can take a minute but the led's will light in "Nightrider / Kitt" sequence

After some minutes when led's go out the copy from SD to EMMc is finished, please remove power, remove SD card and re-power again, the BBB should now boot from EMMc.

Login `debian` / `temppwd`
`sudo su (temppwd)`

If you made a mistake by trying to flash to a 2GB EMMc when EMMc is full all 4 led's will start blinking on/off simultaneously.

If you are not familiar with Linux it will be very hard to edit `/boot/uEnv.txt` now and put `"#"` again at the beginning of the EMMc flasher to de-activate this.

→ `#cmdline=init=/opt/scripts/tools/eMMC/init-eMMC-flasher-v3.sh`

I guess you better start all over again and forget the part about flashing to EMMc or buy a BeagleBone rev. C with 4GB EMMc.

=====

Chapter 4 : updating Debian Linux and installing Music Player Daemon (MPD)

`apt-get update`

This will update the library index of all the Linux software packages which is available on the internet

`apt-get upgrade`

This will upgrade all software to the newest version as the "apt-get update" above also provides version information so Linux knows what needs to be upgraded.

This process can take quite some time and it will ask you if you want to continue so press "y" when asked.

You can also confirm "Yes" automatically by typing `apt-get upgrade -y` so it will do the upgrade without questions. This also works when installing programs with command

`"apt-get install name-of-program -y"`

`apt-get autoremove` (for Debian 8)

`sudo apt autoremove` (for Debian 9)

Removes unnecessary packages which are not needed anymore after the upgrade.

Set correct time and timezone: `dpkg-reconfigure tzdata`

Select your region and then select nearby city which fits your time-zone.

Do you remember the old but very useful DOS program "Norton Commander" ??

Linux has the same called "Midnight Commander" or **mc** in Linux.

If you know how to use it then it is a very easy tool otherwise keep using the typed commands in red text.

How to use **mc** is found here: <https://midnight-commander.org/wiki/doc>

To install:

`apt-get install mc -y` → -y = "Yes" to continue installation

After installation type **mc** and press "Enter" to start the program:

```
root@beaglebone:/home/debian# mc
```

Instead of using "nano" you can now find the file in Midnight Commander and open (edit) it there by pressing "F4". Saving changes is done again by "Ctrl+O" and "Enter"
F3 is a safe way of viewing the file first without risk of changing anything.
F3 also closes the file again.

To install Music Player Daemon (MPD) and the client (MPC):

```
apt-get install mpd mpc -y
```

=====
Chapter 5 : getting your music over LAN from a NAS or other computer

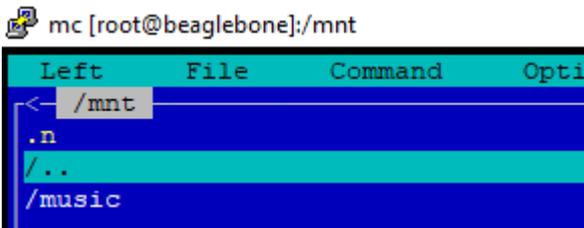
cifs-utils is needed for file access over the network when music files are on a NAS or external computer:

```
apt-get install cifs-utils -y
```

Create a Music directory as folder in /mnt:

```
mkdir /mnt/music
```

This is how it looks like in mc:



Linux needs external drives to be linked to their internal file system by the command "mount" for which the folder /mnt is mostly used.
Linux works with NFS to access external folders so you need to activate this on your NAS or external computer.

To link the external drive automatically during boot we need to set this in the file fstab:

```
nano /etc/fstab
```

```
add line: //IPAddressNAS/Folder /mnt/music cifs  
username=???,password=???,uid=mpd,ioccharset=utf8 0 0
```

(make sure this is in one line but my text editor breaks it in 2 lines when writing this instruction)

```

mc [root@beaglebone]:/etc
GNU nano 2.7.4
# /etc/fstab: static file system information.
#
/dev/mmcblkpl1 / ext4 noatime,errors=remount-ro 0 1
debugfs /sys/kernel/debug debugfs defaults 0 0

//10.0.0.20/Music /mnt/music cifs username=ubuntu,password=ubuntu,uid=mpd,ioccharset=utf8 0 0

```

The username and password is only necessary when you have set so on your shared drives or folders on your NAS to allow access.

To activate the "mounting" of the NAS to /mnt/music:

mount -a

Now you should see your music folders identical as on your NAS: (here in **mc**)

```

mc [root@beaglebone]:/mnt/music
Left  File  Command  Options
< /mnt/music
.n
/..
/ABBA
/ABC
/Adele
/Alanis Morissette
/Alexander O'Neal
/Alicia Keys

```

reboot

Check again folder /mnt/music/ if the Music files/folders are still present. I had some issues keeping "mount -a" stick after reboot so I always check.

Some NAS or OS on external file servers need special options for the mount in /ect/fstab ... here you are on your own but there is enough "fstab" mount information on the internet related to your NAS or file server.

=====

Chapter 6 : setting up MPD for USB DAC

(this is when using an USB DAC only ... for I2S output skip chapter 6 and go to chapter 7)

nano /etc/mpd.conf → change the following
music_directory change to: "/mnt/music"
port "6600" (remove # at start of line)
auto_update "yes" (remove # at start of line)

Audio output: (alsa)
name change to "Name_of_your_DAC"
device "hw:1,0" (make sure it is set as "hw:1,0" and remove # at start of line)
mixer_type "hardware" (remove # at start of line)

Device hw:1,0 is the DAC connected to the USB port of the BeagleBone.

"reboot" or **"sudo /etc/init.d/mpd restart"** to activate the new settings in mpd.conf

Find a suitable MPD app for your Android or IOS device (or Windows client if controlled via a PC) and set the required settings and you should be able to play your music.

Almost every App loads the album covers from the original music folders.
For this the http protocol is used so make sure http is also enabled on the shared music drive/folders.

=====

Chapter 7 : installing Botic driver (needed for I2S output)

To download Botic:

```
wget http://repo.ieero.com/botic/pool/main/l/linux-upstream/linux-image-4.8.13-botic7-rc3_1_armhf.deb
```

You can check here if there are any newer Botic versions to download:

<http://repo.ieero.com/botic/pool/main/l/linux-upstream/>

To unpack (install) Botic:

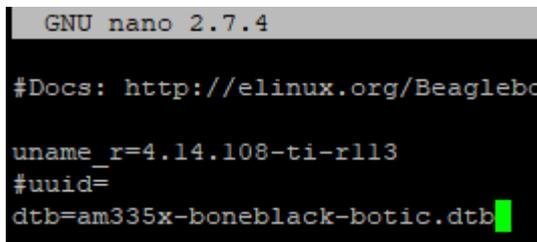
```
dpkg -i linux-image-4.8.13-botic7-rc3_1_armhf.deb
```

To activate Botic:

```
nano /boot/uEnv.txt
```

Remove "#" before dtb= and complete line like this: `dtb=am335x-boneblack-botic.dtb`

And add # at the start of this line: `"enable_uboot_cape_universal=1"` further down this file.



```
GNU nano 2.7.4
#Docs: http://elinux.org/Beaglebo
uname_r=4.14.108-ti-r113
#uuid=
dtb=am335x-boneblack-botic.dtb
```

```
nano /etc/mpd.conf    → change the following
music_directory      change to: "/mnt/music"
port                 "6600"          (remove # at start of line)
auto_update          "yes"           (remove # at start of line)
```

```
Audio output: (alsa)
name                 change to "Name_of_your_DAC"
device               "hw:0,0"    (make sure it is set as "hw:0,0" and remove # at start of line)
mixer_type           "hardware" (remove # at start of line)
```

Device hw:0,0 is for using I2S out from 40 pin header instead of sending audio over USB

reboot

To see the actual (default) Botic settings: `grep . /sys/module/snd_soc_botic/parameters/*`

```

root@beaglebone:/home/debian# grep . /sys/module/snd_soc_botic/parameters/*
/sys/module/snd_soc_botic/parameters/blr_ratio:64
/sys/module/snd_soc_botic/parameters/clk_44k1:22579200
/sys/module/snd_soc_botic/parameters/clk_48k:24576000
/sys/module/snd_soc_botic/parameters/dai_format:16385
/sys/module/snd_soc_botic/parameters/dsd_format_switch:1
/sys/module/snd_soc_botic/parameters/ext_masterclk:3
/sys/module/snd_soc_botic/parameters/pinconfig:default
/sys/module/snd_soc_botic/parameters/serconfig:MMMM
root@beaglebone:/home/debian# █

```

If a Botic parameter needs to be changed: **nano /boot/uEnv.txt**
Please look at <http://bbb.ieero.com/> for possible parameter settings, see also chapter 8.

Add the parameters at the end of this lline: `cmdline=coherent.....`

Example when using no external clocks: → `snd_soc_botic.ext_masterclk=0`

```

###0-Boot Overlays###
cmdline=coherent_pool=1M net.ifnames=0 rng_core.default_quality=100 quiet snd_soc_botic.ext_masterclk=0 █
#In the event of sdid real failures, uncomment this next line

```

Make sure you continue at the same line and add each additional parameter with a space between the parameters if you add more than one parameter.

Result (after **reboot**) when setting new value `snd_soc_botic.ext_masterclk=0` in `uEnv.txt`:

```

root@beaglebone:/home/debian# grep . /sys/module/snd_soc_botic/parameters/*
/sys/module/snd_soc_botic/parameters/blr_ratio:64
/sys/module/snd_soc_botic/parameters/clk_44k1:22579200
/sys/module/snd_soc_botic/parameters/clk_48k:24576000
/sys/module/snd_soc_botic/parameters/dai_format:16385
/sys/module/snd_soc_botic/parameters/dsd_format_switch:1
/sys/module/snd_soc_botic/parameters/ext_masterclk:0
/sys/module/snd_soc_botic/parameters/pinconfig:default
/sys/module/snd_soc_botic/parameters/serconfig:MMMM
root@beaglebone:/home/debian# █

```

Find a suitable MPD app for your Android or IOS device (or Windows client if controlled via a PC) and set the required settings and you should be able to play your music.

Almost every App loads the album covers from the original music folders.
For this the http protocol is used so make sure http is also enabled on the shared music drive/folders.

Now you are ready to play 😊

=====

Chapter 8 : overview of Botic settings and wire connections:

The Botic settings as described at the end of chapter 7 are related to these I/O pins.

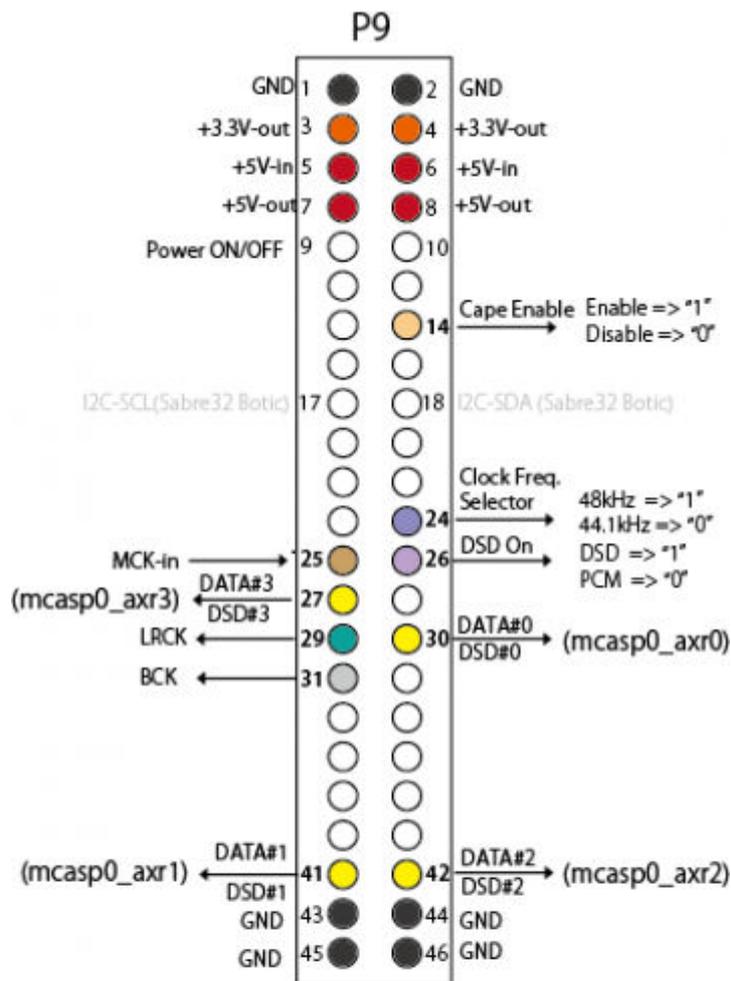
Possible Botic options:

Make sure you first check the current Botic settings with the command:

```
grep . /sys/module/snd_soc_botic/parameters/*
```

Only if these settings do not match with your connections or signals you need to enter the correct settings in /boot/uEnv.txt at cmdline=coherent..... as described in chapter 7.

Here you see the input and output pins on the BBB P9 header:



[Botic 7 kernel parameters]

```
"snd_soc_botic.serconfig=MMII"
```

```
"snd_soc_botic.dsd_format_switch=1"
```

```
"snd_soc_botic.ext_masterclk=3"
```

```
"snd_soc_davinci_mcasep.mute_pin=-1"
```

The green settings are the default settings by Botic:

snd_soc_botic.ext_masterclk=?

- 0 no external clocks, just onboard for 48k freq
- 1 external clock for 44k1 + onboard for 48k
- 2 external clock for 48k only
- 3 external clocks for 44k1 and 48k (default)
- +4 invert polarity of clock selection switch GPIO0_15
- 9 external clock for 44k1 only

snd_soc_botic.dsd_format_switch=?

- 0 DSD switch is disabled
- 1 DSD switch is enabled (default)
- +2 invert polarity of format selection switch GPIO0_14

snd_soc_botic.serconfig=????

- MMMM 8ch I2S output or 4ch DSD output (default)
- I--- 2ch I2S output on mcasep0_axr0 pin
- I-- 2ch I2S output on mcasep0_axr1 pin
- I- 2ch I2S output on mcasep0_axr2 pin
- I 2ch I2S output on mcasep0_axr3 pin
- DD-- 2ch DSD output on mcasep0_axr0 and mcasep0_axr1 pins
- MM-- 4ch I2S output or 2ch DSD output on mcasep0_axr0 and mcasep0_axr1 pins
- MM 4ch I2S output or 2ch DSD output on mcasep0_axr2 and mcasep0_axr3 pins
- S--- 2ch SPDIF output on mcasep0_axr0 pin
- S-- 2ch SPDIF output on mcasep0_axr1 pi

snd_soc_davinci_mcasep.mute_pin=?

(not set by the system, add this parameter when needed)

- 1 mute pin functionality is disabled
- or these additive numbers:
- + 1 signal mute on pin 0
 - + 2 signal mute on pin 1
 - + 4 signal mute on pin 2
 - + 8 signal mute on pin 3
 - + 16777216 invert the mute signal

snd_soc_botic.dai_format=?????

- +1 I2S mode
- +2 Right Justified mode (LSB)
- +3 Left Justified mode (MSB)
- +4 L data MSB after FRM LRC
- +5 L data MSB during FRM LRC
- +0 NB_NF: normal BCLK & LRCK
- +512 NB_IF: normal BCLK, inverted LRCK
- +768 IB_NF: inverted BCLK, normal LRCK
- +1024 IB_IF: inverted BCLK & LRCK
- +4096 CBM_CFM: DAC is BCLK & LRCK master
- +8192 CBS_CFM: DAC is BCLK slave, LRCK master
- +12288 CBM_CFS: DAC is BCLK master, LRCK slave
- +16384 CBS_CFS: DAC is BCLK & LRCK slave

Default value: I2S + NB_NF + CBS_CFS = 1 + 0 + 16384 = 16385

snd_soc_botic.clk_44k1=?????????

Default value: 22579200

snd_soc_botic.clk_48k=?????????

Default value: 24576000

snd_soc_botic.blr_ratio=??

For I2S mode default value is 64.

Other valid values are 32 and 48 (and maybe others).

=====

Chapter 9 : other useful commands

Power down BBB	: shutdown now
PCM output Botic	: cat /proc/asound/Botic/pcm0p/sub0/hw_params (only when music is playing)
MPD status	: systemctl status mpd
MPD restart	: sudo /etc/init.d/mpd restart
system resources	: top ("q" to exit)
Exit SSH console	: exit
check Debian version	: cat /etc/debian_version
check MPD version	: mpd -version

I have a personal issue with my Synology NAS (PC hardware with Xpenology) and can't get http support operational for my shared Music folder so I can not load cover art in my iOS app. But I can set /mnt/music for http and point the app to the BBB IP instead of NAS IP:

apt-get install webfs

nano /etc/webfsd.conf

Change the following:

```
# document root
web_root="/mnt/music"

# port to listen on
web_port="8080" (8080 is an example)
```

service webfs start start the webserver

General note:

As I am not a Linux specialist (merely put all fragments of information together from the internet) I hope this manual is useful for many novice Linux / Botic users.

Sometimes it can be helpful having an extra SD-card and installing an older Linux version.

Like mentioned I had some issues with "mount -a" not sticking after reboot what apparently was not the case with Debian 8 ... why ?? I have no clue but feel free to experiment if you run into issues.

Having 2 cards, one with Debian 8 and one with Debian 9 or one without and one with Botic makes you very flexible for testing until everything works perfect 😊