MERUS™ audio amp HAT ZW

User manual - KIT_40W_AMP_HAT_ZW

About this document

Scope and purpose

This document describes the MERUS™ audio amp HAT for Raspberry Pi Zero W equipped with the Infineon MA12070P proprietary multi-level class D amplifier. Detailed information is provided on the board's content, layout and use. A Raspberry Pi Zero W is mandatory for operation.

Intended audience

This document is intended for anyone working with the MERUS™ audio amp HAT, such as makers and audio engineers.

Attention: ***Read through this user manual before operating the board.***

Attention: ***Observe proper volume and limiter settings handling procedures. Failure to do so may result in damage to components on the board.***

Table of contents

About this document **1**
Table of contents **1**
1 MERUS™ audio amp HAT overview **3**
1.1 Key features 3
1.2 Application and block diagram 4
2 MERUS™ audio amp HAT description **5**
2.1 Features and specifications 5
2.2 Audio performance 5
2.3 Hardware description 5
2.3.1 Board description and pin functions 5
2.3.2 Schematics 7
2.3.3 Required input power 10
2.3.4 I²S digital audio interface configuration 10
2.3.5 GPIO controls 10
2.3.6 Short-circuit protection 10
3 Hardware set-up **11**
4 Software set-up **12**
4.1 Raspbian 12
4.1.1 Official Raspbian method (currently not supported) 12
4.1.2 Infineon MERUS™ Raspbian method (current preferred method for Raspbian) 14
4.1.3 MERUS™ custom kernel building (optional) 15
4.2 Volumio 15
4.2.1 Official Volumio method (currently not supported) 15
4.2.2 Infineon MERUS™ Volumio method (current preferred method for Volumio) 15

4.2.3 AirPlay (shairport-sync) 17
4.3 alsamixer and maximum volume/limiter settings 17
4.3.1 alsamixer controls 17
4.3.2 Volume and limiter settings 19
5 Revision history 21
References 21

1 MERUS™ audio amp HAT overview

The MERUS™ audio amp is a Hardware at the Top (HAT) audio power amplifier board that brings the Infineon proprietary multi-level technology to Raspberry Pi users and makers. It is intended for loudspeaker building and wireless music streaming with minimum size and consumption, state-of-the-art power efficiency and HD audio quality. The MERUS™ audio amp is equipped with the MA12070P class D multi-level amplifier, which can provide up to 40 W instantaneous peak power from the official Raspberry Pi supply, and up to 48 kHz/24 bit music playback. Furthermore, there is no need for extra power supplies, as the MERUS™ audio amp is powered directly from the same supply as the Raspberry Pi Zero W. It is compatible with the most popular audio streaming applications for Raspberry Pi, such as Volumio, Max2Play and JustBoom, and it can be also used in Raspbian as an AirPlay client or Bluetooth audio receiver.

1.1 Key features

The MERUS™ audio amp HAT for Raspberry Pi Zero W has the following features:

- Equipped with MERUS™ MA12070P proprietary multi-level amplifier
- Compatible with both Raspberry Pi Zero and Raspberry Pi Zero W
- Compatible with major streaming applications for Raspberry Pi (Volumio, JustBoom, Max2Play)
- Power input: 5 V/2.5 A (sourced from the same single supply as the Raspberry Pi)
- No need for external or extra power supplies
- Up to 40 W instantaneous peak output power with the Raspberry Pi official 5 V/2.5 A supply
- Up to 48 kHz of sample rate and 24 bits of music playback
- THD + N: 0.077 percent at 7 W/4 Ω/1 kHz
- Full hardware control, customization and error monitoring through Linux alsamixer

1.2 Application and block diagram

The MERUS™ audio amp HAT is meant to be used and controlled with a Raspberry Pi Zero or Raspberry Pi Zero W. The application diagram in Figure 1 shows the simplified connection scheme. The HAT has a boost converter built in, which supplies 20 V to the MA12070P amplifier. It runs in slave mode so all the control and communication is handled by the Raspberry Pi, which is implemented in the software drivers. As shown, most applications will require a host controller such as a phone or computer over a wireless network.

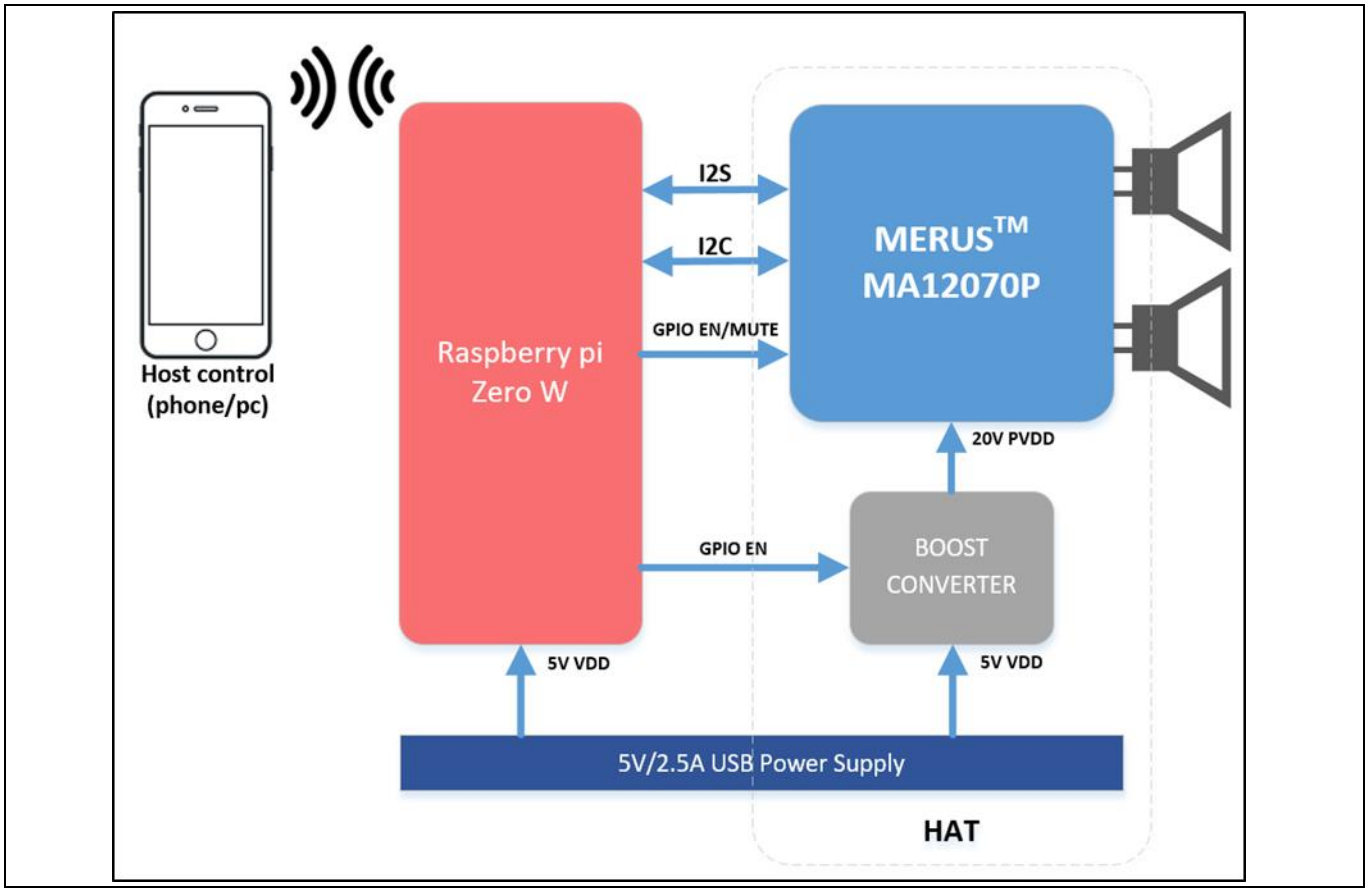


Figure 1 MERUS™ audio amp HAT application diagram

2 MERUS™ audio amp HAT description

2.1 Features and specifications

The MERUS™ audio amp board provides the following features and specifications:

- MERUS™ MA12070P multi-level class D amplifier
- Built-in boost converter from 5 V to 20 V to supply the MA12070P device
- Minimum power supply requirements:
  - micro USB power supply: 5 V to 2.5 A micro USB (no need for extra high voltage power supplies)
  - Battery/power bank: 5 V to 2.1 A 3000 mAh (may vary according to the protection mechanisms of the battery)
- I^2S digital audio input
- I^2C communication for full register map control
- Automatic enable control for booster and amplifier for secure boot-up and shut-down sequences
- Digital limiter for loudspeaker protection
- 2 x Bridge Tied Loaded (BTL) channel default configuration
- Optional 1 x Parallel Bridge Tied Loaded (PBTL) configuration for high-power true wireless stereo applications

2.2 Audio performance

- Output power p-channel (max. instantaneous peak – measured with music): 20 W, RL = 4 Ω
- Output power p-channel (max. average – measured with sine waves): 7 W – THD + N = 0.077 percent, RL = 4 Ω, f = 1 kHz
- THD + N: 0.014 percent – 1 kHz, P_{OUT} = 1 W, RL = 4 Ω
- Idle power consumption (HAT only): 390 mW
- Idle power consumption (HAT + Raspberry Pi): 970 mW
- Overall efficiency (1 W/ch at 4 Ω): 60 percent

2.3 Hardware description

2.3.1 Board description and pin functions

Figure 2 shows the board, pin distribution and functions of the MERUS™ audio amp HAT. It is shown that the design is centered on the MA12070P multi-level amplifier, which is supplied from both the TPS61178 built-in boost converter and the Raspberry Pi 5 V USB power supply. The MA12070P counts with a digital audio processor with master and independent volume controls per channel and with limiters implemented for device protection. For detailed information on MA12070P refer to the product datasheet.

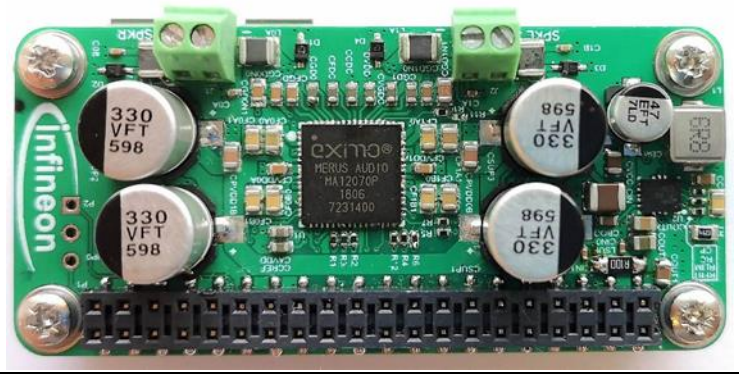


Figure 2 MERUS™ audio amp HAT board overview

Table 1 shows the pin function and description of the MERUS™ audio amp HAT.

Table 1 MERUS™ audio amp HAT pin description

Pin	Name	Type	Description
1	VDD	P	5 V power supply for internal digital circuitry and boost converter
2	VDD_3V3	P	3V3 power supply from Raspberry Pi (not used)
3	VDD	P	5 V power supply for internal digital circuitry and boost converter
4	SDA	I/O	I^2C bus serial data
5	GND	P	–
6	SCL	I/O	I^2C bus serial clock
7	/EN	I	MA12070P enable pin
8	NC	–	–
9	/MU	I	MA12070P mute pin
10	GND	P	–
11	PI_BCLK	I	MA12070P I^2S digital audio serial clock
12	NC	–	–
13	GND	P	–
14	NC	–	–
15	/ERROR	O	MA12070P error output pin (not used)
16	NC	–	–
17	/CLIP	O	MA12070P error output pin (not used)
18	NC	–	–
19	GND	P	–
20	NC	–	–
21	NC	–	–
22	NC	–	–
23	NC	–	–
24	NC	–	–
25	NC	–	–
26	GND	P	–
27	NC	–	–

28	NC	–	–
29	GND	P	–
30	NC	–	–
31	NC	–	–
32	NC	–	–
33	GND	P	–
34	NC	–	–
35	NC	–	–
36	PI_LRCK	I	MA12070P I^2S digital audio word clock
37	NC	–	–
38	NC	–	–
39	PI_DOUT	I	MA12070P I^2S digital audio serial clock
40	GND	P	–

2.3.2 Schematics

Figure 3 and Figure 4 on the following pages show the MERUS™ audio amp HAT schematics.

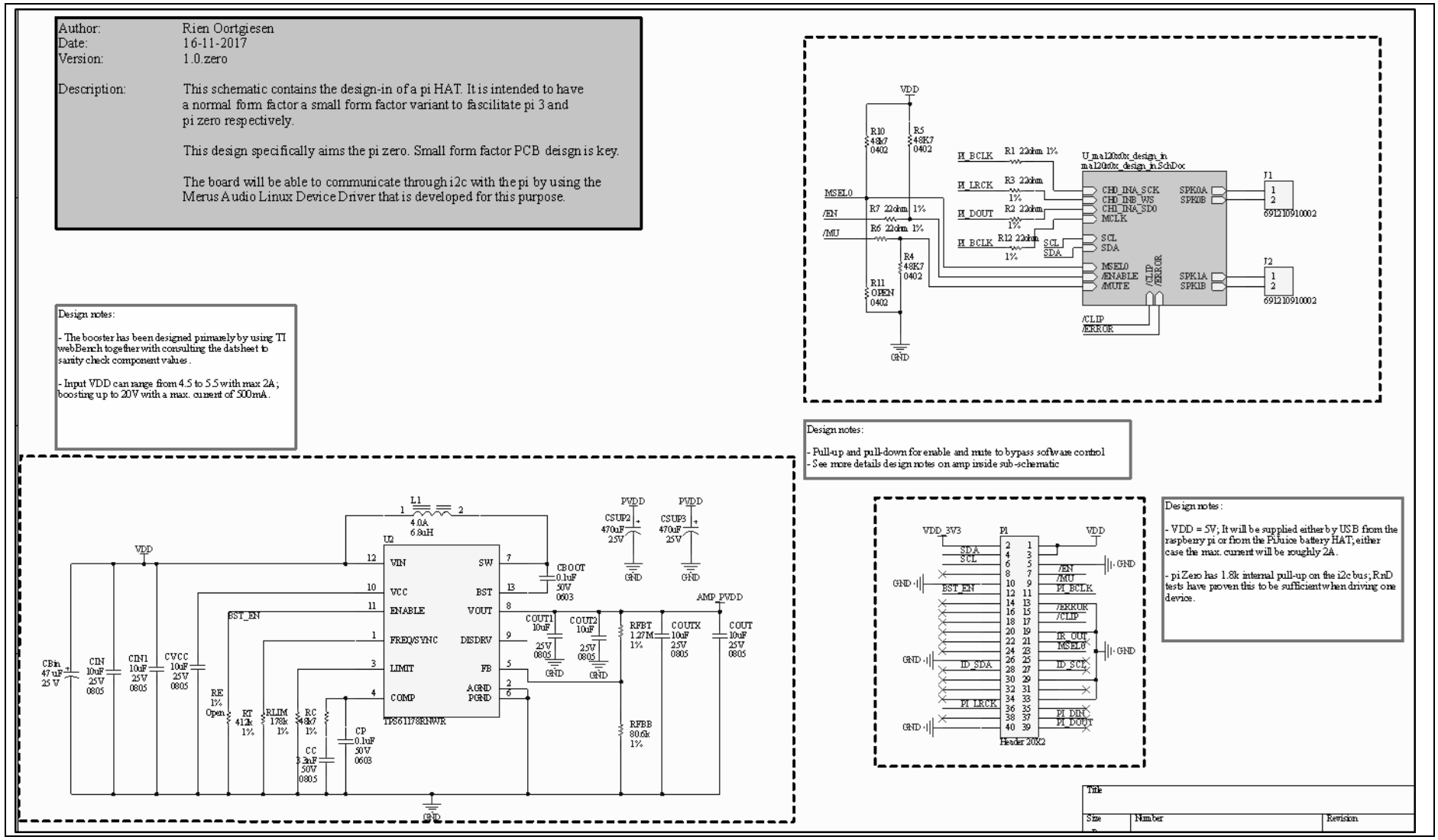


Figure 3 MERUS™ audio amp HAT main schematic

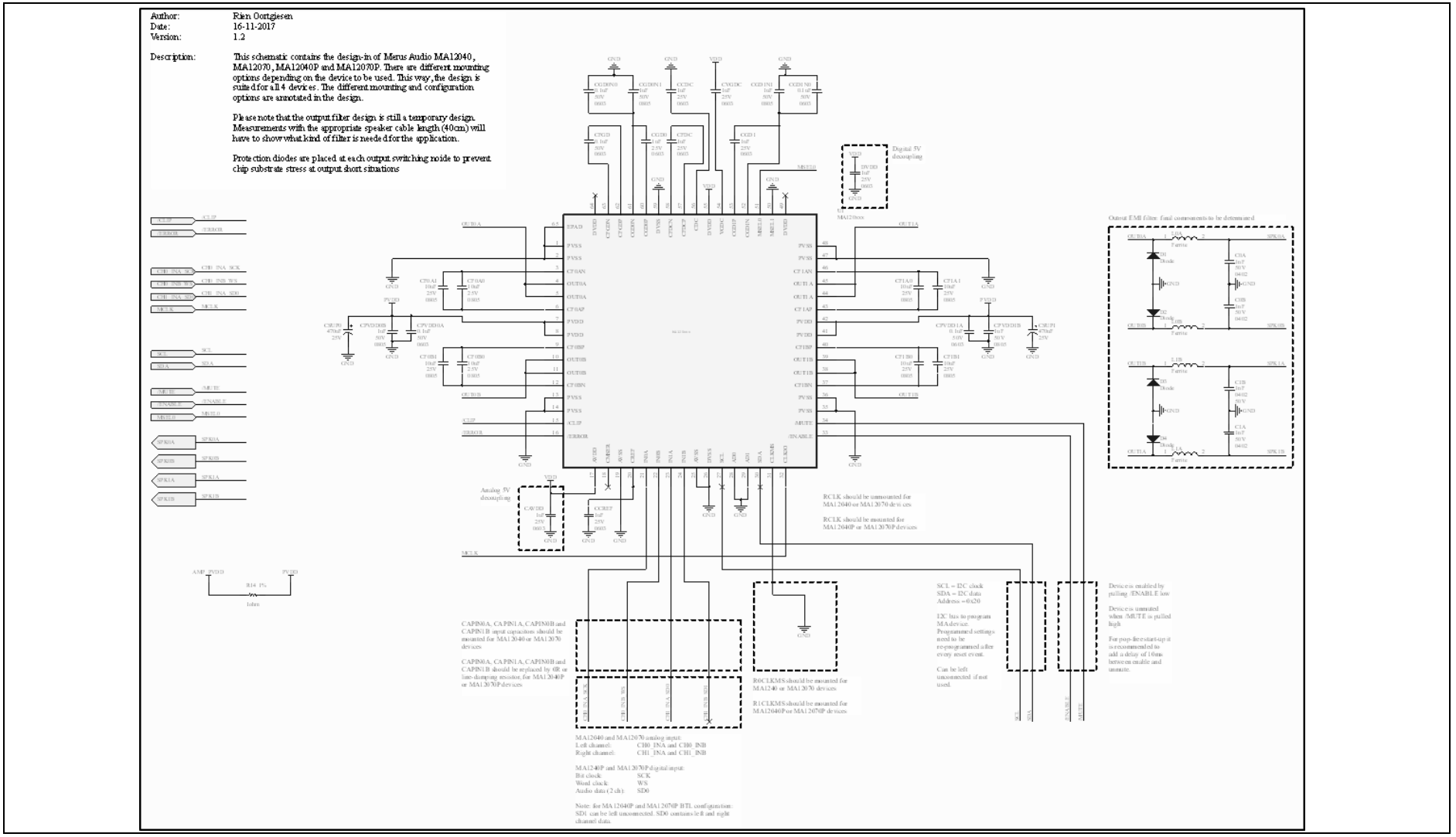


Figure 4 MERUS™ audio amp HAT main schematic

2.3.3 Required input power

The MERUS™ audio amp HAT requires a 5 V USB power supply with a minimum of 2.5 A DC current output. It is designed to be used with the official Raspberry Pi power adapter. MA12070P limiter and volume default settings have been determined for different nominal loudspeaker impedances while used with the Raspberry Pi power supply. Details of these settings are covered in section 4.3.

2.3.4 I²S digital audio interface configuration

The I²S interface is implemented as a slave three-wire communication in the MERUS™ audio amp board. As shown in the schematics, the serial clock line PI_BCLK is also wired to the MA12070P master clock. This method is used as the Raspberry Pi Zero doesn't count with a master clock output for this purpose. Supported sample rates range from 44.1 kHz to 48 kHz and the word length is fixed to 32 bits per channel. The major applications for wireless audio streaming with the Raspberry Pi are able to decode and/or convert several audio formats including MP3, AAC, WAV and FLAC to 32 bits, and therefore this should be set in the required host application, e.g. MPlayer running on Raspbian.

2.3.5 GPIO controls

The MERUS™ audio amp board makes use of some of the available GPIOs from the Raspberry Pi for safe boot-up and shut-down sequences. These are automatically handled by the software drivers. During boot-up, at first the boost converter and MA12070P are disabled. Next, the boost converter will be enabled until the 20 V bypass capacitors are fully loaded. Finally, the MA12070P will be enabled. With regard to the mute control pin, the amplifier will be always muted until PCM streaming is present. This will avoid pop-noises during non-playback time and also during boot-up and shut-down sequences.

2.3.6 Short-circuit protection

The MA12070P amplifier has built-in protection mechanisms such as Under-Voltage Lockout (UVLO), over-temperature warning/error, short-circuit/over-load protection, power stage pin-to-pin short-circuit, and DC protection. Additionally, the MERUS™ audio amp board counts with external protection diodes D1 to D4 at the output-in Figure 4 to avoid inrush currents under any circumstances.

3 Hardware set-up

The MERUS™ audio amp HAT is is designed to be mounted directly on top of a Raspberry Pi Zero, just like other HATs. A female header is used on the board for this purpose. In order to safely separate the HAT from the Raspberry Pi board, 4 mm 2.5 M hex spacers are recommended. Normally 2.5 M screws will be used to attach both boards from bottom (Raspberry Pi) to top (HAT). The length of the screws depends on the attachment below the Raspberry Pi. For this purpose, a minimum space of 4 mm between the Raspberry Pi and the surface below is suggested to provide good air flow without overheating of the Raspberry Pi Zero. Passive loudspeakers should be directly connected to the terminal blocks of each channel. With regard to cable length, the MERUS™ audio amp HAT supports up to 60 cm length where radiated emission tests are necessary. However, in most of its applications this condition won't be needed and cables up to 5 m can be used. For more information on this topic, please refer to 0. Figure 5 and Figure 6 show an assembly example with this hardware set-up.

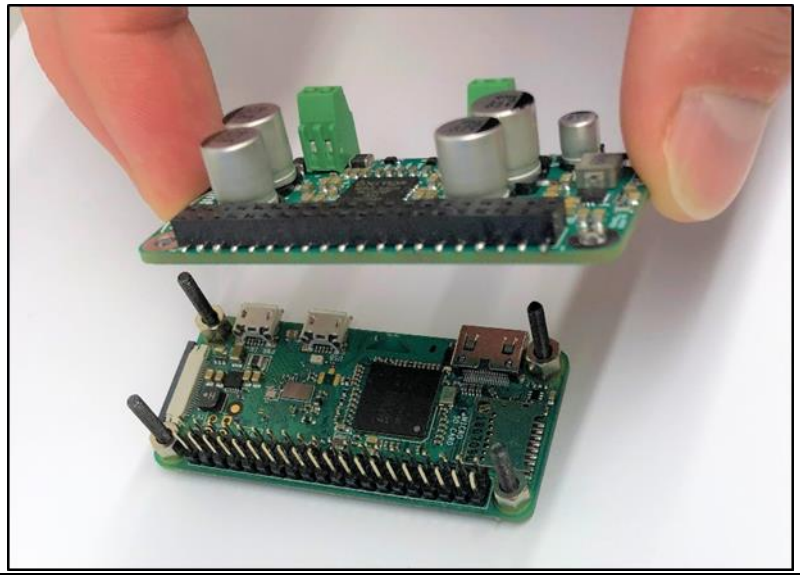


Figure 5 Assembly procedure of the Raspberry Pi + HAT using 2.5 M hex spacers between boards

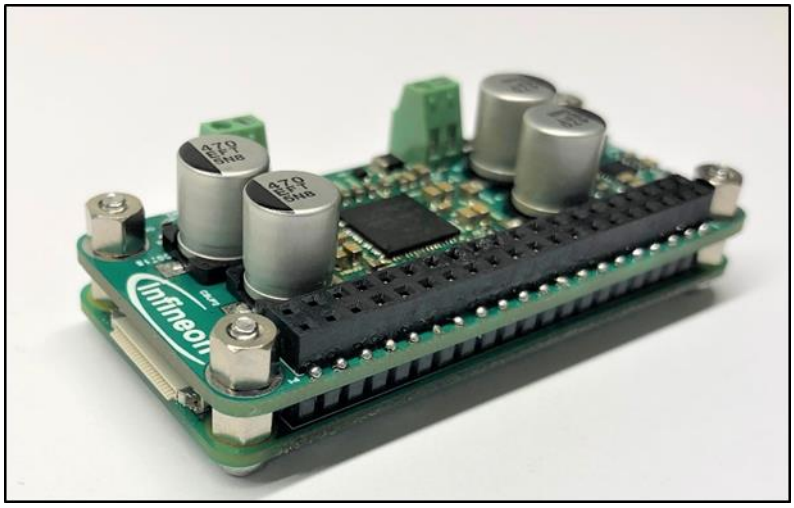


Figure 6 Completed assembly of the Raspberry Pi + HAT using 2.5 M hex spacers between boards

4 Software set-up

This section describes how to install and set up the MERUS™ audio amp drivers and necessary software packages with different Linux distributions and applications. Each distribution may have more than one possible method. First, basic installation processes are covered for the main distributions, and finally alsamixer controls and volume settings for different loudspeaker loads are covered. It is strongly recommended to read the last section before using the MERUS™ audio amp HAT.

4.1 Raspbian

4.1.1 Official Raspbian method (currently not supported)

The MERUS™ audio amp HAT software drivers will have support on the official Raspbian GitHub repository. To install the MERUS™ Raspberry Pi drivers the following procedure should be applied on a Windows, macOS or Linux computer:

1. Download the Raspbian image file from this link.
2. Download balenaEtcher (or any other SD card image burner) from this link, and open it.
3. Insert your SD card into the host machine and burn the Raspbian image file to the card.
4. Before unmounting the SD card:
    - Open the file called “config.txt” located at the card’s boot partition.
    - Replace the line

```
dtparam=audio=on
```

with

```
dtparam=audio=off
```

– this will turn off the built-in sound card of the Raspberry Pi.

- Next, add the following line below the previous modification in order to load the MERUS™ amp driver during boot-up, and save the file.

```
dtoverlay=merus-amp
```

- Configuring the wireless network and SSH:
    - Linux:

In order to stream audio via a wireless network it is necessary to configure the file called “wpa_supplicant.conf” located in the rootfs partition in the etc/wpa_supplicant directory. Open the file with a text editor and add the following content after the last line:

```
GROUP=netdev

network={
```

```
ssid="YOUR_NETWORK_NAME"
  psk="YOUR_PASSWORD"
}
```

- Windows:

In order to stream audio via a wireless network it is necessary to create a file called "wpa_supplicant.conf" located in the boot partition. You can do this by creating a file with a text editor with the following content (make sure the file doesn't have any other extensions than .conf):

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
network={
        ssid="YOUR_NETWORK_NAME"
        psk="YOUR_PASSWORD"
}
```

- To enable SSH access through your host computer create a file called "ssh" inside the boot partition. The file doesn't need any content and must not have any other extension, such as .txt.

5. Once this file has been saved, the SD card can be unmounted and then inserted into the Raspberry Pi Zero W.
6. Insert the SD card into the Raspberry Pi with the HAT attached, and turn it on.
7. Use Angry IP Scanner to find the IP address of the Raspberry Pi.
8. Next, in order to log in remotely to the Raspberry Pi from a Linux or macOS machine, type the following (usually the password is "raspberrypi" if you are logging in for the first time):

```
ssh pi@<ipadress>
```

9. If you are on a Windows machine use "pi" as the user and the Raspberry Pi IP address in the host address/name of any terminal window application such as Putty (usually the password is "raspberrypi" if you are logging in for the first time).
10. Finally, once you have accessed the Raspberry Pi through SSH, type:

```
aplay -l
```

to check that the driver's sound card and codec were loaded. If the process has been successful the terminal should show the following output prompt:

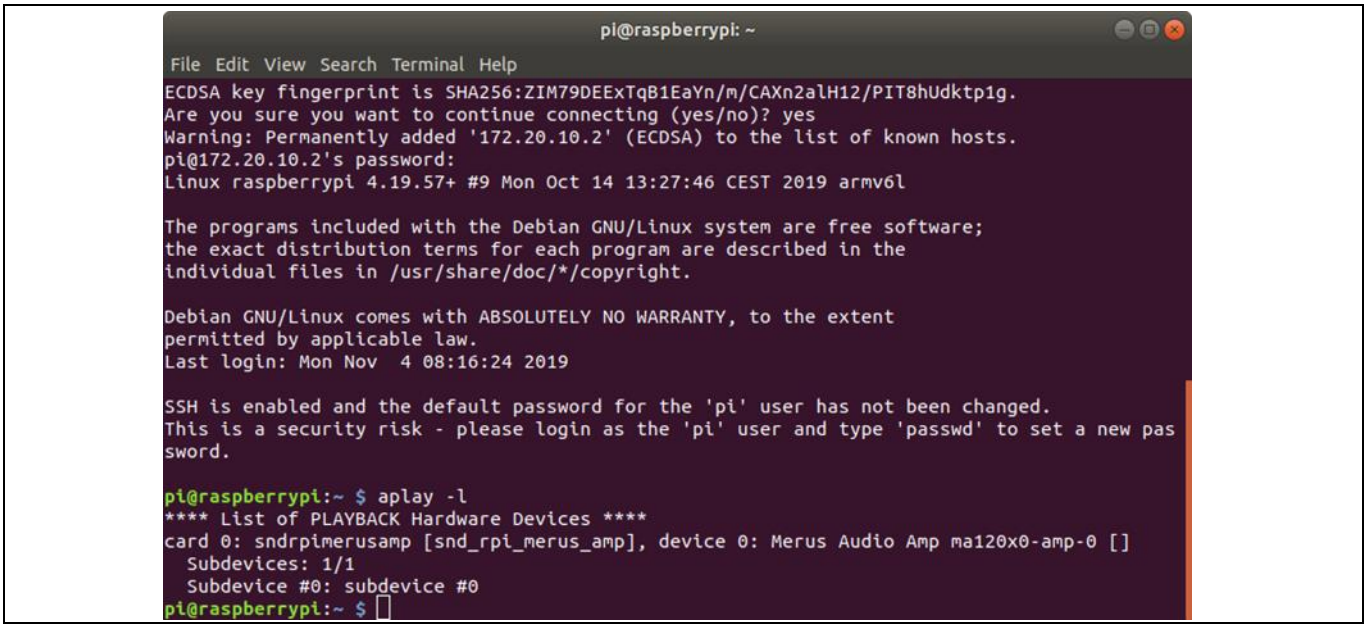


Figure 7 Sound card and codec verification with aplay -l command output

It shows that the default soundcard is “snd_rpi_merus_amp” with “ma120x0-amp-0” codec referring to the Infineon MA120x0P class D amplifier series.

12. Always make sure you have the latest update on your Raspberry Pi by typing the command line:

```
sudo apt update
```

and then:

```
sudo apt full-upgrade
```

When the process is finished, reboot the system.

4.1.2 Infineon MERUS™ Raspbian method (current preferred method for Raspbian)

A customized Raspbian image can be used to install Raspbian in a Raspberry Pi with the MERUS™ audio HAT drivers. This image has been built with kernel source version 4.19.57+. Note that if an upgrade or update is made the MERUS™ might not be compatible with the new version. To install the Raspbian distribution with the MERUS™ audio amp driver built in, the following procedure should be applied on a Windows, macOS or Linux computer:

1. Download the RaspbianMerusAudio.img image file from the Infineon MERUS™ audio amp HAT site.
2. Insert the SD card in the host machine.
3. Use any image burning application to burn the file to an SD card, such as balenaEtcher or Win32diskImager.
4. Follow steps 4 to 11 in section 4.1.1.

4.1.3 MERUS™ custom kernel building (optional)

A custom kernel building of the Raspbian kernel source is possible with the the MERUS™ audio amp HAT drivers. This method allows the user to make modifications not only to the drivers but also to the kernel source while maintaining compatibility with the last kernel version (currently 4.19.57+). This procedure is covered in [2].

4.2 Volumio

4.2.1 Official Volumio method (currently not supported)

The MERUS™ audio amp HAT software drivers will have support on the official Volumio distribution. To install the MERUS™ Raspberry Pi drivers for Volumio the following procedure should be applied on a Windows, macOS or Linux computer:

1. Download the Volumio image file from this link.
2. Download balenaEtcher (or any other SD card image burner) from this link and open it.
3. Insert your SD card in the host machine and burn the downloaded Volumio image file to the card.
4. After the burning process insert the SD card into the Raspberry Pi and turn it on.
5. To configure the wireless network settings of Volumio, connect your host machine or mobile phone to the “Volumio” network and then type http://volumio.local into your browser.
6. Go to the settings section and follow the instructions in the Volumio GUI to finish your wireless configuration. The next time the Raspberry Pi is booted it will connect to the configured network automatically.
7. Finally, go to the settings section and select “Playback options” to configure the MERUS™ audio amp HAT as the system’s sound card. Turn the I^2S option on and choose MERUS™ audio amp as the output device and DAC model.
8. Select “Shutdown” from the menu and then turn on the Raspberry Pi again so the system is rebooted.
9. Follows steps 10 to 14 in section 4.2.2.

4.2.2 Infineon MERUS™ Volumio method (current preferred method for Volumio)

A customized Volumio image can be used to install Volumio in a Raspberry Pi with the MERUS™ audio HAT drivers. This image has been built with Kernel source version 4.14.92+. To install the Volumio distribution with MERUS™ audio amp drivers built in, the following procedure should be applied on a Windows, macOS or Linux computer:

1. Download the Volumio image file from the Infineon MERUS™ audio amp HAT site.
2. Download balenaEtcher (or any other SD card image burner) from this link and open it.
3. Insert your SD card into the host machine and burn the downloaded Volumio image file to the card.
4. After the burning process insert the SD card into the Raspberry Pi and turn it on.
5. To configure the wireless network and audio settings of Volumio, connect your host machine or mobile phone to the “Volumio” network and then type http://volumio.local/ into your web browser to access the

configuration screen. (Depending on the type of machine you are working on, the Volumio configuration screen might appear automatically, without typing http://volumio.local/.)

6. Go through the initial configuration menu and follow the instructions in the Volumio GUI to finish your configuration. This menu will appear automatically the first time you boot the Raspberry Pi:
    - Select the language and press “Next”.
    - Select the name of your device (if you have only one device it is recommended to leave it as “Volumio”) and continue to the next setting.
    - Select the output device: **make sure the “I²S DAC” option is off** and select “snd_rpi_merus_amp” as the output device (usually this option will be already configured). Then, press “Next”.
    - Continue to the network settings and select the network you want to connect to the Raspberry Pi.
    - Optionally configure the music and finish the initial configuration.
7. Next, go to the settings menu (top-right corner) and select playback options. **Make sure the “I²S DAC” option is off** and select “snd_rpi_merus_amp” as the output device (already configured in step 6). In the volume options select “Software” as the mixer type to allow the protection mechanisms of the HAT to work properly. In the audio resampling section turn on the audio resampling option. Select “24” as target bit depth, and “Native” as the target sample rate. **Remember to press “Save” after changing each option.**
8. In order to access the Volumio GUI after the reboot, type http://volumio.local/ or your Raspberry Pi IP address on your host machine’s web browser. You can obtain the IP address with Angry IP Scanner.
9. Select “Shutdown” from the menu and then turn on the Raspberry Pi again so the system is rebooted.

In order to be able to set the MERUS™ amp controls in alsamixer (explained in section 4.3) it is necessary to enable SSH in Volumio. Follow the next steps in order to do this:

10. Turn on the Raspberry Pi (pre-configuration should have been applied according to the steps above).
11. Type “<your Raspberry Pi IP address>/DEV” or “volumio.local/DEV” on your host machine’s web browser and press “ENABLE” under the SSH section. There is no output warning or message after enabling the SSH option, so once the button has been pressed do not expect any messages. For details visit https://volumio.github.io/docs/User_Manual/SSH.html.
12. Next, in order to log in remotely to the Raspberry Pi from a Linux or macOS machine, type the following (including your IP address) in the terminal window:

```
ssh volumio@<ipadress>
```

13. If you are on a Windows machine use “volumio” as the user and the Raspberry Pi IP address in the host address/name of any terminal window application such as Putty. The password is “volumio”.
14. Finally, once you have accessed the Raspberry Pi through SSH, type:

```
aplay -lL
```

to check the driver's sound card and codec were loaded. If the process was successful the terminal should show the same output prompt as in Figure 4.

It shows that the default sound card is "snd_rpi_merus_amp" with "ma120x0-amp-0" codec referring to the Infineon MA120x0P class D amplifier series.

4.2.3 AirPlay (shairport-sync)

Wireless audio streaming in Raspbian is also possible with shairport-sync, which brings AirPlay support to the Raspberry Pi. In order to install shairport-sync type the following command in the Raspberry Pi terminal window:

```
sudo apt-get install shairport-sync
```

Finally, open any AirPlay-compatible streaming application and find your Raspberry Pi host name as an AirPlay device.

4.3 alsamixer and maximum volume/limiter settings

4.3.1 alsamixer controls

The MERUS™ audio amp sound card driver comes with its own set of controls implemented in the alsamixer, which provide direct access to the MA12070P multi-level amplifier registers and digital audio processor. It is strongly recommended to read section 4.3.2 in order to not exceed the maximum power ratings.

By typing the following command (for any Linux distribution distribution) in the Raspberry Pi terminal:

```
alsamixer
```

the following mixer will be shown:

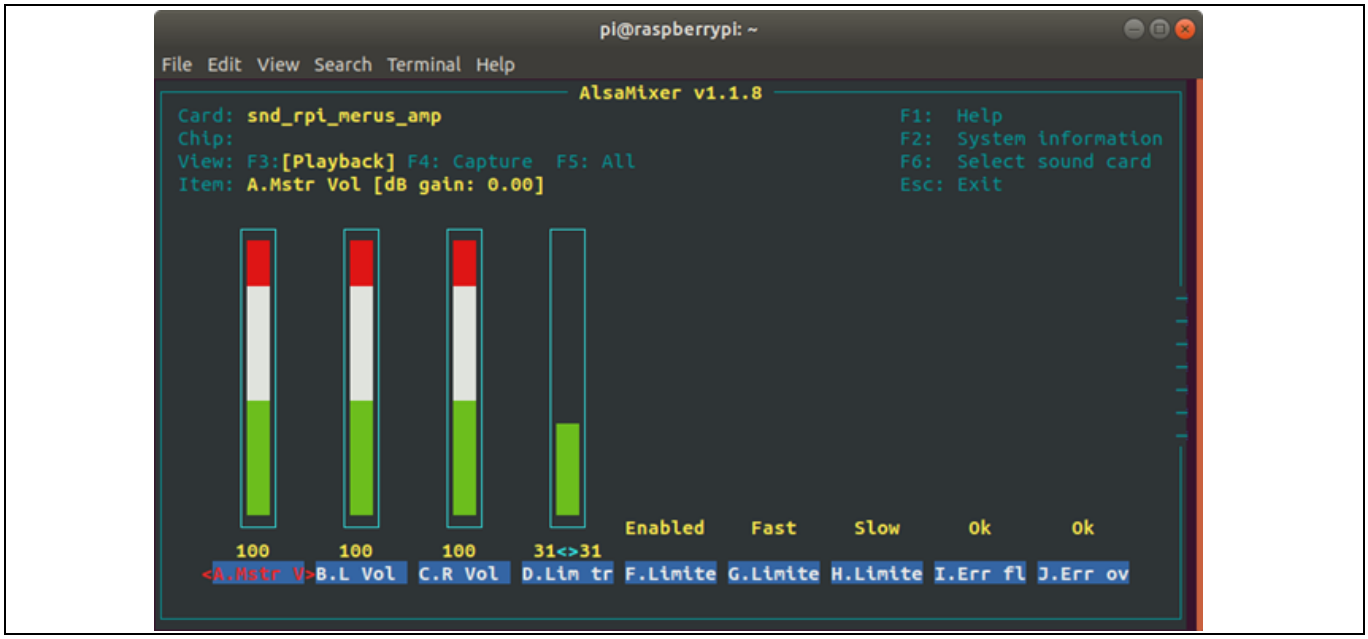


Figure 8 MERUS™ audio amp sound card driver mixer (first page)

By scrolling to the right, the last controls will be shown as the following picture:

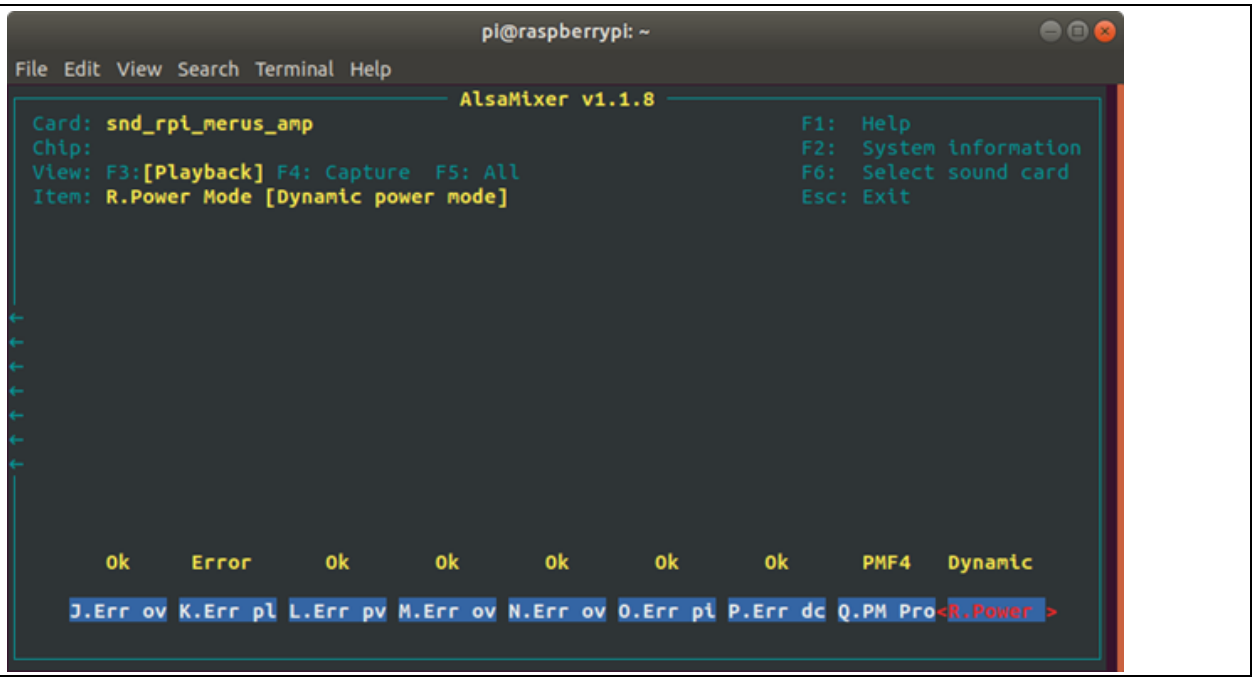


Figure 9 MERUS™ audio amp sound card driver mixer (second page)

It should be noted that all the mixer controls are ordered from left to right and from “A” to “R”. It is recommended to use the item description located at the top-left corner to have a more detailed description of each control and its current status. Increasing master volume and limiter thresholds controls from their default values should be done carefully. Exceeding the recommended settings may damage the devices permanently. For detailed information about volume and limiter settings please refer to section 4.3.2.

Table 2 lists the available alsamixer controls of the MA12070P in the MERUS™ audio HAT. The register and access type is also specified.

Table 2 alsamixer controls and their registers

Item/Control	alsamixer name	MA12070P register	Access type
A	Mstr Volume	0x40	Read/Write
B	L Vol	0x42	Read/Write
C	R Vol	0x43	Read/Write
D	Lim thresh Volume	0x47 and 0x48	Read/Write
E	N/A	N/A	N/A
F	Limiter Enable	0x36	Read/Write
G	Limiter Attck	0x35	Read/Write
H	Limiter Rls	0x35	Read/Write
I	Err flycap	0x6D	Read only
J	Err overcurr	0x6D	Read only
K	Err pllerr	0x6D	Read only
L	Err pvddunder	0x6D	Read only
M	Err overtempw	0x6D	Read only

N	Err overtempe	0x6D	Read only
O	Err pinlowimp	0x6D	Read only
P	Err dcprot	0x6D	Read only
Q	PM Prof	0x1D	Read/Write
R	Power Mode	0x00	Read/Write

Controls I to P show the error status of the MA12070P. If there are no errors an “OK” string should appear in each control. It should be noted that “Err pllerr” will show an error when the Raspberry Pi is booted. After any audio signal has been played the control should show no errors.

4.3.2 Volume and limiter settings

In order to not reach maximum output current limits of the booster and its inductor it is necessary to set the limiter values properly. In relation to this point, a protection mechanism has been implemented in the HAT, which will mute and disable the MA12070P amplifier when the maximum output current is reached. In order to set the correct limiter settings for a specific loudspeaker impedance, the following procedure should be applied:

1. Enter the alsamixer in Raspbian or Volumio.
2. Turn the limiter control “Lim thresh Volume” down to -40 dB.
3. Set the volume control “Mstr Volume” to 0 dB.
4. Turn up the volume of your audio source (phone, computer, etc.) to maximum.
5. Start playing music.
6. Start carefully increasing the limiter control in 1 dB steps while looking at the current limiter level. Here there are two possible results:
    a. When the maximum output current has been reached, the device will mute and it will be disabled.
    b. The Raspberry Pi has been rebooted automatically because of the voltage drop of the power supply being used.
7. Take note of the last limiter threshold value that was reached.
8. Reboot the Raspberry Pi and set the limiter value to 1 or 2 dB less than the last value reached before the reboot.
9. If no limiting at all is desired on the audio signal at full scale, set the volume to the same level as the limiter value found in step 8. However, this will reduce the loudness of the system. If maximum loudness is desired, maintain the volume at 0 dB.
10. If it is necessary or if the device is muted and disabled again, reboot the Raspberry Pi one more time and reduce the limiter threshold by 1 dB more.

Table 3 shows suggested limiter threshold values according to the nominal speaker impedance and its approximate DC resistance value using the official Raspberry Pi power supply. It is worth mentioning that these values may vary, but they are a good starting point to know where the limiter threshold should be set.

Table 3 Suggested limiter threshold values vs. speaker impedance

Loudspeaker impedance	≈ DC resistance	Max. lim. threshold allowed
2 Ω	2 Ω	-16 dB
4 Ω	3.5 Ω	-13 dB
4 Ω	4 Ω	-11 dB
6 Ω	6 Ω	-9 dB
8 Ω	8 Ω	-7 dB

Optionally, if the audio source can't reach full level for any reason, the volume can be increased accordingly. However, in order to keep the booster output current below acceptable limits, the threshold should not be modified.

5 Revision history

Document version	Date of release	Description of changes
1.0	20-08-2019	First release

References

[1] EMC output filter recommendations for MA120XX(P) – https://www.infineon.com/cms/en/product/power/class-d-audio-amplifier-solutions/integrated-class-d-audio-amplifier-ics/ma12070p/

[2] Drivers and kernel building with MERUS™ audio amp HAT Z

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Edition 2019-11-26

Published by

Infineon Technologies AG

81726 Munich, Germany

© 2019 Infineon Technologies AG. All Rights Reserved.

Do you have a question about this document?

Email: erratum@infineon.com

Document reference
UM_1911_PL88_1912_152704

IMPORTANT NOTICE

The information contained in this application note is given as a hint for the implementation of the product only and shall in no event be regarded as a description or warranty of a certain functionality, condition or quality of the product. Before implementation of the product, the recipient of this application note must verify any function and other technical information given herein in the real application. Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind (including without limitation warranties of non-infringement of intellectual property rights of any third party) with respect to any and all information given in this application note.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (**www.infineon.com**).

WARNINGS

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.