Set up a Lean Linux Audio DSP UAC2 Rpi3A+ Gadget

Version:
23/01/19 - All this is work under progress.
Some welcomed feedback issued from a first version of this Howto published on diyaudio.com has been included into this version - thank you to all having commented the first version. This version also has some corrected typos, has been upgraded by some more options and completed by more comments.

Topic:
It will take approx. 30 minutes to set up a basic UAC2 gadget following this HowTo. Optionally, some additional system configuration tweaks may be applied to make the whole system a bit leaner, which will take another hour or so. All the described steps and options of this HowTo have been real-world tested and will result into a well-behaved and functional UAC2 gadget. While playing audio and ssh into the gadget ...
$ systemctl | grep running | wc -l
... shows a total of 13 scopes/services/sockets.

System Actuality:
Raspberry Pi OS imager 1.7.3 / Bullseye V.5.15 (or optionally Kernel V.6.1. - see below in section A).

Ingredients:
Rpi3A+, Raspberry Pi OS Lite (64Bit), CamillaDSP.

Content part I (basic system):
A. Update the basic OS
B. Setup the UAC2 mode
C. CamillaDSP

Content part II (optional goodies):
D. Lean Ux - an attempt to get the system a wee bit leaner, lighter, faster
E. Further options and tweaks
F. System checks

Warning:
This is a kind of shortlist-barebone-HowTo. Some suggestions may be redundant, or not be really elegant, maybe stupid, and some better solutions may exist. Some suggestions will even seem problematic in terms of the integrity of the linux system, especially when applying the modifications of part D of the HowTo. Instead, the specific functionality as an UAC2 gadget is not affected.

Outcome:
Testing this UAC2 setup as a gadget along with a Linux host, everything seems flawless. Instead, using the UAC2 gadget along with a W10 host is flawed by occasional crackles and even brief (up ro some seconds), events of pausing of the audio stream.

About the formating of this HowTo:
```
# Code is formatted in this typeset
# Code bits formatted [LikeThis] are [UserDefined] or [SystemSpecific]
```

Disclaimer:
Take utmost care. Electricity might be lethal, as music at > 200dB SPL. And life as such is life threatening. Proceed at your own risk ...

A: The basic Raspberry OS - updates and options

apt
```
# apt update
# apt full-upgrade
# reboot now
```

rpi-update
Optional, in search for a more recent kernel.
Pros: The g_audio module version might get updated as well
Cons: You might jeopardize further updates/upgrades.
```
# rpi-update next
# reboot now
```

Comment:
Apt will also be used later on in this HowTo. Therefore, run rpi-update as one of the very last steps, e.g. after having performed all other apt-related modifications. Otherwise, apt then may struggle to perform correctly.

B. Setup the UAC2 mode

/boot/config.txt - append to this file
```
# vi /boot/config.txt
dtoverlay=dwc2,dr_mode=peripheral
```

/etc/modprobe.d/g_audio.conf - create this new file
```
# vi /etc/modprobe.d/g_audio.conf
options g_audio c_chmask=3 p_chmask=0 c_srate=44100 c_ssize=2
iProduct=»[AnyLabelYouWant]»
```

comment:
`x_chmask` are bitmasks for the channels, therefore ...
   0->0, 1=ChA, 2=ChB, 3=ChA+ChB
   `c_chmask=3` means capture 2 channels / stereo (channel 1 and 2)
   `p_chmask=0` means playback 0 channels, which removes the gadget from the players list
`x_ssize` is the sample size in bytes, therefore …
   1->8Bit, 2->16Bit, 3->24Bit, 4->32Bit, therefore choose
   `c_ssize=2` in case of S16_LE, or `c_ssize=4` for S32_LE

/etc/modules-load.d/g_audio.conf - create this new file
```
# vi /etc/modules-load.d/g_audio.conf
g_audio
```

/etc/modules-load.d/dwc2.conf - create this new file
```
# vi /etc/modules-load.d/dwc2.conf
dwc2
```

And then ...
```
# reboot now
```

Check the result by ...
```
$ cat /proc/asound/cards
$ grep '' /sys/module/g_audio/parameters/*
```

Eventually also check for updated options of the current g_audio module by ...
```
$ modinfo g_audio
```

Reference:
`https://www.kernel.org/doc/html/v[6.1]/usb/gadget-testing.html`
v[6.1] refers to the actual linux kernel version running on the gadget

C: CamillaDSP forever ...

C1: Install CamillaDSP (CDSP) on your system

apt
install bsdtar
```
# apt install libarchive-tools
```

wget
Download the CDSP tar
```
$ wget https://github.com/Henquist/camilladsp/releases/download/v[1.0.3]/
camilladsp-linux-aarch64.tar.gz
```
v[1.0.3] refers to the CDSP version to download

bsdtar
Extract and install CDSP
```
$ bsdtar -xf camilladsp-linux-aarch64.tar.gz
# mv camilladsp /usr/local/bin
```

C2: Configure CDSP to capture from the gadget USB_in

/home/[YourDirectory]/[CDSP_Directory]/[CDSP_ConfigFile.yml] - create this new file
Make the values of the CDSP configuration file match with the g_audio configuration

```
$ vi /home/[YourDirectory]/[CDSP_Directory]/[CDSP_ConfigFile.yml]
…
devices
  capture_samplerate: [==c_srate]
  …
  capture:
    channels: [has to match c_chmask]
    device: hw:CARD=[input device name from $arecord -l]
    format: [has to match c_ssize]
    type: Alsa
  playback:
    …
…
filters
  …
pipeline
  …
```

For e.g. c_srate=44100, c_chmask=3, c_ssize=2, this might transform into ...
```
…
devices
  capture_samplerate: 44100
  …
  capture:
    channels: 2
    device: hw:CARD=UAC2Gadget,DEV=0
    format: S16_LE
    type: Alsa
  playback:
    …
…
```

```
filters
  …
pipeline
  …
```

C.3. System bootup autostart a CamillaDSP starter script

/usr/lib/systemd/system/[AnyName].service - create this new file

```
# vi /usr/lib/systemd/system/[AnyName].service
[Unit]
Description=CamillaDSP Boot Starter Script
After=multi-user.target

[Service]
ExecStart=/home/[YourDirectory]/[YourStartScript]
Restart=always
RestartSec=1
User=root
Group=root
CPUSchedulingPolicy=fifo
CPUSchedulingPriority=10

[Install]
WantedBy=multi-user.target
```

and then …

```
# systemctl enable [AnyName].service
```

Comment:
Enable this service as one of the very last steps, e.g. after having performed all other modifications. Furthermore, the system and manual startups of CDSP should have proven to run flawlessly before.

D. Lean Ux - an attempt to get the system a wee bit leaner, lighter, faster

D.1. Configuration files mods - Linux general options

/etc/dhcpcd.conf - append to this file
make dhcpcd startup some seconds faster

```
# vi /etc/dhcpcd.conf
noarp
```

/etc/fstab - append to this file
do not write to the sd card, but into a temp file system residing in ram instead

```
# vi /etc/fstab
tmpfs  /tmp       tmpfs  noatime,nodev,nosuid,size=25M 0  0
tmpfs  /var/lock  tmpfs  noatime,nodev,nosuid,size=5M  0  0
tmpfs  /var/log   tmpfs  noatime,nodev,nosuid,size=5M  0  0
tmpfs  /var/run   tmpfs  noatime,nodev,nosuid,size=5M  0  0
tmpfs  /var/tmp   tmpfs  noatime,nodev,nosuid,size=5M  0  0
```

Comment:
In the same logic, do not set up a swap partition. Also, swapping to a file will be disabled later on.

/etc/systemd/journald.conf - edit this file
minimize journaling

```
# vi /etc/systemd/journald.conf
Storage=none
RuntimeMaxUse=0
```

```
SystemMaxUse=0
Audit=no
```

D.2. Configuration files mods - System specific options

/boot/config.txt - edit this file
For a hopefully lean basic RaspberryPi configuration, eg. to disable Rpi-onboard audio devices

```
# vi /boot/config.txt
arm_64bit=1              (most important on a 64-bit system)
audio_pwm_mode=1
camera_auto_detect=0
disable_audio_dither=1
disable_overscan=1
disable_poe_fan=1
disable_splash=1
display_auto_detect=0
dtoverlay=pi3-disable-bt
# dtoverlay=vc4-kms-v3d  (comment this one out)
dtparam=i2c_arm=off
dtparam=i2s=off
dtparam=spi=off
dtparam=audio=off
enable_uart=0
framebuffer_width=320
framebuffer_height=240
gpu_mem=16
hdmi_blanking=1
hdmi_drive=1
hdmi_force_mode=1
hdmi_group=1
hdmi_mode=1
max_framebuffers=2 (comment this one out)
```

/etc/modprobe.d/blacklist.conf - create this new file
avoid loading some modules which are not needed besides the audio UAC2 (and ssh) functionality

```
# vi /etc/modprobe.d/blacklist

### KERNEL MODULES OPTIONALLY NOT TO INSTALL ###

### tuttifrutti
install backlight /bin/false
install bluetooth /bin/false
install bcm2835_mmal_vchiq /bin/false
install btbcm /bin/false
install btqca /bin/false
install btsdio /bin/false
install btrl /bin/false
install drm /bin/false
install drm_panel_orientation /bin/false
install fuse /bin/false
install garp /bin/false
install ip_tables /bin/false
install ipv6 /bin/false
install llc /bin/false
install stp /bin/false
install uio /bin/false
install uio_pdrv_genirq /bin/false
install vc_sm_cma /bin/false
install vc4 /bin/false
install x_tables /bin/false

### sound
```

```
install snd_soc_hdmi_codec /bin/false

### video
install bcm2835_codec /bin/false
install bcm2835_isp /bin/false
install bcm2835_v4l2 /bin/false
install mc /bin/false
install videobuf2_bcm2835_v4l2 /bin/false
install videobuf2_common /bin/false
install videobuf2_dma_contig /bin/false
install videobuf2_memops /bin/false
install videobuf2_vmalloc /bin/false
install videobuf2_v4l2 /bin/false
install videodev /bin/false
install v4l2_mem2mem /bin/false

### !!!  MODULES TO KEEP !!! ###

### aes
# aes_arm64
# aes_generic
# libaes

### af_alg - algif_* - *hash*
# af_alg
# algif_aead
# algif_hash
# algif_skcipher
# gf128mul
# ghash_generic

### brcm* - cfg80211 - rfkill
# brcmfmac
# brcmutil
# cfg80211
# rfkill

### audio and gadget mode - sound - i2c - snd_*
# g_audio
# i2c_bcm2835
# libcomposite
# regmap_i2c
# snd
# snd_bcm2835
# snd_compress
# snd_pcm
# snd_pcm_dmaengine
# snd_soc_bcm2835_i2s
# snd_soc_core
# snd_soc_rpi_wm8804_soundcard
# snd_soc_wm8804
# snd_soc_wm8804_i2c
# snd_timer
# u_audio
# usb_f_uac2

### divers
# cmac
# md4
# md5
```

D.3. Single commands mods - Linux general options

apt

keep as few programs installed as possible - will make faster updates/upgrades

```
# apt purge [xyz]
```

xyz such as ...
avahi-daemon
bash-completion
bind9-host
bluez
ca-certificates
eject
file
gcc
gettext-base
gnupg ?
man-db
manpages
ncurses-term ? --> raspi-gpio
tasksel ?
usbutils
vim-common
zip

reinstall some leaner packages

```
# apt install [xyz]
```

vim-tiny

and finally ...

```
# apt clean
# apt autoremove --purge
```

systemctl

Have as few services as possible active in the system

`# systemctl mask [xyz]` (=byebye forever ...)

xyz such as …
apt-daily.service
apt-daily.timer
apt-daily-upgrade.timer
avahi-daemon.service
console-setup.service
cron.service
dphys-swapfile.service
getty@tty1.service
keyboard-setup.service
man-db.timer
ModemManager
rng-tools-debian.service
rsyslog.service
systemd-journal-flush.service
systemd-journald.service
systemd-journald.socket
systemd-journald-audit.socket
systemd-journald-dev-log.socket
triggerhappy.service
triggerhappy.socket

Check the result by ...

```
# reboot now
$ systemctl | grep running
$ systemctl | grep active
```

Comment:
Some of these services cannot be incactivated by `# systemctl disable [xyz.service]`
Therefore use of the irreversible `# systemctl mask [xyz]` instead
Some services can also be dynamically stopped on demand from within a script

```
# systemctl stop [xyz.service]
```

D.4. Multi-step mods

dphys-swapfile
Remove swapping

```
# dphys-swapfile swapoff
# dphys-swapfile uninstall
# apt purge dphys-swapfile
```

Wifi – use iwd instead of (outdated) wpa-supplicant

```
# apt install iwd
# systemctl enable iwd.service
# systemctl start iwd.service
# iwctl
  - device list
  - station [wlan0] scan
  - statopm [wlan0] get-networks
# vi /var/lib/iwd/[YourNetworkSSID].psk
   [Security]
   Passphrase=[YourPassphrase]
# systemctl mask wpa_supplicant
# systemctl stop wpa_supplicant
# apt purge wpasupplicant
# apt clean
# apt autoremove --purge
# reboot now
# rm -fr /etc/wpa_supplicant
```

Comment:
As long as wpasupplicant is running, iwd will not set up it's `/var/lib/iwd/network.psk` file
Therefore, first prepare iwd to log into a network, then get rid of wpasupplicant, then reboot.

Disable the journaling on the ext4 file system
First mount the sd-cart onto a file system of any linux host
And then ...

```
# tune2fs -O ^has_journal /dev/mmcblk0p2
```

E. Further options and tweaks

E.1. Run audio programs (such as CDSP) on specific and optionally isolated CPUs

The following example isolates CPU 2 and 3 to independently run all threads of specified audio programs. Instead, e.g. CPU 0 will handle all IRQs

[YourStartScript]
Assign a specific program to a singe CPU or to a set of CPU's

```
…
# taskset -c 2-3 camilladsp …
…
```

You may then also optionally want these specific CPU's isolated from other processes.

/boot/cmdline.txt - edit this file
CPU isolation - Paste the following content into the command line:

```
# vi /boot/cmdline.txt
… irqaffinity=0 nohz=on isolcpus=2-3 nohz_full=2-3 rcu_nocbs=2-3 ...

# reboot now
```

Check the result by ...

```
# cat /sys/devices/system/cpu/isolated
$ ps H -q $(pidof -s camilladsp) -o 'pid,tid,cls,rtprio,comm,pcpu,psr'
```

Pro's, con's and compromises:
CPU isolation might not be beneficial. The regular system scheduler is very effective in distributing all load between all available CPU ressources. Some programs such as CDSP or MPD now run theirs process as several threads, all or them being by standard more or less evenly distributed. In practice and as in one of the tested setups, CDSP runs as 3 threads and starts another 3 child threads. Squeezing all these 6 threads onto a single CPU might indeed not be an optimal solution. As a compromise and as described in this example, two CPU's (2-3) might be isolated and assigned to all of the threads of CDSP and it's child threads. This approach still isolates the audio processing while letting the scheduler balance the load between CPU2 and CPU3.

E.2. Throttle the CPU to a constant frequency

apt

```
# apt install cpufrequtils
```

[YourStartScript]

```
…
# cpufreq-set --freq nnn
# cpufreq-set --governor userspace --max nnn --min nnn
…
```

Check the result by …

```
# cat /sys/devices/system/cpu/cpufreq/policy0/cpuinfo_cur_freq
# cat /sys/devices/system/cpu/cpufreq/policy0/scaling_governor
```

Comment:
Avoiding switching (by avoiding the [ondemand] governor) may reduce the probability of xruns. This is because switching may come along with some ms of latency.

F. System checks united

F.1. The UAC2 gadget

```
$ cat /proc/asound/cards
$ grep '' /sys/module/g_audio/parameters/*
$ modinfo g_audio
```

F.2. Services

```
$ systemctl | grep running
$ systemctl | grep active
```

F.3. WiFi

```
$ ip route
$ systemctl | grep iwd
```

F.4. CamillaDSP

```
$ ps H -q $(pidof -s camilladsp) -o 'pid,tid,cls,rtprio,comm,pcpu,psr'# cat
```

F.5. CPU

```
# cat /sys/devices/system/cpu/cpufreq/policy0/cpuinfo_cur_freq
# cat /sys/devices/system/cpu/cpufreq/policy0/scaling_governor
# cat /sys/devices/system/cpu/isolated
```

```
...
```

F.2314298 The little red LED there

```
# isitlit
```