

TUTORIAL: CONFIGURAÇÃO DE DISPOSITIVOS DE ÁUDIO NO RASPBERRY PI – PARTE 1

Bárbara Circe¹, William D’Andrea Fonseca¹, Leonardo Jacomussi¹, Paulo H. Mareze¹

¹ Engenharia Acústica, Universidade Federal de Santa Maria,
Av. Roraima nº 1000, Cidade Universitária, Santa Maria (RS), Brasil, 97105-900
{barbara.circe, will.fonseca, leonardo.jacomussi, paulo.mareze}@eac.ufsm.br

Resumo

O Raspberry Pi é um pequeno e poderoso microcomputador (*single board computer*, SBC). Ele pode ser utilizado tanto de maneira didática nas faculdades de engenharia quanto na automação de tarefas cotidianas. Suas principais vantagens são o custo-benefício, a facilidade de uso e sua integração com diversos dispositivos e sistemas, seja via USB (com o uso de interfaces de áudio), pelo uso de HATs (*Hardware Attached on Top*) ou por GPIO (*General-purpose Input/Output*). Seu sistema operacional (SO) é baseado em Linux, sendo de uso livre. Assim, munido com *software* desenvolvidos em Python, o Raspberry Pi se torna uma poderosa ferramenta em processamento de sinais áudio (com significativo custo reduzido). Em forma de tutorial, este trabalho busca ajudar a comunidade científica na aquisição de sinais. Nesta primeira parte, elucida-se de forma acessível e precisa o passo a passo para identificar características do Raspberry Pi, bem como um conjunto de comandos que servem como ferramentas para investigar e configurar dispositivos de áudio. As instruções neste documento exploram abordagens simples que vão da configuração por meio de GUIs (*Graphical User Interface*) como o Gnome, chegando às etapas de configuração de *hardware* de áudio utilizando-se da arquitetura ALSA (*Advanced Linux Sound Architecture*) de maneira mais aprofundada.

Palavras-chave: Raspberry Pi, dispositivos de áudio, ALSA, tutorial, processamento de sinais.

Abstract

Raspberry Pi is a small and powerful Single Board Computer (SBC). It can be used as much for its didactic capabilities in engineering courses as for its capacity to automate everyday tasks. Its main advantages are the cost-benefit, its ease of use, and its ability to integrate with various devices and systems. Such integration may be via USB (using audio interfaces), HATs (Hardware Attached on Top) or GPIO (General-purpose Input/Output). Its operating system (OS) is Linux based and as such, free to use. Thus, equipped with software developed in Python, the Raspberry Pi becomes a powerful audio signal processing tool at a significantly reduced cost. A tutorial, this work seeks to assist the scientific community with signal acquisition. In this first part, an accessible and precise step-by-step guide to identifying and applying features of the Raspberry Pi is clarified. As well, a set of commands that serve as tools to investigate and configure audio devices are cataloged herein. The instructions within this paper explore simple approaches that go from configuring through GUIs (Graphical User Interface) such as Gnome, to more advanced stages of configuring audio hardware, using the more in-depth ALSA (Advanced Linux Sound Architecture).

Keywords: Raspberry Pi, audio interfaces, ALSA, tutorial, signal processing.

PACS no. 43.38.Md, 43.10.Pr, 43.38.Md, 43.58.Ta, 43.60.-c, 43.60.Qv.

1 Introdução

O Raspberry Pi (Rpi) é um pequeno e poderoso microcomputador *integrado em uma única placa (single board computer, SBC)*. Ele pode ser utilizado tanto de maneira didática nas faculdades de engenharia quanto na automação de tarefas cotidianas, por exemplo. Suas principais vantagens são o custo-benefício, a facilidade de uso e sua integração com diversos dispositivos e sistemas, seja via USB (com o uso de interfaces de áudio), pelo uso de HATs (*Hardware Attached on Top*) ou por GPIO (*General-purpose Input/Output*). Seu sistema operacional (SO) é baseado em Linux, sendo de uso livre. Assim, munido com *software* desenvolvidos em Python, o Raspberry Pi se torna uma poderosa ferramenta em processamento de sinais de áudio.

A placa possui uma infinidade de aplicações, podendo conectar-se a diversos dispositivos capazes de realizar aquisição e/ou a reprodução de sinais de áudio. Seu *framework* de áudio permite uma variedade de configurações, possibilitando o controle de entradas e saídas de diversos dispositivos e aplicações até mesmo de forma simultânea. A primeira parte deste tutorial demonstra como obter informações sobre o modelo do Raspberry Pi, apresentando em seguida o passo a passo para a configuração da comunicação entre *hardware* e a arquitetura de áudio ALSA (*Advanced Linux Sound Architecture*), possibilitando que aplicativos realizem tarefas envolvendo a captura e/ou reprodução de áudio. A segunda parte do tutorial está publicada¹ também nos *proceedings* deste evento (Acústica 2020) [1].

2 Fundamentos

Esta seção aborda aspectos básicos que são fundamentais para a compreensão das aplicações com Rpi.


2.1 O Raspberry Pi

Concebido em 2012, no Reino Unido pela *Fundação Raspberry Pi* [2], o Rpi é um popular microcomputador conhecido pelo seu baixo custo e por sua diversidade de aplicações, especialmente em tarefas que envolvem algum nível de automação [3–6]. Há atualmente pelo menos 9 modelos oficiais disponíveis no mercado², que vão do Raspberry Pi Zero ao Raspberry Pi 4 B. A Figura 1 exibe diferentes vistas do Raspberry PI 3 B+ (à esquerda), o modelo utilizado na investigação e produção deste tutorial, bem como do Raspberry PI 4 B (à direita), o atual mais moderno. Seu sistema operacional oficial é o **Raspberry Pi OS**, anteriormente chamado de Raspbian por ser baseado em Debian Linux.

(a) Raspberry Pi 3 B+ [7].

(b) Raspberry Pi 4 B [8].

Figura 1 – Raspberry Pi: modelos mais recentes.

¹Segunda parte do tutorial .

²Veja o site <https://www.raspberrypi.org/products/>.

A diversidade de entradas e saídas como HDMI, USB e o acoplamento de placas de expansão (HATs) possibilitam a conexão de diversos tipos de dispositivos capazes de captar e/ou reproduzir áudio. Logo, torna-se evidente a necessidade de uma configuração adequada para controlar entradas e saídas desses dispositivos, assim como ferramentas que possibilitam a comunicação entre *hardware* e aplicativo.

2.2 Tipos de dispositivo e aplicações

Há diferentes formas de conectar dispositivos que utilizam áudio no Raspberry Pi. Pode-se citar como exemplo (1) conectar um *headphone* na saída *on-board* através do conector de áudio da placa; (2) utilizar GPIO, seja por cabos ou utilização de placas de expansão; (3) acoplar dispositivos via USB, como interfaces ou placas de áudio mais complexas; além de (4) utilizar o áudio de uma tela através da porta HDMI ou ainda (5) conectar-se a um dispositivo reproduzidor de áudio via *bluetooth*. Somado à variedade de conexões, a *forma de utilização* dos dispositivos de áudio também pode ter outras dependências. Pode-se desejar utilizar uma placa de áudio que obedeça aos comandos do Python [9], por exemplo. Logo, alguns pacotes para gerenciamento dessa relação fazem-se necessários, como os módulos `sounddevice` e `PyAudio` utilizados em conjunto com o `PortAudio`, por exemplo. Resumidamente, a combinação do tipo de dispositivo de áudio e aplicação desejada exige muitas vezes configurações específicas.

Durante a escolha da melhor maneira de gerenciar dispositivos, encontramos certas limitações em diferentes métodos. Como um breve exemplo podemos citar: (1) o Rpi originalmente possui apenas saídas, isso pode ocasionar problemas em aplicações com tempo real e/ou projetos que necessitem de latências muito baixas (além disso, o áudio *on-board* possui apenas 11 bits [10]); (2) HATs são uma ótima opção porém é comum encontrar a um baixo custo aqueles que apenas gerenciam a entrada ou a saída ao invés de gerenciar ambos (a referência [11] contém uma lista de *hardware* compatíveis com o Rpi para diversas aplicações); e (3) algumas interfaces de áudio não são aceitas pelos aplicativos, porém se a ALSA reconhecer mesmo sem reproduzir ou gravar, é um bom sinal: o problema pode ser a falta de algum servidor (`PulseAudio` ou `JACK`) ou de uma biblioteca como o `PortAudio` que consiga lidar com a complexidade da interface e suas aplicações.

As referências [10] e [12] trazem informações valiosas e que podem poupar bastante tempo, sendo respectivamente um repositório de *software* de áudio para Raspberry (além de detalhes sobre sua utilização) e um conjunto de dispositivos USB que verificadamente funcionam com o microcomputador.

2.3 A arquitetura de áudio ALSA

A arquitetura ALSA (*Advanced Linux Sound Architecture*) é “um conjunto de *drivers* de *hardware*, bibliotecas e utilitários que fornecem funcionalidades de áudio e MIDI para o sistema operacional Linux” [13]. Esse subsistema (ALSA) estrutura sobretudo a relação entre o sistema operacional (Linux) e o *hardware*, como mostrado na Figura 2. Isto é, quando um aplicativo necessita enviar ou receber um sinal de áudio, ele necessita de um *conjunto intermediário de instruções* (chamado apenas de *framework* neste artigo), que faz o papel de conexão *software-hardware*. Logo, o uso da ALSA é compulsório.

A utilização apenas da ALSA, sem nenhum outro intermediário, é comumente chamada de *modo nativo*. Em analogia com o Windows³ a ALSA seria como os *drivers*. Entretanto, diferentemente do Windows, há muitos programas que precisam de bibliotecas adicionais de comunicação, isto é, uma camada acima da ALSA. Essa camada é chamada de *servidores de áudio* (ou *sound servers*), que no caso do Linux são o `PulseAudio` [14] e o `JACK` [15], veja a Figura 2 (a).

³No Windows, de forma geral, temos dois tipos *drivers* que conectam os aplicativos aos *hardware* de áudio. O tipo geralmente usado em aplicações profissionais é o ASIO (Audio Stream Input/Output), desenvolvido pela Steinberg, ele busca contornar o sistema operacional (OS) o máximo possível para que se obtenha valores baixos de latência. O outro tipo de *driver* é chamado de WASAPI (Windows Audio Session API), utilizado de forma geral no sistema (em aplicações mais simples) e/ou com *hardware* que não comportem os *drivers* ASIO nativamente (nesse caso existe um *workaround* chamado ASIO4All). Esse tipo de *hardware* provavelmente é compatível com Windows Core Audio (ou WCA) – que utiliza o WASAPI. O WCA é o utilizado no Windows 10 substituindo seus antecessores MME (1991), DirectSound (1995) e WDM (1998).

A Figura 2 (a) esquematiza a relação entre aplicativo e *hardware*, passando pelos servidores de áudio (JACK e PulseAudio) e pela ALSA. Nota-se o caráter complementar das etapas de comunicação da pirâmide. Ter essa noção é importante para a investigação de possíveis falhas, indica-se também *sempre* buscar ler atentamente aos códigos de erros e aos *changelogs*.

A seta vermelha na Figura 2 (a) indica que a ALSA é capaz de se comunicar diretamente com o aplicativo, a depender da compatibilidade e da complexidade no uso de áudio (nesse modo o aplicativo abre mão das funções que estão dentro JACK ou PulseAudio). No entanto, se nenhum *framework* for utilizado, o aplicativo não tem como se comunicar com o hardware, observe a Figura 2 (b).

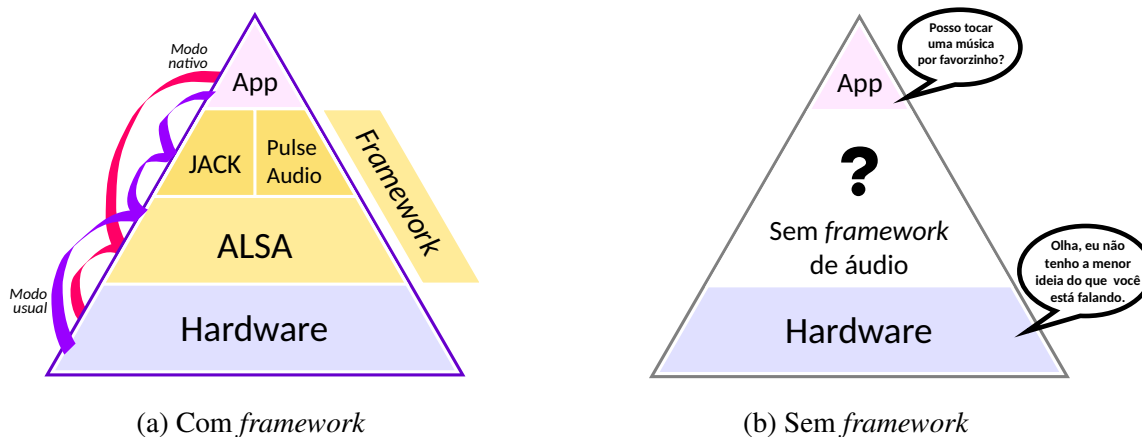


Figura 2 – Relação aplicativo-*hardware* hierarquizada (adaptado de [16]).

A maioria dos aplicativos em Linux utilizarão o PulseAudio. De forma simples, a comunidade [16] compara o PulseAudio como equivalente ao WASAPI do Windows (embora também possa ser comparado com ASIO por suas capacidades). Assim, o JACK é comparado ao *driver* ASIO do Windows, visto que é utilizado em aplicações profissionais, como em programas do tipo DAW (*Digital Audio Workstation*), por exemplo.

2.4 Comandos auxiliares do terminal do Linux

Esta seção apresenta comandos básicos que serão utilizados para acessar informações da ALSA e do PulseAudio. Primeiramente é necessário abrir o terminal do Linux, você pode abri-lo com o atalho **Ctrl** + **Alt** + **T**. Observe os comandos⁴ dispostos na Nota 1.

3 Caracterização de *hardware* e *software*

O Quadro 1 reúne informações sobre o modelo de Raspberry Pi utilizado na produção da primeira parte deste tutorial. Três tipos de dispositivos foram configurados, sendo eles (1) uma placa de expansão (HAT-GPIO, para saída de áudio); (2) uma interface de áudio (via USB); e (3) uma TV (via HDMI). Seus modelos e características podem ser consultados no Quadro 2. Uma das combinações testadas envolvendo a placa de áudio, microfone e a placa amplificadora é apresentada na Figura 3.

⁴Mais comandos podem ser encontrados em

Linux commands: <https://www.raspberrypi.org/documentation/linux/usage/commands.md>.

Nota 1 – Comandos básicos para operação no terminal do Linux.

- i. `<nome-do-comando> --help`:
estampa na tela a ajuda e as opções de utilização do comando.
Exemplo: `alsamixer --help`;
- ii. `man <nome-do-comando>`:
abre o manual de referência do comando desejado.
Exemplo: `man pulseaudio`;
- iii. `lsusb`:
mostra na tela dispositivos USB conectados ao Rpi;
- iv. `find <diretório-de-pesquisa> -name <nome-do-arquivo.extensão>`:
procura um tipo de arquivo pelo nome em um certo diretório. Substituindo o nome do arquivo por um asterisco, buscaremos todos aqueles que possuem a mesma extensão.
Exemplo 1: `find / -name asound.config`.
Exemplo 2: `find / -name *.config` (todos os arquivos .config);
- v. `whereis <arquivos-ou-programas>`:
printa na tela diretórios que contenham arquivos ou programas com o nome procurado.
Exemplo: `whereis python3.5`;
- vi. `cd <diretório>`:
seleciona o diretório inserido como sendo o ativo.
Exemplo: `cd /home/pi/Audio`;
- vii. `dir`:
comando que mostra na tela o conteúdo do diretório ativo.
- viii. `nano`:
ferramenta de leitura e edição de arquivos de texto.
Exemplo: `nano asound.config`
- ix. `sudo`:
comando que força execução de linhas do *bash* como "super usuário"^a

^aO comando `sudo` vem da expressão *superuser do*.Quadro 1 – Especificações do *hardware* e sistema operacional do Raspberry Pi.

Modelo	Raspberry Pi 3 B
CPU	Quad Core 1.2 GHz Broadcom BCM2837 64 bit
Armazenamento	Memória Flash (32 GB Micro SD)
Memória RAM	1 GB
Sistema operacional	Raspbian GNU/Linux 9 (Stretch) 32 bit

Quadro 2 – Modelo e tipo de conexão dos dispositivos testados com o Raspberry Pi 3 B.

Dispositivo	Conexão	Modelo
Placa amplificadora	HAT (GPIO)	Hifiberry Amp2
Interface de áudio	USB	C-Media CM6206
TV	HDMI	Philco PH20U21D

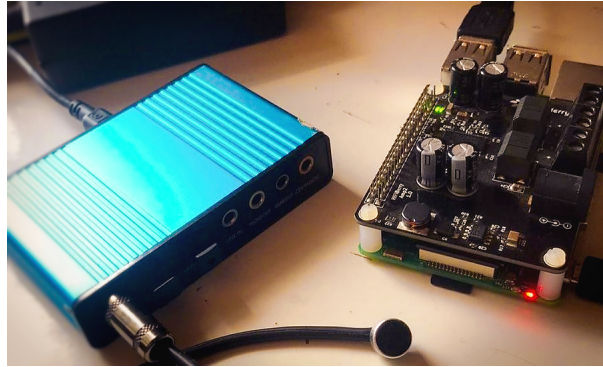


Figura 3 – Combinação de dispositivos conectados ao Raspberry: placa de áudio (USB) C-Media CM6206 (à esquerda) e placa amplificadora (HAT) Hifiberry Amp2 (à direita).

4 Metodologia

Esta seção apresenta os passos para a identificação do Raspberry Pi, bem como a instalação de pacotes essenciais e a configuração de dispositivos pelo Gnome Alsamixer ou diretamente pela ALSA.

4.1 Identificação do Raspberry Pi

O primeiro passo para configuração correta do Raspberry Pi é conhecer características como o modelo utilizado, o tipo e a versão do sistema operacional e sua arquitetura de *hardware*. Abra o terminal (com o comando `Ctrl+Alt+T`) e digite o conteúdo do Código 1. Para não haver dúvidas, simbolizaremos com \$ no início de cada linha um comando a ser escrito no *bash* do sistema. As linhas em que não há o símbolo \$ são o *output* das entradas inseridas no *bash*. O resultado da inserção dos comandos pode ser visto na Figura 4. O primeiro comando (`uname -a`) imprime na tela informações do sistema: versão do *kernel*, *hardware* e outras características. Como resposta, obtemos "armv7l", indicando um processador de 32-bits; isso ocorre porque o sistema operacional é 32-bits⁵, porém os processadores dos Raspberry Pi 3 e 4 são 64-bit, sendo a estrutura *ARM* a base da maioria dos processadores do Rpi. O segundo comando, (`cat /etc/os-release`) informa o tipo e a versão do sistema operacional utilizado, que neste caso é o Raspbian⁶ Stretch versão 9, uma distribuição Linux baseada em Debian e dedicada ao Raspberry. O terceiro comando (`grep Revision /proc/cpuinfo`) informa o código de revisão do microcomputador, com esse código você pode descobrir o modelo e mais informações⁷ como a memória RAM e o país de origem. O quarto comando retorna o modelo do seu Raspberry Pi; neste caso, o modelo utilizado para a criação deste tutorial foi o Raspberry Pi 3 B+.

Código 1 – Comandos que estampam informações sobre o sistema.

```
1 $ uname -a
2 $ cat /etc/os-release
3 $ grep Revision /proc/cpuinfo
4 $ cat /sys/firmware/devicetree/base/model; echo
```

4.2 Pacotes essenciais

O básico para que o Raspberry reproduza áudio pode ser encontrado no Código 2. A primeira linha instala os pacotes `libasound2` e `alsa-utils`, necessários para a utilização da arquitetura ALSA. As duas linhas seguintes instalam pacotes para a leitura de arquivos mp3 e para a conversão de wav para mp3. A última linha é responsável pelo carregamento do driver de áudio da saída *on-board*. Dependendo do gerenciador de ambientes escolhido, algumas funcionalidades como a ALSA e o PulseAudio já vem instaladas, sem a necessidade de refazer essa etapa.

⁵A versão Raspberry Pi OS 64-bits beta está disponível <https://www.raspberrypi.org/forums/viewtopic.php?t=275370>.

⁶Atualmente o nome do sistema operacional é Raspberry Pi OS.

⁷Disponível em <https://www.raspberrypi.org/documentation/hardware/raspberrypi/revision-codes/README.md>

```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~$ uname -a
Linux raspberrypi 4.19.36-v7+ #1213 SMP Thu Apr 25 15:08:02 BST 2019 a
rmv7l GNU/Linux
pi@raspberrypi:~$ cat /etc/os-release
PRETTY_NAME="Raspbian GNU/Linux 9 (stretch)"
NAME="Raspbian GNU/Linux"
VERSION_ID="9"
VERSION="9 (stretch)"
ID=raspbian
ID_LIKE=debian
HOME_URL="http://www.raspbian.org/"
SUPPORT_URL="http://www.raspbian.org/RaspbianForums"
BUG_REPORT_URL="http://www.raspbian.org/RaspbianBugs"
pi@raspberrypi:~$ grep Revision /proc/cpuinfo
Revision       : a02082
pi@raspberrypi:~$ cat /sys/firmware/devicetree/base/model;echo
Raspberry Pi 3 Model B Rev 1.2
pi@raspberrypi:~$

```

Figura 4 – Resultado do *bash* após inserção dos comandos vistos no Código 1.

Código 2 – Comandos para instalar pacotes essenciais e carregar o *driver* de áudio.

```

1 $ sudo apt-get install libasound2 alsa-utils #ALSA utilities
2 $ sudo apt-get install mpg321
3 $ sudo apt-get install lame #converter wav em mp3
4 $ sudo modprobe snd-bcm2835 #carregando o driver

```

Em certas ocasiões apenas alguns aplicativos interagem com o dispositivo desejado; desta forma, considera-se uma boa prática testar a saída ou entrada do dispositivo de áudio após alterações nas configurações, como testar no terminal conforme o Código 3, no Python ou no Youtube via Chromium. O Código 3 apresenta três maneiras de testar a captura e reprodução de áudio usando os comando `arecord` e `aplay`. A porção `-D` controla o dispositivo que será escolhido para realizar a ação, seguido das opções `default`, representando o dispositivo padrão (`hw:1,0`) que controla o *hardware* diretamente ou (`plughw:1,0`) que permite que *plugins* (como o PulseAudio) controlem o *hardware*. O argumento `-f cd` configura o formato de gravação como sendo 16-bits, frequência de amostragem 44.100 Hz e estéreo; `-c2` realiza a reprodução ou a captura em dois canais. Os índices 1,0 indicam a escolha do primeiro canal (0) do *segundo* dispositivo (1).

Código 3 – Exemplos de uso do comandos `aplay` e `arecord` para reprodução e captura de áudio.

```

1 #reproduzindo
2 $ aplay -D default /diretorio/teste1.wav
3 $ aplay -c2 -D hw:1,0 /diretorio/teste1.wav #acessando o hardware diretamente
4 $ aplay -D plughw:1,0 /diretorio/teste1.wav
5 #capturando áudio
6 $ arecord -D default /diretorio/teste.wav
7 $ arecord -f S16_LE -c2 -D hw:1,0 -r 44100 /home/pi/teste.wav
8 $ arecord -D plughw:1,0 --duration=7 -f cd /home/pi/teste.wav

```

4.3 Gerenciador de ambientes (Gnome)

A primeira e mais simples tentativa é buscar reconhecimento do dispositivo via seu gerenciador de ambientes. Há diversos tipos como o **Gnome** e o **PIXEL**, por exemplo. A principal vantagem do gerenciador de ambientes é poder selecionar seu dispositivo de áudio ou pelo menos saber se ele está sendo reconhecido a partir da sua *interface gráfica*⁸ de fácil utilização.

O Gnome é um gerenciador de ambientes utilizado em plataformas Linux. Uma parte dele, o Gnome AlsaMixer, é um gerenciador de dispositivos de áudio (a partir do pacote `alsa-utils`) em forma de GUI. Com ele é possível alterar o dispositivo selecionado de uma maneira muito semelhante ao Windows, podendo-se controlar saídas ou entradas. O Código 4 contém os comandos necessários para a instalação ou atualização do Gnome AlsaMixer.

⁸Também chamada de GUI, do inglês *Graphical User Interface*.

Código 4 – Instalação ou atualização do Gnome AlsaMixer.

```

1 $ sudo apt-get update
2 $ sudo apt-get install gnome-alsamixer #instalar
3 $ sudo apt-get upgrade gnome-alsamixer #atualizar

```

A Figura 5 apresenta a interface gráfica desta ferramenta. No canto superior direito podemos ver um ícone em forma de alto-falante: basta clicar com o botão esquerdo do mouse e as opções para chegar nessa tela aparecem.

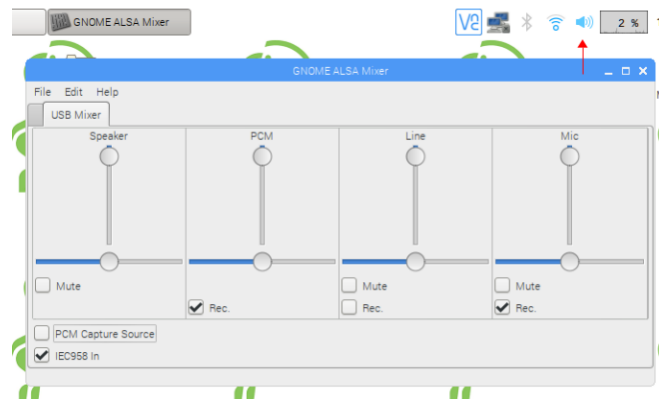


Figura 5 – Interface gráfica do Gnome AlsaMixer.

4.4 Configurando a ALSA

Será abordado nesta seção as dependências da ALSA, sua forma de funcionamento e a configuração de dispositivos via `alsamixer` ou por alteração do `asound.config`, arquivo de configuração da ALSA.

4.4.1 Checando dependências

O `libasound2`, `alsa-oss` e `alsa-utils` são bibliotecas e pacotes de baixo nível que já vem com o Raspbian [17], apoiando a estrutura de controle de áudio (ALSA). Com ela podemos acessar várias funções com os comandos `amixer`, `alsamixer`, `alsactl`, `aplay`, `arecord`, `speaker-test` etc. O Código 5 ensina como instalar⁹ ou atualizar a ALSA. A versão utilizada para criação deste tutorial foi a (.1.3-1).

Código 5 – Comando para atualização do pacote `alsa-utils`.

```

1 $ sudo apt-get install libasound2 alsa-utils #instalar
2 $ sudo apt-get upgrade libasound2 alsa-utils #atualizar

```

4.4.2 Listando dispositivos

Digite o conteúdo do Código 6 para listar dispositivos de reprodução e captura, respectivamente.

Código 6 – Comandos que listam dispositivos para reprodução e captura de áudio, respectivamente.

```

1 $ aplay -l
2 $ arecord -l

```

O resultado do `bash` deve parecer como na Figura 6. Podemos notar que os dispositivos são identificados como “card 0” ou “card 1”. Muitas vezes (e não todas) iremos ver as expressões `card` e `sound card` referindo-se ao que, neste tutorial, chamamos de dispositivos de áudio (e.g., HATs e interfaces de áudio). O primeiro dispositivo é o HiFiBerry¹⁰ Amp2 e o segundo é dispositivo de áudio USB da C-Media¹¹ CM6206.

⁹Para instalação e mais detalhes, visite: https://alsa.opensrc.org/Quick_Install.

¹⁰Mais informações em :

<https://www.hifiberry.com/blog/introducing-the-amp2-more-power-higher-sample-rates-more-affordable/>.

¹¹Informações e *datasheet*: https://www.cmedia.com.tw/products/USB20_FULL_SPEED/CM6206.


```

pi@raspberrypi:~$ aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: sndrpihifiberry [snd_rpi_hifiberry_dacplus], device 0: HiFiBerry pcm512x-hifi-0 []
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 1: Device [USB Sound Device], device 0: USB Audio [USB Audio]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
pi@raspberrypi:~$ arecord -l
**** List of CAPTURE Hardware Devices ****
card 1: Device [USB Sound Device], device 0: USB Audio [USB Audio]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
pi@raspberrypi:~$

```

Figura 6 – Lista de dispositivos de reprodução e de captura de áudio, respectivamente.

Ainda na Figura 6 podemos notar que o dispositivo identificado como “card 0” possui apenas controle para a saída de áudio (*playback*), pois não aparece na lista dos que capturam áudio. Sendo assim, focaremos no reconhecimento da placa de áudio USB da C-Media, ou “card 1”, pois esta possui capacidade tanto de *input* quanto de *output*. A organização da ALSA fica mais compreensível ao se conceituar certos nomes recorrentes, como:

- **card:** é um índice que define a placa de som utilizada, seja ela física (como tipos *onboard*, via USB etc.) ou virtual como dispositivos MIDI virtuais [18];
- **device:** dispositivos do *card*. Podem ser, por exemplo, dispositivos de audio digital (PCM), de controle (*mixers*) ou MIDI [18]; e
- **subdevice:** está associado com as portas do *device*, como por exemplo capturar audio a partir de canais simultâneos [19].

4.4.3 Configurando o alsamixer

Este é um modo de gerenciar dispositivos de áudio a partir do terminal, utilizando um *mixer* da ALSA. O Código 7 mostra a sequência de comandos para controlar e salvar configurações relativas aos dispositivos de áudio. A primeira linha (`alsamixer`) dá origem à tela de controle – a Figura 7 (a) ilustra esse resultado. O destaque em vermelho na parte de cima da Figura 7 (a) identifica o nome do dispositivo e controles que podem ser utilizados para a navegação nesse *mixer*. Circulado em vermelho e na parte inferior, ainda na Figura 7 (a), vemos as letras “MM”, simbolizando que aquele canal está mudo. Para ativar ou desativar o *mute* devemos selecionar o canal (seja ele saída ou entrada) e apertar a tecla **M**. As setas do teclado **↑**, **↓**, **→** e **←** são utilizadas para alterar opções e para selecionarmos o dispositivo, basta apertamos a tecla **F6**, como ilustra a Figura 7 (b).

Código 7 – Comandos para abrir o *mixer* da ALSA e salvar suas configurações, respectivamente.

```

1 $ alsamixer #interface de controle
2 $ sudo alsactl store #salve

```

Estão sendo reconhecidos dois dispositivos: o HAT (da HiFiBerry) e o dispositivo USB (C-Media), acompanhados dos seus respectivos índices de identificação de *sound cards*. Após realizar as modificações necessárias, deve-se digitar o conteúdo da segunda linha do Código 7: `sudo alsactl store`. Esse comando garante que as configurações editadas sejam salvas. O comando `sudo`, neste caso, é estritamente necessário para que sejam salvas as alterações para todos os usuários e suas permissões (observe mais detalhes na Nota 2).

Figura 7 – (a) Controles do alsamixer e (b) seleção de dispositivos (utilize as setas para alternar entre figuras, com o Acrobat Reader, por exemplo).

Nota 2 – Detalhes na utilização do alsamixer.

Neste método, utilizando o `alsamixer`, devemos:

- 1) verificar dispositivos disponíveis de captura e reprodução de áudio, conforme o Código 6;
- 2) abrir *mixer* da ALSA com o comando `alsamixer` e configurar o necessário, como:
 - selecionar o dispositivo de áudio desejado, utilizando a tecla `F6`;
 - aumentar (zerar) o ganho dos canais de entrada ou saída;
 - tirar canais do mudo; e
 - selecionar qual entrada recebe o mic (line ou mic, por exemplo).
- 3) salvar todas as configurações com o comando `sudo alsactl store` do Código 7; e
- 4) testar com o Código 3.

4.4.4 Configurando a ALSA: `asound.config`

Existe uma maneira de configurar a ALSA de forma mais profunda: alterando seu arquivo de configuração. Nesse caso, devemos acessar o terminal na pasta que contém o arquivo `asound.config` e alterá-lo. O Código 8 enumera a sequência a ser seguida para encontrar e abrir o arquivo, sendo este crucial para se modificar o dispositivo de áudio padrão – veja pormenores na Nota 3. Se, por alguma razão ele não esteja na pasta esperada, pode-se usar a sequência de comandos da parte inferior do Código 8. Caso ele tenha sido excluído, recomenda-se recriar um arquivo de mesmo nome na pasta original.

Código 8 – Linhas de comando para se chegar ao arquivo `asound.config`.

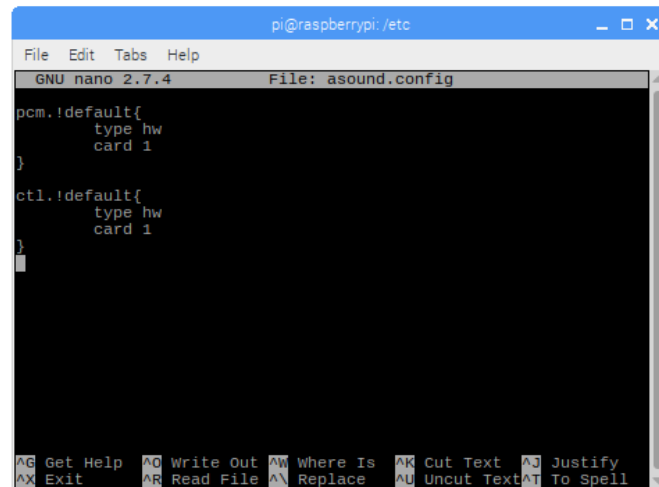
```

1 $ cd /etc/
2 $ dir # mostra conteúdo da pasta, que deve conter asound.config
3 $ sudo nano asound.config #para abrir arquivo
4 #Caso não esteja na pasta /etc/:
5 $ whereis asound.config #uma forma de procurar
6 $ sudo \ find -name asound.config #outra forma de procurar

```

O resultado da linha `sudo nano asound.config` do Código 8 pode ser visto na Figura 8. A sigla `pcm` significa *Pulse Code Modulation*, sendo a forma padrão para se representar digitalmente sinais de áudio em dispositivos, tratando-se de um método utilizado para amostrar e *quantizar* um sinal de áudio (antes analógico). O `pcm` associado ao tipo `hw` dá à ALSA a habilidade de controlar o *hardware* “card 1” criando um *sound card* virtual [20], dando acesso ao *device* e aos *subdevices* definidos pela lógica da ALSA. A porção `ctl` está associada ao controle do *hardware* (como um *mixer*). A palavra *default* representa que essas são as configurações padrão, podendo-se substituir essa palavra por outras para criar

novos perfis `pcm` e `ctl` personalizados para a ALSA, como será demonstrado na Parte 2 deste tutorial [1]. O índice “1” logo após `plughw` (no Código 3) deve ser o índice correspondente ao *card* (Figura 8) que simboliza o dispositivo que se deseja comunicar (veja o Código 3 para realizar testes).



```

pi@raspberrypi: /etc
File Edit Tabs Help
GNU nano 2.7.4 File: asound.conf
pcm.!default{
    type hw
    card 1
}

ctl.!default{
    type hw
    card 1
}
AG Get Help  AR Write Out  AW Where Is  AK Cut Text  AJ Justify
AX Exit      AR Read File  AN Replace  AU Uncut Text AT To Spell

```

Figura 8 – Conteúdo do arquivo `asound.conf` após alteração.

Nota 3 – Configuração da ALSA via `asound.conf`.

Neste método de configuração da ALSA via `asound.conf`, devemos:

- 1) encontrar e abrir o arquivo `asound.conf` (Código 8);
- 2) alterar os índices para aqueles que correspondam ao *card* que desejamos ativar; e
- 3) testar via terminal e outros meios (consulte o Código 3).

Uma outra maneira de alterar qual dispositivo será padrão, é digitando no terminal (*bash*) o endereço `cd /usr/share/alsa` e abrir o arquivo desejado com o comando `sudo nano alsa.conf` e alterar o índice das linhas “`defaults.ctl.card 0`” e “`defaults.pcm.card 0`”. Há ainda inúmeras maneiras de configurar a ALSA – configurações da arquitetura em conjunto com o PulseAudio serão exemplificadas na segunda parte deste tutorial [1].

5 Considerações finais

A interface gráfica é a maneira mais intuitiva de gerenciar dispositivos de áudio, mostrando-se uma ferramenta valiosa para configurações mais simples. Muitas interfaces inclusive utilizam o PulseAudio internamente, poupando o trabalho de configurar em níveis mais baixos. As configurações feitas via terminal podem ser mais complexas, uma vez que podemos acessar e modificar arquivos de configuração de áudio, podendo-se administrar de forma mais direta a relação entre *kernel*, *hardware* e aplicativo. É importante frisar que a configuração muitas vezes deve ser feita a partir de uma investigação que associa certas noções básicas, ao levar em consideração os seguintes itens:

- muitas maneiras de se configurar dispositivos podem ser feitas de forma complementar, ou seja, deve-se concatenar métodos a fim de uma configuração específica;
- a necessidade de instalação de pacotes, bibliotecas e alguns módulos será constante e a checagem deve sempre existir, atentando-se sempre aos *logs* de erro; e
- deve-se compreender que as aplicações podem usar *framework* específicos, como trabalhar apenas através da ALSA (nível de núcleo) ou necessitando da ajuda de um intermediário (*middleware*) como JACK ou PulseAudio.

A segunda parte deste tutorial [1] irá tratar de configurações mais complexas, abordando ferramentas como o PulseAudio e o PortAudio.

Agradecimentos

A autora e os autores gostariam de agradecer ao Curso de Engenharia Acústica da Universidade Federal de Santa Maria (UFSM) por todo apoio institucional e infraestrutura. Ademais, agradecemos também ao Joseph Lacey pela revisão de inglês do *abstract* e à Laís Smeha pela revisão textual.

Referências

- [1] Circe, Bárbara; Fonseca, William D’A.; Jacomussi, Leonardo e Mareze, Paulo H. Tutorial: configuração de dispositivos de áudio no Raspberry Pi – Parte 2. In *Acústica 2020 (XI Congresso Ibérico de Acústica, Tecniacústica 2020 e 51º Congresso Español de Acústica)*, páginas 1–11, Faro, Portugal, Outubro 2020.
- [2] Raspberry Pi Foundation – Teach, Learn, and Make with Rasp. Pi. <https://www.raspberrypi.org>. Acessado em: jun. de 2020.
- [3] Fonseca, William D’A.; Jacomussi, Leonardo e Mareze, Paulo H. Raspberry Pi: A Low-cost Embedded System for Sound Pressure Level Measurement. In *49th Internoise - International Congress and Exposition on Noise Control Engineering*, páginas 1–12, Seul, Córrea do Sul, 2020. Disponível em <http://bit.ly/SLM-Internoise2020>.
- [4] Li, Kongyan; Chan, Chloe H. K.; Rajendran, Vani G.; Meng, Qinglin; Rosskothén-Kuhl, Nicole e Schnupp, Jan W. H. Microsecond sensitivity to envelope interaural time differences in rats. *The Journal of the Acoustical Society of America*, 145(5), 2019. doi: [10.1121/1.5099164](https://doi.org/10.1121/1.5099164).
- [5] Schlieper, F.; Aspöck, L. e Vorländer, M. A binaural demonstrator on a single board computer. In *German Conference on Acoustics - DAGA 2016*, Aachen, Germany, 2016. Disponível em https://pub.dega-akustik.de/DAGA_2016.
- [6] Florentin, Juliette e Verlinden, Olivier. Autonomous wildlife soundscape recording station using Raspberry Pi. In *International Congress on Sound and Vibration - ICSV 24*, London, UK, 2017. Disponível em <https://www.iiav.org>.
- [7] Raspberry Pi 3 Model B. <https://www.raspberrypi.org/products/raspberry-pi-3-model-b>. Acesso em: ago. de 2020.
- [8] Raspberry Pi 4 Model B. <https://www.raspberrypi.org/products/raspberry-pi-4-model-b>. Acesso em: set. de 2020.
- [9] Python Programming Language. <https://www.python.org>. Acessado em: set. de 2020.
- [10] Raspberry Pi and realtime, low-latency audio. https://wiki.linuxaudio.org/wiki/raspberrypi#audio_software_repository. Acesso em: set. de 2020.
- [11] RPi Expansion Boards - eLinux.org. https://elinux.org/index.php?title=RPi_Expansion_Boards#Sound. Acesso em: set de 2020.
- [12] RPi VerifiedPeripherals - eLinux.org. https://elinux.org/RPi_VerifiedPeripherals#USB_Sound_Cards. Acesso em: set. de 2020.
- [13] Alsa Opensrc Org - Independent ALSA and linux audio support site. <https://alsa.opensrc.org/ALSA>. Acesso em: set. de 2020.
- [14] PulseAudio. <https://wiki.archlinux.org/index.php/PulseAudio>. Acesso em: set. de 2020.
- [15] JACK Audio Connection Kit. <https://jackaudio.org>. Acesso em: set de 2020.
- [16] How Linux Audio Works vs. Windows Audio in 2017 – Learn Digital Audio. <https://www.learn-digitalaudio.com/how-linux-audio-works-vs-windows-audio-2017>. Acesso em: ago. de 2020.
- [17] Phillips, Stephen C. Sound configuration on Raspberry Pi with ALSA. <http://blog.scphillips.com/posts/2013/01/sound-configuration-on-raspberry-pi-with-alsa>. Acesso em: ago. de 2020.
- [18] Alsa Opensrc Org - ALSA: Advanced Linux Sound Architecture. <https://alsa.opensrc.org/Device>. Acesso em: set. de 2020.
- [19] Understanding ALSA device, subdevice and cards. <http://delogics.blogspot.com/2014/11/understanding-alsa-device-subdevice-and.html>. Acesso em: ago. de 2020.
- [20] Asoundrc - AlsaProject. <https://www.alsa-project.org/wiki/Asoundrc>. Acesso em: ago, de 2020.