

FIA 2020/22

XII CONGRESSO/CONGRESO IBEROAMERICANO DE ACÚSTICA
XXIX ENCONTRO DA SOCIEDADE BRASILEIRA DE ACÚSTICA - SOBRAC

Florianópolis, SC, Brasil

Aquisição de sinais acústicos utilizando microfones MEMS digitais e plataforma embarcada Teensy (compatível com Arduino)

Mello, F. R.¹; Fonseca, W. D'A.¹; Mareze, P. H.¹

¹ Engenharia Acústica, Universidade Federal de Santa Maria, Santa Maria, RS, Brasil,
{felipe.mello, will.fonseca, paulo.mareze}@eac.ufsm.br

Resumo

Devido ao seu pequeno tamanho e capacidade de montagem em placas de circuito impresso, microfones micro-eleto-mecânicos (MEMS) são amplamente utilizados em dispositivos como *smartphones*, *laptops* e aplicações IoT (Internet of Things). Recentemente, vêm ganhando espaço na indústria de instrumentação para acústica e áudio, tanto em aplicações com poucos sensores (um a quatro), quanto naquelas que envolvem um grande número de transdutores, como, por exemplo, no imageamento acústico (*beamforming*) e gravação imersiva (arranjo ambisonics). Sua vantagem consiste no preço e tamanho reduzidos, aliados a uma resposta em frequência adequada, relação sinal-ruído apropriada e alta padronização (de fabricação). Alguns modelos possuem saída digital, dispensando a necessidade de um conversor analógico-digital. Nesse caso, o sinal é transmitido por meio de um protocolo de comunicação e faz-se necessário um dispositivo capaz de se comunicar com o microfone e adquirir seus sinais. O presente trabalho tem como objetivo demonstrar os aspectos práticos acerca da aquisição dos sinais de um microfone MEMS digital por meio do microcontrolador Teensy. Ele é compatível com Arduino, possuindo a mesma lógica e linguagem de programação, facilitando a adaptação de projetos. Um breve histórico da tecnologia MEMS é apresentado, seguido por uma descrição dos protocolos de comunicação normalmente implementados nos microfones digitais. Na sequência, comenta-se sobre as características técnicas do Teensy 4.0, bem como as ferramentas de processamento de sinais de áudio presentes em sua biblioteca, apresentando um panorama dos tipos de objeto nela contidos. Por fim, é descrito o passo a passo de como configurar o Teensy 4.0 para realizar a aquisição de sinais de um microfone digital I2S (Inter-IC Sound) e transmiti-lo via USB para um computador. Com isso, pode-se plotar os dados ou tecer novos processamentos (como o cálculo do nível de pressão sonora, por exemplo).

Palavras-chave: tutorial, microfones I2S, processamento de sinais, filtros digitais, instrumentação.

PACS: 43.10.Sv, 43.38.Kb, 43.60.-c, 01.30.Rr.

Acoustic signal acquisition using digital MEMS microphones and Teensy 4.0 microcontroller

Abstract

Due to their small size and surface-mounting capability, micro-electrical-mechanical (MEMS) microphones are widely present in gadgets like smartphones, laptops, and IoT (internet of things) applications. Recently, they have been conquering space in the audio and acoustic instrumentation industry, both in applications involving a few sensors (one to four) and in those that require a great number of transducers, such as acoustic imaging (*beamforming*) and immersive recording (*ambisonics* arrays). Their advantages are a low-cost and small size, adequate frequency response, and high manufacturing standardization. Some models have a digital output, dispensing the use of external analog-to-digital converters (ADC). In this case, the microphone signal is transmitted through a digital communication protocol, requiring a controller device for the data acquisition. This work aims to show how to acquire data from a digital MEMS microphone using the Teensy 4.0 microcontroller, an Arduino-compatible board. A brief historical background of the MEMS technology is shown, followed by a description of the communication protocols found in digital models. In the following, Teensy's technical characteristics are detailed, focusing on its library for audio signal processing and providing a quick overview of its available tools. Finally, a tutorial explains how to configure Teensy to acquire signals from an I2S digital MEMS microphone and transmit it through USB to a computer, making it possible to plot the results and do further processing (such as sound pressure level calculations).

Keywords: tutorial, I2S microphones, signal processing, digital filters, instrumentation.



1. INTRODUÇÃO

Com o avanço tecnológico, sucessiva miniaturização e sofisticação dos dispositivos eletrônicos voltados ao mercado consumidor, microfones micro-eleto-mecânicos (ou MEMS¹) vêm ganhando muita popularidade desde o início dos anos 2010. Por volta de 2014, seu número de unidades vendidas superou a dos conhecidos microfones de eletreto [1], e a tendência é que sua presença no mercado só venha a aumentar. Dentre os motivos que levaram a esse crescimento, pode-se destacar o baixo custo, possibilidade de fabricação em massa (com alta padronização), robustez e capacidade de integração em placas de circuito impresso.

Para além da indústria de eletrônicos, microfones MEMS vêm ganhando espaço também nas áreas de áudio profissional e instrumentação em acústica, em especial em aplicações que envolvem um elevado número de transdutores, como *câmeras acústicas* e arranjos para gravação de som imersivo [2–4]. Ademais, constantes avanços em técnicas de fabricação têm permitido a construção de microfones com características acústicas cada vez melhores, abrindo caminho para seu uso em sonômetros e dispositivos de medição profissionais [5–7]. Nesse contexto, os modelos da família XENSIV², da empresa Infineon, e os da família SBM100³ (com transdução ótica), da empresa sensiBel, são exemplos de dispositivos com faixas dinâmicas até então sem precedentes dentre os microfones MEMS.

Este trabalho tem como objetivos apresentar ao leitor um apanhado geral acerca da tecnologia dos microfones MEMS, em especial os modelos com saída digital, e um tutorial de como realizar a aquisição e processamento dos sinais de um microfone MEMS digital utilizando um microcontrolador compatível com Arduino (nomeado de Teensy⁴). O texto está organizado da seguinte maneira: a Seção 2 apresenta um breve histórico da tecnologia de sensores MEMS, com ênfase nos microfones; a Seção 3 descreve os aspectos construtivos dos microfones MEMS, suas características acús-

ticas e os protocolos de comunicação presentes em modelos digitais; a Seção 4 comenta sobre o microcontrolador Teensy 4.0, suas características técnicas e ferramentas para processamento de áudio; a Seção 5 inicia a parte prática, apontando ao leitor os requisitos de *software* e *hardware* necessários ao tutorial; a Seção 6 contém dois exemplos práticos, o primeiro envolvendo a captura do sinal de dois microfones MEMS digitais, e o segundo demonstrando como avaliar a resposta em frequência de um filtro implementado no Teensy; por fim, a Seção 7 encerra o artigo com algumas considerações finais.

2. BREVE HISTÓRICO

A tecnologia dos sistemas micro-eleto-mecânicos (MEMS) é relativamente recente, com os primeiros artigos e análises teóricas datados da década de 1950 [8, 9]. Inicialmente denominados “micromáquinas”, os primeiros sensores MEMS foram concebidos a partir da curiosidade de pesquisadores em saber até que ponto poderia ser utilizada a tecnologia que deu origem aos transistores — a manipulação do silício. Contudo, foi apenas na década de 1960 que os primeiros sensores de pressão baseados em silício passaram a ser desenvolvidos [10, 11] e só na década de 1970 que técnicas de construção mais sofisticadas passaram a ser implementadas (ainda rudimentares, se comparadas às atuais) [12].

Foi com o trabalho de Hohm e Sessler⁵, de 1983, que os sensores acústicos iniciaram sua jornada rumo ao MEMS. Ao longo das décadas de 1980 e 1990, inúmeras pesquisas no tema foram realizadas e formaram a base para a construção dos sensores modernos. Finalmente, a comercialização em massa de microfones MEMS iniciou-se em 2003, com a linha SiSonic da empresa norte-americana Knowles, seguida por empresas como a dinamarquesa Sonion, a alemã Infineon, entre outras. Nesse contexto, a indústria dos dispositivos eletrônicos portáteis (como *smartphones*, *headsets*, *notebooks* etc.) foi uma grande catalisadora para a popularização dos microfones MEMS, sendo a Apple uma das primeiras empresas a adotá-los em seus produtos (a partir do iPhone 4, em 2010 [14]).

¹Do inglês *Micro-Electrical-Mechanical Systems*.

²<https://www.infineon.com/cms/en/product/sensor/mems-microphones>.

³<https://sensibel.com/product>.

⁴<https://www.pjrc.com/teensy>.

⁵Sessler é também um dos inventores dos microfones de eletreto, junto com Jim West [13].

Para mais informações históricas acerca dos microfones MEMS, recomenda-se os artigos de Sessler [15] e Scheeper [16], os quais contêm uma análise detalhada acerca do desenvolvimento dos primeiros microfones MEMS na década de 80 e início dos anos 90. Ademais, os artigos de Elko e Harney [13], Malcovati e Baschiroto [1] e Zawawi et al. [14] oferecem uma visão atualizada do tema, apresentando também aspectos técnicos e construtivos da tecnologia, bem como amplas revisões bibliográfica. Finalmente, o livro de Mikko Suvanto apresenta uma introdução completa aos transdutores acústicos MEMS, incluindo informações sobre sua caracterização eletroacústica e implementação [17]⁶.

3. MICROFONES MEMS

Um microfone é um transdutor, ou seja, um dispositivo capaz de converter uma forma de energia em outra. Neste caso, tem-se um transdutor acústico, responsável por converter um sinal sonoro em um sinal elétrico. Existem diversas formas de se realizar tal conversão e, com isso, existem diversos tipos de microfones. Dentre os mais comuns, pode-se citar:

- **Eletrodinâmicos:** compostos por uma bobina móvel acoplada a um diafragma. As ondas sonoras movem o diafragma, que por sua vez move a bobina, imersa em um campo magnético, gerando assim um sinal de corrente elétrica análogo ao sinal acústico;
- **Piezoelétricos:** baseados em materiais cristalinos que, ao se deformarem pela ação de uma pressão sonora, geram um sinal de tensão análogo ao sinal acústico;
- **Piezo-resistivos:** baseados em materiais que, ao se deformarem, tem sua resistência elétrica alterada. Com isso, por meio de circuitos especiais, é possível traduzir as deformações ocasionadas por uma onda sonora em um sinal elétrico;
- **Óticos:** a transdução ocorre por meio de interferometria. Nesse caso, há um laser apontado para o diafragma (que se move com a onda sonora) e um sensor responsável por captar a luz refletida. Alterações na posição do diafragma

geram alterações na luz refletida, que por sua vez é convertida por um circuito fotodetector em um sinal elétrico análogo; e

- **Capacitivos:** são os mais populares em conjunto com os eletrodinâmicos. Os microfones capacitivos, também conhecidos como condensadores, possuem um diafragma móvel paralelo a uma placa fixa (o *backplate*), ambos condutores, formando um capacitor. Quando uma onda sonora incide no diafragma, este se move, gerando variações de capacitância que são transformadas em um sinal elétrico análogo ao acústico. Nesse grupo encontram-se também os *microfones de eletreto*. Para mais informações, consulte Wong e Embleton [18].

Dentre os princípios de transdução citados, apenas o eletrodinâmico não é utilizado em microfones MEMS. Assim, é possível encontrar transdutores acústicos MEMS piezoelétricos, piezo-resistivos, óticos e capacitivos. Todavia, os condensadores constituem cerca de 80% dos modelos comerciais, uma vez que com esse princípio é possível construir dispositivos com maior sensibilidade, menor consumo de energia e aumentada compatibilidade com produção em massa [1].

A Figura 1 contém uma ilustração de um microfone MEMS capacitivo. Na figura, uma onda sonora incide na porta de entrada (nesse caso, do tipo *bottom-port*, vide Figura 2), movimentando o eletrodo móvel (diafragma) e variando a distância x entre o diafragma e o *backplate* (eletrodo perfurado fixo). A oscilação de x , por sua vez, ocasiona numa variação da capacitância entre os eletrodos, que é convertida em uma variação de tensão V pelo circuito polarizador.

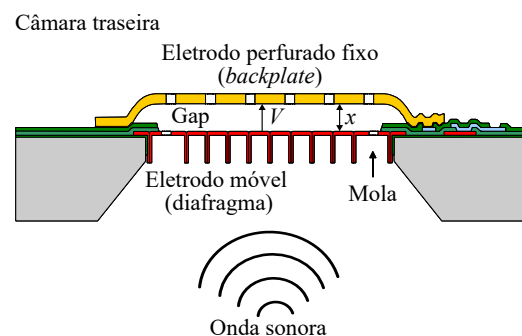


Figura 1: Ilustração de um microfone MEMS capacitivo. V representa a tensão de polarização das placas paralelas e x a distância entre ambas (adaptado de Malcovati & Baschiroto [1]).

⁶Mikko também publicou uma série de vídeos sobre o tema em seu [canal no YouTube](#)

Como comentado na Seção 2, os microfones MEMS são construídos a partir da manipulação de silício. Existem diversas técnicas para tal fim, mas, no geral, as peças constituintes do transdutor (como o diafragma e *backplate*, no caso dos capacitivos) são moldadas em chips de silício, os quais também podem conter (em um mesmo chip) circuitos de condicionamento de sinal e interfaces de saída. No entanto, o mais comum é que os circuitos auxiliares sejam construídos em um chip à parte⁷, que é montado ao lado do transdutor, sendo que todos estão contidos em uma mesma caixa [14] (vide Figura 2). É ainda interessante aclarar que existem MEMS com saída analógica [19, 20] (Figura 4 (a)) e os com saída digital [5, 6, 21] (Figuras 4 (b) e 4 (c)), foco deste artigo.

Deste modo, é possível construir dispositivos de tamanho extramente reduzido, os quais possuem integrado um circuito condicionador de sinal e, em modelos com saída digital, um conversor analógico para digital (ADC) — vide diagrama de blocos na Figura 4 ou Apêndice A. Ademais, microfones MEMS costumam ser baratos, resistentes a choques mecânicos e mudanças de temperatura (sua sensibilidade mantém-se constante com o tempo), e possuem alto nível de padronização (devido à técnica de fabricação). Os modelos digitais, em específico, também possuem alta resistência à interferência eletromagnética.

Todos esses fatores, somados à possibilidade de soldá-los diretamente em placas de circuito impresso, tornaram os microfones MEMS onipresentes em dispositivos eletrônicos portáteis, como *smartphones*, por exemplo. Outrossim, grande interesse tem surgido em sua utilização para fins de medição, especialmente em sistemas que envolvem muitos transdutores, como câmeras acústicas [23], sistemas de monitoramento de ruído [5, 24]

⁷Chamado de ASIC, ou ainda *Application-Specific Integrated Circuit*.

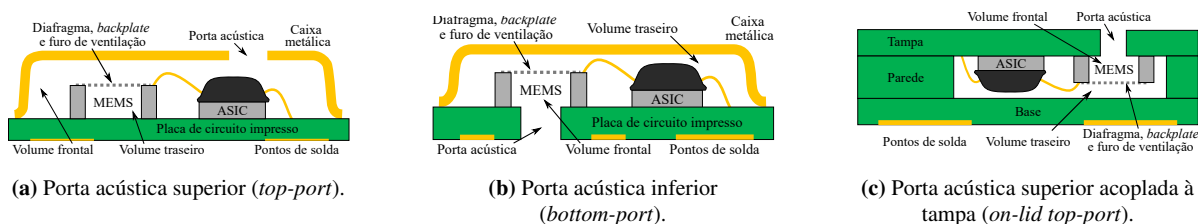


Figura 2: Três configurações internas em microfone MEMS capacitivos (adaptados de Knowles [22]).

e gravadores de som imersivo [25] (como o ZM-1, da Zylia).

3.1 Características acústicas

As características acústicas de um microfone MEMS, em especial sua resposta em frequência e relação sinal-ruído (SNR), estão diretamente relacionadas à configuração interna do dispositivo — em específico a relação entre as dimensões do furo de ventilação e os volumes frontal e traseiro — e propriedades mecânicas da membrana. Em suma, o volume frontal determina a resposta em altas frequências⁸, enquanto que o volume traseiro e furo de ventilação determinam o comportamento em baixas frequências [26]. Microfones MEMS, usualmente, possuem resposta plana entre 100 Hz e 10 kHz (± 3 dB, variando de modelo para modelo, vide Figura 3). A Figura 5 (b) contém a resposta em frequência medida do microfone utilizado neste tutorial⁹.

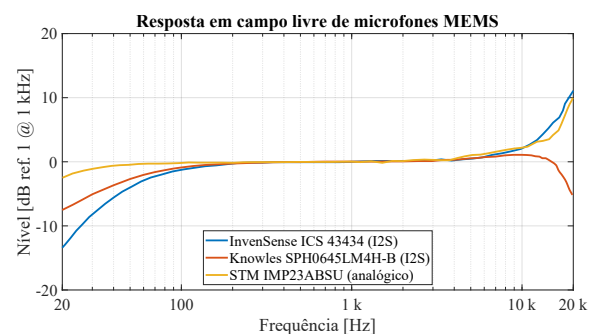


Figura 3: Respostas em frequência dos microfones MEMS InvenSense ICS 43434 (digital), Knowles SPH0645LM4H-B (digital) e STM IMP23ABSU (analógico) retiradas das respectivas folhas de dados.

Por fim, os modelos comercializados atualmente possuem uma das três configurações internas listadas abaixo (e ilustradas na Figura 2):

⁸Microfones MEMS possuem, caracteristicamente, uma ressonância em altas frequências pelo fato de que sua cavidade frontal atua como um ressonador de Helmholtz.

⁹Mais informações sobre a metodologia de medição e resultados serão publicados em um trabalho futuro.

- Porta acústica superior (ou *top-port*); ou
- Porta acústica inferior (ou *bottom-port*); ou
- Porta acústica superior acoplada à tampa (ou *on-lid top port*).

3.2 Microfones digitais – protocolos de comunicação

Os microfones MEMS com saída digital transmitem seus sinais em formato serial por meio de um protocolo de comunicação. Dentre os modelos disponíveis no mercado, existem três protocolos principais:

1. Modulação por densidade de pulso, ou *pulse density modulation* (PDM);
2. Inter-IC Sound, ou I²S; e
3. Multiplexação por divisão temporal, ou *time-division multiplex* (TDM).

Em resumo, microfones PDM enviam sinais com 1 bit de resolução e altíssima taxa de amostragem. Para que esse sinal possa ser processado como áudio, é necessário transformá-lo para o formato PCM (*pulse code modulation*). Tal conversão pode exigir muito poder computacional, não sendo possível realizá-la em *tempo real* na maioria dos microcontroladores [23]. Microfones PDM possuem, normalmente, cinco pinos (vide Figura 4 (b)):

1. **Vdd**: tensão de alimentação (normalmente entre 3,3 V e 5,0 V);
2. **Gnd**: terra;
3. **L/R**: define em qual dos canais o microfone transmitirá seus dados. Normalmente, quando aterrado (estado *low*), envia sinais no canal esquerdo. Quando alimentado (estado *high*), en-

via sinais no canal direito¹⁰;

4. **CLK**: é o pino dedicado ao *clock*. Em eletrônica, o sinal de *clock* constitui-se por um trem de pulsos que oscila entre os estados *high* e *low*, em uma frequência específica, e determina a taxa de funcionamento do sistema como um todo (quantos bits são transmitidos por segundo); e

5. **DATA**: pino para transmissão do sinal do microfone em formato PDM (a taxa de transmissão é definida pelo **CLK**).

Os microfones I²S, por sua vez, já possuem integrada uma interface capaz de transmitir sinais no formato PCM (via protocolo I²S). Em geral, microfones MEMS I²S transmitem sinais com resolução de 16 à 32 bits¹¹, sendo que sua taxa de amostragem depende da frequência de *clock* utilizada no processador (podendo chegar a mais de 96 kHz). Os pinos de um microfone I²S são semelhantes ao PDM, com a adição de um sexto denominado **WS**, que controla qual dos canais está enviando o sinal de áudio em determinado instante (vide Figura 4 (c)). Ademais, certos modelos podem conter ainda um sétimo pino para controle, como definição de modo de funcionamento, por exemplo (**CHIPEN**).

É importante ressaltar que, tanto a interface PDM, quanto a I²S, conseguem transmitir sinais estéreo (dois canais) em uma mesma linha de da-

¹⁰Cabe sempre consultar a folha de dados (*datasheet*), uma vez que podem existir diferenças entre modelos.

¹¹Microfones MEMS I²S trabalham, normalmente, com 32 bits de resolução. Contudo, nem sempre todos os bits estão disponíveis para o sinal de áudio. O modelo SPH0645LM4H da Knowles, por exemplo, reserva apenas 18 bits para o áudio, completando o restante com zeros.

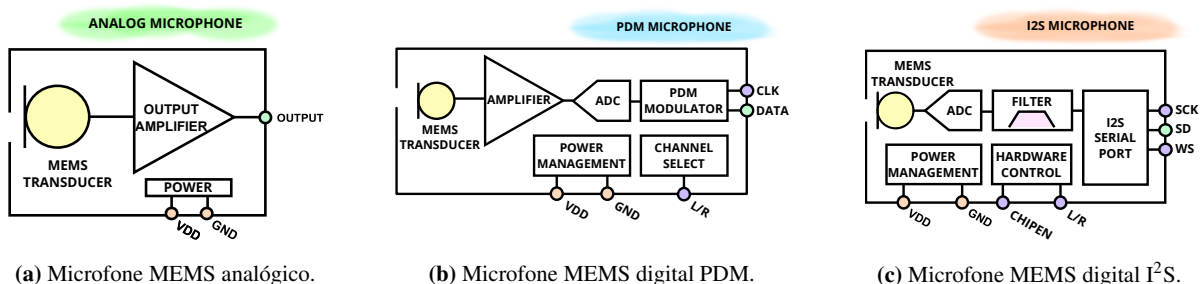


Figura 4: Diagrama de blocos de um microfone MEMS analógico e dois digitais (adaptado de Lewis [21]). Uma versão em forma de circuitos elétricos encontra-se no Apêndice A.

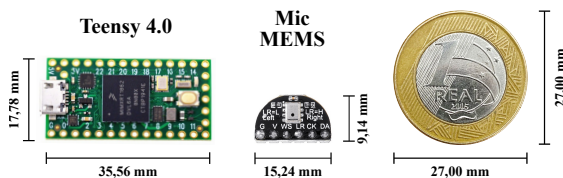


dos¹². Por fim, a interface TDM é muito similar à I²S, com a diferença que esse protocolo permite que até dezesseis microfones compartilhem uma mesma linha de dados, tornando-o atrativo para aplicações multicanal. Para mais informações acerca dos protocolos de comunicação e sua implementação, recomenda-se a leitura dos documentos [21, 27, 28].

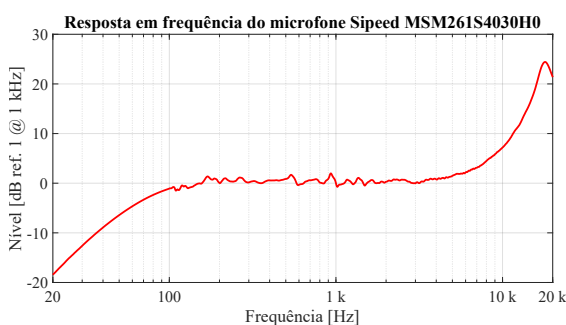
Para realizar a aquisição dos sinais de um microfone digital, faz-se necessário o uso de um microcontrolador (MCU, *microcontroller unit*) ou processador digital de sinais (DSP), responsável por gerar os sinais de *clock* e controle adequados, bem como fornecer energia.

3.3 Modelo utilizado neste trabalho

Para este tutorial, será utilizado o microfone MEMS digital [Sipeed MSM261S4030H0](#), já soldado em uma placa de prototipagem [29]. Este é um microfone I²S, com porta acústica superior, e foi escolhido devido à sua disponibilidade no mercado brasileiro. Na Figura 5 (a) encontra-se uma foto do microfone ao lado do microcontrolador Teensy 4.0 e uma moeda de 1 Real. A Figura 5 (b) contém sua resposta em frequência (medida pelos autores na Universidade Federal de Santa Maria).



(a) Comparação entre as dimensões do Teensy 4.0, microfone MEMS e moeda de 1 Real.



(b) Resposta em frequência medida do microfone MSM261S4030H0 (alisada em 1/8 de oitava).

Figura 5: Dimensões do microfone MEMS Sipeed MSM261S4030H0 e microcontrolador Teensy 4.0 e resposta em frequência do microfone.

4. MICROCONTROLADOR TEENSY 4.0

Para a aquisição dos sinais do microfone I²S será utilizado o microcontrolador Teensy 4.0 [30]. Este é um MCU compatível com Arduino, o que torna mais fácil sua programação.

Em relação às suas especificações técnicas, o Teensy 4.0 possui tamanho reduzido, baixo consumo de potência e vem equipado com o processador Arm Cortex M7, com velocidade de base de 600 MHz, tornando-o um dos microcontroladores mais rápidos e potentes do mercado. Ainda, possui suporte a matemática de ponto flutuante (32 e 64 bits) e à biblioteca CMSIS¹³, contendo várias ferramentas para processamento digital de sinais, como filtros e Transformada Rápida de Fourier (FFT). O Quadro 1 apresenta uma comparação das especificações técnicas de microcontroladores compatíveis com Arduino.

Considerando sua aplicação na área de acústica, o Teensy 4.0 conta com uma vasta biblioteca nativa (e de código aberto) para processamento de sinais de áudio, incluindo suporte ao protocolo I²S e transmissão de áudio via USB (atuando como uma interface de áudio genérica). A biblioteca é composta por objetos, responsáveis por realizar tarefas como aquisição, processamento e transmissão de sinais, e oferece uma interface gráfica para construção de rotinas, chamada [Audio System Design Tool](#), na qual os objetos são representados por blocos funcionais, com entradas e saídas, que podem ser interconectados (vide Figura 6). Outrossim, a interface contém uma breve documentação de cada objeto, descrevendo seus métodos e uso. Ao fim, é possível exportar a configuração em código para Arduino.

5. PREPARAÇÃO DO SISTEMA

Nesta Seção, inicia-se a etapa prática deste trabalho. A Seção 5.1 descreve os requisitos de *software* e a Seção 5.2 ilustra como configurar e programar o *hardware*.

5.1 Requisitos de *software*

Para realizar a aquisição dos sinais de microfones digitais via Teensy, primeiramente faz-se

¹²Em termos práticos, dois microfones podem ser ligados em um mesmo pino do microcontrolador ou processador.

¹³Ou *Common Microcontroller Software Interface Standard*, é um conjunto de ferramentas e funções para processadores da família Arm Cortex.

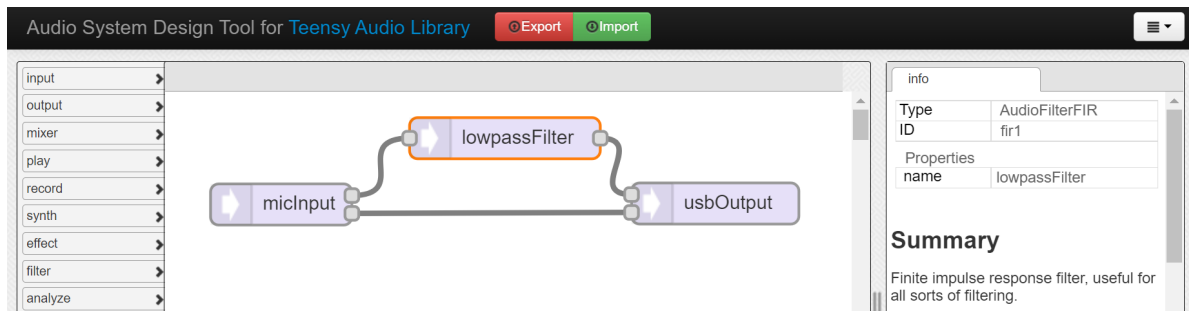


Figura 6: Interface gráfica para construção de rotinas de áudio para o Teensy 4.0. Neste exemplo, o microcontrolador captura o sinal de dois microfones I²S, processa o canal esquerdo com um filtro FIR e transmite o sinal estéreo de saída via USB para o computador. Note que o objeto `lowpassFilter` está selecionado e, à direita, há uma breve descrição de suas funcionalidades.

necessário instalar a plataforma Arduino IDE¹⁴ (utilizada para programar microcontroladores compatíveis), disponível gratuitamente no site <https://www.arduino.cc/en/software> (caso julgue necessário, siga este breve [tutorial](#)). Além disso, o repositório deste artigo no GitHub fornece informações adicionais, acesse em <https://github.com/eac-ufsm/fia2022-mic-mems>.

Na sequência, deve-se instalar o pacote Teensyduino, que habilita a programação de placas Teensy via Arduino IDE. Para isso, acesse o [site oficial](#)¹⁵, baixe o instalador compatível com seu sistema operacional e siga o passo-a-passo descrito para instalação.

Com essas ferramentas, já é possível programar o Teensy 4.0, via USB, utilizando a plataforma do Arduino.

5.2 Configuração do hardware

Aqui, mostrar-se-á como conectar dois microfones I²S ao Teensy e preparar o microcontrolador para enviar e receber sinais de áudio via USB para o computador. Como comentado nas Seções 3.2 e 4, microfones I²S possuem seis pinos e o Teensy 4.0 possui suporte nativo ao protocolo.

¹⁴IDE é a sigla para *Integrated Development Environment*, que diz respeito às plataformas que reúnem as ferramentas necessárias para implementação de *software*.

¹⁵O link é clicável e também pode ser acessado em: https://www.pjrc.com/teensy/td_download.html.

Desse modo, o primeiro passo será conectar corretamente os microfones ao MCU (o que pode ser feito com o auxílio de uma *protoboard* e cabos do tipo *jumper*). A Figura 7 contém um esquemático com as conexões.

Em suma, o Pino L/R (no microfone) define em qual dos canais o microfone enviará seus dados. Assim, quando aterrado (Pino G no Teensy), o sinal será enviado no canal esquerdo e, quando alimentado (Pino 3V no Teensy), será enviado no canal direito. Ademais, os Pinos 20 e 21 do MCU enviam os sinais de WS e *clock*, respectivamente, que são compartilhados por ambos microfones. Por fim, como o protocolo permite que dois canais sejam enviados em uma mesma linha de dados, a saída dos dois microfones (DA) são conectadas na mesma entrada (Pino 8 no Teensy).

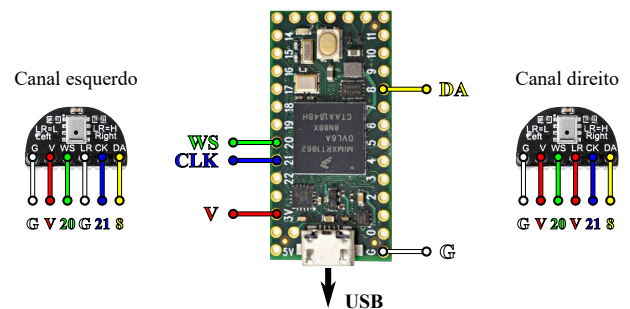
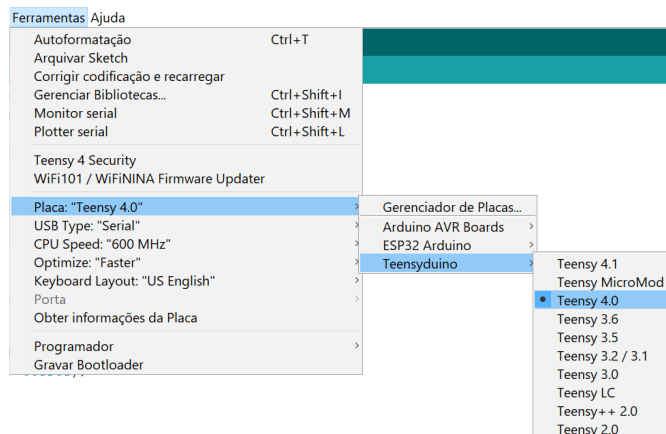


Figura 7: Diagrama de conexões entre dois microfones I²S e o microcontrolador Teensy 4.0. O pino L/R define em qual dos canais o microfone enviará seus dados.

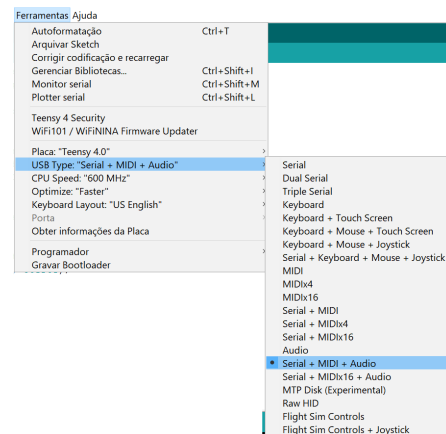
Quadro 1: Comparação de *hardware* entre diversas placas compatíveis com Arduino.

Nome	Processador	Velocidade da CPU	Armazenamento	Memória	Acesso direto à memória (DMA)
Teensy 4.1	NXP iMXRT1062 ^{2,◊}	600 MHz	8 MB (memória <i>flash</i>)	1 MB	32 canais
Teensy 4.0	NXP iMXRT1062 ^{1,◊}	600 MHz	2 MB (memória <i>flash</i>)	1 MB	32 canais
Teensy 3.6	NXP MK66FX1M0 ^{3,◊}	180 MHz	1 MB (memória <i>flash</i>)	256 KB	32 canais
Arduino Due	AT91SAM3X8E	84 MHz	512 KB (memória <i>flash</i>)	96 KB (SRAM)	23 canais
Arduino Mega 2560	ATmega2560	16 MHz	256 KB (memória <i>flash</i>)	8 KB (SRAM)	—

^{1,2} ARM Cortex-M7. ³ ARM Cortex-M4. [◊] FPU = Floating Point Unit. Mais informações podem ser acessadas em: [PJRC store](#), [Arduino.cc](#) e [Board db](#).



(a) Como selecionar o microcontrolador-alvo na Arduino IDE.



(b) Como configurar o Teensy para atuar como uma interface de áudio USB.

Figura 8: Configurações de *hardware* necessárias ao projeto (Arduino IDE).

Finalmente, antes de passar para a programação do microcontrolador de fato, faz-se necessário configurar a Arduino IDE para se comunicar com o Teensy, bem como qual tipo de conexão o Teensy realizará com o computador. Para isso, conecte-o ao computador utilizando um cabo USB e abra a Arduino IDE. Nesta, acesse a barra Ferramentas\Placas\Teensyduino e selecione Teensy 4.0 (vide Figura 8 (a)). Assim, ao compilar um programa, o Arduino IDE automaticamente o fará para a placa selecionada.

O próximo passo é configurar o Teensy para se comportar como uma interface de áudio USB. Com esse objetivo, acesse a barra Ferramentas\Serial Type e selecione Serial + MIDI + Audio (vide Figura 8 (b)). Deste modo, quando o programa for carregado no microcontrolador, o sistema operacional do computador automaticamente o reconhecerá como uma interface de áudio digital e assim será possível capturar os sinais dos microfones facilmente.

6. EXEMPLO PRÁTICO

Nesta Seção serão apresentados dois exemplos de aplicação:

- Captura do sinal de dois microfones e transmissão via USB para o computador (Seção 6.1); e
- Captura, processamento e transmissão via USB de um sinal gerado no computador e enviado ao Teensy — para avaliação da resposta em frequência de um filtro (Seção 6.2).

A geração, gravação e processamento dos sinais

de áudio será realizada via Matlab, utilizando o ITA-Toolbox [31]. Outrossim, para que o ITA-Toolbox e o Matlab consigam se comunicar com interfaces de áudio genéricas (como o Teensy), recomenda-se a instalação do *software* gratuito [ASIO4ALL](#). Finalmente, todos os códigos necessários aos exemplos encontram-se no [GitHub](#), para acesso público, acompanhados de breve documentação e comentários extras.

6.1 Captura e processamento de áudio com microfones MEMS

Para realizar a captura dos sinais de dois microfones MEMS, será implementado o programa ilustrado na Figura 6, com auxílio da ferramenta Audio System Design Tool, descrita na Seção 4. Na Figura 6, da esquerda para a direita, tem-se os seguintes blocos:

1. **micInput:** objeto `AudioInputI2S`, presente na aba `input` sob o nome `i2s`. Sua função é configurar o Teensy para gerar os sinais de *clock* e *WS*, bem como receber os sinais provenientes dos microfones. Note que há dois círculos, em seu lado direito, com cabos conectados a outros objetos. Esses círculos representam o número de saídas e, neste caso, o de cima é o canal esquerdo (numerado como 0), e o de baixo o canal direito (numerado como 1).
2. **lowpassFilter:** objeto `AudioFilterFIR`, presente na aba `filter` sob o nome `fir`. Com este objeto, é possível implementar filtros de resposta impulsiva finita (ou FIR) com ordem máxima de 200. Internamente, o fil-

tro realiza suas operações utilizando ponto-fixo e precisão de 16 bits (`int16`). Neste exemplo, implementou-se um filtro passa-baixas com frequência de corte de 4 kHz, cujos coeficientes foram gerados em Matlab (o código encontra-se no repositório deste artigo no [GitHub](#)). Note que o objeto possui um círculo à esquerda (ou seja, uma entrada) e um à direita (uma saída). Deste modo, só é capaz de filtrar um canal.

3. **usbOutput:** objeto `AudioOutputUSB`, presente na aba `output` sob o nome `usb`. Responsável por fazer o *streaming* de áudio do Teensy para o computador. Note que o objeto possui duas entradas, ou seja, o Teensy é capaz de atuar como uma interface de áudio USB estéreo (duas entradas e duas saídas).

Uma vez com os blocos devidamente ajustados e interconectados, basta clicar no botão “export” (em vermelho) para exportar a configuração em formato de código para Arduino. Cabe ressaltar que a interface gráfica apenas declara os objetos e suas interconexões, sendo que suas configurações ficam à cargo do usuário. O Código 1 contém o programa completo. Note que as conexões também são objetos, denominados `AudioConnection`, cujos argumentos são o objeto e canal de origem, seguidos do objeto e canal de destino, respectivamente. Finalmente, para

carregar o programa no Teensy, basta clicar no botão , presente no canto superior esquerdo da tela.

Para ilustrar o funcionamento do código, reproduziu-se uma varredura exponencial (ou *sweep*, de 20 Hz até 20 kHz) em um alto-falante localizado próximo aos microfones, montados lado a lado em uma *protoboard*. A varredura foi escolhida pois permite visualizar, no tempo, a atuação do filtro e a ressonância característica do microfone (veja o gráfico da Figura 5 (b)). O áudio capturado pelos microfones foi gravado via Matlab, por meio de funções do ITA-Toolbox, e os sinais no tempo podem ser vistos na Figura 9.

6.2 Avaliação da resposta em frequência de filtros implementados no Teensy

Uma vez que o Teensy é capaz de tanto receber, quanto enviar, sinais de áudio via USB, é possível tratá-lo como uma *caixa preta*¹⁶ e medir a resposta impulsiva de filtros implementados na plataforma, via o método da varredura exponencial¹⁷ (20 Hz–20 kHz). As-

¹⁶Ou *black-box*, do inglês.

¹⁷Para mais informações sobre a teoria de medição de respostas impulsivas, recomenda-se o artigo de Müller e Massarani [32]. Ademais, em uma publicação futura o procedimento será pormenorizado.

Código 1: Código para captura do sinal de dois microfones I²S, aplicação de um filtro passa-baixas em um dos canais e transmissão dos sinais via USB.

```
// Inclusão das bibliotecas necessárias
#include <Audio.h>
#include <Wire.h>
#include <SPI.h>
#include <SD.h>
#include <SerialFlash.h>

// Criação dos objetos de áudio
AudioInputI2S micInput;
AudioFilterFIR lowpassFilter;
AudioOutputUSB usbOutput;

// Definição das conexões entre os objetos
AudioConnection patchCord1(micInput, 0, lowpassFilter, 0);
AudioConnection patchCord2(micInObj, 1, usbOutput, 1);
AudioConnection patchCord3(lowpassFilter, 0, usbOutput, 0);

// Coeficientes para o filtro passa-baixas (FIR)
const short firTaps = 122;
const short coeffs[firTaps] = {2, 4, 4, 5, 2, -2, -7, -11, -13, -10, -2, 10, 21, 28, 26, 14, -6, -29, -48,
-54, -42, -11, 30, 69, 93, 89, 52, -11, -83, -139, -158, -124, -41, 73, 181, 247, 237, 144, -17, -201,
-348, -401, -323, -118, 168, 450, 631, 627, 400, -21, -538, -996, -1222, -1068, -454, 598, 1960, 3420,
4719, 5615, 5935, 5615, 4719, 3420, 1960, 598, -454, -1068, -1222, -996, -538, -21, 400, 627, 631, 450,
168, -118, -323, -401, -348, -201, -17, 144, 237, 247, 181, 73, -41, -124, -158, -139, -83, -11, 52, 89,
93, 69, 30, -11, -42, -54, -48, -29, -6, 14, 26, 28, 21, 10, -2, -10, 2, 0};

// Configuração do microcontrolador
void setup() {
  Serial.begin(9600);
  AudioMemory(120);
  lowpassFilter.begin(&coeffs, firTaps);
}

void loop() {
  // Inicia-se o streaming de áudio
}
```

sim, para avaliar o filtro passa-baixas utilizado no exemplo da Seção 6.1, construiu-se o programa ilustrado na Figura 10. Aqui é importante notar a presença do objeto `i2s_dummy`. De acordo com a documentação da biblioteca de áudio do Teensy, os objetos do tipo `AudioInputUSB` e `AudioOutputUSB` são incapazes de iniciar a transmissão de dados por si. Desse modo, recomenda-se adicionar um objeto *dummy*, que pode ser qualquer objeto do tipo `input` ou `output` (excluindo os USB), apenas para dar início à circulação de sinais de áudio.

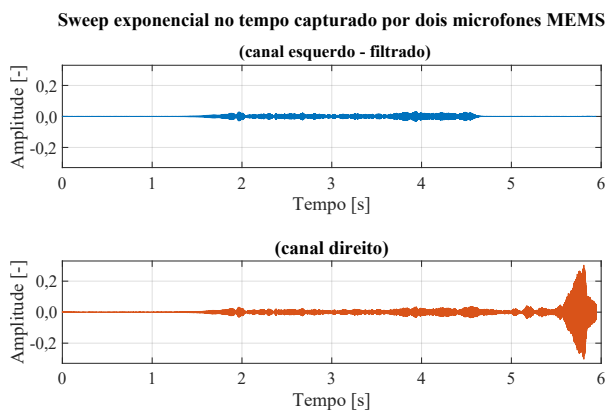


Figura 9: Varredura exponencial capturada no tempo por ambos microfones. Em azul, o sinal filtrado, e em laranja, o sinal sem processamento.

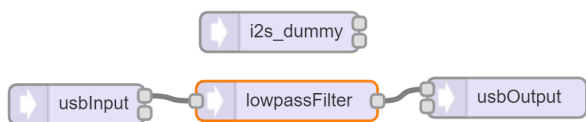


Figura 10: Objetos utilizados no Exemplo 2 e suas interconexões.

Para avaliar a resposta do filtro, escreveu-se um código em Matlab com auxílio de funções do ITA-Toolbox. O sinal de excitação foi uma varredura exponencial com 2^{18} amostras, taxa de amostragem de 44100 Hz e faixa de frequência de 20 Hz à 20 kHz. O sinal foi enviado via USB para o Teensy e recuperado, também via USB, pelo computador. Finalmente, a resposta impulsiva foi calculada dividindo o espectro do sinal gravado pelo espectro do sinal de excitação [32]. Ademais, o filtro implementado no Teensy foi comparado com o original, construído em Matlab. O resultado está ilustrado na Figura 11, na qual é possível verificar que ambas as respostas (na frequência) são muito similares, tanto em magnitude, quanto na fase (nesse gráfico, representada pelo *atraso de grupo* para uma melhor visualização¹⁸).

¹⁸O *atraso de grupo* é a derivada da fase em relação à

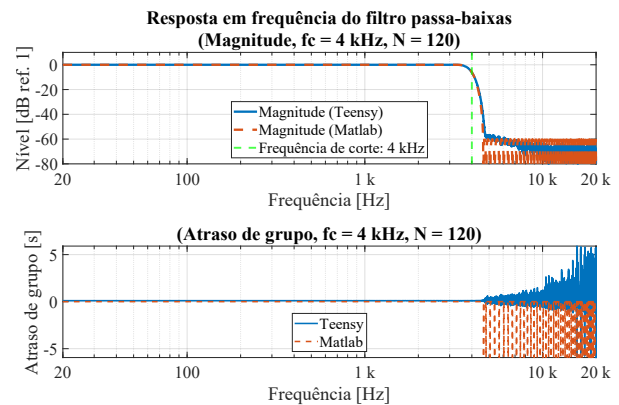


Figura 11: Comparação entre as respostas em frequência e atraso de grupo do filtro projetado no Matlab e implementado no Teensy.

7. CONSIDERAÇÕES FINAIS

Este artigo apresentou um breve resgate histórico da tecnologia dos microfones MEMS, bem como detalhes acerca de suas características construtivas e acústicas. Outrossim, abordou-se os protocolos de comunicação disponíveis nos microfones com saída digital e demonstrou-se como realizar, de forma simples, a aquisição do sinal de microfones MEMS digitais I²S utilizando o microcontrolador Teensy 4.0. Os componentes utilizados possuem um custo relativamente baixo, permitindo que pesquisadores e iniciantes no universo dos microfones MEMS e sistemas embarcados possam prontamente explorar suas possibilidades. Ainda, demonstrou-se como programar e avaliar filtros digitais implementados na plataforma.

Os conteúdos técnico e computacional apresentados podem ser utilizados por educadores em disciplinas de instrumentação, sinais e sistemas e processamento digital de sinais, como exemplos. A partir do apresentado, muitas outras aplicações podem ser desenvolvidas.

Finalmente, os exemplos práticos encontram-se em um repositório no [GitHub](https://github.com/eac-ufsm/fia2022-mic-mems), com todas as funções e códigos utilizados, bem como comentários complementares. Tudo está publicamente disponível para uso em

- <https://github.com/eac-ufsm/fia2022-mic-mems>,

pede-se apenas que ao utilizar os códigos atribua-se uma citação aos autores originais.

frequência e descreve a latência imposta pelo sistema, em segundos, em função da frequência. Filtros do tipo FIR possuem fase linear na banda de passagem e, portanto, um atraso de grupo constante.

8. AGRADECIMENTOS

Os autores agradecem ao curso de Engenharia Acústica (EAC), da Universidade Federal de Santa Maria (UFSM), bem como aos programas FIPE-CT e FIEEX-CT, por financiarem e apoiarem este projeto. Além disso, um agradecimento especial para o Engenheiro Acústico e egresso da EAC que apoiou esta pesquisa, Sergio Aguirre.

REFERÊNCIAS

- [1] Malcovati, P. e Baschiroto, A. The evolution of integrated interfaces for MEMS microphones. *Micromachines*, 9(7):1–20, n. 323, 2018. ISSN 2072-666X. doi: [10.3390/mi9070323](https://doi.org/10.3390/mi9070323).
- [2] Gfai Tech GmbH. Microphone Arrays for Different Applications. Online, 2022. Disponível em: <https://www.gfai.tech.com/products/acoustic-camera/microphone-arrays>. Acesso em junho de 2022.
- [3] CAE Software & Systems. Acoustic Camera – Microphone Arrays for Beamforming. Online, 2022. Disponível em: <https://www.cae-systems.de/en>. Acesso em junho de 2022.
- [4] Zylia. Zylia ZM-1 microphone. Online, 2022. Disponível em: <https://www.zylia.co/zylia-zm-1-microphone.html>. Acesso em junho de 2022.
- [5] Mello, Felipe Ramos; Fonseca, William D’Andrea e Mareze, Paulo Henrique. MEMS digital microphone and Arduino compatible microcontroller: an embedded system for noise monitoring. Em *50th International Congress and Exposition on Noise Control Engineering – Internoise 2021*, págs. 1–12, Washington, DC, USA, Aug. 2021. doi: [10.3397/IN-2021-2557](https://doi.org/10.3397/IN-2021-2557). Disponível em: <https://bit.ly/MEMS-int2021>.
- [6] Mello, Felipe Ramos; Fonseca, William D’Andrea e Mareze, Paulo Henrique. Autonomous noise monitoring system based on digital MEMS microphones: development of a smartphone application for remote communication. Em *51th International Congress and Exposition on Noise Control Engineering – Internoise 2022*, págs. 1–12, Glasgow, Scotland, UK, Aug. 2022.
- [7] Svantek. Noise Monitoring Station Class 1 SV 307A. Online, 2022. Disponível em: <https://svantek.com/products/sv-307a-noise-monitoring-station>. Acesso em junho de 2022.
- [8] Smith, Charles S. Piezoresistance Effect in Germanium and Silicon. *Physical Review*, 94(1):42–49, Apr 1954. ISSN 1536-6065. doi: [10.1103/physrev.94.42](https://doi.org/10.1103/physrev.94.42).
- [9] Paul, William e Pearson, G. L. Pressure Dependence of the Resistivity of Silicon. *Physical Review*, 98(6):1755–1757, Jun 1955. ISSN 1536-6065. doi: [10.1103/physrev.98.1755](https://doi.org/10.1103/physrev.98.1755).
- [10] Pfann, W. G. e Thurston, R. N. Semiconducting Stress Transducers Utilizing the Transverse and Shear Piezoresistance Effects. *Journal of Applied Physics*, 32(10):2008–2019, Oct 1961. ISSN 1089-7550. doi: [10.1063/1.1728280](https://doi.org/10.1063/1.1728280).
- [11] Tufte, O. N.; Chapman, P. W. e Long, Donald. Silicon Diffused-Element Piezoresistive Diaphragms. *Journal of Applied Physics*, 33(11):3322–3327, Nov 1962. ISSN 1089-7550. doi: [10.1063/1.1931164](https://doi.org/10.1063/1.1931164).
- [12] Bogue, Robert. MEMS sensors: past, present and future. *Sensor Review*, 27(1):7–13, Jan 2007. ISSN 0260-2288. doi: [10.1108/02602280710729068](https://doi.org/10.1108/02602280710729068).
- [13] Elko, Gary W. e Harney, Kieran P. A history of consumer microphones: the electret condenser microphone meets micro-electrical-mechanical-systems. *Acoustics Today*, 5(2):4–13, Abril 2009. ISSN 1557-0215. Disponível em: <https://acousticstoday.org/issues/2009AT/Apr2009/#p=1>.
- [14] Zawawi, Siti Aisyah; Hamzah, Azrul Azlan; Majlis, Burhanuddin Yeop e Mohd-Yasin, Faisal. A review of MEMS capacitive microphones. *Micromachines*, 11(5):484(1–26), May 2020. ISSN 2072-666X. doi: [10.3390/mi11050484](https://doi.org/10.3390/mi11050484).
- [15] Sessler, Gerhard M. Silicon Microphones. *Journal of the Audio Engineering Society (JAES)*, 44(1/2):16–22, Fevereiro 1996. ISSN 1549-4950. Disponível em: <https://www.aes.org/e-lib/browse.cfm?elib=7916>.
- [16] Scheeper, P.R.; Donk, A.G.H.; Olthuis, W. e Bergveld, P. A review of silicon microphones. *Sensors and Actuators A: Physical*, 44(1):1–11, Jul 1994. ISSN 0924-4247. doi: [10.1016/0924-4247\(94\)00790-x](https://doi.org/10.1016/0924-4247(94)00790-x).
- [17] Suvanto, Mikko. *The MEMS microphone book*. 1ª ed. Kangasala, Finland: MOSOMIC, 2022. ISBN 978-9529456604. Disponível em: <http://www.mosomic.com/thememsmicrophonebook>.
- [18] Wong, George S. K. e Embleton, Tony F. W. *AIP Handbook of Condenser Microphones — Theory, Calibration, and Measurements*. New York, NY, USA: AIP Press, 1995. ISBN 1-56396-284-5.
- [19] Silva, Lúcia Corrêa. *Caracterização de microfones MEMS analógicos*. Trabalho de conclusão de curso (monografia), Engenharia Acústica, Universidade Federal de Santa Maria, Santa Maria, RS, Brasil, julho 2018. doi: [10.13140/RG.2.2.25995.64808](https://doi.org/10.13140/RG.2.2.25995.64808).
- [20] Djurek, Ivan; Grubesa, Tomislav e Orlic, Niksa. Measurements of analog MEMS microphones. Em Ivšić, Branimir; Petrović, Goran e Dadić, Martin, editores, *2019 2nd International Colloquium on Smart Grid Metrology (SMAGRIMET)*, volume 2, págs. 1–4, Split, Croácia, apr 2019. University of Zagreb, Faculty of Electrical Engineering and Computing, IEEE. doi: [10.23919/smagrimet.2019.8720378](https://doi.org/10.23919/smagrimet.2019.8720378).
- [21] Lewis, Jerad. Analog and Digital MEMS Microphone Design Considerations, 2013. Analog Devices, Technical Article ([MS-2472](https://www.analog.com/en/resources/technical-articles/ms-2472)).
- [22] Knowles. Sisonic design guide, 2017. Application Note ([AN24](https://www.knowles.com/en-us/resources/application-notes/an24)).
- [23] Pflug, Patrick e Krischker, Detlef. Aspects of the use of MEMS microphones in phased array systems.

Em *46th International Congress and Exposition on Noise Control Engineering – Intersound 2017*, págs. 1–11, Hong Kong, China, Aug. 2017. Disponível em: <http://bit.ly/int2017-mems-array>.

- [24] Alsina-Pagès, Rosa Maria; Hervás, Marcos; Duboc, Leticia e Carbassa, Jordi. Design of a Low-Cost Configurable Acoustic Sensor for the Rapid Development of Sound Recognition Applications. *Electronics*, 9(7):1155(1–20), Jul 2020. ISSN 2079-9292. doi: [10.3390/electronics9071155](https://doi.org/10.3390/electronics9071155).
- [25] Fonseca, William D’Andrea; Mello, Felipe Ramos; Carvalho, Davi Rocha; Mareze, Paulo Henrique e Silva, Olavo M. Measurement of car cabin binaural impulse responses and auralization via convolution. Em *2021 Immersive and 3D Audio: from Architecture to Automotive (I3DA)*, págs. 1–13, Bologna, Italy, 2021. IEEE. doi: [10.1109/I3DA48870.2021.9610834](https://doi.org/10.1109/I3DA48870.2021.9610834).
- [26] AN4426 – Tutorial for MEMS microphones. Application Note 4426, STMicroelectronics, Fevereiro 2017. Disponível em: https://www.st.com/resource/en/application_note/dm00103199-tutorial-for-mems-microphones-stmicroelectronics.pdf.
- [27] Philips Semiconductors. I2S Bus Specification (Revised), 1996. [Application note](#).
- [28] Kite, Thomas. Understanding PDM Digital Audio. Application note, Audio Precision, Beaverton, Oregon, Fevereiro 2012. Disponível em: <https://www.ap.com/download/understanding-pdm-digital-audio-2/>.
- [29] MEMSensing Microsystems. **MSM261S4030H0** – I²S digital output MEMS mic. with multi modes (datasheet), v 1.4, 2015.
- [30] PJRC – P. J. Stoffregen. [Teensy 4.0 Development Board](#), [Teensy Audio Library](#), [Audio System Design Tool](#), & [Creating New Audio Objects](#), 2021. Disponível em: <https://www.pjrc.com>.
- [31] Dietrich, Pascal; Guski, Martin; Pollow, Martin; Müller-Trapet, Markus; Masiero, Bruno; Scharrer, Roman e Vorländer, Michael. ITA-Toolbox – An Open Source Matlab Toolbox for Acousticians. Em *Proceedings of the 38th German Annual Conference for Acoustics DAGA 2012 in Darmstadt*, número 38, págs. 151–152, 2012. Disponível em: https://pub.dega-akustik.de/DAGA_2012/data/articles/000164.pdf.
- [32] Müller, Swen e Massarani, Paulo. Transfer-Function Measurement with Sweeps – *Director’s cut including previously unreleased material and some corrections*. *Journal of the Audio Engineering Society (JAES)*, 49 (6):443–471, Jun 2001. ISSN 1549-4950.

A. INTERFACES ELÉTRICAS DE MICROFONES

Neste apêndice encontram-se exemplos de interfaces elétricas com microfones MEMS (analógicos e digitais), bem como capacitivo com polarização

permanente (eletreto) — são complementares à Figura 4.

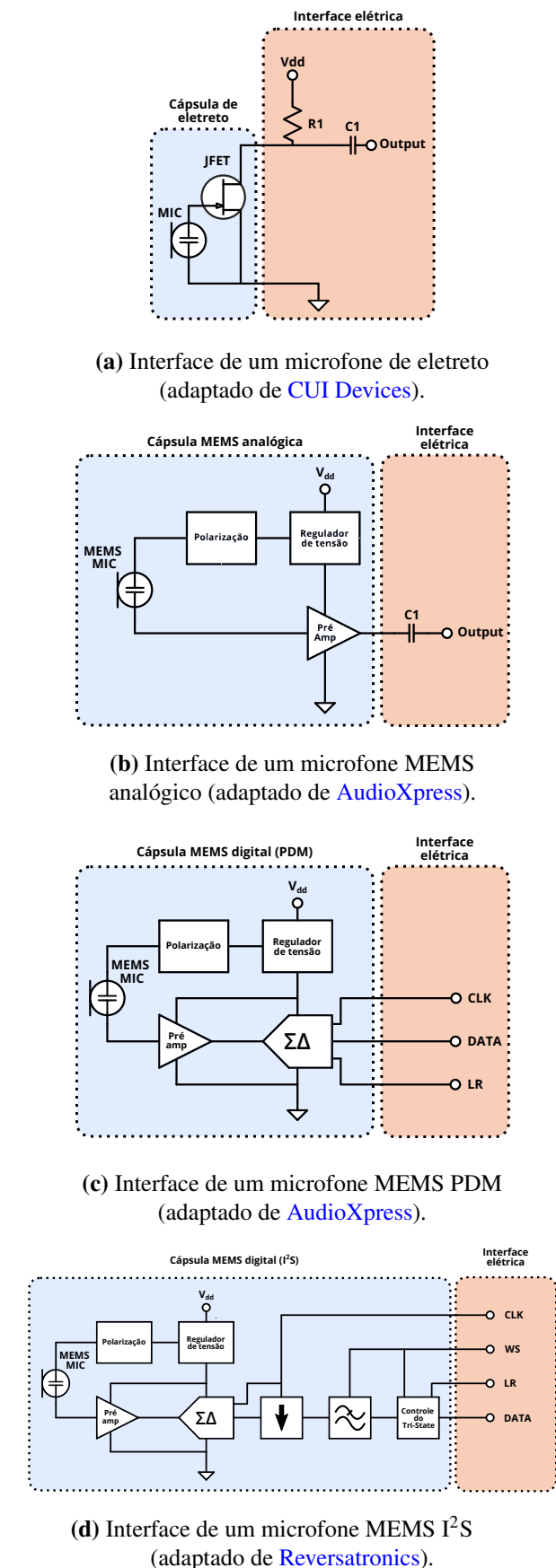


Figura 12: Interfaces elétricas para diversos tipos de microfones capacitivos.