Dispositivo rastreador de movimentos da cabeça baseado em Arduino: construção e utilização em acústica

Bom, E. B.; Fonseca, W. D'A.; Brandão, E.; Mareze, P. H.

Engenharia Acústica, Universidade Federal de Santa Maria, Santa Maria, RS
{enzo.bom, will.fonseca, eric.brandao, paulo.mareze}@eac.ufsm.br.

Resumo

Dispositivos de rastreamento são geralmente instrumentos que fornecem informação de rotação e/ou translação de uma determinada pessoa (ou objeto), podendo esse dado ser utilizado em diversas aplicações. Especificamente associados aos movimentos da cabeça, tais dispositivos são denominados *head-trackers*. Eles são extremamente úteis no contexto de acústica virtual (e auralização), em que o conhecimento da orientação do receptor é de grande importância para medições e processamento. Com base no instrumento desenvolvido no projeto Mr. Head Tracker, de concepção original na Universidade de Graz, na Áustria, cujo o intuito era criar um dispositivo de rastreamento de baixo custo, foi concebido e montado um protótipo funcional utilizando *hardware* e *software* na Universidade Federal de Santa Maria (UFSM). Nesta implementação, utilizando Arduino, foram projetadas uma cadeia de reprodução e uma cadeia de medição/gravação biauriculares. A verificação do funcionamento de tais cadeias foi realizada a partir de procedimentos experimentais, sendo eles testes subjetivos e medições acústicas. Com isso, o objetivo deste artigo é apresentar os passos dessa nova proposta de construção, constando alterações em relação ao projeto original e elaborando detalhes. São apresentados também os resultados de sua aplicação (assim como sua metodologia), compostos de três testes realizados com a cadeia de reprodução proposta, a fim de aclarar seu comportamento e suas limitações. Os resultados demonstram que o dispositivo funcionou de maneira adequada com os códigos computacionais desenvolvidos, apresentando acordo com a teoria e expectativas.

Palavras-chave: tecnologia biauricular, acústica virtual, auralização, rastreamento de movimentos.

PACS: 43.66.Pn, 43.66.Yw, 43.66.Qp, 43.28.Tc, 43.20.Ye.

Arduino-based Head-tracking device: assembly and application in acoustics

Abstract

Tracking devices can be generically defined as instruments that provide position information for a person or object, supplying rotation and/or translation data. Thus, they can be used in different applications in engineering. As they concern head movements specifically, these devices are called head-trackers and are extremely useful in the field of virtual acoustics (and auralization), where it is crucial to know a receiver's position (data used to post-process acoustical information). Based on the low-cost head-tracker developed from the project named Mr. Head Tracker at the University of Graz, Austria, a new hardware-software functional prototype was assembled at the Federal University of Santa Maria (UFSM), Brazil. Two different binaural application chains were designed: reproduction and recording/measurement. The good performance of the systems was assessed through experiments, including subjective tests and acoustical measurements. The aim of this paper is to describe the steps to assemble this new Arduino-based device, including additional instructions due to modifications from the original concept. The methodology and results of the three subjective tests conducted using the reproduction chain are also included and detailed. These tests clarify the behavior and limitations of the system constructed. In general, results have shown that the device works in accordance with the codes developed, matching theory and expectations.

Keywords: binaural technology, virtual acoustics, auralization, moviment tracking device.

DOI: 10.55753/aev.v33e50.82

1. INTRODUÇÃO

No contexto da tecnologia biauricular [1], tanto para reprodução quanto para captação de sinais biauriculares (que são aqueles que carregam os efeitos de difração do corpo do ser humano sobre uma onda sonora incidente) é extremamente importante ter a informação da localização do receptor (que pode ser uma pessoa ou um *manequim*) em termos de rotação e translação. Para isso, muitas vezes são utilizados dispositivos de rastreamento chamados de *head-trackers*. Esses instrumentos têm como finalidade fornecer dados sobre a posição e orientação da cabeça receptora em questão. Tais informações podem ser utilizadas, por exemplo, em aplicações para gerar movimentos em um ambiente virtual a partir de movimentos reais [2] ou fornecer informações de ângulo em medições de *Head Related Transfer Functions*[1] (HRTFs, Funções de Transferência Relativas à Cabeça ou ainda Funções de Transferência Anatômicas) [4]. Essas funções descrevem objetivamente a interação entre a fonte sonora e a antropometria do receptor, de forma que existe, para cada posição angular fonte-receptor no espaço 3D, uma função diferente.

Com a proposta de inicialmente ser utilizado como um instrumento MIDI[2] [5, 6] em uma *Digital Audio Workstation*[3], o projeto original nomeado Mr. Head Tracker[4] foi desenvolvido no *Institute for Electronic Music and Acoustics* (IEM) da Universidade de Graz, na Áustria [8]. Um dos objetivos foi implementar o dispositivo com base em placas Arduino [9] para criar um instrumento de custo reduzido.

Buscando cadeias de reprodução e gravação biauriculares, elaborou-se na Universidade Federal de Santa Maria (UFSM) um protótipo adaptado, mantendo ainda a ideia de dispositivo de baixo custo. Utilizou-se ainda de *software* adicionais para a comunicação com aplicações em Matlab e Pure Data (assim como seria necessário para funcionar adequadamente em DAWs).

[1]Sua versão no domínio do tempo é chamada de *Head Related Impulse Response* ou HRIR [3].
[2]Musical Instrument Digital Interface.
[3]DAW, por exemplo, Reaper, ProTools ou Ableton Live.
[4]O projeto também pode ser acessado em um site Git [7].

O foco principal deste artigo é então o procedimento de construção e implementação desse dispositivo, salientando as diferenças em relação ao projeto original. Apresenta-se também testes realizados para estimar a eficácia do aparato no rastreamento de posições diversas.

2. FUNDAMENTOS

Para melhor entendimento de algumas etapas e conceitos do trabalho, se faz necessário o desenvolvimento de uma breve fundamentação teórica, desenvolvida a seguir.

2.1 Eixos e coordenadas de movimento

No contexto de rastreamento de movimento, além dos graus de liberdade associados aos movimentos de translação nos eixos X, Y, Z, são também definidos movimentos de rotação em torno deles. Esses três movimentos são denominados *yaw, pitch* e *roll*[5], sendo eles respectivamente associados ao eixo Z, Y e X. Considerando o centro geométrico da cabeça de uma pessoa como a origem desses eixos, tem-se que Z é o vetor que aponta para cima, X para a frente da pessoa e Y para a esquerda. Logo, define-se *yaw* como o movimento de virar a cabeça para esquerda e/ou direita, *pitch* como o movimento de olhar para cima e/ou para baixo e *roll* de deitar a cabeça em direção aos ombros. Para facilitar o entendimento, observe a Figura 1 que demonstra a situação citada.

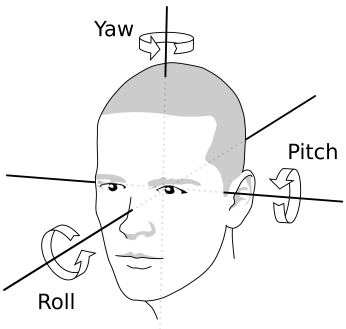


Figura 1: Cabeça centrada na origem do eixo de coordenadas X, Y e Z (movimentos associados *roll*, *pitch* e *yaw*). Retirado de [10].

[5]Termos muito utilizados em navegação, traduções incluem: guinada (*yaw*), inclinação ou arfagem (*pitch*) e rolamento ou giro (*roll*). Utilizar-se-á a nomenclatura em inglês por questões de compatibilidade de *software*.

Por convenção são positivos os valores de *yaw, pitch* e *roll*, respectivamente, quando vira-se a cabeça para esquerda, quando olha-se para baixo e quando deita-se à cabeça em direção ao ombro direito.

2.2 Utilização de MIDI 14 bits: MSB e LSB

Para aumentar a resolução angular do dispositivo de rastreamento, foi adotado o MIDI 14 bits. A criação de mensagens MIDI de 14 bits consiste basicamente da utilização de duas mensagens de 7 bits, obtendo-se $2^{14} = 16.384$ incrementos em vez dos tradicionais 128 incrementos do 7 bits. Utiliza-se o *Most Significant Byte* (MSB)[6] e o *Least Significant Byte* (LSB) que são encontrados a partir de algumas operações binárias [11]. O desenvolvimento desse cálculo é baseado no que consta no artigo do projeto original [8]. O entendimento desse procedimento se faz necessário pois para criar valores de ângulo a partir dos dados do dispositivo o conhecimento do caminho inverso é necessário.

Para converter os valores angulares fornecidos pelo sensor em duas mensagens de 7 bits é necessário primeiro multiplicar esse número pelo fator $2^{14}/360$, em que 2^{14} é quantidade de valores em 14 bits e 360 a amplitude angular desejada. Deve-se então arredondar esse valor para que ele seja inteiro. Em conseguinte, ele é convertido para notação binária para se encontrar o MSB e o LSB da seguinte forma:

$$\mathrm{MSB}_{7\mathrm{bit}} = (v_{14\mathrm{bit}} \gg 7) \;\&\; 0111\;1111_2 \quad (1)$$

e

$$\mathrm{LSB}_{7\mathrm{bit}} = v_{14\mathrm{bit}} \;\&\; 0111\;1111_2, \quad (2)$$

em que $v_{14\mathrm{bit}}$ representa o valor angular já no formato binário de 14 bits, o subíndice $\{\cdot\}_2$ indica que é um número binário, $\gg$ a operação binária de *bitwise right shift* (ou deslocamento de bits para direita) e & a operação binária AND[7].

[6] MSB define o byte mais significante e LSB byte menos significante, ou seja, o octeto de bits. Encontra-se também na literatura msb e lsb para *most* e *least significant bit*, respectivamente. Lembrando que bit significa *binary digit* (0 ou 1).

[7] As duas operações binárias realizadas nessa etapa são explicadas em detalhe no livro *Sistemas Digitais: princípios e aplicações* [12].

Basicamente, a operação & compara um mesmo bit de dois diferentes números, retornando 1 (um) quando ambos forem 1 (um). A operação *shift* serve para deslocar os bits de um número binário em um número desejado de casas (no caso deste projeto é 7). Utilizando o ângulo de 120°, faz-se o exemplo do procedimento para melhor entendimento. A saber:

- Escala-se o valor para 14 bits:
  $120 \cdot 2^{14}/360 \approx 5461,3333$;
- Arredonda-se esse valor encontrado e converte-se em número binário:
  $5461 \rightarrow 0001\;0101\;0101\;0101$;
- Para encontrar o LSB, faz-se a operação da Equação 2, que irá comparar os bits do número obtido com[8] (0000 0000) 0111 1111, retornando o valor 1 (um) quando ambos forem 1 (um). Isto é:

  $0000\;\;0000\;\;0111\;\;1111 \Rightarrow$ Ref.

  $0001\;\;0101\;\;0101\;\;0101 \Rightarrow 5461$

  $0000\;\;0000\;\;0101\;\;0101 \Rightarrow$ Result. LSB
- Para encontrar o MSB, faz-se a operação de *shift* no valor inicial[9],

  $0001\;\;0101\;\;0101\;\;0101 \Rightarrow 5461$

  $\underbrace{0000\;\;000}_{\text{bits inseridos}}\underbrace{0\;\;0010\;\;1010}_{\text{orig. deslocados}}\;\;\underbrace{1010\;\;101}_{\text{descartados}}$

  Realiza-se então novamente a comparação:

  $0000\;\;0000\;\;0111\;\;1111 \Rightarrow$ Ref.

  $0000\;\;0000\;\;0010\;\;1010 \Rightarrow 5461$ com *shift*

  $0000\;\;0000\;\;0010\;\;1010 \Rightarrow$ Result. MSB

A partir desse cálculo são obtidos, então, em valores decimais: LSB = 85 e MSB = 42. É interessante salientar que o número de referência 0111 1111 da operação AND é o valor 127, ou seja, o valor máximo para 7 bits, fazendo então com que os valores de MSB e LSB estejam compreendidos nesse intervalo, pois tudo a partir do oitavo bit irá resultar 0 (zero) depois da operação binária.

[8] Nesse caso, já que um número tem 16 bits e o outro apenas 8 bits, considera-se que à esquerda do número com 8 bits todas as posições são 0 (zero).

[9] Desloca-se o número para direita e à esquerda é completado com zeros.

Para encontrar (ou recuperar) os valores angulares, basta fazer o procedimento inverso. Devido ao uso do sistema binário, a operação de *shift* para direita resulta na divisão por 2 (dois) [12], logo, fazendo-se 7 (sete) deslocamentos, tem-se a divisão do valor por 128. Assim, os cálculos necessário são:

- $128 \cdot 42 + 85 = 5461$;
- Revertendo o fator de escala: $(5461 \cdot 360)/2^{14} \approx 119{,}992675°$.
- Dif.: $120 - 119{,}992675 \approx 0{,}007325°$.

No entanto, adicionalmente, no projeto original a intenção era ter os valores centrados em 0 (zero), ou seja, no intervalo $(-180, 180]$[10] e, por isso, o cálculo de mensagens MIDI para valores angulares ganha uma etapa adicional. Com isso,

$$\theta = \left[\left(\frac{128 \cdot \text{MSB} + \text{LSB}}{2^{14} - 1}\right) - 0{,}5\right] \cdot 360\,, \quad (3)$$

sendo que θ é um ângulo qualquer (as outras variáveis já foram definidas). Nesse caso, a subtração de 0,5 corresponde justamente ao processo de centrar os valores em torno de 0 (zero).

A escolha da escolha de 14 bits em vez de 7 bits se deve a resolução obtida. Considerado-se 360 graus, 7 bits resultaria em $360/2^7 \approx 2{,}81°$, excedendo *Limiar diferencial observável* (ou *just-noticeble difference*, JND) para discernimento de fontes no espaço, que é aproximadamente $0{,}97°$ para o plano horizontal e $3{,}65°$ para o plano vertical (como estudado por Perrott e Saberi [13]). Logo, utilizando-se de 14 bits tem-se $360/2^{14} \approx 0{,}02°$, um valor de resolução suficientemente maior.

3. METODOLOGIA

Nesta seção serão apresentados detalhes sobre a montagem do *head-tracker*, do projeto original para a versão da UFSM. Ademais, inclui a descrição do *hardware* e de alguns testes subjetivos realizados com ele.

[10] Notação de intervalo aberto e fechado, o símbolo (demonstra que -180 não está incluso, enquanto] indica que 180 está incluso.

3.1 Proposta da cadeia de reprodução e da cadeia de medição

Este projeto teve como objetivo final desenvolver duas cadeias biauriculares:

- reprodução e
- medição/gravação.

A ideia do primeiro caso é ter um sistema em que a pessoa, utilizando um *headphone* com o *head-tracker* fixado nele, fizesse movimentos e, a partir da utilização de um banco de dados de HRTFs e dos dados fornecidos pela dispositivo de rastreamento, se criassem áudios biauriculares[11] correspondentes ao movimento, sendo ainda apresentados simultaneamente para o sujeito. O segundo caso consistiu em desenvolver uma cadeia de medição de HRTFs utilizando um *manequim* (como o HATS[12] da Brüel & Kjær [14], por exemplo) em que o *head-tracker* atua como aparato determinante dos ângulos de medição. As duas cadeias propostas podem ser visualizadas, respectivamente, nas Figuras 2 e 3.

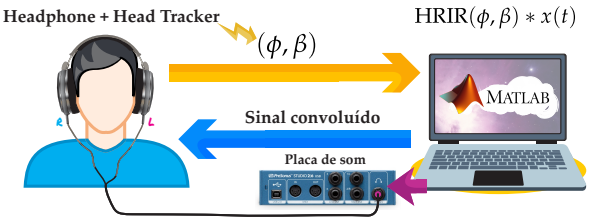


Figura 2: Representação esquemática da cadeia de reprodução implementada.

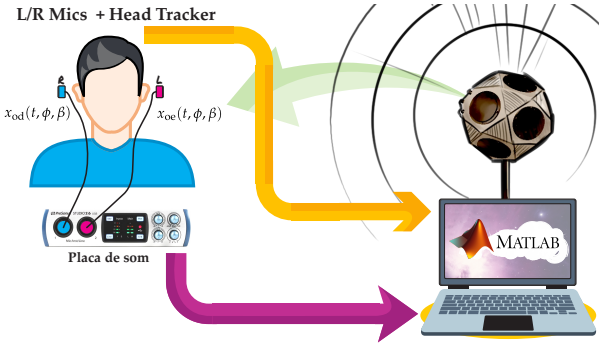


Figura 3: Representação esquemática da cadeia de medição implementada.

[11] Códigos desenvolvidos em Matlab.

[12] *Head and Torso Simulator* ou Simulador de cabeça e tronco. Também encontrado como *cabeça artificial*.

3.2 Arduino e descrição do projeto original

Como supracitado, o objetivo do projeto desenvolvido na Universidade de Graz era criar um instrumento de rastreamento MIDI e, para isso, foram utilizados códigos baseados no projeto HIDUINO[13] [15, 16]. Essa implementação fornece exemplos de códigos, documentação e *firmware*[14] para criar um instrumento MIDI *plug-and-play*[15] a partir de um MCU ATmega16u2 ou ATmega8u2 (*Microcontroller Unit*, citado nesse trabalho também apenas como *chip*). Esse tipo de *chip* é usualmente utilizado para controle da porta USB nativa de placas Arduino, porém, geralmente, os *chips* responsáveis pelo processamento do código dentro do Arduino são do tipo ATmega328 ou similares, veja a Tabela 1.

Para a utilização de um *firmware* de HIDUINO é necessário ter um sistema dois *chips*, sendo um deles responsável para o processamento dos dados e outro responsável pela conexão com a porta USB+MIDI. Este por sua vez será reprogramado, sendo importante ressaltar que não é possível utilizar um *firmware* de HIDUINO com sistemas que possuam apenas uma MCU. As três peças utilizadas no projeto original podem ser visualizadas na Figura 4. Um alternativa seria o uso de outra biblioteca como a USBMIDI Library [17].

O sensor Adafruit BNO055 [18] contém um giroscópio triaxial de 16 bits (com alcance de $\pm 2000^{\circ}$ por segundo), um acelerômetro triaxial de 14 bits, um sensor geomagnético e um chip MCU Cortex M0+ (de 32 bits), como pode ser observado na Figura 5. A taxa de atualização é de 100 Hz, podendo ser entregue em diferentes formatos, escolhidos a partir de linhas de código. Observe na Figura 6 a comparação entre os tamanhos do Arduino UNO (provavelmente o mais popular da série) e o Arduino Pro Mini.

[13] O HIDUINO refere-se ao nome *Human Interface Device* (HID). Ele oferece uma porta USB-MIDI real e não uma conversão serial como os *software* LoopBe1 no Windows e/ou IAC no MacOS. Está disponível em github.com/ddiakopoulos/hiduino.

[14] *Firmware* é uma classe específica de *software* que fornece controle de baixo nível para o *hardware* específico, isto é, um *software* que é embarcado do *hardware*.

[15] Dispositivos que dispensam a utilização de *drivers* adicionais para funcionar.

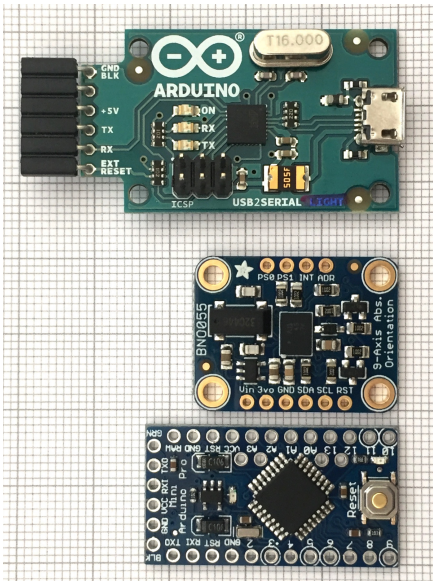


Figura 4: Peças utilizadas no projeto original, de cima para baixo: Arduino USB2 Serial Micro (24×42 mm), sensor Adafruit BNO055 (20×27 mm) e Arduino Pro Mini (18×33 mm), retirado de [8].

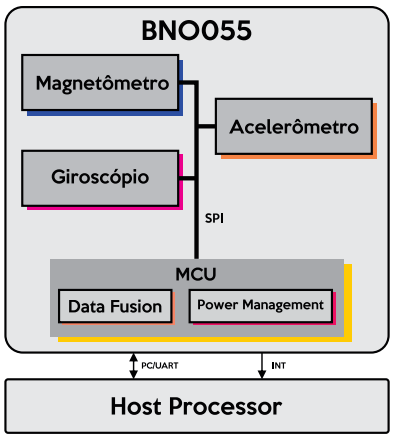


Figura 5: Arquitetura simplificada do elemento sensor BNO055 (adaptado de [18]).

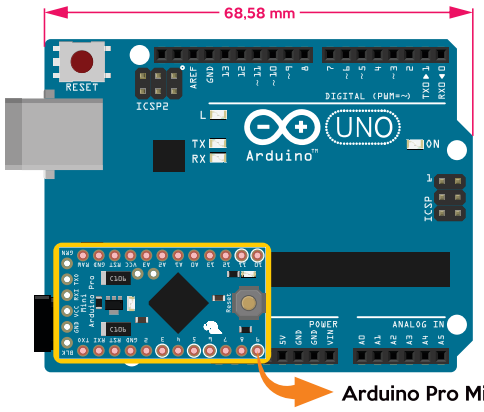


Figura 6: Comparação entre Aruino UNO e Arduino Pro Mini (dados técnicos comparativos na Tabela 1).

A MCU é conectada ao sensor via I^2C e realiza o processamento dos dados a partir do código desenvolvido. Dessa maneira, fornece mensagens MIDI CC (*Control Change* ou ainda conhecida como *Continuous Control*) para o computador, geradas com a utilização da biblioteca MIDI.

O Arduino Pro Mini (ATmega328) tem o objetivo de ser pequeno e compacto, logo, essa versão não possui portas USB nativas, tornando necessário então conectá-lo a um conversor USB. Nesse caso, o Arduino USB2Serial Micro, que possui nativamente uma porta USB e um MCU ATmega16u2, pode ser usado para fazer a comunicação entre o *chip* principal e o computador. É então possível realizar o procedimento de reprogramá-lo com um *firmware* de HIDUINO para ser reconhecido pelo computador como um dispositivo MIDI *plug-and-play*. O Arduino Nano (similar ao Pro Mini), embora tenha uma porta de comunicação USB nativamente, ainda necessitaria do módulo USB2Serial para que fosse *plug-and-play*. Além disso, o tamanho físico do Nano é maior do que o Pro Mini.

A vantagem de ter o aparato funcionando como um dispositivo MIDI é porque dessa maneira tem-se compatibilidade com várias DAWs. Isso possibilita a utilização dele com *plug-ins* de áudio, como por exemplo, de Ambisonics, AmbiX [21–23] ou o SPARTA [24, 25], que permitem criar, editar e reproduzir faixas de áudio tridimensionais. Ademais, sabendo como as mensagens MIDI foram criadas, é possível convertê-las em valores angulares (em graus), de forma que elas possam ser utilizadas em códigos computacionais desenvolvidos em Matlab [26] (e/ou outros *software*/linguagens).

Cabe aqui salientar que a família de placas Arduino e Arduino *compatible* é grande, tendo aplicações em áudio, acústica e vibrações com aquisição de sinais analógicos (com as placas Uno, Due [27] e Teensy 3.6 [28], por exemplo) e digitais (com as placas Zero, MKR1000 [29] e Teensy 4.0 [20], por exemplo), veja a Tabela 1. Os sites Adafruit [30] e Sparkfun [31] são ótimos locais para os leitores conhecerem modelos, projetos, tutoriais, circuitos e aplicações com essas placas (assim como sensores e novidades).

Como pode ser observado na Figura 7, o dispositivo conta também com uma *interface de usuário* equipada com um botão do tipo *push button* (em azul, que faz parte do processo de calibração) e de um botão DIP *switch* de 2 bits (que é responsável por escolher alguns modos de operação do dispositivo). O procedimento de calibração consiste, basicamente nos seguintes passos:

- Olhar para a posição frontal desejada e pressionar o botão *push button* por mais de um segundo;
- Olhar para baixo e (sem ficar/estar pressionando), apertar o botão com um clique simples.

Tabela 1: Especificações de diferentes modelos de placas Arduino ou Arduino *compatible* (adaptado de [19, 20]).

Name	Processor	Op./Input Volt.	CPU Speed	Analog In/Out	Dig. IO/PWM	EEPROM [kB]	SRAM [kB]	Flash [kB]	USB	UART
Micro	ATmega32U4	5 V / 7-12 V	16 MHz	12/0	20/7	1	2,5	32	Micro	1
MKR1000	SAMD21 Cortex-M0+	3,3 V/ 5V	48MHz	7/1	8/4	-	32	256	Micro	1
Pro	ATmega168 ATmega328P	3,3 V / 3,35-12 V 5 V / 5-12 V	8 MHz 16 MHz	6/0	14/6	0,512 1	1 2	16 32	-	1
Pro Mini	ATmega328P	3,3 V / 3,35-12 V 5 V / 5-12 V	8 MHz 16 MHz	6/0	6/0	1	2	32	-	1
Zero	ATSAMD21G18	3,3 V / 7-12 V	48 MHz	6/1	14/10	-	32	256	2 Micro	2
Uno	ATmega328P	5 V / 7-12 V	16 MHz	6/0	14/6	1	2	32	Regular	1
Due	ATSAM3X8E	3,3 V / 7-12 V	84 MHz	12/2	54/12	-	96	512	2 Micro	4
Mini	ATmega328P	5 V / 7-9 V	16 MHz	8/0	14/6	1	2	32	-	-
Nano	ATmega168 ATmega328P	5 V / 7-9 V	16 MHz	8/0	14/6	0,512 1	1 2	16 32	Mini	1
MKRZero	SAMD21 Cortex-M0+ 32 bit[1]	3,3 V	48 MHz	7 (ADC 8/10/12 bit)/1 (DAC 10 bit)	22/12	No	32 KB	256	1	1
Teensy 3.6	NXP MK66FX1M0[2]	3,3 V	180 MHz	25 (2x 13 bit)/ 2 (12 bit)	62/22	4	256	1024	Micro	6
Teensy 4.0	NXP iMXRT1062[3]	3,3 V	600 MHz	14 (2 x 12 bit)/0	41/31	64	1024	2048	Micro	5

[1] Low-power ARM MCU; [2]ARM Cortex-M4 with FPU$^\diamond$; [3]ARM Cortex-M7 with FPU; $^\diamond$FPU = Floating Point Unit

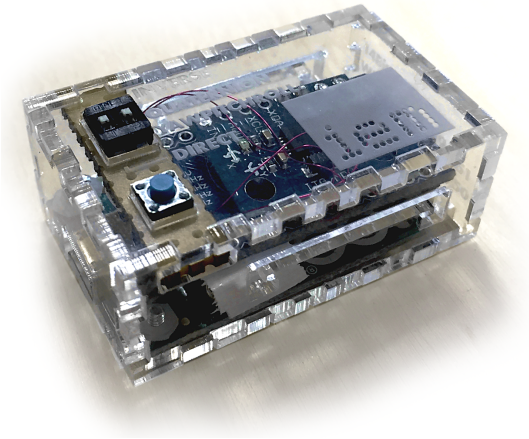


Figura 7: Protótipo da Universidade de Graz do MrHeadTracker (adaptado de [7]).

Essa calibração fica salva na memória do Arduino e a posição frontal pode ser reiniciada (*reset*) apertando (sem ficar pressionando) o botão novamente. É interessante fazer esse processo completo periodicamente – ou então toda vez que se usa o dispositivo.

Os modos de operação citados correspondem às posições das duas chaves que a peça *switch* possui. A Chave 1 é responsável por alternar o funcionamento do *head-tracker* entre modo direto (OFF) e indireto (ON) (`invSwitch`, dentro do código), sendo esses modos responsáveis por fazer o dispositivo funcionar como um *pointing device* ou para compensar a rotação do movimento realizado, respectivamente. Isso significa que, quando há um movimento, para o primeiro caso, dentro do computador se terá um movimento correspondente no mesmo sentido e, no segundo caso, se terá um movimento correspondente no sentido contrário. No modo direto, mexer a cabeça 30° para esquerda (sentido com valores positivos) gerará um valor de 30°, no entanto, no indireto, esse mesmo movimento irá gerar um valor de -30°. A Chave 2 é responsável por alternar a informação enviada pelo *head-tracker* entre *quaternions* (ON) e ângulos de Euler (OFF) (`quatSwitch`, dentro do código), sendo o primeiro um dado mais robusto, utilizado para processamento de localização quando o dispositivo pode estar em qualquer lugar no espaço e, o segundo, por exemplo, para a utilização do dispositivo em um lugar fixo em que a pessoa mexe apenas a cabeça.

3.3 Protótipo UFSM desenvolvido

A montagem do novo protótipo do dispositivo consistiu basicamente da utilização dos mesmos equipamentos, porém, com a dificuldade de encontrar disponível uma peça do projeto original, a interface USB utilizada foi um conversor FTDI FT232RL [32], que tem a pinagem semelhante ao do Arduino USB2 Serial Micro e também conecta perfeitamente com o Arduino Pro Mini, sendo as três peças utilizadas apresentadas na Figura 8.

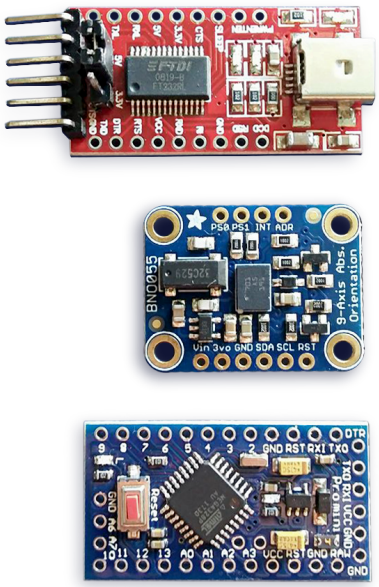


Figura 8: *Hardware* utilizados para a montagem do experimento, de cima para baixo: FT232RL, sensor BNO055 e Arduino Pro Mini.

Embora somente com um *hardware* de alteração do projeto original, foi encontrado o problema quanto à utilização de *firmware* de HIDUINO com apenas um *chip* para processamento, fato que, *a priori*, não era conhecido. Diferente do conversor Arduino que possui uma MCU do tipo ATmega16u2, possibilitando modificações, a placa FT232RL possui apenas um *chip* simples FTDI que serve somente como uma ponte para ligar uma placa que não possui porta USB ao computador, sem possibilidade de alteração do seu *firmware*. Ou seja, inviabiliza o funcionamento do aparelho como um dispositivo *plug-and-play*, assim, tornando-se necessária a utilização de *software* adicionais que recebam e traduzam as informações chegando pela porta

USB (previamente criadas com funções da biblioteca MIDI). Tendo em vista essa limitação, foi utilizado o Hairless MIDI [33], *software* cuja funcionalidade é organizar os dados da porta USB em mensagens MIDI. Para transmiti-las para outras aplicações, como por exemplo Matlab e Pure Data (Pd) [34], se faz necessária a utilização ainda de um outro *software* que funcione como um *cabo MIDI virtual*, ou seja, que faça o computador interpretar o sistema como se de fato um instrumento MIDI estivesse conectado na porta USB e estivesse enviando suas próprias mensagens. Para esse fim, foi utilizado o LoopBe30 [35], cujas portas virtuais de entrada e saída são reconhecidas nos *software* de recepção, para que possam ser utilizadas como seria com qualquer outro dispositivo MIDI.

3.4 Montagem e utilização

Visto que conversor FTDI possui seis pinos machos já soldados, foram soldados no Arduino Pro Mini conectores fêmea para que houvesse uma conexão estável e sem maus contatos. As peças conectadas e sendo alimentadas com tensão da porta USB do computador via cabo USB mini podem ser observadas na Figura 9.

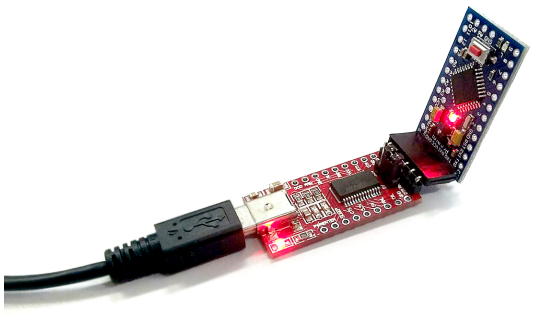


Figura 9: Arduino Pro Mini conectado ao conversor FT232RL da FTDI.

Para utilização do código disponibilizado no *website* do projeto foi necessário fazer o *download* do *software* de Arduino, além de algumas bibliotecas, sendo elas: MIDI, EEPROM, Adafruit Unified Sensor Library e também a Adafruit BNO055 Library. Após a devida instalação da IDE do Arduino, instalaram-se tais bibliotecas. Essa etapa pode ser realizada incluindo elas na pasta *libraries* dentro da pasta instalada Arduino, ou então, dentro da IDE[16], acessando o menu Sketch e então Incluir biblioteca → Adicionar biblioteca .zip e escolher o arquivo `.zip` baixado.

Para que fosse possível utilizar o dispositivo com o *software* Hairless MIDI, foi necessário fazer uma breve alteração no código original, em que era inicializada apenas uma conexão MIDI com o comando `MIDI.begin`, sendo na nova versão necessário inicializar também uma conexão serial com o comando `Serial.begin`, devendo, ao fim, existir a seguinte sequência do Código 1.

Código 1: Adição de comando[18] para inicializar conexão serial após a inicialização da conexão MIDI.

```
MIDI.begin(MIDI_CHANNEL_OMNI);
Serial.begin(115200);
```

A diferença é que, como as mensagens são enviadas via comunicação serial e devem ser traduzidas dentro do Hairless MIDI, essa conexão teve que ser previamente estabelecida. No caso do projeto original, como o dispositivo se comporta igual a um dispositivo MIDI por conta do *firmware* de HIDUINO, tal conexão não necessita ser feita visto que as mensagens enviadas, mesmo sendo elas enviadas serialmente, já estão organizadas de acordo com o protocolo MIDI.

Depois disso, para transferir o *sketch*[17] fornecido[18] na página *online* do projeto (com a adaptação supracitada já realizada), foi selecionada (no *software* do Arduino, menu Ferramentas) a placa "Arduino Pro ou Pro Mini", o processador nesse caso é o ATmega328P (5 V, 16 MHz). A porta serial é reconhecida quando o conversor FT232RL (junto ao Arduino) é conectado ao computador, sendo nesse caso a porta COM3 (lembrando que é necessário clicar no botão de carregar). Essas configurações podem ser visualizadas na Figura 10.

Após ter sido realizada essa etapa, foi feita a conexão das peças de acordo com o diagrama apresentado na Figura 11, sendo elas devidamente soldadas.

[16] *Integrated Development Environment*.
[17] Termo utilizado também para se referir à um código (ou projeto) de Arduino.
[18] `MrHeadTrackerBNO055Switchable.ino`.

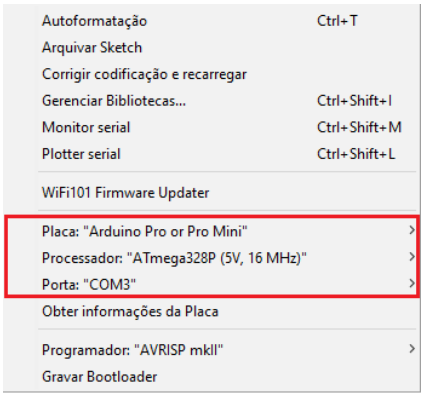


Figura 10: Configurações no *software* do IDE para transferência do *sketch* para o Arduino Pro Mini.

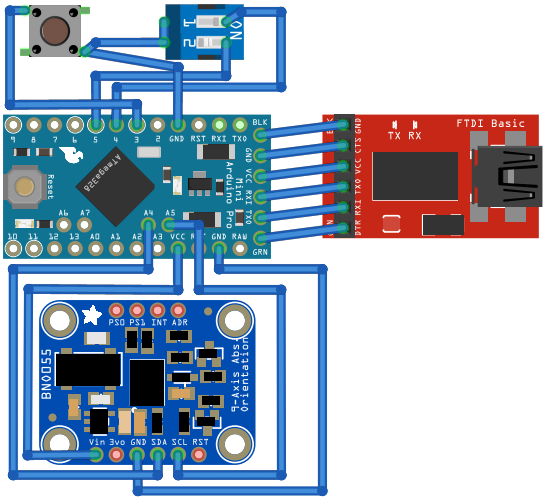


Figura 11: Diagrama de conexão dos *hardware* do *head-tracker*.

É extremamente importante comentar que o dispositivo só começa a enviar mensagens após uma calibração ter sido efetuada. Para esta etapa é interessante já ter o *software* Hairless MIDI aberto (tela de *debug*), de forma que é possível verificar se o aparelho começa, de fato, a enviar as mensagens. Após aberto, é possível perceber que o Hairless MIDI não possui muitas configurações, como mostrado na Figura 12. Entretanto, é essencial deixar ativada a caixa Serial <-> MIDI Bridge ON. É de suma importância ajustar o Baudrate[19] (em File → Preferences...) correspondentemente ao utilizado na rotina do Arduino. Configura-se também no menu Serial Port a porta COM associada ao dispositivo (assim como na IDE do Arduino).

[19]Baudrate é o número de vezes em um segundo que um sinal em um canal de comunicação muda.

A tela de *debug* também serve para verificar se o dispositivo esta enviando dados no formato de Euler ou em Quaternions. No primeiro caso, aparecem mensagens apenas dos controladores 16, 48, 17, 49, 18 e 50. No segundo caso, aparecem também dos controladores 19 e 51. Em MIDI Out, deve-se configurar uma das portas do dispositivo MIDI virtual, sendo neste caso do LoopBe30.

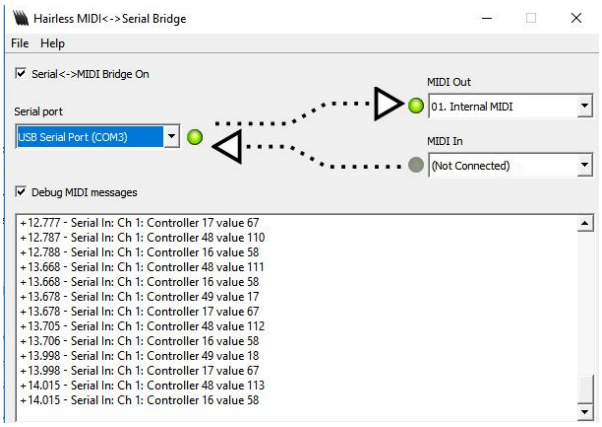


Figura 12: Interface do *software* Hairless MIDI.

É importante salientar que o LoopBe30 possui um detector de *feedback* MIDI, ou seja, ele reconhece quando uma mensagem MIDI fica "presa" entre a entrada e saída do dispositivo virtual (semelhante ao efeito de microfonia), assim, usa da função *mute* nas portas MIDI para que o computador não fique lento. Com o sensor BNO055, essa situação acontece pois a taxa de *uptade* dele é de 100 Hz, provavelmente ultrapassando o limite de memória que as especificações MIDI aceitam por segundo, acionando assim o mecanismo de defesa mesmo que não esteja acontecendo um *feedback* real. Dessa forma, foi necessário desativar esse detector, porém certificando-se que o fenômeno não estava de fato acontecendo. A versão comercial LoopBe30 teve de ser utilizada neste projeto, visto que a versão gratuita, o LoopBe1, não possui essa função de desabilitar o detector de *feedback* [35], sendo inviável na utilização com o sensor BNO055.

Para utilizar o dispositivo com o Matlab, existem algumas funções específicas do *software* para comunicação com instrumentos MIDI, sendo elas:

- `mididevinfo`: mostra todos os instrumentos MIDI e portas (*in/out*) disponíveis;
- `midideviceo`: permite selecionar e atribuir à uma variável o dispositivo MIDI disponível desejado; e
- `midireceive`: permite receber as mensagens MIDI do dispositivo selecionado.

Munido dessas funções foi possível fazer a leitura das mensagens enviadas pelo *head-tracker* e então gerar valores de ângulo a partir dos valores MIDI recebidos utilizando a Equação 3.

A partir dos diagramas apresentados nas Figuras 13 e 14 é possível observar a diferença entre as duas cadeias de utilização do *head-tracker*, isto é, projeto do IEM e o projeto da UFSM (demais pormenores podem ser verificados no trabalho de Bom [36]).

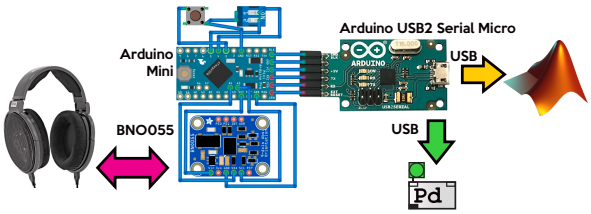


Figura 13: Diagrama da utilização do *head-tracker* com sistema original IEM.

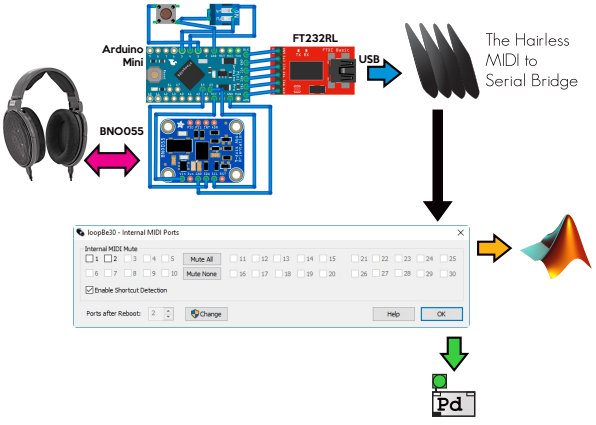


Figura 14: Diagrama da utilização do *head-tracker* com sistema da UFSM.

3.5 Testes subjetivos utilizando o dispositivo

Para verificar o funcionamento do dispositivo em conjunto com os códigos computacionais desenvolvidos, foram realizados alguns testes subjetivos[20] com uma versão simplificada da cadeia de reprodução proposta, assim como um teste em um formato mais similar ao proposto, de forma a compreender alguns detalhes do sistema.

Todos foram testes de localização de fontes virtuais geradas nas rotinas desenvolvidas em Matlab utilizando bancos de dados de HRTFs [37]. A ideia geral do primeiro teste, denominado aqui como Teste 1, foi apresentar ao sujeito áudios espaciais referentes à oito posições de fonte, as quais eram indicadas em uma folha de teste e eram as únicas opções possíveis de marcação do participante que, por sua vez, deveria fazê-la de acordo com a sua percepção para cada reprodução do áudio, sendo a ilustração utilizada uma versão impressa da imagem apresentada na Figura 15.

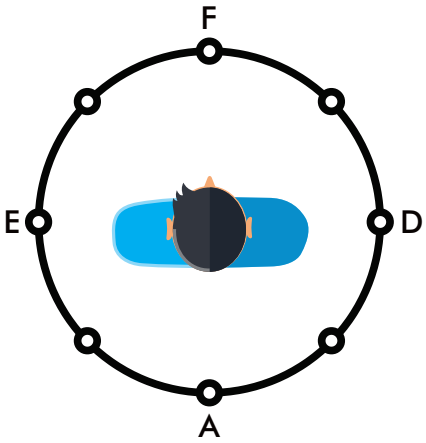


Figura 15: Imagem apresentada aos sujeitos para localizar a fonte sonora em sua volta.

As posições foram variadas apenas no plano horizontal (movimentos em torno do eixo Z, *yaw*), mantendo sempre a elevação nula. Além disso, foi utilizado um sistema de dicas sobre tais posições: para as três primeiras reproduções, a dica é que a fonte poderia estar apenas na parte frontal da imagem, assim como as próximas três seriam apenas na parte traseira e as últimas três poderiam estar em qualquer uma das posições. Todos os casos poderiam incluir as posições de 90° e 270°, sendo elas, respectivamente, esquerda e direita. As posições de fonte eram

[20] Eles não foram realizados no mesmo dia e não necessariamente pelos mesmos sujeitos. Além disso, não foi averiguada a condição auditiva dos voluntários.

geradas com o *head-tracker* montado em uma *jog wheel*, peça giratória de uma controladora MIDI[21] geralmente utilizada para discotecagem, de forma que essa se assemelhasse a uma *turntable* e facilitasse a movimentação do dispositivo, sendo isso efetivo para avaliar alguns aspectos do funcionamento do *head-tracker*. Foram colocadas algumas demarcações em azul referentes à 0°, 90°, 180° e 270° para guiar os sujeitos durante os testes, observe a Figura 16.

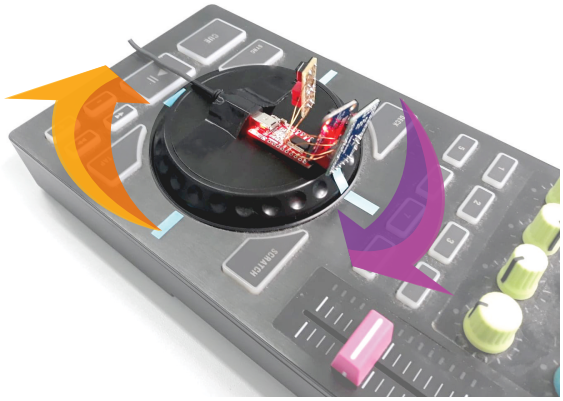


Figura 16: *Head-tracker* montado em *jog wheel*, voltado para a posição 0°.

Foi também realizado um teste, agora denotado como Teste 2, em que o sujeito deveria ouvir o áudio espacial porém, diferente do anterior, indicar a posição da fonte utilizando o *head-tracker* montado, da mesma maneira, na peça giratória supracitada, sendo preservadas também as dicas quanto às posições que, dessa vez, não eram discretas, podendo os sujeitos posicionar o dispositivo em qualquer lugar.

Foi utilizado, em ambos os testes, um *headphone* Sennheiser HD-650, cujo o filtro de compensação foi obtido no banco de dados de HRTFs de Fabian [37, 38] e um sinal musical, sendo, o volume geral do computador ajustado em 70% e do Matlab em 22% no *mixer* do computador. No primeiro teste o áudio foi reproduzido com a função *sound* do Matlab que, posteriormente, foi substituída pela função *play* do ITA Toolbox[22] que demonstrou maior controle e qualidade. Ambos os testes contaram com 20 participantes, sendo eles, em sua maioria, alunos do curso de Engenharia Acústica da UFSM [40].

Finalmente, a partir dos resultados obtidos nos Testes 1 e 2, que demonstraram algumas limitações em evidenciar o bom funcionamento do sistema utilizando o *head-tracker* devido aos fenômenos acústicos (ou de audição) que serão citados na seção seguinte, foi decidido conceber e aplicar um terceiro teste, denominado Teste Final. O objetivo desse último teste foi minimizar erros relacionados à tais fenômenos (aos quais ambos testes anteriores estavam mais suscetíveis) que na verdade não estavam associados ao funcionamento do dispositivo junto às rotinas desenvolvidas, podendo assim mascarar o resultado que indicaria o desempenho correto do sistema.

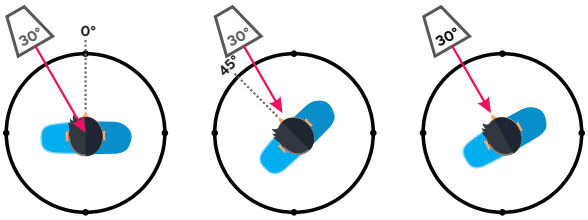
A ideia desse teste foi apresentar para o sujeito a fonte em uma determinada posição, de forma que, assim como no Teste 2, ele deveria sua localização, porém, dessa vez, virando-se em um banco giratório, utilizando o *head-tracker* fixado à um *headphone* (observe a Figura 17). Após o primeiro movimento, o sujeito ouviria a reprodução da fonte fixada na mesma posição, que seria calculada compensando a rotação realizada por ele, sendo o procedimento repetido até que o participante tivesse a sensação de estar de frente para a fonte sonora.

Figura 17: Sujeito utilizando o *headphone* com o *head-tracker* fixado para o Teste Final (em uma sala de aula comum).

[21] As mensagens nativas do *jog wheel* não foram utilizadas, ele serviu apenas como base girante.

[22] *Toolbox* de acústica para Matlab desenvolvido na Universidade de Aachen na Alemanha [39].

Um exemplo prático é apresentado para melhor entendimento: o sujeito virado para a posição frontal (0°) escuta com o *headphone* a fonte ser reproduzida na posição 30° (à esquerda), e então, após a reprodução cessar, o sujeito se move na tentativa de ficar de frente para a fonte sonora e atinge a posição 45°. Como a fonte se mantém na mesma posição e a rotação do sujeito é compensada na implementação do código, a próxima reprodução da fonte via *headphone* será em 345° (15° à direita). Após o sinal cessar novamente, o sujeito se vira para a posição correta e, ao final da reprodução nesta posição, confirma estar de frente para a fonte. A situação é ilustrada na Figura 18.



(a) Reprodução 1 do Teste Final. **(b)** Reprodução 2 do Teste Final. **(c)** Reprodução 4 do Teste Final.

Figura 18: Exemplificação do Teste Final da cadeia de reprodução.

O teste consistiu de possibilidades ilimitadas para o sujeito localizar as posições de fonte, de forma que, ao restringir a possível posição da fonte após consecutivas tentativas e reproduções, a taxa de acertos fosse maximizada. Como este processo eventualmente poderia ser demorado, foi escolhido o número de três posições para cada sujeito, em que cada uma delas tal procedimento deveria ser repetido, sendo elas: 60°, 230° e 330°.

O *headphone*, seu ajuste de volume e a maneira de reprodução foram idênticos ao do Teste 2. Da mesma forma, para cada participante era solicitado que sentasse voltado para uma parede específica e então era realizada a calibração do *head-tracker*. Para todos eles foi solicitado também que virassem cabeça e tronco juntos. Era realizada uma breve explicação do teste com uma ilustração no quadro negro para que compreendessem de maneira correta o procedimento. Ainda, era solicitado que tentassem evitar movimentos menores após estabelecerem-se em uma determinada posição após cada reprodução, pois essas poderiam significar alguma alteração no dado de orientação para geração da próxima posição de fonte.

4. RESULTADOS E DISCUSSÕES

É interessante comentar, primeiramente, que as dicas utilizadas nos testes foram motivadas pelas confusões geralmente presentes no contexto de acústica, sendo algumas delas bastante conhecidas e estudadas. As principais delas são a *reversão frente/costas*, que faz que a pessoa tenha impressão que uma fonte em sua frente está, na verdade, atrás (e vice-versa) [41], e o *cone de confusão*, região cônica cuja a localização é dificultada em torno de 90° e 270° , devido a ambiguidade nas pistas que o cérebro utiliza para localizar uma fonte no espaço [42]. Considerando que a ideia principal dos testes era evidenciar o bom funcionamento do *head-tracker* com os códigos desenvolvidos, tais erros deveriam ser minimizados, o que pôde ser feito restringindo a região de resposta dos sujeitos a partir das dicas mencionadas.

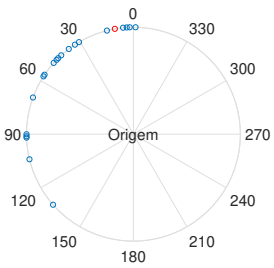
O Teste 1 citado gerou os resultados apresentados na Tabela 2, em que as linhas representam as marcações em uma determinada posição da folha de teste e as colunas representam o número de vezes que a reprodução com a fonte em uma determinada posição foi marcada nas posições disponíveis na folha, sendo importante salientar que uma das posições de reprodução era repetida, totalizando em um número maior de colunas do que linhas.

Tabela 2: Resultados obtidos no Teste 1 de localização de fonte (utilizando a Figura 15 e as dicas).

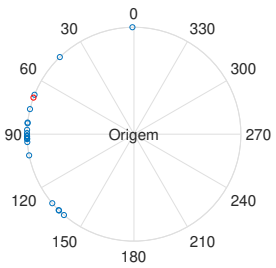
		R1	R2	R3	R4	R5	R6	R7	R8	R9
		46°	0°	90°	270°	134°	180°	314°	226°	180°
P1	46°	13	1	0	0	0	0	0	0	4
P2	0°	1	13	0	0	0	0	0	0	9
P3	90°	6	0	20	0	6	0	0	0	0
P4	270°	0	1	0	11	0	0	10	14	0
P5	134°	0	0	0	0	13	2	0	0	4
P6	180°	0	0	0	0	0	18	0	0	3
P7	314°	0	3	0	0	0	0	6	2	0
P8	226°	0	1	0	9	1	0	4	4	0

Como a diagonal principal é o número de acertos (células em azul e rosa), pode-se perceber com evidência que a utilização das dicas aumentou significativamente o número de acertos quando comparado às três últimas reproduções (células em rosa) em que a fonte poderia estar em qualquer uma das oito posições. Observando a tabela, nota-se que o pior caso das seis primeiras reproduções obteve apenas 11 acertos, e o melhor caso das três últimas reproduções apenas 6 acertos. Em especial, é interessante comparar a reprodução da fonte em 180° que, para a etapa com dicas obteve 18 acertos, porém na etapa sem dicas obteve apenas 3. Esses resultados demonstraram que, de fato, isolando as possibilidades de respostas dos sujeitos, tem-se um menor erro associado às confusões, fazendo com que o maior número de acertos pudesse comprovar que o sistema de rastreamento estava funcionando de maneira adequada com os códigos implementados.

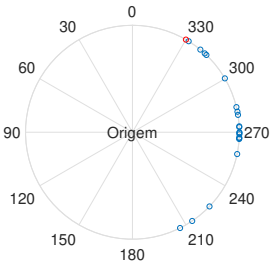
No Teste 2, como não havia um número discreto de opções de marcações, gerou-se uma nuvem de resultados em torno das posições reais da fonte que, em muitos casos, foi bastante dispersa pois os sujeitos, em grande parte, aparentaram não levar em consideração as dicas como no primeiro teste. Os resultados gráficos para nove posições são mostrados na Figura 18. Os pontos em vermelho representam a posição real da fonte e os pontos azuis, as respostas dos sujeitos.



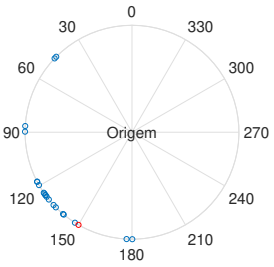
(a) Resultados para ângulo de azimute da fonte 10°.



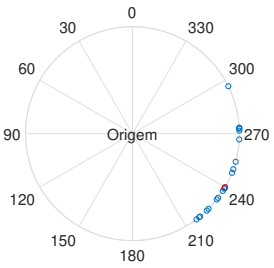
(b) Resultados para ângulo de azimute da fonte 70°.



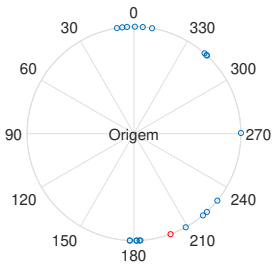
(c) Resultados para ângulo de azimute da fonte 330°.



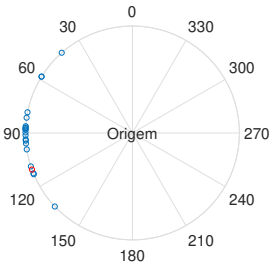
(d) Resultados para ângulo de azimute da fonte 150°.



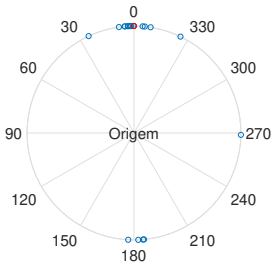
(e) Resultados para ângulo de azimute da fonte 240°.



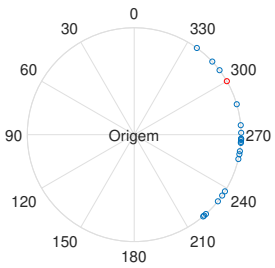
(f) Resultados para ângulo de azimute da fonte 200°.



(g) Resultados para ângulo de azimute da fonte 110°.



(h) Resultados para ângulo de azimute da fonte 0°.



(i) Resultados para ângulo de azimute da fonte 300°.

Figura 18: Resultados obtidos no Teste 2 de localização de fonte utilizando o *head-tracker*.

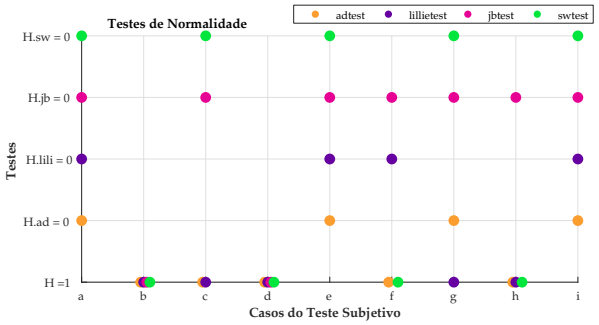


Figura 19: Testes de normalidade nos dados de (a) a (i) para o Teste 2.

Fica evidente pelos resultados[23] mostrados que os sujeitos tiveram mais dificuldade em localizar a posição exata da fonte, sendo isso possivelmente associado à falta de opções discretas de marcação (se comparado com o Teste 1). Também deve-se comentar sobre a possível aparição de outro fenômeno acústico que pode acontecer na audição espacial, o *erro de localização* ou *localization blur* [43]. Diferente das confusões que tem a ver com a ambiguidade das pistas que o cérebro utiliza para localização (reversão frente/costas e cone de confusão que, por sua vez, podem ter sido influentes neste teste também), ele está associado à sensibilidade espacial do sistema auditivo, ou seja, à capacidade de discriminar[24] ou não uma fonte em dois pontos relativamente próximos. Dessa maneira, isso pode ter feito com que os sujeitos não tenham conseguido definir exatamente uma posição para a fonte, podendo julgar que ela estava, por exemplo, em uma região entre $\pm 10°$ da posição verdadeira. Esse comportamento é evidente em alguns dos gráficos em que existem várias respostas muito próximas da posição real da fonte.

Para observar a nuvem de resultados, quatro testes de normalidade foram aplicados para cada ângulo, estes testes são indicados geralmente para conjuntos com poucas observações, são eles: Anderson-Darling (`adtest`) [44], Lilliefors (`lilietest`) [45], Jarque-Bera (`jbtest`) [46] e Shapiro-Wilk (`swtest`) [47]. A hipótese nula (H=0) indica que o teste não pode rejeitar a hipótese de que a distribuição seja gaussiana (ou normal), com uma certa significância (95% neste caso). Por simplicidade, apenas os resultados dos testes estão mostrados na Figura 19. Por exemplo, para o caso (e) da Figura 18 todos os quatro testes *concordam*, ao apresentar H=0. Não ter distribuição normal não indica erro, mas que o conjunto possui outra distribuição ou ainda que o conjunto deve ter mais observações para uma conclusão aprimorada acerca da média ($\bar{x}$) e variância (σ_x^2). Além disso, como pode ser verificado nos casos (f) e (h), existiu reversão frente/costas, como pode ser corroborado pela variância alta na Tabela 3, fazendo com que alguns testes obtivessem H=1. No caso (h), por exemplo, removendo reversões e *outliers*[25], todos os testes obtém H=0 (nesse caso o conjunto foi reduzido para 13 observações).

Para avaliar as respostas com valores numéricos, foi gerada também a Tabela 3 com a *média aritmética* das respostas, variância, mediana, melhor acerto, assim como o *Mean Absolute Error* (MAE ou erro médio absoluto), calculado a partir de

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N} |x_i - x_t|\,, \qquad (4)$$

em que N é o número total de valores medidos (ou observações), x_i os valores medidos e x_t o valor real (ou correto). Os valores foram arredondados para ficar sem casas decimais.

Tabela 3: Valores comparativos de desempenho para o Teste 2, casos de (a) a (i) da Figura 18.

Valor real (°)	10	70	330	150	240	200	110	0	300
Caso	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)
Média (°)	50	91	274	104	247	304	89	45	263
Variância (°)	37	32	33	74	22	107	20	70	30
Mediana (°)	45	91	271	125	240	317	89	5	266
MAE (°)	43	31	56	49	18	111	24	45	42
Melhor acerto (°)	14	68	328	148	240	209	108	0	307

[23] Com figuras de vetoriais, é possível realizar o *zoom* para observar melhor os detalhes nos gráficos.

[24] Por isso utiliza-se o termo *blur*, como se estivesse *borrado*, *nublado* ou ainda *embaçado*.

[25] Valor aberrante ou valor atípico, apresentando um grande afastamento em relação aos demais do conjunto.

De acordo com esses valores, observa-se grande dispersão de resultados, demonstrando *erro* de localização. É importante lembrar que a avaliação é apenas auditiva, com isso, nenhuma pista visual pode ajudar ao sistema de localização. Da perspectiva de análise dos resultados, os gráficos, testes de normalidade, média, variância, mediana e MAE são complementares, visto que os parâmetros isoladamente podem levar a conclusões errôneas. Da perspectiva do funcionamento do protótipo, observa-se um ótimo acordo com a teoria e expectativas de resultados.

Outro ponto importante a ser ressaltado é que, da maneira como o *head-tracker* estava preso ao *jog wheel*, que basicamente era com fitas adesivas, alguma força era exercida sobre tal sistema, fazendo com que ele se movimentasse involuntariamente. Dessa forma, percebeu-se que ao fim de alguns testes, mesmo com o dispositivo voltado para a fita azul referente à posição de 0°, o valor de ângulo lido no computador era diferente, podendo chegar até a 6°, fazendo-se concluir que, possivelmente, a força do cabo USB estivesse alterando a posição original de fixação do *head-tracker*.

Para averiguar esse comportamento, fez-se uma trajetória com ele montado (da mesma maneira que durante os testes) e salvaram-se os dados angulares para posterior processamento, na qual o ponto de começo era 0°, assim como o ponto final, sendo a fita azul a referência. Ao fim desse teste, pode-se perceber que, devido à força exercida pelo cabo, a posição do *head-tracker* era levemente alterada, podendo causar, então, uma deriva (*offset*) nos valores armazenados no teste de cada sujeito, ainda que fosse realizada a calibração da posição frontal para todos participantes, pois eles tinham as fitas azuis em torno do *jog wheel* com referência. O resultado de uma dessas trajetórias observadas pode ser visto na Figura 20, em que, ao começo do percurso, o valor alinhado com a fita azul era de 0,27° e, ao fim, de 3,68°. Lembrando que este foi um problema no experimento e não no protótipo em si.

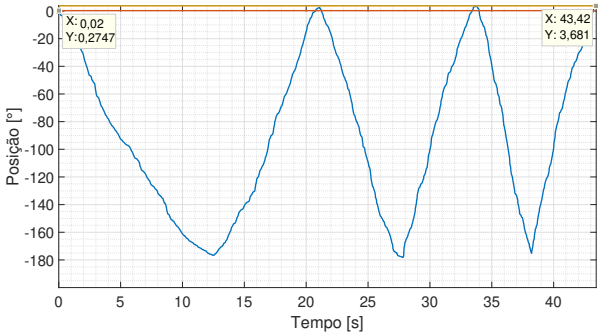


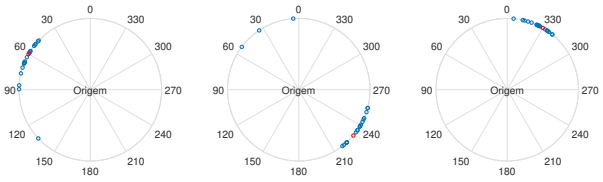
Figura 20: Trajetória do *head-tracker* montado no *jog wheel* ao longo do tempo.

Deve-se considerar também que um estudo realizado pelos desenvolvedores originais do projeto [8] demonstrou que o dispositivo possui um desvio padrão (para cinco trajetórias consecutivas em torno dos três ângulos de rotação *yaw, pitch* e *roll*) compreendido entre 0,5° e 2,5°. Porém, como não são apresentados estudos com trajetórias maiores no trabalho, nada garante que esse erro não possa ser maior em percursos mais extensos. Todavia, deve-se pontuar que este é um projeto de relativo baixo custo.

Esse teste demonstrou que para a utilização da cadeia de reprodução de maneira realista, ou seja, sem opções discretas de marcação, algum critério a mais deveria ser utilizado para restringir as respostas dos sujeitos para minimizar os erros associados tanto às confusões de localização quanto ao *blur*, e, por esse motivo, o Teste Final foi elaborado.

Nos resultados do Teste Final, como era esperado, obtiveram-se repostas (de 20 sujeitos) participantes bem mais próximas dos valores reais de fonte, uma vez que eles puderam restringir a possível posição da fonte após cada reprodução e eliminar ambiguidades (como nos casos de reversão frente/costas) com movimentos intermediários durante o teste. O diagrama contendo a posição real da fonte em vermelho e as respostas dos sujeitos em azul é apresentado na Figura 21.

É possível notar que, excluindo alguns *outliers*, as respostas são substancialmente menos dispersas que no Teste 2, cuja dinâmica de localização de fonte era similar ao Teste Final.



(a) Resultados para a posição 60°.

(b) Resultados para a posição 230°.

(c) Resultados para a posição 330°.

Figura 21: Resultados obtidos do Teste Final e distribuição dos dados desconsiderando os *outliers*.

Para quantificar de forma objetiva os resultados, foram calculados a média, a variância, a mediana, o erro médio absoluto (MAE) e o melhor acerto para cada posição, os dados estão apresentados na Tabela 4. Os valores entre parênteses são referentes aos cálculos **desconsiderando** os *outliers*.

Tabela 4: Dados extraídos do Teste Final (em negrito sem *outliers*).

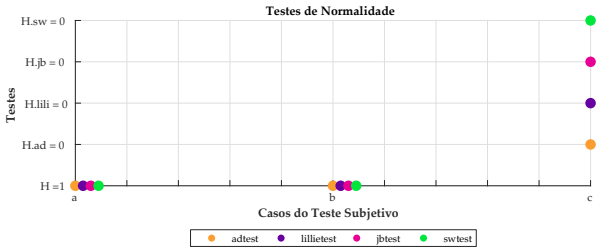
Valor real (°)	60	230	330
Caso	(a)	(b)	(c)
Média (°)	67 (**63**)	260 (**237**)	334
Variância (°)	20 (**12**)	58 (**12**)	9
Mediana (°)	65 (**63**)	240 (**239**)	334
MAE (°)	13 (**10**)	34 (**12**)	8
Melhor acerto (°)	63	234	328

Observa-se pelos resultados apresentados na tabela que as respostas para cada posição de fonte foram pouco dispersas, o que pode ser evidenciado principalmente pelos baixos valores de variância (dos casos desconsiderando os *outliers*). O erro calculado também apresentou valores relativamente baixos, porém talvez seja uma tarefa difícil eliminá-lo completamente pois provavelmente ele esteja associado ao fenômeno *blur*, ao qual testes subjetivos estão suscetíveis.

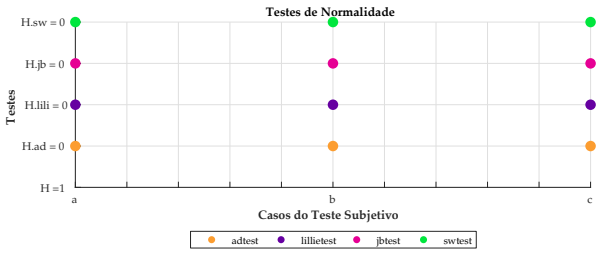
Ao comparar os valores sem *outliers* da Tabela 4 com os dados da tabela referente ao Teste 2, é possível perceber uma melhora considerável no desempenho dos sujeitos – exceto na linha referente ao *melhor acerto*, na qual em ambos os casos foram bem próximos do valor real da fonte.

É importante salientar que os valores sem considerar os *outliers* não foram calculados para o Teste 2 pois não faria sentido, pois grande parte dos dados seria desconsiderada, além de que o Teste 2 era naturalmente suscetível aos erros provenientes das confusões acústicas da audição. O Teste Final, por sua vez, tinha justamente o intuito de evitar tais erros.

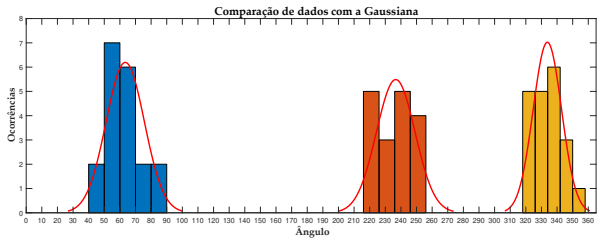
Do mesmo modo que no Teste 2, foram realizados também para o Teste Final os testes de normalidade para as respostas com e sem os *outliers*. Os dados sem *outliers* organizados em barras também foram comparados à curva Gaussiana. Esses gráficos podem ser vistos na Figura 22.



(a) Com *outliers*.



(b) Sem *outliers*.



(c) Sem *outliers*, comparação com a Gaussiana.

Figura 22: Testes de normalidade nos dados de (a) a (c) do Teste Final.

No caso desconsiderando os *outliers*, todos os testes apresentaram a mesma resposta, indicando que o conjunto de dados possui prova-

velmente uma distribuição normal. Isso reflete a situação de que humanos, em geral, possuem um comportamento semelhante quando submetidos às mesmas condições, fato evidenciado pela psicoacústica [1]. Todavia, para uma estabilidade estatística, mais experimentos devem ser realizados. Estes resultados corroboram para demonstrar a eficácia do rastreador construído.

5. CONSIDERAÇÕES FINAIS

O protótipo pode ser utilizado considerando as limitações apresentadas neste artigo, tanto na cadeia de gravação quanto na de reprodução propostas.

Apesar das alterações do protótipo UFSM, em relação ao projeto original, pôde-se obter um sistema semelhante cujo funcionamento é idêntico, devendo apenas serem utilizados os *software* adicionais que, eventualmente, podem ser substituídos por outras alternativas encontradas ao longo de novos desenvolvimentos.

Muito embora os resultados do Teste 1 tenham demonstrado que é possível reduzir os erros de localização restringindo as possibilidades de marcação, exaltando então o funcionamento do *head-tracker*, o Teste 2 ainda apresentou resultados que demonstram a dificuldade na localização quando não há demais pistas (*cues*). Fato inerente ao sistema auditivo e não ao rastreador de posição.

O Teste Final, por sua vez, reduziu consideravelmente a dificuldade de localização dos sujeitos, tornando as respostas mais próximas da posição real da fonte e, finalmente, indicando que o sistema proposto funciona de maneira adequada, ainda que certos aprimoramentos devam ser realizados. O desempenho de testes subjetivos pode ser aprimorado se HRTFs não generalizadas forem usadas [48], reduzindo talvez o erro que ainda se demonstrou presente no Teste Final.

Aplicações como a avaliação de HRTFs para cabeças artificiais ou humanas podem ser realizadas com o protótipo proposto, caso a incerteza associada não seja um impedimento. Isso pode ser cumprido com uma câmara anecoica e um aparato para girar ou a fonte sonora ou a cabeça. Caso seja utilizado em pessoas, o rastreamento da cabeça pode promover correções devido a dificuldade de se ficar parado estaticamente por muito tempo [49, 50].

Novos desenvolvimentos estão previstos para estudar a viabilidade da comunicação *wireless* (*bluetooth*), buscando evitar a incerteza apresentada no Teste 2 (referente a utilização de cabos). No entanto, é interessante observar que isso poderá causar outras incertezas em decorrência da velocidade de transmissão. Ademais, recomenda-se os testes com novas plataformas Arduino com a evolução dos MCUs.

REFERÊNCIAS

1. BLAUERT, Jens. *Communication Acoustics*. Berlin: Springer-Verlag, 2005. ISBN 9783540221623.

2. VORLÄNDER, Michael. *Auralization: Fundamentals of Acoustics, Modelling, Simulation, Algorithms and Acoustic Virtual Reality*. 1. ed. Berlin, Germany: Springer, 2007. 355 p. ISBN 978-3540488293.

3. XIE, Bosun. *Head-related transfer function and virtual auditory display*. Second edition. [S.l.]: J. Ross Publishing, 2013. ISBN 978-1604270709.

4. BLAUERT, Jens. *Spatial Hearing - Revised Edition: The Psychophysics of Human Sound Localization*. Revised. [S.l.]: The MIT Press, 1997. 508 p. ISBN 9780262024136.

5. GUERIN, Robert. *MIDI power*. Boston, MA: Thomson Course Technology, 2006. ISBN 978-1598630848.

6. RATTON, Miguel. *MIDI Total - Fundamentos e Aplicações*. [S.l.]: Editora Música & Tecnologia, 2005. ISBN 978-8589402057.

7. RUDRICH, Daniel; ZAUNSCHIRM, Markus; ROMANOV, Michael. *MrHeadTracker Wiki/Git. git.iem.at/DIY/MrHeadTracker/wikis*. Acessado em dez. 2018.

8. ROMANOV, Michael; BERGHOLD, Paul; FRANK, Matthias; RUDRICH, Daniel; ZAUNSCHIRM, Markus; ZOTTER, Franz. Implementation and Evaluation of a Low-Cost Headtracker for Binaural Synthesis. In: *Audio Engineering Society Convention 142*. [s.n.], 2017. Disponível em: http://www.aes.org/e-lib/browse.cfm?elib=18567.

9. MONK, Simon. *Programming Arduino : getting started with Sketches*. New York: McGraw-Hill, 2016. ISBN 978-1259641633.

10. JANTUNEN, Tommi; MESCH, Johanna; PUUPPONEN, Anna; LAAKSONEN, Jorma. On the rhythm of head movements in finnish and swedish sign language sentences. In: *Speech Prosody 2016: Proceedings of the 8th International Conference on Speech Prosody, Boston University, USA, 31 May-3 June 2016*. [S.l.: s.n.], 2016. p. 850–853. doi: 10.21437/SpeechProsody.2016-174.

11. COOK, Mike. *Arduino music and audio projects*. Berkeley, CA New York, NY: Apress/Springer, 2015. ISBN 978-1484217207.

12. TOCCI, Ronald J.; WIDMER, Neal; MOSS, Greg. *Digital systems : principles and applications*. [S.l.]: Prentice Hall, 2011. ISBN 978-0135103821.

13. PERROTT, David R.; SABERI, Kourosh. Minimum audible angle thresholds for sources varying in both elevation and azimuth. *The Journal of the Acoustical Society of America*, v. 87, n. 4, p. 1728–1731, 1990. doi: 10.1121/1.399421.

14. BRÜEL & KJÆR. *HATS Tipo 4128-C - Simulador de cabeça e tronco. www.bksv.com*. Acessado em dez. 2018.

15. DIAKOPOULOS, Dimitri. *HIDUINO GitHub. github.com/ddiakopoulos/hiduino*. Acessado em dez. 2018.

16. DIAKOPOULOS, Dimitri; KAPUR, Ajay. HIDUINO: A firmware for building driverless USB-MIDI devices using the Arduino microcontroller. In: *New Interfaces for Musical Expression (NIME) 2011*. Oslo, Norway: [s.n.], 2011. p. 405–408. doi: 10.5281/zenodo.1177995.

17. GUADALUPI, Arturo. *MIDIUSB Library for Arduino. github.com/arduino-libraries/MIDIUSB*. Acessado em dez. 2018.

18. SENSORTEC, Bosch. *Datasheet: BNO055 - Intelligent 9-axis absolute orientation sensor*. 2014.

19. Arduino. *Borad Comparison and Specs. www.arduino.cc/en/products/compare*. Acessado em dez. 2018.

20. STOFFREGEN, Paul. *Teensy Project. www.pjrc.com/teensy*. Acessado em dez. 2018.

21. KRONLACHNER, Matthias. *AmbiX v0.2.8 - Ambisonic plug-in suite (website). www.matthiaskronlachner.com*. Acessado em dez. 2018.

22. KRONLACHNER, Matthias. *AmbiX GitHub. github.com/kronihias/ambix*. Acessado em dez. 2018.

23. NACHBAR, Christian; ZOTTER, Franz; DELEFLIE, Etienne; SONTACCHI, Alois. AmbiX - A Suggested Ambisonics Format. In: *Ambisonics Symposium 2011*. Lexington, Kentucky, US: [s.n.], 2011. Disponível em: https://iaem.at/ambisonics/proceedings-of-the-ambisonics-symposium-2011.

24. Spatial Audio Real-time Applications (SPARTA). *An open-source VST audio plug-in suite for spatial audio production, reproduction and visualisation (website). http://research.spa.aalto.fi/projects/sparta_vsts/*. Acessado em dez. 2018.

25. MCCORMACK, Leo; POLITIS, Archontis. SPARTA & COMPASS: Real-time implementations of linear and parametric spatial audio reproduction and processing methods. In: *AES Conference on Immersive and Interactive Audio*. York, UK: [s.n.], 2019. Disponível em: http://www.aes.org/e-lib/browse.cfm?elib=20417.

26. MathWorks. *Matlab. www.mathworks.com*. Acessado em jun. 2018.

27. LAND, Bruce. *Arduino Oscilloscope: 688000 samples/sec*. *hackaday.io/project/425-arduino-oscilloscope-688000-samplessec*. Acessado em dez. 2018.

28. ZORZO, Arthur; FONSECA, William D'A. Estudo da técnica de identificação de sistemas implementada em microcontroladores Arduino Due e Teensy 3.6. *Acústica e Vibrações*, v. 32, n. 49, p. 5–14, dez. 2017. ISSN 1983-442X.

29. IGOE, Tom. *The Inter-IC Sound (I2S) Protocol*. *tigoe.github.io/SoundExamples/i2s.html*. Acessado em dez. 2018.

30. ADAFRUIT. *www.adafruit.com*. Acessado em dez. 2018.

31. SPARKFUN. *www.sparkfun.com*. Acessado em dez. 2018.

32. Future Technology Devices International Ltd (FTDI). *FT232RL Datasheet, Drivers, FAQ & Application Note*. *www.ftdichip.com/Products/ICs/FT232R.htm*. Acessado em dez. 2018.

33. Hairless MIDI to Serial Bridge. *projectgus.github.io/hairless-midiserial*. Acessado em nov. 2018.

34. Pure Data (Pd). *puredata.info*. Acessado em out. 2018.

35. SCHMITT, Daniel. *Nerds.de - Audio & MIDI Particles - LoopBe30 and LoopBe1*. *www.nerds.de/en/loopbe30.html*. Acessado em fev. 2018.

36. BOM, Enzo. *Desenvolvimento de cadeia de medição e reprodução biauricular utilizando dispositivo de rastreamento da cabeça*. Monografia (Trabalho de Conclusão de Curso) — Universidade Federal de Santa Maria, Santa Maria, RS, Brasil, 2018.

37. BRINKMANN, Fabian; LINDAU, Alexander; WEINZIERL, Stefan; GEISSLER, Gunnar; PAR, Steven; MÜLLER-TRAPET, Markus; OPDAM, Rob; VORLÄNDER, Michael. *The FABIAN head-related transfer function data base, Manual/Tech. Report, Technische Universität Berlin,*. [S.l.], 2017. doi: 10.14279/depositonce-5718.

38. LINDAU, Alexander; WEINZIERL, Stefan; MAEMPEL, H. J. FABIAN - An instrument for software-based measurement of binaural room impulse responses in multiple degrees of freedom. *24. Tonmeistertagung - VDT International Convention*, 2006. Disponível em: https://bit.ly/FABIAN-BRIR.

39. DIETRICH, P.; MASIERO, B.; MÜLLER-TRAPET, M.; POLLOW, M.; SCHARRER, R. Matlab Toolbox for the Comprehension of Acoustic Measurement and Signal Processing. In: *German Congress on Acoustics – DAGA*. Berlin, Alemanha: [s.n.], 2010. p. 517–518. ISBN 978-3-9808659-8-2. Disponível em: http://www.ita-toolbox.org/.

40. PAIXÃO, Dinara Xavier da; FONSECA, William D'Andrea. A experiência do ensino de graduação em Engenharia Acústica no Brasil. In: *FIA 2018 - XI Congreso Iberoamericano de Acústica; X Congreso Ibérico de Acústica; 49º Congreso Español de Acústica - TecniaAcustica'18*. Cadiz, Espanha: [s.n.], 2018. Disponível em: http://www.sea-acustica.es/fileadmin/Cadiz18/ENA-0_001.pdf.

41. MØLLER, Henrik. Fundamentals of binaural technology. *Applied Acoustics*, v. 36, p. 171–218, 1992. doi: 10.1016/0003-682X(92)90046-U.

42. PULKKI, Ville; KARJALAINEN, Matti. *Communication acoustics: an introduction to speech, audio and psychoacoustics*. [S.l.]: John Wiley & Sons, 2015. ISBN 978-1-118-86654-2.

43. DANIEL, Adrien. *Spatial Auditory Blurring and Applications to Multichannel Audio Coding*. Tese (Doutorado) — Université Pierre et Marie Curie - Paris VI, Paris, França, 2011. Disponível em: https://tel.archives-ouvertes.fr/tel-00623670/.

44. PETTITT, A. N. Testing the Normality of Several Independent Samples Using the Anderson-Darling Statistic. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, v. 26, n. 2, p. 156–161, 1977. doi: 10.2307/2347023.

45. LILLIEFORS, Hubert W. On the Kolmogorov-Smirnov Test for Normality with Mean and Variance Unknown. *Journal of the American Statistical Association*, Taylor & Francis, v. 62, n. 318, p. 399–402, 1967. doi: 10.1080/01621459.1967.10482916.

46. JARQUE, Carlos M.; BERA, Anil K. A Test for Normality of Observations and Regression Residuals. *International Statistical Review / Revue Internationale de Statistique*, [Wiley, International Statistical Institute (ISI)], v. 55, n. 2, p. 163–172, 1987. ISSN 03067734, 17515823. doi: 10.2307/1403192.

47. ROYSTON, J. P. An Extension of Shapiro and Wilk's W Test for Normality to Large Samples. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, v. 31, n. 2, p. 115–124, 1982. doi: 10.2307/2347973.

48. BOMHARDT, Ramona. *Anthropometric Individualization of Head-Related Transfer Functions Analysis and Modeling*. Tese (Doutorado) — RWTH Aachen University, Aachen, Alemanha, 2017. ISBN: 978-3-8325-4543-7. Disponível em: http://publications.rwth-aachen.de/record/699551.

49. MASIERO, B.; DIETRICH, P.; POLLOW, M.; FELS, J.; VORLÄNDER, M. Design of a Fast Individual HRTF Measurement System. In: *German Congress on Acoustics – DAGA*. Darmstadt, Alemanha: [s.n.], 2012. Disponível em: http://pub.dega-akustik.de/DAGA_2012.

50. RICHTER, Jan-Gerrit. *Fast measurement of individual Head-Related Transfer Functions*. Tese (Doutorado) — RWTH Aachen University, Alemanha, 2019. doi: 10.18154/RWTH-2019-04006.