



Física experimental com Arduino: ondas em uma corda tensionada

Experimental Physics with Arduino: waves in a tensioned string

Itamar V. de Sousa Jr.^{*1}, José O. S. Miranda¹, Alexandro C. S. Nascimento¹, Francisco R. V. Araújo²

¹Instituto Federal de Educação, Ciência e Tecnologia do Piauí - *Campus* Parnaíba, PI, Brasil

²Instituto Federal de Educação, Ciência e Tecnologia do Piauí - *Campus* São Raimundo Nonato, PI, Brasil

Recebido em 03 de Maio de 2020. Revisado em 21 de Junho de 2020. Aceito em 22 de Junho de 2020.

Neste trabalho foi desenvolvido um aparato para estudo do som emitido por cordas vibrantes utilizando a plataforma Arduino. Cordas de aço e nylon foram fixadas a um dispositivo que permitia aplicar uma força de tração às cordas e investigar a relação entre esta força e a frequência do som emitido utilizando em conjunto com o Arduino, respectivamente, um sensor de força baseado no módulo HX711 e um sensor de áudio baseado no circuito MAX4466. O sinal de áudio foi obtido a partir da conversão analógico-digital (A/D) da tensão elétrica no pino de saída do sensor de som. Para a obtenção de um sinal mais fidedigno, foram implementadas estratégias de programação que permitiram aumentar a velocidade de conversão A/D e diminuir o impacto da latência de comunicação Arduino-PC. Após análise espectral dos sinais de áudio, verificamos uma boa concordância entre previsões teóricas e os resultados obtidos experimentalmente mostrando, assim, que a abordagem adotada pode ser utilizada tanto no desenvolvimento de aulas experimentais como também na confecção de instrumentação para pesquisa na área de acústica.

Palavras-chave: ondas estacionárias, Arduino, acústica, aquisição de dados.

In this work we have developed an apparatus for studying the sound emitted by vibrating strings using the Arduino platform. Steel and nylon strings were attached to a device that made it possible to apply a strain to the strings and investigate the relationship between this force and the frequency of the sound emitted using, together with the Arduino, respectively, a force sensor based on the HX711 and an audio sensor based on the MAX4466 circuit. The audio signal has obtained from the analog-digital (A/D) conversion of the electrical voltage at the output pin of the sound sensor. To obtain a more reliable signal, we implemented programming strategies that allowed us to increase the A/D conversion speed and reduce the impact of Arduino-PC communication latency. After spectral analysis of the audio signals, we found a good agreement between theoretical predictions and the results obtained experimentally showing, thus, that the adopted approach can be used both in the development of experimental classes as well as in the manufacture of instrumentation for research in the acoustics.

Keywords: standing waves, Arduino, acoustics, data acquisition.

1. Introdução

A realização de experimentos é parte fundamental da construção do conhecimento científico e, atualmente, importantes teorias são verificadas a partir dos mais variados experimentos que ganham atenção da mídia e alimentam o imaginário popular. No ensino de Física, a realização de experimentos didáticos também desempenha um papel fundamental para o aprendizado, constituindo-se de um excelente recurso de ensino que permite aos estudantes uma compreensão mais profunda dos conteúdos.

No Brasil, a Lei de Diretrizes e Bases da Educação Nacional (LDB - Lei nº 9394/96) estabelece como uma das finalidades do ensino a compreensão dos fundamentos científico-tecnológicos dos processos produtivos, relacio-

nando a teoria com a prática, no ensino de cada disciplina [1]. Contudo, a realidade de grande parte das escolas é bem diferente e, muitas vezes, a realização de aulas práticas não ocorre devido à inexistência de laboratórios e materiais didáticos.

Diversas abordagens surgem como alternativas na tentativa de desenvolver experimentos de baixo custo e simulações computacionais que permitam a realização de aulas práticas e ajudem os estudantes a compreender os mais variados fenômenos e conceitos físicos em locais onde não é possível a aquisição de laboratórios didáticos comerciais com esta finalidade [2–4]. Neste contexto, a plataforma Arduino [5] tem se mostrado uma ferramenta muito útil e importante para professores e estudantes de Física, principalmente por conta de sua versatilidade.

*Endereço de correspondência: itamarvieira@ifpi.edu.br.

O uso da plataforma Arduino facilita a concepção e a realização de experimentos de Física por disponibilizar o *software* e o *hardware* que permitem o controle e o processamento de dados obtidos a partir de sensores externos, ou mesmo em suas portas I/O. Além disso, a plataforma Arduino possui comunicação USB, facilitando a interação com computadores, e é compartilhada sob uma licença gratuita (*Creative Commons*), contando com o desenvolvimento colaborativo de milhões de usuários e uma enorme disponibilidade de informações na internet.

Diversos trabalhos estão sendo desenvolvidos com o intuito de elaborar experimentos de Física de baixo custo utilizando esta plataforma. Os experimentos propostos são direcionados para uma grande variedade de temas dentro da Física tais como mecânica [6–9], termodinâmica [10–12], eletricidade [13–15], óptica [16, 17] e física moderna [18, 19]. A grande quantidade de informações disponíveis sobre Arduino na internet simplifica seu uso e permite o rápido desenvolvimento de protótipos. Existem inúmeras bibliotecas¹ nos mais diversos repositórios que facilitam a configuração da placa e o desenvolvimento de *firmwares*² para a execução das mais variadas atividades.

Em contrapartida, esta relativa facilidade para encontrar informações sobre a plataforma Arduino com bibliotecas e rotinas “prontas” pode ocultar do estudante ou professor de Física desavisado os mecanismos utilizados para implementar determinada tarefa. O uso destas bibliotecas oculta do usuário informações que podem ser importantes em experimentos de Física. Um bom exemplo são as funções `millis()` e `micros()`, que fornecem o tempo (em milissegundos e microssegundos, respectivamente) desde que a placa foi inicializada e são algumas vezes utilizadas para a determinação do intervalo de tempo entre eventos, mas sem deixar claro para o usuário a forma que são implementadas, omitindo, por exemplo, a resolução temporal desta contagem. Outra situação comum em que pode haver problemas é a utilização da função `analogRead()`, que implementa, em uma única linha de código, todos os passos necessários à leitura de uma tensão elétrica em uma entrada analógica da placa, mas omitindo informações como a resolução desta leitura e o tempo necessário para que ela ocorra de forma correta.

A precisão dos experimentos de Física realizados com a plataforma Arduino pode ser melhorada se o *hardware* utilizado for melhor conhecido pelo usuário, já que estratégias de programação que permitam obter maior fidedignidade dos resultados experimentais podem ser implementadas de forma mais eficaz.

Neste contexto, o presente trabalho tem como objetivo principal desenvolver, utilizando a plataforma Arduino, um aparato que permita estudar os modos normais de

vibração de uma corda tensionada e, além disso, discutir alguns tópicos importantes relacionados às estratégias de programação que podem ter impactos diretos sobre os resultados experimentais quando realiza-se medidas usando esta plataforma.

2. Conhecendo o Arduino

O projeto Arduino foi desenvolvido com o objetivo de facilitar a prototipagem de sistemas eletrônicos microcontrolados. Inicialmente o projeto visava permitir que artistas e pessoas com pouco conhecimento em eletrônica pudessem desenvolver tarefas como controlar sistemas de iluminação e outras automações simples sem a necessidade de recorrer a projetos customizados desenvolvidos por profissionais [5].

Existem diversos modelos de placas Arduino no mercado, no entanto, apesar de possuírem características diferentes, compartilham um conjunto de circuitos eletrônicos semelhantes, cujo principal componente é um microcontrolador.

Um microcontrolador pode ser entendido como um computador de chip único, onde temos encapsulados num circuito integrado a unidade lógica aritmética (ULA), memórias, barramentos de comunicação, além de periféricos programáveis para entrada e saída de dados (I/O).

Conhecer a arquitetura do microcontrolador presente em sua placa Arduino, mesmo que de maneira superficial, é fundamental para o pesquisador que deseja desenvolver instrumentação a partir desta plataforma. Por exemplo, se o microcontrolador da placa utilizada é de 8 bits, o maior valor que pode ser manipulado em passo único (numa única instrução ou ciclo de *clock*) é o número inteiro 255. Qualquer variável definida pelo usuário em seu código que assuma um valor superior a este necessitará de mais tempo de processamento. Isto significa que, para aplicações em que o tempo é uma variável crítica, até mesmo a escolha dos tipos de variáveis definidas pelo usuário terá impacto sobre os resultados. De modo que o conhecimento dos periféricos disponíveis no microcontrolador e sua forma de funcionamento ajuda a determinar a melhor forma de utilizar a placa Arduino ou optar por um modelo adequado a cada situação.

Para que os microcontroladores operem corretamente, é necessário um conjunto auxiliar de circuitos eletrônicos responsáveis por funções como regulagem de tensão, geração de sinal de *clock*, proteção térmica, entre outras. É neste ponto que o projeto Arduino apresenta uma grande vantagem, pois integra na mesma placa de circuito impresso todos os componentes necessários ao funcionamento do microcontrolador. Na figura 1, em destaque, estão os principais componentes da placa Arduino Mega 2560.

¹Coleção de subprogramas utilizados no desenvolvimento de um *software*.

²Conjunto de instruções, análogo a um programa, que é gravado no microcontrolador. Por simplicidade, pode-se entender o *firmware* como um programa que é executado diretamente na *hardware*, sem a necessidade de um sistema operacional.

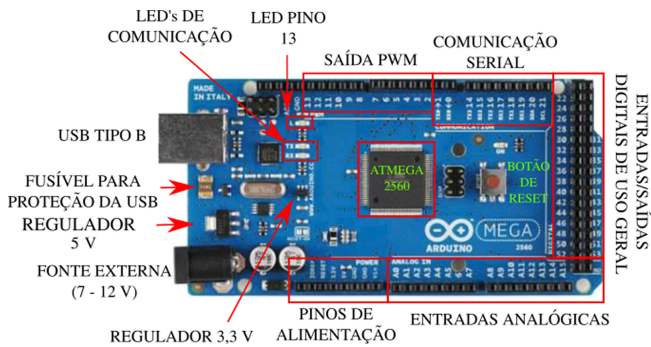


Figura 1: Placa Arduino Mega 2560. Em destaque estão os principais componentes que compõem esta placa.

2.1. Programação do Arduino

Para a criação dos *firmwares*, o projeto Arduino disponibiliza um ambiente de desenvolvimento integrado (IDE, para seu acrônimo em língua inglesa) próprio, que permite ao usuário programar o microcontrolador da placa. Este IDE é o software que interpreta as instruções escritas pelo usuário em uma linguagem de programação própria, baseada em C/C++, compila³ e executa a gravação.

A gravação dos *firmwares* nos microcontroladores das placas Arduino dispensa dispositivos externos. Em geral, as placas possuem circuitos eletrônicos auxiliares que permitem receber⁴, pela porta USB, os códigos gerados pelo IDE em um PC. Os microcontroladores das placas Arduino são configurados de fábrica com um *bootloader*, que é um pequeno *firmware*, gravado em um espaço reservado, que se inicializa sempre que a placa Arduino é ligada e verifica se deve carregar um novo *firmware* no microcontrolador. Esta é uma das grandes vantagens da plataforma Arduino em relação às outras placas de desenvolvimento, e foi o que permitiu sua rápida popularização.

3. Ondas em uma corda tensionada

O modelo mais simples que permite a compreensão da propagação de uma onda em uma corda tensionada é aquele em que considera-se uma perturbação unidimensional e assume-se que os deslocamentos da corda a partir da sua posição de equilíbrio são pequenos. Para esta configuração, a propagação da onda considerada é descrita por uma equação diferencial do tipo:

$$\frac{\partial^2 y(x, t)}{\partial x^2} = \frac{1}{v^2} \frac{\partial^2 y(x, t)}{\partial t^2}, \tag{1}$$

onde $y(x, t)$ é o deslocamento transversal no ponto x da corda considerada no instante de tempo t e v é a

³O termo *compilar* é utilizado para designar o processo de geração do código hexadecimal (*firmware*) que será gravado na memória de programa do microcontrolador.

⁴Nas placas Uno e Mega isto é feito utilizando um microcontrolador auxiliar, o ATmega16U2 que é responsável pela interface USB-Serial.

velocidade de propagação. É importante ressaltar que esta equação descreve um grande número de fenômenos ondulatórios, inclusive as ondas eletromagnéticas e sonoras (vibrações mecânicas longitudinais, tridimensionais e com velocidades de propagação dependentes das condições do meio em que se propagam). Sendo μ a densidade linear de massa e T a força de tensão, a velocidade de propagação pode ser escrita como:

$$v = \sqrt{\frac{T}{\mu}}. \tag{2}$$

A solução geral da equação (1) pode ser obtida pelo método de separação de variáveis, onde considerando-se uma corda de comprimento L presa nas suas extremidades, tem-se as seguintes condições de contorno:

$$y(0, t) = y(L, t) = 0, \tag{3}$$

assumindo que a corda é deslocada da sua posição de equilíbrio e solta em seguida, começando a vibrar no instante $t = 0$, as condições iniciais são dadas por:

$$y(x, 0) = y_0(x) \quad \text{e} \quad \frac{\partial y}{\partial t}(x, 0) = v_0(x), \tag{4}$$

onde $y_0(x)$ é uma função não nula que descreve a configuração da corda em $t = 0$ e $v_0(x)$ é a velocidade de propagação inicial. A partir destas informações é possível determinar uma solução geral para a equação (1):

$$y(x, t) = \sum_{n=1}^{\infty} \sin n\pi \frac{x}{L} \left[A_n \cos n\pi \frac{x}{L} + B_n \sin n\pi \frac{x}{L} \right], \tag{5}$$

onde $n = 1, 2, 3, \dots, \infty$. Os coeficientes A_n e B_n dados por:

$$A_n = \frac{2}{L} \int_0^L y_0(x) \sin n\pi \frac{x}{L} dx, \tag{6}$$

$$B_n = \frac{2}{n\pi v} \int_0^L v_0(x) \sin n\pi \frac{x}{L} dx, \tag{7}$$

são determinados utilizando-se o “truque de Fourier”.

3.1. Ondas estacionárias e modos normais de vibração

Sejam $y_1(x, t)$ e $y_2(x, t)$ duas funções que descrevem duas ondas que propagam-se em sentidos opostos e possuem as mesmas amplitudes e constantes de fase nulas, ou seja:

$$y_1(x, t) = A \cos(kx - \omega t), \tag{8}$$

$$y_2(x, t) = A \cos(kx + \omega t), \tag{9}$$

somando estas duas funções, tem-se uma outra função dada por:

$$\begin{aligned} y'(x, t) &= y_1(x, t) + y_2(x, t), \\ y'(x, t) &= 2A \cos(kx) \cos(\omega t), \end{aligned} \tag{10}$$

chamada de onda resultante. Como a onda resultante é o produto de uma função que depende de x por uma função que depende de t , não há propagação, resultando em uma onda estacionária.

As frequências dos modos normais de vibração, f_n , formam o espectro de vibração da corda e são dadas por:

$$f_n = \frac{nv}{2L} = \frac{n}{2L} \sqrt{\frac{T}{\mu}}, \quad (11)$$

onde n representa os modos normais de vibração com modo fundamental para $n = 1$. Desta forma, as frequências dos modos normais de vibração são múltiplos inteiros do modo fundamental.

4. Aparato experimental e calibração

O aparato experimental pode ser visto na figura 2 e é composto pelos seguintes itens:

1. Sensor de Som: pequena placa de circuito impresso contendo um microfone de eletreto conectado ao circuito integrado MAX4466 [20];
2. Sensor de Força: célula de carga conectada ao módulo HX711 [21];
3. Suporte para as cordas com regulador na extremidade;
4. Arduino Mega2560;
5. Computador.

Escolhemos dois tipos de cordas de violão para nossas análises, uma de nylon e a outra de aço, que são fixadas ao suporte de madeira tendo uma de suas extremidades presa ao sensor de força (2) e a outra ao regulador (3) que permite variar a tensão mecânica ao qual a corda é submetida. O comprimento da corda entre esses dois pontos é dado por $L = 66$ cm.

Com um auxílio de uma balança de precisão e uma régua, determinamos a densidade linear de massa das duas cordas, sendo $\mu_N = 0,511$ g/m para a corda de nylon e $\mu_A = 0,664$ g/m para a corda de aço. Tais valores serão usados para o cálculo de velocidade de propagação

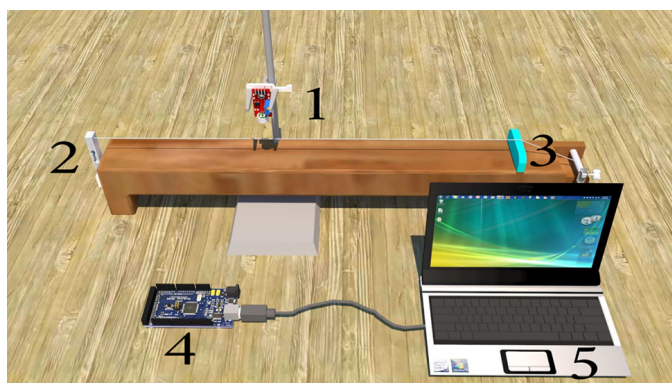


Figura 2: Representação esquemática do aparato experimental utilizado.

da onda na corda assim como serão referências para verificar nossas implementações no processo de aquisição de dados.

Antes de efetuarmos as medidas em nosso aparato, apresentaremos os processos de calibração que serão descritos nas seções 4.1 e 4.2.

4.1. Calibração do sensor de força

O sensor de força utilizado consiste de uma célula de carga com capacidade máxima de 20 kg conectada a um módulo HX711⁵. Quando submetida à uma força, a célula de carga sofre pequenas deformações que alteram sua resistência elétrica. Esta alteração é traduzida em um sinal elétrico que é amplificado, convertido num sinal digital pelo circuito integrado HX711 e enviado à placa Arduino.

O sinal digital enviado ao Arduino deve então ser interpretado para que possamos determinar a intensidade desta força. Para esta finalidade, utilizou-se a biblioteca *hx711.h* disponível em [22]. Esta biblioteca implementa o algoritmo que permite determinar o fator de calibração da célula de carga utilizado um objeto de peso conhecido.

4.2. Calibração do sensor de som

O sensor de som utilizado consiste numa pequena placa de circuito impresso contendo um microfone de eletreto conectado ao circuito amplificador MAX4466. Para determinar o ganho e a resposta em frequência deste sensor foi montado um aparato auxiliar que consistia de um gerador de áudio e um osciloscópio mostrado na figura 3.

O sensor foi posicionado à uma distância fixa do alto-falante e os dois canais do osciloscópio foram utilizados

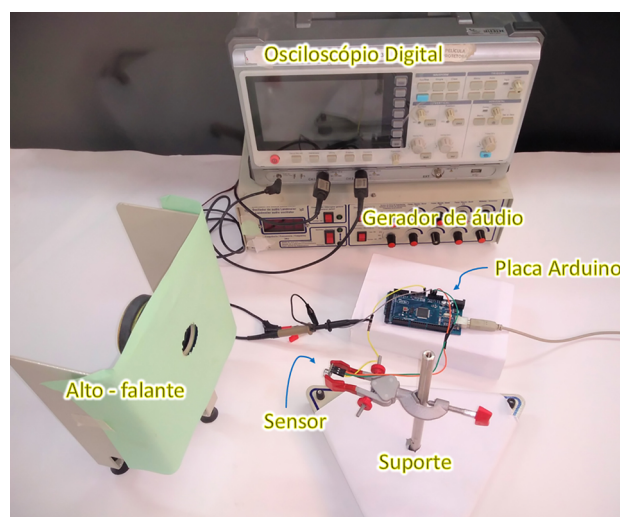


Figura 3: Proposta utilizada para a calibração do aparato experimental.

⁵Circuito integrado dotado de um amplificador de ganho programável e conversor Analógico-Digital de 24 bits que é frequentemente utilizado para aquisição de sinais de células de carga.

para comparar o áudio capturado pelo sensor com o sinal emitido pelo alto-falante. O sensor foi então ajustado para ter melhor desempenho na faixa de frequências do som emitido pelas cordas utilizadas no experimento.

5. Resultados e Discussão

Para melhor compreensão da nossa análise, dividimos a apresentação dos resultados da seguinte forma: Na subseção 5.1 investigaremos o comportamento mecânico das cordas usando o sensor de força, onde determinaremos a velocidade de propagação da onda na corda pela equação 2; ; Na subseção 5.2 discutiremos os fatores que interferem na aquisição do sinal, investigando a latência na comunicação serial e fazendo uma breve digressão sobre as estratégias utilizadas na otimização do programa. Na seção 5.3 determinaremos a velocidade de propagação das ondas a partir dos espectros de frequências obtidos quando as cordas são submetidas à diferentes trações. Finalmente, discutiremos a precisão das medidas comparando os valores das velocidades de propagação da onda nas cordas pelos dois métodos, além de determinarmos os valores das densidades lineares de cada corda a partir dos espectros de frequência.

5.1. Comportamento mecânico das cordas

Inicialmente investigamos o comportamento das cordas em função da tensão mecânica aplicada submetendo-as a diferentes valores de forças de tração cujas intensidades foram acompanhadas em função do tempo utilizando-se para tanto o Arduino e o Sensor de Força. Observamos que a corda de nylon sofre um processo de relaxação mecânica após a aplicação de uma força, conforme mostrado na figura 4(a). O mesmo fenômeno não é observado, de forma tão acentuada, quando utilizamos cordas de aço, conforme figura 4(b). Tal diferença ocorre pelo fato de o nylon ser um polímero e, como tal, sofrer deformações de forma diferente dos sólidos cristalinos em função dos processos de relaxação moleculares que ocorrem nesses materiais [23].

Com base nesses resultados, estabeleceu-se um protocolo de medição que consiste em aguardar um intervalo de aproximadamente 150 segundos entre as medidas de força para que a corda, assim, pudesse atingir o novo estado de equilíbrio mecânico e, dessa forma, obtermos valores constantes de força.

Para os valores de tensão que as cordas foram submetidas (indicadas na figura 4 com valores dados na tabela 1) e usando o valor da densidade linear para cada corda, $\mu_N = 0,511 \text{ g/m}$ para o nylon e $\mu_A = 0,664 \text{ g/m}$ para o aço, determinamos os valores de velocidade da onda na corda para cada tração, conforme equação 2. Esses valores também estão mostrados na tabela 1.

Na parte final dessa seção tomaremos esses valores de velocidade e força para verificar a precisão das medidas

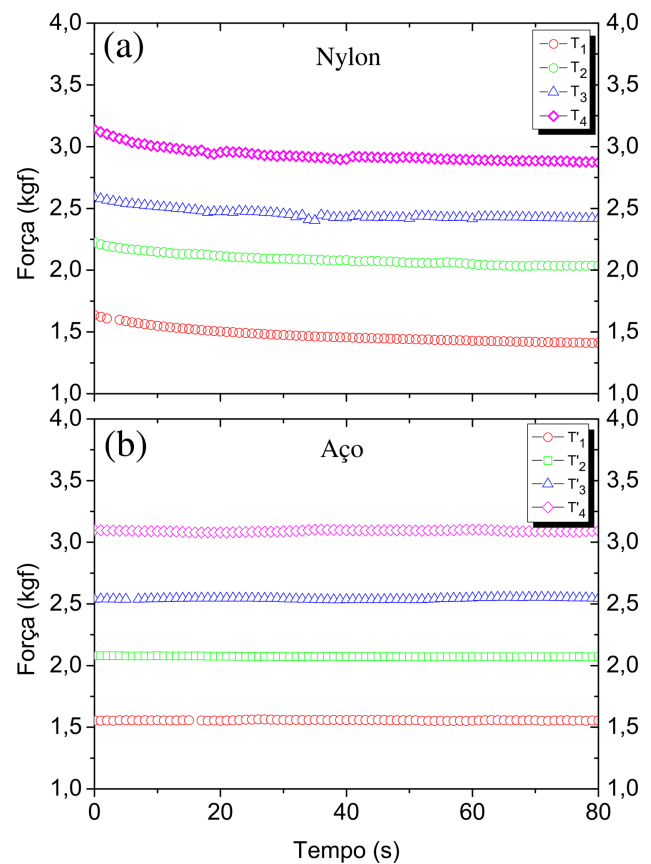


Figura 4: Gráfico da força em função do tempo para uma corda (a) nylon e (b) aço. Os valores de força medidos são mostrados na tabela 1.

Tabela 1: Correspondentes velocidades de propagação da onda nas cordas de nylon e aço para cada valor de tração.

	Tração (kgf)	Velocidade (m/s)
Nylon	$T_1 = 1,40$	$v_1 = 163,9$
	$T_2 = 2,00$	$v_2 = 195,9$
	$T_3 = 2,40$	$v_3 = 214,6$
	$T_4 = 2,80$	$v_4 = 231,8$
Aço	$T'_1 = 1,55$	$v'_1 = 151,3$
	$T'_2 = 2,07$	$v'_2 = 174,8$
	$T'_3 = 2,55$	$v'_3 = 194,1$
	$T'_4 = 3,07$	$v'_4 = 212,9$

de frequência após as implementações na comunicação serial entre o Arduino e demais periféricos.

5.2. Captura do sinal de áudio com o Arduino

Para capturar o sinal de áudio, optou-se inicialmente pela abordagem mais simples, onde o pino de saída do sensor de som era conectado a uma das entradas analógicas da placa Arduino que realizava a leitura e transferia o valor via comunicação serial para o PC.

Contudo, esta abordagem se mostrou ineficiente, pois o sinal obtido não representava de forma satisfatória a

forma de onda. Para compreender a causa do problema dividimos a análise em duas partes, onde investigamos a influência da latência da comunicação serial e a velocidade de conversão do ADC interno da placa. Os resultados desta análise serão apresentados a seguir.

5.2.1. Impacto da comunicação serial

Foi observado, nas medidas iniciais, que a comunicação Arduino-PC tinha grande impacto sobre a amostragem do sinal de áudio, o que pode ser observado na figura 5, onde o Arduino realiza a conversão AD utilizando a função interna da IDE [`analogRead()`] e envia o valor para o PC utilizando outra função interna [`Serial.print()`] a uma taxa de 9600 bps⁶ para cada amostra do sinal coletada no sensor quando este recebe um som de frequência 500 Hz.

Utilizando esta metodologia, vemos que o sinal capturado pelo Arduino contém pontos que estão em um ciclo diferente da onda sonora. Suspeitamos que isso ocorria em função do protocolo de comunicação utilizado pela IDE do Arduino. Na tentativa de compreender melhor este problema, utilizamos taxas de comunicação mais altas até o limite disponível. A figura 6 mostra a comparação entre o sinal reconstruído a partir do envio dos valores da leitura AD realizada pelo Arduino utilizando a taxa de 115 kbps e o sinal obtido com osciloscópio para um áudio de 500 Hz.

À medida que aumentamos a taxa de comunicação serial, a resolução do sinal obtido com o Arduino melhorava significativamente, confirmando assim nossa hipótese acerca do impacto da comunicação serial. Contudo, as taxas maiores que 115 kbps não apresentaram melhora significativa. Mesmo utilizando a maior taxa de comunicação possível (2 Mbps), o sinal reconstruído ainda continha pouca correlação com a onda original, conforme vemos na figura 7.

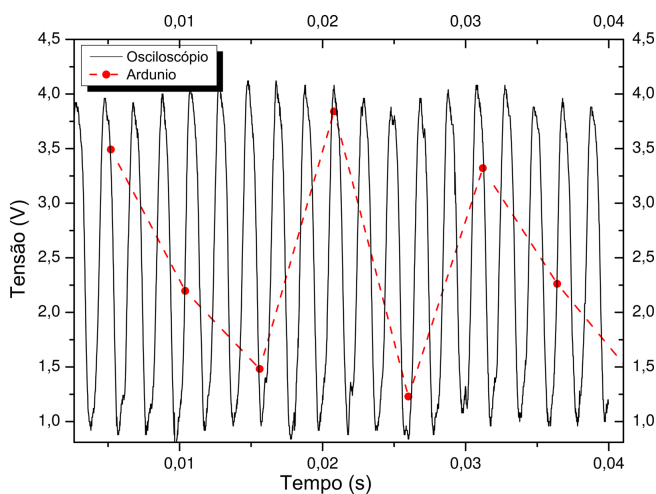


Figura 5: Sinal senoidal a partir de um gerador de frequência coletado com o Arduino e Osciloscópio simultaneamente.

⁶bits por segundo.

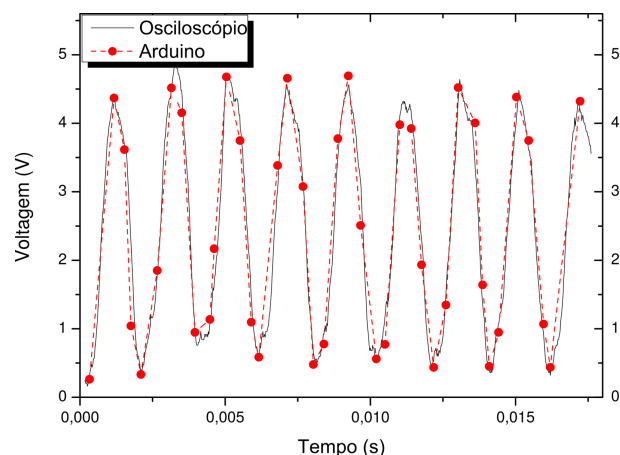


Figura 6: Comparação entre a forma de onda reconstruída a partir dos dados obtidos pelo Arduino (pontos vermelhos) e enviados para o PC a uma taxa de 115 kbps e o mesmo sinal obtido em um osciloscópio (curva preta).

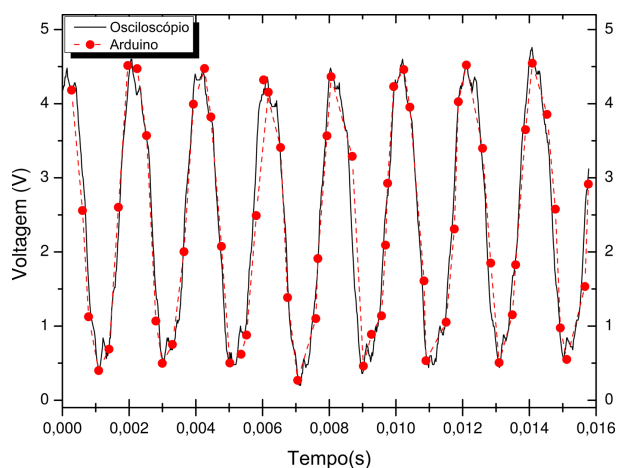


Figura 7: Comparação entre a forma de onda reconstruída a partir dos dados obtidos pelo Arduino (pontos vermelhos) e enviados para o PC a uma taxa de 2 Mbps e o mesmo sinal obtido em um osciloscópio.

Neste ponto, suspeitou-se que o fator que poderia estar interferindo na amostragem do sinal de áudio já não seria mais a comunicação, mas sim, a velocidade de conversão AD. Para confirmar, realizamos alguns testes e ajustes cujos resultados serão mostrados na próxima seção.

5.2.2. Conversão Analógico-Digital

Para determinar o tempo necessário para uma conversão Analógico-Digital, realizamos um teste que consistia em efetuar uma leitura AD utilizando a função interna da IDE [`analogRead()`] e, logo em seguida, alternar o estado de um pino⁷ gerando uma onda quadrada cujo período foi acompanhado com um osciloscópio. O resultado pode

⁷Isso foi feito utilizando uma macro que altera diretamente os registradores do microcontrolador. Com isso, o tempo necessário para executar o *toggle* de estado era da ordem dos nanossegundos e poderia ser desprezado nesta análise.

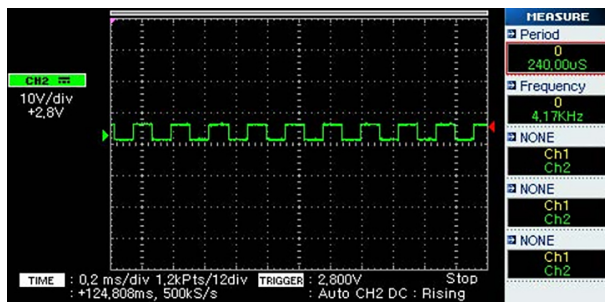


Figura 8: Onda quadrada utilizada como referência para investigar a velocidade de conversão AD. Cada transição de nível lógico corresponde à uma leitura AD utilizando a função interna da IDE `analogRead()`.

ser visto na figura 8 onde observamos que o tempo para uma conversão era cerca de 120 μ s.

Este valor está em acordo com o calculado a partir das informações do *datasheet* [24], pois o conversor AD do microcontrolador Atmega2560 necessita de 13 ciclos de *clock* para efetuar a conversão e a função `analogRead()` efetua uma configuração que faz com que o *clock* do ADC seja de aproximadamente 115 kHz. Com esta configuração o número de pontos coletados por segundo é aproximadamente 8333 o que, para uma onda de 500 Hz, significa que teremos menos de 17 pontos por ciclo, o que seria insuficiente para reconstruir os harmônicos de maior frequência de um som.

Como não podemos alterar de forma simples os parâmetros implementados pela função `analogRead()`, decidimos configurar o ADC utilizando os registradores do microcontrolador destinados a esta tarefa. Conforme descrito no *datasheet*, o microcontrolador Atmega2560 da placa Arduino Mega conta com um conversor Analógico-Digital de aproximações sucessivas que utiliza *clock* derivado do sinal principal (gerado pelo cristal de 16 Mhz) e cuja frequência é determinada pela configuração dos bits ADPS⁸ do registrador ADCSRA⁹. Além dessa configuração, utilizou-se também o modo “ADC *Free Running*”, que consiste fazer o ADC converter continuamente a entrada e lançar uma interrupção¹⁰. ao final de cada conversão. O ADC foi então configurado para operar no modo 8 bits, isto permitiu utilizar uma frequência de *clock* de 500 KHz, permitindo uma taxa de amostragem de aproximadamente 38,4 KHz, ou seja, cerca de 38.400 amostragens por segundo. A configuração deste modo pode ser verificada no material suplementar.

Por fim, com o objetivo de melhorar ainda mais a qualidade do sinal obtido, utilizamos uma estratégia de programação que consistia em utilizar no programa uma

matriz de valores para armazenar em *buffer*¹¹ os dados coletados. A figura 9 mostra a forma da onda capturada utilizando as configurações descritas até aqui.

Dessa forma, podemos concluir que a qualidade do sinal obtido é afetada por dois fatores: a configuração do Arduino por meio das funções da IDE e a comunicação serial dessa plataforma com o PC durante a medida. Em processos de medida que envolvem curtos intervalos de tempo, esse cuidado deve ser levado em conta.

5.3. Espectro sonoro

Após o ajuste do sistema para minimizar os efeitos tanto de configuração de funções da IDE como comunicação serial no processamento dos dados medidos pelo Arduino, realizamos medidas do som emitido tanto pelas cordas de nylon como de aço, quando postas a oscilar, e seus respectivos espectros de frequência para os mesmos valores de força medidos pelo sensor de força e dispostos na tabela 1. Todos os espectros são mostrados na Fig. 10.

Na primeira coluna, Fig. 10(a)-(d), temos as medidas realizadas para a corda de nylon, onde são indicados os valores de força para a qual a corda foi submetida. Na outra coluna, 10(e)-(h), temos os espectros para a corda de aço. Esses dados foram coletados no monitor serial do Arduino e posteriormente usamos o software SciDAVis para determinar os espectros de frequência.

Conforme prevê a equação (5), a onda sonora complexa mostrada no *insert* de cada figura é composta pela superposição dos seus harmônicos. São esses harmônicos que determinam a propriedade denominada timbre. Podemos notar que o timbre é modificado tanto para o tipo de corda como para a tensão mecânica ao qual as mesmas estão submetidas.

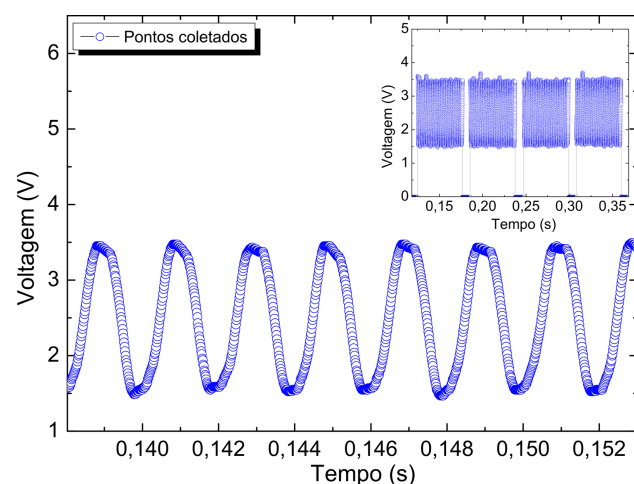


Figura 9: Na parte superior vemos a conversão dos AD dos dados armazenados em *buffer*. Cada bloco separado corresponde a uma quantidade de 4000 pontos armazenados. Na parte inferior vemos a ampliação desses dados mostrando a onda que foi armazenada.

⁸ADC *Prescaler Select Bits*. Para definir o *prescaler* são utilizados os bits ADPS2, ADPS1 e ADPS0.

⁹ADC *Control and Status Register A*.

¹⁰Uma interrupção pode ser entendida como uma sinalização para que o programa interrompa seu fluxo e seja desviado para atender um dado conjunto de instruções [25]

¹¹Parte da memória utilizada para guardar dados temporariamente enquanto estes se transferem de um dispositivo a outro.

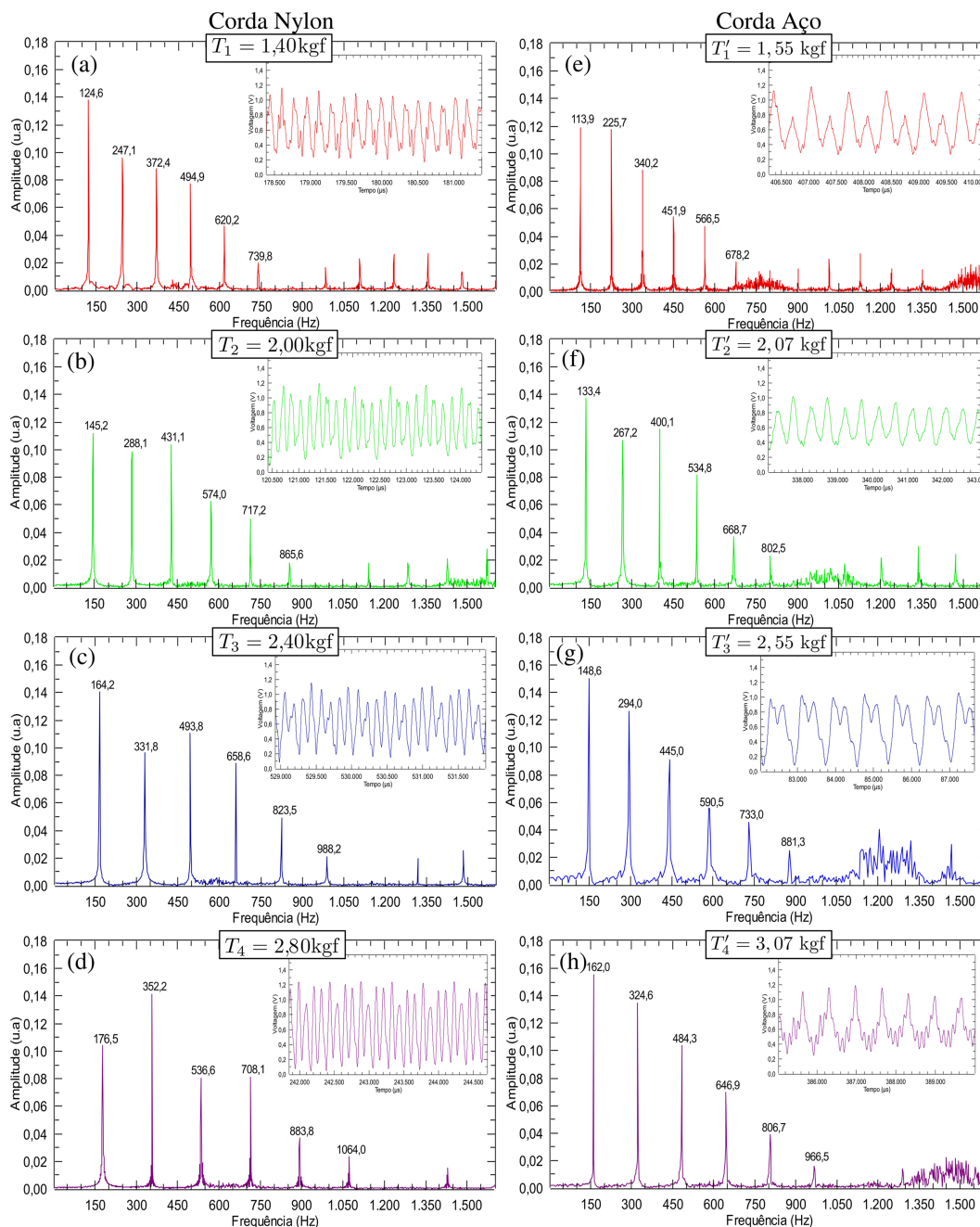


Figura 10: Espectro de frequência e suas respectivas ondas (*insert*) para as cordas de nylon (a)-(d) e aço(e)-(h) para as tensões indicadas em cada gráfico.

É importante destacar que a construção das formas de ondas apresentadas na Fig. 10 é realizada selecionando os dados na porta serial de forma a manter apenas os valores contidos em *buffer* que correspondem à medição. Desse modo, eliminamos os pontos correspondentes aos intervalos de tempo onde era efetuada a transmissão dos valores do Arduino para o PC, evitando os problemas descritos anteriormente.

Os valores das frequências dos picos subsequentes para todas as medidas estão de acordo com a equação 11, isto é, as frequências dos demais modos de vibração são múltiplos inteiros da frequência fundamental. Outro fato

a ser evidenciado é que quando a tensão na corda é aumentada, a frequência do modo fundamental também aumenta e, conseqüentemente, a frequência dos demais modos, como previsto pela equação (11).

Como forma de validar os resultados obtidos para os espectros sonoros, levando em consideração as implementações de programação propostas nesse trabalho, determinaremos tanto as velocidades de propagação das ondas como as densidades lineares a partir dos dados da figura 10. Primeiramente, levantamos os gráficos da frequência em termos dos modos normais de vibração para todas as forças aplicadas tanto na corda de nylon

como na de aço. Esses resultados são apresentados na figura 11.

Realizamos um ajuste linear (retas coloridas na figura 11) para o conjunto de pontos de cada força à qual as cordas foram submetidas. Segundo a equação 11, a inclinação de cada reta é dada por $v/2L$, sendo v a velocidade da onda propagante na corda para cada valor de tensão e $L = 66$ cm é o comprimento da corda. Dessa forma, obtemos todos os valores de velocidade para as duas cordas a partir dos espectros de frequência e comparamos essas velocidades com os valores obtidos anteriormente utilizando o sensor de força. Esses valores são mostrados na tabela 2.

Como última análise, a partir dos valores de velocidades calculadas utilizando o sensor de som, construímos um gráfico (figura 12) das trações nas cordas, aqui convertidas para newtons, em função de v^2 , uma vez que a equação 1 pode ser escrita como: $T = \mu v^2$.

O resultado apresentado na figura 12 nos dá diretamente as densidades lineares de cada corda pela inclinação de cada curva. Os valores obtidos para as cordas de aço e de nylon são 0,658 e 0,469 g/m, respectivamente. Es-

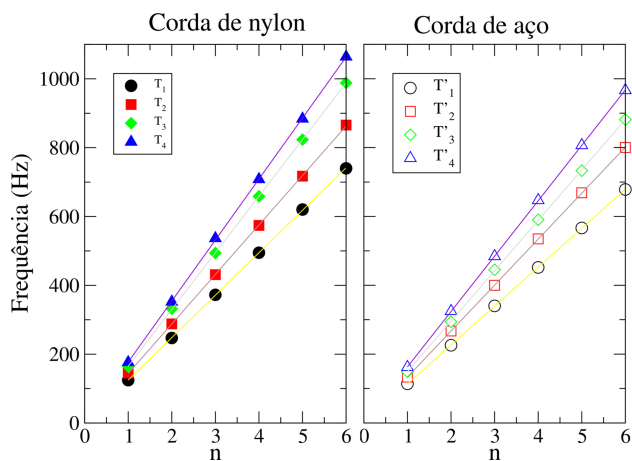


Figura 11: Frequência em função dos modos normais de vibração para corda de nylon e aço submetida a diferentes forças.

Tabela 2: Correspondentes velocidades de propagação da onda na corda obtidas pelo sensor de força e sensor de som para cada valor de tração.

		Sensor força $v = \sqrt{T/\mu}$	Sensor som $v = (2L/n)f_n$
	Tração (kgf)	v (m/s)	v (m/s)
Nylon	1,40	163,9	162,9
	2,00	195,9	189,8
	2,40	214,6	217,2
	2,80	231,8	234,0
Aço	1,55	151,3	149,2
	2,07	174,8	176,3
	2,55	194,1	193,3
	3,07	212,9	212,4

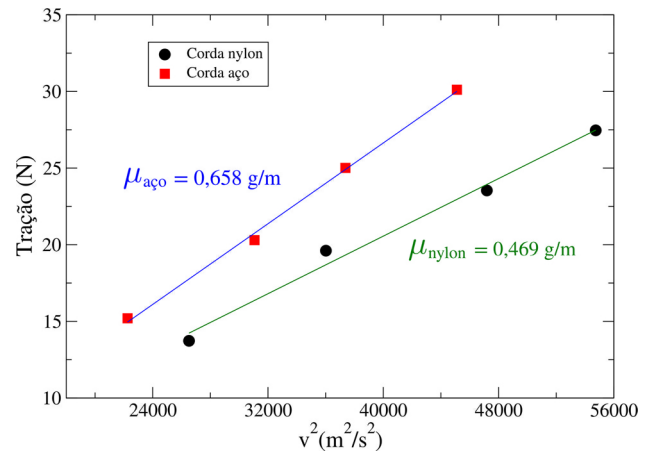


Figura 12: Tração em função de v^2 . As inclinações (retas azul e verde) dão os valores de densidade linear para cada corda.

ses valores diferem pouco daqueles medidos inicialmente com a balança de precisão e régua. Tais resultados reforçam a validade da metodologia empregada na realização deste trabalho mostrando que, para o desenvolvimento de experimentos mais precisos com o Arduino, é importante aprofundar os conhecimentos acerca do funcionamento da placa e da interface de programação (IDE) utilizada.

6. Considerações finais

Desenvolvemos um aparato experimental, semelhante ao braço de um violão, para o estudo de ondas em cordas tensionadas utilizando, como interface de coleta de dados, a plataforma Arduino. Para medidas de tensão mecânica na corda, utilizamos um sensor de força composto por uma célula de carga e o circuito HX711, enquanto a captação do som foi realizada com um sensor composto por um microfone e o circuito amplificador MAX4466. Ambos os dispositivos foram conectados ao Arduino permitindo a visualização da forma de onda, bem como, obtenção das velocidades de propagação das ondas nas cordas em diferentes situações. A partir das medidas da amplitude da onda sonora em função do tempo obtivemos, através de uma análise espectral, os modos normais de vibração da corda que dão origem ao timbre. O sistema foi ajustado de modo a reduzir os impactos da latência na comunicação serial e conversão AD na captura do sinal sonoro. Mostramos que o conhecimento mais aprofundado tanto do *hardware* quanto da configuração da IDE são fatores determinantes para uma coleta de dados mais precisa, principalmente, quando desenvolvemos experimentos onde o tempo é uma variável crítica. Para validar nossos resultados, determinamos a velocidade de propagação das ondas e a densidade linear para as cordas de nylon e aço por meio dos modos normais de vibração (harmônicos) levantados no espectro de frequência. Para o caso da densidade, observamos uma concordância muito boa entre os resultados obtidos com esta técnica e a medida direta da densidade linear (utilizando a rela-

ção massa/comprimento). O baixo custo de montagem e obtenção dos materiais do aparato experimental proposto neste trabalho, assim como o protocolo de medida, contribui para tornar possível o estudo de ondas sonoras emitidas por cordas ou outros dispositivos permitindo a realização de vários experimentos na área de acústica, ou mesmo outros experimentos que envolvam conversão AD de sinais oscilantes como a tensão elétrica em circuitos RLC sem a necessidade de utilização de equipamentos sofisticados de laboratório. Por fim, destacamos a grande capacidade da plataforma Arduino na prototipagem e desenvolvimento de experimentos para investigação de Leis Físicas, reafirmando a relevância do uso desse sistema no ensino de Física.

Referências

- [1] BRASIL. Lei n. 9394, 20 de dezembro de 1996. Brasília, 1996. Disponível em: http://www.planalto.gov.br/ccivil_03/leis/19394.htm.
- [2] M. Fonseca, N.L. Maidana, E. Severino, S. Barros, G. Senhora e V. R. Vanin, *Revista Brasileira de Ensino de Física* **35**, 4503 (2013).
- [3] F.S. Rocha, G.F. Marranghello e M.M. Lucchese, *Caderno Brasileiro de Ensino de Física* **31**, 98 (2014).
- [4] J.C. Santos e A.G. Dickman, *Revista Brasileira de Ensino de Física* **41**, e20180161 (2019).
- [5] <https://www.arduino.cc/en/guide/introduction>, acessado em 03/04/2019.
- [6] M.A. Cavalcante, C.M. Rodrigues e L. Pontes, *Caderno Brasileiro de Ensino de Física* **30**, 579 (2013).
- [7] M.A. Cavalcante, R. Peçanha e A.C. Teixeira, *Revista Brasileira de Ensino de Física* **35**, 3502 (2013).
- [8] M. Varanis, A.L. Silva e A.G. Mereles, *Revista Brasileira de Ensino de Física* **40**, e1304 (2018).
- [9] A. Tunyagi, K. Kandrai, Z. Fülöp, K. Kapusi e A. Simon, *Physics Education* **53**, 035028 (2018).
- [10] A. Hilberer, G. Laurent, A. Lorin, A. Partir, J. Bobroff, F. Bouquet, C. Even, J.M. Fischbach, C.A. Marrache-Kikuchi, M. Monteverde et al., *Papers in Physics* **10**, 100007 (2018).
- [11] J. M. Cardoso e M. Zannin, *Revista Brasileira de Ensino de Física* **41**, e20190028 (2019).
- [12] F.S. Oliveira, L.B. Calheiro, D.F. Bozano, N.C.G. Errobidart, M.I.A. Jardim, D.D. Reis e A.M.B. Goncalves, *Physics Education* **54**, 065012 (2019).
- [13] M.A. Cavalcante, C.R.C. Tavolaro e E. Molisani, *Revista Brasileira de Ensino de Física* **33**, 4503 (2011).
- [14] G. Dionisio e L.E.S. Spalding, *Revista Brasileira de Ensino de Física* **39**, e1501 (2017).
- [15] A.A. Moya, *Physics Education* **54**, 015005 (2018).
- [16] K. Atkin, *Physics Education* **53**, 025003 (2018).
- [17] W.P.S. Freitas, C.R. Cena, D.C.B. Alves e A.M.B. Gonçalves, *Physics Education* **53**, 035034 (2018).
- [18] S. Silveira e M. Girardi, *Revista Brasileira de Ensino de Física* **39**, e4502 (2017).
- [19] I.N. Oliveira, J.A.P. Ramos, W.L. Silva, V.D. Chaves e C.A.O. Melo, *Revista Brasileira de Ensino de Física* **42**, e20190105 (2020).
- [20] Datasheet MAX4466, disponível em: <https://pdf1.all-datasheet.com/datasheet-pdf/view/460080/MAX4466.html>, acessado em 30/04/2020.
- [21] Datasheet HX711, disponível em: <https://www.alldatasheet.com/datasheet-pdf/pdf/1132222/AVIA/HX711.html>, acessado em 30/04/2020.
- [22] <https://www.arduino-libraries.info/libraries/hx711-arduino-library>, acessado em 17/08/2019.
- [23] W.D. Callister Jr. e D.G. Rethwisch, *Ciência e engenharia de materiais – uma introdução* (LTC, Rio de Janeiro, 2016).
- [24] Datasheet microcontroller, disponível em: https://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561_datasheet.pdf, acessado em 29/04/2020.
- [25] <https://www.arduino.cc/reference/pt/language/functions/external-interrupts/attachinterrupt/>, acessado em 10/09/2019.