



.....
Cícero Jailton de Moraes Souza^{1#} , Ibson José Maciel Leite¹, Mário Antonio Alves Monteiro¹

¹Instituto Federal de Pernambuco, campus Pesqueira, Pesqueira, PE, Brasil.
.....

Palavras-chave

Arduino
Excel
aceleração constante
aquisição automática de dados

Resumo

A aquisição e a análise automáticas de dados em experimentos didáticos, com o uso de ferramentas como o Arduino e planilhas eletrônicas, têm sido cada vez mais exploradas em diferentes áreas do ensino de física. Essa abordagem permite análises experimentais quantitativas com precisão, facilita a visualização e a análise dos dados, desenvolve habilidades de resolução de problemas e familiariza os alunos com tecnologias acessíveis. Neste trabalho, aplicamos essas ferramentas a um experimento de movimento uniformemente acelerado, no qual foi analisado o movimento de uma esfera de aço descendo e rolando sobre um trilho de alumínio inclinado. Para a obtenção e análise dos dados, foram utilizadas uma placa Arduino Uno, o *streamer* de dados do Microsoft Excel e sete sensores HW-201. Os sensores foram fixados ao trilho e conectados ao Arduino, permitindo a detecção da passagem da esfera e registrando os instantes de tempo em que isso ocorreu. A integração do Excel com o Arduino possibilitou a monitoração e a visualização em tempo real dos dados, além da criação de tabelas e gráficos dinâmicos para uma análise mais detalhada dos resultados.

1. Introdução

A integração de tecnologias no ambiente de aprendizagem tem se tornado uma prioridade crescente na instrumentação para o ensino, visando preparar os estudantes para um mundo cada vez mais digital e conectado [1]. Nesse contexto, os microcontroladores têm despontado como ferramentas promissoras, capazes de estimular a criatividade, o pensamento computacional e a resolução de pro-

blemas de maneira prática e envolvente.

O uso de microcontroladores vem se tornando cada vez mais frequente no ensino de física, devido às diversas vantagens que oferecem. Eles possibilitam a construção de experimentos interativos e personalizados de maneira

O uso de microcontroladores possibilita a construção de experimentos interativos e personalizados de maneira acessível

de maneira acessível, permitindo que os alunos participem ativamente do processo de criação e desenvolvimento dos experimentos. Além disso, o uso de microcontroladores permite a coleta e a análise de da-

Autor de correspondência. E-mail: cicero.souza@pesqueira.ifpe.edu.br

Este é um artigo de acesso livre sob licença Creative Commons



<https://creativecommons.org/licenses/by-nc-sa/3.0/>

Copyright © 2024, Copyright by Sociedade Brasileira de Física. Printed in Brazil.

dos em tempo real, o que possibilita uma interatividade com o experimento e maior precisão na obtenção de resultados. Segundo Platt [1], um microcontrolador é:

Um chip que funciona como um computador. Você escreve um programa composto de instruções que o microcontrolador possa entender e então as copia para alguma memória no chip. (...) Um microcontrolador é versátil e potente, mas ele é mais adequado em algumas situações que em outras. Ele é ideal para aplicar operações lógicas do tipo 'se isto acontecer faça aquilo, mas se aquilo outro acontecer faça outra coisa'.

O Arduino é um exemplo de plataforma eletrônica de código aberto. Ele consiste em uma placa de circuito contendo um *chip* microcontrolador, podendo este ser de diferentes empresas, de modo que existem vários tipos de placas no mercado [1]. Trata-se de uma ferramenta versátil e acessível, que permite a integração de sensores para realizar medições e controlar dispositivos. Quando aplicado ao ensino de física, o uso do Arduino possibilita a construção de experimentos de baixo custo que permitam a coleta e análise de dados automatizada, sendo cada vez mais comum encontrar experimentos didáticos de física que utilizam essa ferramenta [2, 3].

Uma nova ferramenta que vem ganhando cada vez mais espaço no ensino de física é o *streamer* de dados, uma funcionalidade da planilha eletrônica Excel, um software desenvolvido pela Microsoft que faz parte do pacote Office 365 [4]. O *streamer* de dados do Excel proporciona uma abordagem inovadora e interativa para a coleta e a análise de dados em experimentos didáticos. Ao combinar as funcionalidades do Excel com sensores, microcontroladores e dispositivos de medição em geral, professores e alunos podem explorar conceitos físicos de maneira prática e envolvente. Com essa ferramenta, é possível registrar medidas como temperatura, pressão, aceleração e outras grandezas físicas relevantes de maneira precisa e praticamente em tempo real. Embora seja possível copiar os dados e as informações do *serial monitor* do Arduino e colá-los em uma planilha do Excel, ao se utilizar o *streamer* de dados, todo o processo de coleta torna-se mais simples e automatizado. Além disso, os recursos de visualização de dados do Excel, como gráficos e tabelas dinâmicas, podem auxiliar na interpretação dos resultados e na identificação de padrões.

A coleta e a análise automáticas de dados em experimentos de física com foco educacional é uma metodologia que vem ganhando destaque nos últimos anos. Com o uso de sensores, é possível medir grandezas físicas e visualizar seus resultados por meio de gráficos. Particularmente, novos trabalhos têm surgido na área de ensino de física utilizando o Arduino e o *streamer* de

dados do Excel em experimentos didáticos. O uso dessas ferramentas tem ocorrido nas mais diversas áreas do ensino de física, como cinemática [5, 6], calorimetria [7, 8], oscilações e ondas [9, 10] e circuitos elétricos [11], o que demonstra o potencial dessas tecnologias.

Neste trabalho, apresentamos uma aplicação do Arduino junto ao *Streamer de Dados* do Excel para o estudo de um experimento de movimento uniformemente acelerado, no qual analisamos o movimento de uma esfera de aço descendo e rolando sobre um trilho de alumínio inclinado.

2. Conexões, programa e montagem do experimento

Neste trabalho, utilizamos a placa Arduino Uno e o Excel, versão Microsoft 365. Além disso, a montagem do experimento é composta pelos itens mostrados na Fig. 1, que são: uma esfera de aço de 18 mm de diâmetro, um trilho de alumínio de 2 m de comprimento, 7 sensores HW-201 e um *buzzer* de 5 V. Também usamos um suporte de laboratório para suspender o trilho, alguns prendedores de plástico e fita adesiva para fixar os sensores ao trilho, fios e uma placa *protoboard* para realizar as conexões entre os sensores e o Arduino.

O sensor HW-201 é um módulo eletrônico que utiliza radiação de comprimento de onda no infravermelho para detectar obstáculos à sua frente. Ele é normalmente utilizado em robôs seguidores de linha e outros projetos de robótica que precisam realizar a detecção de objetos ou determinar a velocidade de rotação de motores [12]. O sensor funciona emitindo um sinal infravermelho e detectando a reflexão desse sinal em objetos próximos. Ele possui um LED emissor e um receptor (fototransistor) que detecta a reflexão do sinal. Quando o sinal é refletido de volta para o sensor, ele é convertido em um sinal elétrico que pode ser lido por um microcontrolador, como o Arduino. O HW-201 conta ainda com um circuito inte-

A integração entre o Arduino e o streamer de dados do Excel transforma os experimentos didáticos, proporcionando coleta automatizada e análise de dados em tempo real



Figura 1 - Alguns dos materiais usados no experimento, tais como: esfera de aço, trilho de alumínio, sensores HW-201 e um buzzer 5 V.

grado LM393, que é um comparador de tensão que facilita sua conexão com o Arduino [13]. O módulo pode ser alimentado com uma fonte de 3-5 V DC e possui uma distância de detecção que pode variar de 2 a 30 cm, dependendo do ajuste de sensibilidade [12].

O *buzzer* ativo 5 V é um componente eletrônico que possui um circuito oscilador embutido que, quando opera sob uma tensão de 5 V de corrente contínua, emite um sinal sonoro de frequência única. Costuma ser utilizado em projetos eletrônicos de alarmes, sistemas de sinalização, impressoras, projetos de robótica etc. [14]. Uma maneira de conectar o *buzzer* ativo 5V ao Arduino é ligando o pino positivo e o pino negativo ao pino digital e ao pino GND do Arduino, respectivamente.

O esquema da montagem do experimento é apresentado na Fig. 2. O trilho de alumínio ficou com uma das extremidades apoiada na superfície da mesa, enquanto o corpo do trilho foi suspenso pelo suporte, de modo que o ângulo de inclinação podia ser ajustado simplesmente movimentando-se o suporte. O ângulo de inclinação foi determinado a partir do comprimento do trilho e da altura a que um dos seus lados foi suspenso, o que forneceu uma incerteza de $\pm 0,1^\circ$. Os sensores HW-201 foram fixados ao trilho com prendedores de plástico e fita adesiva, sendo colocados à mesma distância um do outro (30,0 cm \pm 0,5 cm) e em uma posição tal que a esfera, ao rolar pelo trilho, passasse à frente do sensor, de modo que ele pudesse detectar sua passagem.

O diagrama das conexões dos sensores HW-201 com o Arduino é apresentado na Fig. 3. Nesse experimento, foram utilizados 7 sensores, sendo que cada um teve seu pino VCC ligado ao pino de 5V do Arduino e o pino GND ligado ao GND do Arduino. Para outras aplicações, a quantidade de sensores pode ser alterada, dependendo da necessidade do experimentador. Os

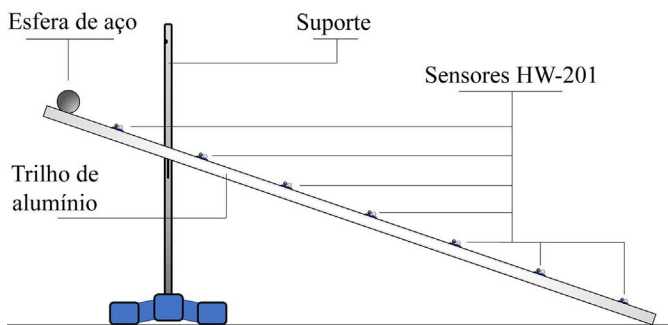


Figura 2 - Esquema da montagem do experimento. O trilho de alumínio foi elevado por um suporte e a esfera de aço desce pelo trilho passando à frente dos sensores HW-201, que detectam sua passagem.

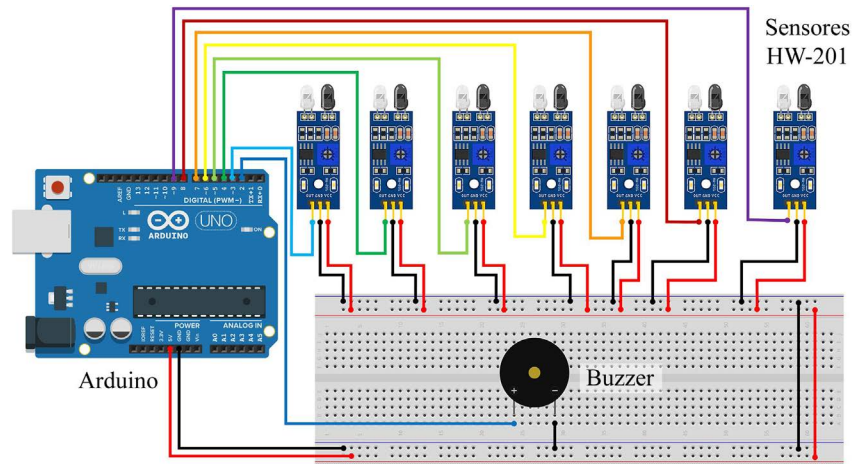


Figura 3 - Diagrama de conexões dos sensores e do *buzzer* ao Arduino. Todos os sensores HW-201 são conectados em paralelo aos pinos de 5 V e GND do Arduino; além disso, o pino OUT de cada sensor é conectado a um pino digital. O Buzzer Ativo 5V é conectado ao GND e a uma das entradas digitais do Arduino.

pinos de saída (OUT) dos sensores foram conectados aos pinos digitais, do pino 3 ao pino 9 do Arduino. As pernas positiva e negativa do *buzzer* foram conectadas, respectivamente, à entrada digital 2 e ao GND do Arduino, que, por sua vez, foi conectado a um notebook com o pacote Office 365 instalado.

Após todos os sensores serem colocados em suas posições e as conexões terem sido realizadas de maneira adequada, é possível iniciar a coleta de dados. Com o Arduino conectado ao *notebook* e com a porta serial e a placa corretas selecionadas no IDE (*integrated development environment* - ambiente de desenvolvimento integrado) do Arduino, o passo seguinte é carregar o programa (ver Apêndice) na placa Arduino.

O algoritmo funciona da seguinte maneira: toda vez que a esfera metálica passa na frente de um dos sensores, o *buzzer* emite um sinal sonoro e o instante de tempo em que isso ocorreu é registrado pelo Arduino, usando a função *micros()*, que registra o instante de tempo em microssegundos. Esses tempos são divididos por 1000, para que o tempo seja transformado em milissegundos, e são impressos usando a função *println()*. Quando a esfera passa pelo segundo sensor, ou por qualquer outro sensor, o mesmo processo acontece: o *buzzer* emite um som e o instante de tempo é registrado. A emissão do som pelo *buzzer* tem apenas fins didáticos, para mostrar que o sensor detectou a passagem da esfera, não interferindo na medida realizada.

É importante destacar que existe um tempo mínimo entre uma detecção e outra devido ao *delay* de 100 ms no programa. Esse intervalo de tempo é necessário para que o programa não interprete a passagem por um mesmo sensor como sendo duas passagens. Dessa maneira, a esfera não pode estar muito lenta, de modo que a passagem por um sensor demore mais que 100 ms (o que ocasionaria duas medidas de tempo em

um mesmo sensor), nem pode estar muito rápida, de modo que a passagem por dois sensores consecutivos demore menos que 100 ms (situação na qual o tempo de um dos sensores não seria registrado). Nos testes realizados, foram feitas medidas cujos ângulos de inclinação do trilho variaram de 3,0° a 20,0°, e em nenhuma das medidas houve problemas com o *delay* do programa. Para ângulos maiores ou menores que esses, caso surja algum problema desse tipo, basta apenas ajustar o tempo de *delay* no programa, de acordo com a necessidade do experimentador.

Com o Arduino conectado e com o programa carregado, é necessário conectá-lo ao Excel. Para isso, selecionamos a aba do Excel *streamer* de dados e clicamos em “Conectar um Dispositivo”, onde de imediato já aparece a opção de selecionar a placa Arduino vinculada ao notebook. Quando a conexão acontece, quatro novas planilhas são criadas automaticamente. Essas planilhas servem para fazer a comunicação de entrada ou saída de dados com o Arduino e também para configurar essa entrada ou saída. Após a conexão ser concluída, clica-se em “Data de Início” e a captura de dados se inicia. As medidas de tempo realizadas pelo Arduino são então enviadas para o Excel e automaticamente começam a preencher a nova planilha chamada “Dados em”, como mostrado na Fig. 4.

Os dados enviados para o Excel são aqueles que são impressos no Arduino com a função *println()*. No caso da Fig. 4, são mostrados a coluna TIME, com os instantes de tempo registrados pelo Excel, e mais 10 canais de entrada de dados, de modo que é possível imprimir o valor de até 10 variáveis, sendo que os instantes de tempo medidos em milissegundos são impressos no canal 1 (canal CH1, na coluna B, linhas 8 a 22). Esses valores de tempo do Arduino começam a ser contabilizados a partir do momento em que o programa é carregado na placa, ou, se já estiver carregado previamente, o tempo começa a contar a partir do momento em que a placa é conectada ao notebook.

O número de canais e o número de linhas de dados podem ser editados na planilha “Configurações”. Cada vez que a esfera passa por um dos sensores, um novo dado de tempo é enviado para o Excel, que atualiza a célula B5 e as células B8 a B22. Com essa configuração, portanto, podemos visualizar os dados referentes à captura de 15 valores de tempo, a partir dos quais, para novas medidas, os tempos mais antigos são descartados e não aparecem mais na planilha. Como na nossa aplicação utilizamos apenas 7 sensores de detecção, analisamos apenas os 7 instantes de tempo medidos mais recentes.

A Fig. 5 apresenta o experimento montado.

Dados Em (De Arduino Uno (COM11))										
Os dados provenientes da fonte de dados atual serão exibidos abaixo conforme sejam recebidos.										
Dados Atuais										
TIME	CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8	CH9	CH10
17:28:09,22	2352959									
Dados Históricos										
TIME	CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8	CH9	CH10
17:28:09,22	2352959									
17:28:08,88	2352623									
17:28:08,50	2352237									
17:28:08,06	2351803									
17:28:07,52	2351269									
17:28:06,85	2350598									
17:28:05,44	2349179									
17:28:05,34	2349079									
17:28:05,24	2348978									
17:28:05,13	2348878									
17:28:05,04	2348777									
17:28:04,93	2348677									
17:28:04,83	2348577									
17:28:04,73	2348476									
17:28:04,63	2348376									

Figura 4 - Planilha de entrada de dados do Excel já preenchida com valores medidos.

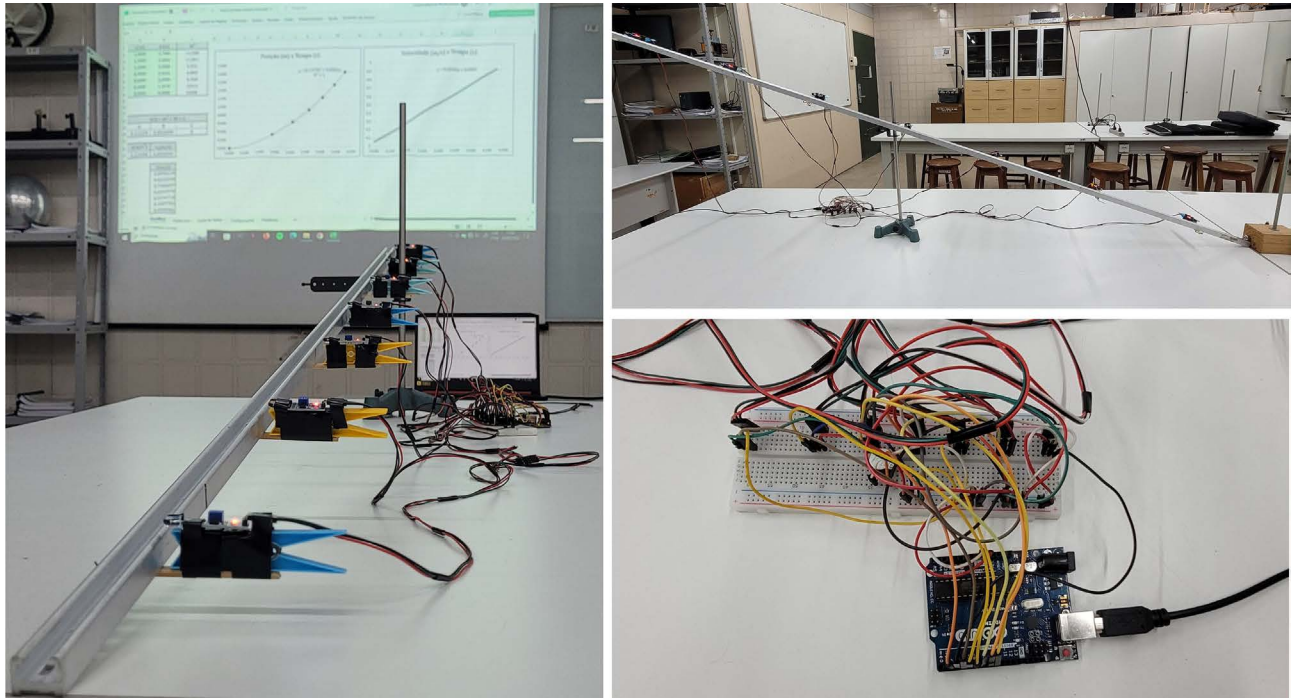


Figura 5 - Montagem do experimento. Sete sensores HW-201, fixados no trilho de alumínio inclinado, foram conectados a uma *protoboard* e ao Arduino, que por sua vez foi conectado a um computador com o Microsoft Office 365 instalado. A planilha do Excel pode ser projetada para uma melhor visualização dos dados.

3. Análise dos resultados

Soltando-se a esfera da extremidade mais elevada do trilho, ela percorrerá uma distância de $15,0 \pm 0,5$ cm até o primeiro sensor. Nesse experimento, consideramos a posição do primeiro sensor como a posição inicial $s_0 = 0,0 \pm 0,5$ cm, de modo que os outros sensores estavam nas posições $s_1 = 30,0 \pm 0,5$ cm, $s_2 = 60,0 \pm 0,5$ cm; $s_3 = 90,0 \pm 0,5$ cm, $s_4 = 120,0 \pm 0,5$ cm, $s_5 = 150,0 \pm 0,5$ cm e $s_6 = 180,0 \pm 0,5$ cm. As incertezas das posições dos sensores foram estimadas com base nas fixações dos sensores, que podiam ter uma variação de $\pm 0,5$ cm. Quando a esfera passa pelo primeiro sensor, este registra o instante de tempo t_0 e, ao passar pelos outros sensores, são registrados os respectivos instantes de tempo t_1, t_2, \dots, t_6 . De acordo com a Fig. 4, portanto, esses instantes de tempo, contados a partir da conexão do Arduino, seriam iguais aos valores das células $t_0 = B14 = 2349179$ ms = 39,153 min, $t_1 = B13 = 2350598$ ms = 39,177 min, ..., $t_6 = B8 = 2352959$ ms = 39,216 min. Por motivos didáticos, optamos por utilizar não os valores dos tempos das células, mas sim os intervalos em segundos $\Delta t_i = (t_i - t_0)/1000$, de modo que o tempo inicial usado é $\Delta t_0 = (t_0 - t_0)/1000 = 0,000$ s, e os outros tempos são $\Delta t_1 = (t_1 - t_0)/1000, \dots, \Delta t_6 = (t_6 - t_0)/1000$. Assim, para um ângulo de inclinação $\theta = 8,0^\circ$, foram obtidos os seguintes resultados mostrados na Tabela 1.

Esses valores foram impressos em outra planilha do Excel, onde a análise de dados em tempo real foi realizada. Assim, a partir dos dados da Tabela 1, o Excel

realiza automaticamente os cálculos de velocidade e de aceleração do movimento, bem como os gráficos de posição e velocidade em função do tempo.

Para obter uma melhor visualização da relação entre posição e tempo, criamos um gráfico de dispersão com linha de tendência polinomial de ordem 2, usando os dados de posição S em metros e Δt em segundos da Tabela 1. O resultado é mostrado na Fig. 6, juntamente com os dados obtidos para outros três ângulos.

Pelo ajuste das curvas dos gráficos da Fig. 6, podemos observar que uma linha de tendência quadrática entre a posição e o tempo se ajusta muito bem aos dados medidos, o que é o esperado para um movimento com aceleração constante. Se compararmos as curvas dos ajustes quadráticos que aparecem na Fig. 6 com a função do movimento com aceleração constante, dada pela Eq. (1),

Tabela 1: Valores de posição, tempo e tempo ao quadrado obtidos para um ângulo $\theta = 8,0^\circ$.

s (m)	Δt (s)	Δt^2 (s ²)
0,00	0,000	0,000
0,30	0,607	0,368
0,60	0,947	0,897
0,90	1,217	1,481
1,20	1,441	2,076
1,50	1,648	2,716
1,80	1,829	3,345

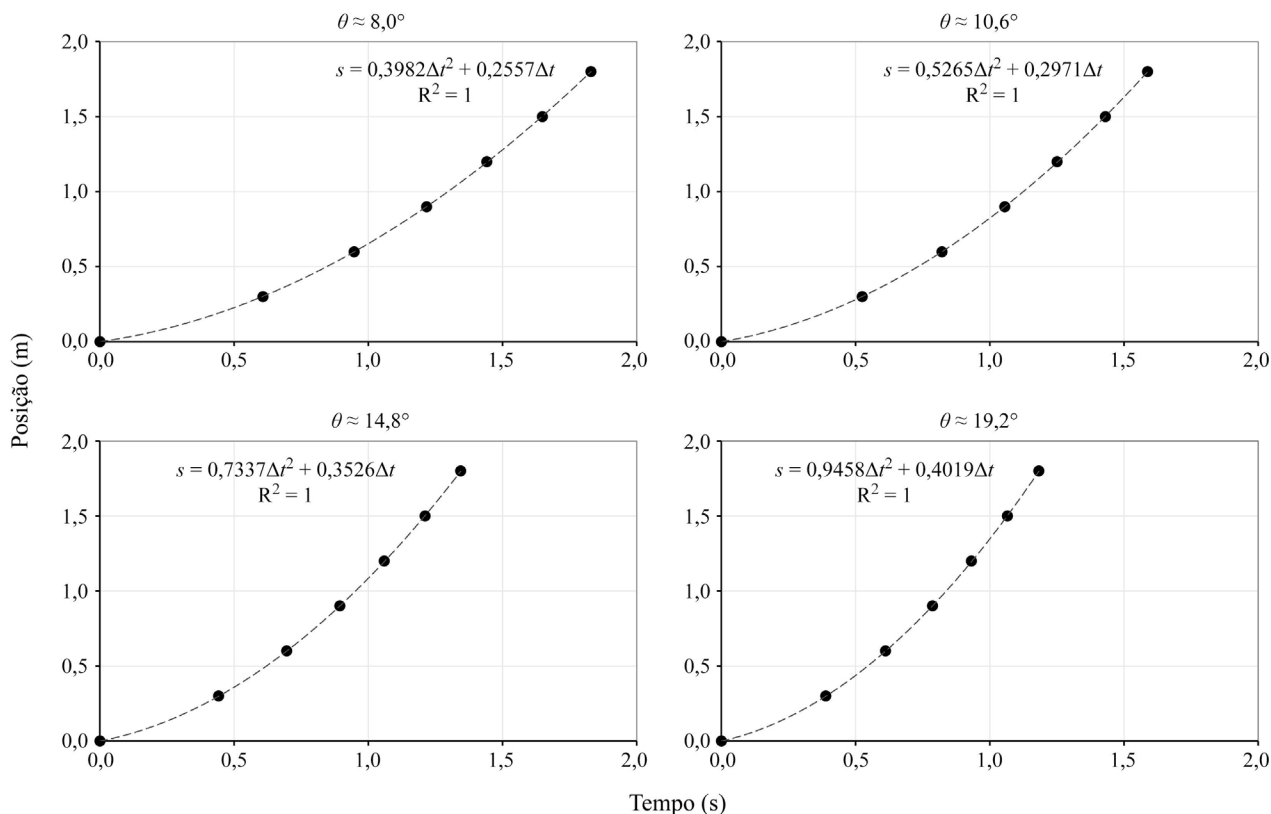


Figura 6 - Gráficos da posição em função do tempo para o movimento da esfera rolando pelo trilho inclinado em diferentes ângulos.

$$s(t) = s_0 + v_0 t + \frac{1}{2} a t^2. \quad (1)$$

Para os dados da Tabela 1, por exemplo, é possível determinar para o ângulo $\theta = 8,0^\circ$ que $s_0 = 0,00$ m, $v_0 = 0,2557$ m/s e $a = 0,7964$ m/s². Assim, a função da posição para este movimento é:

$$s(t) = 0,2557t + 0,3982t^2. \quad (2)$$

Analisando a relação entre as acelerações e os ângulos de inclinação do trilho conforme os gráficos apresentados na Fig. 6, podemos ver, como esperado, que para ângulos maiores as acelerações também são maiores, assim como as velocidades iniciais da esfera ao passar pelo primeiro sensor.

O R^2 (coeficiente de determinação) é uma medida estatística utilizada para avaliar a qualidade de ajuste de um modelo de regressão aos dados observados. No contexto de um gráfico feito no Excel, o R^2 é uma medida de quão bem uma linha de tendência se ajusta aos pontos da curva. O R^2 varia entre 0 e 1, sendo que quanto mais próximo de 1 for o valor do R^2 , melhor o ajuste do modelo aos dados. No caso dos resultados de nosso experimento que estão sendo mostrados nos gráficos da Fig. 6, $R^2 = 1$ para todos os ângulos, o que indica um bom ajuste quadrático da curva.

Podemos ver que a velocidade v_0 no primeiro sensor é diferente de zero em todos os casos, pois ao passar pela posição em que o primeiro sensor foi colocado a esfera já está se movimentando. Uma vez tendo determinado a velocidade inicial e a aceleração, podemos fazer um gráfico da velocidade em função do tempo, utilizando os dados extraídos da Eq. (2) e considerando que, para o movimento uniformemente acelerado,

$$v(t) = v_0 + at. \quad (3)$$

A Fig. 7 apresenta os gráficos da velocidade em função do tempo para os mesmos quatro ângulos apresentados na Fig. 6.

Mantendo a mesma escala nos eixos horizontal e vertical, é possível visualizar que o gráfico com a reta de maior inclinação é, como esperado, o que tem a maior aceleração e o que possui o maior ângulo de inclinação do trilho em relação ao eixo horizontal.

O ponto diferencial de nosso experimento é que tanto os dados da Tabela 1 quanto os gráficos das Figs. 6 e 7 são atualizados em tempo real e de maneira automática, conforme a esfera vai passando pelos sensores. Para que isso ocorra, basta vincular um gráfico de dispersão com linha de tendência no Excel aos dados de posição obtidos da maneira discutida no início desta seção. Fazendo-se isso, o gráfico da Fig. 6 será

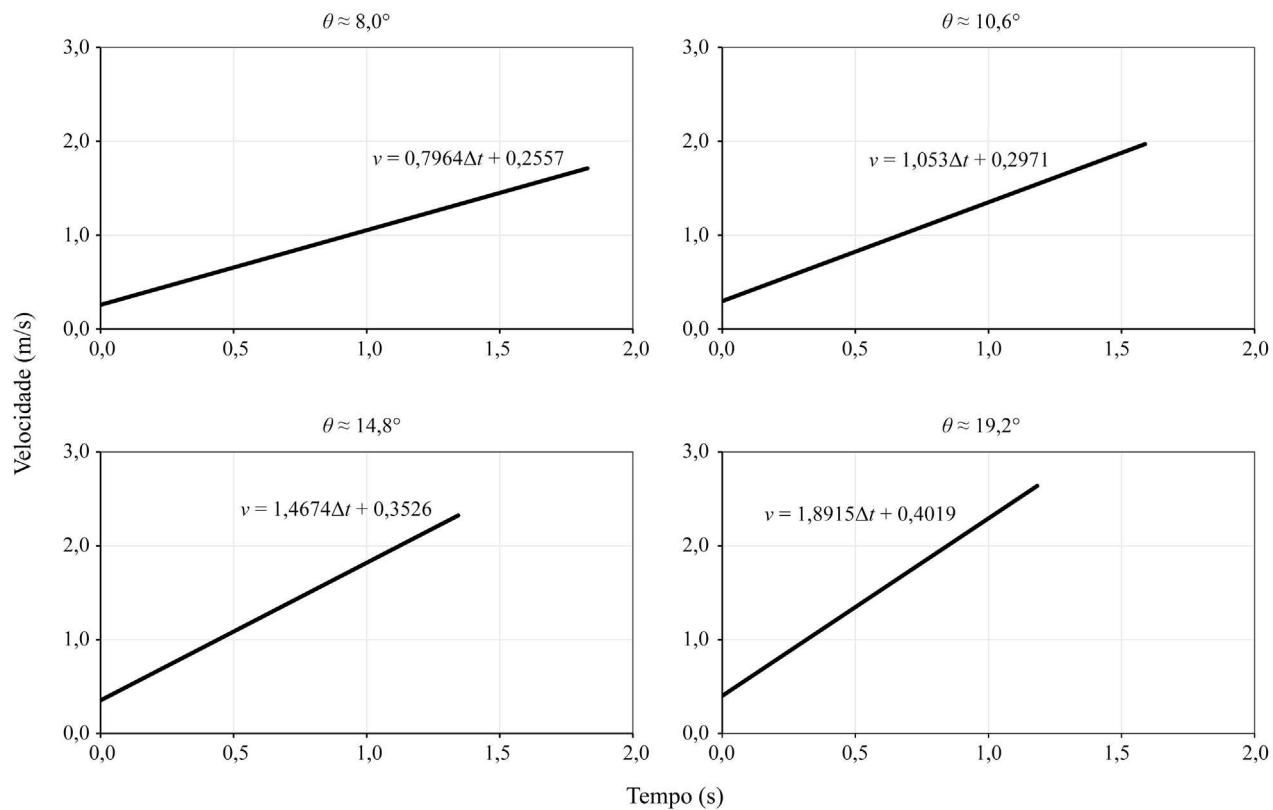


Figura 7 - Gráficos da velocidade a partir dos dados obtidos da aceleração e velocidade inicial.

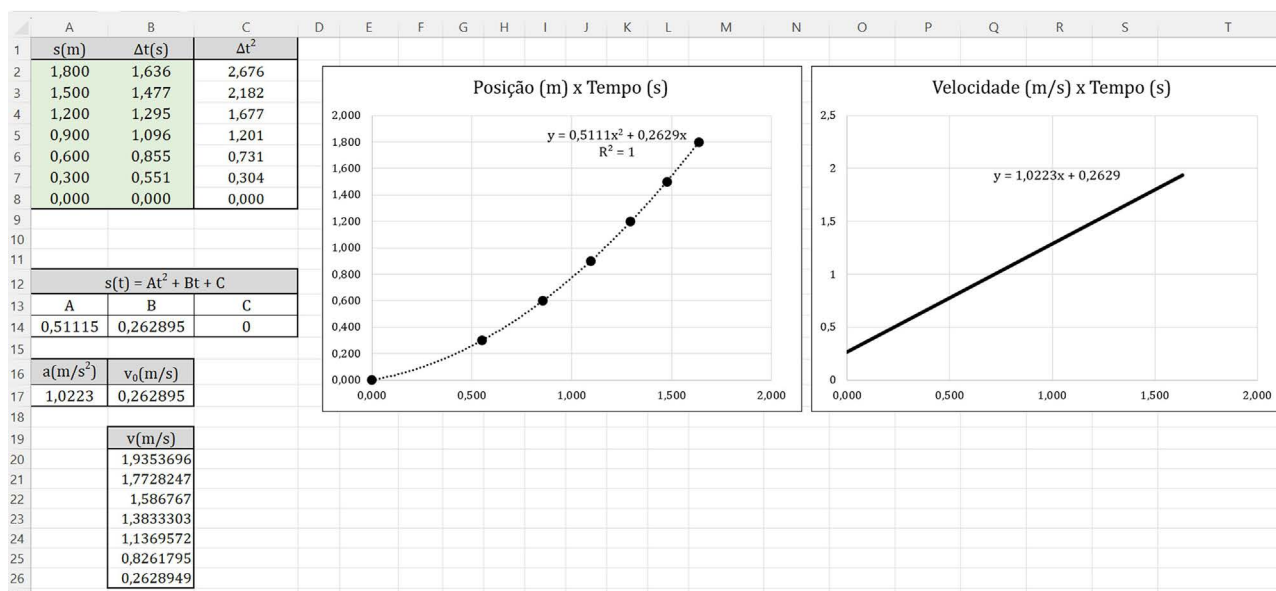


Figura 8 - Planilha com a análise em tempo real dos dados obtidos no experimento.

gerado automaticamente à medida que os sensores detectarem a esfera. A partir desse gráfico, será possível visualizar o ajuste quadrático da linha de tendência, de modo a obter a Eq. (2). No entanto, para criar o gráfico da velocidade correspondente é necessário extrair, também automaticamente, os dados de aceleração e velocidade inicial da esfera. Esses dados não

podem ser extraídos de maneira automatizada do gráfico de dispersão, mas é possível utilizar uma das funções do Excel para isso.

Nas versões mais recentes do Microsoft Excel, a função PROJ.LIN é utilizada para realizar uma regressão linear e prever valores futuros com base em uma série de dados existente. Essa função usa os valores

conhecidos de x e y para calcular a inclinação (coeficiente angular) e a interceptação (coeficiente linear) da linha de regressão, usando o método dos mínimos quadrados [16]. Em seguida, ela utiliza esses coeficientes para estimar o valor de y correspondente ao novo valor de x . Assim, como a posição é uma função quadrática do tempo, do tipo

$$s(t) = At^2 + Bt + C, \quad (4)$$

é possível determinar os coeficientes A , B e C usando PROJ.LIN. Nesse caso, os valores do eixo y correspondem aos valores das posições dos sensores; então, para se usar a função deve-se definir os valores y como os dados da coluna de posição. Já no caso dos valores x , uma vez que a relação entre y e x é quadrática, deve-se definir como valores de x duas colunas: a coluna com os dados do tempo e a coluna com esses tempos elevados ao quadrado. Por exemplo, estando os dados da posição nas células A2:A8,¹ os dados de instantes de tempo nas células B2:B8 e esses tempos elevados ao quadrado nas células C2:C8, podemos fazer uma regressão com PROJ.LIN e descobrir os valores dos coeficientes. Nesse exemplo, deve-se digitar o seguinte código em uma célula vazia:

= PROJ.LIN(A2 : A8; B2 : C8; FALSO)

O termo “FALSO” deve ser colocado para explicitar que a curva deve interceptar os eixos no tempo zero e na posição zero, ou seja, estamos definindo que a contagem do movimento se iniciou quando a esfera passou pelo primeiro sensor na posição $s_0 = 0$ m e no instante $t_0 = 0$ s, de modo que o coeficiente C deverá ser zero. Caso não se queira definir isso, basta substituir “FALSO” por “VERDADEIRO” e o Excel calculará o valor de C que melhor se ajusta aos dados.

Assim, os valores dos coeficientes da função quadrática são calculados de maneira automática. Isso significa que, ao se fazerem novas medidas, os valores da coluna de tempo e da coluna de tempo elevado ao quadrado vão se atualizando, o que também gera imediatamente uma atualização na regressão da curva e nos coeficientes calculados. Como esses coeficientes estão relacionados à aceleração a e à velocidade inicial v_0 , esses valores também mudam automaticamente, assim como os gráficos de posição em função do tempo e de velocidade em função do tempo. A Fig. 8 mostra como ficou a planilha do Excel e os gráficos com a cap-

tura e a análise dos dados em tempo real para uma determinada medida.

Portanto, toda vez que a esfera passar por um sensor, um novo instante de tempo é medido e substituído o valor de tempo mais antigo da planilha. Essa alteração gera também uma variação nos coeficientes da curva ajustada e nos gráficos produzidos. Os valores corretos dos coeficientes e os gráficos finalizados só são gerados quando a esfera tiver passado pelos sete sensores.

4. Considerações finais

Os microcontroladores, em especial o Arduino, têm se destacado como ferramentas promissoras no ensino de física devido à sua versatilidade, baixo custo e fácil acesso. Eles possibilitam que professores criem experimentos mais interativos e personalizados, mais adequados às suas necessidades e às de seus estudantes. O *streamer* de dados do Excel, por sua vez, apresenta-se como outro importante instrumento no ensino experimental de física, proporcionando uma abordagem inovadora e interativa para a coleta e a análise de dados. Combinado com sensores, microcontroladores e dispositivos de medição, ele permite a visualização e a análise dos dados em tempo real, eliminando a necessidade de anotações manuais e facilitando a interpretação dos resultados por meio de gráficos e tabelas dinâmicas.

Neste artigo, exploramos a aplicação do Arduino em conjunto com o *streamer* de dados do Excel em um experimento de física, especificamente no estudo do movimento uniformemente acelerado de uma esfera em um trilho inclinado. Demonstramos que por meio da integração entre essas duas ferramentas é possível obter resultados precisos, visualizar os dados imediatamente e realizar análises mais detalhadas dos fenômenos físicos estudados.

Consideramos que a utilização do Arduino em conjunto com o Excel para aquisição e análise automática de dados é uma abordagem promissora no ensino de física, pois proporciona uma experiência prática e envolvente para os alunos, o que pode permitir uma melhor compreensão dos conceitos físicos estudados. Essa integração tecnológica contribui para o desenvolvimento de habilidades essenciais e prepara os estudantes para enfrentar os desafios de um mundo cada vez mais tecnológico e digital.

Recebido em: 15 de Novembro de 2023

Aceito em: 2 de Agosto de 2024

Nota

¹A referência “A2:A8” indica um intervalo de células que começa na coluna A, na linha 2, e vai até a coluna A, na linha 8.

Referências

- [1] C. Platt, *Eletrônica para Makers: Um Manual Prático Para o Novo Enthusiasta de Eletrônica* (Novatec, São Paulo, 2016).
- [2] C.J.M. Souza, L.A. Silva, I.J.M. Leite, M.A.A. Monteiro, *Rev. Bras. Ens. Fis.* **43**, e20210125 (2021). [doi](#)
- [3] P.D.M. Pereira, M.S.Silva, *Rev. Bras. Ens. Fis.* **43**, e20210189 (2021). [doi](#)
- [4] MICROSOFT, *O que é o Streamer de Dados?* MICROSOFT, 2023. Disponível em <https://support.microsoft.com/pt-br/office/o-que-%C3%A9-o-streamer-de-dados-1d52ffce-261c-4d7b-8017-89e8ee2b806f>, acesso em 3 março 2023.
- [5] M.E. Hadi, A.E. Moussaouy, A. Ouariach, R. Essaadaoui, A. Hachmi, e cols., *Phys. Educ.* **55**, 055008 (2020). [doi](#)
- [6] T. Alexandru-Constantin, R. Ileana-Constanța, *IOP Conf. Ser.: Mater. Sci. Eng.* **1256**, 012025 (2022). [doi](#)
- [7] W. Vallejo, C. Diaz-Uribe, C. Fajardo, *Heliyon* **6**, e03591 (2020). [doi](#)
- [8] R. Essaadaoui, A.E. Moussaouy, M.E. Hadi, A. Ouariach, A. Hachmi, e cols., In: *9th Int. Conf. Inf. Educ. Technol.*, Okayama, Japan, 2021, [doi](#).
- [9] A. Teiermayer, *Phys. Educ.* **54**, 065021 (2019). [doi](#)
- [10] F. Vitkóczy, K. Piláth, *Int. J. Innov. Educ. Res.* **8**, 60 (2020). [doi](#)
- [11] M.E. Hadi, A. Ouariach, R. Essaadaoui, A.E. Moussaouy, O. Mommadi, *Phys. Educ.* **56**, 045015 (2021). [doi](#)
- [12] UsinaInfo, *Sensor de Obstáculo Reflexivo Infravermelho HW-201 com Ajuste de Sensibilidade*. Disponível em <https://www.usinainfo.com.br/sensor-de-linha-arduino/sensor-de-obstaculo-reflexivo-infravermelho-hw-201-com-ajuste-de-sensibilidade-8379.html>, acesso em 3 março 2023.
- [13] M. Hero, *Sensor de Obstáculo Infravermelho IR* (Maker Hero, Palhoça, 2023). Disponível em: <https://www.makerhero.com/produto/sensor-de-obstaculo-infravermelho-ir/>, acesso em 28 junho 2023.
- [14] UsinaInfo, *Buzzer Ativo 5V Bip Contínuo PCI 12mm*. Disponível em <https://www.usinainfo.com.br/buzzer/buzzer-ativo-5v-bip-continuo-pci-12mm-2988.html>, acesso em 31 março 2023.
- [15] H.M. Nussenzveig, *Curso de Física Básica: Mecânica* (Blucher, São Paulo, 2013), v. 1, p. 20.
- [16] Microsoft, PROJ.LIN (Função PROJ.LIN), disponível em <https://support.microsoft.com/pt-br/office/proj-lin-fun%C3%A7%C3%A3o-proj-lin-84d7d0d9-6e50-4101-977a-fa7abf772b6d>, acesso em 30 maio 2023].