

APRIMORAMENTO DO EXPERIMENTO DO TRILHO DE AR COM SISTEMAS EMBARCADOS

IMPROVEMENT OF THE AIR TRACK EXPERIMENT WITH EMBEDDED SYSTEMS

Mayron Rubens Miranda Limaverde[mayronlimaverde@gmail.com]

Flávio Alberto de Lima Oliveira[flavio.albert@hotmail.com]

Daniel Fonseca de Souza[danielsouza@ifam.edu.br]

Instituto Federal do Amazonas (IFAM)

Av. Gov. Danilo Areosa 1672, Manaus, AM, Brasil

Recebido: 26/12/25

Aceito: 07/04/26

Resumo

Os experimentos atuais de trilhos de ar, apesar de serem funcionais podem ser melhorados e aperfeiçoados. O trabalho busca atualizar e aprimorar o experimento didático conhecido como "Trilho de Ar", utilizado no ensino de cinemática. O objetivo é melhorar sua funcionalidade e torná-lo mais atrativo aos estudantes, integrando tecnologias modernas de hardware e software. Inicialmente, foram identificadas limitações na interface e na capacidade do experimento original de fornecer variáveis relevantes, como velocidade e aceleração. Para solucionar essas questões, foram empregadas plataformas de prototipagem, conectadas a sensores infravermelhos. A interface do sistema foi projetada para ser simples e intuitiva. Essa abordagem visa facilitar o aprendizado prático e conectar o conhecimento teórico à aplicação experimental. Os resultados incluem a implementação de sensores mais eficientes, o desenvolvimento de uma interface amigável e a integração dos dados captados com um sistema de exibição em tempo real. Assim, o Trilho de Ar é transformado em um dispositivo IoT, expandindo suas possibilidades didáticas e contribuindo para o ensino de Física em laboratórios educacionais.

Palavras-Chave: Trilho de Ar; Cinemática; IoT; Arduino; Ensino de Física.

Abstract

The current air track experiments, although functional, can be improved and refined. This work aims to update and enhance the didactic experiment known as the "Air Track," used in the teaching of kinematics. The goal is to improve its functionality and make it more appealing to students by integrating modern hardware and software technologies. Initially, limitations in the original experiment's interface and its capacity to provide relevant variables, such as velocity and acceleration, were identified. To address these issues, prototyping platforms were used, connected to infrared sensors. The system interface was designed to be simple and intuitive. This approach aims to facilitate practical learning and bridge the gap between theoretical knowledge and experimental application. The results include the implementation of more efficient sensors, the development of a user-friendly interface, and the integration of captured data with a real-time display system. Consequently, the Air Track has been transformed into an IoT device, expanding its educational possibilities and contributing to the teaching of physics in educational laboratories.

Keywords: Air Track; Kinematics; IoT; Arduino; Physics Education.

1. Introdução

O aprendizado prático possibilita ao estudante relacionar conceitos físicos abstratos com situações concretas de aplicação, desenvolvido em diversas etapas da formação de um aluno, entre elas podemos destacar: o conhecimento teórico (tradicional), adquirido via aulas, nas quais os professores transmitem os fundamentos conceituais do conteúdo, situando os alunos em um assunto específico, e o conhecimento prático, onde há a aplicação do conhecimento em experimentos e atividades práticas, sendo assim, os alunos tendem a resolver problemas de cunho real e não mais teórico. Parte-se da premissa de que os alunos compreendam os conceitos ao visualizar e aplicar em atividades práticas ou experimentais. Essas atividades experimentais são o que moldam o aluno, tornando-o mais competente para resolver possíveis problemas da sua área de estudo, de acordo com Dewey (1916), “Nós só aprendemos fazendo”, o que evidencia a importância dessa atividade ser inserida no processo educativo ao longo da trajetória de estudos. Dentro das dependências do IFAM-CMDI, mais especificamente no laboratório de física, há uma gama de experimentos, todos com sua importância em propor desafios e demonstrar fenômenos físicos. Mesmo os experimentos sendo funcionais, há uma grande oportunidade de torná-los mais atrativos aos alunos, mediante aperfeiçoamentos que agreguem novas funcionalidades. Um desses experimentos é o Trilho de Ar, que utiliza ar com a finalidade de reduzir significativamente o atrito existente entre o trilho e o cavaleiro (uma peça metálica que é o objeto de análise dos sensores) para estudo dos fenômenos relacionados à cinemática. Após um objeto passar pelos sensores localizados no trilho, o software do experimento, registra os tempos e como é conhecida a distância, torna-se possível com auxílio das equações da cinemática, calcular outras grandezas de interesse, tais como: velocidade e aceleração. Mesmo sendo funcional, o Trilho de Ar apresenta diversas limitações. Sua interface não é convidativa e exibe poucas variáveis de interesse, embora seja possível extrair um conjunto mais amplo de informações experimentais, o próprio software não as demonstra por limitações funcionais. Sua própria interface em geral não é amigável por uma série de comandos para se visualizar informações que poderiam ser mostradas aos usuários de maneira mais simples e rápida. Esse trabalho justifica-se pela observação do estado dos experimentos, que mesmo funcionais, apresentam diversas limitações em sua arquitetura de hardware/software. Na atualidade, em que os equipamentos eletrônicos se destacam pela apresentação atrativa de dados, tabelas e informações, os experimentos atuais do IFAM se encontram em desvantagem se comparados com experimentos mais modernos. Por isso esse trabalho busca atualizar o experimento Trilho de Ar, tornando-o mais eficaz para o ensino da cinemática. As melhorias e adições serão tanto na área de hardware quanto na de software, esses aperfeiçoamentos buscam tornar-se o experimento mais atrativo e útil aos estudantes, permitindo visualizar e compreender de maneira clara e eficiente a área de estudo da cinemática.

2. Referencial Teórico

A transformação de objetos comuns em objetos inteligentes capazes de gerar e compartilhar dados é um fenômeno que se torna cada vez mais frequente. Essa transformação é proporcionada pela aplicação da Internet das Coisas (IoT). A Internet das Coisas constitui uma ponte entre o mundo físico e o digital, oferecendo a capacidade de monitorar e controlar o ambiente em tempo real. Essa mudança ocorre com o emprego de hardware e software modernos que vêm com a capacidade de conectar dispositivos tornando-os IoT (Magrini, 2018; Motta, 2021). A base inicial do projeto se relaciona ao uso de sistemas embarcados, que podem ser definidos como dispositivos computacionais responsáveis por realizar diversas operações específicas dentro de um sistema maior. De acordo com Wolf (2012) eles são o “núcleo invisível de muitas inovações que moldam o mundo moderno”, essa capacidade de realizar diversas operações de maneira rápida e acessível torna os embarcados uma opção ideal para serem implementados em projetos. Não apenas sua simplicidade, mas também o fato de boa parte dos embarcados populares serem de valor acessível e contar com pelo menos uma IDE (ambiente de desenvolvimento integrado) que é uma plataforma virtual que permite aos usuários

programarem seus embarcados para executar seus comandos. Normalmente, cada IDE utiliza uma linguagem de programação específica, tais como C e C++ que oferecem vantagens como alto desempenho, gerenciamento de memória e orientação a objetos, por outro lado a linguagem Python oferece maior facilidade para programação “a simplicidade do Python e a disponibilidade de bibliotecas o tornam mais atrativo”, essa facilidade também o torna mais acessível a diversas bibliotecas, permitindo aplicar funções específicas de maneira mais rápida (Oliphant, 2007; Delgado, 2018). As plataformas escolhidas para o desenvolvimento do projeto foram: Arduino, ESP32 e Raspberry, todas possuindo seu devido embarcado e uma IDE para a programação do dispositivo. Segundo Sebesta (2016), a escolha da linguagem de programação deve levar em conta fatores como eficiência e tempo de desenvolvimento. Apesar de semelhantes em propósito, cada uma das plataformas oferece vantagens e desvantagens diferentes, permitindo alcançar diferentes resultados dependendo da aplicação (Alencar, 2024). Em relação aos trabalhos relacionados, pode-se citar: “Análise do movimento do móvel usando o Trilho de Ar e a placa Arduino como aquisição de dados” (Cavalcanti, 2022), “Construção de um Trilho de Ar com o uso do Arduino: uma proposta para o ensino de função afim e quadrática” (Santos, 2019) e “Análise do movimento do móvel usando o Trilho de Ar e a placa arduino como aquisição de dados” (Cavalcanti, 2016), todos esses trabalhos têm como elemento em comum a escolha do mesmo experimento, mas a metodologia e a finalidade dos trabalhos divergem entre si. Com base comparativa, segue a Tabela 1 com as divergências e semelhanças entre os trabalhos, a fim de simplificar sua identificação, os mesmos serão apresentados como: A, B, C seguindo a ordem respectiva em que foram declarados no parágrafo anterior, e este trabalho, por sua vez, será referido como D.

Tabela 1- Comparação Entre Trabalhos Relacionados

Tabela Comparativa Sobre os Trabalhos			
Trabalho	Objetivo	Embarcado	Modelo do sensor
A	Propor uma prática pedagógica experimental com Arduino para ensinar cinemática.	Número de embarcados: 1	Sensor LDR
		Nome: Arduino	
B	Construir um Trilho de Ar de baixo custo com Arduino, para ensino de funções afins e quadráticas.	Número de embarcados: 1	Sensor Ultrassônico
		Nome: Arduino	
C	Propor prática pedagógica para atrair a atenção de alunos com Trilho de Ar.	Número de embarcados: 1	Sensor LDR
		Nome: Arduino	
D	Atualizar Trilho de Ar com embarcados diversos e criação de interface interativa.	Número de embarcados: 3	Sensor Infravermelho

		Nome: Arduino, ESP32 e Raspberry	
--	--	--	--

3. Metodologia

No experimento do Trilho de Ar, será detectado o movimento do cavaleiro e após o fim do percurso, os sensores irão indicar em um módulo as variáveis de interesse, como por exemplo: o tempo que o cavaleiro leva para completar o percurso. O processo ocorre por meio de uma série de sensores que se encontram no decorrer do trilho, onde os sensores captam e registram o tempo de deslocamento do carrinho entre os sensores posicionados em locais específicos e, em seguida, os valores obtidos são apresentados em uma estrutura chamada de Multicronômetro Digital. Mesmo essa estrutura sendo capaz de mostrar os resultados, sua interface apresenta limitações por ser pouco intuitiva para novos usuários. Uma proposta de melhoria do Trilho de Ar é o aprimoramento do hardware. Nessa etapa o foco é encontrar um sensor eficaz, capaz de detectar o carrinho em movimento, outro ponto a ser levado em consideração refere-se ao tamanho dos sensores. Para isso, a escolha do sensor não foi realizada diretamente no trilho, mas em um pequeno protótipo em escala reduzida, utilizado para testes preliminares. Em seguida, iniciou-se a fase da programação do protótipo, feita a partir da IDE do Arduino, utilizando a linguagem C++, o código desenvolvido tem como base o funcionamento do trilho original, sendo capaz de analisar as informações provenientes do carrinho e calcular os valores das variáveis de forma equivalente ao sistema original. A interface utilizada no Arduino corresponde ao próprio terminal da IDE, o que a torna pouco receptiva e interativa para o usuário. Para que os dados obtidos possam ser realmente apresentados de maneira ideal, a próxima etapa é a integração da IDE do Arduino ao VS Code, que consiste em uma IDE diferente, na qual será possível implementar a programação de uma interface própria para a apresentação e organização de dados. Ambas as IDEs têm suas respectivas funções, cada uma realizando uma tarefa diferente. A IDE do Arduino é programada para a obtenção e organização de dados, por meio dela que os sensores infravermelhos serão capazes de obter os dados básicos que serão apresentados. A IDE do VS Code tem a função de utilizar os dados obtidos para calcular as variáveis de interesse, gerar gráficos em tempo real com os valores calculados e, por fim, apresentar de maneira organizada todos os dados obtidos. A integração de todos esses processos possibilita um aprimoramento significativo para o trilho ar, promovendo a organização de dados, a geração de gráficos que favorecem a compreensão dos fenômenos, uma interface de fácil interação e, principalmente, a transformação do Trilho de Ar convencional em um Trilho de Ar IoT.

3.1 Hardware Do Protótipo

Para a montagem do hardware do protótipo, a base utilizada é a mesma do Trilho de Ar padrão, ou seja, em uma superfície com medidas fixas escolhidas pelo usuário, são posicionados um número de quatro sensores infravermelhos conectados ao Arduino, Figura 1, onde o sensor 1 é utilizado para a inicialização da medição do tempo, já os sensores 2 a 4 para a marcação no trilho para obter os dados sobre o carrinho em movimento na distância em que estiverem posicionados.

Com o objetivo de permitir a visualização da atividade de cada sensor, há um LED complementar ligado a todos os sensores, de modo que, ao ativar um sensor pela passagem do carrinho, o LED é ligado como um identificador visual do funcionamento dos sensores. As distâncias cobertas seguem a seguinte lógica: Distância A: Sensor 1 a 2; Distância B: Sensor 2 a 3 e Distância C: Sensor 3 a 4.

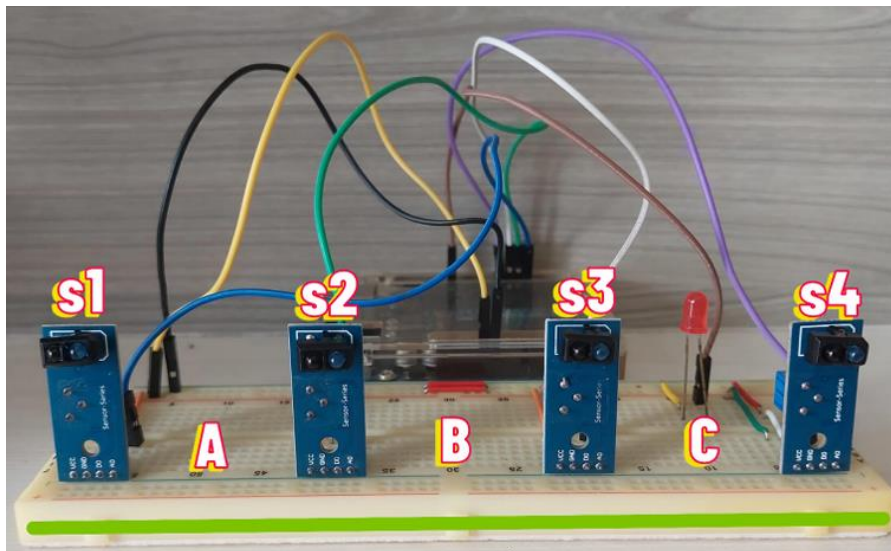


Figura 1 - Identificação dos Elementos

3.2 Software do Protótipo

A implementação do software é dividida em duas IDEs. A primeira é realizada na IDE do próprio Arduino, utilizando a linguagem C++, Figura 2, a qual está diretamente conectada ao circuito. Nessa etapa, o código é responsável por gerenciar o controle da coleta de dados dos sensores, definindo a função de cada sensor e sua relação com os demais sensores no circuito. Além disso, o código realiza a organização inicial dos dados extraídos, preparando-os para serem exportados.

```

1  float startTime;
2  float stopTime;
3  float seg2; // segundos cronômetro
4  float seg3;
5  float seg4;
6  float tempoTT;
7  int var;
8
9  const int pinoLed = 12; // Pino digital do LED
10
11  const int pinoSensor1 = 4; // Sensor 1
12  const int pinoSensor2 = 5; // Sensor 2
13  const int pinoSensor3 = 6; // Sensor 3
14  const int pinoSensor4 = 7; // Sensor 4
15
16  void setup() {
17      pinMode(pinoSensor1, INPUT);
18      pinMode(pinoSensor2, INPUT);
19      pinMode(pinoSensor3, INPUT);
20      pinMode(pinoSensor4, INPUT);
21      pinMode(pinoLed, OUTPUT);
22      digitalWrite(pinoLed, LOW);
23      Serial.begin(9600);
24  }

```

Figura 2 - Inicialização Arduino IDE

Em seguida, os dados são encaminhados para a segunda IDE chamada VS Code, a qual possui duas funções principais. A primeira consiste em utilizar os dados captados na IDE anterior para serem aplicados em equações da cinemática, onde são calculadas as variáveis de interesse, como velocidade, aceleração em cada um dos trechos previamente definidos. A segunda função refere-se ao desenvolvimento da interface gráfica, que possibilitará aos usuários visualizar de forma organizada as variáveis de interesse.

3.3 Integração Hardware e Software

Com as etapas de hardware e software concluídas, torna-se necessário estabelecer uma integração entre ambas para o funcionamento geral do protótipo, conforme ilustrado na Figura 3. A primeira conexão ocorre entre o circuito que abriga os sensores e a plataforma Arduino. Sua relação é estabelecida de forma direta, uma vez que os sensores estão conectados a um microcontrolador Arduino que nativamente já dispõe de uma conexão com sua IDE via porta USB (Souza, 2018).

A integração com o VS Code apresenta maior complexidade, pois o circuito já se encontra associado à IDE do Arduino, ou seja, o VS Code deve ser capaz de estabelecer conexão, tanto com o circuito quanto com a própria IDE do Arduino. Esse processo para a conexão entre eles ocorre em duas etapas: inicialmente o VS Code conecta-se ao circuito por meio da porta USB, estabelecendo a conexão. A segunda etapa, consiste em conectar o VS Code à IDE do Arduino, onde os dados são transmitidos. Estabelecida a integração entre circuito, Arduino e o VS Code, temos o protótipo completamente funcional.

```
import streamlit as st
import serial
import time
import pandas as pd
import matplotlib.pyplot as plt

# Configura a conexão com o Arduino
def inicializar_arduino(porta='COM9', baudrate=9600):
    try:
        arduino = serial.Serial(porta, baudrate)
        time.sleep(2) # Aguarda estabilizar a conexão
        st.write("Conectado ao Arduino!")
        return arduino
    except serial.SerialException:
        st.error("Erro: não foi possível abrir a porta COM9. Verifique a conexão e a porta.")
        return None

# Função para ler os valores do Arduino
def ler_dados_arduino(arduino):
    if arduino and arduino.in_waiting > 0:
        linha = arduino.readline().decode('utf-8').strip()
        valores = linha.split(",")
        if len(valores) == 4:
            try:
                seg2, seg3, seg4, tempoTT = map(float, valores)
                return seg2, seg3, seg4, tempoTT
            except ValueError:
                return None
    return None
```

Figura 3 - Conexão Arduino e VS Code

4. Resultado e Discussão

4.1 Escolha dos Sensores

Os sensores originais do Trilho de Ar são do tipo fotoelétricos, mas devido ao seu grande tamanho e à impossibilidade de conexão direta com a placa Arduino, tornou-se necessária a busca por sensores alternativos. Os sensores selecionados foram aqueles que atendiam aos seguintes critérios: dimensões inferiores às dos sensores originais, compatibilidade de conexão com a placa Arduino e a capacidade de detectar a passagem do cavaleiro em movimento. Os sensores escolhidos que possuíam essas características foram: LDR, ultrassônico e infravermelho. Cada sensor apresenta uma metodologia própria para a aquisição dos dados.

Após a realização de testes e comparações, foi observado diferenças quanto a capacidade e à confiabilidade na obtenção dos dados. O sensor ultrassônico mostrou-se incapaz de detectar o carrinho quando em altas velocidades. Já o sensor LDR, embora funcional e empregado em trabalhos de pesquisa similares, apresentou inconsistências na detecção do carrinho, não em função da velocidade, mas por depender da luminosidade do ambiente, além de exigir o uso de algum sensor extra como fonte de luz para seu adequado funcionamento.

O sensor infravermelho foi o que apresentou maior consistência, atendendo perfeitamente aos critérios estabelecidos e não demonstrou inconsistências durante os testes. Devido a esses fatores ele foi o escolhido e empregado oficialmente no protótipo (Laudares, 2004; Gubbi, 2013).

4.2 Escolha da plataforma de prototipagem

De acordo com Gubbi (2013), um equipamento baseado em Internet das Coisas (IoT) é definido como um conjunto de dispositivos físicos integrados a sensores, software e outras tecnologias que sejam capazes de se comunicar através da internet e interagir com o ambiente à sua volta. De maneira direta, a abordagem usada para cumprir com esses requisitos foi o emprego de plataformas de prototipagem como Arduino, ESP32 e Raspberry Pi, todas oferecendo facilidade para emprego e manuseio de sensores, conexão a plataforma IDE e métodos para a conexão via internet. Como as três plataformas avaliadas atendem aos requisitos necessários, foram observadas as características individuais de cada uma. O Raspberry Pi, embora seja o mais potente entre os três dispositivos analisados, foi descartado por se assemelhar mais a um minicomputador do que a uma plataforma de prototipagem. Tanto o ESP32 quanto o Arduino atendem perfeitamente às tarefas propostas. Entretanto, o espectro decisório entre eles foi a IDE utilizada. No caso do Arduino, a Arduino IDE, apresenta uma ampla gama de bibliotecas e métodos de conexão com VS Code. Além disso, a relação nativa entre a plataforma de prototipagem Arduino e sua IDE é o fator determinante, pois permite que a maioria de todos os processos de integração sejam feitos de maneira simples, sem a necessidade de empregar bibliotecas adicionais.

4.3 Interface

Ao desenvolver uma interface, devem ser considerados alguns fatores, tais como: qual é o objetivo da interface, quais os tipos de usuários envolvidos e se a interface é interativa e de fácil compreensão. De acordo com Norman (2013), o design de uma interface deve ser semelhante ao modelo mental do usuário, de modo a facilitar a criação de conexões e associações mais intuitivas entre o usuário e a interface. O objetivo da interface é apresentar, de maneira simples e objetiva, as variáveis de interesse, permitindo assim que os usuários identifiquem as informações desejadas sem muito esforço cognitivo. O público esperado são alunos que podem ou não ter familiaridade com a área da cinemática. Quanto à escolha do design, esta foi feita para evitar a sobrecarga de informações, optando-se pela divisão em três seções. A primeira seção consiste na apresentação do tempo individual de travessia do cavaleiro nos trechos A, B, C, bem como do tempo total para a conclusão do trajeto. Essa área visa fornecer ao usuário os valores gerais de tempo. A segunda seção da interface, apresentada na Figura 4, é responsável por apresentar todas as variáveis de interesse, a saber: tempo, velocidade e aceleração. Essas variáveis são calculadas para cada um dos trechos previamente

definidos, permitindo uma compreensão mais abrangente dos fenômenos ocorridos ao longo do circuito. A visualização dessas informações é realizada a partir da geração de uma tabela, na qual cada informação pode ser visualizada e encontrada de forma mais organizada e de fácil identificação. A terceira seção, ilustrada na Figura 5, é dedicada à geração de gráficos, o que permite ao aluno compreender os fenômenos físicos, não apenas a partir de valores numéricos, mas também por meio de uma representação visual dos resultados obtidos.

Trecho A (sensor2)	Trecho B (sensor3)	Trecho C (sensor4)	Tempo Total (tempoTT)
0.36 s	0.31 s	0.30 s	0.97 s
	0	1	2
Tempo Total (s)	2.54	0.96	0.97
Velocidade Total (cm/s)	4.7244	12.5	12.3711
Aceleração Total (cm/s ²)	1.86	13.0208	12.7537
Tempo Sensor 2 (s)	0.79	0.3	0.36
Velocidade Sensor 2 (cm/s)	5.0633	13.3333	11.1111
Aceleração Sensor 2 (cm/s ²)	6.4092	44.4444	30.8642

Figura 4 - Monitoramento de Variáveis

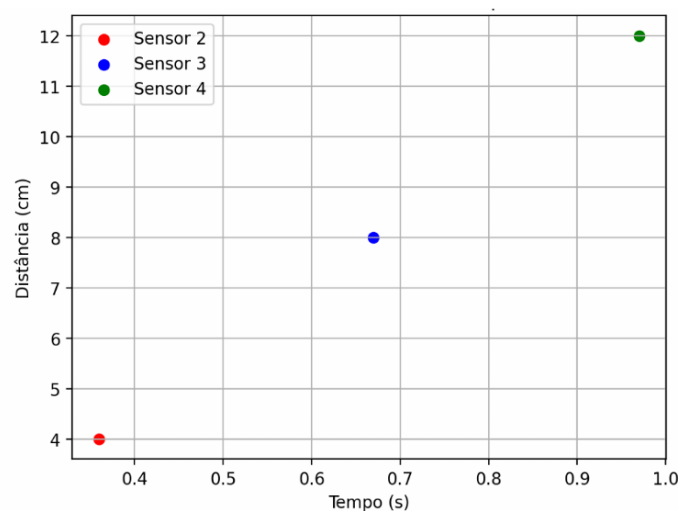


Figura 5 - Gráfico de Distância x Tempo

4.4 Setup Final

Com a integração de todos os componentes apresentados anteriormente, o setup do protótipo consiste em dois núcleos que são: o hardware e o software. O núcleo de hardware final consiste em um circuito composto por quatro sensores infravermelhos, distribuídos a uma distância igual entre eles. O circuito por sua vez está conectado ao microcontrolador final por meio de vinte jumpers, sendo utilizado o Arduino Uno. A integração entre esses componentes e o controle do seu funcionamento ocorre na IDE do Arduino, onde se encontram os comandos responsáveis pela operação dos sensores.

O núcleo de software consiste no código principal desenvolvido no VS Code, já descrito anteriormente, responsável por calcular as variáveis de interesse. Esse código utiliza cinco

bibliotecas, sendo elas: Streamlit, que permite o desenvolvimento aplicações web de maneira rápida e simples; Serial (Pyserial), responsável por possibilitar a comunicação entre os dispositivos de hardware e software; Time, responsável por manipular variáveis de valor temporal; Pandas, empregada no carregamento e na manipulação de estruturas de dados, como DataFrames (tabelas); e, por fim, Matplotlib é utilizado para gerar visualizações gráficas e gráficos em Python. Após a inicialização das bibliotecas, o código procede com a inicialização da conexão entre o Arduino e VS Code por meio da biblioteca Serial. Em seguida, cria-se uma função para ler os dados do Arduino. Uma vez estabelecida a conexão e a troca de dados, o Streamlit é utilizado para a geração e organização das tabelas, gráficos e cartões nos quais os valores são apresentados. Posteriormente, são aplicadas as fórmulas matemáticas necessárias para o cálculo das variáveis de interesse.

4.5 Aplicações no Estudo da Física no IFAM

Com a finalização do experimento e a validação de seu funcionamento, torna-se viável sua aplicação no estudo da cinemática, mais especificamente nos conteúdos de Movimento Retilíneo Uniforme e Movimento Retilíneo Uniformemente Variado. Dessa forma, o sistema permite que os alunos investiguem movimentos como a aceleração constante, bem como estudem a relação entre aceleração, velocidade inicial, velocidade final, deslocamento e outras variáveis associadas, como tempo e distância. No âmbito do Campus do IFAM, é esperado que esse projeto seja utilizado como facilitador do processo de ensino-aprendizagem, auxiliando os alunos na compreensão dos conteúdos da cinemática, por meio de um dispositivo mais simples que o Trilho de Ar convencional e que oferece um retorno mais eficiente e detalhado das informações experimentais.

5. Conclusão

O desenvolvimento do protótipo baseado no experimento Trilho de Ar demonstrou que a integração entre hardware e software pode transformar ferramentas educacionais tradicionais em dispositivos inteligentes, mais eficientes e atrativos, permitindo que os alunos não apenas compreendam os conceitos teóricos, mas também realizem atividades práticas utilizando recursos tecnológicos mais interativos. Por meio da aplicação de sensores infravermelhos e do uso de plataformas como Arduino e VS Code, foi possível criar um sistema que melhora significativamente a experiência de aprendizado na área da cinemática. As melhorias incluem uma interface gráfica intuitiva, responsável por organizar e apresentar os dados de forma clara e objetiva, tornando a compreensão dos conceitos físicos mais acessível para os estudantes. Além disso, a utilização de gráficos em tempo real e a capacidade de realizar cálculos avançados de grandezas associadas ao movimento proporcionam uma experiência mais interativa e didática, na qual os usuários tendem a apresentar um melhor entendimento dos fenômenos ocorridos. Este trabalho reforça a importância da adoção de tecnologias modernas, como IoT e sistemas embarcados, no ensino das ciências exatas, evidenciando como essas soluções podem tornar a educação mais envolvente, dinâmica e eficaz. Futuras implementações podem aprofundar a conectividade e os níveis de automação do sistema, expandindo as possibilidades de aprendizado e de aplicação em outros contextos educacionais.

6. Referências

- Alencar, A. et al. (2024). A influência da linguagem de programação no direcionamento acadêmico para um curso de dois ciclos. In: Workshop sobre Educação em Computação – DF, Brasília: 2024. Anais... Porto Alegre: Sociedade Brasileira de Computação, p. 794-805.
- Cavalcanti, D. et al. (2016). Análise do movimento do móvel usando o trilho de ar e a placa Arduino como aquisição de dados. Dissertação (Mestrado em Ensino de Física) - Universidade Federal de Alagoas.

- Cavalcanti, D. et al. (2022). Análise do movimento do móvel usando o trilho de ar e a placa Arduino como aquisição de dados. *RECIMA21 - Revista Científica Multidisciplinar*, 3(2), 2675-6218.
- Delgado, A. L. (2018). Programação Básica em C++. Acesso em 25 dez., 2024, <https://www.inf.ufpr.br/ci208/notasaula>.
- Dewey, J. (1916). *Democracy and education: an introduction to the philosophy of education*. New York: Macmillan.
- Gubbi, J. et al. (2013). Internet of things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7), 1645–1660.
- Laudares, F.; Lopes, M. & Cruz, F. (2004). Usando sensores magnéticos em um trilho de ar. *Revista Brasileira de Ensino de Física*, 26(3), 233-236.
- Magrani, E. (2018). *A internet das coisas*. Rio de Janeiro: Fundação Getúlio Vargas.
- Motta, R.; Oliveira, K. & Travassos, G. (2021). IoT roadmap: Support for internet of things software systems engineering. *arXiv preprint, arxiv:2103.04969*, Acesso em 25 dez., 2024, <https://arxiv.org/abs/2103.04969>.
- Norman, D. (2013). *The design of everyday things (revised and expanded edition)*. Basic books.
- Oliphant, T. E. (2007). Python for scientific computing. *Computing in science & engineering*, 9(3), 10–20.
- Santos, V. C. et al. (2019). Construção de um trilho de ar com o uso do Arduino: uma proposta para o ensino de função afim e quadrática. Dissertação (Mestrado Profissional em Matemática em Rede Nacional) - Universidade Federal Rural do Rio de Janeiro.
- Sebesta, R. W. (2016). *Concepts of programming languages*. Pearson Education India.
- Souza, V. (2018). *Programando o ESP32 no Arduino*. Clube de Autores.
- Wolf, W. (2012). *Computers as components: principles of embedded computing system design*. San Francisco: Morgan Kaufmann.