UNIVERSIDADE FEDERAL DO PAMPA

BRUNO FLORES MOUCHET

DESENVOLVIMENTO DE UM CRONÔMETRO COM SENSORES INFRAVERMELHOS COMO PLATAFORMA ABERTA E ACESSÍVEL PARA EXPERIMENTOS DE CINEMÁTICA COM TRILHOS DE AR

Bagé
2025

BRUNO FLORES MOUCHET

DESENVOLVIMENTO DE UM CRONÔMETRO COM SENSORES INFRAVERMELHOS COMO PLATAFORMA ABERTA E ACESSÍVEL PARA EXPERIMENTOS DE CINEMÁTICA COM TRILHOS DE AR

Trabalho de Conclusão de Curso apresentado ao Curso de Física da Universidade Federal do Pampa, como requisito parcial para obtenção do Título de Licenciado em Física.

Orientador: Prof. Dr. Paulo Henrique Guadagnini

Bagé
2025

Ficha catalográfica elaborada automaticamente com os dados fornecidos
pelo(a) autor(a) através do Módulo de Biblioteca do
Sistema GURI (Gestão Unificada de Recursos Institucionais).

M924d Mouchet, Bruno Flores
Desenvolvimento de um cronômetro com sensores infravermelhos como plataforma aberta e acessível para experimentos de cinemática com trilhos de ar / Bruno Flores Mouchet.
49 p.

Trabalho de Conclusão de Curso (Graduação)-- Universidade Federal do Pampa, FÍSICA, 2025.
"Orientação: Paulo Henrique Guadagnini".

1. Ensino de Física. 2. Automação. 3. Tecnologias Educacionais. I. Guadagnini, Paulo Henrique. II Título.

BRUNO FLORES MOUCHET

DESENVOLVIMENTO DE UM CRONÔMETRO COM SENSORES INFRAVERMELHOS COMO PLATAFORMA ABERTA E ACESSÍVEL PARA EXPERIMENTOS DE CINEMÁTICA COM TRILHOS DE AR

Trabalho de Conclusão de Curso apresentado ao Curso de Física da Universidade Federal do Pampa como requisito parcial para obtenção do Título de Licenciado em Física.

Trabalho de Conclusão de Curso defendido e aprovado em: 02/07/2025.

Banca examinadora:

_____

Prof. Dr. Paulo Henrique Guadagnini

Orientador

UNIPAMPA

_____

Prof. Ms. Francisco Machado da Cunha

UNIPAMPA

_____

Prof. Ms. Ricardo Gomes Lopes

CFES

Assinado eletronicamente por **PAULO HENRIQUE GUADAGNINI**, **PROFESSOR DO MAGISTERIO SUPERIOR**, em 07/07/2025, às 18:39, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

Assinado eletronicamente por **FRANCISCO MACHADO DA CUNHA**, **PROFESSOR MAGISTERIO SUPERIOR - SUBSTITUTO**, em 07/07/2025, às 23:42, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

Assinado eletronicamente por **Ricardo Gomes Lopes**, **Usuário Externo**, em 08/07/2025, às 17:17, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

RESUMO

Este trabalho apresenta o desenvolvimento de um cronômetro de baixo custo, baseado na Plataforma Microcontroladora Arduino e sensores infravermelhos, para a realização de experimentos de cinemática com trilhos de ar. O objetivo é oferecer uma alternativa acessível e de fácil replicação aos dispositivos comerciais utilizados em laboratórios de ensino, cujos custos costumam ser elevados. O sistema desenvolvido permite medições precisas de tempo em experimentos de Movimento Retilíneo Uniforme (MRU) e, potencialmente, de Movimento Retilíneo Uniformemente Variado (MRUV), utilizando quatro sensores infravermelhos reflexivos posicionados ao longo do trilho. Os dados coletados são exibidos em um display LCD, dispensando conhecimentos prévios em programação por parte do usuário. O projeto adota uma abordagem de código aberto, permitindo modificações e aprimoramentos futuros. Além de detalhar o processo de construção, o trabalho inclui a validação do instrumento por meio da comparação entre os dados obtidos pelo cronômetro e um analisador lógico, evidenciando as limitações e a exatidão do sistema. Os resultados indicam que a proposta contribui significativamente para tornar mais acessível a realização de práticas experimentais no Ensino de Física, fomentando a aprendizagem por meio da integração de tecnologia em atividades laboratoriais.

Palavras-Chave: Ensino de Física; Automação; Tecnologias Educacionais.

ABSTRACT

This work presents the development of a low-cost stopwatch based on the Arduino microcontroller platform and infrared sensors, intended for conducting kinematics experiments with air tracks. The objective is to offer an accessible and easily replicable alternative to commercial devices commonly used in teaching laboratories, which are often expensive. The developed system enables precise time measurements in experiments involving Uniform Rectilinear Motion (URM) and, potentially, Uniformly Accelerated Rectilinear Motion (UARM), using four reflective infrared sensors positioned along the track. The collected data is displayed on an LCD screen, requiring no prior programming knowledge from the user. The project follows an open-source approach, allowing for future modifications and enhancements. In addition to detailing the construction process, the work includes the validation of the instrument through a comparison between the data obtained by the stopwatch and a logic analyzer, highlighting the system's limitations and accuracy. The results indicate that the proposal significantly contributes to making experimental activities in Physics Education more accessible, fostering learning through the integration of technology into laboratory practices.

Keywords: Physics Education; Automation; Educational Technologies.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

ADC – Analog-to-Digital Converter (Conversor Analógico-Digital)

AMS1117 – Regulador de Tensão Linear

ATMEL – Empresa de semicondutores, microcontroladores e componentes eletrônicos

AVR – Regulador Automático de Tensão

BOOTLOADER – Carregador de Inicialização

CIDEPE – Centro Industrial de Equipamentos de Ensino e Pesquisa

DC – Corrente Contínua

EEPROM – Memória Somente Leitura Programável Apagável Eletricamente (Electrically Erasable Programmable Read-Only Memory)

ICSP – Programação Serial no Circuito (In-Circuit Serial Programming)

IoT – Internet das Coisas (Internet of Things)

ISR – Rotina de Serviço de Interrupção

LCD – Liquid Crystal Display

LDR – Light Dependent Resistor (Resistor Dependente da Luz)

LED – Light Emitting Diode (Diodo Emissor de Luz)

MHz – Unidade de Frequência (um milhão de vezes por segundo)

MHS – Movimento Harmônico Simples

MIPS – Milhões de Instruções por Segundo

MRU – Movimento Retilíneo Uniforme

MRUV – Movimento Retilíneo Uniformemente Variado

OC – Open Collector

PPM – Partes por Milhão

PU – Pull-Up

PWM – Modulação por Largura de Pulso

SI – Sistema Internacional de Unidades

SRAM – Memória Estática de Acesso Aleatório

TTL – Transistor-Transistor Logic

VR – Resistor Variável

SUMÁRIO

1 INTRODUÇÃO

Ao longo do curso, a participação do autor em projetos de eletrônica e programação foi proeminente. Destaca-se, inicialmente, o engajamento no Programa Institucional de Bolsas de Iniciação à Docência (PIBID), no âmbito do qual foi concebido e implementado um curso básico de introdução à robótica. O público-alvo desta iniciativa compreendeu professores e alunos da Escola Estadual de Ensino Médio Dr. Luiz Maria Ferraz (CIEP), situada em Bagé, RS. Esta vivência revelou-se de suma importância, uma vez que não apenas solidificou a compreensão dos temas abordados, mas também ofereceu a valiosa oportunidade de exercer a docência, fomentando o aprimoramento profissional na área educacional.

Durante os estágios obrigatórios do curso de Licenciatura em Física (Estágio I, II e III), continuou-se desenvolvendo projetos em eletrônica. No Estágio I, por exemplo, foi desenvolvida uma haste automatizada equipada com sensor de distância e motor, além de terem sido ministradas aulas sobre circuitos elétricos, abordando o uso de resistores, capacitores, entre outros componentes eletrônicos. Dadas essas experiências e os aprendizados nos componentes de laboratórios de física, surgiu o intuito de construção de sensores de movimento que, além de terem um custo baixo, sejam de simples manutenção.

Dessa forma, a justificativa deste projeto de conclusão de curso fundamenta-se na criação de um material que traga benefícios para a Universidade, considerando que trilhos de ar comercializados por empresas têm um custo elevado. Assim, a proposta é desenvolver um instrumento com código aberto, acessível e replicável para uso no laboratório de física.

Nesse contexto, propõe-se a implementação de um sistema que permita a coleta de dados. Embora o foco inicial seja a cinemática, a aplicação dessa abordagem pode ser estendida a experimentos de dinâmica. No entanto, essa possibilidade é considerada para futuros desenvolvimentos, sendo que, para esta primeira versão, o escopo está adequadamente delimitado à cinemática. Assim, o projeto se apresenta como uma contribuição relevante tanto para o aprimoramento das atividades práticas quanto para a acessibilidade de instrumentos.

Na seção seguinte, são apresentados os objetivos do presente trabalho e, na sequência, um levantamento de trabalhos relacionados que envolvem a utilização da Plataforma Microcontrolada Arduino em experimentos com trilho de ar. Será realizada

uma apresentação do kit de trilho de ar utilizado atualmente nos laboratórios de física do Campus Bagé da Unipampa. Apresentado o aparato experimental utilizado atualmente, será demonstrada a metodologia experimental, incluindo a utilização de 4 sensores switch E18-D80NK-N e a automação com o Arduino. Por fim, será descrita a produção de dados experimentais para a realização de análises estatísticas com analisador lógico, comparando a incerteza e o erro de medidas com o aparato experimental construído.

2 OBJETIVOS

2.1 Objetivo geral

O objetivo geral deste trabalho é desenvolver um material complementar para o laboratório de física, utilizando sensores infravermelhos para realizar medições com material de baixo custo.

2.2 Objetivos específicos

Os objetivos específicos são experimentações utilizando o arduino no Movimento Retilíneo Uniforme (MRU) e com potencialidades para Movimento Retilíneo Uniformemente Variado (MRUV), plano inclinado e queda livre com a utilização do display LCD integrada a Plataforma Microcontroladora Arduino. A proposta é que o usuário possa realizar medições mais interativas em cinemática sem a necessidade de conhecimento prévio em programação, bastando apenas observar no display LCD o registro de tempo. Com isso, é fácil perceber que o custo do projeto é baixo em comparação com materiais desenvolvidos por empresas, que geralmente têm um custo elevado e não oferecem uma interface tão amigável, pois são projetados de forma integrada. Por outro lado, o arduino proporciona flexibilidade para modificações e melhorias. Assim, o projeto visa servir como um recurso didático para a Universidade e está aberto a melhorias e aperfeiçoamentos.

3 TRABALHOS RELACIONADOS

Uma breve pesquisa foi realizada em um repositório associado com o Instituto Brasileiro de Informação em Ciência e Tecnologia (Ibict), denominado Oasisbr. De acordo com o site oficial:

> O Portal Brasileiro de Publicações e Dados Científicos em Acesso Aberto (Oasisbr) é uma iniciativa do Instituto Brasileiro de Informação em Ciência e Tecnologia (Ibict) que reúne a produção científica e os dados de pesquisa em acesso aberto, publicados em revistas científicas, repositórios digitais de publicações científicas, repositórios digitais de dados de pesquisa e bibliotecas digitais de teses e dissertações. Deste modo, o Oasisbr tem por objetivo reunir, dar visibilidade e acesso à boa parte dos conteúdos científicos produzidos por pesquisadores que atuam nas instituições brasileiras e portuguesas, publicados em sistemas agregadores de produção e dados científicos. Por meio de uma única interface, o Oasisbr dá acesso às mais diversas tipologias documentais que contém informações científicas, a saber: artigos científicos, livros, capítulos de livros, artigos apresentados em conferências, conjuntos de dados de pesquisa, preprints, dissertações, teses, trabalhos de conclusão de curso, etc. Ademais, o Oasisbr também dá acesso ao conteúdo científico presente no Repositório Científico de Acesso Aberto de Portugal (RCAAP). (Oasisbr, 2025)

A pesquisa realizada no repositório com as palavras chave trilho de ar; Arduino; E18-D80NK; delimita-se a pesquisas realizadas em apenas instituições brasileiras e portuguesas, ou seja, os resultados encontrados vão estar associados a um contexto nacional e na língua portuguesa.

Após uma breve revisão, foram encontradas duas dissertações relacionadas ao uso do Arduino no ensino de física e matemática. A primeira, intitulada "Construção de um Trilho de Ar com o uso do Arduino: Uma Proposta para o Ensino de Função Afim e Quadrática" (Santos, 2019). O trabalho descreve a construção de um trilho de ar de baixo custo para reduzir o atrito de objetos em movimento. O Arduino foi programado para operar com o sensor ultrassônico HC-SR04, que envia 8 pulsos de 40 kHz em apenas 10 microssegundos. Esse sensor mede a distância de um objeto com base no tempo que a onda sonora leva para retornar após refletir no objeto. Sabendo que a velocidade do som no ar é de 340 m/s (ou 0,034 cm/microssegundo), o tempo necessário para o acionamento do sensor é considerado desprezível. Com esse sensor ultrassônico, ele obtém dados para estudar funções afim e quadrática.

A segunda dissertação, intitulada "Análise do Movimento do Móvel Usando o Trilho de Ar e a Placa Arduino como Aquisição de Dados" (Cavalcanti, 2017). O trabalho também propõe a construção de um trilho de ar de baixo custo, porém utilizando um sensor LDR (Light Dependent Resistor), que é sensível à luz e varia sua resistência conforme a intensidade luminosa recebida. Quanto maior a luz incidente, menor a resistência elétrica. O sensor LDR foi programado com o Arduino para controlar a coleta de dados. Esse tipo de sensor é amplamente utilizado na educação devido às suas diversas aplicações, permitindo medir (MRU) e (MRUV) na cinemática.

Com base na revisão bibliográfica identificou-se que o diferencial deste TCC em relação às dissertações apresentadas está na escolha do sensor, na interface e na proposta de código aberto. Este projeto emprega um sensor de nível industrial, o Sensor Infravermelho Reflexivo Industrial E18-D80NK, que, integrado ao Arduino, realiza medições precisas de tempo e permite a visualização no display LCD. Utilizando o trilho de ar já disponível na universidade, o sistema contará com quatro sensores estrategicamente posicionados ao longo do trilho para medir registro de tempo. A programação foi feita no Arduino e desenvolvida para exibir os dados no display LCD, podendo assim com esses dados construir gráficos. O código desenvolvido será aberto para futuras modificações e aprimoramentos, permitindo que os estudantes e professores modifiquem ou adaptem e expandam o sistema conforme as necessidades de seus estudos e experimentos.

4 TRILHO DE AR

Este trabalho é desenvolvido com base no trilho de ar criado pela empresa Cidepe (Centro Industrial de Equipamentos de Ensino e Pesquisa). Sendo que a Cidepe é uma empresa de referência em instrumentos educacionais em instituições de ensino no Brasil e no exterior.

O trilho de ar na Figura 1 apresenta o mesmo modelo no qual serão instalados os sensores E18-D8NK. Dessa forma, será possível realizar as medições e as marcações de templo.

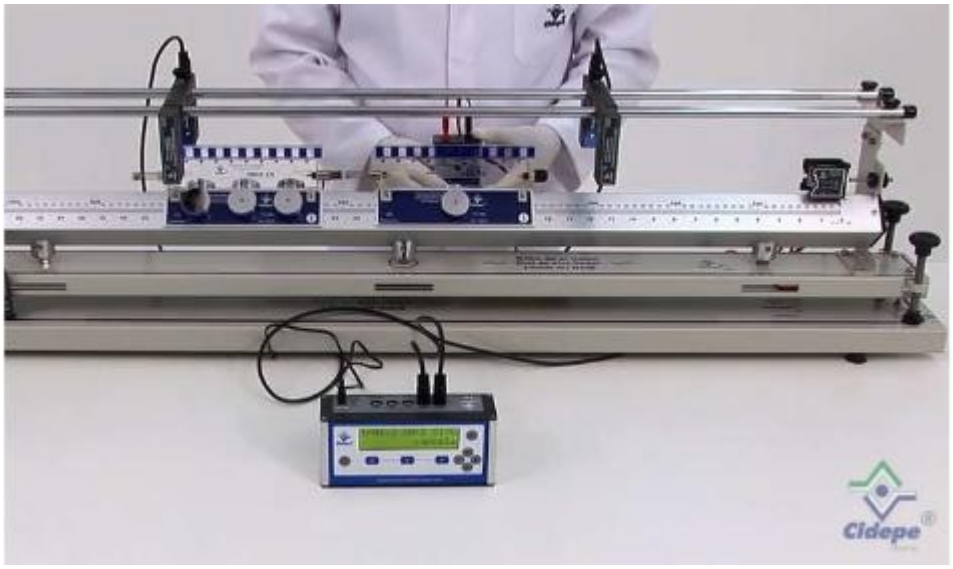
Figura 1 – Laboratório Cidepe

Fonte: Cidepe (2025)

As Figuras 2 e 3 apresenta o equipamento completo utilizado no laboratório de física, incluindo seus principais componentes, trilho de ar, multicronômetro, rolamento, sensores e unidade de fluxo. Neste trabalho, será utilizado apenas o trilho de ar,

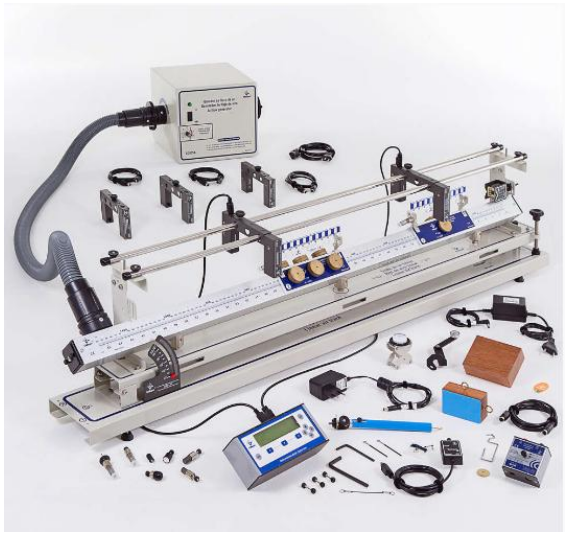
substituindo os sensores originais.

Figura 2 – Trilho de ar Cidepe



Fonte: Cidepe (2025)

Figura 3 – Trilho de ar, multicronômetro, rolagem, sensores



Fonte: Cidepe (2024)

Este produto, desenvolvido pela empresa, é destinado ao estudo experimental em laboratórios de física, permitindo a realização de experimentos que abrangem uma ampla gama de tópicos. No campo da cinemática, ele permite o estudo de conceitos como referencial, posição, movimento e trajetória. Também possibilita a observação do movimento de um corpo, análise de trajetória e deslocamento, e compreensão da diferença entre deslocamento e distância percorrida. Além disso, o sistema de

referência cartesiano no plano é utilizado para entender grandezas escalares e vetoriais.

No caso do Movimento Retilíneo Uniforme (MRU) o trilho de ar permite a construção de tabelas e gráficos relacionando posição e tempo, auxiliando na determinação da velocidade média e na definição da equação horária do MRU. É possível verificar as características do MRU e aprofundar o estudo de conceitos fundamentais sobre o movimento retilíneo uniforme. Para o Movimento Retilíneo Uniformemente Variado (MRUV) o produto permite a criação de tabelas e gráficos de posição em função do tempo e a análise da linha de tendência dos pontos gráficos, que pode ser visualizada por meio de planilhas eletrônicas.

O experimento também permite observar como a posição varia em relação ao tempo no MRUV, com a construção de gráficos específicos de velocidade versus tempo, possibilitando o cálculo da velocidade média e a obtenção da equação de Torricelli, uma equação independente do tempo. Em termos de Dinâmica, o produto facilita o estudo da segunda lei de Newton, permitindo a análise da relação entre força e aceleração, além de experimentos que exploram a conservação de energia, o coeficiente de restituição e a quantidade de movimento e energia cinética durante colisões inelásticas.

É possível coletar dados antes e após a colisão de objetos, compreendendo o conceito de sistema e observando o comportamento da quantidade de movimento e da energia cinética. O equipamento também permite avaliar o coeficiente de restituição entre dois corpos e comparar a quantidade de movimento e a energia cinética antes e após uma colisão elástica ou inelástica. Na Ondulatória, o produto possibilita a determinação da constante elástica de um sistema massa-mola em um Movimento Harmônico Simples (MHS), através da medição de períodos e do cálculo de valores médios.

Também é possível determinar a constante elástica da mola por meio de processos dinâmicos. Esse conjunto de funcionalidades faz do equipamento uma ferramenta abrangente e essencial para o ensino e a experimentação em física, permitindo a realização de uma série de estudos experimentais fundamentais de maneira prática e precisa. (Cidepe, 2025). Neste trabalho, o foco principal será o desenvolvimento e análise dos Movimentos Retilíneo Uniforme (MRU).

5 METODOLOGIA EXPERIMENTAL

Para a construção do sistema, vamos utilizar principalmente os seguintes materiais:

- Sensor Reflexivo Infravermelho Distância Ajustável E18-D80Nk,
- Plataforma Microcontrolada Arduino
- Display Lcd Keypad Shield 16x2

5.1 Sensor Reflexivo Infravermelho Distância Ajustável E18-D80Nk

O Sensor Reflexivo Infravermelho Distância Ajustável E18-D80Nk, mostrado na Figura 4, é um dispositivo para detecção de distância projetado para identificar objetos dentro de uma faixa ajustável de 6 cm a 80 cm, com saída lógica TTL *Transistor-Transistor Logic* (Lógica Transistor-Transistor). A distância de detecção é ajustada por meio de um potenciômetro VR (resistor variável), enquanto o status de detecção é indicado por um LED integrado.

Figura 4 – Sensor Reflexivo Infravermelho Distância Ajustável E18-D80Nk

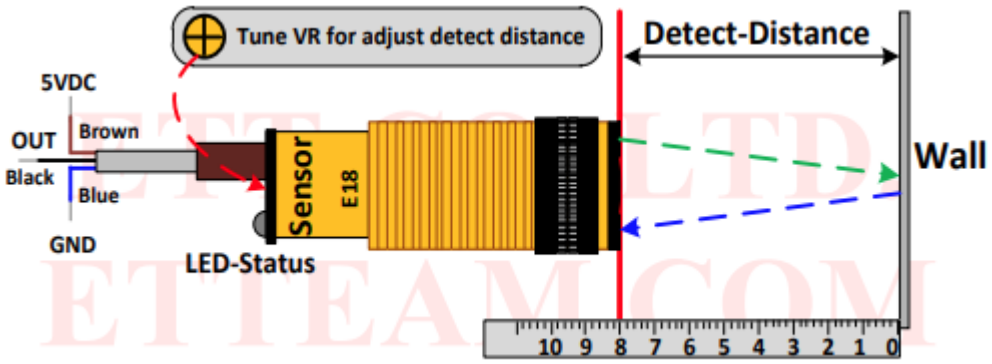
Fonte: Eletrogate (2025)

Este sensor é ideal para a detecção de materiais opacos ou com baixa transparência à luz, sendo que a cor preta é a mais recomendada, pois proporciona melhor absorção do infravermelho, aumentando a precisão do dispositivo. A saída de sinal do sensor é do tipo Open Collector e requer a conexão de um resistor Pull-Up de 10 K na saída para correto funcionamento. O nível de sinal é digital TTL, com valores

de 0 para GND (Terra) e 1 para 5V. O dispositivo opera com uma fonte de alimentação de 5V DC e consome uma corrente de 100mA, garantindo eficiência em aplicações que exigem monitoramento preciso de presença e proximidade. Para configurar o sensor de detecção de distância, iniciou-se a conexão da fonte de alimentação de 5V ao cabo marrom do sensor e do GND ao cabo azul (Figura 5). Em seguida, posicionou-se a parte frontal do sensor de frente para o chão, que apresentava uma superfície preta, a fim de aumentar a precisão da detecção. Com uma régua, foi medida a distância desejada para a detecção e o sensor foi mantido nessa posição por alguns instantes. Após essa etapa, ajustou-se o potenciômetro VR localizado na extremidade do sensor, observando a resposta do LED. Essa observação permitiu identificar quando o LED indicava a detecção na distância previamente configurada, confirmando que o sensor estava calibrado corretamente para a distância necessária (ETT, 2025).

Com base em experimentos realizados, foi observado que a cor do chão, da parede ou de qualquer material utilizado para refletir o sensor pode afetar a precisão da detecção de distância. Quando a superfície de absorção é de cor clara, a mínima distância de detecção do sensor aumenta, o que resulta em uma distância de detecção especificada pelo usuário sendo inferior à distância mínima real do sensor.

Figura 5 – Configuração da detecção de distância do sensor infravermelho



Fonte: ETT (2025)

Portanto, recomenda-se o uso de superfícies escuras, como paredes de cor preta, para garantir que o sensor funcione de maneira ideal. Caso a superfície utilizada não seja suficientemente escura, o usuário deve estar ciente de que a distância de detecção real pode ser maior do que a esperada. Além disso, o usuário deve testar e

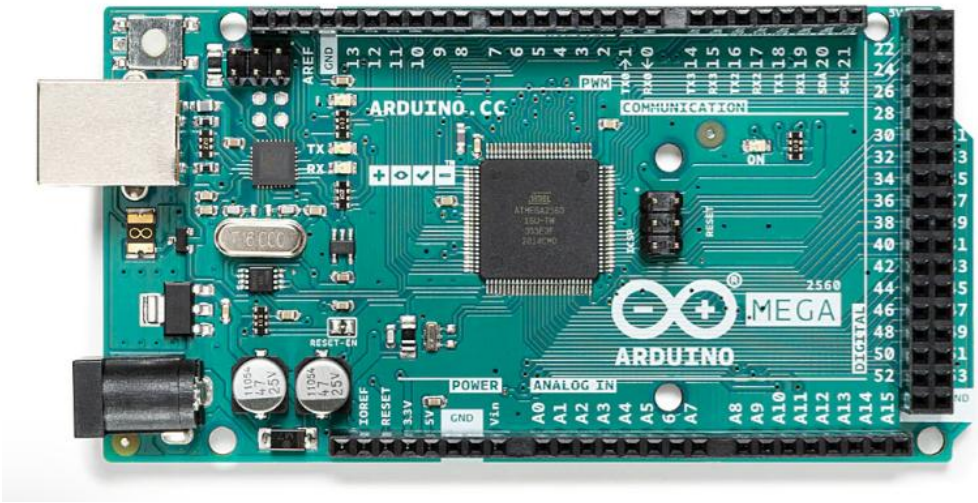
ajustar a distância de detecção de acordo com as características do material refletor, pois cada cor de parede resulta em diferentes níveis de reflexão. Para uma configuração precisa, recomenda-se seguir os passos de 1 a 5 do processo de ajuste descrito anteriormente. Com base nos experimentos realizados no *(Manual of IR Sensor Switch E18-D80NK-N),* a menor distância de detecção observada em uma parede preta é de 6 cm. Se os resultados estiverem de acordo com o passo 5 do procedimento de ajuste, significa que o sensor está funcionando corretamente e está pronto para uso e conexão. (ETT, 2025).

5.2 Plataforma Microcontrolada Arduino

O Arduino é uma plataforma de *hardware Open Source*, de fácil utilização, ideal para a criação de dispositivos que permitam interação com o ambiente. Esses dispositivos possibilitam utilizar como entrada sensores de temperatura, luz, som, etc., e como saída LEDs, motores, displays, alto-falantes, entre outros, criando, dessa forma, possibilidades ilimitadas. A plataforma utiliza-se de uma camada simples de software implementada na placa, que é um *bootloader*, e uma interface amigável no computador que usa a linguagem *processing*, baseada na linguagem C/C++, a qual também é open source. Através do *bootloader*, dispensa-se o uso de programadores para o chip, no caso, a família AVR do fabricante ATMEL facilitando ainda mais o seu uso, uma vez que não exige compiladores ou hardware adicional. Neste ambiente de desenvolvimento, são disponibilizadas bibliotecas que permitem o interfaceamento com outros hardwares, possibilitando o completo desenvolvimento de aplicações simples ou complexas em qualquer área. (Souza, 2011). Sendo utilizado nesse trabalho o Arduino Mega 2560, pois:

> O Arduino Mega 2560 é uma placa micro controladora baseada no ATmega2560. Possui 54 pinos de entrada/saída digitais (dos quais 15 podem ser usados como saídas PWM), 16 entradas analógicas, 4 UARTs (portas seriais de hardware), um oscilador de cristal de 16 MHz, uma conexão USB, um conector de alimentação, um conector ICSP e um botão de reset. Ela contém tudo o que é necessário para suportar o microcontrolador; basta conectá-lo a um computador com um cabo USB ou alimentá-lo com um adaptador CA/CC ou bateria para começar. A placa Mega 2560 é compatível com a maioria dos shields projetados para o Uno e com as antigas placas Duemilanove ou Diecimila. (Arduino Mega 2560 Rev3 Official Store, 2025)

Figura 6 – Arduino Mega 2560



Fonte: Arduino Official Store (2025a)

A Figura 6 mostra o Arduino Mega 2560 e possui internamente o ATmega 2560, que é um microcontrolador robusto, impulsionado por um processador RISC que alcança até 16 milhões de instruções por segundo (MIPS). Ele oferece ampla capacidade de armazenamento com 256 KB de memória flash para programas, 8 KB de memória SRAM para dados temporários e 4 KB de EEPROM para dados persistentes. Para controle de tempo e eventos, possui dois Timers/Contadores de 8 bits e dois de 16 bits, além de um contador em tempo real. Sua capacidade de conversão analógico-digital é notável, com um conversor ADC de 10 bits e 16 canais. Para modulação por largura de pulso (PWM), o chip dispõe de quatro canais de 8 bits e doze canais de 16 bits. A conectividade é garantida por quatro interfaces seriais, uma interface I2C e uma interface SPI, entre outras funcionalidades adicionais. A placa Arduino Mega 2560 foi projetada para lidar com projetos de maior complexidade. Com um total de 54 pinos digitais de Entrada/Saída e 16 entradas analógicas, ela é a escolha ideal para aplicações como impressoras 3D e sistemas robóticos. O coração do Arduino Mega 2560 é o microcontrolador ATmega 2560, que opera a uma frequência de 16 MHz. A placa inclui reguladores de tensão, sendo um de 5V (AMS1117 – 1A) e outro de 3,3V (LpP2985 – 150mA). (Eletrogate, 2025).

5.3 Display Lcd Keypad Shield 16x2

Para a interface de comunicação, utilizou-se o display LCD Keypad Shield 16x2

(Figura 7) para a interação do usuário com o experimento, o qual permite, ao interagir com os botões, acessar a segunda página do LCD onde se encontra o restante dos valores de tempo. Para utilizar o sensor, é necessário que, ao programar os botões no software, seja incluída a biblioteca “LiquidCrystal.h”, que é a configuração padrão para os displays. Utilizou-se o datasheet do display LCD para a programação dos botões (LCD Keypad Shield DFR0009), que se encontra nas referências para acesso caso haja necessidade.

Figura 7 – Display LCD Keypad Shield 16x2

Fonte: D-Robotics (2025)

Em suas especificações, ele opera com tensão de 5V, possui 6 botões de pressão e pinagem analógica expandida com configuração padrão conforme a Figura 8.

Figura 8 – Configuração pinagem do LCD

Pin	Function
Analog 0	Button (Select Up, Right, Down and Left)
Digital 4	DB4
Digital 5	DB5
Digital 6	DB6
Digital 7	DB7
Digital 8	RS (Data or Signal Display Selection)
Digital 9	Enable
Digital 10	Backlit Control

Fonte: D-Robotics (2025)

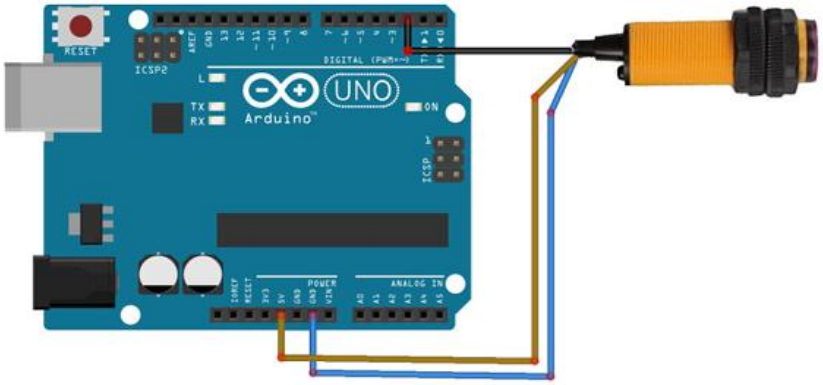
Para auxiliar na compreensão e implementação, um código fonte elaborado pelo autor, que demonstra a configuração dos botões no software, está disponível na

referência citada. (D-Robotics, 2025).

5.4 Sistema Geral de Detecção de Objetos com Sensores E18-D80nk

Utilizando os sensores para medir a passagem de objetos em movimento, o esquema de montagem para um sensor está sendo disponibilizado na Figura 9. Os demais sensores seguem o mesmo padrão de conexão e, para melhor fixação, pode-se usar uma placa universal ou, de forma mais básica, uma protoboard. Os sensores foram configurados para detectar a presença do objeto e marcar o tempo de passagem com o auxílio do Arduino. Com esses quatro sensores, o objetivo é medir a diferença no tempo registrado por cada um deles.

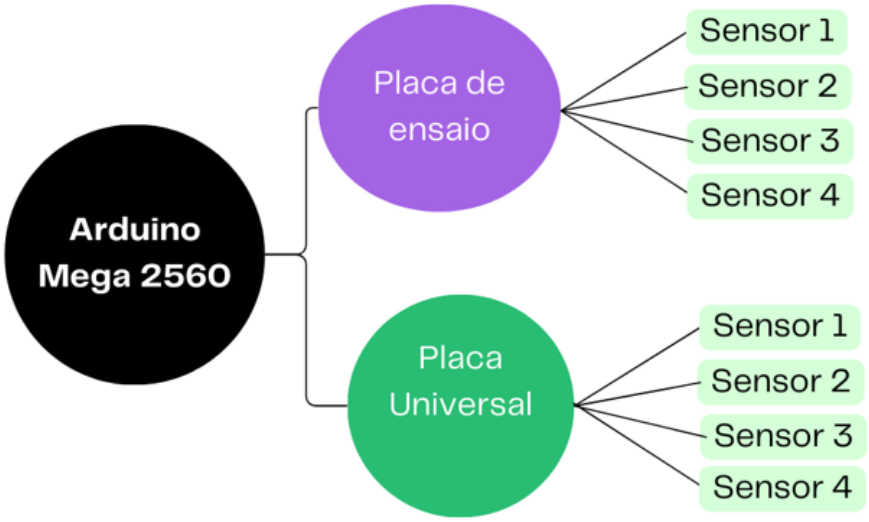
Figura 9 – Conexão do sensor no Arduino



Fonte: Autoria própria (2025)

A configuração é padrão para os quatro sensores (Figura 10), por exemplo, em uma protoboard ou placa universal, envolve conectar o fio marrom (5V) e o fio azul (GND) à placa alimentada pelo 5V e GND do Arduino. Essa ligação pode ser feita em série diretamente para energizar os sensores. Em seguida, o fio de saída (OUT - preto) dos sensores foi conectado a portas digitais do Arduino que suportam a função de interrupção, essenciais para a leitura precisa do sinal. A disposição geral dos sensores pode seguir o arranjo ilustrado nas imagens a seguir, mas é flexível e pode ser adaptada conforme a preferência do usuário. Independentemente da fixação escolhida, é fundamental que as distâncias entre os sensores sejam devidamente calibradas.

Figura 10 – Diagrama de construção



Fonte: Autoria própria (2025)

Figura 11 – Sensores frontal

Fonte: Acervo da pesquisa (2025)

Visualizando os sensores de frente (Figura 11) e fixando-os da melhor forma possível, podemos utilizar a câmera de um aparelho qualquer que capture imagem. Nesse caso, foi utilizada uma câmera de celular para identificar a localização do emissor de infravermelho, apresentado na Figura 12.

Figura 12 – Identificação do emissor infravermelho

Fonte: Acervo da pesquisa (2025)

Após a identificação, observa-se a cor roxa, é necessário que se faça a medida da distância entre os sensores (distância entre dois emissores), pois para o MRU é necessário que se tenha distâncias iguais entre os sensores, que nesse caso foi de 10 cm (Figura 13).

Figura 13 – Medida das distâncias entre os emissores

Fonte: Acervo da pesquisa (2025)

Lembrando que deve estar em metros para o SI, essa modificação pode ser ajustada automaticamente no código elaborado conforme disponibilizado no Apêndice A. Caso o leitor queira modificar as distâncias, isso pode ser feito de forma simples,

mudando diretamente no código desenvolvido.

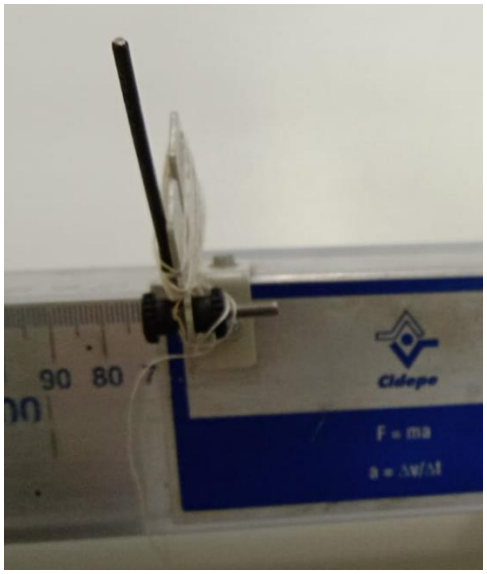
Com o LCD Keypad Shield encaixado no Arduino Mega 2560, antes da detecção e conforme especificado no código do apêndice A, a tela do LCD apresenta a configuração mostrada na figura 14.

Figura 14 – Tela inicial LCD

Fonte: Acervo da pesquisa (2025)

O objeto a ser medido neste caso do trilho de ar comercializado pela Cidepe é uma pequena haste de cor preta para melhor identificação com infravermelho, conforme a Figura 15.

Figura 15 – Objeto de detecção



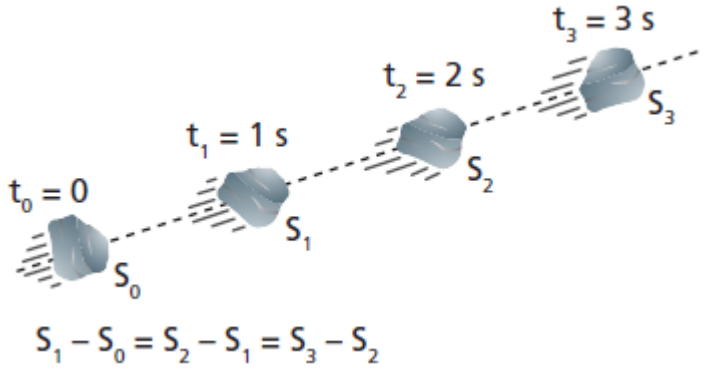
Fonte: Acervo da pesquisa (2025)

6 SISTEMA DE DETECÇÃO MRU

Para introduzirmos o sistema de medição faz-se necessário verificar algumas definições do MRU. “Movimento Retilíneo Uniforme (MRU) é aquele em que a velocidade escalar instantânea constante e diferente de zero, de modo que móvel sofre iguais variações de espaço em iguais intervalos de tempo.” (Doca; Biscuola, Villas Bôas, 2012).

A trajetória pode ser de qualquer tipo, seja em linha reta ou em curva, sem qualquer limitação. Por exemplo, imagine uma pedra arremessada no espaço, onde a gravidade é mínima, ela se moverá em linha reta e com velocidade constante. Isso significa que, a cada intervalo de tempo, ela cobrirá a mesma distância (Doca; Biscuola, Villas Bôas, 2012).

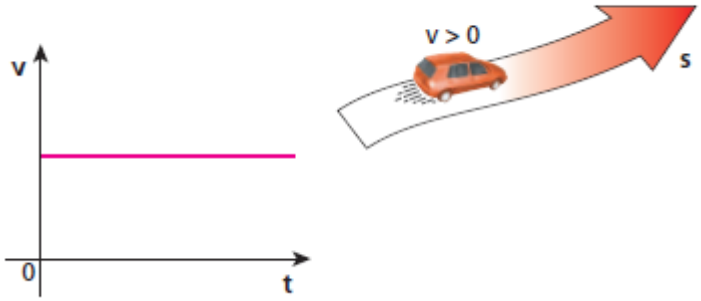
Figura 16 – Movimento Uniforme



Fonte: Doca, Biscuola e Villas Bôas (2012)

A representação gráfica da velocidade em um movimento uniforme, a velocidade escalar instantânea permanece sempre a mesma em qualquer ponto do tempo. A Figura 17 mostra o gráfico que representa essa velocidade em função do tempo.

Figura 17 – Gráfico velocidade x tempo



Fonte: Doca, Biscuola e Villas Bôas (2012)

A velocidade escalar é constante e diferente de zero, o que nos leva à conclusão de que o movimento é uniforme. A velocidade escalar é positiva e, por isso, concluímos que o movimento se dá no sentido da trajetória (movimento progressivo). (Tópicos de física. v.1, 2012. p. 35.)

Assim, pode-se escrever, no intervalo de $t\ a\ t_0$ a função horária do espaço

$$V = \frac{\Delta s}{\Delta t} \rightarrow V = \frac{s - s_0}{t - t_0} = \frac{s - s_0}{t}$$

$$s - s_0 = vt \quad \rightarrow \quad s = s_0 + vt$$

$$\Delta s = vt \qquad (1)$$

A equação (1) obtida é a função horária dos espaços para qualquer movimento uniforme. Onde S_0 é o espaço em $t_0 = 0$, ou seja, o espaço inicial; V é a velocidade escalar e S é o espaço num instante t qualquer (Doca; Biscuola; Villas Bôas, 2012. p. 35). A partir dessas definições, utilizamos a equação (1) para fazer as medidas com um móvel no trilho de ar simulando um movimento uniforme. De acordo com os dados obtidos, temos, nas Tabelas1 e 2:

Tabela 1 – Registro individual dos sensores

Tempos registrados para cada sensor			
$t_1 = 6{,}99s$	$t_2 = 7{,}13s$	$t_3 = 7{,}27s$	$t_4 = 7{,}41s$

Fonte: Autoria própria (2025)

Tabela 2 – Intervalos de tempo entre os sensores

Intervalo de tempo entre os sensores		
$\Delta t_1 = t_2 - t_1 = 0{,}14s$	$\Delta t_2 = t_3 - t_2 = 0{,}14s$	$\Delta t_3 = t_4 - t_3 = 0{,}14s$

Fonte: Autoria própria (2025)

Cálculo da velocidade escalar (Tabela 3): Como o movimento é uniforme, a velocidade é constante e pode ser calculada para cada trecho entre os sensores usando a equação (1), onde $\Delta s = 0{,}1m$ (distância fixa entre os sensores) e Δt é o intervalo de tempo entre os sensores.

Tabela 3 – Velocidade escalar entre dois registros

Velocidade escalar		
$V_1 = \frac{0{,}1}{0{,}14} \approx 0{,}714\ m/s$	$V_2 = \frac{0{,}1}{0{,}14} \approx 0{,}714\ m/s$	$V_3 = \frac{0{,}1}{0{,}14} \approx 0{,}714\ m/s$
$t_1\ a\ t_2$	$t_2\ a\ t_3$	$t_3\ a\ t_4$

Fonte: Autoria própria (2025)

Como os intervalos de tempo são iguais e o movimento é uniforme, as velocidades calculadas são idênticas, confirmando que a velocidade escalar é constante:

$$V \approx 0{,}714\ m/s$$

E usando a função horaria do espaço,

$$s(t) = s_0 + vt \qquad (2)$$

onde $S_0 = 0\ m$ no início do movimento, V é a velocidade escalar $(0{,}714\ m/s)$ e t é o tempo. Portanto, a função horária fica: $s(t) = 0 + 0{,}724.t = 0{,}714.t$ e o tempo total para percorrer a distância total $(0{,}3\mathrm{m})$, usando a função horaria $S(t) = 0{,}3m$:

$$0{,}3 = 0{,}714\ .\ t_{total}$$

$$t_{total} = 0{,}42\ s$$

Os dados mostram que o tempo total registrado é: $t_4 - t_1 = 7{,}41 - 6{,}99 = 0{,}42\ s$ que é o tempo que o móvel percorreu na distância total implicando em uma velocidade média total:

$$V_{total} = \frac{0{,}3\ m}{0{,}42\ s} \approx 0{,}714\ m/s$$

A Tabela 4 reúne os resultados das medições das grandezas.

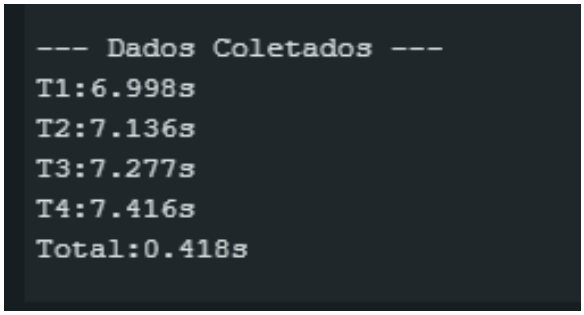
Tabela 4 – Grandezas obtidas

Grandeza	Valor
Distância total, S_{total}	0,3 m
Tempo total	0,42 s
Velocidade escalar média total, V_{total}	0,714 m/s

Fonte: Autoria própria (2025)

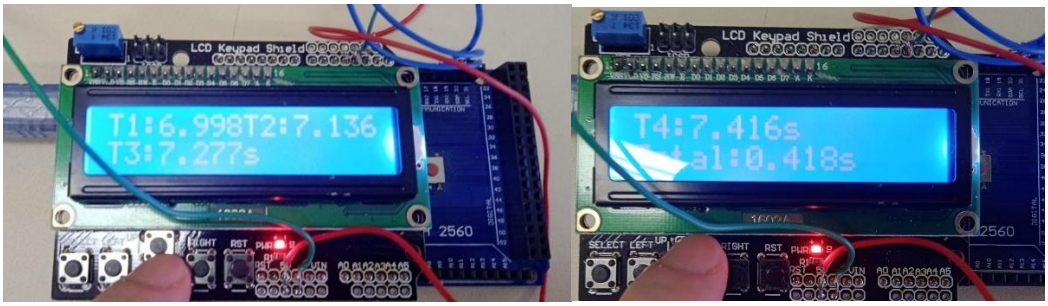
O móvel percorreu $0{,}3\ m$ em $0{,}42\ s$, mantendo uma velocidade constante de ≈ $0{,}714\ m/s$. Como os intervalos de tempo e distâncias são proporcionais, o movimento é retilíneo e uniforme (MRU). Se a velocidade variasse entre trechos, os Δt seriam diferentes, indicando aceleração. Neste caso, todos os dados estão consistentes com o MRU, como observado nas Figuras 18 e 19.

Figura 18 – Monitor serial



Fonte: Acervo da pesquisa (2025)

Figura 19 – Medida no Display LCD



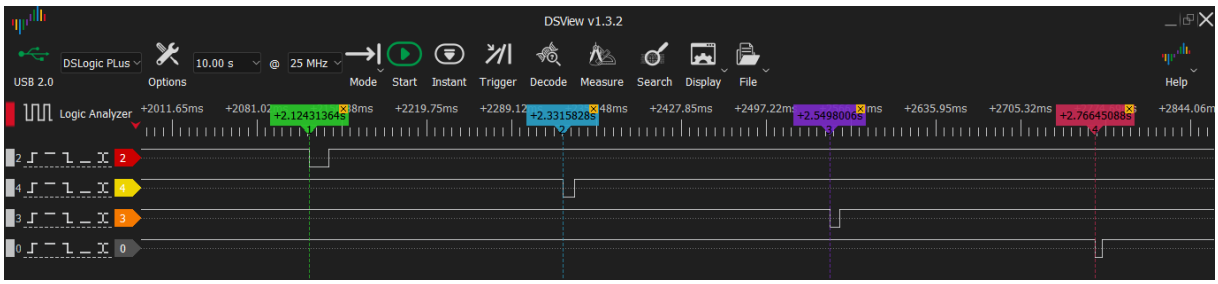
Fonte: Acervo da pesquisa (2025)

7 VALIDAÇÃO DO INSTRUMENTO

Para a validação do protótipo, utilizou-se um analisador lógico com o objetivo de testar a precisão do sensor em relação ao Arduino. Como já se espera, o Arduino apresenta uma perda no tempo de resposta, visto que necessita interpretar comandos e realizar a comunicação com o display LCD. Devido a essa característica, apesar de o sistema funcionar adequadamente para os experimentos, foi necessário quantificar a perda de precisão do instrumento com base nos dados coletados. É importante lembrar que o analisador lógico, por ser um dispositivo desenvolvido especificamente para essa finalidade e com poucas restrições de processamento em comparação com o Arduino, certamente proporcionaria uma precisão de medição maior e praticamente absoluta. Dessa forma, os valores obtidos pelo analisador lógico foram considerados como o valor teórico, enquanto as medições realizadas pelo Arduino representaram o valor experimental.

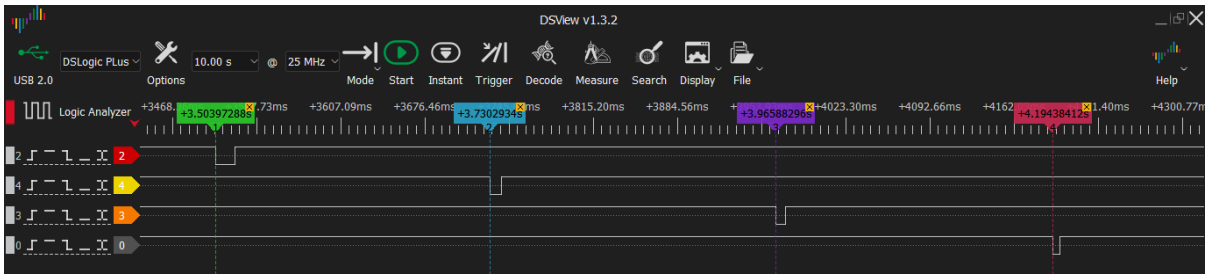
Nas Figuras 20, 21, 22 e 23, é possível observar as medidas realizadas no analisador lógico e no Arduino. Usando o dispositivo detectou-se como objetivo as variações nas tensões que permanece no estado alto ($\approx 5V$) nas portas digitais durante um período de medição de 10 segundos, utilizando uma frequência de amostragem máxima de 25 MHz. Essa frequência significa que o analisador lógico coleta 25 milhões de amostras por segundo, resultando em um intervalo de tempo de 40 nanossegundos entre cada amostra. Dessa forma, o analisador lógico pode distinguir eventos ou mudanças de estado com uma separação mínima de 40 nanossegundos, o que é importante para a detecção de *glitches* (pulsos curtos e inesperados) em sinais digitais. Uma frequência de amostragem muito baixa poderia impedir o registro de um *glitch* rápido que ocorresse entre duas amostras.

Figura 20 – Teste experimental I



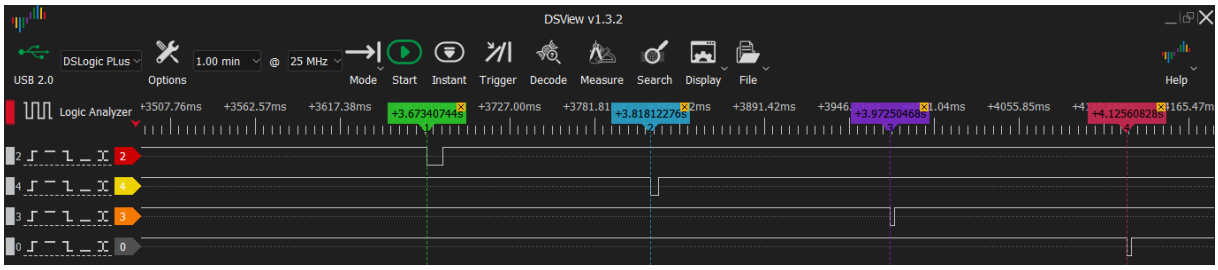
Fonte: Acervo da pesquisa (2025)

Figura 21 – Teste experimental II



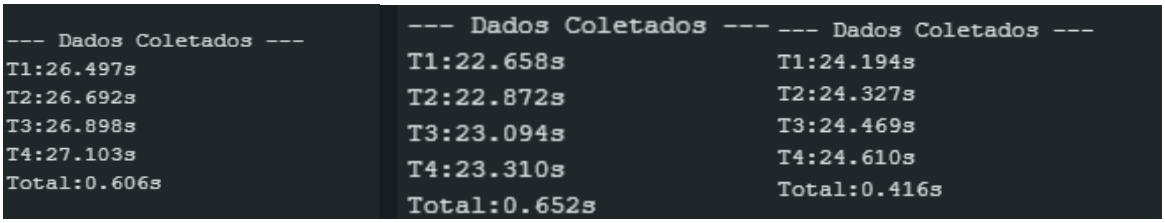
Fonte: Acervo da pesquisa (2025)

Figura 22 – Teste experimental III



Fonte: Acervo da pesquisa (2025)

Figura 23 – Dados coletados: Serial monitor teste 1,2 e 3



Fonte: Acervo da pesquisa (2025)

Como observado nas figuras, percebem-se variações na sensibilidade à tensão entre os sensores. O sensor quatro, por exemplo, apresenta uma queda de tensão mais rápida em comparação com o terceiro, segundo e primeiro. Através de alguns ajustes no potenciômetro dos sensores, foi possível obter uma queda de tensão “menos abrupta”, pois as quedas anteriores não forneciam informações confiáveis. Como as medidas necessitam de uma queda de tensão para ter as marcações do tempo, quando essa queda é ativada o sensor registra o tempo em que o objeto passou e o analisador lógico se mostrou útil para identificar quaisquer problemas como glitches e permitiu corrigir essas inconsistências.

Para obter medidas de precisão diretamente no Arduino, a coleta de informações precisas torna-se complicada devido a uma certa "lentidão" inerente ao seu processamento, que é compartilhado entre o hardware e o display LCD. Portanto, recomenda-se que o usuário utilize o analisador lógico caso deseje ter uma noção dessa perda de precisão ao usar o Arduino.

Com o desvio relativo percentual, equação (3), e comparando os intervalos de tempo medidos pelo Analisador Lógico (Δ_{DS}) com os intervalos de tempo medidos pelo Arduino (Δ_{AD}) para cada teste estão apresentados nas Tabelas 5 e 6. Isso permite uma avaliação de concordância entre as medições dos dois dispositivos, mostradas nas Tabelas 7, 8, 9 e 10.

$$Desvio\ Relativo\ Percentual = \left[\frac{\Delta_{AD} - \Delta_{DS}}{\Delta_{DS}}\right].100\% \quad (3)$$

Tabela 5 – Dados: Analisador Lógico DSview

Analisador lógico Δ_{DS}		
Teste 1	Teste 2	Teste 3
T1 = 2,12431364 s	T1 = 3,50397288 s	T1 = 3,67340744 s
T2 = 2,3315828 s	T2 = 3,7302934 s	T2 = 3,81812276 s
T3 = 2,5498006 s	T3 = 3,96588296 s	T3 = 3,97250468 s
T4 = 2,76645088 s	T4 = 4,19438412 s	T4 = 4,1256828 s
Δ_{1DS1} = 0,20726916	Δ_{2DS1} = 0,22632052	Δ_{3DS1} = 0,14471532
Δ_{1DS2} = 0,2182178	Δ_{2DS2} = 0,23558956	Δ_{3DS2} = 0,15438192
Δ_{1DS3} = 0,21665028	Δ_{2DS3} = 0,22850116	Δ_{3DS3} = 0,15317812

Fonte: Autoria própria (2025)

Tabela 6 – Dados: Arduino

Arduino Δ_{AD}		
Teste 1	Teste 2	Teste 3
T1 = 26,497 s	T1 = 22,658 s	T1 = 24,194 s
T2 = 26,692 s	T2 = 22,872 s	T2 = 24,327s
T3 = 26,898 s	T3 = 23,094 s	T3 = 24,469 s
T4 = 27,103 s	T4 = 23,310 s	T4 = 24,610 s
Δ_{1AD1} = 0,195	Δ_{2AD1} = 0,214	Δ_{3AD1} = 0,133
Δ_{1AD2} = 0,206	Δ_{2AD2} = 0,222	Δ_{3AD2} = 0,142
Δ_{1AD3} = 0,205	Δ_{2AD3} = 0,216	Δ_{3AD3} = 0,141

Fonte: Autoria própria

Como foi explicado inicialmente uma hipótese plausível para o desvio observado nas medições do Arduino em comparação com as do analisador lógico está nas limitações do próprio Arduino, do display LCD e dos sensores. Diversos fatores contribuem para essa diferença, destacando-se a resolução do timer do Arduino através da função millis(), que utiliza timers internos para medir intervalos de tempo. A resolução desses timers é finita e depende da frequência do clock do microcontrolador. A função millis() tem uma resolução de 1 milissegundo e introduz um erro de quantização com viés negativo, pois o arredondamento de cada leitura de tempo é feito para o múltiplo inferior de 1ms. Para futuras construções do instrumento, seria mais adequado usar a função micros(), que oferece uma resolução superior de 4 microssegundos.

Tabela 7 – Desvio Relativo Percentual

Teste 1	Teste 2	Teste 3
$Desvio\ 1 = \left[\frac{0{,}195 - 0{,}2726916}{0{,}2726916}\right].100\%$ $\approx 5{,}92\%$	$Desvio\ 1 = \left[\frac{0{,}214 - 0{,}22632052}{0{,}22632052}\right].100\%$ $\approx 5{,}44\%$	$Desvio\ 1 = \left[\frac{0{,}133 - 0{,}14471532}{0{,}14471532}\right].100\%$ $\approx 8{,}10\%$
$Desvio\ 2 = \left[\frac{0{,}206 - 0{,}2182178}{0{,}2182178}\right].100\%$ $\approx 5{,}60\%$	$Desvio\ 2 = \left[\frac{0{,}222 - 0{,}23558956}{0{,}23558956}\right].100\%$ $\approx 5{,}77\%$	$Desvio\ 2 = \left[\frac{0{,}142 - 0{,}15438192}{0{,}15438192}\right].100\%$ $\approx 8{,}02\%$
$Desvio\ 3 = \left[\frac{0{,}205 - 0{,}21665028}{0{,}21665028}\right].100\%$ $\approx 5{,}38\%$	$Desvio\ 3 = \left[\frac{0{,}216 - 0{,}22850116}{0{,}22850116}\right].100\%$ $\approx 5{,}47\%$	$Desvio\ 3 = \left[\frac{0{,}141 - 0{,}15317812}{0{,}15317812}\right].100\%$ $\approx 7{,}95\%$

Fonte: Autoria própria (2025)

Tabela 8 – Validação do instrumento I

Teste 1			
Intervalo	Analisador lógico Δ_{1DS}	Arduino Δ_{1AD}	Desvio relativo %
1	Δ_{1DS1} = 0,20726916	Δ_{1AD1} = 0,195	5,92
2	Δ_{1DS2} = 0,2182178	Δ_{1AD2} = 0,206	5,60
3	Δ_{1DS3} = 0,21665028	Δ_{1AD3} = 0,205	5,38

Fonte: Autoria própria (2025)

Tabela 9 – Validação do instrumento II

Teste 2			
Intervalo	Analisador lógico Δ_{2DS}	Arduino Δ_{2AD}	Desvio relativo %
1	Δ_{2DS1} = 0,22632052	Δ_{2AD1} = 0,214	5,44
2	Δ_{2DS2} = 0,23558956	Δ_{2AD2} = 0,222	5,77
3	Δ_{2DS3} = 0,22850116	Δ_{2AD3} = 0,216	5,47

Fonte: Autoria própria (2025)

Tabela 10 – Validação do instrumento III

Teste 3			
Intervalo	Analisador lógico Δ_{3DS}	Arduino Δ_{3AD}	Desvio relativo %
1	Δ_{3DS1} = 0,14471532	Δ_{3AD1} = 0,133	8,10
2	Δ_{3DS2} = 0,15438192	Δ_{3AD2} = 0,142	8,02
3	Δ_{3DS3} = 0,15317812	Δ_{3AD3} = 0,141	7,95

Fonte: Autoria própria (2025)

O percentual de erro está abaixo de 10%, o que é aceitável para a validação do instrumento. Outra observação importante é que o Arduino Mega 2560 usa um ressonador cerâmico (modelo CSTCE16M0V53-R0) como sua base de tempo. Embora ressonadores cerâmicos sejam mais baratos, menores e mecanicamente mais robustos que os cristais de quartzo, eles sacrificam precisão e estabilidade.

Cristais de quartzo, por outro lado, utilizam a propriedade piezoelétrica para gerar uma frequência muito estável e precisa, sendo a escolha preferida para aplicações onde a temporização exata é crítica, como relógios de tempo real ou comunicações de alta velocidade. Um ressonador cerâmico funciona de forma semelhante a um cristal, utilizando uma frequência dentro do componente elétrico, mas, diferentemente do cristal, que tem uma tolerância de frequência de 10 a 30 PPM, um ressonador cerâmico possui uma tolerância de frequência de 0,5% ou 5.000 PPM, geralmente usada em aplicações de microprocessadores onde a estabilidade absoluta não é crucial. (ECS INC INTERNATIONAL, 2025).

As especificações do ressonador CSTCE16M0V53-R0 mostram que sua tolerância de frequência é de +/-0,50% máximo, significando que a frequência real pode variar entre 15,92 MHz e 16,08 MHz (Figura 24). E ainda, há uma variação de frequência de +/-0,30% devido à temperatura de operação (de -20°C a 80°C) e +/-0,20% devido ao envelhecimento do componente ao longo do tempo. Somando essas tolerâncias, a frequência real do clock do Arduino pode variar em até aproximadamente 1% do seu valor nominal de 16 MHz, contribuindo significativamente para os erros nas medições de tempo (Murata, 2025).

A latência do software é outro fator relevante. A leitura do estado do pino do sensor e o registro do tempo envolvem a execução de código, o que introduz um atraso que pode variar ligeiramente a cada leitura. Essa variação é possivelmente crítica se outras tarefas, como a atualização do display LCD, estiverem sendo executadas simultaneamente no microcontrolador. Essa latência, por não ser constante, contribui para imprecisões nos intervalos de tempo medidos, reforçando que o Arduino não foi projetado especificamente para análise de sinais de alta precisão. Como sugestão para esses erros de latência, recomenda-se que, dentro da Rotina de Serviço de Interrupção (ISR), apenas o mínimo necessário seja executado, como salvar o valor de millis() ou micros() em uma variável separada que para atualizar o display devem ser movidas para o loop principal do programa, fora da ISR. Embora o Arduino possa ser programado para "detectar" sinais em seus pinos, a taxa de amostragem máxima que ele consegue atingir é limitada por sua arquitetura e pela forma como o software é executado.

Figura 24 – Especificações do ressonador CSTCE16M0V53-R0

Product Type	Ceramic Resonator (CERALOCK)
Series	CSTCE_V
Frequency	16.000MHz
Frequency Tolerance	+/-0.50% max.
Operating Temperature Range	-20°C~80°C
Frequency Shift by Temperature	+/-0.30% max.
Frequency Aging	+/-0.20% max.
Resonant Impedance (R1)	40ohm max.
Built-in Load Capacitance (CL1/CL2)	15pF
Shape	SMD
Wash	Not available
L x W (size)	3.2x1.3mm
Mass	17mg

Fonte: Murata (2025)

Recomenda-se a utilização de modelos de Arduino equipados com cristal de quartzo, idealmente com uma tolerância de 20 PPM, em detrimento do ressonador padrão do mega. Para um aprimoramento e expansão de funcionalidades em aplicações de IoT, pode-se considerar o uso do ESP32. O Arduino Cloud oferece suporte a uma vasta gama de placas de desenvolvimento baseadas em ESP32/ESP8266. Os chips ESP são adequados para qualquer projeto de IoT e sua programação pode ser realizada utilizando a linguagem Arduino (C++). A configuração dessas placas no Arduino Cloud é um processo rápido e simplificado (Arduino Official Store, 2025b). Para informações adicionais sobre as placas ESP, é possível consultar a página de referência (Arduino Official Store, 2025a).

8 CONSIDERAÇÕES FINAIS

Este trabalho representa um "ponto de partida", demonstrando como instrumentos de laboratório de física básica podem ser recriados utilizando eletrônica e programação simples, de baixo custo e com recursos acessíveis. Sua natureza de código aberto e manutenção descomplicada facilita a replicabilidade, tornando-o um recurso valioso para aplicação em contextos escolares e acadêmicos para professores que desejam reproduzir e aprimorar esses experimentos em suas escolas e universidades, as quais geralmente já possuem ou podem adquirir os materiais necessários. Adicionalmente, as dissertações referenciadas neste trabalho de conclusão de curso oferecem um guia para a construção de um trilho de ar do zero, utilizando materiais de baixo custo, disponível em formato de manual. Embora este estudo centra no Movimento Retilíneo Uniforme (MRU) como exemplo, a abordagem pode ser expandida para experimentos mais complexos. A flexibilidade na escolha do sensor também permite a adaptação conforme a disponibilidade tecnológica.

Este trabalho destaca-se na facilidade de calibração das distâncias e troca entre os sensores, que pode ser realizada de forma intuitiva utilizando a câmera de um dispositivo móvel para identificar a emissão do infravermelho. A flexibilidade na fixação dos sensores expande o sistema para diversos experimentos, incluindo queda livre e plano inclinado, conforme a necessidade do usuário. As explicações detalhadas sobre os erros, obtidas pela comparação entre as medições do analisador lógico e as realizadas experimentalmente com o Arduino, demonstram a validação do protótipo e servem como exemplos práticos dessas imprecisões. Conforme abordado na seção 7, as possíveis causas desses erros incluem a escolha do microcontrolador, que utiliza um ressonador cerâmico em vez de um cristal de quartzo, resultando em perda de precisão. Consequentemente, a latência do software, decorrente do processamento, comunicação e atualização das informações no display LCD Keypad Shield, contribui para essas variações. Sugere-se, portanto, a substituição da função millis() pela micros() no código, uma alteração simples que não compromete a estrutura do programa e o desvio relativo percentual é aceitável para fins educacionais. Essas são hipóteses plausíveis que se relacionam diretamente com as limitações da arquitetura e do processamento do microcontrolador. Por fim, este trabalho visa oferecer às escolas de ensino básico e universidades uma alternativa mais acessível e econômica para a obtenção de aparatos experimentais, com potencialidades de experimentação

de MRUV, queda livre e plano inclinado. Existe também a possibilidade da construção de um suporte para o acoplamento dos sensores que podem ser feito em uma impressora 3D.

REFERÊNCIAS

ARDUINO Official Store. **Arduino documentation**, 2024a. Disponível em: https://docs.arduino.cc/ Acesso em: 28 nov. 2024.

ARDUINO Official Store. **ESP32/ESP8266**, 2025a. Disponível em: https://docs.arduino.cc/arduino-cloud/guides/esp32/ Acesso em: 10 jun. 2025.

ARDUINO Official Store. **Arduino Cloud**, 2025b. Disponível em: https://docs.arduino.cc/arduino-cloud/guides/esp32/ Acesso em: 10 mai. 2025.

CAVALCANTI, Deiverson Rodrigo Candido. **Análise do Movimento do Móvel Usando o Trilho de Ar e a Placa Arduino como Aquisição de Dados.** Dissertação (Mestrado) – Universidade Federal de Alagoas, Maceió. 2017. Disponível em: https://if.ufal.br/pt-br/pos-graduacao/mnpef/institucional/banco-de-dissertacoes/DeiversonRodrigoDissertao.pdf Acesso em: 28 nov. 2024.

CIDEPE**,** Centro Industrial de Equipamentos de Ensino e Pesquisa. Disponível em: https://www.cidepe.com.br/index.php/br/empresa Acesso em: 11 nov. 2024.

DOCA, Ricardo Helou; BISCUOLA, Gualter José; VILLAS BÔAS, Newton. **Tópicos de física**. v. 1. 21. ed. São Paulo: Saraiva Didático, 2012. p. 34-57.

DREASOURCELAB, 2025. **DSview User Guide v1.3.2.** Disponível em: https://www.dreamsourcelab.com/ Acesso em: 18 mai. 2025.

ELETROGATE, 2024. Disponível em: www.eletrogate.com Acesso em: 28 nov. 2024.

ECS Inc. International. **Quartz Crystals Vs. Ceramic Resonator**, 2025. Disponível em: https://ecsxtal.com/334-quartz-crystals-vs-ceramic-resonator/ Acesso em: 23 de maio de 2025.

ETT, empresa líder na indústria de microeletrônica, produção de produtos na área de microeletrônica e sistema de controle automático**. Manual of IR Sensor Switch E18-D80NK-N**, 2024. Disponível em: https://www.ett.co.th/ Acesso em: 28 de novembro de 2024.

MURATA, 2025. **Data Sheet (CSTCE16M0V53-R0)**. Disponível em: https://www.murata.com/en-us/api/pdfdownloadapi?cate=&partno=CSTCE16M0V53-R0. Acesso em: 23 mai. 2025.

OASISBR, Portal Brasileiro de Publicações e Dados Científicos em Acesso Aberto, 2024. **Mecanismo de busca multidisciplinar que permite o acesso gratuito à produção científica de autores vinculados a universidades e institutos de pesquisa brasileiros**. Disponível em: https://oasisbr.ibict.br/vufind/ Acesso em: 28 de novembro de 2024.

SANTOS, Vinicius da Cunha. **Construção de um trilho de ar com o uso do arduino**: uma proposta para o ensino de função afim e quadrática. 2019. 65 f. Dissertação (Mestrado Profissional em Matemática em Rede Nacional – PROFMAT) Instituto de

Ciências Exatas, Universidade Federal Rural do Rio de Janeiro, Seropédica, 2019. Disponível em: https://abrir.link/SHcmG Acesso em: 28 nov. 2024.

SOUZA, A. R. de *et al*. A placa Arduino: uma opção de baixo custo para experiências de física assistidas pelo PC**. Revista Brasileira de Ensino de Física**, v. 33, n. 1, p. 01-05, jan. 2011. Disponível em: https://abrir.link/NqYxE Acesso em: 28 nov. 2024.

TINKERCAD, empresa líder global em tecnologia de projeto e criação. **Introdução ideal à Autodesk**, 2024. Disponível em: https://www.tinkercad.com/dashboard Acesso em: 28 nov. 2024.

D-ROBOTICS. **LCD Keypad Shield (DFR0009)**, 2011. Disponível em https://l1nq.com/k04iR. Acesso em: 08 mai. 2025.

APÊNDICE A – Código Computacional para o Cronômetro com Sensores Infravermelho

O código foi desenvolvido para cronometrar a passagem de um objeto por uma sequência de quatro sensores e exibe os tempos no display LCD e no monitor serial. Nele, são registrados os valores de tempo.

```
#include <LiquidCrystal.h>

// Configuração do LCD Keypad Shield em que são definidos os pinos do Arduino que
//estão conectados no LCD, nesse caso do Mega 2560 é apenas encaixar LCD no Mega.
//Mais informações das definições das variáveis podem ser obtidas na biblioteca do LCD.

    const int rs = 8, en = 9, d4 = 4, d5 = 5, d6 = 6, d7 = 7;
    LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

// Mapeamento dos botões para definição no display
    #define btnRIGHT  0
    #define btnUP     1
    #define btnDOWN   2
    #define btnLEFT   3
    #define btnSELECT 4
    #define btnNONE   5

//Para a configuração dos sensores as portas digitais escolhidas são de acordo com
//Arduino Mega 2560 pois, comportam a função de interrupção. Na escolha de um
//diferente deve ser pesquisado na documentação do Arduino escolhido.
const int sensor1 = 3;
const int sensor2 = 18;
const int sensor3 = 19;
const int sensor4 = 20;


// Variáveis do tipo unsigned long (inteiros longos não negativos) para armazenar os
```

```
//momentos em que cada sensor foi acionado. O qualificador volatile é importante porque
//essas variáveis serão modificadas dentro das rotinas de interrupção (ISRs), e o
// compilador precisa saber que seus valores podem mudar fora do fluxo normal.

volatile unsigned long t1 = 0, t2 = 0, t3 = 0, t4 = 0;
volatile bool detected1 = false, detected2 = false, detected3 = false, detected4 = false;
int currentPage = 0;
bool measurementComplete = false;

// Distâncias entre sensores ajuste SI: (10cm = 0.10m)
const float distancia_S1_S2 = 0.10;
const float distancia_S2_S3 = 0.10;
const float distancia_S3_S4 = 0.10;

void setup() {

// Na inicialização do LCD pode ser escrito de forma diferente que se encontra nesse
//código, mudando no comando lcd.print(”.....”) .
 lcd.begin(16, 2);
 lcd.print("Sistema  Pronto");
 delay(1000);
 lcd.clear();
 lcd.print("Aguardando...");

// Configurando o pino do sensor como uma entrada. INPUT_PULLUP ativa um resistor
// interno de pull-up, o que significa que o pino estará em nível HIGH por padrão, e um
//sinal LOW será detectado quando o sensor for ativado (por exemplo, quando um
//objeto passa e fecha um contato). Isso é repetido para os quatro sensores.

 pinMode(sensor1, INPUT_PULLUP);
 pinMode(sensor2, INPUT_PULLUP);
 pinMode(sensor3, INPUT_PULLUP);
 pinMode(sensor4, INPUT_PULLUP);
```

```
 // A função attachInterrupt() é usada para configurar interrupções externas no Arduino.
//Ela permite que o microcontrolador responda imediatamente a eventos externos
//(como mudanças em pinos digitais) sem precisar verificar constantemente o estado do
//pino no loop principal. O uso de attachInterrupt() neste código é apropriado para a
//aplicação de medição de tempos entre ativações de sensores, seguindo a documentação
//do Arduino que está disponível nas referências. As interrupções garantem precisão na
//captura dos eventos enquanto permitem que o microcontrolador execute outras tarefas
//quando não está respondendo aos sensores.
  attachInterrupt(digitalPinToInterrupt(sensor1), sensor1_ISR, FALLING);
  attachInterrupt(digitalPinToInterrupt(sensor2), sensor2_ISR, FALLING);
  attachInterrupt(digitalPinToInterrupt(sensor3), sensor3_ISR, FALLING);
  attachInterrupt(digitalPinToInterrupt(sensor4), sensor4_ISR, FALLING);
  Serial.begin(9600);
}
void loop() {
  int button = readButtons();

// Esta parte do código é responsável por gerenciar a navegação entre as páginas do
//display LCD e reiniciar o sistema quando necessário, baseado nos botões pressionados
//pelo usuário.
  if (button != btnNONE) {
    if (button == btnRIGHT || button == btnUP) {
      currentPage = 1;
      updateDisplay();
    }
    else if (button == btnLEFT || button == btnDOWN) {
      currentPage = 0;
      updateDisplay();
    }
    else if (button == btnSELECT) {
      resetSystem();
    }
    delay(200);
```

```
  }

// verifica se todos os quatro sensores foram ativados e, nesse caso, executa os cálculos
// e atualizações finais do sistema.
  if (!measurementComplete && detected1 && detected2 && detected3 && detected4) {
    measurementComplete = true;
    calculateAndDisplay();
    updateDisplay();
  }
}
//Esta função lê o valor analógico do pino A0 (analogRead(0)), que é tipicamente usado
//pelo LCD Keypad Shield para detectar qual botão está pressionado. Cada botão possui
//um resistor diferente em série, resultando em diferentes níveis de tensão (e, portanto,
//diferentes leituras analógicas) quando pressionado. A função então compara o valor
//lido com faixas predefinidas para determinar qual botão foi pressionado e retorna a
//constante correspondente definida anteriormente (btnRIGHT, btnUP, etc.). Se o valor
//lido for maior que 1000, nenhum botão é considerado pressionado (btnNONE).

int readButtons() {
  int adc_key_in = analogRead(0);
  if (adc_key_in > 1000) return btnNONE;
  if (adc_key_in < 50)   return btnRIGHT;
  if (adc_key_in < 200)  return btnUP;
  if (adc_key_in < 400)  return btnDOWN;
  if (adc_key_in < 600)  return btnLEFT;
  if (adc_key_in < 800)  return btnSELECT;
  return btnNONE;
}

void updateDisplay() {
  lcd.clear();
  if (!measurementComplete) {
    if (detected1) {
```

```
    lcd.print("Medicao em curso");
    lcd.setCursor(0, 1);
    lcd.print("Sensores: ");
    lcd.print(detected1 + detected2 + detected3 + detected4);
    lcd.print("/4");
   }
   return;
  }
  if (currentPage == 0) {

   // Página 1 - Tempos iniciais
   lcd.setCursor(0, 0);
   lcd.print("T1:"); lcd.print(t1/1000.0, 3); lcd.print("s");
   lcd.setCursor(8, 0);
   lcd.print("T2:"); lcd.print(t2/1000.0, 3); lcd.print("s");
   lcd.setCursor(0, 1);
   lcd.print("T3:"); lcd.print(t3/1000.0, 3); lcd.print("s");
  } else {
   // Página 2 - Tempos finais
   lcd.setCursor(0, 0);
   lcd.print("T4:"); lcd.print(t4/1000.0, 3); lcd.print("s");
   lcd.setCursor(0, 1);
   lcd.print("Total:"); lcd.print((t4-t1)/1000.0, 3); lcd.print("s");
  }
}
void calculateAndDisplay() {

  // Cálculos dos deltas
  float deltaT1 = (t2 - t1) / 1000.0;
  float deltaT2 = (t3 - t2) / 1000.0;
  float deltaT3 = (t4 - t3) / 1000.0;
  // Exibe no Serial Monitor
  Serial.println("\n--- Dados Coletados ---");
```

```
  Serial.print("T1:"); Serial.print(t1/1000.0, 3); Serial.println("s");
  Serial.print("T2:"); Serial.print(t2/1000.0, 3); Serial.println("s");
  Serial.print("T3:"); Serial.print(t3/1000.0, 3); Serial.println("s");
  Serial.print("T4:"); Serial.print(t4/1000.0, 3); Serial.println("s");
  Serial.print("Total:"); Serial.print((t4-t1)/1000.0, 3); Serial.println("s");
}

void resetSystem() {
  t1 = t2 = t3 = t4 = 0;
  detected1 = detected2 = detected3 = detected4 = false;
  measurementComplete = false;
  lcd.clear();
  lcd.print("Sistema reiniciado");
  delay(1000);
  lcd.clear();
  lcd.print("Aguardando...");
  Serial.println("Sistema reiniciado. Aguardando nova medicao...");
}
// Estas quatro funções são Rotinas de Serviço de Interrupção (ISRs) que são acionadas
//automaticamente quando ocorre uma mudança de estado nos pinos dos sensores
//configurados. Cada função segue a mesma estrutura lógica, mas trata um sensor
//específico. Essa estrutura é Comum das ISRs todas as funções possuem o mesmo padrão
//de funcionamento: Verificação de primeira ativação; Utiliza-se uma condição if
//(!detectedX) para garantir que só registre o tempo na primeira ativação de cada sensor
//Isso evita múltiplos registros caso o sensor seja ativado várias vezes pois quando o
//objeto retorna no trilho de ar corre esse risco. O registro do Timestamp armazena o
//tempo atual (em milissegundos) usando millis() nas variáveis t1 a t4 estas variáveis
//são do tipo volatile unsigned long para garantir acesso seguro entre a ISR e o loop
//principal. A marca de Detecção setam a flag correspondente (detected1 a detected4)
//para true indica ao sistema que este sensor específico foi ativado. Atualização do
//Display updateDisplay() para refletir imediatamente a mudança no LCD que mostra ao
//usuário quais sensores já foram ativados.
```

```
void sensor1_ISR() { if (!detected1) { t1 = millis(); detected1 = true; updateDisplay(); }
}
void sensor2_ISR() { if (!detected2) { t2 = millis(); detected2 = true; updateDisplay(); }
}
void sensor3_ISR() { if (!detected3) { t3 = millis(); detected3 = true; updateDisplay(); }
}
void sensor4_ISR() { if (!detected4) { t4 = millis(); detected4 = true; updateDisplay(); }
}
```