

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
ESCOLA POLITÉCNICA E DE ARTES
GRADUAÇÃO EM ENGENHARIA DA COMPUTAÇÃO



SISTEMA DE CONTROLE DE TEMPERATURA PARA CHURRASQUEIRA

MARCUS VINICIUS DE SOUZA SANTOS

GOIÂNIA, 2023

MARCUS VINICIUS DE SOUZA SANTOS

SISTEMA DE CONTROLE DE TEMPERATURA PARA CHURRASQUEIRA

Trabalho de Conclusão de Curso apresentado à Escola Politécnica e de Artes, da Pontifícia Universidade Católica de Goiás, como parte dos requisitos para a obtenção do título de Bacharel em Engenharia de Computação.

Orientador(a): Prof.^a Gustavo Vinhal

GOIÂNIA, 2023

MARCUS VINICIUS DE SOUZA SANTOS

SISTEMA DE CONTROLE DE TEMPERATURA PARA CHURRASQUEIRA

Este Trabalho de Conclusão de Curso julgado adequado para obtenção do título de Bacharel em Engenharia de Computação, e aprovado em sua forma final pela Escola de Politécnica e de Artes, da Pontifícia Universidade Católica de Goiás, em ____/____/_____.

Prof.^a Ma. Ludmilla Reis Pinheiro dos Santos
Coordenadora de Trabalho de Conclusão de Curso

Banca Examinadora:

Orientador: Prof. Gustavo Siqueira Vinhal

Prof. Me. Fernando Gonçalves Abadia

Prof. Me. Rafael Leal Martins

GOIÂNIA
2023

Dedico este trabalho às pessoas que colaboraram com seu desenvolvimento, de maneira direta ou indireta e que sempre me motivaram durante essa jornada.

AGRADECIMENTOS

Gostaria de expressar minha profunda gratidão a todos os professores da PUC Goiás que me guiaram ao longo de toda a minha jornada acadêmica. Em especial, quero destacar o Prof. Gustavo Siqueira Vinhal, que esteve ao meu lado desde o momento em que esta ideia surgiu, lá na disciplina de Projeto de Componentes e Circuitos, até o desfecho deste trabalho. Sua orientação foi fundamental para o sucesso deste projeto. Não posso deixar de mencionar minha família e amigos, que sempre esteve ao meu lado, oferecendo apoio, especialmente nos momentos mais desafiadores.

A todos vocês, meu mais sincero e profundo agradecimento.

RESUMO

Este trabalho tem como objetivo realizar um estudo acerca do sistema de controle de temperatura, com foco na sua aplicação em uma churrasqueira. Para atingir esse propósito, foi conduzida uma análise sobre o assunto e, com base nessa pesquisa, foi desenvolvido uma implementação utilizando um Arduino. A implementação inclui componentes que simplificam o processo, como o módulo leitor de temperatura MAX6675, que fornece a leitura da temperatura de um termopar já em grau Celsius. Por meio dessa implementação, foi possível analisar as operações típicas de um sistema de controle de temperatura, monitorar a temperatura do ambiente, verificar se esta está de acordo com a temperatura desejada e efetuar os ajustes necessários na temperatura ambiente por meio de atuadores.

Palavras chaves: Sistema controle de temperatura, Churrasqueira, Arduino, Modulo MAX6675; Termopar; Atuador.

ABSTRACT

The aim of this work is to conduct a study on the temperature control system, with a focus on its application in a barbecue. To achieve this objective, an analysis was conducted on the subject, and based on this research, an implementation was developed using an Arduino. The implementation includes components that simplify the process, such as the MAX6675 temperature sensor module, which provides temperature readings from a thermocouple already in Celsius degrees. Through this implementation, we were able to analyze the typical operations of a temperature control system, monitor the ambient temperature, check if it aligns with the desired temperature, and make necessary adjustments to the ambient temperature using actuators.

Keywords: Temperature Control System; Barbecue; Arduino; MAX6675 Module; Thermocouple; Actuator.

LISTA DE ILUSTRAÇÕES

Figura 1: Arduino Uno R3.....	21
Figura 2: Modulo MAX66 e termopar tipo K.....	22
Figura 3: Ventoinha120mm 12v.....	22
Figura 4: Mini Bomba de Água 12V RS385.....	23
Figura 5: Modulo relé 2 canais 5V.....	23
Figura 6: Display LCD 16x2.....	24
Figura 7: Interface Arduino IDE.....	25
Figura 8: Interface do ambiente de circuitos Tinkercad.....	25
Figura 9: Circuito adaptado no Tinkercad.....	26
Figura 10: Display LCD recebendo todas entradas para seu funcionamento.....	27
Figura 11: Protobord com os componentes de entrada.....	27
Figura 12: Acionamento dos atuadores.....	28
Figura 13: Placa Arduino com todos os componentes conectados.....	28
Figura 14: inclusão de bibliotecas.....	29
Figura 15: Página de download GitHub.....	29
Figura 16: Passo para adicionar biblioteca.....	30
Figura 17: Janela para selecionar biblioteca.....	30
Figura 18: Simulação do sistema em estado baixo.....	37
Figura 19: Simulação do sistema em estado estável.....	37
Figura 20: Simulação do sistema em estado alto.....	38

LISTA DE ABREVIATURAS

GND – Graduated Neutral Density filter
PWM – Pulse Width Modulation
USB – Universal Serial Bus
RPM – Rotações Por Minuto
VAC – Tensão Alternada
VCC – Tensão em Corrente Contínua
IDE – Integrated Development Environment
LCD - Liquid Crystal Display

SUMÁRIO

1. INTRODUÇÃO	12
1.1 Objetivos	13
1.1.1 Geral.....	13
1.1.2 Específico.....	13
1.2 Organização do trabalho	14
2. REFERENCIAL TEÓRICO.....	15
2.1 Churrasco.....	15
2.2 Tipos de Churrasqueiras	16
2.3 Sistema de Controle de Temperatura	17
2.4 Arduino	18
2.5 Trabalhos Correlatos	19
3. MATERIAIS E MÉTODOS.....	21
4. DESENVOLVIMENTO PRÁTICO.....	23
4.1 Componentes utilizados	23
4.2 Softwares Utilizados	27
4.3 Implementação.....	28
5. CONSIDERAÇÕES FINAIS	39
5.1 Trabalhos futuros	39
5.2 Melhorias técnicas.....	39
REFERÊNCIAS.....	40
APÊNDICES.....	44
APÊNDICE A – CONFIGURAÇÃO DOS PINOS.....	44

APÊNDICE B – VARIÁVEIS DE CONTROLE	44
APÊNDICE C – INICIANDO OBJETOS	45
APÊNDICE D – FUNÇÃO SETUP	45
APÊNDICE E – FUNÇÃO LOOP	46
APÊNDICE F – FUNÇÃO INTERRUPTÃO POR TIMER.....	47
APÊNDICE G – FUNÇÃO INTERRUPTÃO PELO BOTÃO MAIS	48
APÊNDICE H – FUNÇÃO INTERRUPTÃO PELO BOTÃO MENOS	48
APÊNDICE I – FUNÇÃO SIMULA TEMPERATURA.....	49

1. INTRODUÇÃO

Segundo Nogueira (2020), o churrasco é, até hoje, reconhecido como um ritual ancestral que mantém a tradição de reunir pessoas em torno de uma fogueira. Embora no passado tenha sido essencial para celebrar conquistas e marcar momentos cruciais na sobrevivência da humanidade, nos dias de hoje, o seu propósito fundamental continua sendo unir indivíduos em torno do prazer da convivência e da boa comida.

A temperatura é um fator essencial para o sucesso de um churrasco, afetando diversos aspectos do processo de preparo. Manter uma temperatura adequada na grelha é crucial para garantir o cozimento uniforme da carne, evitando partes cruas ou excessivamente cozidas. Além disso, a temperatura influencia diretamente a textura desejada da carne, proporcionando suculência em temperaturas mais baixas e uma superfície mais crocante em temperaturas mais altas. Além do aspecto culinário, o controle preciso da temperatura é vital para a segurança alimentar. Utilizar um termômetro de carne para verificar se a temperatura interna atingiu níveis seguros ajuda a prevenir doenças transmitidas por alimentos. Em resumo, a atenção à temperatura durante o churrasco não apenas assegura resultados gustativos ideais, mas também garante que a carne seja preparada de maneira segura e saudável (HANNA® INSTRUMENTS, 2019).

Um sistema de controle de temperatura desempenha um papel central em vários setores. Ele é essencial para manter a segurança, preservar a qualidade de produtos e amostras, otimizar a eficiência energética e criar ambientes de conforto. Seja em processos industriais, edifícios, laboratórios, ou instalações de saúde, o controle preciso da temperatura é crucial para proteger pessoas, equipamentos e bens, bem como para garantir a eficácia de experimentos e processos de fabricação. Em resumo, o controle de temperatura é um componente fundamental em diversas áreas, afetando a segurança, a qualidade e a eficiência (CRAVO, 2022).

Um sistema de controle de temperatura é projetado para manter a temperatura de um ambiente ou processo dentro de uma faixa desejada. Isso é feito por meio de um processo de feedback, onde um sensor de temperatura mede constantemente a temperatura atual. Os dados do sensor são enviados a um controlador, que compara a temperatura medida com a temperatura de referência predefinida. Com base nessa diferença, o controlador envia

comandos para um dispositivo chamado atuador, que ajusta o sistema de aquecimento ou resfriamento para corrigir a temperatura e trazê-la de volta para a faixa desejada. Esse ciclo de medição e correção contínuo permite que o sistema de controle de temperatura mantenha um ambiente ou processo dentro das especificações desejadas (CRAVO, 2022).

O Arduino está relacionado à sua acessibilidade e facilidade de uso, tornando a eletrônica e a programação acessíveis a uma ampla gama de pessoas. Além disso, o Arduino permite a rápida prototipagem de projetos, economizando tempo e recursos. A comunidade ativa e o caráter de código aberto incentivam a colaboração e o compartilhamento de conhecimento. O Arduino é usado em diversas aplicações, de automação residencial a robótica, promovendo a versatilidade. Sua capacidade de estimular a criatividade e inovação resulta no desenvolvimento de novas tecnologias e soluções personalizadas em eletrônica e automação (MAKIYAMA, 2022).

No contexto apresentado, este projeto busca responder à seguinte pergunta: - **É possível implementar um sistema de controle de temperatura utilizando Arduino?**

1.1 Objetivos

1.1.1 Geral

Desenvolver um sistema de controle utilizando a plataforma Arduino para monitorar e regular a temperatura em churrasqueiras. A implementação inclui sensores e algoritmos de controle, visando otimizar o processo de churrasco, proporcionando praticidade e segurança aos usuários."

1.1.2 Específico

Analisar microcontroladores, com destaque para o Arduino, explorando seus potenciais aplicações práticas. A implementação de interrupções, tanto internas quanto externas, é considerada para aprimorar o desempenho do sistema. Adicionalmente, pretende-se realizar simulações do projeto em um ambiente virtual, proporcionando uma avaliação abrangente antes da implementação física.

1.2 Organização do trabalho

a organização segue a seguinte forma: no Capítulo 2, é realizada uma revisão dos tópicos importantes que contribuíram para o desenvolvimento do estudo, descrevendo as pesquisas conduzidas e o conhecimento adquirido para a conclusão do projeto. O Capítulo 3 aborda a organização do projeto de pesquisa e suas diferentes partes. No Capítulo 4, são apresentados os detalhes da execução do projeto, incluindo suas partes, as soluções propostas, tanto em termos de código quanto de hardware, e a explicação de cada ponto. No Capítulo 5, são apresentadas as conclusões obtidas a partir dos experimentos Feitos, bem como recomendações para aprimoramentos em trabalhos futuros.

2. REFERENCIAL TEÓRICO

Este capítulo traz conceitos e definições necessárias, para a compreensão do trabalho.

2.1 Churrasco

O churrasco é uma técnica de preparação de carne que envolve grelhar ou assar diretamente sobre uma fonte de calor, como carvão, madeira ou gás, geralmente ao ar livre. Essa prática culinária é valorizada em diversas culturas, especialmente na brasileira, argentina e uruguaia, e é considerada uma tradição social que reúne amigos e familiares ao redor de uma churrasqueira (DOS SANTOS, 2023).

A seleção de cortes de carne desempenha um papel crucial, com opções populares incluindo picanha, contrafilé, costelas, frango e linguiça. Cada corte possui características únicas de sabor e textura, sendo fundamental prestar atenção à qualidade da carne, com ênfase na marmorização (gordura intrínseca na carne) para garantir suculência e sabor. A preparação da carne muitas vezes envolve o uso de marinadas, temperos secos, como pimenta e alho, ou simplesmente sal grosso, mas é importante evitar excessos para não mascarar o sabor natural (DOS SANTOS, 2023).

Ao escolher entre carvão e churrasqueira a gás, considera-se a tradição do sabor defumado conferido pelo carvão versus a conveniência do gás. A preparação da churrasqueira, seja com carvão ou gás, requer atenção ao acendimento correto, utilizando métodos como acendedores de carvão, maçarico ou o "método do vulcão". A temperatura da churrasqueira é ajustada controlando a altura da grelha, a quantidade de carvão ou gás, influenciando diretamente no resultado do cozimento (DOS SANTOS, 2023).

Quanto às técnicas de grelha, a escolha entre o método direto, posicionando a carne sobre as brasas, e o método indireto, assando ao lado das brasas, depende do tipo de carne e sua espessura. Inclinando a carne na grelha ajuda a controlar o cozimento e evitar chamas excessivas, garantindo que o sabor não seja comprometido. O tempo de cozimento varia conforme o tipo de carne e o ponto desejado, podendo ser monitorado com um termômetro de carne (DOS SANTOS, 2023).

Além da carne, é comum incluir acompanhamentos como farofa, pães e molhos, como chimichurri e vinagrete, para enriquecer a experiência. Saladas frescas, legumes grelhados, queijos e vegetais também são apreciados como complementos (DOS SANTOS, 2023).

2.2 Tipos de Churrasqueiras

A escolha do tipo de churrasqueira envolve diversos fatores, como espaço disponível, orçamento, preferências de sabor e necessidades específicas do local de uso. Cada tipo de churrasqueira apresenta vantagens e desvantagens, atendendo a diferentes estilos de churrasco (QUINTOANDAR, 2023).

A churrasqueira de alvenaria é um modelo tradicional construído com tijolos à vista, reconhecido por sua durabilidade e resistência ao calor. Ideal para cozinhar grandes quantidades de carne, demanda um investimento significativo em construção e espaço, sendo preferida por aqueles que buscam uma churrasqueira permanente e robusta (QUINTOANDAR, 2023).

A opção pré-moldada é prática e econômica, feita de blocos de concreto com estrutura de chaminé e coifa eficientes. Pronta para instalação, pode ser personalizada com diversos materiais de acabamento e desmontada conforme necessário (QUINTOANDAR, 2023).

A churrasqueira embutida é uma escolha inteligente para espaços limitados, integrando-se à parede e proporcionando um visual moderno. Adequada para áreas gourmet e varandas, é mais acessível em custo e montagem rápida em comparação com as de alvenaria (QUINTOANDAR, 2023).

A churrasqueira de aço ou portátil é prática e versátil, leve e desmontável, facilitando o transporte para acampamentos ou churrascos com amigos. Sem necessidade de obras, é uma opção econômica e conveniente (QUINTOANDAR, 2023).

A churrasqueira a bafo é ideal para assar lentamente carnes, peixes e frangos, reduzindo o consumo de carvão. Apresenta uma tampa para criar um ambiente de cozimento abafado, embora não seja adequada para espetos e não seja recomendada para ambientes internos (QUINTOANDAR, 2023).

O modelo a gás é reconhecido por sua ecologia, facilidade de limpeza e controle preciso da temperatura. Disponível em diferentes configurações, como embutida, de bancada ou portátil, requer espaço e suprimento de gás, sendo mais caro em comparação com versões elétricas (QUINTOANDAR, 2023).

A churrasqueira elétrica é prática para espaços reduzidos, como sacadas de apartamento, fácil de instalar e transportar, gerando menos fumaça. No entanto, alguns consideram que a carne não adquire o sabor autêntico de churrasco nesse modelo (QUINTOANDAR, 2023).

A churrasqueira giratória é elétrica, destacando-se pela rotação automática dos espetos para um cozimento uniforme da carne. Embora possa não agradar esteticamente a todos, oferece a conveniência de não exigir a virada manual dos espetos, permitindo que o cozinheiro se concentre em outros aspectos do preparo (QUINTOANDAR, 2023).

2.3 Sistema de Controle de Temperatura

Um sistema de controle de temperatura é essencial para regular a temperatura em diferentes ambientes ou dispositivos, assegurando que ela permaneça dentro dos limites desejados (BERTULUCCI, 2016). Nesse contexto, os elementos principais desempenham papéis específicos para garantir um controle eficaz.

O sensor de temperatura é o ponto de partida, encarregado de medir a temperatura atual do ambiente ou dispositivo. Existem vários tipos de sensores, como termopares, termistores e sensores de temperatura de platina (PT100). Esses dispositivos convertem a temperatura em sinais elétricos, como voltagem,

resistência ou corrente, fornecendo dados cruciais para o controle (BERTULUCCI, 2016).

Dentro dessa variedade de sensores, o Termopar Tipo K se destaca. Amplamente utilizado em diversas aplicações devido à sua habilidade de medir temperaturas em uma vasta faixa, de -270°C até 1.372°C , o Termopar Tipo K é essencial para sistemas de controle de temperatura em setores que vão desde a indústria alimentícia até a indústria aeroespacial (BORGES, 2019).

O controlador é o cérebro do sistema, recebendo os dados do sensor de temperatura e comparando-os com o valor de referência (setpoint) estabelecido pelo usuário. O Arduino, uma plataforma de prototipagem eletrônica de código aberto, é um exemplo de controlador que simplifica a criação de projetos eletrônicos interativos e controlados por computador (BAEZA, 2009).

Após a análise do controlador, o sistema parte para a ação com o atuador. Este componente crucial executa ações para ajustar a temperatura conforme as decisões do controlador. Se a temperatura estiver abaixo do setpoint, um sistema de aquecimento é acionado; se estiver acima, entra em ação um sistema de resfriamento. Diversos dispositivos, como resistências elétricas ou compressores de ar-condicionado, podem ser utilizados como atuadores, dependendo da natureza do sistema (BERTULUCCI, 2016).

A escolha entre sistemas de malha aberta e malha fechada é vital para a eficácia do controle. Em um sistema de malha fechada, o controlador constantemente compara a temperatura atual com o valor de referência, realizando ajustes contínuos para manter a temperatura desejada. Essa abordagem é mais precisa e amplamente utilizada em aplicações críticas, garantindo um controle mais efetivo (BERTULUCCI, 2016).

Finalmente, a interface de controle é a interface que permite aos operadores interagirem com o sistema. Isso inclui a configuração do setpoint desejado, a visualização da temperatura atual e a monitorização do status do sistema. A interface de controle pode variar de simples, como a de um termostato residencial, a mais avançada em contextos industriais (BERTULUCCI, 2016).

2.4 Arduino

O Arduino é uma plataforma de prototipagem eletrônica que combina hardware e software de código aberto. Criado para ser acessível e fácil de usar,

o Arduino é amplamente utilizado por entusiastas, makers e profissionais em uma variedade de campos.

No núcleo do sistema está uma placa de circuito integrado (microcontrolador), como o Arduino Uno ou Arduino Mega. Essas placas são projetadas para serem programadas e executar tarefas específicas, interagindo com o ambiente físico por meio de pinos de entrada/saída.

O desenvolvimento de software ocorre no Ambiente de Desenvolvimento Integrado (IDE) Arduino, onde os usuários escrevem código em uma linguagem simplificada baseada em C/C++. Este código é então carregado na placa Arduino, permitindo que ela execute diversas funções.

Os pinos de entrada/saída (I/O) nas placas Arduino permitem a conexão com uma variedade de dispositivos, como sensores e atuadores. Isso possibilita a criação de projetos interativos que respondem a estímulos do ambiente.

O ecossistema Arduino é enriquecido por uma ampla gama de sensores e módulos pré-fabricados, que facilitam a expansão das capacidades das placas. Sensores de temperatura, luz, movimento, entre outros, estão prontamente disponíveis para uso em projetos.

Além disso, o Arduino possui uma comunidade ativa e global. Essa comunidade compartilha projetos, tutoriais e fornece suporte, criando um ambiente propício para a aprendizagem e a colaboração.

Os usos do Arduino são diversos, desde projetos de automação residencial e robótica até aplicações em arte interativa e educação. Sua versatilidade e acessibilidade o tornam uma ferramenta valiosa para transformar ideias em protótipos funcionais, impulsionando a inovação no campo da eletrônica (Makiyama, 2022).

2.5 Trabalhos Correlatos

Há uma grande quantidade de trabalhos que implementam sistemas de controle de temperatura. Araújo (2020) o utiliza para o controle de temperatura para estufas.

Oliveira Junior, Oliveira e Marques (2019) mostram a utilização desse sistema para monitorar e controlar a temperatura de um aquário.

Krinski (2018) utilizou o sistema de controle de temperatura para automatizar e controlar o processo de cozimento (brasagem) de cereais

maltados, fervura e resfriamento de mosto, para a criação de líquidos fermentados alcoólicos, como a cerveja.

Já Borges Filho (2020) utilizando esse sistema projetou um sistema de para monitoramento e controle remoto do processo de fermentação de cerveja artesanal.

3. MATERIAIS E MÉTODOS

A pesquisa, em termos de sua natureza, constitui um resumo temático. Seu propósito é a explanação da esfera de conhecimento relacionada ao projeto, visando à análise e correlação das áreas de conhecimento pertinentes ao projeto, contribuindo para a compreensão das causas e explicações envolvidas (WAZLAWICK, 2014).

Em relação aos objetivos, a pesquisa é composta por uma abordagem exploratória e outra descritiva. A parte descritiva se concentra na obtenção de dados mais robustos acerca do tema, sem intervenção do pesquisador, com o intuito de apresentar os fatos de forma objetiva e imparcial (WAZLAWICK, 2014).

A fase exploratória da pesquisa pode ser considerada como o ponto de partida no processo investigativo, muitas vezes desprovida de um objetivo ou hipótese previamente definidos por parte do autor (WAZLAWICK, 2014). Essa abordagem é flexível, uma vez que explora diversos aspectos relacionados aos fenômenos ou fatos estudados (GIL, 2017), possibilitando a formulação de novas hipóteses.

No que tange aos procedimentos técnicos, a pesquisa adotará uma abordagem bibliográfica e experimental. A pesquisa bibliográfica demanda a análise de artigos, teses, livros e outras fontes de referência. A pesquisa experimental, por sua vez, implica a manipulação de variáveis experimentais sob o controle do pesquisador (WAZLAWICK, 2014).

A pesquisa experimental envolve a provocação de alterações no ambiente de estudo, seguida da análise dos resultados para verificar se estão em conformidade com as expectativas (WAZLAWICK, 2014). Além disso, conforme Gil (2017), uma pesquisa experimental deve incluir pelo menos um experimento que seja responsável por criar a situação investigada.

Quanto à abordagem, a pesquisa é qualitativa, uma vez que emprega métodos que visam a aplicação de medidas padronizadas e sistemáticas, facilitando a comparação por meio da análise de dados estatísticos (AUGUSTO, 2012).

No que diz respeito aos métodos de pesquisa, nossa abordagem começará com uma análise bibliográfica, seguida por experimentos. Nossa revisão bibliográfica envolverá a consulta de livros, revistas, dissertações e artigos acadêmicos pertinentes ao tema do sistema de controle de temperatura.

Inicialmente, nos aprofundaremos nos conceitos relacionados aos sistemas de controle de temperatura, com base em fontes confiáveis. Em seguida, avançaremos para a exploração das estratégias de implementação desse sistema, utilizando a linguagem de programação C++.

Os experimentos serão conduzidos em um ambiente controlado, obedecendo a todas as medidas de segurança necessárias. O sistema utilizado para a implementação do código será um computador com o sistema operacional Windows 11, equipado com as seguintes configurações: Processador Ryzen 7 2700x, Placa de vídeo RX 6700XT 12GB, SSD M.2 de 1TB, 16 GB de memória RAM DDR4 3600Mhz. Para a construção do protótipo, serão utilizados componentes como o Arduino Uno R3 Atmega 328p, fonte de alimentação de 12V, 2 relés de 5V 10A, sensor de temperatura do tipo termopar tipo K, ventoinha de 12V, minibomba de água Arduino de 12V RS385, Display LCD 16X2 e tubulações de borracha.

4. DESENVOLVIMENTO PRÁTICO

Na fase experimental, o projeto visa criar um protótipo em escala reduzida, integrando componentes simplificados, para simular o desempenho do nosso sistema de controle de temperatura. Dessa forma, poderemos executar as funções principais do sistema com eficiência, tendo em mente as dificuldades e limitações dos componentes utilizados.

4.1 Componentes utilizados

Para desenvolvimento do trabalho será utilizado um Arduino Uno R3, um sensor de temperatura, teremos atuadores para poder fazer o controle da temperatura, uma ventoinha para subir a temperatura e uma bomba de água para diminuir, relés para acionar os atuadores.

De acordo com o Arduino o Uno R3 é um microcontrolador, baseado no microcontrolador ATmega328P, que pode ser visto na figura 1, ele pode ser alimentado por meio de uma conexão USB (5V) ou uma fonte externa de 7-12V. O Arduino Uno R3 possui 14 pinos digitais, dos quais 6 podem ser usados como saídas PWM (Modulação por largura de pulso) e 6 entradas analógicas. Seu microcontrolador ATmega328P tem 32KB de memória flash para armazenar o código do programa e opera com um oscilador cerâmico de 16 MHz.



Figura 1: Arduino Uno R3. Fonte: (https://m.media-amazon.com/images/I/51hehXTFKVL._SX522_.jpg)

Este modelo oferece uma porta USB para programação e alimentação, juntamente com a capacidade de ser alimentado por um adaptador AC-DC ou uma bateria compatível. O Arduino Uno R3 também possui um conector ICSP para programação e atualização de firmware, bem como um botão de reset que permite reiniciar o microcontrolador.

Em termos de capacidades elétricas, os pinos digitais do Arduino Uno R3 podem fornecer uma corrente de saída máxima de 20 mA por pino, enquanto os pinos 3.3V podem fornecer até 50 mA. Além disso, possui 2KB de memória SRAM e 1KB de memória EEPROM.

Para a medição de temperatura, será utilizado o módulo de leitura Max6675 e um termopar tipo K, mostrados na figura 2, ele é um dispositivo compacto e preciso projetado para medir temperaturas entre 0°C e 1024°C usando termopares do tipo K. Ele oferece comunicação SPI (3 fios), detecção de quebra do termopar e é compatível com microcontroladores como Arduino, Raspberry Pi e PIC. Inclui bornes para conexão da sonda de temperatura e é fácil de usar. Suporta alimentação de 3,0V a 5,5V, com uma corrente de operação de 50mA.

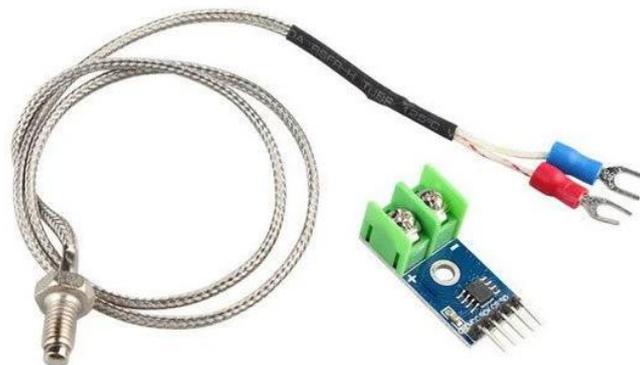


Figura 2: Modulo MAX66 e termopar tipo K. Fonte: (https://www.eletrogate.com/termopar-tipo-k-0-a-800c-modulo-de-leitura-max6675?utm_content=&utm_term=&gad=1)

Para aumentar a temperatura será usado uma ventoinha 120mm12v, mostrada na figura 3, essa ventoinha trabalhar com a tensão de 12V, corrente 0,28 Amperes, potência de 3,36 W, rotação de 2400 Rpm



Figura 3: Ventoinha120mm 12v. Fonte: (https://http2.mlstatic.com/D_NQ_NP_657336-MLB46776243067_072021-O.webp)

Para diminuir a temperatura, será usado a Mini Bomba de Água 12V RS385, que pode ser vista na figura 4, Ela tem alto fluxo, impulsionando até 2 litros por minuto, possui tamanho compacto, pesa 100g e opera com uma faixa de tensão de 9V a 15V, oferecendo uma elevação máxima de até 3 metros e altura de aspiração de até 2 metros.



Figura 4: Mini Bomba de Água 12V RS385. Fonte: (https://www.usinainfo.com.br/1017396-thickbox_default/mini-bomba-de-agua-para-arduino-12v-rs385-2lmin.jpg)

Para o acionamento dos atuadores será usado um modulo relé 2 canais 5V, mostrado na figura 5, esse modulo possui 2 canais independentes, sua

tensão de sinal é de 5V, ele suporta cargas de 30VCC a 10A ou 250VCA a 10A e tem um tempo de resposta rápido, variando entre 5 a 10ms.



Figura 5: Modulo relé 2 canais 5V. Fonte: (<https://www.piscaled.com.br/modulo-rele-2-canais-5v-220v/110v-10a>)

O Display 16x2 será utilizado para mostrar as informações, esse possui 16 caracteres por 2 linhas, possui uma alimentação 5V.

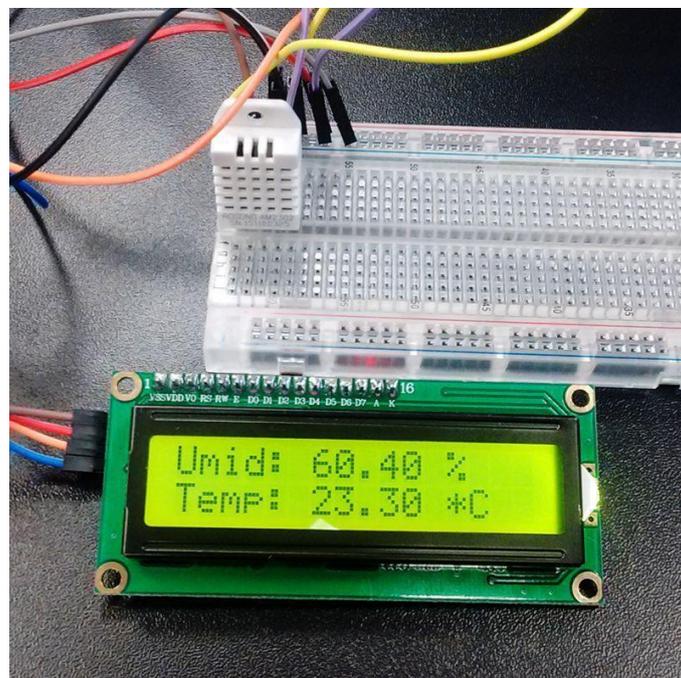
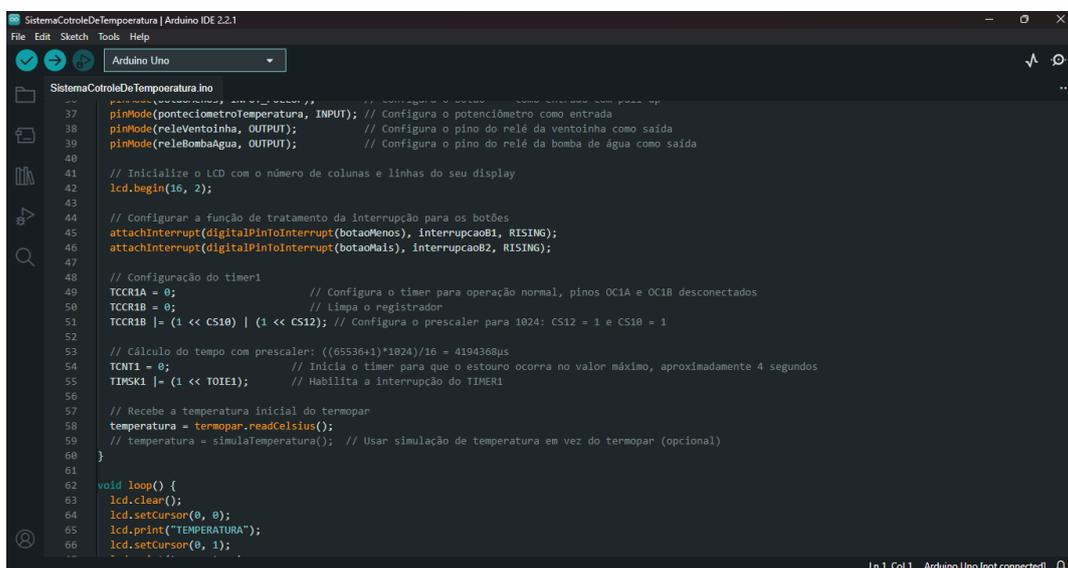


Figura 6: Display LCD 16x2. Fonte: (https://images.tcdn.com.br/img/img_prod/715570/display_lcd_16x2_verde_com_backlight_94_3_20201213232555.jpg)

4.2 Softwares Utilizados

Na implementação, foi utilizado a Arduino IDE, versão 2.2.2, uma plataforma de desenvolvimento de código aberto. Esta ferramenta facilita a gravação de código na placa de maneira intuitiva, proporcionando uma experiência aprimorada em comparação ao ambiente de desenvolvimento padrão das placas Arduino. A Arduino IDE oferece uma interface de usuário mais amigável, um editor de código mais robusto, suporte a depuração, gerenciamento simplificado de bibliotecas e compatibilidade com uma ampla gama de placas.



```
SistemaControleDeTemperatura.ino
37 pinMode(potenciometroTemperatura, INPUT); // Configura o potenciômetro como entrada
38 pinMode(releVentoinha, OUTPUT); // Configura o pino do relé da ventoinha como saída
39 pinMode(releBombaAgua, OUTPUT); // Configura o pino do relé da bomba de água como saída
40
41 // Inicialize o LCD com o número de colunas e linhas do seu display
42 lcd.begin(16, 2);
43
44 // Configurar a função de tratamento da interrupção para os botões
45 attachInterrupt(digitalPinToInterrupt(botaoMenos), InterrupcaoB1, RISING);
46 attachInterrupt(digitalPinToInterrupt(botaoMais), InterrupcaoB2, RISING);
47
48 // Configuração do timer1
49 TCCR1A = 0; // Configura o timer para operação normal, pinos OC1A e OC1B desconectados
50 TCCR1B = 0; // Limpa o registrador
51 TCCR1B |= (1 << CS10) | (1 << CS12); // Configura o prescaler para 1024: CS12 = 1 e CS10 = 1
52
53 // Cálculo do tempo com prescaler: ((65536+1)*1024)/16 = 4194368µs
54 TCNT1 = 0; // Inicia o timer para que o estouro ocorra no valor máximo, aproximadamente 4 segundos
55 TIMSK1 |= (1 << TOIE1); // Habilita a interrupção do TIMER1
56
57 // Recebe a temperatura inicial do termopar
58 temperatura = termopar.readCelsius();
59 // temperatura = simulaTemperatura(); // Usar simulação de temperatura em vez do termopar (opcional)
60 }
61
62 void loop() {
63 lcd.clear();
64 lcd.setCursor(0, 0);
65 lcd.print("TEMPERATURA");
66 lcd.setCursor(0, 1);
67 }
```

Figura 7: Interface Arduino IDE. Arquivo Pessoal

Também foi utilizado o Tinkercad para a montagem e teste do circuito e código, ele é uma plataforma online desenvolvida pela Autodesk, projetada para a criação de modelos 3D e a simulação de circuitos eletrônicos. Esta ferramenta oferece aos usuários uma extensa biblioteca de componentes eletrônicos prontos para uso, permitindo a montagem e simulação de circuitos de forma intuitiva.

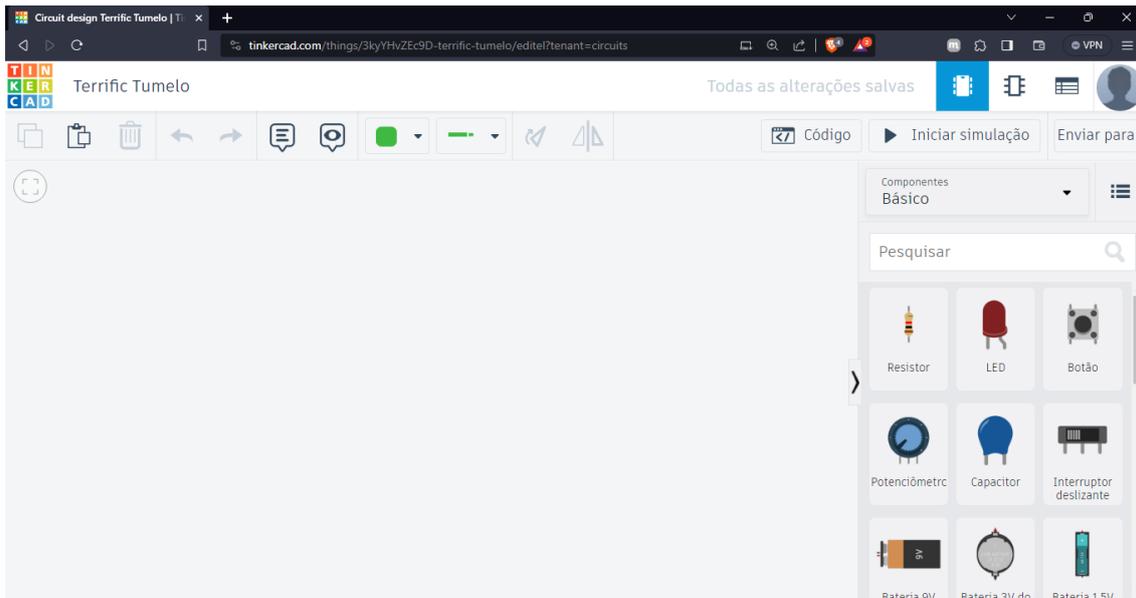


Figura 8: Interface do ambiente de circuitos Tinkercad. Arquivo Pessoal

4.3 Implementação

A implementação se iniciou pela montagem dos componentes na plataforma Tinkercad, por conta de limitações da plataforma, não foi possível adicionar alguns componentes previstos no projeto, porém foram feitas adaptações para que esses componentes fossem substituídos por outros que funcionam de forma similar.

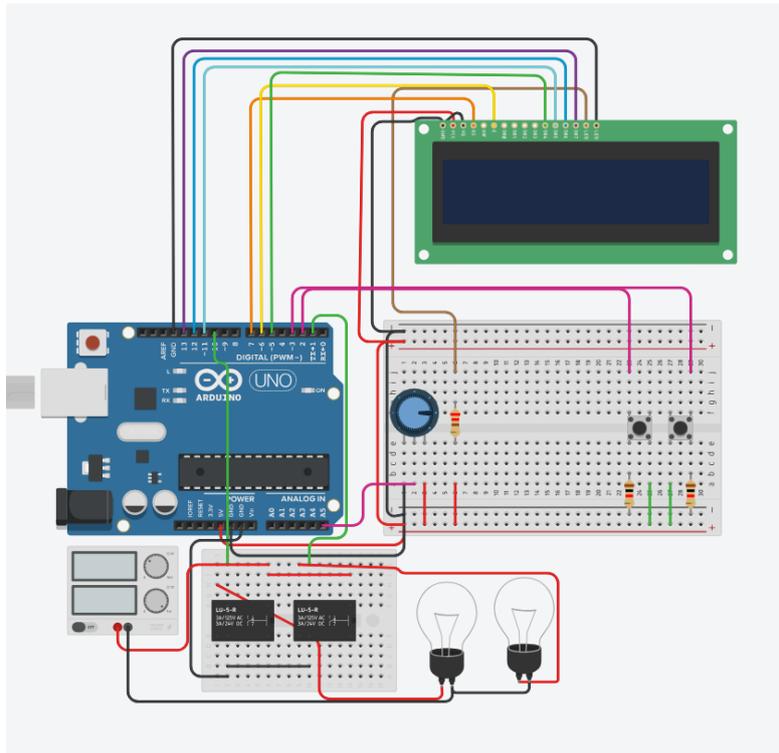


Figura 9: Circuito adaptado no Tinkercad. Arquivo Pessoal

Foi montado o display LCD 16x2, para o seu funcionamento foi necessário ligar os pinos GND, V0 o último pino LED na saída GND da placa Arduino, o pino Led restante foi ligado em um resistor que está ligado a uma fonte 5V, os pinos DB4, DB5, DB6, DB7, RS e E estão ligados a entradas digitais do Arduino.

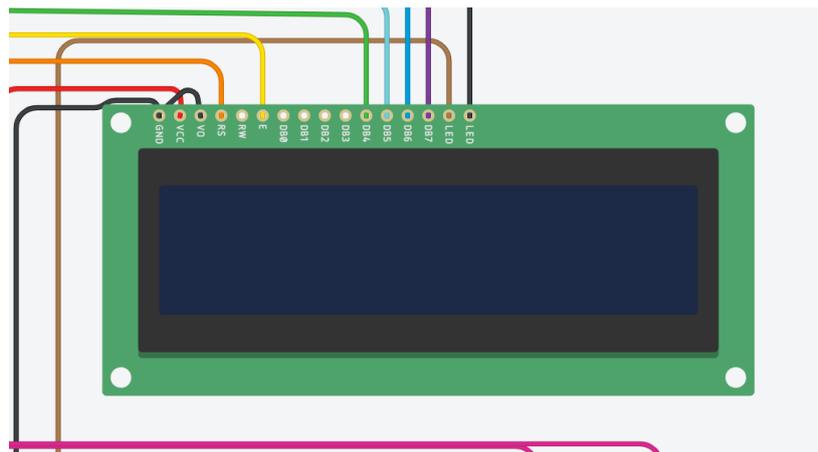


Figura 10: Display LCD recebendo todas entradas para seu funcionamento. Arquivo Pessoal

Foi montado em um protobord os inputs do projeto, um potenciômetro para simular a temperatura lida pelo modulo MAX6675, apesar de ser um input diferente, dentro do código tem uma função que faz o input funcionar de for

similar ao do modulo MAX6675. Também foram adicionados 2 botões que fazem as interrupções que controlam o valor do set point, o primeiro botão aumenta e o segundo diminui o set point, os botões foram montados de forma que quando pressionados manda um sinal 5V para o Arduino.

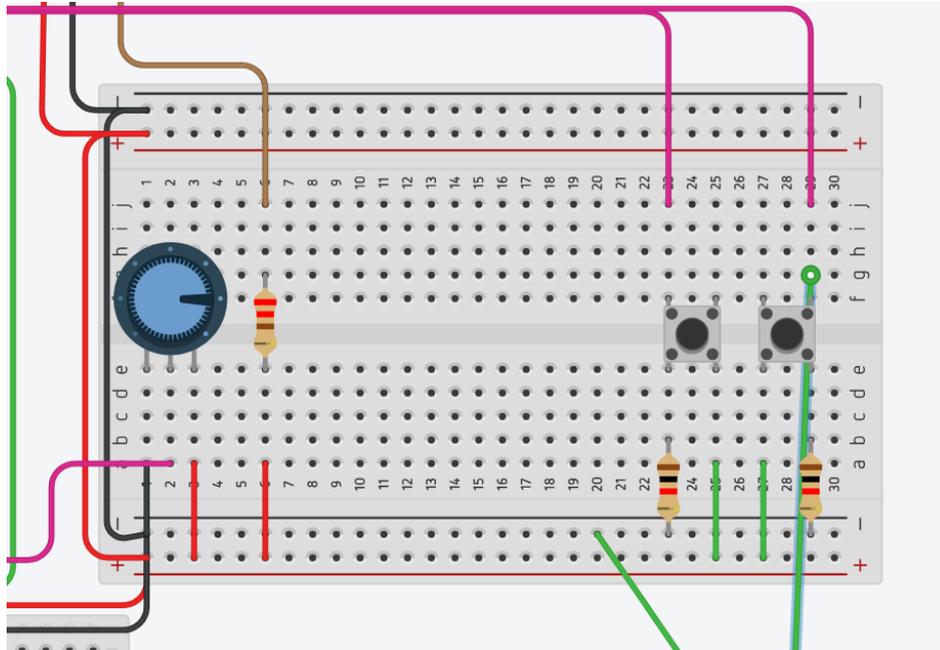


Figura 11: Protobord com os componentes de entrada. Arquivo Pessoal

Para o acionamento dos atuadores, em vez de usar o modulo relé 2 canais 5V, foi necessário montar em um mini protobord dois relés de forma que ele se comportasse igual o modulo, recebendo a alimentação GND da placa Arduino, Recebendo também a tensão de saída dos relés de uma fonte externa e o sinal de acionamento dos relés vindo do Arduino. Como na plataforma Tinkercad não tinha os atuadores necessários para o projeto, foram adicionadas duas lâmpadas para mostrar que o acionamento dos atuadores está funcionando de forma correta.

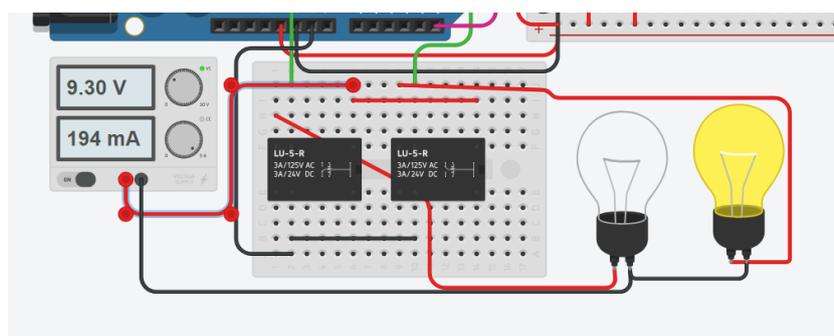


Figura 12: Acionamento dos atuadores. Arquivo Pessoal

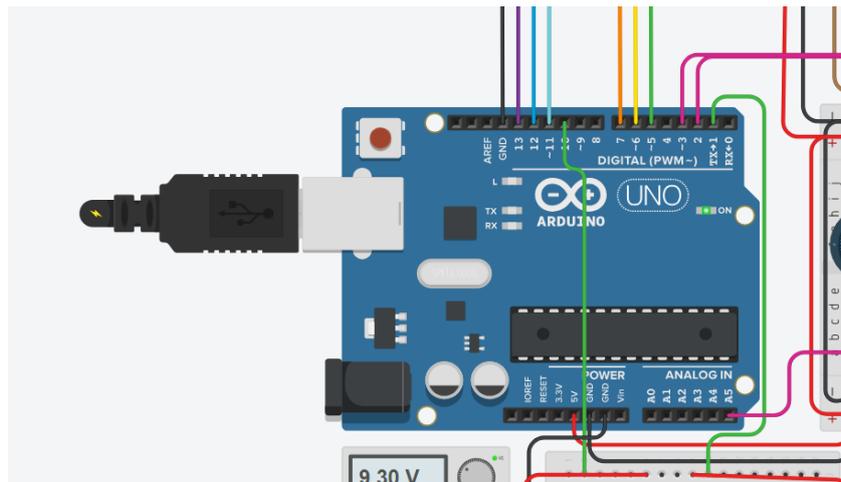


Figura 13: Placa Arduino com todos os componentes conectados. Arquivo Pessoal

Após montar todo o circuito, começou a parte da implementação do código, nessa etapa também foi necessário fazer algumas adaptações, em razão da plataforma Tinkercad não possuir a biblioteca “max6675.h” e não ser possível instalá-la na plataforma, sendo assim as linhas de código relacionadas a biblioteca tiveram que serem comentadas, foi criada uma função que a partir do valor do potenciômetro, simula um temperatura.

Na figura 14 mostra a parte do código responsável por incluir as bibliotecas “LiquidCrystal.h”, que é responsável por fazer o controle do display LCD, ela simplifica a tarefa de enviar informações para LCD. E a biblioteca “max6675.h” ela permite a leitura precisa da temperatura de sensores termopar e converte os dados em valores de temperatura em graus Celsius, essa biblioteca não vem inclusa nas bibliotecas do Arduino, então é necessário fazer a instalação dela.

```
1 #include <LiquidCrystal.h>
2 //#include "max6675.h"
```

Figura 14: inclusão de bibliotecas. Arquivo Pessoal

Para fazer a instalação da biblioteca “max6675.h” foi necessário fazer o *download* dela através do link do GitHub, disponibilizado pelo próprio criador.

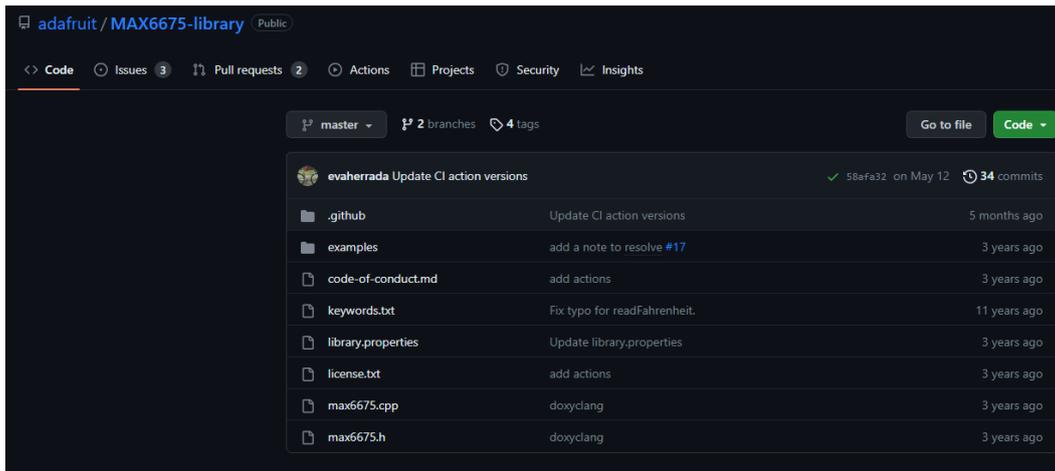


Figura 15: Página de download GitHub (<https://github.com/adafruit/MAX6675-library>)

Após ter feito o download, a inclusão vai ser feita no Arduino IDE, na barra de ferramentas seleccione a opção Sketch, depois include Library e add .ZIP Library.

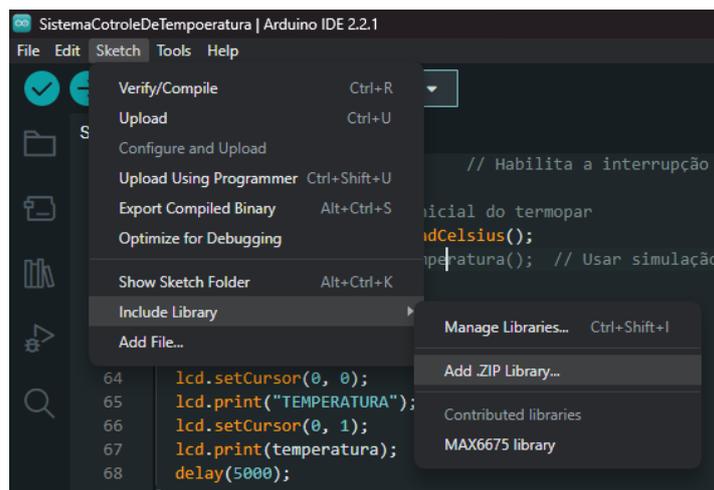


Figura 16: Passo para adicionar biblioteca. Arquivo Pessoal

Ira abrir uma janela para que você escolha a biblioteca, seleccione a biblioteca e clique em abrir e a biblioteca vai ser instalada.

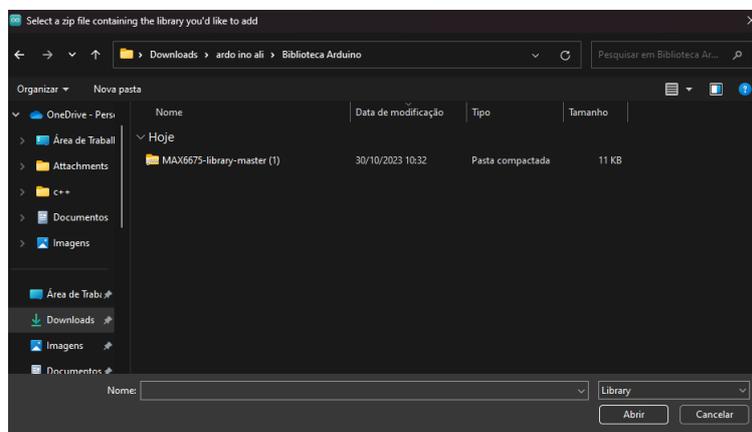


Figura 17: Janela para seleccionar biblioteca. Arquivo Pessoal

O código abaixo define o que está ligado em cada pino da placa Arduino, o pino 2 é o botão de que aumenta o valor do set point, o pino 3 é o botão de que diminui o valor do set point, o pino A5 está ligado o potenciômetro que simula a temperatura, nos pinos 7,6,5,11,12 e 13 estão ligados o LCD, o 10 esta ligado o relé da ventoinha, no 1 esta ligado o relé da bomba d'agua, os pinos 8, 9 e 4 estão ligados no modulo MAX6675.

```
// Configuração dos pinos
const int botaoMais = 2;      // Pino do botão "+" para aumentar o setpoint
const int botaoMenos = 3;    // Pino do botão "-" para diminuir o setpoint
const int ponteciometroTemperatura = A5; // Pino do potenciômetro para simular a temperatura
const int lcdRs = 7;        // Pino RS do display LCD
const int lcdEn = 6;       // Pino EN do display LCD
const int lcdD4 = 5;       // Pino D4 do display LCD
const int lcdD5 = 11;      // Pino D5 do display LCD
const int lcdD6 = 12;      // Pino D6 do display LCD
const int lcdD7 = 13;      // Pino D7 do display LCD
const int releVentoinha = 10; // Pino de controle da ventoinha
const int releBombaAgua = 1; // Pino de controle da bomba de água
const int ktcSO = 8;       // Pino digital do termopar (SO)
const int ktcCS = 9;       // Pino digital do termopar (CS)
const int ktcCLK = 4;      // Pino digital do termopar (CLK)
```

Foram criadas as variáveis de controle, a variável setPoint representa o valor de temperatura desejado, o valor que o sistema vai tentar manter, a variável temperatura representa a temperatura atual do ambiente, a variável temperaturaSimulada representa a temperatura que ira ser simulada pela função de simular temperatura, a variável contadorDeInterrupcao representa quantas vezes ocorreu uma interrupção por overflow do timer 1.

```
// variaveis de controle
int setPoint = 80;          // Valor desejado de temperatura
int temperatura = 0;       // Valor atual da temperatura lida do termopar
int temperaturaSimulada = 0; // Valor simulado da temperatura
int contadorDeInterrupcao = 0; // Contador de interrupção
```

É mostrado no código abaixo a parte que inicializa os objetos do modulo MAX6675 e o display LCD com os seus pinos que foram definidos na parte de configuração de pinos. Ela também mostra a variável interruptFlag, que é um sinalizador se ocorreu interrupção.

```
// Inicializa o objeto Sensor do termopar
//MAX6675 termopar(ktcCLK, ktcCS, ktcSO);
// Inicializa o objeto LiquidCrystal para controle do display LCD
LiquidCrystal lcd(lcdRs, lcdEn, lcdD4, lcdD5, lcdD6, lcdD7);
```

```
volatile bool interruptFlag = false; // Sinalizador de interrupção para ajuste do setpoint
```

O código a seguir mostra a função setup, nela primeiro são configurados as entradas e saídas do código, como entrada ficou configurado os pinos do botaoMais, botaoMenos e pontecimetroTemperatura. Como saída ficou releVentoinha e releBombaAgua. Após isso é feito a inicialização do LCD com o número de linhas e colunas do LCD. Nela também é feito a configuração de interrupção dos botões botaoMais e botaoMenos, configuração da interrupção do timer, prescaler definido em 1024 o timer iniciando no 0, com essa configurações e seguindo a formula tempo = ((valor máximo +1)* prescaler)/frequencia as interrupções aconteceram em aproximadamente 4 segundos e por fim o sistema pega a temperatura atual do ambiente para inicializar a variável temperatura.

```
void setup() {
  // Configura as entradas e saídas do código
  pinMode(botaoMais, INPUT_PULLUP);    // Configura o botão "+" como entrada com pull-up
  pinMode(botaoMenos, INPUT_PULLUP);   // Configura o botão "-" como entrada com pull-up
  pinMode(pontecimetroTemperatura, INPUT); // Configura o potenciômetro como entrada
  pinMode(releVentoinha, OUTPUT);      // Configura o pino do relé da ventoinha como saída
  pinMode(releBombaAgua, OUTPUT);     // Configura o pino do relé da bomba de água como saída

  // Inicialize o LCD com o número de colunas e linhas do seu display
  lcd.begin(16, 2);

  // Configurar a função de tratamento da interrupção para os botões
  attachInterrupt(digitalPinToInterrupt(botaoMenos), interrupcaoB1, RISING);
  attachInterrupt(digitalPinToInterrupt(botaoMais), interrupcaoB2, RISING);

  // Configuração do timer1
  TCCR1A = 0;          // Configura o timer para operação normal, pinos OC1A e OC1B desconectados
  TCCR1B = 0;          // Limpa o registrador
  TCCR1B |= (1 << CS10) | (1 << CS12); // Configura o prescaler para 1024: CS12 = 1 e CS10 = 1

  // Cálculo do tempo com prescaler: ((65536+1)*1024)/16 = 4194368µs
  TCNT1 = 0;          // Inicia o timer para que o estouro ocorra no valor máximo, aproximadamente 4
segundos
  TIMSK1 |= (1 << TOIE1); // Habilita a interrupção do TIMER1

  // Recebe a temperatura inicial do termopar
  //temperatura = termopar.readCelsius();
  temperatura = simulaTemperatura(); // Usar simulação de temperatura em vez do termopar
}
```

A função loop, mostrada no código abaixo tem a função de mostrar a temperatura no LCD.

```

void loop()
{
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("TEMPERATURA");
  lcd.setCursor(0, 1);
  lcd.print(temperatura);
  delay(5000);
}

```

A função mostrada no código abaixo é a função de interrupção por timer, quando ocorre o estouro do timer 1, ela é chamada, ela incrementa o contador de interrupção, faz a leitura da temperatura, faz a verificação se a temperatura está abaixo do setpoint, se estiver ela desliga a bomba de agua e liga a ventoinha, exibe no LCD que a temperatura está baixa e mostra a temperatura, caso a temperatura não esteja a baixo do set point é verificado se a temperatura está acima do setpoint, caso esteja a ventoinha é desligada, a bomba de agua é acionada, exibe no LCD que a temperatura está alta e mostra a temperatura, ela finaliza reiniciando a contagem do timer 0.

```

// Função de interrupção para o timer1
ISR(TIMER1_OVF_vect) {
  contadorDeInterrupcao++;
  //temperatura = termopar.readCelsius();
  temperatura = simulaTemperatura(); // Usa simulação de temperatura em vez do termopar

  // Verifica se a temperatura está abaixo do setpoint
  if (temperatura < setPoint) {
    digitalWrite(releBombaAgua, LOW);
    digitalWrite(releVentoinha, HIGH);

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("TEMPERATURA");
    lcd.setCursor(0, 1);
    lcd.print("BAIXA: ");
    lcd.setCursor(8, 1);
    lcd.print(temperatura);
    delay(5000);
  } else if (temperatura > setPoint) { // Verifica se a temperatura está acima do setpoint
    digitalWrite(releVentoinha, LOW);
    digitalWrite(releBombaAgua, HIGH);

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("TEMPERATURA");
    lcd.setCursor(0, 1);
    lcd.print("ALTA: ");
    lcd.setCursor(7, 1);

```

```

    lcd.print(temperatura);
    delay(5000);
}

TCNT1 = 0; // Reinicia a contagem do timer1
}

```

No código a seguir mostra a função de interrupção pelo botão mais, ela exhibe que está sendo feito o incremento no setpoint, faz o incremento do setpoint e sinaliza que ocorreu a interrupção.

```

// Função de interrupção para o botão "+" (aumenta o setpoint)
void interrupcaoB2() {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("SETPOINT +");
    setPoint++;
    lcd.setCursor(0, 1);
    lcd.print("SETPOINT:");
    lcd.setCursor(10, 1);
    lcd.print(setPoint);
    interruptFlag = true;
    delay(10000); // Aguarda 10 segundos antes de permitir outra interrupção
}

```

A função de interrupção pelo botão menos é mostrada no código abaixo, ela exhibe que está sendo feito o decremento no setpoint, faz o decremento do setpoint e sinaliza que ocorreu a interrupção.

```

// Função de interrupção para o botão "-" (diminui o setpoint)
void interrupcaoB1() {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("SETPOINT -");
    setPoint--;
    lcd.setCursor(0, 1);
    lcd.print("SETPOINT:");
    lcd.setCursor(10, 1);
    lcd.print(setPoint);
    interruptFlag = true;
    delay(10000); // Aguarda 10 segundos antes de permitir outra interrupção
}

```

No código a seguir é mostrado a função simulaTemperatura, ela recebe o valor do potenciometroTemperatura, e divide por 10 assim podendo simular a temperatura de 0 até 102 graus.

```

// Função para simular a leitura da temperatura (usada apenas para testes)
int simulaTemperatura() {
    temperaturaSimulada = analogRead(ponteciometroTemperatura) / 10;
}

```

```
return temperaturaSimulada;  
}
```

Na figura 18 mostra o comportamento do sistema quando é iniciado, em temperatura ambiente, abaixo da temperatura de referência 79 graus Celsius.

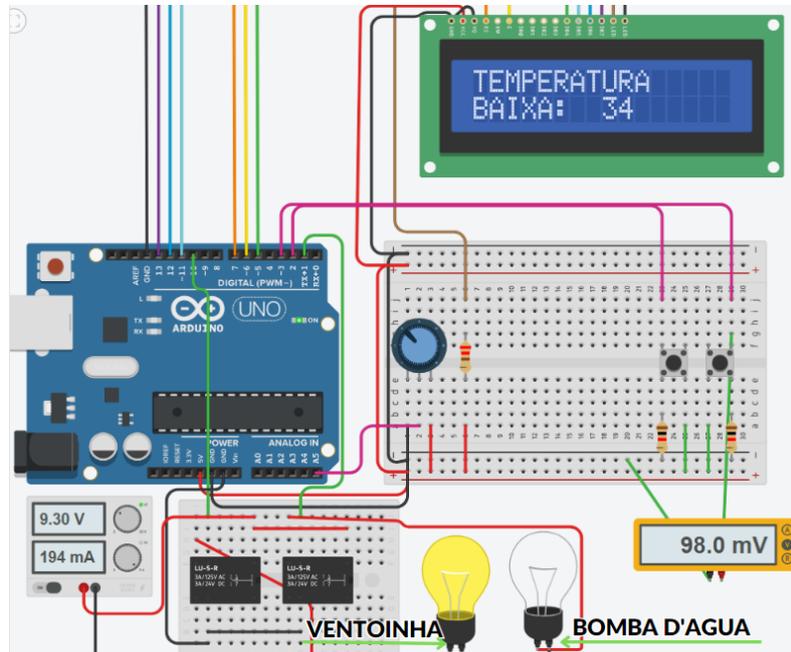


Figura 18: Simulação do sistema em estado baixo. Arquivo Pessoal

Na figura 19 é mostrado o comportamento do sistema quando a temperatura está na temperatura de referência.

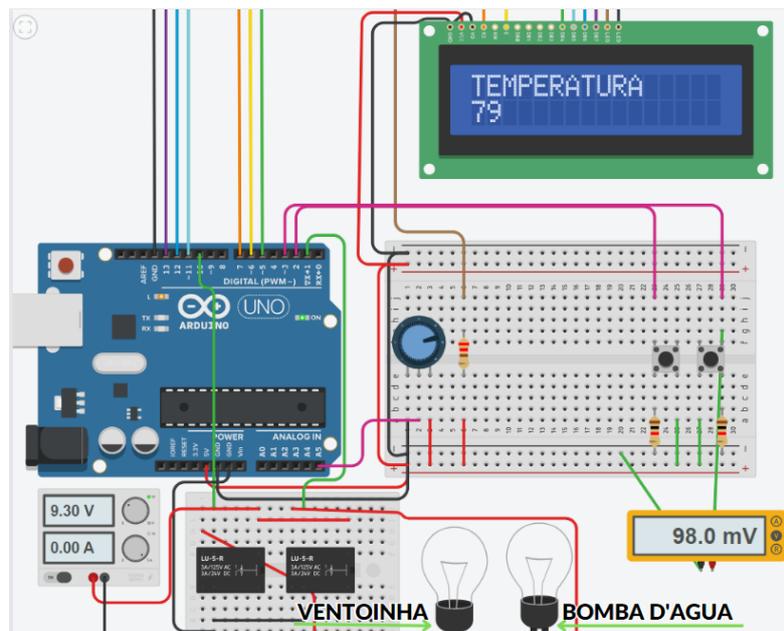


Figura 19: Simulação do sistema em estado estável. Arquivo Pessoal

Na figura 20 mostra o comportamento do sistema quando a temperatura está acima da temperatura de referência.

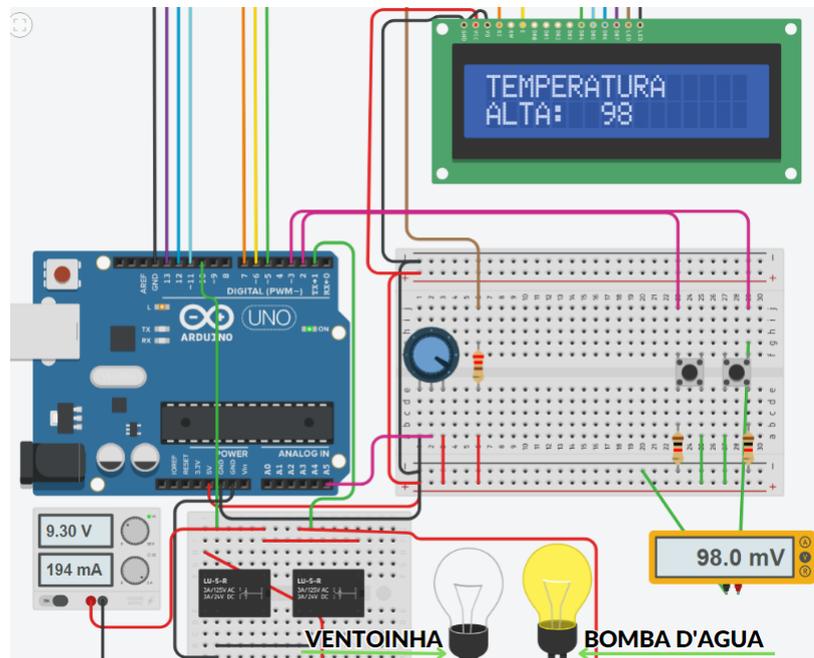


Figura 20: Simulação do sistema em estado alto. Arquivo Pessoal

5. CONSIDERAÇÕES FINAIS

Considerando os objetivos estabelecidos, é possível afirmar que foram alcançados de maneira parcial. Embora tenha sido atingido os objetivos de desenvolver um sistema de controle de temperatura utilizando Arduino, capaz de realizar as funções básicas esperadas e com interrupções internas e externas implementadas, é importante ressaltar que os testes foram conduzidos apenas em ambientes virtuais.

Essa abordagem virtual impediu uma avaliação efetiva da eficiência do sistema em um ambiente real, utilizando os componentes físicos especificados no trabalho. Durante os testes, surgiram desafios devido às limitações de orçamento, o que impossibilitou a implementação física do projeto. Como alternativa, o sistema foi simulado em um ambiente virtual. No entanto, esse ambiente não incluía todos os componentes originais do projeto, exigindo a substituição por outros componentes para simular o comportamento esperado. Essas dificuldades ressaltam a necessidade de conduzir futuras avaliações em ambientes físicos para garantir a validade e eficiência do sistema proposto.

5.1 Trabalhos futuros

Diante da complexidade da tarefa que é o desenvolvimento do sistema descrito neste trabalho, há espaço para aprimoramentos que podem tornar o funcionamento desse sistema de controle de temperatura ainda mais eficaz. É necessário explorar algumas melhorias técnicas que têm o potencial de elevar a qualidade da implementação do sistema.

5.2 Melhorias técnicas

Implementação de um mapeamento de tempo, com um mapeamento de tempo abriria espaço para a criação de funções que melhorariam o controle da temperatura.

Implementação da funcionalidade de poder girar a carne em tempos em um tempo escolhido.

A implementação de rotinas para carnes mais comuns, onde o usuário precisasse apenas selecionar o tipo de carne e o sistema já identificaria a temperatura ideal, tempo em que ela deve ser virada e o tempo de preparo.

REFERÊNCIAS

8 tipos de churrasqueira: qual o melhor modelo?. [S. l.], 3 mar. 2023. Disponível em: <https://meulugar.quintoandar.com.br/tipos-de-churrasqueiras/>. Acesso em: 19 out. 2023.

A IMPORTÂNCIA da Temperatura da Carne Grelhada na Segurança Alimentar. [S. l.], 23 ago. 2019. Disponível em: <https://hannainst.com.br/a-importancia-da-temperatura-da-carne-grelhada-na-seguranca-alimentar/>. Acesso em: 30 nov. 2023.

ARAÚJO, Guilherme Matheus de. **Controle de Temperatura de Estufas Utilizando Método de Controle Supervisório**. 2020. Trabalho de conclusão de curso (Bacharelado em Sistemas de Informação) - Universidade Federal Rural de Pernambuco, [S. l.], 2020. Disponível em: https://repository.ufrpe.br/bitstream/123456789/4166/1/tcc_guilhermematheusdearaujo.pdf. Acesso em: 1 nov. 2023.

BAEZA, Jorge Pomares. **Manual de arduino**. Vol. 1. ed. Espanha: GITE – IEA, 2009. Disponível em: <https://rua.ua.es/dspace/bitstream/10045/11833/1/arduino.pdf>. Acesso em: 19 out. 2023.

BERTULUCCI , Cristiano. Conheça mais sobre o controle da temperatura. In: Conheça mais sobre o controle da temperatura. [S. l.], 19 dez. 2016. Disponível em: <https://qualidadeonline.wordpress.com/2016/12/19/conheca-mais-sobre-o-controle-da-temperatura/>. Acesso em: 12 out. 2023.

BORGES, Felix. **DESENVOLVIMENTO E IMPLEMENTAÇÃO DE UM SISTEMA DE MEDIÇÃO PARA AJUSTE DE SENSORES DE TEMPERATURA**. Orientador: Luan de Campos Corrêa. 2019. Artigos (TCC) (Bacharelado Engenharia Mecânica) - Satc - Educação, Tecnologia e Inovação, [S. l.], 2019. Disponível em: <https://rua.ua.es/dspace/bitstream/10045/11833/1/arduino.pdf>. Acesso em: 19 out. 2023.

BORGES FILHO, Mario Divino. **Sistema de monitoramento e controle remoto para nanocervejarias**. 2020. Trabalho de conclusão de curso (Graduação em Engenharia de Controle e Automação) - Universidade Federal de Uberlândia, [S. l.], 2020. Disponível em: <https://repositorio.ufu.br/bitstream/123456789/31349/4/SistemaDeMonitoramento.pdf>. Acesso em: 1 nov. 2023.

CRAVO, Edilson. **Sistema de controle de temperatura: o que é e sua importância**. [S. l.], 1 ago. 2022. Disponível em: <https://blog.kalatec.com.br/sistema-controle-temperatura/>. Acesso em: 1 nov. 2023.

DOS SANTOS, Carlos Vinícius. A Química da Combustão. In: A Química da Combustão. [S. l.], 12 out. 2023. Disponível em: <https://www.omundodaquimica.com.br/curiosidade/churrasco>. Acesso em: 12 out. 2023.

KRINSKI, Jackson. **MODELAGEM E CONTROLE DE TEMPERATURA DO PROCESSO DE BRASSAGEM PARA A PRODUÇÃO DE CERVEJA**. 2018. Trabalho de conclusão de curso (Bacharelado em Engenharia Elétrica) - Universidade Federal do Paraná, [S. l.], 2018. Disponível em: <https://www.eletrica.ufpr.br/tcc/2018/2s/Jackson%20Krinski/Modelagem%20e%20controle%20de%20temperatura%20do%20processo%20de%20Brassagem%20para%20a%20produ%C3%A7%C3%A3o%20de%20cerveja%20Jackson%20Krinski%20GRR%2020118289.pdf>. Acesso em: 1 nov. 2023.

MAKIYAMA, Marcio. **O que é arduino, para que serve, benefícios e projetos [Exemplos]**. [S. l.], 29 nov. 2022. Disponível em: <https://victorvision.com.br/blog/o-que-e-arduino/>. Acesso em: 6 nov. 2023.

NOGUEIRA, Marcos. **Churrasco: o ritual pré-histórico que ainda nos reúne em torno da fogueira**. [S. l.], 1 abr. 2020. Disponível em: <https://super.abril.com.br/especiais/churrasco-nao-e-tecnologia-e-feiticaria/>. Acesso em: 1 nov. 2023.

OLIVEIRA JUNIOR, Aguinaldo José Rodrigues de; OLIVEIRA, Eduardo Henrique Hurla de; MARQUES, Lucas Damazio. **Sistema de monitoramento e controle de temperatura da água de um aquário**. 2019. Trabalho de conclusão de curso (Tecnólogo em Automação Industrial) - Universidade Tecnológica Federal do Paraná, [S. l.], 2019. Disponível em: https://repositorio.utfpr.edu.br/jspui/bitstream/1/16956/1/PG_DAELE_2019_2_03.pdf. Acesso em: 1 nov. 2023.

MAKIYAMA, Marcio. **O que é arduino, para que serve, benefícios e projetos [Exemplos]**. [S. l.], 29 nov. 2022. Disponível em: <https://victorvision.com.br/blog/o-que-e-arduino/>. Acesso em: 18 dez. 2023.

APÊNDICES

APÊNDICE A – CONFIGURAÇÃO DOS PINOS

```
// Configuração dos pinos
const int botaoMais = 2;           // Pino do botão "+" para aumentar o setpoint
const int botaoMenos = 3;         // Pino do botão "-" para diminuir o setpoint
const int ponteciometroTemperatura = A5; // Pino do potenciômetro para simular
a temperatura

const int lcdRs = 7;              // Pino RS do display LCD
const int lcdEn = 6;              // Pino EN do display LCD
const int lcdD4 = 5;              // Pino D4 do display LCD
const int lcdD5 = 11;             // Pino D5 do display LCD
const int lcdD6 = 12;             // Pino D6 do display LCD
const int lcdD7 = 13;             // Pino D7 do display LCD

const int releVentoinha = 10;     // Pino de controle da ventoinha
const int releBombaAgua = 1;     // Pino de controle da bomba de água
const int ktcSO = 8;              // Pino digital do termopar (SO)
const int ktcCS = 9;              // Pino digital do termopar (CS)
const int ktcCLK = 4;             // Pino digital do termopar (CLK)
```

APÊNDICE B – VARIÁVEIS DE CONTROLE

```
// variáveis de controle
int setPoint = 80;                // Valor desejado de temperatura
int temperatura = 0;              // Valor atual da temperatura lida do termopar
int temperaturaSimulada = 0;     // Valor simulado da temperatura
int contadorDeInterrupcao = 0;   // Contador de interrupção
```

APÊNDICE C – INICIANDO OBJETOS

```
// Inicializa o objeto Sensor do termopar
//MAX6675 termopar(ktcCLK, ktcCS, ktcSO);

// Inicializa o objeto LiquidCrystal para controle do display LCD
LiquidCrystal lcd(lcdRs, lcdEn, lcdD4, lcdD5, lcdD6, lcdD7);

volatile bool interruptFlag = false; // Sinalizador de interrupção para ajuste do
setpoint
```

APÊNDICE D – FUNÇÃO SETUP

```
void setup() {
  // Configura as entradas e saídas do código
  pinMode(botaoMais, INPUT_PULLUP);      // Configura o botão "+" como
  entrada com pull-up
  pinMode(botaoMenos, INPUT_PULLUP);     // Configura o botão "-" como
  entrada com pull-up
  pinMode(ponteciometroTemperatura, INPUT); // Configura o potenciômetro
  como entrada
  pinMode(releVentoinha, OUTPUT);        // Configura o pino do relé da ventoinha
  como saída
  pinMode(releBombaAgua, OUTPUT);        // Configura o pino do relé da bomba
  de água como saída

  // Inicialize o LCD com o número de colunas e linhas do seu display
  lcd.begin(16, 2);

  // Configurar a função de tratamento da interrupção para os botões
  attachInterrupt(digitalPinToInterrupt(botaoMenos), interrupcaoB1, RISING);
  attachInterrupt(digitalPinToInterrupt(botaoMais), interrupcaoB2, RISING);
}
```

```

// Configuração do timer1
TCCR1A = 0;           // Configura o timer para operação normal, pinos
OC1A e OC1B desconectados
TCCR1B = 0;           // Limpa o registrador
TCCR1B |= (1 << CS10) | (1 << CS12); // Configura o prescaler para 1024:
CS12 = 1 e CS10 = 1

// Cálculo do tempo com prescaler: ((65536+1)*1024)/16 = 4194368µs
TCNT1 = 0;           // Inicia o timer para que o estouro ocorra no valor
máximo, aproximadamente 4 segundos
TIMSK1 |= (1 << TOIE1); // Habilita a interrupção do TIMER1

// Recebe a temperatura inicial do termopar
//temperatura = termopar.readCelsius();
temperatura = simulaTemperatura(); // Usar simulação de temperatura em vez
do termopar
}

```

APÊNDICE E – FUNÇÃO LOOP

```

void loop()
{
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("TEMPERATURA");
  lcd.setCursor(0, 1);
  lcd.print(temperatura);
  delay(5000);
}

```

APÊNDICE F – FUNÇÃO INTERRUPTÃO POR TIMER

```
// Função de interrupção para o timer1
ISR(TIMER1_OVF_vect) {
    contadorDeInterrupcao++;
    //temperatura = termopar.readCelsius();
    temperatura = simulaTemperatura(); // Usa simulação de temperatura em vez
do termopar

    // Verifica se a temperatura está abaixo do setpoint
    if (temperatura < setPoint) {
        digitalWrite(releBombaAgua, LOW);
        digitalWrite(releVentoinha, HIGH);

        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("TEMPERATURA");
        lcd.setCursor(0, 1);
        lcd.print("BAIXA: ");
        lcd.setCursor(8, 1);
        lcd.print(temperatura);
        delay(5000);
    } else if (temperatura > setPoint) { // Verifica se a temperatura está acima do
setpoint
        digitalWrite(releVentoinha, LOW);
        digitalWrite(releBombaAgua, HIGH);

        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("TEMPERATURA");
        lcd.setCursor(0, 1);
        lcd.print("ALTA: ");
        lcd.setCursor(7, 1);
```

```

    lcd.print(temperatura);
    delay(5000);
}

TCNT1 = 0; // Reinicia a contagem do timer1
}

```

APÊNDICE G – FUNÇÃO INTERRUPTÃO PELO BOTÃO MAIS

```

// Função de interrupção para o botão "+" (aumenta o setpoint)
void interrupcaoB2() {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("SETPOINT +");
    setPoint++;
    lcd.setCursor(0, 1);
    lcd.print("SETPOINT:");
    lcd.setCursor(10, 1);
    lcd.print(setPoint);
    interruptFlag = true;
    delay(10000); // Aguarda 10 segundos antes de permitir outra interrupção
}

```

APÊNDICE H – FUNÇÃO INTERRUPTÃO PELO BOTÃO MENOS

```

// Função de interrupção para o botão "-" (diminui o setpoint)
void interrupcaoB1() {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("SETPOINT -");
    setPoint--;
    lcd.setCursor(0, 1);
    lcd.print("SETPOINT:");
    lcd.setCursor(10, 1);
}

```

```
lcd.print(setPoint);  
interruptFlag = true;  
delay(10000); // Aguarda 10 segundos antes de permitir outra interrupção  
}
```

APÊNDICE I – FUNÇÃO SIMULA TEMPERATURA

```
// Função para simular a leitura da temperatura (usada apenas para testes)  
int simulaTemperatura() {  
    temperaturaSimulada = analogRead(ponteciometroTemperatura) / 10;  
    return temperaturaSimulada;  
}
```