

Sistema de Resfriamento com Cooler por Arduino

**CURITIBA
2023**

**Centro Universitário UniDomBosco
Núcleo de Ciências Exatas e Tecnológicas
Tecnólogo em Análise e Desenvolvimento de Sistemas**

Sistema de Resfriamento com Cooler por Arduino

Projeto do curso de Análise e Desenvolvimento
de Sistemas desenvolvido pelos alunos
Lucas Iaremchuk Gomes, Guilherme Penso,
Emanoel andre Hacke de Britto, Murilo Lustosa,
Kauan Vinicius de Moraes, Maurício de Oliveira
sob orientação do Prof. Edson Pedro Ferlin

**CURITIBA - PR
2023**

SUMÁRIO

1.INTRODUÇÃO	4
2 DESCRIÇÃO DO PROBLEMA	5
2.1 Contextualização	5
2.2 Problema	5
2.3 Objetivos	5
2.4 Justificativa	6
3.HARDWARE, SOFTWARE E CIRCUITO	7
3.1 HARDWARE	7
3.2 SOFTWARE	14
3.2 CIRCUITO	18
4. RESULTADOS	19
4.1 Testes	19
4.2 Análise de Dados	22
5. DIÁRIO DE BORDO	23
6. CONCLUSÃO	25

1. INTRODUÇÃO:

O presente relatório tem como objetivo apresentar um projeto que simula um sistema de resfriamento de um processador, utilizando a plataforma Arduino. A proposta consiste na utilização de uma pastilha Peltier para simular a geração de calor, juntamente com a aplicação de coolers para o resfriamento adequado do sistema.

A pastilha Peltier, um dispositivo semicondutor, desempenha um papel crucial nessa simulação, sendo conectada e constantemente monitorada por um sensor de temperatura. O sensor de temperatura ativa os coolers controlados pelo Arduino, proporcionando a refrigeração necessária para a pastilha e evitando o sobreaquecimento do sistema como um todo. Ademais, o projeto inclui uma caixa de proteção, que além de garantir a segurança dos componentes, também contribui para o eficiente resfriamento dos mesmos.

No decorrer deste relatório, serão detalhadas as especificações técnicas empregadas no projeto, incluindo informações sobre a capacidade de resfriamento da pastilha Peltier, a potência e eficiência dos coolers utilizados, bem como a capacidade de monitoramento e controle oferecida pelo Arduino. Além disso, serão abordados o circuito eletrônico desenvolvido para a interconexão dos componentes, o código de programação implementado para o controle do sistema e os resultados obtidos por meio da simulação.

Por fim, serão apresentadas as conclusões derivadas do desenvolvimento do projeto, incluindo considerações sobre a eficácia do sistema de resfriamento proposto, sua aplicabilidade em cenários reais e possíveis melhorias e aprimoramentos que poderiam ser implementados para otimizar ainda mais o desempenho do sistema.

2. DESCRIÇÃO DO PROBLEMA:

2.1 Contextualização

Com o constante avanço tecnológico na área de processadores, surge a necessidade de desenvolver soluções eficientes para o resfriamento desses componentes. O aumento da potência e do desempenho dos processadores resulta em uma geração de calor significativa durante as operações, o que pode levar ao superaquecimento e comprometer o funcionamento adequado do sistema.

Nesse contexto, é fundamental realizar estudos e experimentos para avaliar diferentes métodos de resfriamento e sua capacidade de manter a temperatura do processador dentro de limites seguros. Este trabalho tem como objetivo simular a geração de calor de um processador e analisar a eficácia dos coolers controlados pelo Arduino no resfriamento desse sistema simulado.

2.2 O Problema

O problema central que motiva este trabalho consiste em investigar a capacidade dos coolers controlados pelo Arduino em resfriar um sistema simulado de geração de calor. Para simular a geração de calor, será utilizado um elemento térmico conectado ao Arduino. O desafio reside em determinar se os coolers controlados pelo Arduino são capazes de resfriar adequadamente o sistema simulado, evitando o superaquecimento e mantendo a temperatura controlada.

2.3 Objetivos

O objetivo principal deste trabalho é avaliar a eficácia dos coolers controlados pelo Arduino no resfriamento de um sistema simulado de geração de calor. Para alcançar esse objetivo, os seguintes objetivos específicos foram estabelecidos:

2.3.1 Pesquisar e analisar as características técnicas dos coolers disponíveis, levando em consideração fatores como capacidade de fluxo de ar, potência de resfriamento e compatibilidade com o sistema proposto.

2.3.2 Projetar e implementar o circuito eletrônico necessário para interconectar os componentes do sistema, garantindo o correto funcionamento e controle dos coolers e do elemento térmico.

2.3.3 Desenvolver um código de programação para o Arduino, permitindo o monitoramento da temperatura por meio de um sensor e o acionamento dos coolers de forma adequada, mantendo a temperatura do sistema simulado controlada.

2.3.4 Realizar simulações e experimentos para avaliar a capacidade dos coolers controlados pelo Arduino em resfriar o sistema simulado, analisando a redução de temperatura e a estabilidade do sistema.

2.4 Justificativa

A relevância deste projeto reside na importância de encontrar soluções eficientes de resfriamento para processadores, a fim de evitar o superaquecimento e garantir o desempenho adequado desses componentes. A simulação da geração de calor do processador permite testar a eficácia dos coolers controlados pelo Arduino e contribui para o desenvolvimento de técnicas de resfriamento mais eficientes e econômicas.

3. HARDWARE, SOFTWARE E CIRCUITO:

3.1 HARDWARE:

ARDUINO UNO

O Arduino Uno é baseado no microcontrolador ATmega328P, que integra uma unidade central de processamento (CPU), memória e periféricos em um único chip. Embora seja uma placa microcontroladora, ela compartilha muitos princípios básicos da arquitetura de computadores.

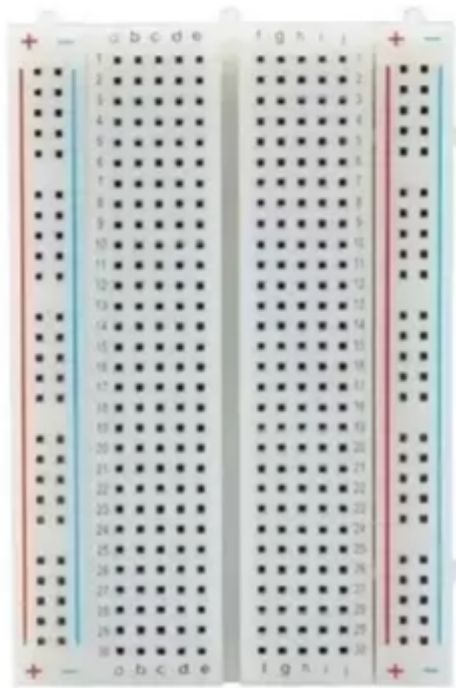


Especificações:

- Microcontrolador: ATmega328
- Tensão de Operação: 5v
- Tensão de entrada (recomendada): 7-12v
- Pinos de entrada/saída: 14 (dos quais 6 podem ser PWM);
- Pinos de entrada analógica: 6
- Corrente DC por pino I/O: 40mA
- Corrente DC para pino 3,3v: 50mA
- Memória Flash: 32KB (dos quais 0,5KB são usados pelo Bootloader)
- SRAM: 2KB - EEPROM: 1KB
- Velocidade do Clock: 16MHz
- Dimensões 5,3 cm x 7,3 cm x 1,2 cm

PROTOBOARD

A protoboard é uma placa de plástico com uma matriz de furos condutores interligados. Cada furo é conectado eletricamente a outros furos de mesma linha vertical ou horizontal. Essa estrutura de contatos permite que componentes eletrônicos, como resistores, capacitores, transistores e cabos jumper, sejam inseridos e conectados de maneira temporária e livre



Especificações:

- Quantidade de pontos: 400
- Possui 2 barramentos para alimentação
- 30 colunas e 10 linhas
- Aceita Fios de 29AWG até 20AWG
- Material base: ABS
- Material conexão: Bronze banhado com níquel
- Diâmetro do furo: 0,8mm²
- Dimensões Aprox.: 83.5 x 54.5 x 8.5mm

PASTILHA PELTIER

A pastilha Peltier, também conhecida como placa Peltier, é um dispositivo termoelétrico que aproveita o efeito Peltier para transferir calor de um lado para o outro. Ela é composta por materiais semicondutores, geralmente bismuto-telureto (Bi_2Te_3), intercalados com materiais condutores. A pastilha Peltier consiste em dois lados principais: o lado quente (fonte de calor) e o lado frio (fonte de resfriamento). Quando uma corrente elétrica é aplicada à pastilha, ocorre o efeito Peltier, onde calor é transferido do lado frio para o lado quente ou vice-versa, dependendo da direção da corrente elétrica.



Especificações:

- Modelo: TEC1-12706;
- Faixa de temperatura: -30° a 70° Celsius;
- Tensão de operação: 0-15,2VDC;
- Corrente de operação: 0-6A;
- Potência máxima: 60W;
- Comprimento do Fio: 30cm;
- Dimensões: 40 x 40mm.

COOLER

Dispositivo de resfriamento projetado para remover o calor excessivo gerado pelos componentes eletrônicos de um computador.

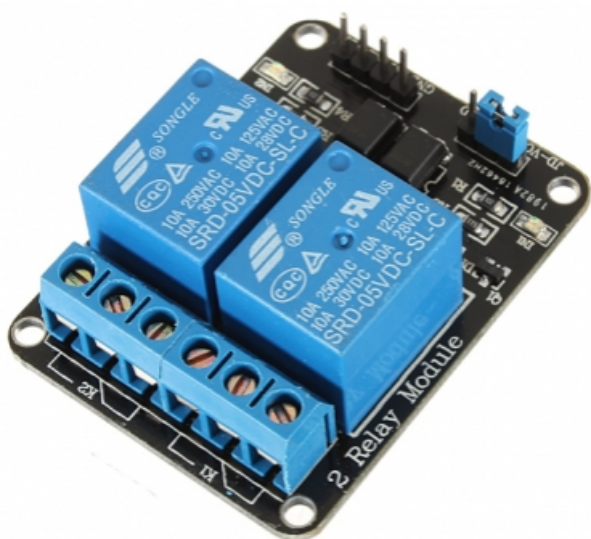


Especificações:

- Dimensões: 6x6x2 cm
- Tensão: 12V
- Velocidade: 4300 rpm
- Fluxo de ar: 18,2 cfm
- Ruído: 26dB

MÓDULO RELÉ

O módulo relé consiste em uma placa de circuito impresso que possui um relé e outros componentes necessários para seu funcionamento adequado. O relé é o componente principal do módulo e é responsável pelo controle das cargas elétricas. O relé é composto por uma bobina e contatos elétricos. Quando uma corrente elétrica é aplicada à bobina, ela gera um campo magnético que atrai os contatos, conectando ou desconectando os terminais dos contatos elétricos. Essa ação do relé é usada para controlar a passagem de corrente para a carga elétrica.

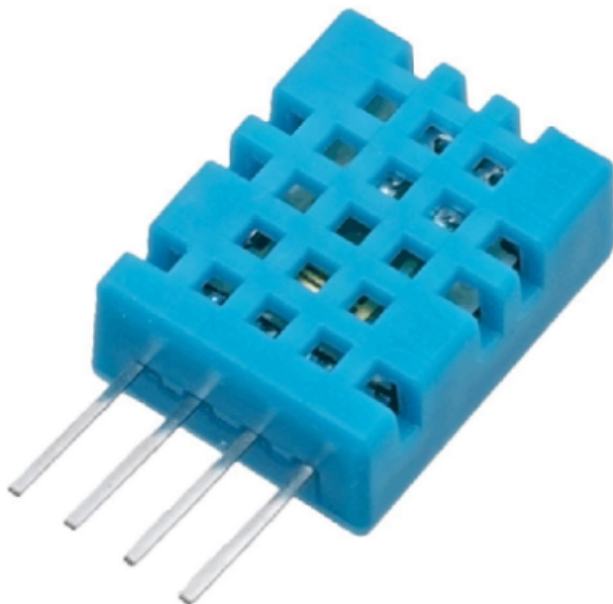


Especificações:

- Modelo: SRD-05VDC-SL-C
- Tensão de operação: 5VDC
- Permite controlar cargas de 220V AC
- Corrente típica de operação: 15~20mA
- LED indicador de status
- Pinagem: Normal Aberto, Normal Fechado e Comum
- Tensão de saída: (30 VDC a 10A) ou (250VAC a 10A)
- Furos de 3mm para fixação nas extremidades da placa
- Tempo de resposta: 5~10ms
- Dimensões: 51 x 38 x 20mm
- Peso: 30g

SENSOR DHT11

O Sensor de Umidade e Temperatura – DHT11 é um dos componentes mais utilizados em projetos que envolva medição de temperatura e umidade ambiente. Este sensor faz medições de temperatura de 0° até 50° celsius e mede a umidade do ar nas faixas de 20% a 90%. A precisão (margem de erro) do sensor para medição de temperatura é de aproximadamente 2° celsius e para umidade é de 5%.



Especificações:

- Tensão de Alimentação: 3 a 5V;
- Faixa de Medição de Umidade: 20 a 90% UR;
- Faixa de Medição de Temperatura: 0 a 50°C;
- Precisão de Medição da Umidade: +- 5% UR;
- Precisão de Medição de Temperatura: +- 2°C.

DISPLAY LCD 16X2 e MÓDULO I2C

O display LCD 16x2 é um componente eletrônico com capacidade para exibir 16 caracteres em duas linhas. É amplamente utilizado em projetos eletrônicos, como microcontroladores e placas Arduino. Cada caractere é formado por uma matriz de 5x8 pixels. É fácil de usar, permite exibir texto claro e legível, e pode ser retroiluminado para facilitar a visualização em ambientes com pouca luz.



Especificações:

- Cor backlight Azul
- Cor escrita Branca
- Possui adaptador display I2C integrado
- Possui potenciômetro para ajuste do contraste
- Tensão de operação 5v
- 2 Linhas
- 16 Colunas
- Dimensões 80 x 36 x 12mm
- Área visível 64,5 x 16mm

3.2 SOFTWARE:

Código:

```
#include <LiquidCrystal_I2C.h>
#include <DHT.h>

#define DHTPIN 4
#define DHTTYPE DHT11
#define RELAY 5
#define COOLER1 2
#define COOLER2 3

DHT dht(DHTPIN, DHTTYPE); // Inicializa o sensor DHT com o pino especificado e
o tipo de sensor

LiquidCrystal_I2C lcd(0x27, 16, 2); // Inicializa o LCD I2C com o endereço
especificado e o tamanho do display

unsigned long previousMillis = 0;
const long interval = 10000;
bool cooler1On = false;
bool cooler2On = false;
bool peltierOn = true;
bool peltierOffDelay = false;
unsigned long peltierOffStartTime = 0;
const unsigned long peltierOffDelayTime = 10000;

void setup() {
  pinMode(RELAY, OUTPUT);
```

```

pinMode(COOLER1, OUTPUT);
pinMode(COOLER2, OUTPUT);

digitalWrite(RELAY, LOW);
peltierOn = true; // Define a peltier como ligada ao iniciar o programa

dht.begin(); // Inicializa o sensor DHT
lcd.begin(16, 2); // Inicializa o display LCD com o número de colunas e linhas
especificados
lcd.backlight(); // Liga o backlight do LCD
}

void loop() {
    float h = dht.readHumidity(); // Lê a umidade do sensor DHT
    float t = dht.readTemperature(); // Lê a temperatura do sensor DHT

    lcd.setCursor(0, 0); // Define a posição do cursor no LCD
    lcd.print("Temp: ");
    lcd.print(t); // Exibe a temperatura no LCD

    lcd.setCursor(0, 1); // Define a posição do cursor no LCD
    lcd.print("Hum: ");
    lcd.print(h); // Exibe a umidade no LCD

    if (t > 30) { // Verifica se a temperatura está acima de 30 graus Celsius
        digitalWrite(COOLER1, HIGH); // Liga o cooler 1
        cooler1On = true;
    } else {
        digitalWrite(COOLER1, LOW); // Desliga o cooler 1
        cooler1On = false;
    }
}

```

```

if (t > 35) { // Verifica se a temperatura está acima de 35 graus Celsius
    digitalWrite(COOLER2, HIGH); // Liga o cooler 2
    cooler2On = true;
} else {
    digitalWrite(COOLER2, LOW); // Desliga o cooler 2
    cooler2On = false;
}

```

```

unsigned long currentMillis = millis(); // Obtém o tempo atual em milissegundos

```

```

if (peltierOn) { // Verifica se a peltier está ligada
    if (t >= 45) { // Verifica se a temperatura está acima ou igual a 45 graus Celsius
        if (!peltierOffDelay) { // Se o atraso para desligar a peltier ainda não estiver em andamento
            peltierOffDelay = true;
            peltierOffStartTime = currentMillis; // Armazena o tempo atual como o início do atraso para desligar a peltier
        } else if (currentMillis - peltierOffStartTime >= peltierOffDelayTime) {
            digitalWrite(RELAY, HIGH); // Desliga a peltier
            peltierOn = false; // Define a peltier como desligada
            peltierOffDelay = false; // Reinicia o atraso para desligar a peltier
            previousMillis = currentMillis;
        }
    } else {
        peltierOffDelay = false; // Reinicia o atraso para desligar a peltier
        previousMillis = currentMillis;
    }
} else {
    if (t < 30) { // Verifica se a temperatura está abaixo de 30 graus Celsius
        digitalWrite(RELAY, LOW); // Liga a peltier
    }
}

```



```
peltierOn = true; // Define a peltier como ligada
}
}
}
```

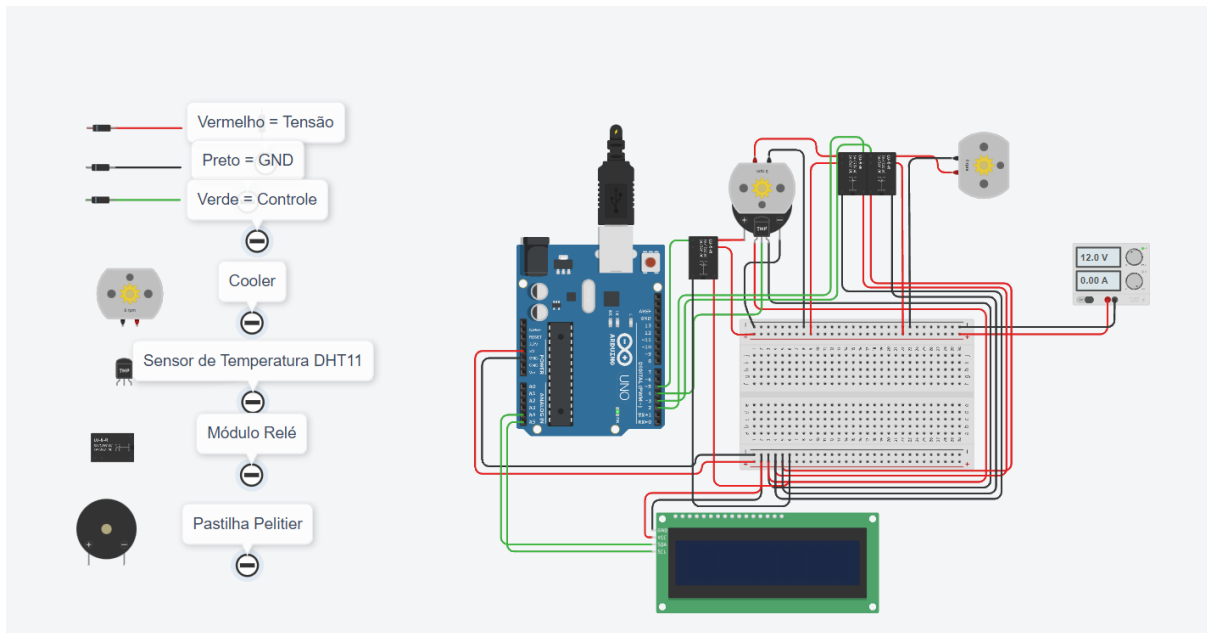
Na função "setup()", são realizadas as configurações iniciais do programa. Os pinos do relé, dos coolers e do display LCD são configurados como saída. Além disso, o relé é definido como desligado e são feitas as inicializações do sensor DHT11 e do display LCD.

A função "loop()" é o coração do programa, sendo executada continuamente. Em cada iteração, ela realiza as seguintes tarefas:

- Lê a temperatura e umidade ambiente utilizando o sensor DHT11.
- Escreve a temperatura e umidade no display LCD para visualização.
- Controla o funcionamento do primeiro cooler com base na temperatura lida. Se a temperatura estiver acima de 30°C, o cooler é ligado; caso contrário, é desligado.
- Controla o funcionamento do segundo cooler de forma similar ao primeiro, mas com um ponto de ativação de 35°C.
- Verifica se a peltier (componente de refrigeração) está ligada. Se estiver, é feita uma verificação adicional para verificar se a temperatura atingiu o limite de 45°C.
- Caso a temperatura tenha atingido o limite e não haja um atraso de desligamento programado, o relé é acionado para desligar a peltier. Em seguida, as variáveis relacionadas são atualizadas.
- Caso a temperatura ainda esteja abaixo do limite, as variáveis são atualizadas normalmente.
- O código utiliza variáveis globais para rastrear o estado dos coolers, da peltier e do tempo de execução. Essas variáveis são usadas para controlar o funcionamento do sistema de resfriamento com base nas condições definidas.

Em resumo, o programa monitora a temperatura ambiente e ativa os coolers conforme a necessidade para manter o ambiente resfriado. Além disso, existe uma proteção que desliga a peltier se a temperatura atingir um limite específico.

3.3 CIRCUITO

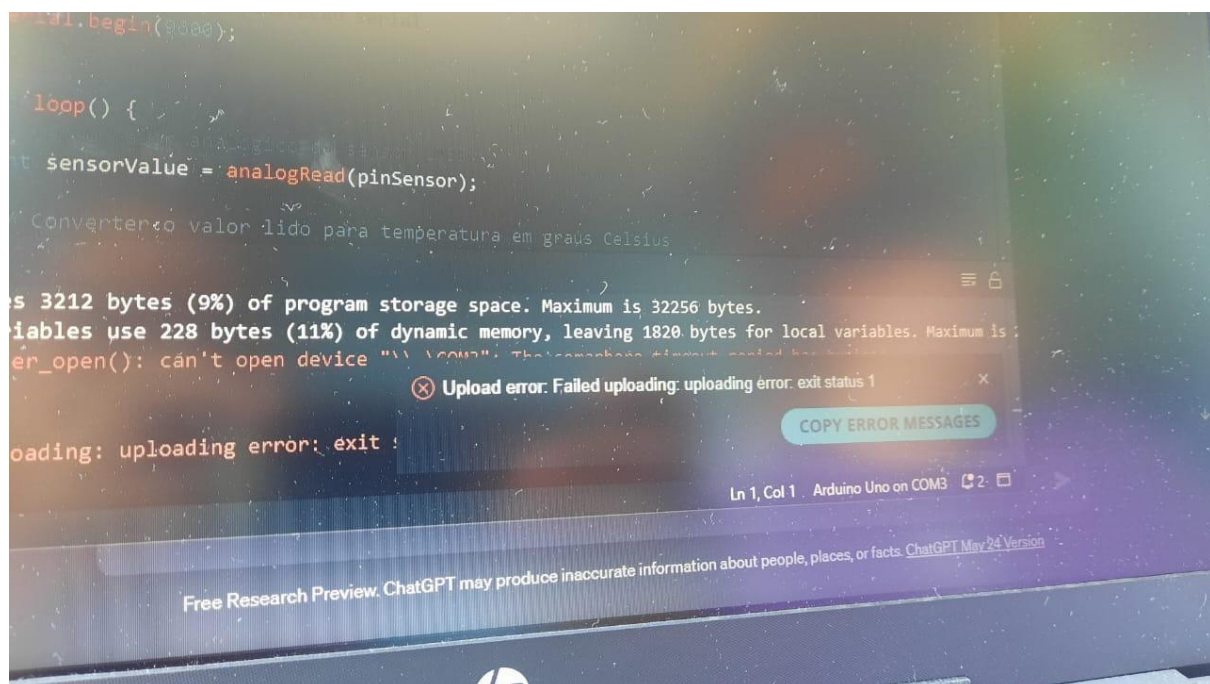


4. RESULTADOS

4.1 Testes

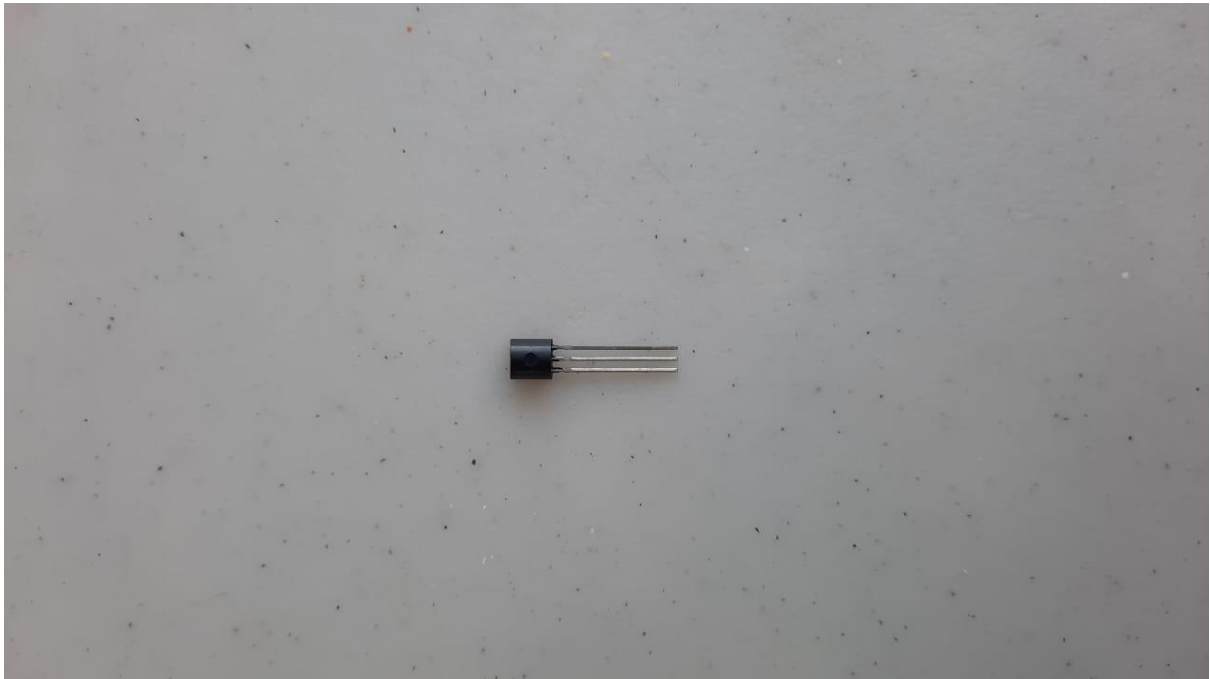
- Erro driver arduino

Após plugar o arduino pelo usb no computador, houve um erro ao tentar o upload do software pelo IDE. Se mostrou necessário a instalação manual do driver CH341 para o funcionamento correto da comunicação da IDE e o arduino UNO no windows 11.



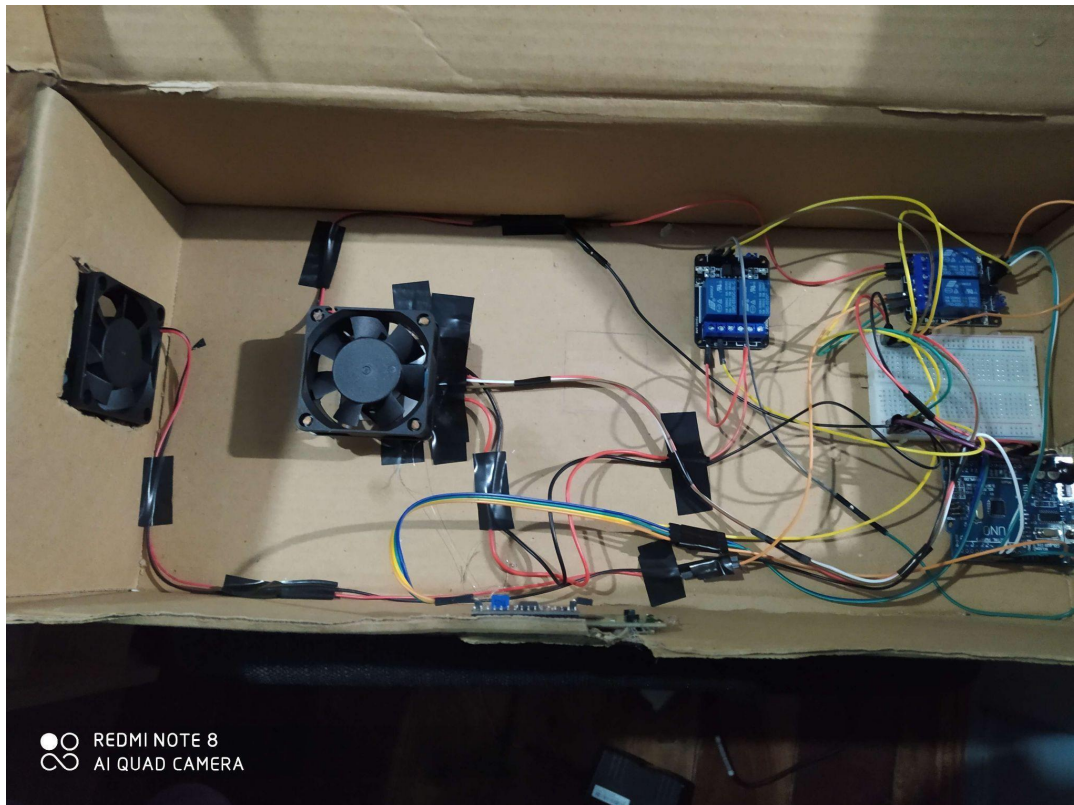
- Erro com sensor LM35

Inicialmente o planejamento era a utilização do sensor lm35 para a verificação da temperatura, porém ocorreram instabilidades durante os testes, possivelmente um mau funcionamento do sensor, foi realizada a troca para o sensor dht11.



- Teste finais

Após a troca de todas as peças necessárias, foi realizada a montagem em uma caixa e realizados os testes novamente para confirmar a funcionalidade.



4.2 Análise de dados

O sistema foi testado em um ambiente controlado, onde foram simuladas diferentes condições de temperatura para verificar o desempenho do sistema de refrigeração. Durante os testes, foram coletados dados de temperatura, umidade e o estado dos coolers, bem como o tempo de acionamento e desligamento da peltier.

Os resultados obtidos demonstraram um funcionamento eficaz do sistema de controle de refrigeração com base na leitura de temperatura. Quando a temperatura ambiente ultrapassava 30 graus Celsius, o cooler 1 era acionado para resfriar o ambiente. Além disso, quando a temperatura ultrapassava 35 graus Celsius, o cooler 2 era ligado para intensificar o resfriamento.

A peltier, que simula a geração de calor de um processador, era acionada quando a temperatura ambiente caía abaixo de 30 graus Celsius. Quando a temperatura atingia ou ultrapassava 45 graus Celsius e persistia por um período de 10 segundos, a peltier era desligada até que para evitar superaquecimento. Após ela retornar a uma temperatura menor que 30 graus, a peltier era religada automaticamente.

Os resultados também foram acompanhados pela exibição das informações de temperatura e umidade no display LCD. O display mostrava a temperatura atual e a umidade relativa, permitindo um monitoramento constante das condições ambientais.

Durante os testes, observou-se que o sistema de refrigeração respondia de forma rápida e eficiente às variações de temperatura. Os coolers e a peltier eram acionados e desligados conforme as condições estabelecidas no código, garantindo um controle preciso da temperatura ambiente simulada.

Os resultados obtidos confirmam a viabilidade do sistema proposto, demonstrando sua capacidade de simular a geração de calor de um processador e controlar o resfriamento com base na leitura de temperatura e umidade. Além disso, a exibição das informações no display LCD facilita o acompanhamento e monitoramento das condições ambientais em tempo real.

No entanto, é importante ressaltar que os resultados obtidos foram baseados em testes realizados em um ambiente controlado e podem variar em diferentes condições de uso. Portanto, é recomendável realizar testes adicionais em situações mais próximas da realidade para validar o desempenho do sistema.

5. DIÁRIO DE BORDO

26/05

Nos reunimos no Discord e organizamos a ideia do projeto e separamos as peças necessárias, fazendo um rascunho e teste teórico pelo site Tinkercad sobre as entradas, voltagens e quantidade de cabos necessários para o projeto. Também foi decidido usar uma caixa para colocar as peças com a intenção de simular um gabinete.

29/05

Lucas foi na 24 de Maio comprar as seguintes peças:

- 2 Coolers;
- 1 Pastilha Peltier;
- 1 Arduino Uno;
- 1 Sensor de Temperatura LM35;
- 1 Tela LED;
- Conjunto de 40 Cabos Macho / Fêmea;
- 2 Relê Duplos;

01/06

Desde a compra dos materiais, fomos juntando vídeos no YouTube e separamos de acordo com as nossas peças. Após filtrar de acordo com os testes na IDE do Arduino, finalizamos a parte inicial do código relacionado ao sensor de temperatura e acionamento dos coolers e pastilha peltier.

03/06

Foi marcado para nos reunirmos e montar as peças no sábado na casa do Guilherme. Montamos e conectamos os cabos de acordo com o teste teórico do Tinkercad, testando a voltagem de cada peça e soldado os cabos. Também verificamos que a Tela LED precisava de outro módulo para se comunicar com o Arduino e resolvemos comprar a peça depois.

Após as conexões serem testadas, foi feita uma tentativa de upload pelo IDE do Arduino para o Arduino, porém ocorreu um erro e foi necessário procurar a solução, após 1 hora de várias tentativas, descobrimos que o erro era devido a falta de Driver do Arduino no Windows 11, que não foi reconhecido pelo sistema automaticamente.

Ao terminar o Upload e testar, houve outro erro, este era relacionado ao não acionamento dos coolers após a temperatura estipulada no código, que foi resolvido ao trocar os cabos positivo e negativo de lugar.

Testando várias vezes, fizemos alguns ajustes finos devido ao sensor de temperatura LM35 estar muito inconstante ao abaixar e subir a temperatura como se estivesse falhando. Verificamos que a lógica estava funcionando e alteramos os planos para comprar outra peça como sensor de temperatura.

Por fim, ajeitamos na Caixa as peças com cola quente e foi passado uma pasta térmica na Pastilha Peltier para não esquentar tanto.

05/06

Lucas foi comprar as peças restantes, e foi comprado o sensor DHT11 e o display com módulo I2C integrado.

06/06

Foi realizada a montagem das outras peças e adicionado na programação a lógica relacionada a tela LED para visualização.

14/06

Revisamos antes da apresentação se tudo estava de acordo no local, onde ocorreu um erro de execução devido a falta de biblioteca no notebook usado para refazer o upload, então baixamos as bibliotecas necessárias e funcionou como esperado

6. CONCLUSÃO

Neste projeto, foi desenvolvido um sistema de controle de refrigeração que simula a geração de calor de um processador e controla a temperatura com base na leitura de um sensor de temperatura e umidade. Os testes realizados mostraram que o sistema foi eficiente no controle da temperatura ambiente, acionando os coolers conforme necessário.

Em suma, o sistema de controle de refrigeração desenvolvido neste projeto apresentou resultados promissores no controle da temperatura simulada de um processador. Porém apesar de demonstrar um desempenho satisfatório, é importante ressaltar que em uma aplicação real seria necessário aprimorar o sistema de controle de refrigeração. A utilização de um cooler de maior potência e um dissipador de calor adequado seriam indispensáveis para garantir uma dissipação eficiente do calor gerado pelo processador. Essas melhorias seriam fundamentais para evitar o risco de superaquecimento e garantir o funcionamento adequado do sistema em um cenário de uso real.