

UNIVERSIDADE FEDERAL DE ITAJUBÁ



Engenharia de Controle e Automação
ECOP04 (Programação Embarcada) - 2021.1

Projeto Final da Disciplina **Protótipo de *Cooling Pad***

Desenvolvido por:

Gabriel Cortes Cassiano Pereira

ccpereiragabriel@gmail.com

Julho - 2021
Itajubá

1. INTRODUÇÃO

O projeto desenvolvido e aqui documentado consiste no protótipo de um *cooling pad* (base refrigerada) para notebooks, bem como sua simulação. Um *cooling pad* é uma mesa de apoio com ventoinhas acopladas em sua estrutura que auxiliam na ventilação e resfriamento do notebook.



Figura 1 - Exemplo de base refrigerada (NBC 100BK - www.c3technology.com.br)

O principal conceito do *cooling pad* desenvolvido é que ele possua diferentes modos de operação, sendo eles:

- Modo desligado;
- Modo manual;
- Modo automático.

Em cada um dos modos o *cooling pad* irá interagir de uma forma distinta com o usuário e irá assumir diferentes comportamentos, com o intuito de se utilizar diferentes componentes analógicos/digitais e empregar os diversos conceitos vistos e absorvidos na disciplina de ECOP04 (Programação Embarcada) e seu respectivo laboratório, ECOP14.

O *cooling pad* deve possuir uma interface gráfica amigável, que permita ao usuário utilizá-lo sem dificuldades, bem como ser bastante responsivo para melhor experiência. E o mais importante, ele deverá ter um impacto significativo no resfriamento do notebook para que se justifique seu uso. Todos esses pontos foram abordados no projeto e serão detalhados neste relatório.

2. ESBOÇO DO PROJETO

2.1. Componentes utilizados e suas funcionalidades

Todo o projeto foi desenvolvido através do software **PICSimLab**, desenvolvido por L. C. Gamboa. Utilizou-se o kit educacional **PICGenios** juntamente com o **PIC18F4520**, ambos abordados na disciplina.

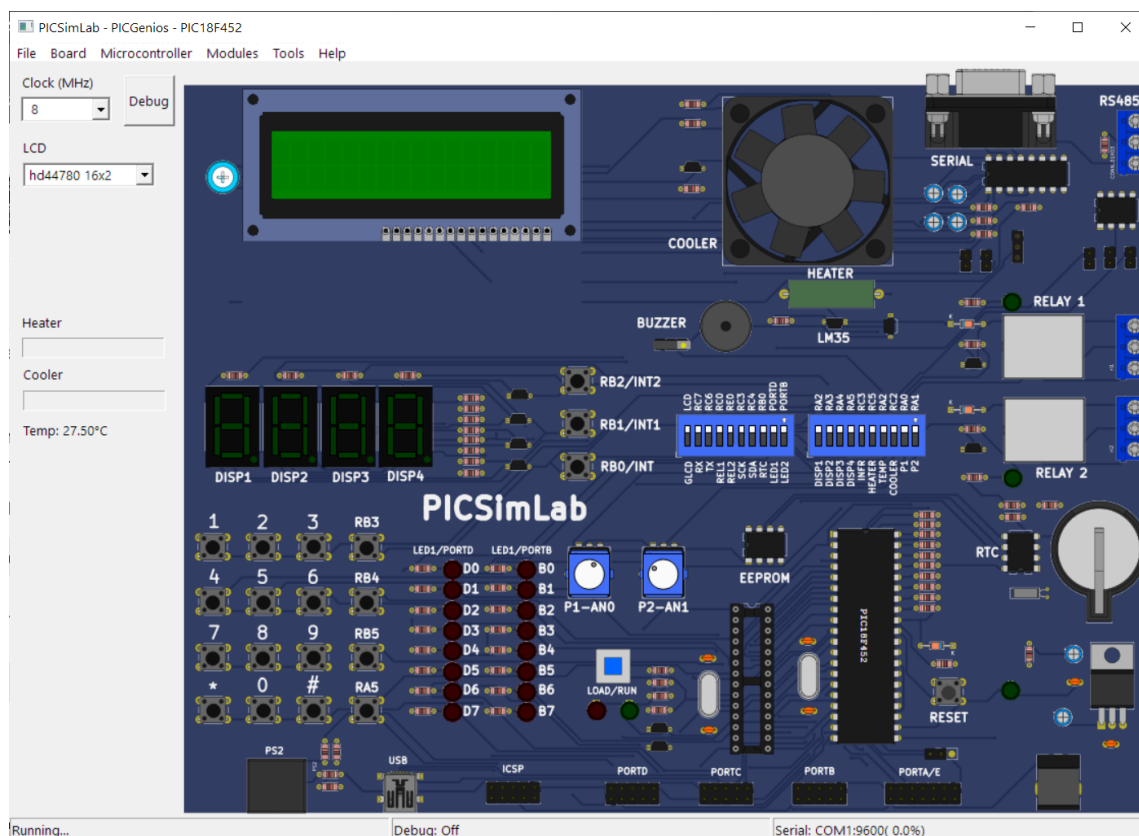


Figura 2 – PICSimLab com a placa PICGenios

A interface do *cooling pad* foi desenvolvida com o **display LCD HD44780 16x2**, apresentando instruções e informações para o usuário durante seu funcionamento. O LCD também possui um menu de seleções de modo para que o usuário selecione um dos três modos existentes. Para auxiliar na interface se utilizou **LEDs e displays de 7 segmentos**. Seu intuito é de exibir alguns dados adicionais de extrema importância para o projeto.

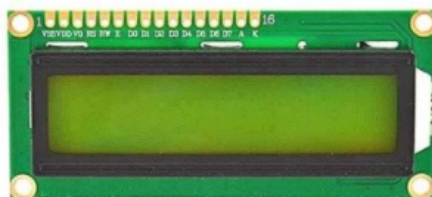


Figura 3 –LCD 16x2 genérico

A navegação pelos menus e algumas outras interações com o *cooling pad* foram realizadas através do **teclado matricial** presente na placa, o qual o usuário pode utilizar com facilidade. Além disso, um dos **potenciômetros** foi implementado para que o usuário realize configurações nos modos de funcionamento do protótipo.

O *cooler* da placa é utilizado para simular os *coolers* do *cooling pad*, e a **resistência de aquecimento** para simular o aquecimento do notebook. Um ponto importante do projeto é a utilização do **sensor LM35** para medir a temperatura e determinar o impacto do *cooling pad* no notebook. Note que no projeto o LM35 mede a temperatura da placa, mas considere que é a temperatura de um notebook para fins de simulação.

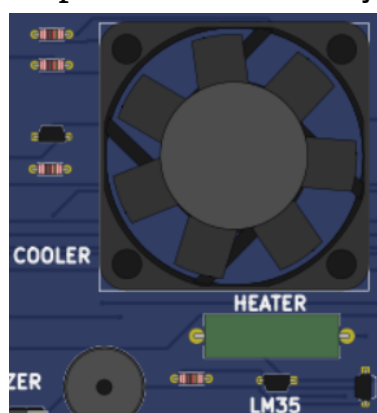


Figura 4 –Conjunto Cooler, Resistência e sensor LM35

2.2. Modos de funcionamento

Em relação aos diferentes modos de funcionamento do protótipo, o modo desligado simula o comportamento do notebook sem o auxílio da base refrigerada.

O modo manual permite ao usuário definir a velocidade da ventoinha da base refrigerada e apresentar o seu resultado na performance do notebook. A velocidade será controlada pelo potenciômetro, podendo ir de 0 até 100%.

O terceiro e último modo é o modo automático. Nele o usuário define um *set point* de temperatura de trabalho, e o próprio *cooling pad* irá controlar a velocidade da ventoinha para que se atinja e mantenha a temperatura desejada. A temperatura de *set point* é definida com o potenciômetro pelo usuário.

Por fim, em qualquer um dos modos de funcionamento será possível ativar/desativar a resistência de aquecimento (*heater*) para simular atividades intensivas do notebook e aquecê-lo. Isso é essencial para testar o funcionamento de todos os modos.

3. DESENVOLVIMENTO DO PROJETO

3.1. Menu

O primeiro elemento a se planejar foram os menus os quais seriam mostrados no display LCD. Por se tratar de um cooling pad não era necessário nada muito complexo, apenas uma interface que fosse intuitiva para o usuário. Pensando na questão do display 16x2 a ideia dos menus é:

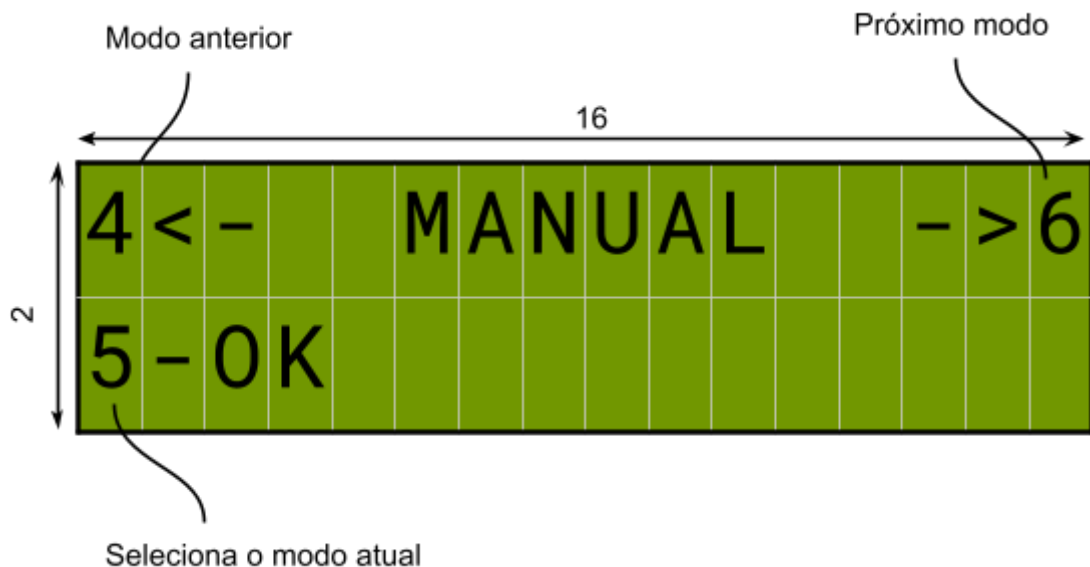


Figura 5 -Esboço do Menu LCD

Os números são referentes às teclas do teclado matricial que devem ser pressionadas para realizar tal ação. Dessa forma o programa deve ficar em um looping no menu até que se escolha um dos três modos:

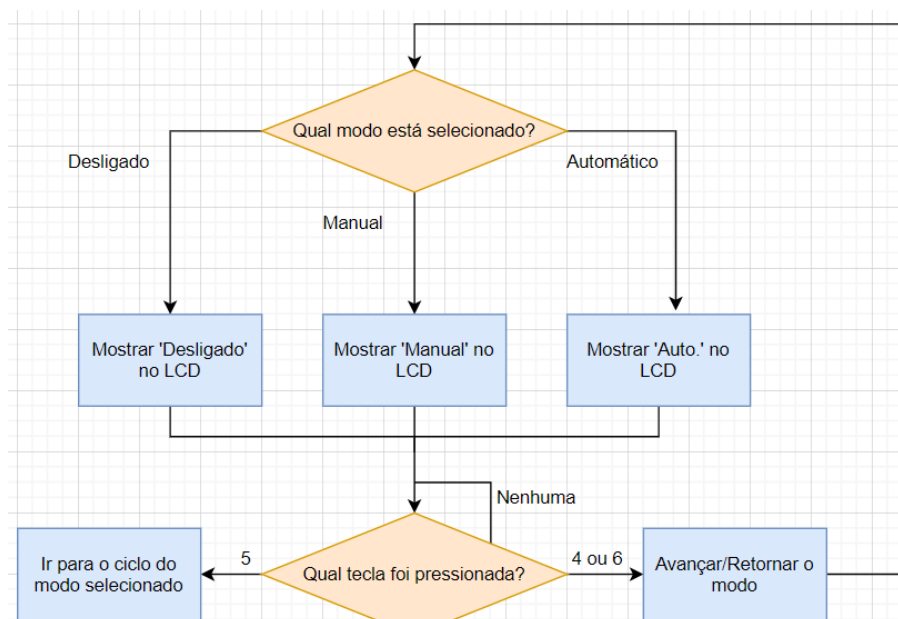


Figura 6 -Fluxograma do Menu

O próximo passo foi definir o funcionamento de cada um dos três modos que estarão disponíveis. Eles foram projetados na ordem do mais simples ao mais complexo, começando pelo Desligado, depois Manual e por fim Automático. Dessa forma foi possível ir implementando cada um dos componentes e funcionalidades da placa aos poucos, aproveitando o resultado obtido no modo mais simples e o reutilizando nos modos mais complexos.

3.2. Modo desligado

O modo desligado tem como objetivo simular o aquecimento do notebook, sem a ação do *Cooling Pad*, para posteriormente verificar o quanto a base refrigerada influencia na performance de temperatura do notebook.

Selecionando o modo o programa entrará no respectivo ciclo referente ao mesmo. Nesse ciclo a informação principal a ser mostrada no LCD é a temperatura medida pelo sensor LM35, que no caso simula a temperatura do notebook.

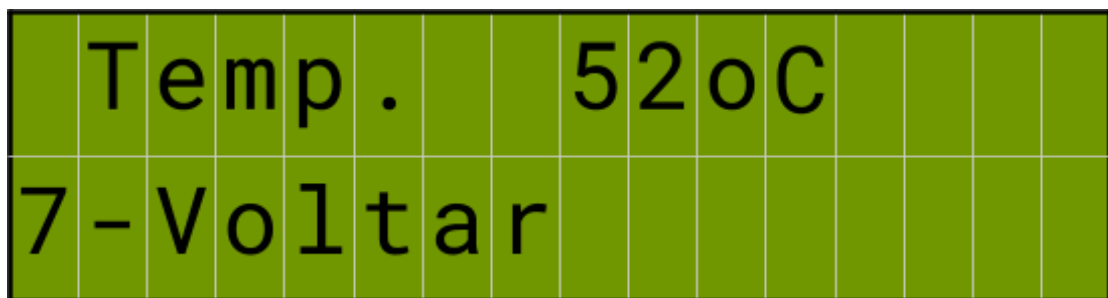


Figura 7 -Esboço do modo Desligado no LCD

Para se ler a temperatura do notebook, o primeiro passo foi entender como ler os dados do sensor LM35 e também como interpretá-los. Pelo esquemático da placa do PICGenios verifica-se que a leitura do sensor deve ser realizada pelo pino AN2 do PIC.

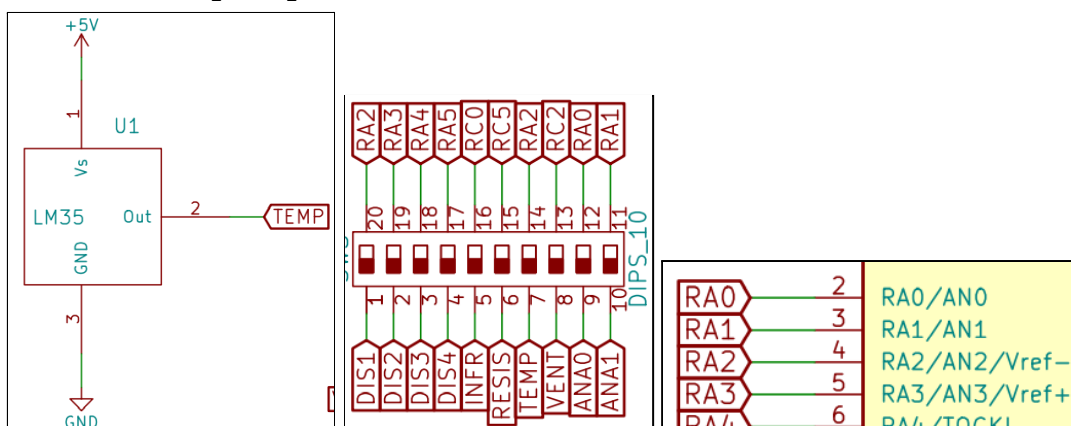


Figura 8 -Caminho do sensor LM35 até o PIC18F4520

Como se trata de uma entrada analógica, configura-se também o conversor AD do PIC para se realizar a leitura dos dados do sensor. Pelo datasheet do PIC verifica-se que o conversor AD é controlado pelo registrador ADCON0. Configurando os bits de seleção de canais corretamente e através do GO e ADON é possível realizar a leitura da temperatura.

REGISTER 19-1: ADCON0: A/D CONTROL REGISTER 0

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON
bit 7							bit 0

Figura 9 – Bits de controle do registrador ADCON0

Com a leitura de temperatura funcionando, o próximo passo foi simular o aquecimento do notebook. Isso foi feito através da resistência de aquecimento ativada pelo bit 5 do registrador C. Isso é verificado no esquemático da placa:

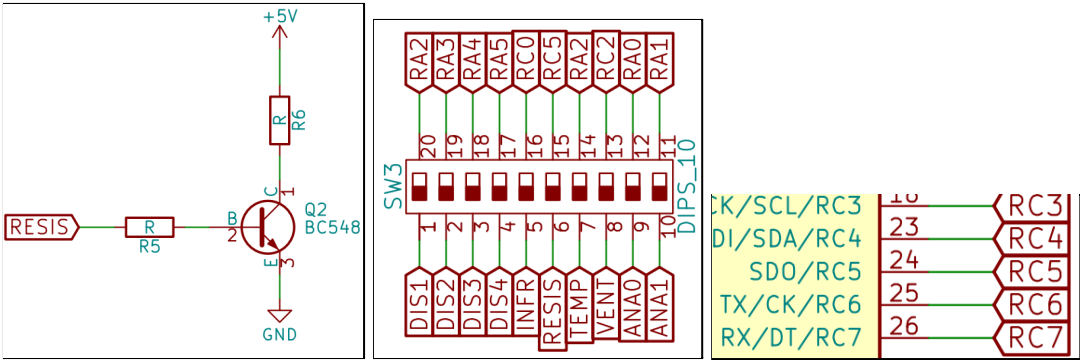


Figura 10 – Caminho da resistência de aquecimento até o PIC18F4520

O aquecimento será iniciado pressionando a tecla 8 para um melhor controle. Dessa forma o modo desligado está finalizado. Foi muito importante se iniciar por esse modo, pois todas essas funcionalidades serão replicadas nos modos Manual e Automático.

3.3. Modo manual

No modo manual a ventoinha foi inserida no conjunto de componentes. O *Cooling Pad* deve ter sua velocidade controlada manualmente pelo usuário. Isso foi realizado com o uso de um dos potenciômetros.

Nesse modo foram inseridos os displays de 7 segmentos para mostrar mais informações que não couberam no LCD. Os displays foram utilizados para mostrar a velocidade em porcentagem das ventoinhas.

Para colocar a ventoinha da placa para funcionar utiliza-se o bit 2 do registrador C do PIC.

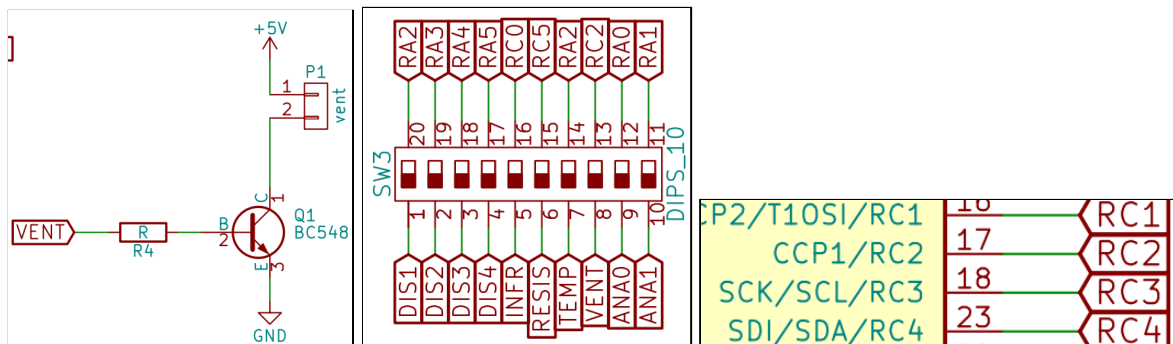


Figura 11 –Caminho da ventoinha até o PIC18F4520

Para ativar a ventoinha foi bem simples. Porém, apenas ligá-la ou desligá-la não bastava para o projeto. Era necessário controlar sua velocidade de rotação. Para isso se utilizou um **sinal PWM**. Note, observando o esquemático acima, que o pino RC2 é compartilhado com CCP1. O CCP1 é responsável pelo controle PWM, portanto é possível utilizá-lo para controlar a velocidade da ventoinha do *Cooling Pad*.

Com o cooler funcionando, o último passo era implementar o controle da velocidade através do potenciômetro P1 da placa. Ele está conectado no pino RA0 do PIC18F4520, portanto basta ler as informações desse pino para obtermos o valor do potenciômetro.

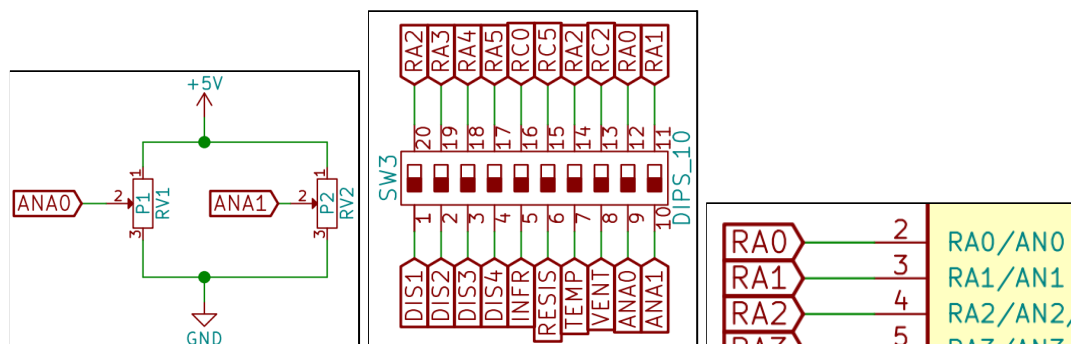


Figura 12 –Caminho do potenciômetro P1 até o PIC18F4520

Utilizou-se novamente o conversor AD para a leitura desse dado. Testando esses dados constatou-se que a informação chegava em um range de 0 a 1023. Para converter isso para porcentagem foi simples. Para facilitar o cálculo apenas se dividiu o valor lido por 10, resultando em valores de 0% a 102,3%. Os valores maiores que 100% foram tratados para serem também considerados 100%. Por fim a porcentagem obtida foi transferida para um sinal PWM que controla a velocidade da ventoinha.

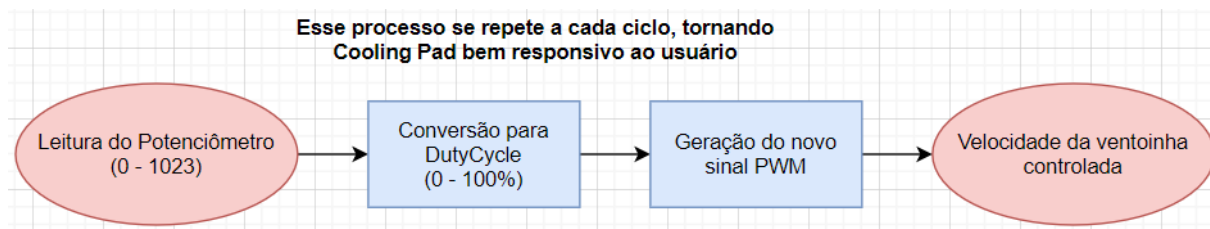


Figura 13 –Fluxograma do controle da ventoinha

Com o controle da ventoinha sendo feito pelo potenciômetro, o modo Manual foi concluído. Esses conceitos e funcionalidades serão reutilizados no modo Automático, que é o último e mais complexo.

3.4. Modo Automático

No modo automático o usuário define um *set point* utilizando o potenciômetro. Esse *set point* corresponde à temperatura que se deseja alcançar e manter no notebook. A complexidade está justamente em relacionar a temperatura medida com a velocidade da ventoinha.

Abordando a questão da relação temperatura/velocidade, essa relação é quase linear. Realizando alguns testes obteve-se em quais temperaturas o “notebook” estabilizou para alguns *duty cycles* (que controla a velocidade do cooler):

Duty Cycle (%)	Temperatura (°C)
80	38
70	43
60	49
50	54
40	59
30	64

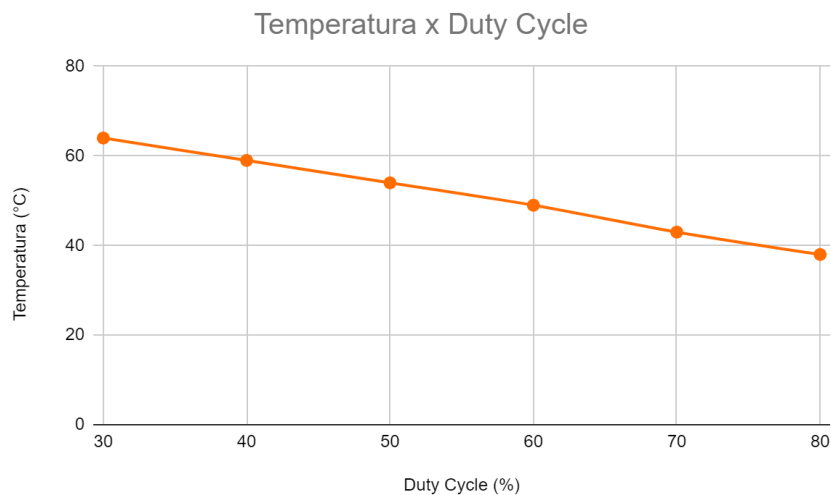


Figura 14 -Gráfico da relação Temperatura x Duty Cycle

A relação entre ambos pode ser considerada linear, o que por sua vez permite montar uma expressão para o *duty cycle* em função da temperatura de *set point*. $54a + b = 50$, $59a + b = 40$. Disso obtemos $a = -2$, $b = 158$, portanto:

$$dutyCycle = -2setPoint + 158$$

Onde o *duty cycle* indica a velocidade do cooler e o *set point* a temperatura que se deseja alcançar e estabilizar. Um problema, porém, é que para valores de temperatura acima de 80°C o *duty cycle* daria negativo, o que não é possível fisicamente. Portanto limita-se a temperatura de *set point* para 80°C e altera-se levemente a equação: $dutyCycle = -2setPoint + 160$.

Porém não é interessante o *cooling pad* trabalhar na velocidade de estabilização fixa. É necessário que esse aumento de velocidade seja gradual e que inclusive seja maior em alguns casos nos quais a temperatura do notebook ultrapasse o *set point* por algum motivo. Para isso multiplicamos o resultado por uma relação de *temperatura medida/set point*.

$$dutyCycle = (-2setPoint + 160) \times \frac{temperatura\ medida}{set\ point}$$

Em código:

```
if (tmpi >= 30) {
    dutyCycle = (160 - 2*setPoint) * ((float) tmpi/ (float) setPoint);
    pwmSet(dutyCycle); //Ajusta ciclo de trabalho do PWM
}
```

Onde *tmpi* é a temperatura medida. Converte-se para float pois os dados obtidos do sensor são inteiros, bem como o *set point* que será abordado agora.

Como dito anteriormente, o *set point* será definido pelo próprio usuário utilizando o potenciômetro. Restringiu-se o *set point* para valores entre 40°C e 80°C. O limite de 80 já foi explicado anteriormente, e o de 40 pois é o mínimo de temperatura que um notebook é capaz de atingir durante tarefas cotidianas.

Assim a conversão da leitura do potenciômetro para um *set point* em °C se dá da seguinte forma:

$$0 \text{ unidades} = 40^{\circ}\text{C} \quad \rightarrow \quad \text{setPoint} = 40 + \frac{(1023-0)}{(80-40)} \times \text{medida do potenciômetro}$$

$$1023 \text{ unidades} = 80^{\circ}\text{C} \quad \text{setPoint} = 40 + 25 \times \text{medida do potenciômetro}$$

Dessa forma o *cooling pad* conseguiu então ler o *set point* que é definido pelo usuário e trabalhar para manter a temperatura do notebook nesse valor através de cálculos e controle de sinais PWM. Resumindo a lógica para uma melhor compreensão:

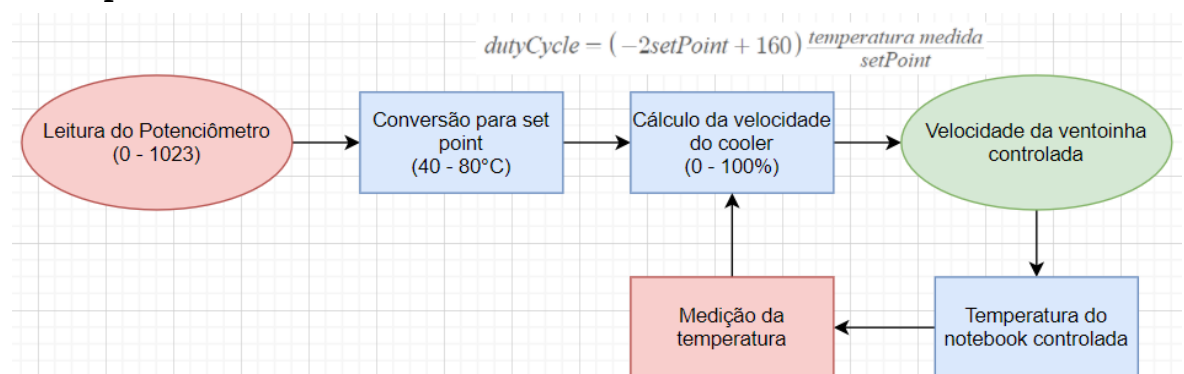


Figura 15 - Fluxograma do controle automático do cooler

Os displays de 7 segmentos foram utilizados para mostrar qual *set point* está selecionado pelo usuário, e qualquer alteração do mesmo é atualizada no mesmo instante.

3.5. Ajustes menores

Finalizado o desenvolvimento do projeto, constatou-se pequenas melhorias que poderiam ser feitas. Elas serão citadas aqui.

3.5.1. LEDs no modo desligado

Foram inseridos LEDs no modo desligado como uma forma de aviso. Eles começam a piscar quando a resistência de aquecimento é ativada. Servem para indicar que o notebook começou a aquecer, mas que não há nenhum auxílio de refrigeração devido ao funcionamento do modo.

3.5.2. Botão de ajuda rápida

Foi inserido um botão de ajuda nos modos manual e automático. Eles percorrem um texto no LCD explicando bem brevemente o uso dos potenciômetros nesses modos. Isso foi feito para dar um leve auxílio para o usuário, já que não há nenhum indicativo sobre os potenciômetros no menu.

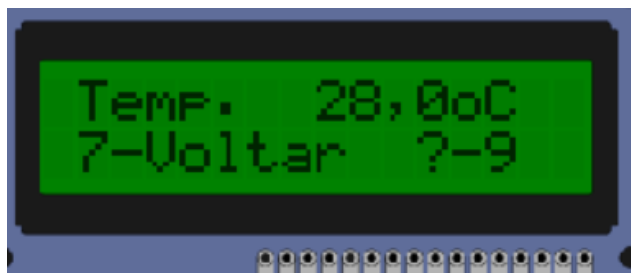


Figura 16 -Botão de ajuda (tecla 9)

3.6. Anexos

Ao final do projeto serão anexados para consulta o esquemático do PICGenios e um passo a passo do funcionamento do projeto do *cooling pad*. Caso surja a necessidade durante a leitura deste relatório basta ir para o final do arquivo.

4. CONCLUSÃO

O desenvolvimento do projeto se mostrou extremamente interessante e de muito aprendizado. Nele foi possível colocar em prática muito do conhecimento visto durante o semestre. Empenhou-se a criatividade necessária para se conceber a ideia, a parte teórica para planejá-la no papel e a parte prática desenvolvendo o protótipo no simulador.

O *cooling pad* não foi desenvolvido em uma placa física devido às limitações do regime de tratamento excepcional da UNIFEI. Ele foi desenvolvido em um software de simulação. Todavia foi possível aprender muito sobre o funcionamento de sistemas embarcados e como abordar projetos nessa área. A limitação não prejudicou a experiência.

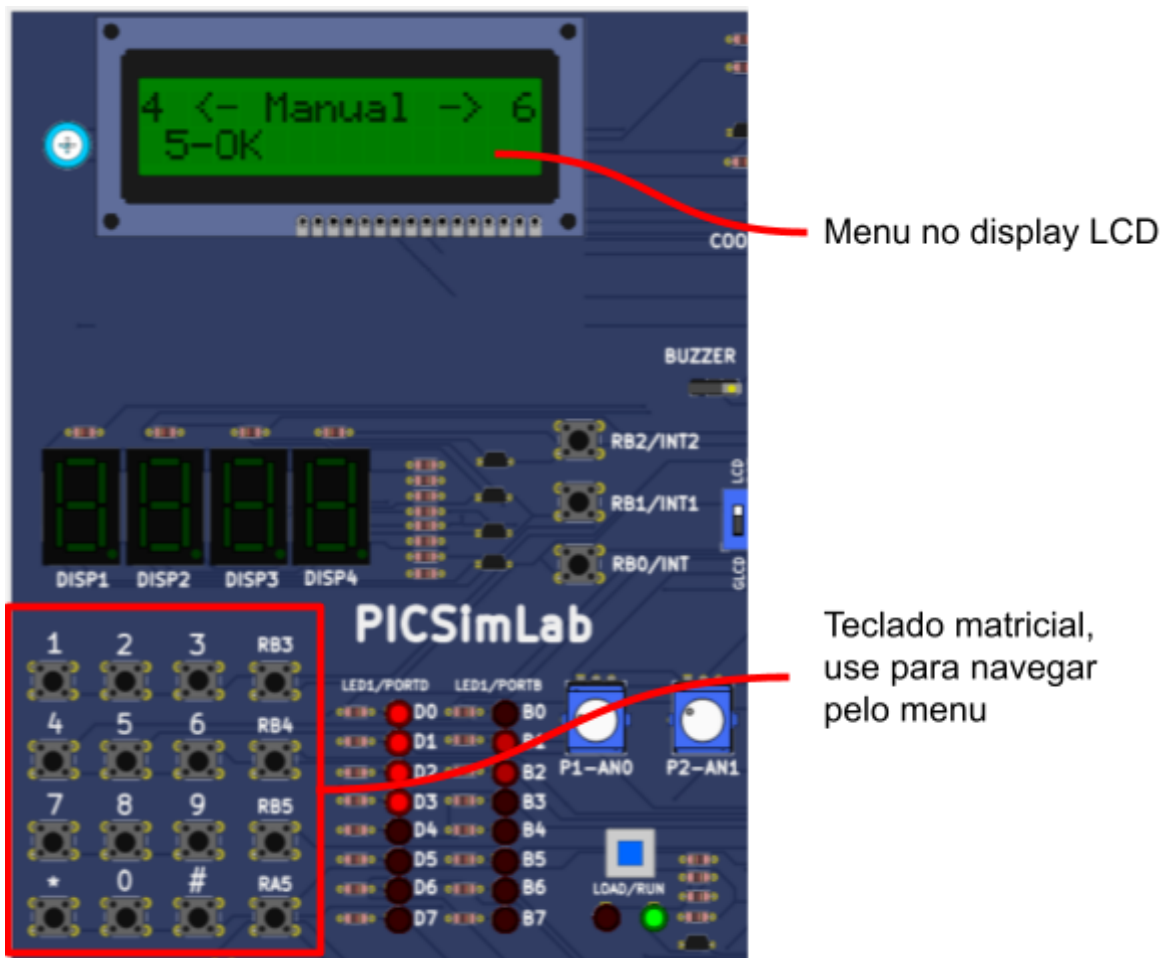
Focando mais especificamente na base refrigerada, é possível afirmar que o projeto atendeu às expectativas. Todas as funcionalidades planejadas estão presentes no *cooling pad*, bem como algumas pequenas melhorias. É possível realizar aperfeiçoamentos no projeto, tanto na parte dos componentes utilizados quanto na parte da programação também. Mas no geral ele atendeu bem às expectativas e agregou muito conhecimento durante sua execução.

ANEXOS

Passo a passo do funcionamento

O funcionamento do *cooling pad* é extremamente simples já que o seu objetivo é facilitar a vida do usuário.

1º Passo - Navegue no menu utilizando as teclas indicadas nos displays. São apenas três modos, então não tem como se perder. Selecione um dos modos.



2º Passo - Modo desligado

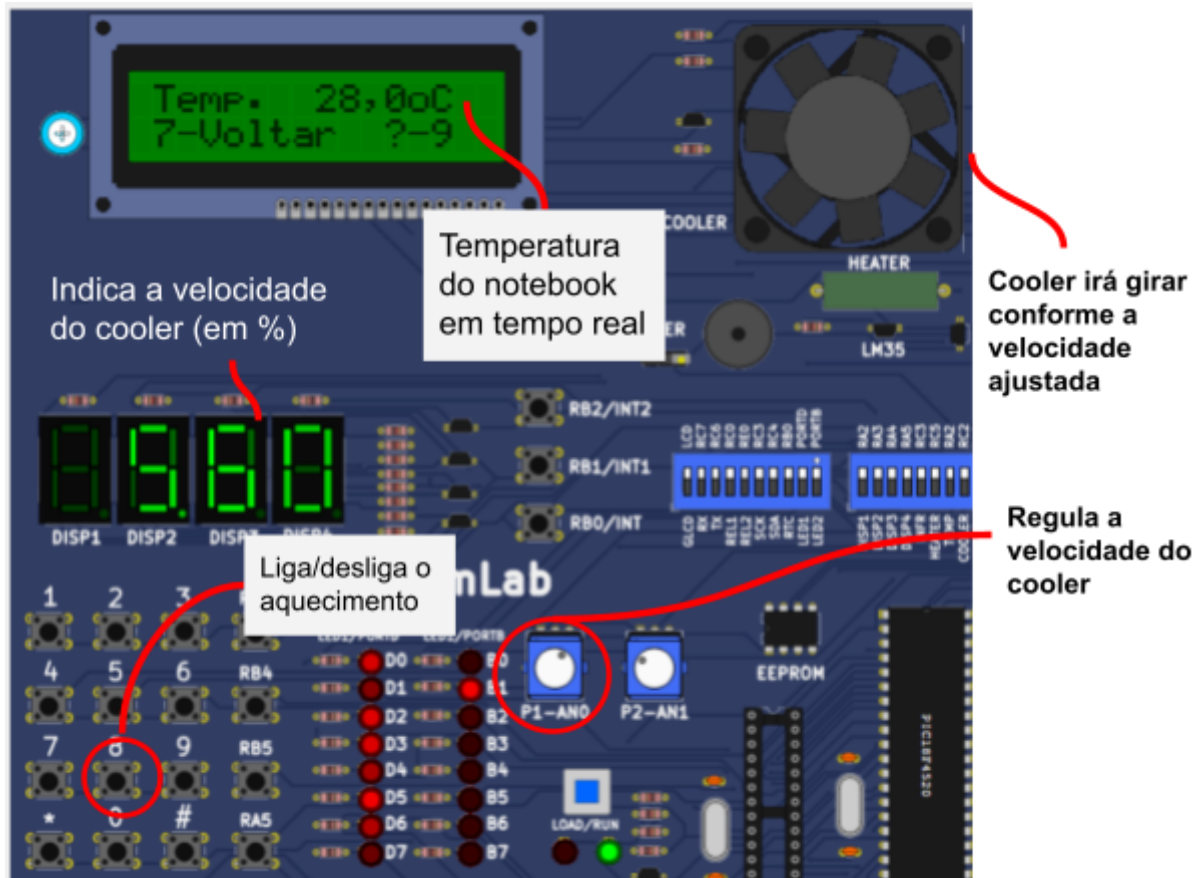


LCD mostra a temperatura em tempo real do notebook

LEDs piscarão indicando que o notebook está aquecendo

Pressione 8 para simular/parar o aquecimento do notebook

3º Passo - Modo manual



4º Passo - Modo automático



