Universidade Federal de Itajubá - UNIFEI

Relatório: Máquina de Caça-Níquel

Marcos Barbosa (2020016324)
marcos.barbosa@unifei.edu.br

Itajubá/2021

Marcos Barbosa (2020016324)
marcos.barbosa@unifei.edu.br

Relatório: Máquina de Caça-Níquel

Relatório apresentado às disciplinas de Programação Embarcada e Laboratório de Programação Embarcada, ministrados pelo Professor Otávio de Souza Martins Gomes como parte da avaliação do curso de Engenharia de Computação.

Itajubá/2021

Sumário

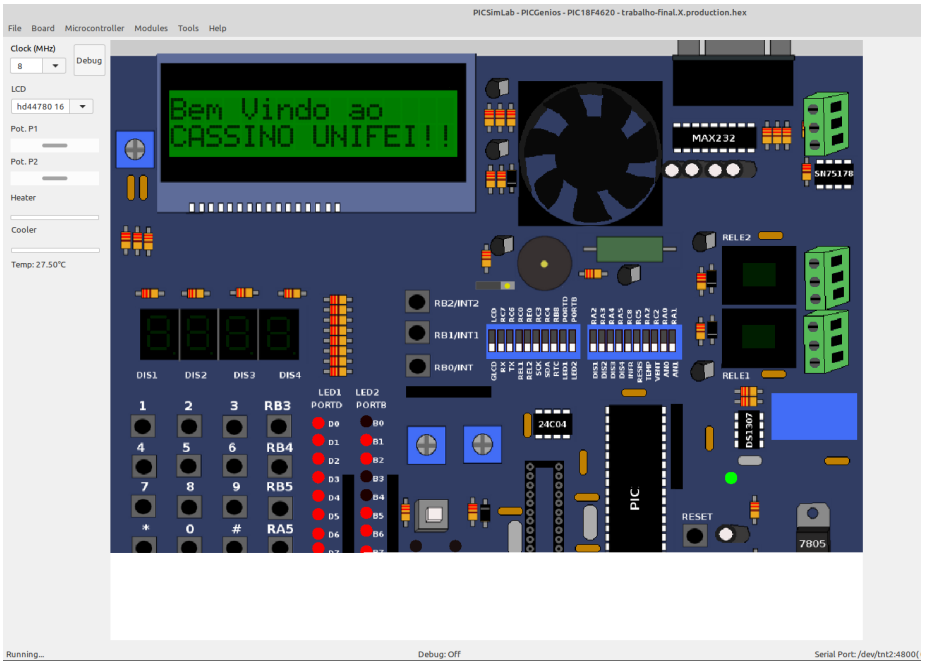
Introdução

Durante as aulas da disciplina de Programação Embarcada foi proposto a elaboração de um trabalho que pudesse reunir todos os conhecimentos adquiridos na disciplina, bem como a produção de um projeto prático e didático.

As ferramentas utilizadas foram o software MPLAB-X para a criação do código-fonte e o microcontrolador PIC18F4520, da família de 8 bits e núcleo de 14 bits, fabricado pela empresa MICROCHIP.

Para os testes e simulações foi utilizado o software brasileiro PICSimLab, acrônimo de “PIC Simulator Laboratory”, desenvolvido pelo professor Luis Claudio Gambôa Lopes, que emula a placa PICGenios e possui suporte ao software MPLAB-X.

Figura 1: Tela inicial do projeto



Fonte: Acervo do autor

Para o tema foi escolhido o desenvolvimento de “Máquinas de Caças-Níqueis", as mais famosas máquinas dos casinos, o desenvolvimento deste projeto visa apresentar de forma didática esse famoso embarcado com mais de um século de existência.

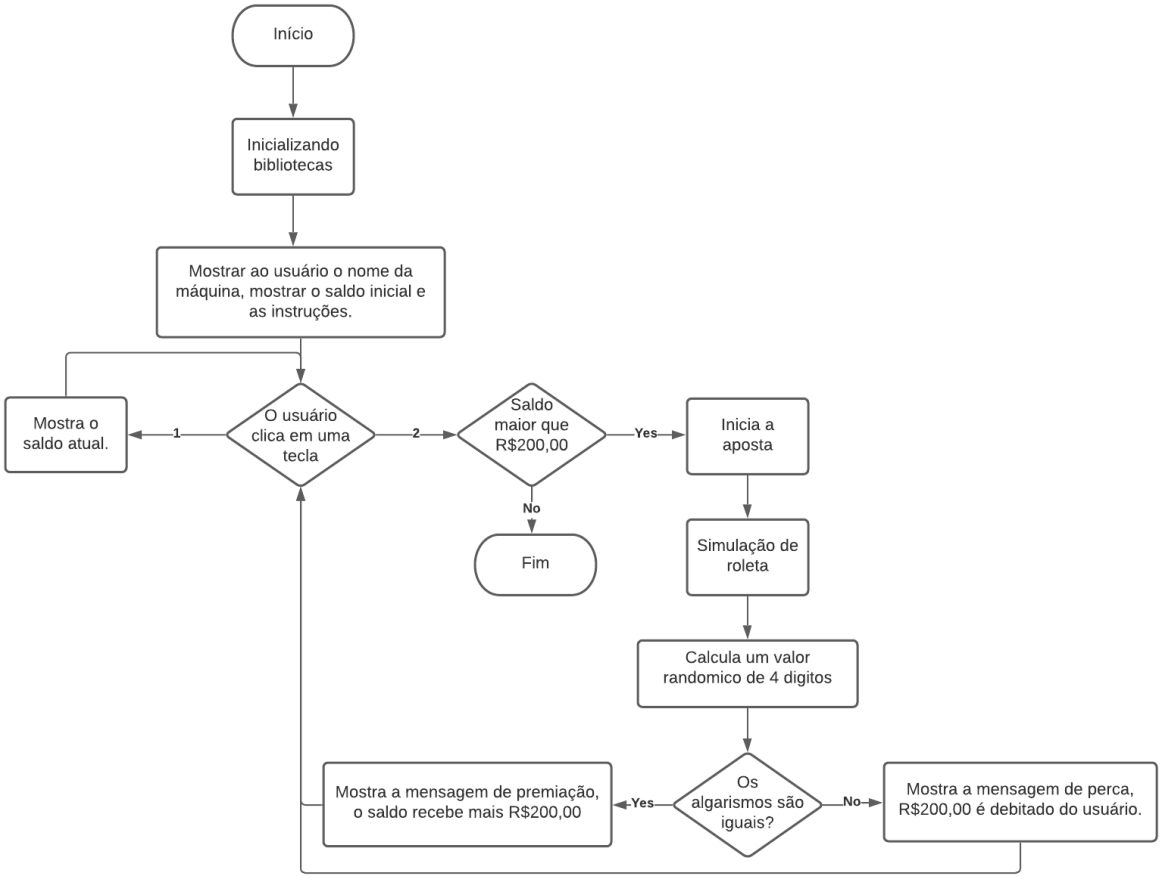
O projeto utiliza de diversos mecanismos da placa PICGenios:

- **Display LCD:** Será o menu e guiará o usuário durante a experiência de uso.

- **Display de 7 Segmentos:** Simulará a roleta, apresentando números que poderão fornecer sequências premiadas.
- **LEDs:** Será utilizado para simular animações antes do giro da “roleta”.
- **Cooler:** Com o mesmo objetivo dos leds, servirá para simular uma animação da “roleta”.
- **Teclado:** Será responsável pela comunicação entre embarcado e usuário, sendo assim, a interatividade do sistema.

O fluxograma do projeto pode ser visto logo abaixo:

Figura 2: Fluxograma do sistema embarcado



Fonte: Acervo do autor

Desenvolvimento

O sistema é dividido em etapas, sendo basicamente separadas entre: inicialização do sistema, instruções de uso, mostrar saldo e apostar. Essas etapas foram definidas para apresentar a melhor experiência de uso ao usuário, proporcionando comunicações dentro das etapas, como o saldo insuficiente para uma nova aposta ou o ganho.

I. Inicialização do sistema

Nessa etapa é utilizado o display 16x2 do PICGenios, ele é usado para mostrar uma mensagem de inicialização da aplicação, mostrando o título da aplicação, veja a figura 1 acima.

Foram criadas várias funções nessa etapa, são elas:

textInit(): Função responśavel por transmitir a mensagem inicial do projeto.

escreve(char text[]): Função que converte um texto simples ao display de 16x2.

delay(): Função que dura cerca de 2s, responável por dar tempo entre outras funções.

II. Instruções de uso

As instruções de uso são mostradas logo após a inicialização do sistema, para isso é utilizado novamente o display de 16x2, ele mostra as opções da aplicação, são elas:

- 1 - Apostar
- 2 - Ver Saldo

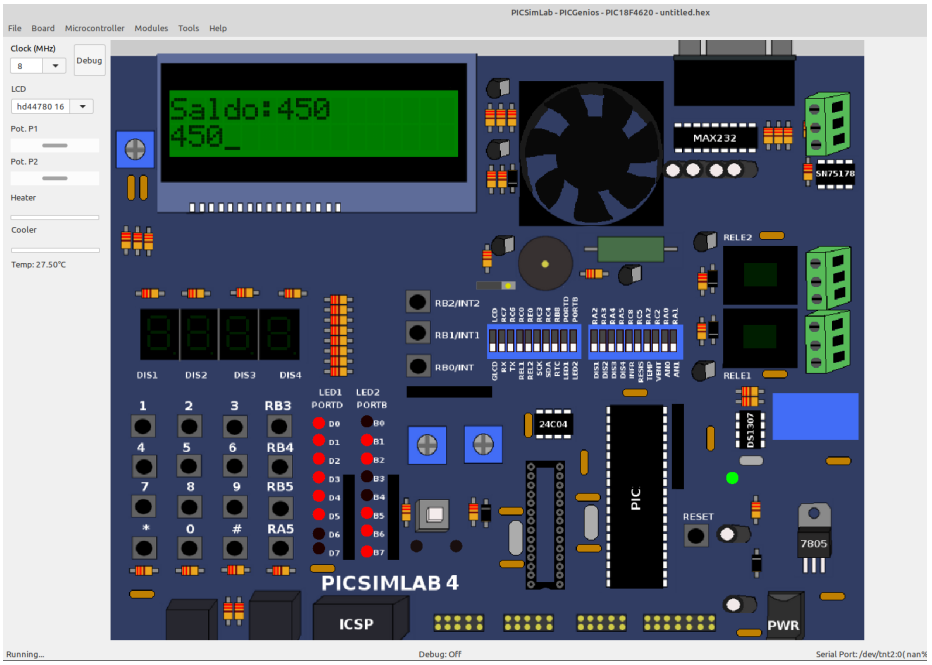
Foram criadas várias funções nessa etapa, são elas:

textInstrutions(): Função responsável por mostrar de frma visual as opções do sistema embarcado.

III. Ver Saldo

Essa função permite ao usuário saber o saldo atual dele no jogo.

Figura 3: Tela de saldo atual



Fonte: Acervo do autor

Foram criadas várias funções nessa etapa, são elas:

textSaldo(): Função que mostra ao usuário seu saldo atual.

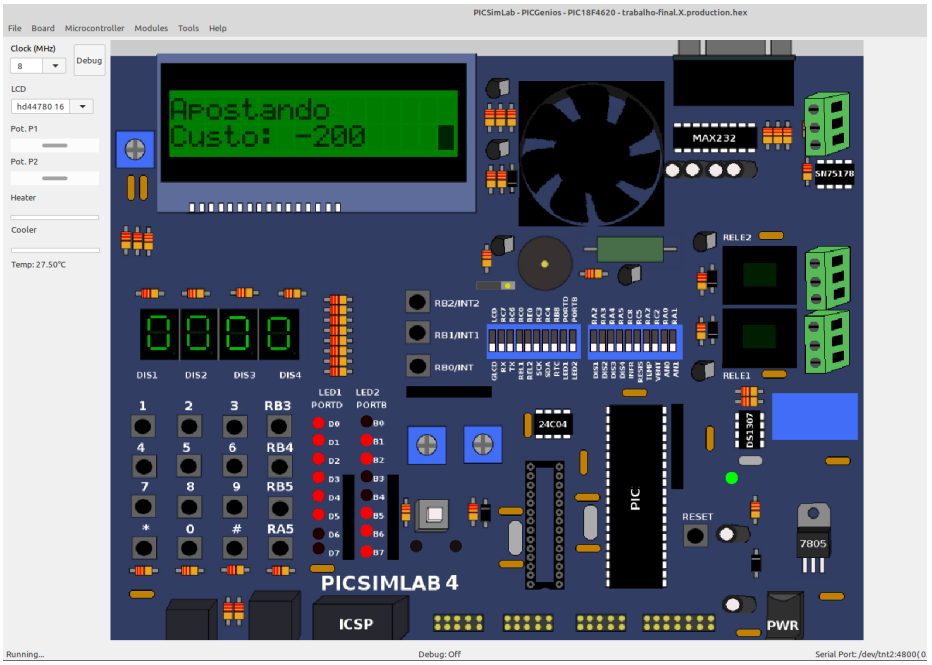
IV. Apostar

Essa é a principal funcionalidade da aplicação e nessa etapa todos os componentes do projeto são usados.

Ele primeiramente mostra uma mensagem ao usuário que a aposta está sendo iniciada, depois ele inicia uma animação da roleta, utilizando o barramento de leds e o cooler.

Depois ele gera um número aleatório de 4 dígitos, que é mostrado dos 4 displays de 7 segmentos, caso os 4 valores forem iguais, por exemplo: 4444 ou 1111, o usuário ganha uma bonificação, mas caso perca, por exemplo: 4325 ou 7888, o usuário tem uma quantidade debitada de sua conta. Caso a quantidade de saldo do usuário seja inferior ao valor da aposta mínima o usuário não consegue mas apostar, com isso a máquina transmite um “Game Over” ao usuário, não permitindo mais apostas.

Figura 4: Tela de aposta



Fonte: Acervo do autor

Foram criadas várias funções nessa etapa, são elas:

textApostando(): Função responsável por iniciar as apostas e controlar o cooler para a animação da roleta.

textGameOver(): Função chamada caso o saldo do usuário seja inferior ao da aposta mínima (R$200,00), ela transmite a mensagem de que o jogo terminou.

verso(): Função utilizada para a animação do barramento de LEDs, iniciando de cima para baixo, simulando a animação de roleta.

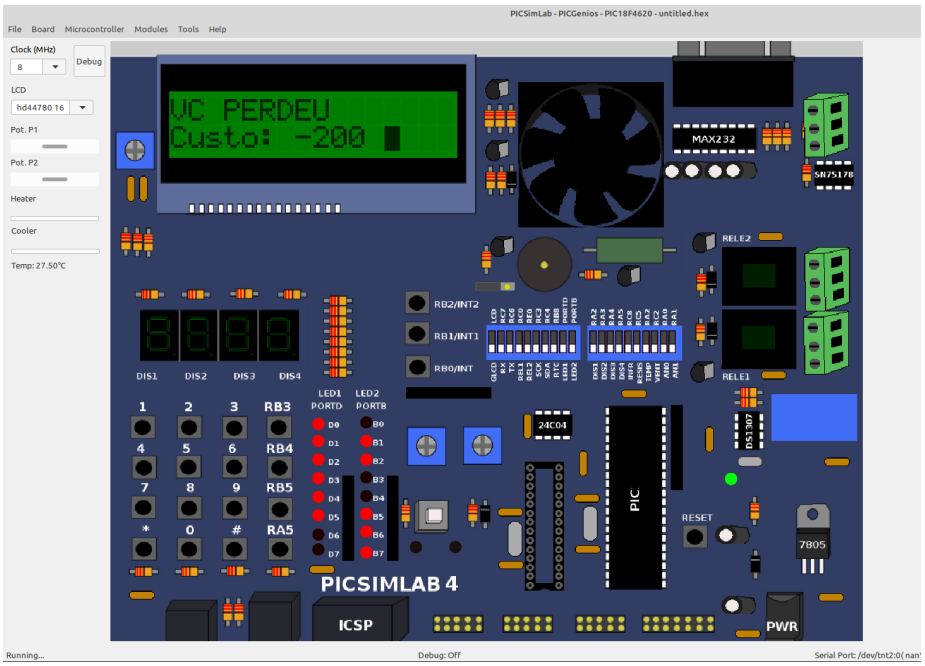
inverso(): Função utilizada para a animação do barramento de LEDs, iniciando de baixo para cima, simulando a animação de roleta.

apostando(): Função responsável por randomizar um valor de 4 dígitos e mostrar aos 4 dispositivos de display de 7 segmentos.

textWin(): Função responsável por anunciar a vitória ao usuário caso o número sorteado seja com os 4 dígitos iguais, como o número 1111 ou 4444. Além de adicionar o fundo de vitória.

textLost(): Função responsável por transmitir a mensagem de que o valor sorteado não foi acertado, assim, o crédito ao usuário será debitado e uma mensagem de perda será mostrada.

Figura 5: Tela de perda



Fonte: Acervo do autor

V. Main

Essa função serve para juntar todas e inicializar as bibliotecas do sistema, permitindo utilizar o cooler, display, leds entre outros. São elas:

- **pwmInit()**
- **ssdInit()**
- **lcdInit()**
- **kpInit()**

Além da função responsável para a detecção de uma tecla:

kpDebouce(): Função responsável por prevenir erros, permite a execução de uma função apenas se um determinado tempo se passou.

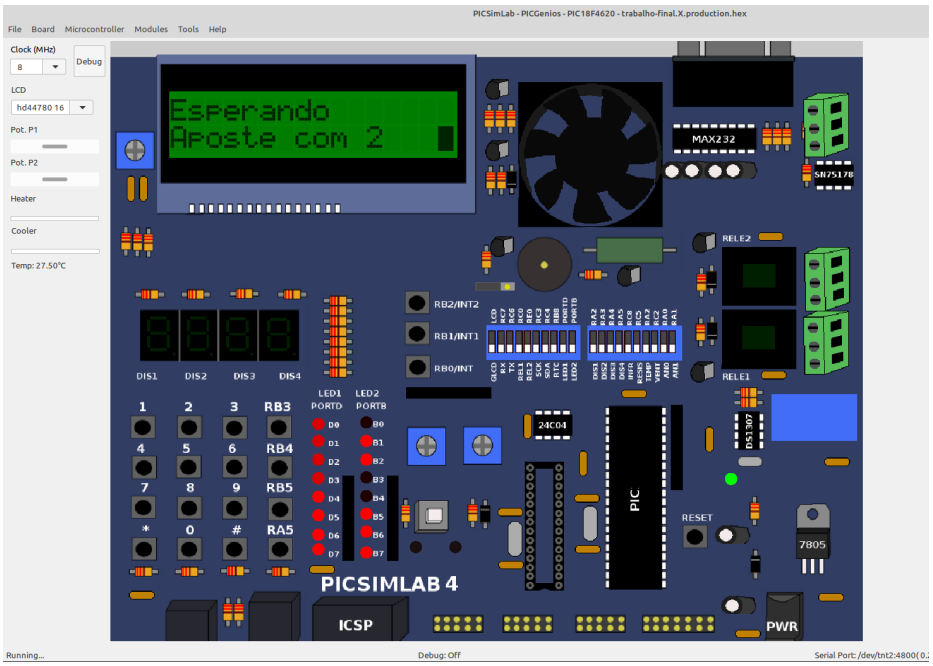
kpRead(): Responsável por ler uma tecla.

bitTst(teclaLida, bit): Responsável por verificar se determinada tecla foi pressionada.

E uma função extra:

textWait(): Responsável por apresentar uma mensagem aguardando a escolha do usuário, fornecendo uma sugestão.

Figura 6: Tela de espera



Fonte: Acervo do autor

Para o desenvolvimento de todas as funções foi criada uma biblioteca, chamada de apostas.cpp, que inclui todas as funcionalidades da aplicação, permitindo maior manutenção e legibilidade do código a longo prazo.

Conclusão

A execução deste projeto proporcionou uma experiência prática e concisa dos elementos apresentados ao decorrer da disciplina, apesar da dificuldade ocasionada pelo distanciamento social e as medidas sanitárias, com o cessar de aulas presenciais, o uso de simuladores e de conteúdo audiovisual conseguiram ajudar a superar esse desafio.

O projeto ainda proporcionou experiências com o audiovisual, para as produções de vídeos explicando o decorrer do projeto, aliado a ferramentas de versionamento, como o git e de versionamento em nuvem, como o GitHub.