

UNIVERSIDADE FEDERAL DE ITAJUBÁ



JOGO DA VELHA

ECOP04 – PROGRAMAÇÃO EMBARCADA

ECOP14 – LABORATÓRIO DE PROGRAMAÇÃO EMBARCADA

Pierre Willian Graciano da Silva – 2020002534

Objetivo: Este relatório tem como objetivo descrever e explicar o funcionamento do código e simulação no PicsimLab. Para isso, foi usado o microcontrolador PIC18F4520 e a placa PICGenios. Este relatório junto do código conta com a nota parcial da P2 das disciplinas Programação Embarcada e Laboratório de Programação Embarcada.

Palavras-chave: PicsimLab, PICGenios, PIC18F4520, Programação Embarcada, Jogo da velha.

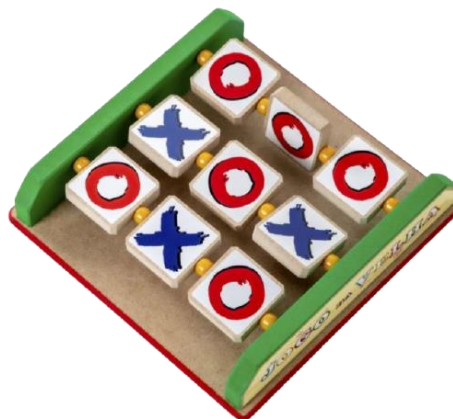
Introdução

Para começar falando de programação embarcada ou sistemas embarcados, é necessário citar sua definição que é um sistema adaptado para uma placa ou microprocessador de um aparelho, como por exemplo eletrodomésticos e máquinas.

Nas disciplinas ECOP04 e ECOP14 foram retratadas simulações diante do aplicativo PicsimLab, devido às aulas estarem sendo ofertadas remotamente. O código produzido em C era feito e compilado no software MPLAB X IDE, que gerava o arquivo .hex para ser simulado ou gravado na placa.

O simulador PicsimLab conta com diversos processadores e placas diferentes, além disso conta com periféricos como teclado matricial, LEDs, cooler, relés, display de 7 segmentos, display LCD, buzzer e potenciômetro. Por outro lado, no MPLAB, foi disponibilizadas diversas bibliotecas, entre elas “ssd.h”, “bits.h”, “keypad.h” etc. Cada uma com uma função específica no código, como por exemplo a keypad.c e keypad.h, eram responsáveis por habilitar funções relacionadas ao funcionamento do teclado, já a ssd.c e ssd.h eram relacionadas ao display de 7 segmentos.

Com isso, foi solicitado aos alunos que fizessem um produto usando o PICGenios e o PIC18F4520, e utilizando periféricos da placa. Nesse trabalho foi optado por fazer um jogo da velha, clássico Tic Tac Toe, e foram utilizados os componentes Teclado Matricial, Cooler, Relé, LCD e LEDs.



Desenvolvimento

O trabalho em questão de refere ao jogo da velha, simulado no PicsimLab e codificado no MPLAB, assim o usuário escolhe a posição desejada, como por exemplo linha um e coluna 2, logo irá ter que apertar o botão na mesma posição no teclado matricial, ou seja, botão 2.

A tela inicial do jogo é composta por um desenho de uma bruxa, que passa do lado direito para o lado esquerdo da tela. O desenho foi feito através do site <https://maxpromer.github.io/LCD-Character-Creator/> na qual é possível fazer o “desenho” e o site dá em formato de bit. Para isso, fiz uma junção de 6 vetores, criando um retângulo de 3x2. Cada vetor foi unido para antes ter o desenho apresentado no display.

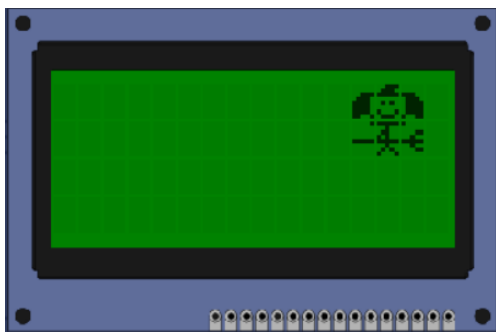


Figura 1 – Tela inicial

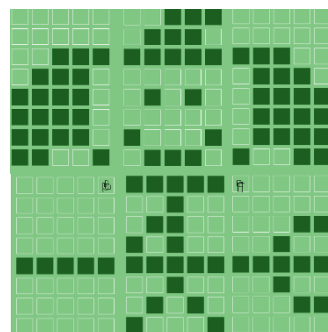


Figura 2 – Formação do desenho

Logo após o aparecimento da imagem, no display são apresentadas informações sobre o jogo e sobre o desenvolvedor. É necessário que o usuário aperte # para prosseguir para as instruções. Nas instruções é informado aos jogadores a função dos botões, que servem para posicionar X ou O.

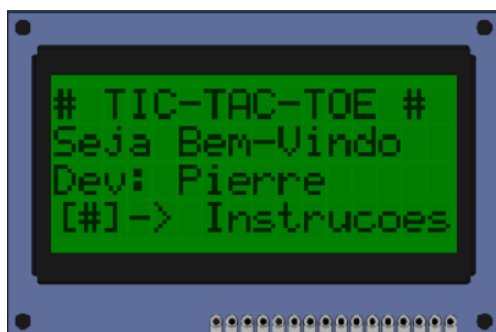


Figura 3 – Menu inicial

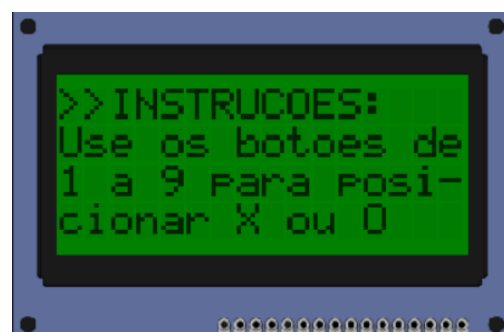


Figura 4 – Instruções

Para o usuário posicionar X ou O, como já explicado no início deste tópico, é necessário escolher o botão referente à posição desejada (caso o botão seja pressionado duas vezes ele irá mudar de X par O e vice versa), como explicado na imagem:



Figura 5 – Display com posições



Figura 6 – Teclado Matricial

O fim do jogo é representado por pelo acionamento dos relés ao lado do cooler e pelo acionamento do cooler, que girará por alguns segundos. Após o fim do código, basta o usuário apertar o botão RESET para reiniciar o código.

- **Bibliotecas e funções**

Começando pelas bibliotecas utilizadas no código, elas foram fornecidas pelo professor Otávio, exceto delay.h e lcd_Position.h, feitas por mim.

"bits.h" – Função do bitSet;

"lcd.h" – Usada para uso do display LCD;

"pwm.h" – PWM usado para uso do cooler;

"delay.h" – Feita para criar contagens de tempo;

"keypad.h" – Usada para uso do teclado;

"io.h" – Usada para entrada e saída;

"lcd_Position.h" – Define posição usando linha e coluna;

<pic18f4520.h> - Própria biblioteca do microcontrolador.

As funções desenvolvidas no código foram:

void simbolo (void) – Feita para impressão da imagem da velha;

void menu_inicial (void) – Feita para impressão do menu inicial;

void instruções (void) – Feita para impressão das instruções;

void inicio_do_jogo (void) – Feita para impressão das posições e algumas funções;

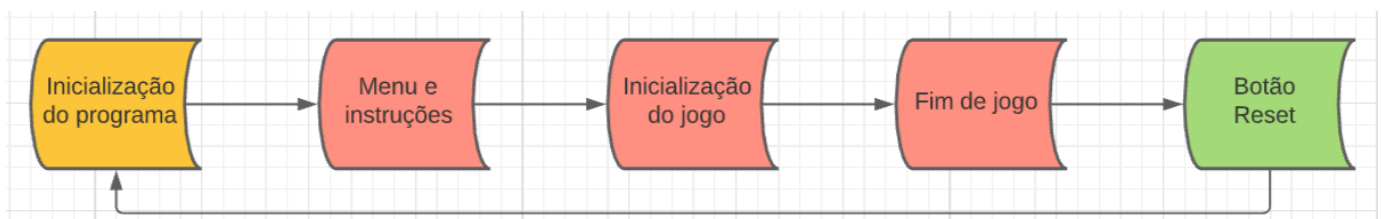
void atraso_ms (void) – Delay em milissegundos;

void lcdPosition (line,column) – Posiciona caractere de acordo com a coordenada;

void cooler (int v) – Feita para configuração da ventoinha;

void fim_de_jogo (void) – Feita para acionamento da ventoinha, LEDs e relés.

Fluxograma



O programa inicia imprimindo uma imagem de uma bruxa simbolizando o jogo da velha, seguida da impressão do menu e das impressões, etapas já esclarecidas no relatório.

Em seguida há o desenvolvimento da função principal. Nela ocorre primeiramente a declaração de variáveis e declaração de bits e pinagens. Em seu loop infinito, é implementado leitura do teclado para posicionar o X ou O utilizando a função `lcdPosition`. Após todas as posições estarem preenchidas com X ou O, o programa vai para a função `fim_de_jogo`. A função final é lida no fim do código, e apresenta os seguintes comandos.

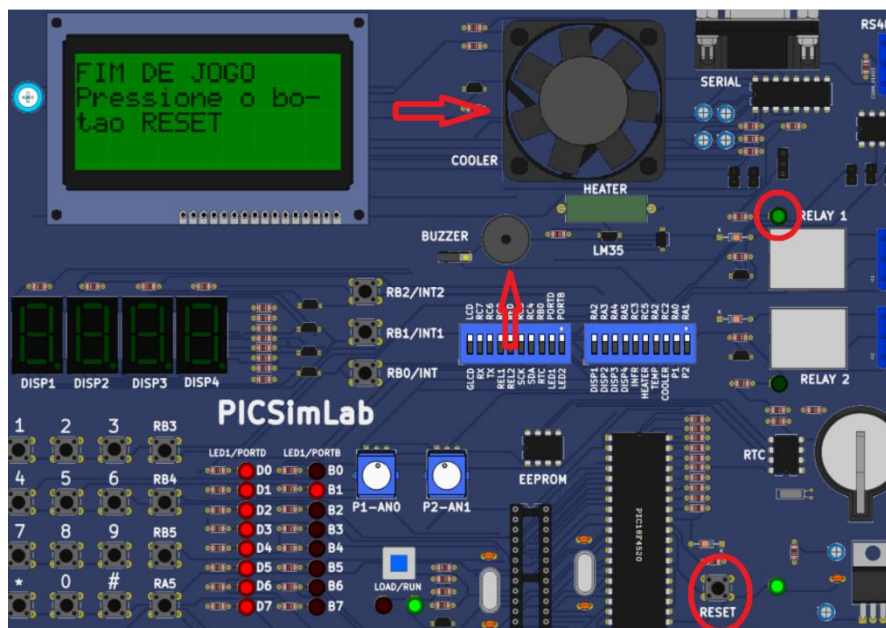


Figura 7 –
Simulação da placa
rodando a função
`fim_de_jogo`

Nessa função ocorre o acionamento da ventoinha para resfriamento da placa para evitar o superaquecimento, também são ligados LEDs e o relé que confirmam que o programa foi finalizado.

No display LCD terá a mensagem “Pressione o botão RESET”. Esse comando é necessário para que o usuário reinicie o código, e assim consiga jogar o jogo da velha novamente.

Conclusão

O projeto e sua ideia em si contribuem para que os alunos das disciplinas ECOP04 e ECOP14 se tornem melhores programadores e com uma visão mais ampla do que é a programação de sistemas embarcados, que é de grande importância nos dias de hoje.

Sobre as dificuldades, pôde-se observar que, durante o uso do teclado matricial, ocorreram diversos erros na impressão dos símbolos X e O. Também houve dificuldade em achar um algoritmo que reconhecesse a posição adequada de cada símbolo para que assim reconhecesse o vencedor. Além disso, também houve dificuldade em poupar linhas e funções durante o código, ficando um código muito repetitivo.