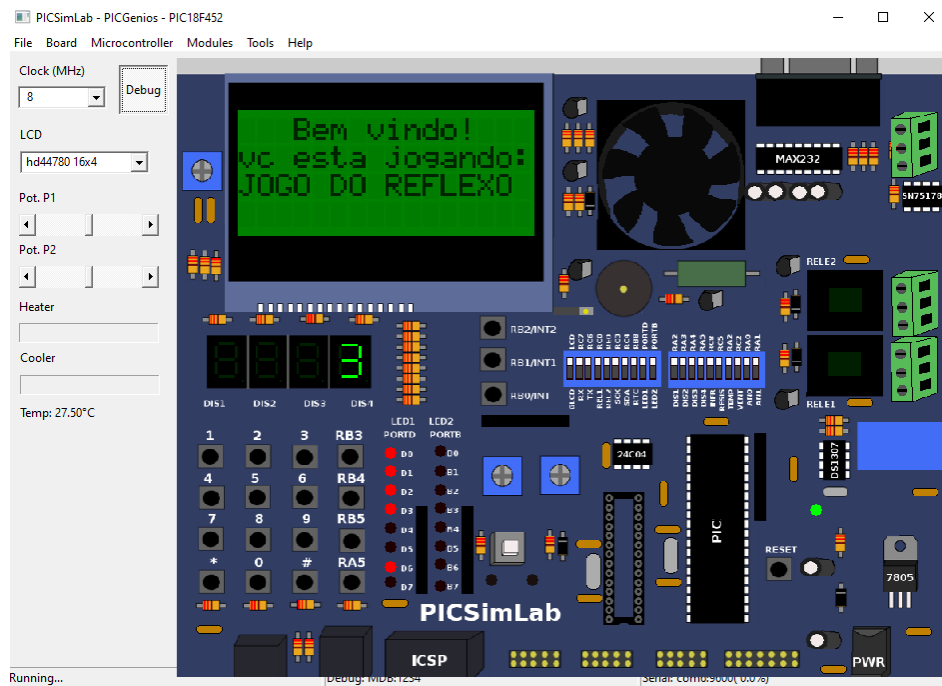


Jogo do Reflexo



André Casadei Marques – 2019010640

Laboratório de programação embarcada

ECOP14 – T03

Itajubá

2020

Introdução

Como projeto final da disciplina de Laboratório de Programação Embarcada - ECOP14 foi proposto aos discentes que desenvolvessem um programa utilizando o simulador PICSimLab de Luis Claudio Gambôa Lopes que simula uma placa didática da PICGenios utilizando o Microcontrolador PIC18F4520 da Microchip. Neste documento estará disponível uma breve explicação dos códigos e bibliotecas utilizadas, bem como a descrição do objetivo do jogo e as dificuldades e soluções encontradas durante o desenvolvimento do projeto.

Objetivos

- Aplicar os conhecimentos da aula teórica de programação embarcada, utilizando o simulador PICSimLab na linguagem C.
- Criar uma aplicação que utiliza os periféricos, interfaces e barramentos da placa.
- Explicar o código e o processo de criação do projeto mostrando as dificuldades e as suas respectivas soluções.

Ressalvas

Outra parte da proposta é a publicação de um vídeo mostrando o funcionamento do código criado, a publicação no LinkedIn e a publicação no GitHub com os arquivos do projeto.

Links:

https://www.linkedin.com/posts/andr%C3%A9-marques-9b4795200_como-projeto-final-da-disciplina-pr%C3%A1tica-activity-6742562905180532736-3ffW

https://github.com/AndreCasadeiMarques/Jogo-do-Reflexo_ProjetoFinalECOP14

Objetivo do jogo

O objetivo do jogo é medir o reflexo do usuário. O programa fará com que os LEDs pisquem na placa e o jogador tem 30 tentativas disponíveis (exibidas no Display de 7 segmentos) para marcar o maior número de pontos, pressionando a tecla correspondente no teclado após cada acendimento de LED. No final de cada partida, é exibido no LCD a pontuação total.

Interfaces utilizadas

LEDS DO PORTD:

Acende um LED por vez e apaga rapidamente, exigindo um bom reflexo do jogador. Eles fornecem ao usuário qual tecla deve ser acionada;

TECLADO:

A partir das teclas, o jogador indica o número do LED que piscou.

DISPLAYS DE 7 SEGMENTOS:

No display de 7 segmentos será a mostrado o número de cada ciclo (serão 60 no total com a duração de aproximadamente 1 segundo cada).

LCD 16x4:

O LCD será a interface explicativa para o usuário. Assim, responsável por informar a pontuação do jogador no final de cada partida.

Explicação do código

Bibliotecas utilizadas: “stdio.h”, “stdlib.h”, “math.h”, “time.h”, “config.h”, “delay.h”, “lcd.h”, “pic18f4520.h”, “teclado.h”.

Contagem no display de 7 segmentos



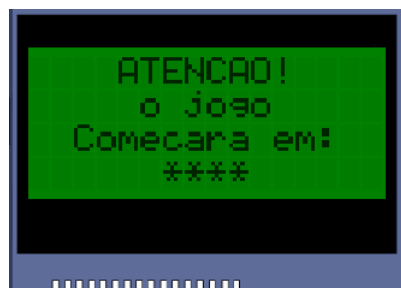
```
for (aux = 0; aux < 5; aux++) { //exibe os segundos no display de 7 segmentos
    PORTD = values7seg2[cont2];
    BitSet(PORTA, 5);
    atraso_ms(10);
    BitClr(PORTA, 5);

    PORTD = values7seg2[cont1];
    BitSet(PORTA, 4);
    atraso_ms(10);
    BitClr(PORTA, 4);
}
```

Esse código é responsável pela contagem no display de 7-seg, as primeiras quatro linhas dentro do for é responsável por mostrar o número no Display “DIS3” e as outras quatro no Display “DIS4”, os valores no vetor “value7seg2” são os valores em hexadecimal de 0 a 5 para o primeiro dígito e de 0 a 9 para o segundo dígito.

Como o intervalo desejado era de aproximadamente 1,5 segundos por contagem as funções de “atraso_ms()” permitem que seja mostrado o primeiro dígito por 10 ms e o segundo por 10 ms, fazendo isso 5 vezes no “for” da variável “aux”.

Comunicação com o usuário através do LCD



```

54 void print1(){
55     lcd_init(); //Config LCD
56     lcd_cmd(L_CLR);
57     lcd_cmd(L_L2);
58     lcd_str(" Loading...");
59     atraso_ms(1000);
60
61     lcd_cmd(L_CLR);
62     lcd_cmd(L_L1);
63     lcd_str(" Bem vindo!");
64     lcd_cmd(L_L2);
65     lcd_str("vc esta jogando:");
66     lcd_cmd(L_L3);
67     lcd_str("JOGO DO REFLEXO");
68     atraso_ms(2200);
69
70     lcd_cmd(L_CLR);
71     lcd_cmd(L_L1);
72     lcd_str(" Pressione");
73
74     lcd_cmd(L_L2);
75     lcd_str(" a tecla");
76     lcd_cmd(L_L3);
77     lcd_str(" correspondente");
78     lcd_cmd(L_L4);
79     lcd_str(" ao LED aceso");
80     atraso_ms(2200);
81
82     lcd_cmd(L_CLR);
83     lcd_cmd(L_L1);
84     lcd_str(" ATENCAO!");
85     lcd_cmd(L_L2);
86     lcd_str(" o jogo");
87     lcd_cmd(L_L3);
88     lcd_str(" Comecara em:");
89     lcd_cmd(L_L4);
90     lcd_str(" *****");
91     atraso_ms(2000);

```

Esse é o código responsável por exibir as mensagens no display LCD, as funções no LCD foram disponibilizadas pelo professor, com exceção da função “atraso_ms” que foi criada durante uma aula, que basicamente são três estruturas de repetição, que se aproximam de 1 milissegundo, o parâmetro de entrada é a quantidade desejada em ms.

Interação com o teclado matricial e LEDs



```

182     valorLed = rand() % 7 + 1;
183     TRISB = 0x00; //config da porta B como Saída
184     PORTB = 0b00000000; //apaga os LEDs no PORTB
185     atraso_ms(100); //tempo que os LEDs ficam apagados
186     PORTB = valuesLED[valorLed]; //porta B recebe valor do LED aleatório
187     atraso_ms(300); //tempo que o LED fica aceso
188     PORTB = 0b00000000; //apaga os LEDs no PORTB
189     TRISB = 0xF8; //config a porta B como entrada
190     TRISD = 0x0F; //config a porta D como entrada
191
192     valorTeclado = leTeclado(); //Lê o teclado
193
194     if (7 - valorTeclado == valorLed) { //compara o valor recebido do teclado
195         if (pontuacaoUn < '9'){
196             pontuacaoUn++; //incrementa o contador de pontos da unidade
197         }
198         else{
199             pontuacaoUn = '0';
200             pontuacaoDz++;
201         }
202     }
203     TRISD = 0x00; //config porta D como Saída

```

Na primeira linha a variável de controle recebe um valor aleatório de 1 a 7 que será responsável por controlar o espaço do vetor “ValuesLED” a ser acessado, a partir disso os valores em hexadecimal são atribuídos ao PORTB ligando o LED correspondente.

Depois o PORTB é zerado e é configurado o PORTD para entrada no objetivo de atribuir na variável o “valorTeclado” obtida à partir da função de leitura do teclado

Finalmente os valores são comparados e caso sejam iguais a pontuação é incrementada (de 1 a 9 para unidade e depois caso seja >9 a dezena é incrementada para exibir no a pontuação final do jogo). Observação: a comparação da tecla com o valor do LED está na sintaxe “7 - valorTeclado == valorLed”, o valor de 7 é porque os números para leitura do teclado estão em ordem contrária a dos LEDs, ou seja o mais significativo vira o menos significativo apenas na comparação.

Função de leitura do teclado:

```
95 unsigned char leTeclado(){
96     int aux4;
97     unsigned char aux3, valorT = 255;
98     for (aux4 = 0; aux4 < 80; aux4++){
99         aux3 = tc_tecla(); //lê o teclado
100         if(aux3!=255){
101             valorT = aux3;
102         }
103     }
104     return valorT;
105 }
```

A leitura do teclado é feita de forma matricial na placa. No nosso código, é verificado o valor do PORTD para retornar o valor na função “BitTst”, tanto a função leTeclado como “BitTst” foram fornecidas pelo professor Otávio, fizemos apenas algumas alterações.