

Universidade Federal de Itajubá



Máquina de Sorvete

ECOP04 – Programação Embarcada

ECOP14 – Laboratório de Programação Embarcada

Charbel Daher Boulos

2020025270

1. Introdução

- **1.1 - Objetivo**

Esse projeto tem como o objetivo desenvolver todas as técnicas de sistemas embarcados estudadas ao longo do período de 2021 em uma única proposta que foi produzida no MPLAB IDE e reproduzida no simulador do PicSimLab.

- **1.2 - Contextualização**

No primeiro semestre de 2021 foi proposto um projeto para as disciplinas de ECOP04 e ECOP14, ministradas pelo discente Otávio de Souza Martins Gomes. As aulas trabalhavam com o MPLAB IDE para desenvolver os códigos e com o PicSimLab para reproduzir e simular o que seria feito na placa do PICGenios. Além disso, os códigos eram feitos na linguagem C e reproduzidos no micro controlador PIC18F4520.



Figura 1 – MPLAB IDE

Ao longo do semestre foi demonstrado o funcionamento de diversas aplicações da placa, entre eles: LCD, LED's, Display de 7 segmentos, Teclado matricial, potenciômetro, ventoinha (cooler), entre outros.

2. Desenvolvimento

- **2.1 – Moldes base**

Este projeto simula o funcionamento de uma máquina de sorvete de casquinha como as das figuras abaixo:



Figura 2 e 3 – Máquinas de sorvete

• 2.2 – Funcionamento

O programa tem início com uma mensagem de boas vindas e aguarda o usuário pressionar o botão “#” para escolher o sabor da massa de sorvete. Após efetuar essa ação, o menu de opções ficará disponível enquanto os botões “1”, “2” ou “3” não forem pressionados.

Quando o usuário escolher seu sabor e aparecerá seu respectivo valor no display de 7-segmentos e então, ele poderá escolher quão cremoso será a massa e, para isso, irá utilizar o potenciômetro que varia de 0% a 5% de cremosidade. Porém, esse percentual irá afetar no tempo de produção de sorvete, caso esse número esteja entre 0 e 1,9, será impresso a seguinte mensagem no LCD “Aguarde Rapidamente estará pronto”, o relé 1 irá piscar, indicando que o processo irá começar e os LED’s começarão a ascender de maneira sequencial ligeiramente rápido. Caso a cremosidade esteja entre 2% e 5%, o LCD terá a seguinte mensagem “Aguarde Estará pronto em instantes...”, o relé 1 irá piscar com a mesma função de indicar que irá começar e os LED’s se ascenderão também de maneira sequencial, no entanto, mais lentamente.

Junto com os LED’s, a ventoinha também entra em funcionamento após a escolha do sabor e da cremosidade. O cooler, neste caso, tem duas velocidades, se for escolhido um percentual até 1,9, ela irá girar com uma velocidade média, pois na simulação exige menos da máquina. Já para a situação a qual a cremosidade seja maior ou igual a 2, a ventoinha terá uma velocidade de 80% do seu total, uma vez que, quanto mais cremoso, maior o rendimento da máquina.

Ao fim deste processo, o relé 2 irá piscar indicando o fim e por último, o LCD imprime a seguinte mensagem “Volte sempre” seguido de um desenho de uma casquinha de sorvete e uma mensagem para apertar o botão de “reset” para recomeçar.

• 2.3 – Fluxograma do funcionamento

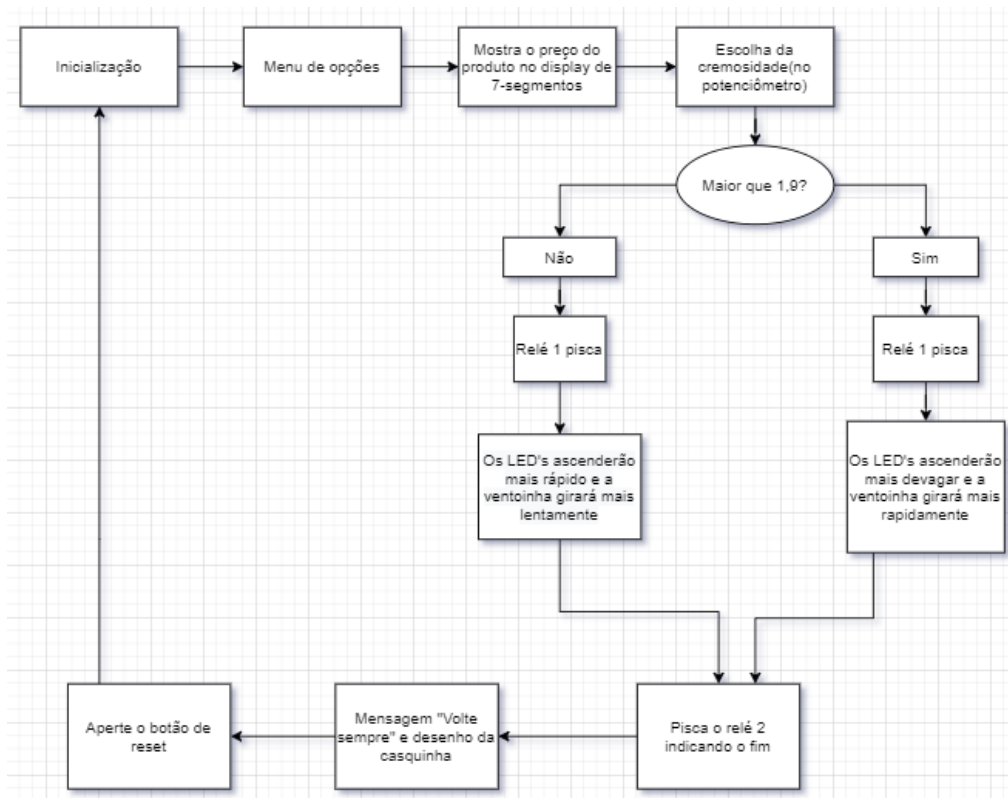


Figura 10 – Fluxograma do sistema

• 2.4 – Exemplo da simulação

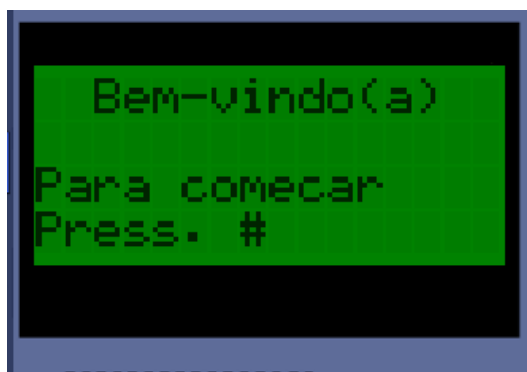


Figura 4 – Mensagem de início

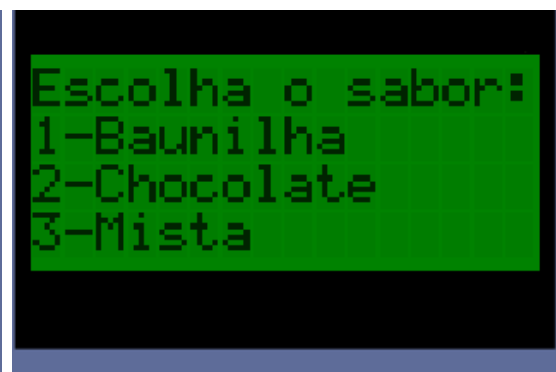


Figura 5 – Menu de opções



Figura 6 – Display de 7-seg mostrando o preço do sorvete

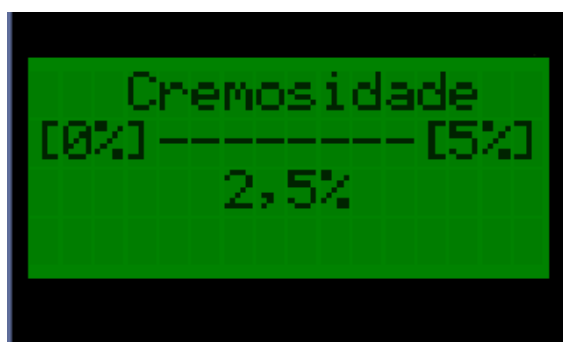


Figura 7 – Cremosidade do sorvete que é escolhida pelo potenciômetro

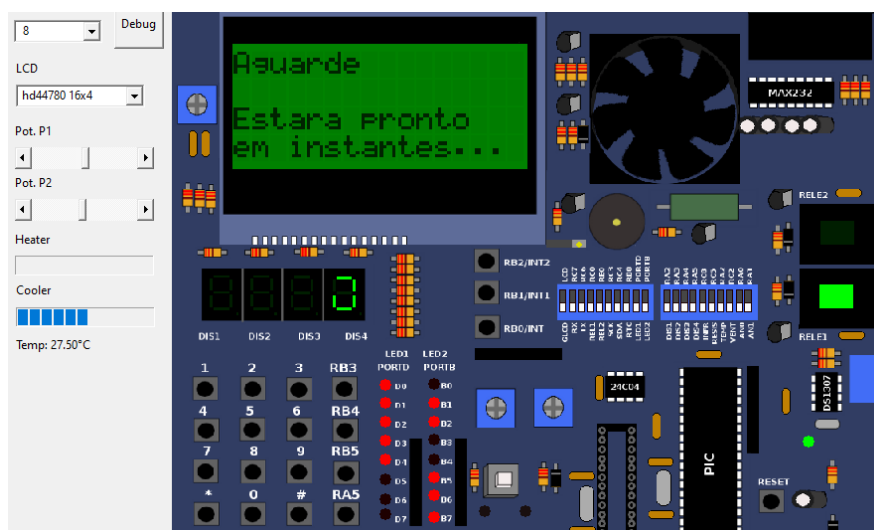


Figura 8 – Sorvete em andamento – Enquanto os LED's piscam ordenadamente, a ventoinha é ligada e o relé 1 também pisca



Figura 9 – Mensagem final para recomençar o processo

• 2.5 – Bibliotecas e funções

As seguintes bibliotecas foram utilizadas ao longo do programa:

- *pic18f4520.h*

Contém definições das portas utilizadas no simulador da placa.

- *config.h*

Contém definições do micro controlador PIC18F4520.

- *bits.h*

Contém definições para operações de bits.

- *lcd.h*

Contém funções para manipulação do LCD, dentre elas:

- void lcdInit(void) : inicializa o lcd;
- void lcdCommand(unsigned char cmd) : passa um comando para impressão;
- void lcdData(unsigned char valor) : imprime apenas um caractere;
- void lcd_str(const char* str) : imprime uma string;

- *keypad.h*

Contém funções para operar com o teclado matricial, dentre elas:

- void kpInit(void) : inicializa o teclado;
- unsigned int kpRead(void) : retorna a tecla pressionada;

- void kpDebounce(void) : faz a leitura das teclas e evita o problema de bouncing;

- *ssd.h*

Contém funções para controlar o display de 7 segmentos, dentre elas:

- void ssdInit(void) : inicializa os terminais de dados e de controle;
- void ssdDigit(char val, char pos) : recebe um número entre 0 e F e mostra ele na posição passada entre 0 e 3;
- void ssdUpdate(void) : desliga o número atual, atualiza e liga o próximo;

- *PF.h*

Contém funções diversas que auxiliaram no programa, dentre elas:

- void lcdPosition(unsigned char, unsigned char) : define a linha e coluna que a mensagem será impressa no lcd;
- void tempo(unsigned char x) : produz um atraso por meio de loop
- void printf(char v[]) : imprime o menu de opções no lcd;
- void clear(int, int, int, int) : limpa a tela lcd da linha x até a xF e da coluna y até a yF;
- void LigarLed(char, char) : liga sequencialmente os LED's;

- *adc.h*

Contém algumas funções para usar o potenciômetro, dentre elas:

- void adc_init(void) - não foi utilizada;
- unsigned int adc_amostra(unsigned char canal) : retorna o valor em relação a posição que o potenciômetro se encontra;

- *itoa.h*

Contém apenas uma função que transforma o valor retornado da função adc_amostra em um caractere.

- void itoa(unsigned int val, char* str);

- *pwm.h*

Contém funções que tem acesso às saídas PWM, dentre elas:

- void pwmInit(void) : inicializa as saídas PWM;
- void pwmSet(unsigned char por cento) : faz a ventoinha girar com uma velocidade relacionada com “por cento” que é passada como parâmetro;
- void pwmFrequency(unsigned int freq) - não foi utilizada;

3. Conclusão

O intuito deste projeto era desenvolver os conhecimentos passados ao longo do semestre sobre sistemas embarcados e ao fim deste projeto, foi possível compreender um pouco melhor o funcionamento da placa PICGenios, de seus componentes e do micro controlador PIC18F4520.

No entanto, ao longo de todo o processo de criação, diversas dúvidas foram surgindo que foram sanadas em aulas de dúvidas, assim, percebe-se o grande aprendizado sobre sistemas embarcados que este projeto gerou.