UNIVERSIDADE FEDERAL DE ITAJUBÁ

ENGENHARIA DE COMPUTAÇÃO

VINÍCIUS MONTEGLIONE DE OLIVEIRA (2019004714)

SIMULAÇÃO DE CAIXA ELETRÔNICO

Simulação de um caixa eletrônico utilizando o microcontrolador PIC18F4520 no kit PICGenios.

Itajubá, MG

2020

1. INTRODUÇÃO

A partir da necessidade da criação de um projeto utilizando-se do microprocessador PIC18F4520 e dos conceitos adquiridos na disciplina ECOP04 – Programação Embarcada e ECOP14 – Laboratório de Programação Embarcada, eu e o aluno Luis Felipe Buzo decidimos por desenvolver um simulador de caixa eletrônico. A inspiração surgiu pela semelhança da disposição do kit PICGenios com a de um caixa eletrônico.

Figura 1: Caixa eletrônico à esquerda e parte do kit PICGenios à direita.

Antes da realização do projeto, estabelecemos uma visão geral da proposta do projeto, ou seja, o que ele executará e qual seria a função de cada componente envolvido. Suas funções seriam: acessar a conta do usuário, verificar o saldo da conta e realizar saques e depósitos. Além disso, para uma maior imersão, definimos que o saldo seria diferente para cada conta acessada. Além disso, o saldo variaria conforme as operações de saque e depósito fossem executadas pelo usuário. Para os componentes da placa, atribuímos as seguintes funções:

- LCD: responsável pela maior parte da comunicação visual e exibirá o menu do caixa e as informações relativas aos processos, como por exemplo o saldo da conta, o número da conta, as teclas referentes a cada função, etc.
- DISPLAY 7-SEG: utilizado para mostrar o valor final do saldo após as operações.

- TECLADO: utilizadas para navegar no menu do caixa, digitar o número da conta, selecionar quantias em dinheiro e confirmar operações.
- LEDs: Serão utilizados como uma “animação de carregamento”, que irá acender de cima parar baixo e de baixo para cima enquanto a conta do usuário é acessada.

Tendo definido o funcionamento do projeto, podemos partir para a programação. Todo o projeto foi desenvolvido utilizando-se do MPLAB X IDE, na linguagem de programação C e com a utilização das bibliotecas desenvolvidas pelo professor Rodrigo Almeida e fornecidas para a disciplina, além do simulador Picsimlab.

2. DESENVOLVIMENTO

Primeiramente, antes de se realizar o código, foram desenvolvidas as etapas da interação entre o usuário e o simulador. Para tal, desenvolveu-se a seguinte interação:

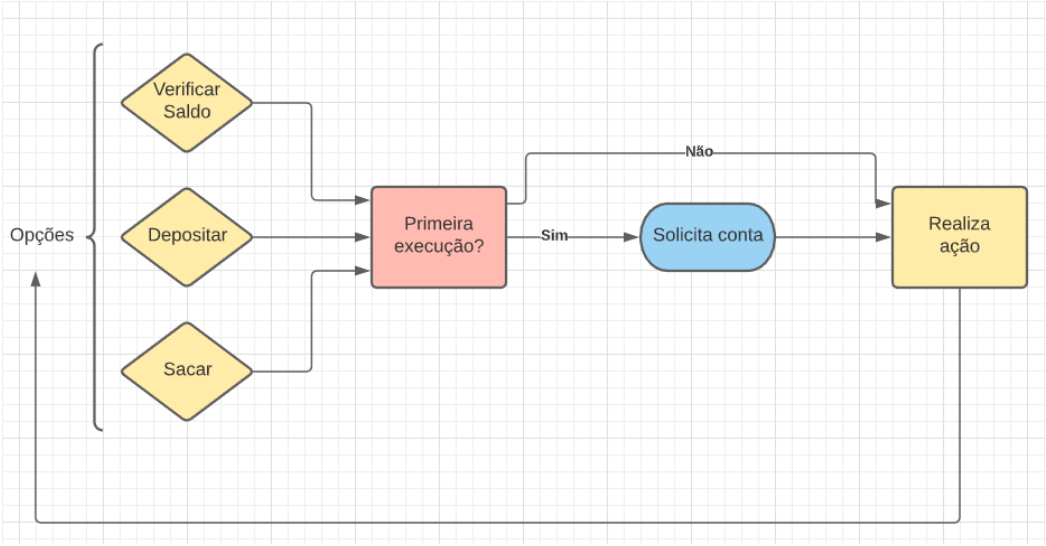

Figura 2: Diagrama da interação entre usuário e simulador.

Assim, temos que o programa terá o seguinte loop: usuário escolhe a ação; é verificada se é a primeira execução – se sim, solicita o número da conta de 4 digitos para usuário, se não, realiza a opção escolhida diretamente. Em seguida, após ter a ação realizada, o programa volta para a etapa inicial, solicitando que

usuário selecione uma das opções. Os menus serão dispostos da seguinte forma:

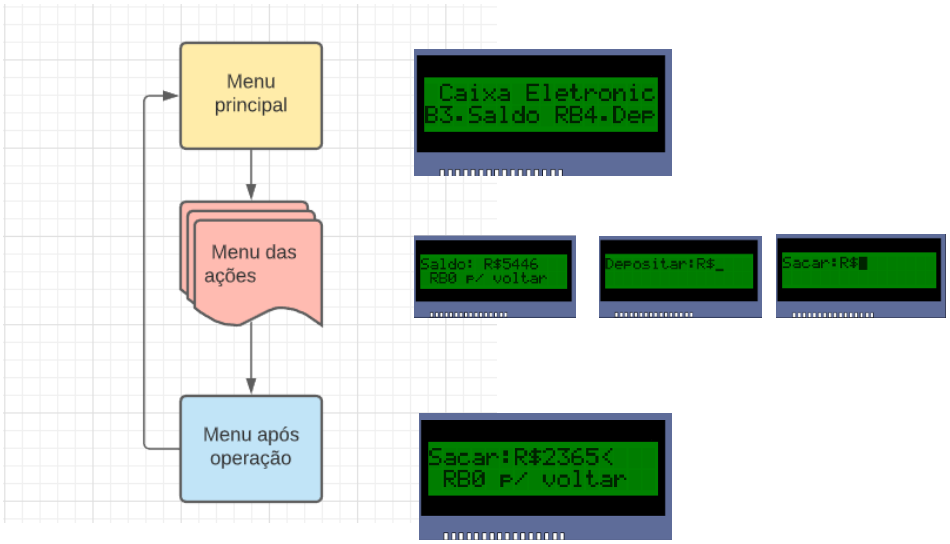

Figura 3: Ordem dos menus.

Entrando em detalhes no código, é importante destacar que as bibliotecas utilizadas foram as seguintes:

```
#include "config.h"
#include "pic18f4520.h"
#include "delay.h"
#include "lcd.h"
#include "teclado.h"
#include "caixaeletronica.h"

```

Figura 4: Bibliotecas utilizadas.

Delas, a “caixaeletrica.h” foi desenvolvida por nós, enquanto as outras foram fornecidas pelo professor, como citado anteriormente. Para verificar se é a primeira execução do usuário criamos uma “flag” que, se é a primeira execução tem valor 0, caso contrário valor 1. No geral, as variáveis utilizadas foram:

```
    struct valor saldo; //sera praticamente uma variavel gloabal que sera alterada conforme usuario sacar ou depositar
    struct valor dep; //armazena quanto usuario quer depositar
    struct valor sac; //armazena quanto usuario quer sacar
    struct valor conta; //armazena numero da conta do usuario
    unsigned char est; //armazena qual tecla foi pressionada (RB3, RB4 ou RB5)
    int flag = 0; //para ver se conta ja foi inserida ou nao dentro do loop infinito

```

Figura 5: Variáveis utilizadas e seus propósitos.

Para cada opção no menu principal, atribuímos uma tecla, sendo: RB3 para saldo, RB4 para depósito e RB5 para saque. No LCD, há a exibição das opções se deslocando da direita para a esquerda que permanecem em loop até que uma das teclas seja pressionada. Para a verificação desta condição, utilizamos a cláusula *if* juntamente à função *BitTst()*, que retornará se a tecla foi

pressionada. Logo em seguida, há uma cláusula *if* que verifica a condição da “flag”.

```
41      if (!BitTst(PORTB, 3) || !BitTst(PORTB, 4) || !BitTst(PORTB, 5)) { //se alguma RB foi pressionada
42          est = PORTB; //armazena qual das 3 foi pressionada
43
44          if (flag == 0) { //pede a conta se a pessoa esta na primeira execucao
45              TRISB = 0xF8; //teclado numerico
46
47              lcd_cmd(L_CLR);
48              lcd_cmd(L_L1);
49              lcd_str("Conta: ");
50
51              saldo = leconta(0); //usuario tecla numero de sua conta e isso gera um valor que será armazenado em saldo
52              for(int h = 0; h < 4; h++){ //reverte saldo para conta e armazena na variavel conta
53                  conta.x[h] = saldo.x[h] - h;
54              }
55
56              lcd_cmd(L_L2);
57              lcd_str("Aguarde...");
58              piscaLed(); //pisca leds em sequencia mostrando um "carregamento"
59
60          }
61
```

Em seguida, verifica-se qual das três teclas foi pressionada também 3 cláusulas *if* e a função *BitTst()* e , ao final, atualiza a *flag*. Um exemplo dessa verificação é o seguinte:

```
111         if (!BitTst(est, 5)) { //pede quanto o usuario quer sacar e subtrai esse valor em saldo
112             TRISB = 0xF8; //teclado numerico
113
114             lcd_cmd(L_CLR);
115             lcd_cmd(L_L1);
116             lcd_str("Sacar:R$");
117             sac = leconta(1); //le quantia a sacar
118             saldo = subtrai(saldo, sac); //subtrai quantia no saldo
119
120             TRISA = 0x20; //configuracao teclas extras para receber RB0
121             TRISB = 0x3F;
122
123             lcd_cmd(L_L2);
124             lcd_str(" RB0 p/ voltar         ");
125             exibeSeg(saldo); //enquanto usuario nao clica em RB0 mostra o saldo no display de 7 segmentos
126
127             BitClr(PORTA, 5); //para limpar leds de 7 segmento
128             BitClr(PORTA, 4);
129             BitClr(PORTA, 3);
130             BitClr(PORTA, 2);
131
132             lcd_cmd(L_CLR); //limpar
133             PORTB = 0xFF;
134         }
135         flag = 1; //flag nao sera mais zero entao nao vai mais pedir para colocar numero da conta na execucao
```

Por fim, desenvolvemos uma biblioteca “caixaeletronica.h” com funções para que o código ficasse mais “limpo”. Ela possui as seguintes funções: *piscaLed()*, que realiza o acendimento sequencial dos LEDs como uma animação de carregamento; *exibeSeg()* que exibe o saldo no display de 7 segmentos; *leconta()*, que lê 4 digitos do teclado e retorna a struct valor ou gera conta dependendo do parâmetro; *soma(),* que realiza a soma do saldo com o depósito; *subtrai()*, que subtrai o valor sacado do saldo. As três últimas funções foram as mais difíceis de implementar, uma vez que realizamos o armazenamento das quantias em um vetor. Assim, foi necessário realizar somas e subtrações de vetores, que ocorre de posição em posição, com a necessidade de verificar os possíveis *carry in* ou *carry out*.