

UNIVERSIDADE FEDERAL DE ITAJUBÁ – UNIFEI
CAMPUS ITAJUBÁ

Isaac do Prado Almeida

**Pet Virtual para PICGenios com
PIC18F4520 utilizando o Simulador
PicSimLab**

ITAJUBÁ - MG

2021

Isaac do Prado Almeida

Pet Virtual para PICGenios com PIC18F4520 utilizando o Simulador PicSimLab

Documento técnico realizado para as disciplinas de Programação Embarcada e Laboratório de Programação Embarcada, apresentada como requisito de avaliação da Universidade Federal de Itajubá.

Professor: Otávio de Souza Martins Gomes

ITAJUBÁ – MG

2021

SUMÁRIO

RESUMO	4
1. PLANEJAMENTO	4
2. ESTRUTURA DO PROJETO	5
3. DESENHANDO NO LCD	6
4. DIFICULDADES E SOLUÇÕES	7
5. FUNCIONAMENTO DO PET	8

RESUMO

O projeto consiste em um Pet Virtual, ao estilo Tamagotchi, onde o usuário tem que cuidar de um Pet (dando comida, atenção, entre outras funções). Neste projeto foi utilizado os softwares MPLAB X IDE v5.40, com o compilador XC8 v2.20 e o simulador do microcontrolador PICGenios, o Picsimlab utilizando o PIC18F4520. O projeto será utilizado como 60% da nota da disciplina de Programação Embarcada (ECOP04 e ECOP14) da Universidade Federal de Itajubá (UNIFEI), utilizando a linguagem C++.

Todas bibliotecas utilizadas no projeto estão junto a pasta do mesmo, a autoria dessas estão asseguradas ao Professor Rodrigo Maximiliano Antunes de Almeida e ao criador do Picsimlab, Luis Claudio Gambôa Lopes.

Os periféricos utilizados no projeto são:

- Barramento de Leds;
- Display LCD 16x4;
- Teclado Matricial;
- Cooler;
- Display 7-segmentos.

1 - PLANEJAMENTO

Primeiramente, foi elaborado um escopo do projeto, para a definição das funções de cada periférico.

- Barramento de Leds: demonstrar a barra de Fome e Fun (diversão) do Pet;
- Display LCD 16x4: exibição do Pet e de todas as funções aplicadas nele;
- Teclado Matricial: utilizado para recolher a função que o usuário quer utilizar;
- Cooler: utilizado como uma demonstração de ação, toda ação entre o usuário e o Pet, o cooler irá girar;
- Buzzer: acionado quando o usuário perde (game-over)

2 - ESTRUTURA DO PROJETO

Ao iniciar o desenvolvimento do projeto no MPLAB X IDE, foi adicionado todas as bibliotecas necessárias para a implementação, e também incluídas na *main.c*.

```
// Projeto Final - Tamagotchi
#include "pic18f4520.h"
#include "bits.h"
#include "config.h"
#include "io.h"
#include "keypad.h"
#include "lcd.h"
#include "atraso.h"
#include "pwm.h"
```

E também, definidas as variáveis para facilitar os comandos hexadecimais do LCD.

```
// Variáveis do LCD
#define L_ON      0x0F
#define L_OFF     0x08
#define L_CLR     0x01
#define L_L1      0x80
#define L_L2      0xC0
#define L_L3      0x90
#define L_L4      0xD0
#define L_CR      0x0F
#define L_NCR     0x0C
#define L_CFG     0x38
```

A partir desses passos é iniciada a função *main*, onde é inicializado todas as bibliotecas e criado as variáveis de Fome e Fun, que seriam designadas para deterem o valor que seria colocado nos LEDs para as barras de status do Pet. E para isso, é criada as funções *fomeLed* e *funLed* para fazer essa comunicação entre a placa e o código.

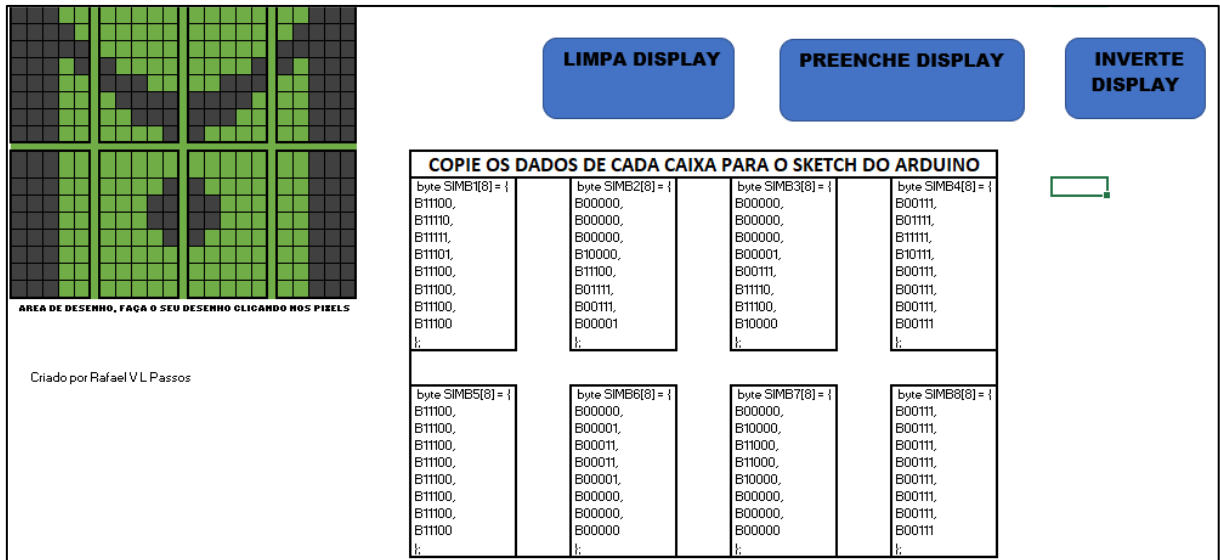
```
// Função responsável por sincronizar os leds do PORTD com a fome
void fomeLed(unsigned char fome) {
    TRISD = 0x00; // setar TRIBD para saída
    switch (fome) {
        case '8':
            bitSet(PORTD, 0);
            bitSet(PORTD, 1);
            bitSet(PORTD, 2);
            bitSet(PORTD, 3);
            bitSet(PORTD, 4);
            bitSet(PORTD, 5);
            bitSet(PORTD, 6);
            bitSet(PORTD, 7);
            break;
    }
}
```

E nesta função é utilizado o switch case, para que cada valor de fome/fun ligue uma quantidade de LEDs (por exemplo, fome = 8 vai ligar todos LEDs, pois o Pet está sem fome).

3 – DESENHANDO NO LCD

O LCD (Liquid Crystal Display) é a base desse projeto, sendo o periférico mais utilizado e que mais teve enfoque. Pois, nele, aconteceu as diversas animações criadas para cada função do programa. Desde as animações de início, até as de alimentação e carinho. Para isso, para facilitar os desenhos nos displays de 5x8 foi utilizado um prom excel para criação em binário destes.

A utilização desse programa foi de suma importância para o desenvolvimento desse projeto, onde consegue dar uma visibilidade muito maior que outros softwares para criação de caracteres. O projeto teve um total de 9 caracteres de animação, que estão no github no documento de texto denominado “*logos.txt*”.



4 – DIFICULDADES E SOLUÇÕES

A maior obstáculo no projeto, sem sombra de dúvidas, foi em encontrar uma maneira de contar o tempo para diminuição das variáveis de Fome e Fun, sem o conhecimento profundo de *clock*, porém, entre vários testes, a solução foi encontrada de uma maneira extremamente simples. Foram criadas 3 variáveis: 2 ints, cont1 e cont2 e um float, tempo.

```
if (cont1 == 250) { // contador de tempo para Fome
    fome--;
    cont1 = 0;
}
if (cont2 == 350) { // contador de tempo para Fun
    fun--;
    cont2 = 0;
}
for (tempo = 0; tempo < 40; tempo++); // +- 10ms
```

E as variáveis cont1 e cont2 foram somadas +1 a cada ciclo do for(;;).

5 – FUNCIONAMENTO DO PET

É utilizado as teclas 1, utilizada para alimentar o Pet com um Cookie, e a tecla 4, utilizada para fazer carinho no Pet. Os barramentos de LED indicam as necessidades do Pet. As ações do usuário serão respondidas com o Cooler, caso o Pet esteja com fome e ele conseguir dar comida, o Cooler irá girar, a mesma coisa ocorre quando o Pet está carente, se o usuário fizer carinho, o Cooler irá girar. E, caso o usuário se distraia e não consiga cuidar do seu Pet da maneira correta, o Buzzer irá apitar e o jogador irá perder (game-over).