

Laboratório de Programação Embarcada – Projeto final

Autor: Victor Eduardo Cauvilla de Oliveira

Calculadora programável com PicGenius

Este projeto foi desenvolvido no final da disciplina de Laboratório de Programação Embarcada, com a motivação de fazer uma aplicação utilizando o PicGenius e que seja largamente utilizada no dia a dia.

A calculadora vem pré-programada com 6 operações, sendo elas:

- Adição
- Subtração
- Multiplicação
- Divisão
- Potência
- Resto da divisão

Mas ela vem também com a facilidade de implementar novas operações sem muito trabalho, bastando criar a função desejada no arquivo ‘operacoes.h’ e seu algoritmo em ‘operacoes.c’.

O código da calculadora foi escrito por mim mesmo, mas foi altamente baseado nos códigos disponibilizados para estudo pelos professores Otávio Gomes e Rodrigo Almeida, docentes da Universidade Federal de Itajubá.

Para realizar este projeto, necessita-se que tenha em mãos um simulador da placa de desenvolvimento PicGenius ou senão, a própria de maneira física e seu gravador de códigos.

Passo 1 – Componentes necessários

Primeiro se necessita decidir quais componentes serão usados nesse projeto. Foi-se escolhido o uso do LCD 16x4 e os 10 botões que já vem fixados na placa.

Passo 2 – Modularização dos códigos do projeto

Para a calculadora funcionar, se faz necessário a presença de alguns códigos de configuração para os componentes, sendo eles:

- ‘defines.h’, biblioteca responsável por toda a configuração do PIC e de suas definições.
- ‘LCD.h’, biblioteca responsável pela configuração e comunicação do PIC com o LCD.
- ‘teclado.h’, biblioteca responsável pela configuração e comunicação do teclado com o PIC.
- ‘operacoes.h’, biblioteca onde será hospedado todos os algoritmos de funções matemáticas que nossa calculadora realizará.

Passo 3 – Construindo as bibliotecas

➤ ‘defines.h’:

Essa biblioteca contém as definições básicas de funcionamento do PIC utilizado na placa de desenvolvimento, assim como de algumas operações BitWise utilizadas.

➤ ‘LCD.h’:

Essa biblioteca é composta por 8 funções, onde:

- nextLine: Responsável por pular a linha no LCD.
- LCD_conf: Responsável pelas configurações do LCD, quando necessário.
- lcd_wr: Responsável pela comunicação do valor enviado do LCD e o valor mostrado no LCD.
- lcd_cmd: Responsável pelo comando enviado ao LCD, e o que será feito com ele.
- lcd_dat: Responsável pela escrita de caráter no LCD.
- lcd_init: Responsável por todas as configurações iniciais necessárias para o funcionamento do LCD.
- lcd_str: Responsável pelo envio de caracteres pertencentes a uma string para o lcd_dat.
- Atraso_ms: Responsável por causar um atraso na leitura, durante a execução do código.

➤ -‘teclado.h’

Esta biblioteca possui duas funções, sendo elas:

- tc_tecla: Responsável pela leitura da tecla pressionada pelo usuário.
- teclaLeitura: Responsável pela configuração necessária antes de se esperar o pressionamento de uma tecla, e então, enviar o valor pressionado de volta ao código.

➤ -‘operacoes.h’

Esta biblioteca possui 3 funções principais e incontáveis outras para os algoritmos de operações matemáticas. As 3 funções principais são:

- myAtoi: Responsável por receber uma string lida pelo teclado e retornar um inteiro equivalente.
- myItoa: Responsável por transformar um inteiro recebido numa cadeia de caracteres, para futura impressão no LCD.
- operacoes: Responsável pelo recebimento de caracteres de representação numérica do código, seu tratamento, a operação realizada e então, o retorno de uma cadeia de caracteres representando o resultado.

Passo 4 – Construindo a função main

Neste passo, foi-se construído a função main, seguindo o seguinte algoritmo:

1. Realiza todas as configurações necessárias.
2. Pede-se o primeiro número da operação.
3. Pede-se o segundo número da operação.
4. Pede-se o código da operação.
5. Imprime o resultado da operação realizada.
6. Aguarda o reset da calculadora para realizar a próxima operação.

Passo 7 – Funcionamento

Com tudo pronto, basta carregar o código na PicGenius e realizar suas operações, como no exemplo de uso abaixo, onde foi realizada uma operação de soma, representada pelo código 1111, entre os valores 1234 e 4321.

