não UNIVERSIDADE DO VALE DO ITAJAÍ

ESCOLA DO MAR, CiÊNCIA E TECNOLOGIA

curso de Engenharia de Computação

miCROCONTROLADORES

Implementação do jogo asteiróide em um microcrontrolador Pic18F4520

por

Diogo Agenor Marchi

George de Borba Nardes

Itajaí (SC), Junho de 2020

Universidade do Vale do Itajaí

ESCOLA DO MAR, CIÊNCIA E TECNOLOGIA

Curso de Engenharia de Computação

miCROCONTROLADORES

Implementação do jogo asteiróide em um microcrontrolador Pic18F4520

por

Diogo Agenor Marchi

George de Borba Nardes

Relatório apresentado como requisito parcial da disciplina Microcontroladores do Curso de Engenharia de Computação para análise e aprovação.

Professor Responsável: Paulo Roberto Valim

(obs.: modelo baseado no usado na disciplina de Projeto de Sistemas Embarcados)

Itajaí (SC), Junho de 2020

Sumário

[1 INTRODUÇÃO 3](#_Toc42454900)

[1.1 Descrição do projeto 4](#_Toc42454901)

[1.2 Objetivo geral 4](#_Toc42454902)

[1.3 Delimitação do escopo 4](#_Toc42454903)

[1.4 Solução Proposta 4](#_Toc42454904)

[1.5 ANÁLISE DE VIABILIDADE 5](#_Toc42454905)

[2 projeto 6](#_Toc42454906)

[2.1 VISÃO GERAL 6](#_Toc42454907)

[2.2 ANÁLISE DE REQUISITOS 6](#_Toc42454908)

[2.2.1 Requisitos funcionais 6](#_Toc42454909)

[2.2.2 Requisitos não funcionais 7](#_Toc42454910)

[2.2.3 Regras de negócio 7](#_Toc42454911)

[2.3 ARQUITETURA DE HARDWARE 8](#_Toc42454912)

[2.4 ARQUITETURA DE SOFTWARE 9](#_Toc42454913)

[2.4.1 Diagrama de Caso de Uso 10](#_Toc42454914)

[2.4.2 Diagrama de Classe 10](#_Toc42454915)

[2.4.3 Diagrama de Estado 12](#_Toc42454916)

[2.4.4 Fluxograma 12](#_Toc42454917)

[2.5 PLANEJAMENTO 13](#_Toc42454918)

[2.6 CRONOGRAMA 14](#_Toc42454919)

[3 DESENVOLVIMENTO 15](#_Toc42454920)

[3.1 IMPLEMENTAÇÃO 15](#_Toc42454921)

[3.2 VERIFICAÇÃO 15](#_Toc42454922)

[3.3 RESULTADOS 15](#_Toc42454923)

[4 Considerações Finais / CONCLUSÕES 16](#_Toc42454924)

1. INTRODUÇÃO
   1. Descrição do projeto

O projeto escolhido envolve o desenvolvimento do jogo de asteroide para um microcontrolador PIC. O objetivo do jogo é fazer com que a nave desvie dos asteroides que estiverem vindo em sua direção. Caso ocorra uma colisão com asteroide o jogo é finalizado. O controle da nave é feito por um GAMEPAD e o jogo é exibido em um display LCD ambos controlado por um microcontrolador.

Como o jogo será executado em um microcontrolador, algumas limitações deverão ser consideradas, como a falta de um SO para gerenciamento de entradas e saídas. Além disso, existe a necessidade de que seja considerado o funcionamento do jogo como um sistema executando várias tarefas ao mesmo tempo. Será feito um chaveamento entre funções para que o usuário tenha a impressão de que todas as funções do jogo estão sendo executadas paralelamente.

* 1. Objetivo geral

Este projeto de conclusão da matéria microcontroladores, tem como objetivo aplicar alguns conhecimentos adquiridos durante o semestre. O tema escolhido pelos alunos foi motivado por englobar alguns dos principais assuntos tratados, como integração do PIC18F4520 com o display HD44780, módulo de tratamento de interrupções externas e interrupções internas do PIC. Além disso, o desenvolvimento de um sistema utilizando task scheduling.

* 1. Delimitação do escopo

Será usado um display de 16 x 4(LM041L) como saída, e como entrada um keypad alterado para as necessidades do jogo. O keypad terá setas no eixo vertical e horizontal, uma tecla “enter” e uma tecla “escape”. O microcontrolador que será usado é o PIC18f4520. Todos os componentes podem ser encontrados no ambiente de simulação Proteus.

* 1. Solução Proposta

Foram feitas algumas pesquisas sobre jogo de Asteroide com o display LCD, e pelo que foi achado, existem jogos apenas com o microprocessador Arduino UNO ou MEGA, então, o asteroide com o PIC18f4520 seria um diferencial, e também, este projeto movimentará a nave nos dois eixos, eixo x e eixo y, em um display LCD 16x4.

Tabela 1– Comparativo da solução proposta com soluções existentes

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome** | **Microcontrolador** | **Display** | **movimento** |
| [SanticN4N](https://www.instructables.com/member/SanticN4N/) | Arduino Uno | LCD16X2 | Sensor movimento |
| Este projeto | PIC18F4520 | LCD16X4 | Controle com botão |

* 1. ANÁLISE DE VIABILIDADE

O projeto escolhido é viável como prova de conceito pois engloba diversos assuntos discutidos durante o semestre. Pode-se citar os conceitos de interrupção, timer e integração com dispositivos externos como principais motivos de viabilidade. Além disso, outro fator que viabiliza é que o nível de dificuldade do projeto fica dentro do esperado levando em conta o tempo disponível para execução.

1. projeto
   1. VISÃO GERAL

O projeto escolhido envolve o desenvolvimento do jogo de asteroide para um microcontrolador PIC. O objetivo do jogo é fazer com que a nave desvie dos asteroides que estiverem vindo em sua direção. O controle da nave é feito por um GAMEPAD e o jogo é exibido em um display LCD. Será usado um display de 16 x 4(LM041L) como saída, e como entrada um keypad alterado para as necessidades do jogo. O keypad terá setas no eixo vertical e horizontal, uma tecla “enter” e uma tecla “escape”.

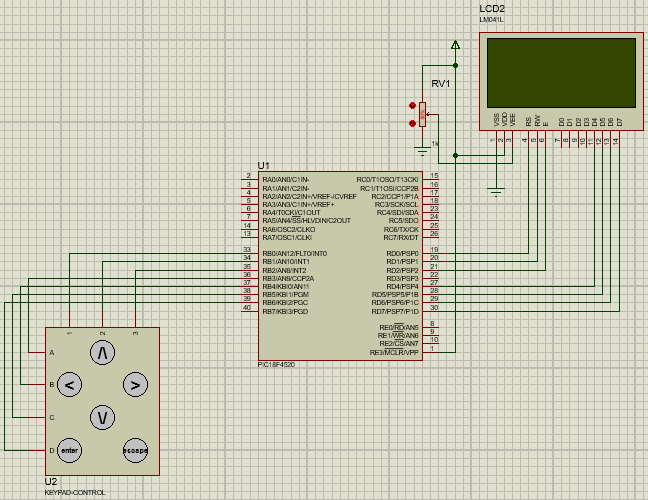


Figura 1- Componentes: Keypad, PIC e Display

* 1. ANÁLISE DE REQUISITOS
     1. Requisitos funcionais
* RF01: O sistema deve permitir que o usuário navegue pelo menu principal;
* RF02: O sistema deve permitir que o usuário selecione uma opção do menu principal;
* RF03: O sistema deve permitir que o usuário retorne ao menu principal a qualquer momento;
* RF04: O sistema deve incrementar a pontuação durante o jogo.
* RF05: O sistema deve permitir que o usuário movimente a nave no espaço em todas as direções.
* RF06: O sistema deve identificar uma colisão da nave com um asteroide.
  + 1. Requisitos não funcionais
* RNF01: o sistema será prototipado no microcontrolador PIC18F4520;
* RNF02: o sistema será simulado na ferramenta de simulação Proteus;
* RNF03: o código da aplicação será escrito em linguagem c;
* RNF04: será utilizado o ambiente de desenvolvimento MPLAB;
* RNF05: o protótipo deverá ter uma vazão de pelo menos 10 frames/segundo (conforme a aplicação);
* RNF06: o sistema deverá ser exibido no display LM041L 16x4
* RNF07: as entradas de usuário serão por meio de um teclado.
* RNF08: o teclado de entrada deve possuir os seguintes botões: CIMA, BAIXO, ESQUERDA, DIREITA, ENTER e ESCAPE.
  + 1. Regras de negócio
* RN01: O menu principal deve ter três opções: jogar, instruções e créditos;
* RN02: A pontuação deve ser incrementada a cada asteroide desviado;
* RN03: Na ocorrência de uma colisão o jogo deve ser interrompido.
* RN04: Ao ser interrompido por uma colisão, o sistema deve informar a pontuação final.
* RN05: Ao ser interrompido por uma colisão e apresentar a pontuação final o sistema deve voltar ao menu principal.
* RN06: A nave poderá ser movimentada nos eixos XY.
  1. ARQUITETURA DE HARDWARE

A figura 2 apresenta um esquemático do projeto de como será ligado os componentes e o que será usado.

A Figura 3 apresenta do diagrama de blocos do projeto. O bloco GAMEPAD é uma matriz de botões adaptada para as necessidades do jogo. O bloco DISPLAY LM041L será o display LCD no qual o jogo será exibido. O bloco de keypad terá as ações controladas pelo bloco Módulo Timer, que é interno ao PIC.

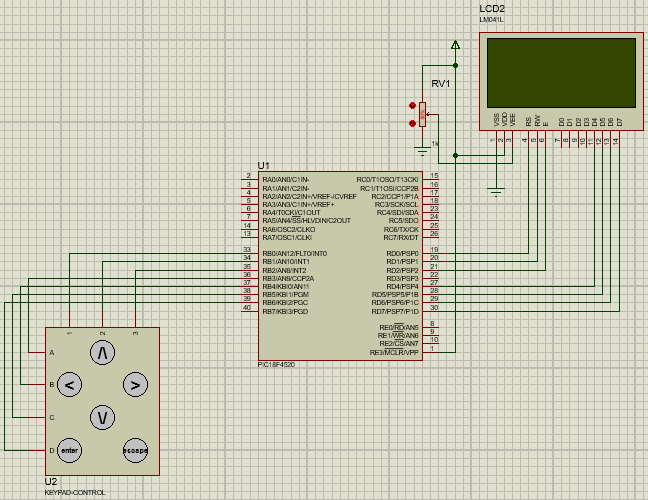


Figura 2 – Esquemático

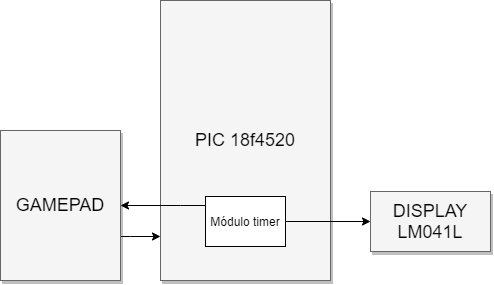


Figura 3- Diagrama de Bloco do Projeto

A saída do GAMEPAD é conectada as portas de entrada e saída do PIC. O estado das teclas será verificado de tempo em tempo e a cada movimento de tecla disponível, há uma atualização nos valores a ser exibido no display.

* 1. ARQUITETURA DE SOFTWARE

Nesta seção será apresentado como planejamos construir o software para o projeto. Como ainda houve nenhuma implementação, pode ser que o que for descrito abaixo sofra algumas alterações conforme limitações ou novas ideias forem surgindo. Ao finalizar a implementação, a arquitetura descrita será revisada e ajustada conforme o resultado final, para que seja uma descrição com maior fidelidade.

O software será planejado da seguinte forma: algumas funções de print de tela, como menu, créditos e instruções, onde essas funções serão chamadas conforme for clicado no keypad control e isso trará ao usuário um software de tempo real. Terá a biblioteca keypad.h/keyppad.c onde nela estará descrita função que lerá as entradas. No arquivo task\_maneger.h/task\_maneger.c foi criado funções, de funcionamento do jogo. O timer foi programado entre 10-20ms para ficar verificando quais botões foram clicados.

Ao compilar o programa, as primeiras linhas de código são para fazer as configurações iniciais do display e logo em seguida entra no task\_manager.c onde existe o while de funcionamento. Nesta função o que aparece por primeiro é o menu, lá ele pode navegar para cima e para baixo para escolher algumas das opções, ao decidir-se da opção o usuário poderá apertar enter para entrar em outra função, pois ao apertar enter a função menu retorna uma variável “OP” para o while do funcionamento do jogo, e nesse while faz uma verificação dessa variável e conforme for o resultado toma uma decisão de ir para a função instruções por exemplo.

Se o código está no ambiente “créditos” e o usuário aperta a tecla “>” o código entende que isso não fará nada, pois neste ambiente o usuário poderá ler quem produziu o projeto e logo após terminar a leitura, ele poderá apertar a tecla “escape” para voltar ao menu, neste ambiente qualquer tecla diferente da “esc” não faz nada.

* + 1. Diagrama de Caso de Uso

A figura 4 mostra os casos de uso envolvidos na interação do usuário com o sistema.

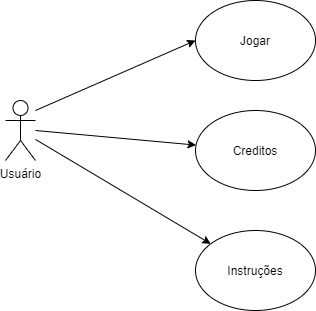


Figura 4 - Diagrama Casos de Uso

O usuário terá apenas 3 ações possíveis no sistema: jogar, visualizar créditos e visualizar instruções.

* + 1. Diagrama de Classe

A Figura 5 apresenta o diagrama de classe, as classes são os arquivos do software. O arquivo principal é o “main.c”. É dele que parte todas as chamadas de funções e procedimentos. O arquivo “config.h” é onde será parametrizado as configurações do microcontrolador. O arquivo “task\_maneger” é o que realizará o controle dos diferentes estados do sistema. Ele fará uso dos arquivos “lcd” e “keypad” para interagir com os componentes externos display e teclado respectivamente.

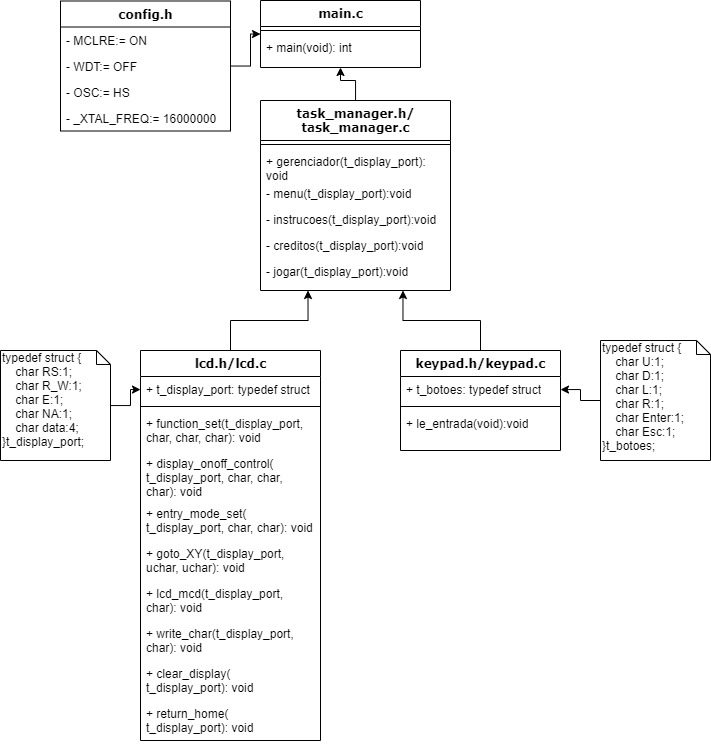


Figura - Diagrama de Classe Completo

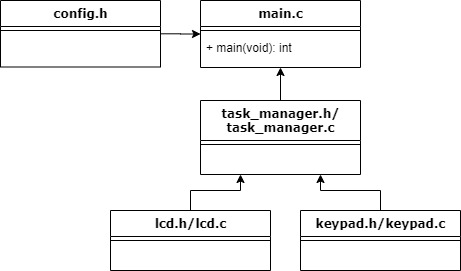


Figura 6 - Diagrama de Classe Simplificado

* + 1. Diagrama de Estado

O diagrama de estado da Figura 6 apresenta todos os estados atingíveis durante a execução do jogo. Fica implícito que cada estado pode ser interrompido ao microcontrolador ser desenergizado.

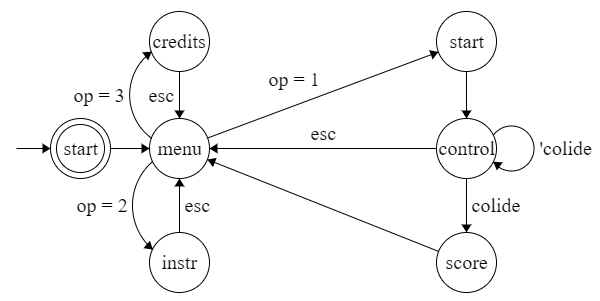


Figura 7- Diagrama de estados do jogo

* + 1. Fluxograma

A fluxograma da figura 7 apresenta todos os passos que o usuário poderá dar ao iniciar o jogo. Ao início do programa, ele começa no menu, que tem 4 possibilidades para o usuário escolher: jogar, instruções, créditos ou finalizar.

Cada tela que ele entrar, se apertar a tecla “escape”, voltará ao menu. Ao entrar no jogo, o usuário comandará a nave através do keypad control, se bater a nave no asteroide, mostrará a pontuação na tela e voltará ao menu. Ao finalizar o programa, o PIC entra em modo Sleep e fica “funcionando” com muito pouco consumo, caso o usuário queira voltar a jogar, basta apertar a tecla “ON” que energizará o PIC novamente.

Texto preto sobre fundo branco

Descrição gerada automaticamente

Figura - Fluxograma

* 1. PLANEJAMENTO

O jogo será feito na linguagem C no MPLAB X com integração do Proteus com alguns requisitos descritos anteriormente. Para verificar se os requisitos poderão ser atendidos, foi feito uma tabela de testes abaixo:

Quadro 1 – Plano de verificação

|  |  |  |
| --- | --- | --- |
| **Requisito** | **Procedimento de verificação/teste** | **Resultado esperado** |
| RF01 | Apertar nas setas do keypad control | Que o cursor modifique |
| RF02 | Apertar enter em alguma das linhas do display | Que troque de tela |
| RF03 | Apertar escape na tela do jogo | Volte ao menu |
| RF04 | Ao passar por asteroides sem colidir | Incremente a pontuação |
| RF05 | Apertar nas setas do keypad control | Nave se mova |
| RF06] | Ao colidir a nave com um asteroide | Acabe o jogo e mostre a pontuação final |
| RNF01 | Verificar o microcontrolador | Ser o PIC18F4520 |
| RNF02 | Verificar a ferramenta de simulação | Ambiente Proteus |
| RNF03 | Verificar linguagem de codificação | Linguagem C |
| RNF04 | Verificar ambiente de desenvolvimento | MPLAB X |
| RNF05 | Verificar vazão do protótipo | 10 frames/segundo |
| RNF06 | Verificar em qual display está sendo apresentado o jogo | LM041L 16x4 |
| RNF07 | Apertar nas teclas | Verificar se mudou algo no display |
| RNF08 | Verificar as teclas do keypad control | Ter as teclas CIMA, BAIXO, DIREITA, ESQUERDA, ENTER e ESCAPE. |

* 1. CRONOGRAMA

Para a primeira semana foi feito uma introdução do que será o projeto, mostrando o que será apresentado e quais componentes seriam usados.

Para a segunda semana será entregue este documento mostrando umas especificações a mais do projeto, como a arquitetura do software e como é pensado a implementação

Terceira, quarta e quinta semana será o tempo para a implementação do código. Quinta semana será a entrega do documento final com o código pronto.

Quadro 6 – Cronograma de execução

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Atividade** | **Sem. 1** | **Sem. 2** | **Sem. 3** | **Sem. 4** | **Sem. 5** | **Entregável** |
| Introdução do projeto | X |  |  |  |  | Documento |
| Especificação |  | X |  |  |  | Documento |
| Implementação |  |  | X | X | X | Código |
| Documentação final |  |  |  |  | X | Documento final e apresentação do projeto |

1. DESENVOLVIMENTO

O projeto foi desenvolvido no MPLAB X na linguagem C com integração de componentes do Proteus. O programa é constituído por 4 bibliotecas: config.h, keypad.h, lcd.h e task\_manager.h.

Config.h é uma biblioteca de configurações do PIC, como por exemplo a velocidade que ele vai trabalhar que é 16MHz, o enable do timer, entre outras configurações.

Keypad.h tem nela a Struct do teclado matricial e a função de leitura de botões. A função funciona da seguinte maneira, é ativado a primeira linha pela energia do pic, e verifica se alguma das colunas dessa linha está ativo, se caso estiver significa que o botão dessa linha e da coluna ativa foi pressionado, logo na struct desse botão é colocado em 1, caso contrario faz nada e faz o mesmo procedimentos para as demais linhas.

Lcd.h é uma biblioteca de configurações do display, como por exemplo printar algum caracter no display, limpar, colocar em alguma posição especificas, entre outros.

O task\_manager.h é todo o funcionamento, nesta função está contido as demais funções como: créditos, instruções, finalizar e jogar. Nesta biblioteca tudo acontecer a partir da função gerenciamento() que chama o menu, e quando for apertado enter, retornará um valor de opção (linha que cursor estava). Conforme for o valor de retorno, chamará uma outra função, se caso vir a opção=2 por exemplo, chamará a função créditos.

* 1. IMPLEMENTAÇÃO

Nesta seção, apresente o máximo de detalhes referentes ao que foi implementado. Descreva o fluxo de ferramentas utilizados e apresente trechos representativos do código fonte (não precisa incluir todo o código). Também apresente fotos do protótipo outras e evidências de implementação.

* 1. VERIFICAÇÃO

Apresente os procedimentos de verificação utilizados e os resultados obtidos com a aplicação do plano de verificação. Transcreva o quadro apresentado na seção anterior e acrescente uma coluna para descrever os resultados obtidos. É importante analisar o grau de atendimento de cada requisito. Também apresente imagens de experimentos de verificação, tais como diagramas de formas de onda de simulação, capturas de tela de console e equipamentos de instrumentação, fotos da operação do protótipo. Procure descrever como a imagem evidencia o cumprimento dos requisitos.

* 1. RESULTADOS

Apresente os resultados experimentais com as métricas de avaliação do protótipo, discutindo ao máximo os resultados obtidos. Procure também resumir esses resultados em tabelas.

1. Considerações Finais / CONCLUSÕES

No documento de projeto, esta seção deve ser denominada “Considerações Finais”. Nela, apresente suas considerações, identificando limitações encontradas na elaboração no projeto e soluções a serem adotadas na implementação. Procure discutir também qualquer aspecto julgue pertinente ao trabalho.

No relatório final, após a implementação, renomeie este capítulo para “Conclusões”. Procure resumir os principais resultados e conhecimentos produzidos. Também procure discutir as limitações do trabalho e indicar futuros trabalhos que possam ser feitos a partir dos estudos, implementações e experimentos já realizados, incluindo melhorias e desdobramentos.

Referências

Apresente as referências citadas no documento seguindo as normas da ABNT para elaboração de referências (NBR 6023 de agosto de 2002, ou posterior).