

MICROCONTROLADORES – 1º LABORATÓRIO

Flavio Henrique A. dos Santos

flaviohenriqu@gmail.com

Marsol L. Araújo

marsollaraujo@gmail.com

Departamento de Engenharia Elétrica (DEL) - UFS
Cidade Universitária Prof. José Aloísio de Campos, Jardim Rosa Elze,
CEP 49100-000 – São Cristóvão – SE

Resumo: Este artigo tem o intuito de demonstrar os resultados obtidos nas experiências realizadas no 1º Laboratório da disciplina de Microcontroladores. Sendo utilizado um *protoboard* com alguns periféricos associados ao microcontrolador DSPIC30F4011, um gravador de PIC, os *Softwares MPLAB C30 e PICKit2*. Os programas foram implementados em linguagem C. Foram desenvolvidas atividades práticas envolvendo conceitos básicos sobre a utilização dos pinos do DSPIC30F4011 e de alguns periféricos associados a ele.

Palavras Chaves: Microcontrolador, PIC, DSPIC30F4011, linguagem C.

Abstract: This article has the objective to show the results of an experience that was realized on the 1º Laboratory of Microcontrollers. As used one *protoboard* with some peripherals associated at the microcontroller DSPIC30F4011, a recorder of PIC, the softwares MPLAB C30 and PICKit2. These programs were implemented in the C language. Were developed practice activities involving the basics concepts about the use of pins of the DSPIC30F4011 and of some peripherals associated with it.

Keywords: Microcontroller, PIC, DSPIC30F4011, C language.

1 INTRODUÇÃO

Microcontroladores são componentes eletrônicos dotados de “inteligência” programável. Podem apresentar dentro de seu encapsulamento um chip CPU, memórias RAM e ROM, interface serial, interface paralela, temporizadores e circuito de interrupção. São aplicados no controle de processos lógicos, ou seja, controle de periféricos como, LEDs, displays, teclas, rês, entre outros. Devido a isso, os microcontroladores apresentam instruções para o controle de entradas e saídas, para setar e limpar bits individuais e executar outras operações orientadas a bit.

Os microcontroladores da família dsPIC30F, projetados pela empresa Microchip Technology, são processadores de 16 bits com arquitetura Harvard modificada com um conjunto de instruções estendido. Eles compartilham as vantagens de um processador digital de sinal, que é a capacidade de processamento, com os atributos de controle de um microcontrolador. Por isso são conhecidos também como controlador digital de sinal (DSC – *Digital Signal Controller*). Apresentam uma alta integração e alta performance.

O microcontrolador utilizado nesse laboratório foi um DSC, mais especificamente o dsPIC30F4011. Seu diagrama de pinos é apresentado na Figura 1 a seguir.

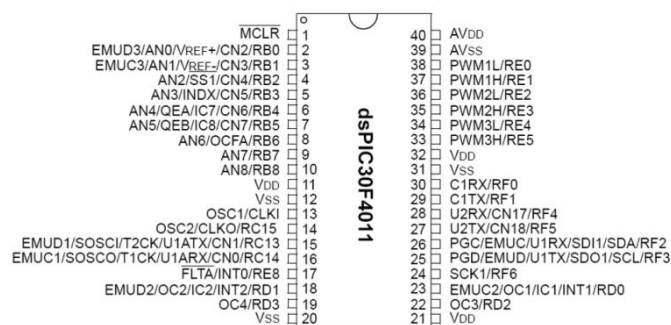


Figura 1 - Diagrama de pinos.

O tamanho do espaço de memória de programa do DSPIC30F4011 é de 16K palavras, isto é, como cada palavra corresponde a 3 bytes, a memória é de 48KB. Em relação ao número de portas, este apresenta as seguintes portas: B, C, D, E e F. Os registradores são o TRISX, PORTX, LATX, e o ADPCFG, cada um com uma devida função. O TRISX e o ADPCFG controlam a operação dos pinos da porta A/D, enquanto que o PORTX e o LATX controlam a direção.

2 MOTIVAÇÃO E OBJETIVOS

Este trabalho tem o objetivo geral de familiarização com o microcontrolador DSPIC30F4011. Os objetivos específicos são listados a seguir:

- Utilização da linguagem C para o dsPIC;
- Utilização dos pinos do DSPIC30F4011;
- Utilização de teclas, displays de sete segmentos, LEDs e outros periféricos;
- Criatividades.

3 METODOLOGIA

O projeto dos experimentos do 1º Laboratório de Microcontroladores consistiu em desenvolver programas em linguagem C a partir dos problemas propostos envolvendo o DSPIC30F4011.

Em geral, cada problema foi analisado e suas respectivas soluções foram sugeridas através de um fluxograma. Feito isso, os fluxogramas foram transcritos em C para o *Software MPLAB C30* e testados suas sintaxe. Depois, os códigos dos devidos programas foram gravados no DSPIC30F4011 e executados em um *protoboard* com os periféricos necessários.

4 DESENVOLVIMENTO DOS PROJETOS

4.1 EXPERIÊNCIA 1

Para primeira experiência foi solicitado a elaboração de um programa que simule uma tranca digital. Para isso, são necessárias duas teclas de controle, uma para escolha do número e outra para confirmação da escolha; quatro *displays* de 7 segmentos, para exibição dos caracteres; e um *buzzer*. Os detalhes do programa são listados a seguir:

- Quando se digita o número 1450 (só nessa ordem 1, 4,...) o código secreto, os *displays* devem mostrar os números digitados se deslocando de direita para esquerda e depois do último dígito mostrar a palavra “dSPIC30F4011” deslocando-se para a esquerda tendo 4 espaços vazios antes de cada repetição. Sinal de que está correto o número digitado;
- Se digitar um número de 4 dígitos diferente de 1450 (a verificação do número deve ser depois de entrarem os 4 dígitos), os *displays* devem mostrar os números digitados se deslocando de direita para esquerda e depois do último dígito mostrar a palavra “ErrO” piscando a intervalos de 0,5 seg aproximadamente;
- Se digitar três vezes o número errado deverá ser ativado o buzzer durante 5 segundos e novamente fica habilitado para entrar o número;

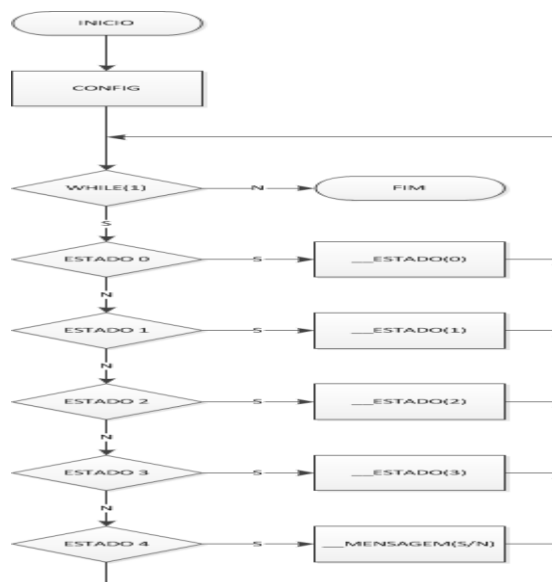


Figura 1 – Fluxograma Experiência 1

Dividiu-se todo o problema em Máquina de Estado Finito (FMS), primeiramente consideramos cinco estados para a máquina principal e dentro do método `__estados()` elaboramos outra FMS para apresentação da senha selecionada. No último estado realizamos a verificação da senha, se a mesma estiver incorreta incrementamos um contador para analisarmos posteriormente as três tentativas frustradas de entrada da senha, para ativarmos o buzzer durante cinco segundos e voltarmos ao estado inicial. Logo em seguida temos um método para exibir a mensagem de “ErrO” juntamente com o código informado piscando e se deslocando, caso o senha esteja incorreta, caso contrário exibirá a sequência “1450dSPIC30F4011” se deslocando. Como implementamos o problema com FMS quando estamos no último estado e pressionamos o botão que representa o Enter do sistema retornamos ao estado inicial, para informarmos a senha novamente.

Como criatividade da experiência 1, criamos uma máquina de estado com 32 estados para deslocarmos a mensagem de sucesso da senha a princípio de baixo para cima e em seguida da direita para a esquerda com toda a palavra, criando assim uma perspectiva diferente do deslocamento da palavra.

4.2 EXPERIÊNCIA 2

A segunda experiência consistiu em implementar o resultado de uma combinação de ativação e/ou desativação de duas teclas (*D1* e *F6*) de entrada. Os comandos de cada combinação são aplicados em periféricos (*displays* de 7 segmentos e *buzzer*) e seus resultados são apresentados na Tabela 1.

<i>D1</i>	<i>F6</i>	<i>Display_1</i>	<i>Buzzer</i>	<i>Display_2</i>
0	0	Mostra o “0”	Ligado 2 seg	Mostra o “0”
0	1	Mostra o “1”	Ligado 2 seg	Mostra o “2”
1	0	Mostra o “3”	Desligado	Mostra o “4”
1	1	Mostra o “5”	Desligado	Mostra o “6”

Tabela 1

O fluxograma proposto para essa experiência é apresentado na Figura 2.

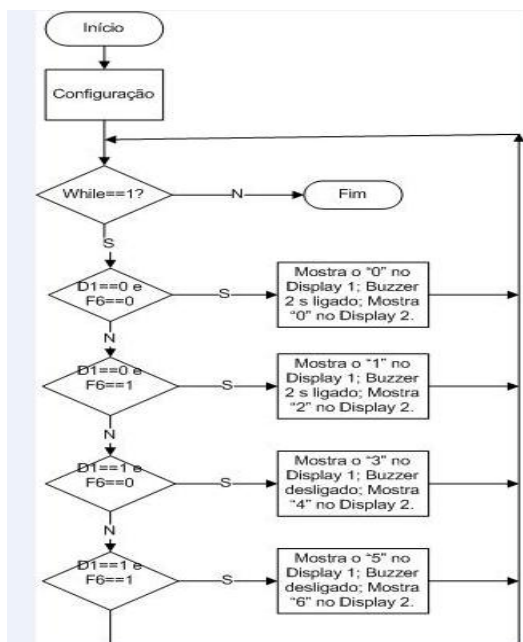


Figura 3 – Fluxograma Experiência 2

Como pode ser observado na Figura 3, para cada condição existe um comando específico. No desenvolvimento do código, esses comandos específicos consistiam em enviar instruções a uma função, chamada de `__ligarDisplay()`, e executá-la. Tais instruções eram um vetor de duas posições com os valores a ser exibido nos *displays*, um vetor de sete posições com as representações numéricas de 0 a 6, e duas variáveis de controle. Em relação à criatividade, foi adicionado mais uma tecla (*F0*) para, quando pressionada, ativar e *buzzer*, fazendo-o “tocar” uma sequência de notas musicais. Para isso, foram definidas frequências, representando notas musicais, que quando combinadas representaram uma melodia. Como o *buzzer* possui uma frequência específica, ocorreu uma variação na melodia.

4.3 EXPERIÊNCIA 3

A terceira experiência é baseada em apresentar os comandos a serem executados quando uma tecla (*D1*) é pressionada seguidas vezes. De acordo com o fluxograma (Figura 4), quando o sistema é iniciado todos os *LEDs* devem estar desligados. Quando se pressiona e solta a tecla uma primeira vez, todos os *LEDs* se ativam. Pressionando uma segunda vez (a qualquer momento), os *LEDs* de 0 a 7 se ativam sequencialmente em intervalos de 500 *ms*. Pressionando uma terceira vez (a qualquer momento), os *LEDs* de 7 a 0 se ativam sequencialmente em intervalos de 500 *ms*. Uma quarta vez (a qualquer momento), os *LEDs* se ativam em pares (0,7), (1,6), (2,5) e (3,4) a intervalos de 500 *ms*. E por fim, uma quinta vez (a qualquer momento), o sistema retorna ao primeiro estágio.

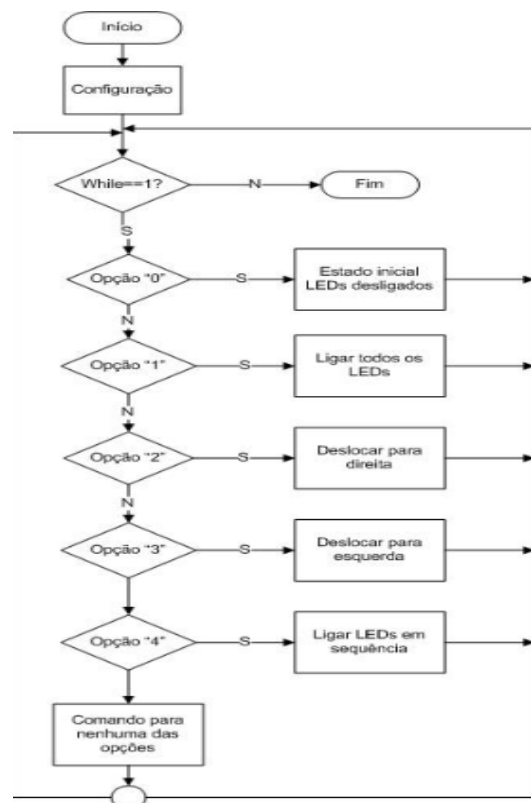


Figura 4 – Fluxograma Experiência 3

Para desenvolvimento do código foi utilizado Máquina de Estado Finito (FMS). A cada estado as instruções eram executadas e o sistema era “atualizado” para saber qual o próximo estado. No primeiro caso, o sistema apenas recebe um comando para todos os *LEDs* estarem apagados e atualiza. No segundo, o sistema verifica se a tecla (*D1*) foi pressionada e, quando liberada, recebe um comando para todos os *LEDs* estarem ligados e atualiza. No terceiro caso, verifica se a tecla (*D1*) foi pressionada e, quando liberada, executa a função `__deslocaEsquerda()` e se atualiza. No quarto, verifica também se a tecla (*D1*) foi pressionada e, quando liberada, executa a função `__deslocaDireita()` e se atualiza. No quinto, também verificação da tecla e execução da função `__sequencia()`. E por último, sistema reiniciado.

Para criatividade, foi adicionado mais uma tecla (*F6*) para controle da intensidade luminosa dos *LEDs*. Para isso foi utilizado modulação PWM. Definiu-se alguns *duty cycle* para implementarmos a variação da luminosidade dos *LEDs*, que quando estava no estado inicial, com todos os *LEDs* ligados, pressionasse a tecla SW1, uma função contendo a variação da intensidade luminosa será ativada.

5 RESULTADOS E DISCUSSÃO

Analisando cada experiência separadamente, a primeira apresentou resultados satisfatórios, tendo o correto funcionamento do programa e da criatividade desenvolvida. Na segunda, o programa foi executado normalmente, apresentando os resultados esperados, porém, durante a criatividade, as notas musicais tocadas pelo *buzzer* apresentavam muito ruído, causados pelo fato do *buzzer* não ser o dispositivo ideal para a atividade, já que este trabalha com uma única frequência. Em relação à terceira experiência, o programa também teve o funcionamento correto, e assim como na segunda experiência, a criatividade também apresentou um problema. Não era nítida a variação da intensidade da luminosidade dos *LEDs*, em razão de uma conexão errada na montagem do circuito. Todas as experiências foram divididas em funções de modo a facilitar seu entendimento e possível reutilização do código.

6 CONCLUSÃO

O foco deste laboratório consiste no aprendizado do funcionamento básico do microcontrolador DSPIC30F4011. As experiências realizadas serviram para aprimoramento do conhecimento adquirido em sala de aula sobre utilização dos pinos, registradores, portas de entrada e saída, além do aprendizado na utilização correta dos periféricos e gravador de PIC.

7 REFERÊNCIA BIBLIOGRÁFICA

[Hidalgo, 2013.1] Hidalgo, A. R. 2013.1, Apostilas da disciplina Microcontroladores da Universidade Federal de Sergipe ministrada pelo Prof. Dr. Antônio Ramirez Hidalgo.

[Souza, 2008] Souza, V. A. Programação em C para o DSPIC: Fundamentos – São Paulo: Ensino Profissional, 2008.