



Tutorial de PIC18F aplicado a Monitoria de Sistemas Embarcados do IFCE

Thiago O. Rodrigues – thiagoliveira08@gmail.com

Jorge F.M.C. Silva– jf.engtelecom@gmail.com

Katielle D. Oliveira – katielledantas@gmail.com

Stephanie A. Braga – stephanie.abraga@gmail.com

Antônio H. S. Lira – lira.hugo92@gmail.com

Caio R. F. Barbosa – caio.raveli@gmail.com

Jose W. M. Menezes – wally@ifce.edu.br

Carlos M. J. M. D. Junior – cmauriciojd@gmail.com

Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE), Departamento de Telemática.

Avenida 13 de maio, 2081, Benfica.

60040-531 – Fortaleza – Ceará

Resumo: *Associar os conteúdos vistos e sala de aula com a prática tem sido um grande desafio. Esse trabalho tem como objetivo mostrar uma maneira de dar a alunos de Engenharia, com foco em Sistemas Embarcados, uma ferramenta de aprendizado alternativo ao visto nas aulas, apresentando um tutorial desenvolvido no programa de monitoria das disciplinas de Sistemas Embarcados e Microcontroladores e pelo Grupo de Desenvolvimento de Sistemas de Telecomunicações e Sistemas Embarcados do Instituto Federal de Educação, Ciência e Tecnologia do Ceará.*

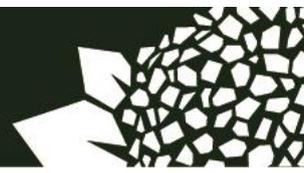
Palavras-chave: *Tutorial, Monitoria, Microcontroladores, Sistemas Embarcados.*

1. INTRODUÇÃO

O Grupo de Desenvolvimento de Sistemas de Telecomunicações e Sistemas Embarcados (GDESTe), grupo criado no Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE), exige que os membros pesquisadores desenvolvam atividades relacionadas à Pesquisa e Inovação como formação complementar ao que é estudado na faculdade e no desenvolvimento de novos projetos.

A dificuldade encontrada para o desenvolvimento dessas atividades se deve ao fato de existirem muitas dificuldades para a montagem de experimentos na área de *hardware*, devido à complexidade dos projetos que exigem infraestrutura adaptada a essa realidade. Além disso, a programação para esses dispositivos exige um conhecimento da arquitetura usado no desenvolvimento das atividades.

Além disso, a montagem e desmontagem desses ensaios ao longo da disciplina torna-se complicado pelo número de alunos que fazem disciplinas ligadas a essa área ao longo dos semestres. Devido a isso, o departamento adquiriu kits para auxiliar no ensino



dessas disciplinas. Aliado a isso, foi iniciado monitorias nas áreas de Microcontroladores e Sistemas Embarcados, com o objetivo de minimizar os problemas encontrados pelos alunos e para confecção de materiais que auxiliarão no ensino dessas disciplinas.

Com isso, o objetivo desse trabalho é expor um tutorial e explicar o funcionamento do kit, e por conseguinte, os periféricos usados, como usar recursos oferecidos pelas arquiteturas trabalhadas nas disciplinas, produção e documentação de códigos exemplos. Esses trabalhos estão distribuídos entre a modelagem do kit no *software* Proteus, a produção de códigos exemplos, o comentários dessas aplicações e a disponibilização deles para os alunos.

2. REVISÃO BIBLIOGRÁFICA

2.1. Microcontrolador PIC e o compilador C18

Os microcontroladores da família PIC são divididos nas séries *PIC10*, *PIC12*, *PIC16*, *PIC18*, *PIC24*, *dsPIC30*, *dsPIC33* e *PIC32*. Cada uma dessas séries contém diversos modelos como diferentes características, mas também com várias semelhanças. (ZANCO, 2010)

Neste trabalho optamos por usar o *PIC18F4550*, da família *PIC18*. Esta escolha deve-se ao fato deste modelo ser de uso bastante comum no mercado e por ser o microcontrolador contido na placa do *kit* escolhido. Este microcontrolador possui 40 pinos divididos em cinco portas de entrada e saída. Possui vários periféricos como os módulos *Timer0*, *Timer1*, *Timer2*, *Timer3*, conversor analógico/digital, módulo comparador, módulo de detecção de alta e baixa voltagem, três pinos de interrupção externa, módulo e *EUSART*. (ZANCO, 2010)

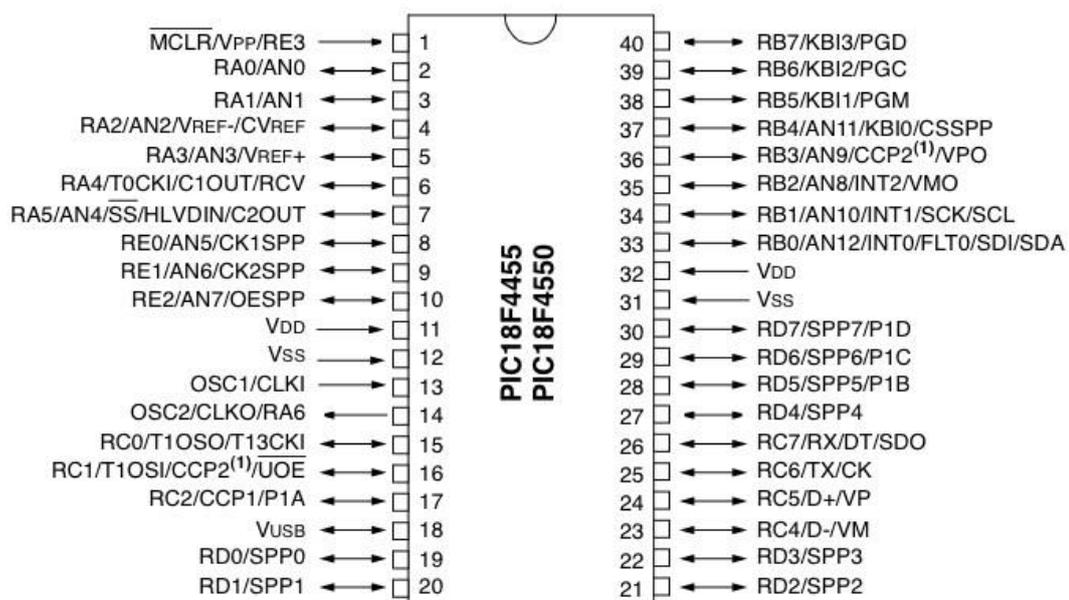


Figura 1 – Mapa de pinos do PIC18F4550

O MPLAB C18 é um compilador C destinado aos microcontroladores da família PIC18. Ele é compatível com o padrão ANSI, permite misturar códigos C e Assembly em um único projeto, permite configurar níveis de otimização de código e dispõe de uma vasta biblioteca. (MIYADAIRA, 2009)

2.2. MPLAB IDE

O MPLAB IDE, desenvolvido pela *Microchip*, é um ambiente de desenvolvimento integrado, que permite a integração com o compilador C18 ou outro compilador. Este software permite gerenciar o projeto, compilar, *debugar* e gravar o microcontrolador. (MIYADAIRA, 2010)

2.3. Dispositivos periféricos

2.3.1. Displays

Displays são componentes opto-eletrônicos capazes de mostrar informações, tais como medidas e *status* vindos de circuitos.

O display de LCD alfanumérico é uma interface de saída capaz de mostrar diversos caracteres do código ASCII. Através de um controlador interno, é capaz de reconhecer instruções predefinidas.

O display de 7 segmentos consiste em um conjunto de *leds* (diodo emissor de luz) todos com um dos terminais em comum. É capaz de exibir números de 0 a 9 e os caracteres A, B, C, D, E e F. Podem ser do tipo ânodo comum ou cátodo comum como exibido na figura 2.

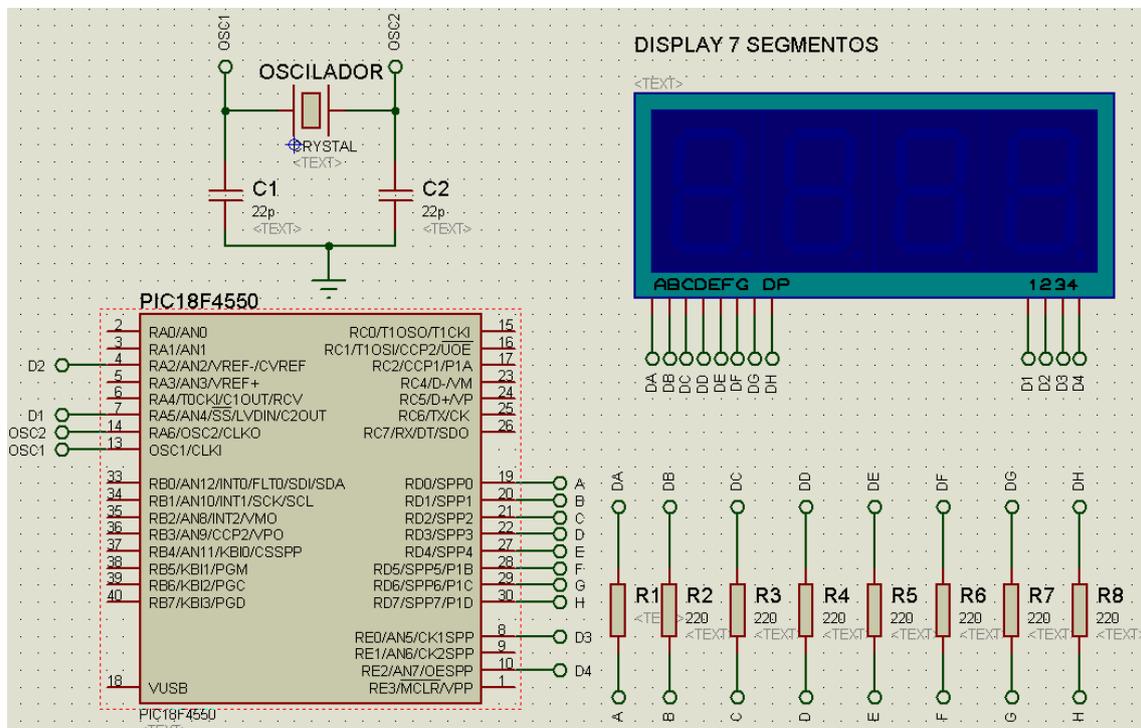


Figura 2 - Display de 7 segmentos

2.3.2. Teclado matricial

As teclas são geralmente organizadas em matrizes, podendo ser mecânicas ou sensíveis a toque. Cada tecla é ligada a uma linha e uma coluna e quando uma tecla é pressionada, a linha e a coluna correspondentes a tecla são conectadas, e essa conexão é detectada pelo circuito controlador. O esquema de ligação é mostrado na figura 3.

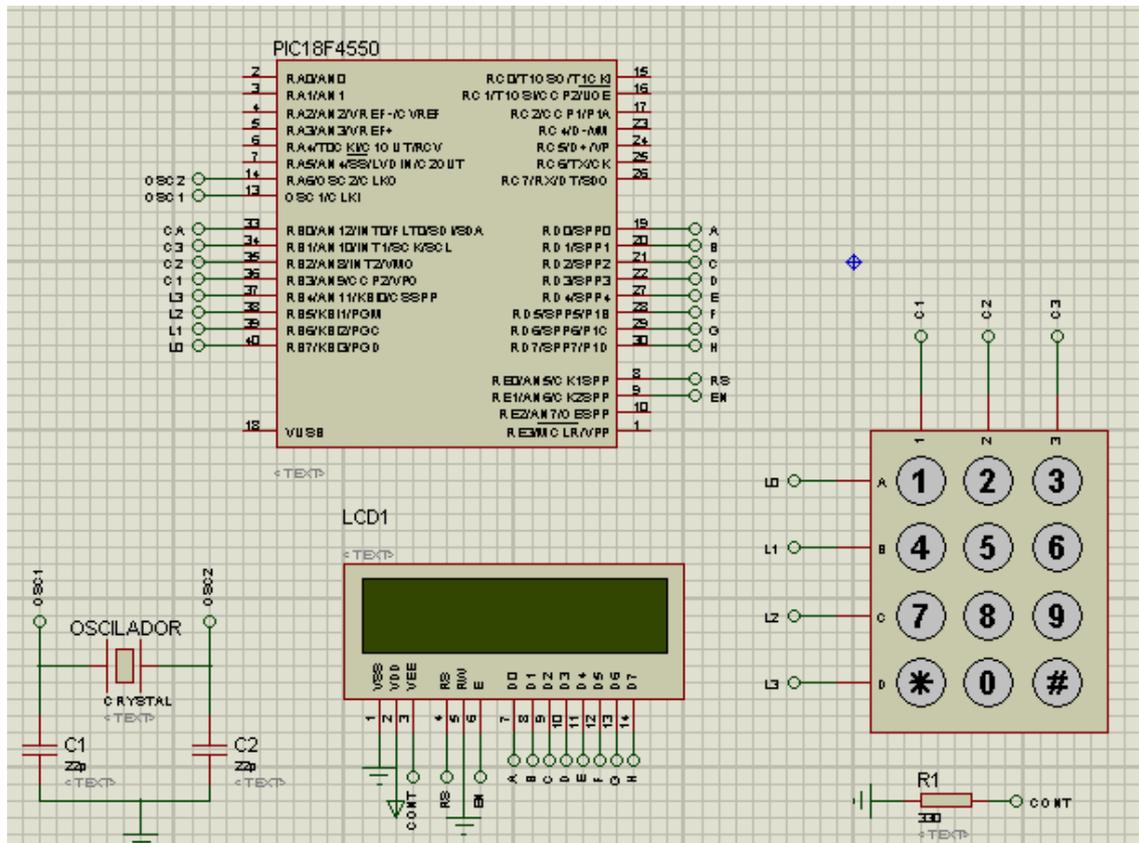


Figura 3 – Teclado matricial

2.3.3. PWM

O *PWM* (Modulação por largura de pulso) é um método que consiste em controlar a tensão disponível para a carga, modificando a largura de pulso do sinal e consequentemente a potência entregue, modulando a sua razão cíclica (*duty-cycle*). Quanto maior for a razão, maior a potência disponível. A forma de onda gerada na saída é semelhante ao que é mostrado na figura 4, abaixo:

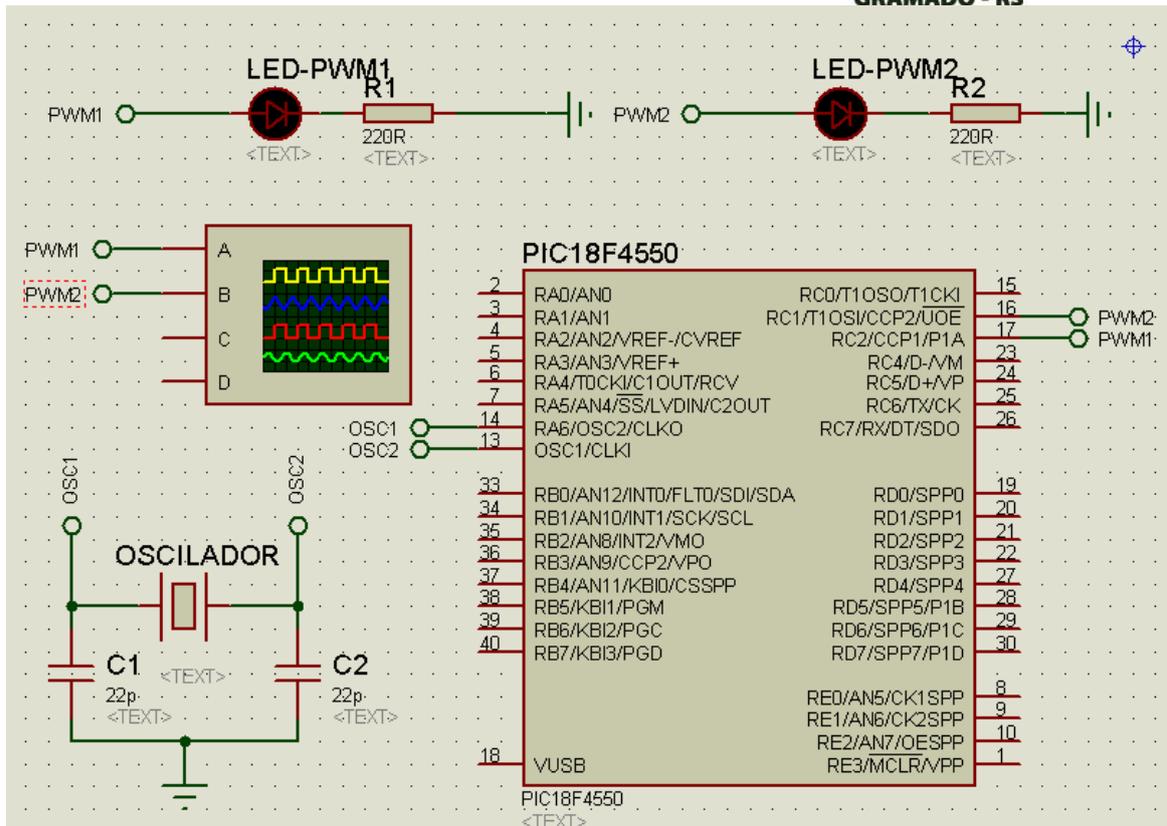


Figura 4 – Sinal modulado visto pelo osciloscópio

2.3.4. Interrupções

As interrupções permitem suspender a qualquer instante a execução de um programa. Por isso, desempenham importantes funções importantes no desenvolvimento dos projetos de Sistemas Embarcados. O processador retorna ao ponto do programa onde estava antes de a interrupção ser gerada. (MIYADAIRA, 2010)

As fontes podem ser externos, como a mudança de estado de um pino de entrada através de interrupções externas, como também de um evento interno, como estouro de *timer*, conversão A/D etc.

As interrupções externas podem ocorrer ou quando houver alterações no estado do pino, sendo assim denominadas interrupções por borda, ou pelo nível lógico, sendo assim denominadas interrupções por nível. Abaixo, tem-se uma exemplo, Figura 5, para testar esse recurso.

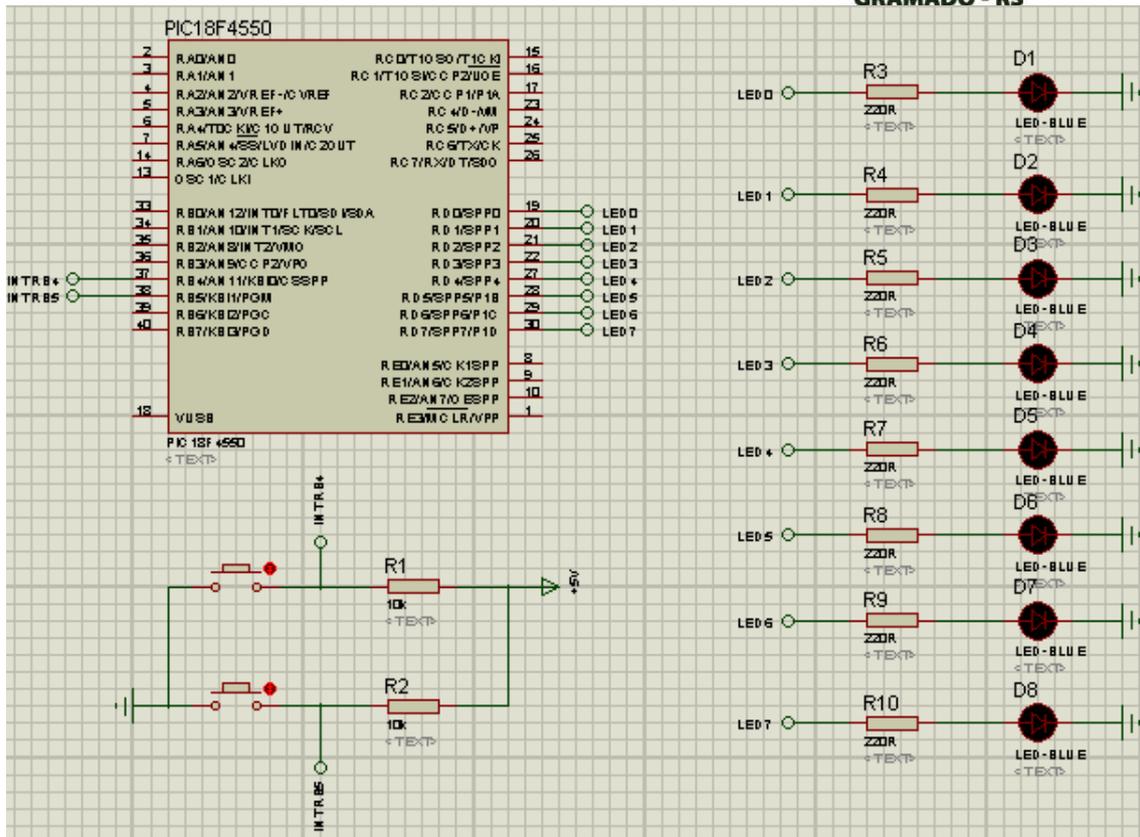


Figura 5 – Exemplo para interrupção da porta B.

2.4. Kit Exto XM11 NEO201

O *kit Exto XM11 NEO201* é composto por diversos periféricos já montados na placa de desenvolvimento. Isso facilita o desenvolvimento de diversas atividades, por reduzir o tempo para a confecção dos circuitos que serviriam apenas para efeito de testes.

O *kit* possui uma interface de comunicação RS232 que converte sinais da faixa de tensão *TTL* para as faixas de tensão deste padrão.

A placa conta com um display alfanumérico de LCD de duas linhas por 16 colunas que possibilita a escrita de mensagens, amostragens e resultados de testes. A comunicação do microcontrolador e o display é feita de maneira paralela, no modo de 8 *bits*. Os quatro *displays* de sete segmentos, do tipo ânodo comum, são multiplexados e o envio de dados para eles é feito através da porta *B* do microcontrolador.

Em uma matriz de 4 linhas por 4 colunas são dispostos 16 botões de modo a formar um teclado matricial que pode ser utilizado através do método de varredura.

O *kit* contém ainda um filtro de sinal *PWM* para geração de sinal analógico. O sinal é aplicado a um filtro passa-baixas de segunda ordem. O sinal DC, na saída, na saída do filtro, é proporcional ao *duty-cycle* do sinal *PWM*.

Outros dispositivos de entrada e saída também formam este *kit*, são eles: 8 leds, 1 ventoinha, 1 potenciômetro, 1 resistência para aquecimento, um sensor de temperatura e *driver* para motor de passo. Uma visão geral do *kit* é mostrada na figura 6.

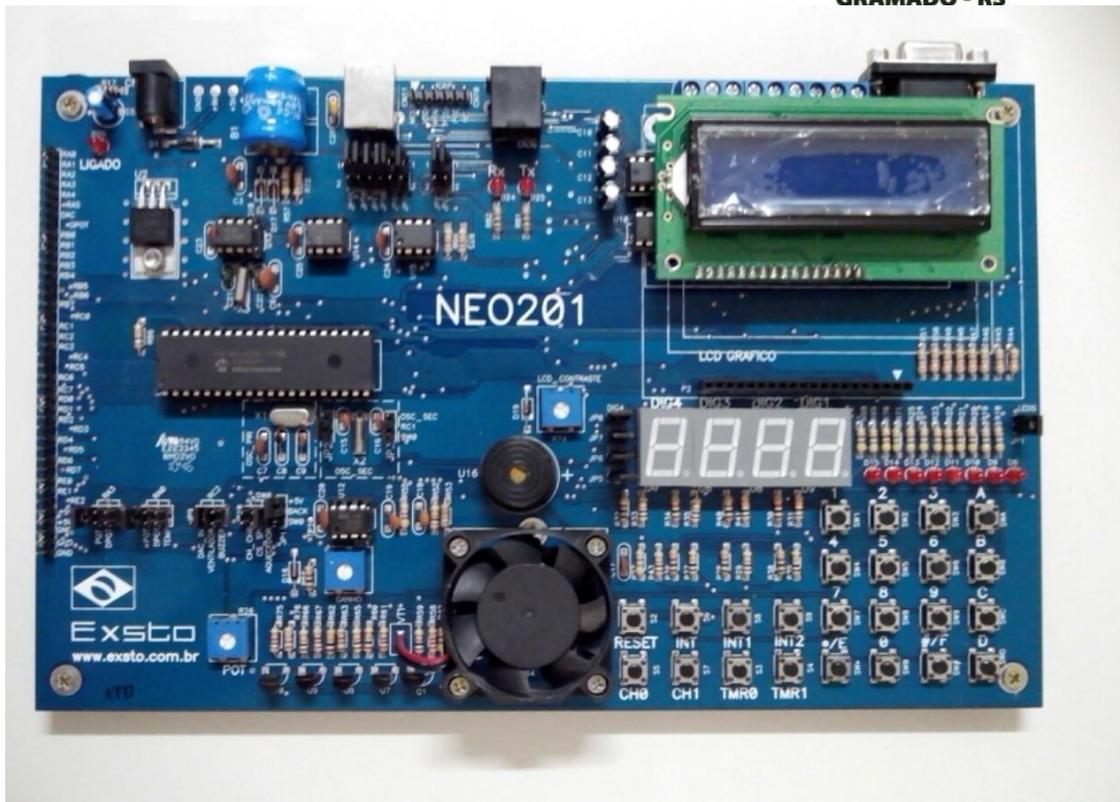


Figura 6 – Kit Exsto NEO201

2.5. Documentação de código usando o Doxygen

O Doxygen é uma ferramenta de documentação para *C++*, *C*, *Java*, *Objective C*, *Python*, *IDL* além de algumas extensões de *C#*, *PHP* e *D*. Esta documentação é vital quando falamos em reutilização de código. Um exemplo do uso dessa ferramenta é mostrada na Figura 7.

A estratégia de documentar o código permite a elaboração de uma boa descrição do programa, além de permitir que o usuário do código cliente tenha uma visão geral dos programas ou da *API (Application Programming Interface)*.

O *Doxygen* é uma ferramenta gratuita disponível para os sistemas operacionais *Windows*, *Linux* e *Mac OS*. Para gerar a documentação de maneira correta, o usuário deve seguir determinados padrões, cada um para finalidades distintas. Os padrões são descritos no tutorial apresentado por esse trabalho. Para fazer uma descrição breve, por exemplo, basta seguir o modelo apresentado pela Figura 7 a seguir.



file:///H:/IFCE/Monitoria/MATERIAL - PIC18F4550/CODIGO MPLAB C18/Documentação/html/exemplo4_8c.html

Grupo de Desenvolvimento em
Sistemas de Telecomunicações
e Sistemas Embarcados

Exemplo 4 PIC18F4550-C18 1.0
Monitoria de Sistemas Embarcados

GDESTE

Página Principal | **Arquivos**

Lista de Arquivos | Arquivos Membros

H: > IFCE > Monitoria > MATERIAL - PIC18F4550 > CODIGO MPLAB C18 > PRATICA 4 - LCD e KEYPAD >

Referência do Arquivo exemplo4.c

Vá para o código-fonte deste arquivo.

Funções

void	delay100us (void)	DELAY FUNCTIONS.
void	delay5ms (void)	
void	initKeypad ()	KEYPAD FUNCTIONS.
unsigned char	scanningKeypad ()	
void	WriteLCD (unsigned char data, unsigned char CD)	LCD FUNCTIONS.
void	setCursorLCD (unsigned char x, unsigned char y)	
void	clearLCD (void)	
void	initLCD (void)	

 Figura 7- Modelo gerado pelo *Doxygen*.

2.6. Proteus

O Proteus é um *software* de simulação e de prototipação desenvolvido pela *Labcenter Electronics*. É composto por dois componentes: o *Isis* e o *Ares*. O *Ares* é utilizado para prototipação de placas de circuitos impressos enquanto o *Isis* é um poderoso simulador com recursos bem robustos.

No *Isis* é possível fazer simulações de circuitos microcontrolados, analógicos e digitais. Ele possui uma vasta biblioteca de componentes, como microcontroladores das famílias PIC e ATmega, além de componentes eletrônicos como resistores, capacitores e indutores. Para circuitos de potência ele dispõe de relés e optoacopladores dentre outros componentes mecânicos e ópticos.

O *Isis*, mostrado na Figura 8, possui ferramentas que permitem que o usuário faça uma análise das grandezas contidas no projeto, como tensão, corrente e frequência, através de voltímetros, amperímetros, osciloscópios, frequencímetro e analisadores de protocolos SPI (*Serial Peripheral Interface*), IC (*Inter-Integrated Circuit*) e Serial UART (*Universal Asynchronous Receiver/Transmitter*).

Neste trabalho, o Proteus, através da ferramenta *Isis*, é utilizado como uma ferramenta didática para a simulação de práticas e exercícios, dispensando a montagem dos circuitos na etapa inicial do aprendizado e facilitando o entendimento. Isso é feito também devido ao fato da placa ter muitos pinos compartilhados, podendo prejudicar o aprendizado de alguns alunos que não são, ou não estão, familiarizados com esse tipo de ferramenta, portanto o Proteus visa “dividir” esse kit de acordo com o andamento da disciplina.

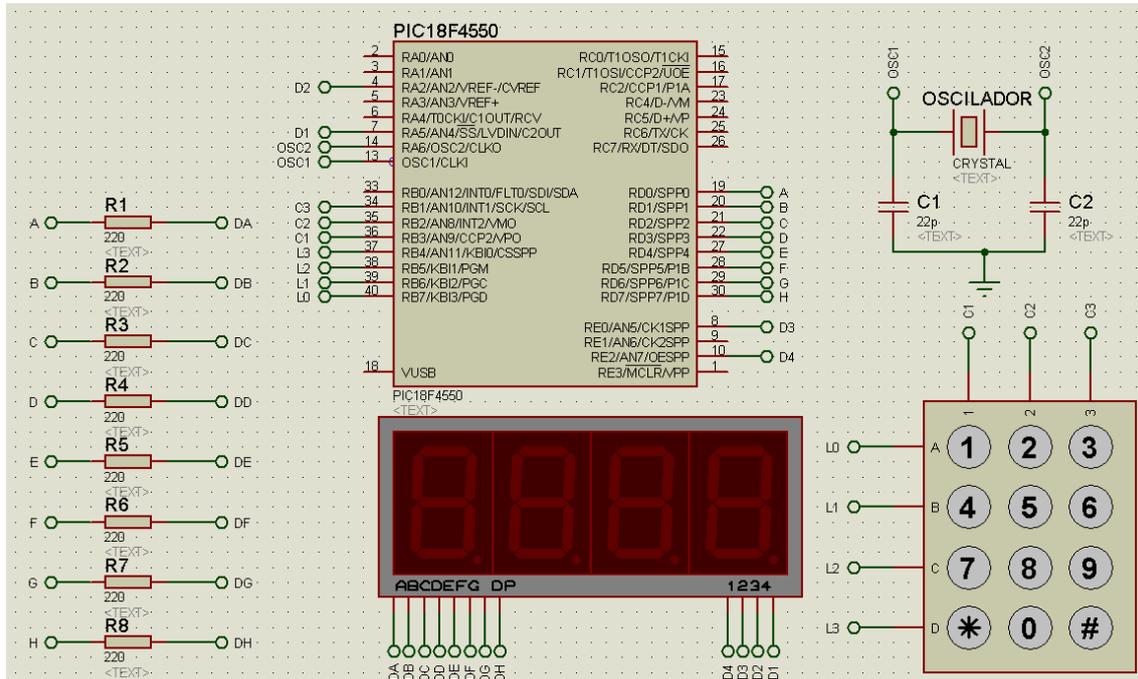


Figura 8 – Ambiente de simulação do *Proteus (Isis)*

2.7. O Tutorial

O uso das ferramentas descritas neste trabalho possibilitou o desenvolvimento de material didático com o propósito de facilitar aos monitores e estudantes de disciplinas relacionadas a Sistemas Embarcados e microcontroladores novos métodos de estudos, bem como aqueles que se interessar em fazer uso desse material para iniciar seus estudos em projetos de pesquisa ou desenvolvimento.

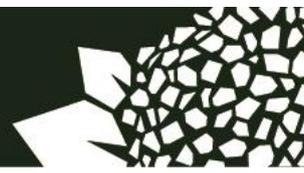
No tutorial é descrito como são integrados o *Proteus*, o *MPLAB* e o *Doxigen* à monitoria. Diversas práticas são montadas, com o circuito montado no *Proteus*, o projeto pronto no *MPLAB* e os códigos documentados pelo *Doxygen*.

Além disso, é mostrado como identificar os periféricos dentro do kit, como eles foram desenhados no *Proteus* de acordo com a aplicação, uso dos recursos da arquitetura como interrupções, PWM e conversores analógico-digitais, alguns códigos exemplos e a documentação desses códigos.

O tutorial é disponibilizado para os alunos via download, tanto pelo sítio do curso de engenharia de telecomunicações quanto pelo sítio do grupo de pesquisa GDESTe.

3. RESULTADOS E DISCUSSÃO

O trabalho desenvolvido até agora nos dá dimensão do que se pode alcançar com iniciativas interessantes. No programa de Monitoria não é ensinado apenas a programação para Sistemas Embarcados, mas sim a manusear equipamentos de bancada como osciloscópios, fontes, geradores de função, dentre outros. Isso se faz necessário vide o pouco tempo em sala com o professor, logo a monitoria proporciona uma jornada semanal de doze horas a mais em sala de aula focado no aprendizado e desenvolvimento de atividades.



Outro fator é o tempo destinado ao desenvolvimento de material didático, oito horas semanal, que reforçam as atividades passadas pelo professor titular da disciplina. Esse fator ganha força quando se tem grupos de pesquisa envolvidos no auxílio do desenvolvimento deste material, focando, assim, numa maximização de esforços em prol de um ensino melhor.

4. CONCLUSÃO

Ao final desse trabalho, destacamos o empenho dos envolvidos no processo, professores, monitores e alunos para realizar um trabalho com qualidade para ser aplicado em três turmas de Sistemas Embarcados em diferentes cursos superiores do IFCE e duas turmas de Microcontroladores também nessa mesma realidade. Esses esforços se dividem em elaboração dos esquemáticos no Proteus, elaboração dos códigos exemplos, teste dos códigos nos kits e documentação destes no *Doxygen* para disponibilizar para os alunos.

Como trabalhos futuros, pretende-se estender esse modelo a arquitetura ARM CORTEX-M3, ARM9 e ARM11, AVR ATMEGA16 e FPGA, com foco na descrição de circuitos digitais em VHDL.

REFERÊNCIAS BIBLIOGRÁFICAS

DOXYGEN. **Overview**. Disponível em:

<<http://www.stack.nl/~dimitri/doxygen/manual/index.html>> Acesso em: 22 jun. 2013.

LABCENTER ELETRONICS. **The VSM Advantage**. Disponível em:

<http://www.labcenter.com/products/vsm/vsm_overview.cfm> Acesso em: 22 jun. 2013.

MIYADAIRA, Alberto Noboru. Microcontroladores PIC18 Aprenda e Programe em Linguagem C. 1. ed. São Paulo: Editora Érica Ltda, 2010. 400 p, il.

OLIVEIRA, A. S; Andrade, F. S. Sistemas Embarcados Hardware e Firmaware na Prática. 1. ed. São Paulo: Erika, 2006. 315p, il.

ORDONEZ, E. D. M; PENTEADO, C. G; SILVA, A. C. R. Microcontroladores e FPGAs Aplicações em Automação. 1. ed. São Paulo: Novatec, 2005. 378p, il.

PEREIRA, Fábio. Microcontrolador PIC 18 Detalhado. 1. Ed. São Paulo: Editora Érica Ltda, 2010. 304 p, il.

SOUZA, Daniel Rodrigues de; SOUZA, David José de; LANINIA, Nicolás César. Desbravando o Microcontrolador PIC18 - Recursos Avançados. ed. São Paulo: Editora Érica Ltda, 2010. 336 p, il.

ZANCO, Wagner da Silva. Microcontroladores PIC18 com Linguagem C uma abordagem prática e objetiva com base no PIC18F4520. 1. ed. São Paulo: Editora Érica Ltda, 2010. 448 p, il.



TUTORIAL PIC18F APPLIED TO MONITORING EMBEDDED SYSTEMS OF IFCE

***Abstract:** Join the contents seen and classroom with the practice has been a challenge. This paper aims to show a way to give engineering students, with a focus on embedded systems, a learning tool alternative to that seen in class, presenting a tutorial monitoring program developed in the disciplines of Embedded Systems and Microcontrollers and the Group development of Telecommunication Systems and Embedded Systems, Federal Institute of Education, Science and Technology of Ceará.*

***Key-words:** Tutorial, Monitoring, Microcontrollers, Embedded Systems.*