

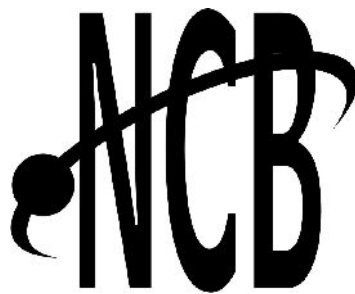
VIDAL PEREIRA

MICROCONTROLADORES PIC 16F E 18F TEORIA E PRÁTICA



NCB

MICROCONTROLADORES PIC 16F E 18F TEORIA E PRÁTICA



Instituto NCB

www.newtoncbraga.com.br
contato@newtoncbraga.com.br

Microcontroladores PIC 16F e 18F – Teoria e Prática

Autor: Vidal Pereira da Silva Jr.

São Paulo - Brasil - 2013

Palavras-chave: Eletrônica - Engenharia Eletrônica - Componentes – Microcontroladores

Copyright by
INSTITUTO NEWTON C BRAGA.
1ª edição

Todos os direitos reservados. Proibida a reprodução total ou parcial, por qualquer meio ou processo, especialmente por sistemas gráficos, microfílmicos, fotográficos, reprográficos, fonográficos, videográficos, atualmente existentes ou que venham a ser inventados. Vedada a memorização e/ou a recuperação total ou parcial em qualquer parte da obra em qualquer programa juscibernético atualmente em uso ou que venha a ser desenvolvido ou implantado no futuro. Essas proibições aplicam-se também às características gráficas da obra e à sua editoração. A violação dos direitos autorais é punível como crime (art. 184 e parágrafos, do Código Penal, cf. Lei nº 6.895, de 17/12/80) com pena de prisão e multa, conjuntamente com busca e apreensão e indenização diversas (artigos 122, 123, 124, 126 da Lei nº 5.988, de 14/12/73, Lei dos Direitos Autorais).

Diretor responsável: Newton C. Braga

Diagramação e Coordenação: Renato Paiotti

Dedicatória

À minha esposa Giane
e as minhas filhas Isabella e Nathália,
que são minhas fontes de energia
para viver cada vez mais.

Avisos importantes

1) Sobre as informações aqui apresentadas e garantias de qualquer tipo:

O autor acredita que todas as informações aqui apresentadas estão corretas e podem ser utilizadas para qualquer fim legal. Entretanto, não existe qualquer garantia, explícita ou implícita, de que o uso de tais informações conduzirá ao resultado desejado.

2) Sobre dúvidas relativas ao assunto

A aquisição deste livro não implica no direito do leitor de obter atendimento pessoal sobre dúvidas ou outros questionamentos referentes ao assunto, bem como suporte no uso das ferramentas apresentadas, as quais são gratuitas ou versões de demonstração.

3) Sobre os arquivos para download que acompanha este livro

Os arquivos que acompanham este material têm todos os exemplos já digitados para uso, e mais alguns arquivos auxiliares como databooks de componentes, manuais de uso, entre outros.

Todos os programas são grátis (apenas o compilador C é uma versão demo) e podem ser baixados da internet no site de seus respectivos fornecedores.

Para copiá-los para seu computador e instalar os programas, veja o anexo I no final deste livro.

Para Baixar o compilador acesse: <http://www.ccsinfo.com>

Objetivo deste material

Permitir ao estudante aprender sobre o funcionamento dos microcontroladores PIC em geral, das famílias 16F e 18F, de forma rápida e simples.

Para tal usaremos a linguagem C, o que reduz bastante o tempo de aprendizado e permitindo fácil transição para outros modelos de pic's.

Metodologia

Este material de estudo está dividido em 8 capítulos, que o estudante deve acompanhar na ordem proposta, e 2 anexos auxiliares:

- I) Introdução aos microcontroladores e linguagens de programação
- II) A linguagem C básica com exercícios simples para fixação dos conceitos
- III) Programa mínimo em C para compilação e simulação dos exemplos
- IV) O ambiente de desenvolvimento e simulação 'Mplab' em C com os exemplos básicos dos capítulos II e III
- V) A arquitetura dos Pics e seus periféricos mais usuais
- VI) A linguagem C implementada pela CCS para os pic's, com detalhes

das diferenças entre as diversas famílias.

- VII) Arquivo de definições dos modelos usados nos exemplos: 18F458, 16F877 e 16F877A
- VIII) Exemplos práticos com esquemas e programas para estudo de alguns dos periféricos estudados (exemplos baseados no 16F877, 16F877A e 18F458)

Sobre a abordagem utilizada neste método de estudo

Desde 1988, ministrando os mais diversos tipos de treinamentos, posso assegurar que a abordagem tradicional usada pela maioria dos autores (geralmente indicada por editoras ou manuais de redação) transmite o assunto, mas de forma seqüencial, um tópico de cada vez.

No método que uso atualmente nos meus treinamentos, pude constatar que se transmitirmos os tópicos de forma paralela, isto é, se formos abordando uma parte de cada área, um pouco por vez, o estudante vai assimilando mais facilmente, pois consegue “enxergar”, passo a passo, o fim do túnel.

Em nosso caso, podemos dividir o treinamento em vários tópicos:

- A linguagem de programação 'C'
- O hardware do PIC
- O ambiente de desenvolvimento
- Esquemas elétricos dos exemplos

Para permitir que o estudante realmente assimile o conhecimento transmitido, vamos abordando todos os tópicos simultaneamente, permitindo ao aluno ir praticando desde o início do treinamento, sem ficar muito tempo apenas na teoria.

Desta forma, ao invés de transmitirmos o conhecimento, primeiro analisando somente o hardware, depois o software, depois as ferramentas e somente ao final os exemplos, vamos mesclando cada um dos tópicos aos poucos, e o aluno com certeza assimilará mais facilmente.

Espero que o leitor aprecie este trabalho, e tenha o melhor aproveitamento possível.

Vidal

Site do autor: www.vidal.com.br

Índice

I - Introdução aos microcontroladores e linguagens de programação.....	12
Os microcontroladores.....	12
A linguagem C	12
Método de estudo.....	12
II - A linguagem C básica.....	14
II.1 - Iniciação à linguagem C.....	14
II.2 - Algumas regras comuns para a programação em 'C'.....	14
II.3 - Modelo básico de um programa em C.....	14
II.4 - Comentários.....	15
II.5 - Diretivas de compilação.....	16
II.6 - Indicador de fim de instrução.....	16
II.7 - Definição de variáveis, constantes e identificadores.....	17
II.7.1 – Sinalizadores de números negativos e positivos.....	17
II.7.2 - Seqüência de declaração de variáveis e constantes.....	18
II.7.3 - Atribuindo valores.....	18
II.7.4 – Atribuindo valores iniciais na declaração.....	18
II.7.5 – IMPORTANTE:	18
II.7.6 - Como escrever os nomes de variáveis, constantes e funções.....	19
II.7.7 – Typedef - Redefinindo tipos.....	19
II.8 - Funções e rotinas.....	20
II.8.1 - Funções especialmente desenvolvidas para os PIC's.....	20
II.9 - Expressões numéricas e de string (caracteres).....	21
II.10 - Operadores lógicos e aritméticos básicos da linguagem C.....	21
II.10.1 - Precedencia (ou prioridade) de operadores.....	23
II.10.2 - Conversão de tipos (type casting).....	23
II.11 - Matrizes.....	25
II.11.1 - Matrizes bidimensionais.....	26
II.12 - Controle do programa em C.....	26
II.12.1 - Blocos de declarações.....	26
II.12.2 - Bloco IF (executa se a condição for verdadeira).....	27
II.12.3 - Bloco FOR (executar por um certo número de vezes).....	30
II.12.4 - O condicional WHILE (enquanto).....	31
II.12.5 - O condicional DO . . . WHILE (faça ... enquanto).....	33
II.12.6 – O comando BREAK	34
II.12.7 – O comando CONTINUE.....	34
II.12.8 - O condicional SWITCH.....	35
II.12.9 - O comando RETURN.....	36
II.13 - Abreviações úteis para instruções aritméticas.....	37
II.13.1 - Incremento e Decremento.....	37
II.13.2 - Combinando abreviações.....	38
II.13.3 – Operações com resultado na mesma variável.....	38
II.14 - Variáveis locais, variáveis globais e parâmetros.....	38
II.14.1 - Variáveis Globais.....	38
II.14.2 - Variáveis Locais.....	38
II.14.3 - Variáveis como parâmetros.....	39
II.15 - A variável tipo VOID e os protótipos de funções.....	40
II.15.1 - Protótipos de funções.....	40
II.16 - Estruturas.....	41

II.17 - Unions.....	42
II.18 - A função MAIN ().....	43
II.19 - Exemplos de programas simples.....	43
III - Programa mínimo em C.....	46
IV - Usando o Mplab 7.62 em C.....	50
IV.1 - Conceitos básicos.....	50
IV.2 - O “Projeto” no MpLab.....	50
IV.3 - Criando o projeto com o Project Wizard.....	51
IV.4 - Simulando o programa.....	58
IV.5 – Verificando o registro PORTD durante a simulação.....	60
V - Os Microcontroladores PIC e seus periféricos mais usuais - Famílias 16F e 18F - Teoria de funcionamento.....	63
V.1 - Introdução.....	63
V.2 – Circuito mínimo.....	64
V.3 – Memória de Programa.....	64
V.4 – Memória de dados.....	65
V.5 – Memória EEPROM de dados.....	65
V.6 – Circuito de Reset e Clock.....	66
V.7 – Multiplicação 8 bits x 8 bits por hardware.....	66
V.8 – Interrupções	67
V.8.1 -Trabalhando com interrupções de alta e baixa prioridade.....	68
V.9 – Fusíveis de configuração.....	68
V.10 – O port A e suas funções especiais.....	69
V.10.1 – Algumas funções de acesso ao portA digital.....	69
V.11 – O port B e suas funções especiais.....	70
V.12 – O port C e suas funções especiais.....	70
V.13 – Os ports D e E com suas funções especiais.....	71
V.14 – Interrupções externas.....	71
V.15 – Timer 0.....	72
V.16 – Timer 1.....	73
V.16.1 – Funções para controle do timer 1.....	73
V.17 – Timer 2.....	73
V.18 – Timer 3 - Apenas na linha 18F.....	74
V.19 – O conversor A/D.....	74
V.20 – A comunicação serial Assíncrona.....	76
V.20.1 – Funções para controle da comunicação serial.....	77
V.21 – Tipos de osciladores.....	77
V.22 – O Watch Dog.....	78
V.22.1 - O Watch Dog da família 16F.....	78
V.22.2 - O Watch Dog do pic 18F458.....	79
V.23 – Brown-out Reset.....	80
V.24 – O modo Power-Down, ou ‘Sleep’.....	80
V.25 – Power-up Timer.....	81
V.26 – Oscilator Start-up Timer.....	81
V.27 – Módulo MSSP como SPI.....	82
V.28 – Módulo MSSP como I2C.....	82
V.29 – Módulo CCP como ‘Capture Mode’.....	83
V.30 – Módulo CCP em ‘Compare mode’.....	83
V.31 – Módulo CCP em ‘Pwm Mode’.....	83
V.32 – Low voltage Detector (LVD) -	84

V.33 – Comparador analógico (linha 16F87xA e 18F458)	85
V.34 – Interfaces CAN, USB, ETHERNET,..... -	86
VI - A linguagem C e os PIC's	87
VI.1 - Diretivas de compilação	87
VI.1.1 - #asm #endasm	87
VI.1.2 - #case	87
VI.1.3 - #define 'nome' 'seqüência'	88
VI.1.4 - #include <arquivo>	88
VI.1.5 - #fuses 'opções'	88
VI.1.6 - #ifdef #endif	88
VI.1.7 - #INT #####, onde ##### indica o nome da rotina	89
VI.1.8 - #Priority – Prioridades por software	89
VI.1.9 - #ROM	90
VI.1.10 - #use delay (clock= 'valor do clock em Hz')	90
VI.1.11 - #use Fast_IO(port)	90
VI.1.12 - #use standard_IO(port) - Default	90
VI.1.13 - #use rs232 (BAUD = taxa, XMIT = pinoTx, RCV = pinoRx, BITS = n)	91
VI.1.14 - #byte nome = endereço do byte	91
#bit nome = endereço do byte . número do bit	91
VI.2 - Funções escritas para os PIC's	91
VI.2.1 - Funções matemáticas	92
VI.2.2 - Funções de manipulação de bit	92
VI.2.3 - Funções de tempo	93
VI.2.4 - Funções para interrupções	93
VI.2.5 - Funções para o canal A/D	94
VI.2.6 - Funções para EEPROM de dados interna	95
VI.2.7 - Funções do Timer 0 (São similares para os demais timers)	95
VI.2.8 - Funções do canal de comunicação serial	96
VI.2.9 - Funções de uso geral	97
VII - Os arquivos de definições dos PICs	98
VIII – Exemplos práticos dos principais periféricos estudados no cap V	99
VIII.1 – Introdução	99
VIII.2 – Usando displays de cristal liquido como auxilio de	99
VIII.2.1 - O display LCD 16X2	99
VIII.3 – Exemplos do Capítulo II	101
VIII.4 – Exemplo dos Capítulo III	101
VIII.5 – Exemplos práticos para treinamento	101
VIII.5.1 – Usando saídas digitais	102
VIII.5.2 – Usando entradas e saídas digitais	103
VIII.5.3 – Usando o conversor de analógico para digital com resolução de 8 bits	103
VIII.5.4 – Display LCD 16x2 com interface de 4 bits	105
VIII.5.5 - Usando o conversor de analógico para digital com resolução de 10 bits e visualização dos resultados no LCD	105
VIII.5.6 – Usando a EEPROM de dados	106
VIII.5.7 – Usando o timer 0 em 8 bits com clock externo	107
VIII.5.8 – Usando o timer 1 (16 bits) com clock interno	107
VIII.5.9 – Usando a interrupção externa INTO no pino RB0	108
VIII.5.10 – Enviando um byte pelo canal serial	109
VIII.5.11 – Recebendo um byte pelo canal serial	110
VIII.5.12 – Comunicação serial SPI por hardware	111

VIII.5.13 – Comunicação serial I2C.....	111
VIII.5.14 – CCP em modo “PWM”.....	112
VIII.5.15 – Multiplexação de displays de 7 segmentos.....	113
VIII.5.16 – Teclado matricial 3x4.....	114
VIII.5.17 – Comunicação serial SPI por software.....	114
Anexos.....	116

I - Introdução aos microcontroladores e linguagens de programação

Os microcontroladores

Desde meu primeiro livro, em 1.988, utilizo o termo “Microcomputador-de-um-só-chip” para definir os microcontroladores.

A principal característica do microcontrolador esta em reunir em um só chip todos os periféricos necessários para o projeto e fabricação de dispositivos eletrônicos dos mais diversos tipos, desde simples sinalizadores e luzes pisca-pisca até equipamentos médicos sofisticados.

Hoje temos microcontroladores de apenas 6 pinos, minúsculos, ideais para incluir inteligência em dispositivos mecânicos em geral (dispositivos mecatrônicos) e até chips com as mais de 80 pinos, com as mais variadas capacidades, diversos tipos de interfaces (USB, Ethernet, CAN, ...); conversores analógico-digitais, entre outros.

Para a sua empreitada neste mundo maravilhoso de projetos com microcontroladores, alguns pré-requisitos são necessários:

1. Conhecimentos de nível médio de [eletrônica analógica](#) e [digital](#);
2. Facilidade de uso de computadores da linha PC baseados em Windows, para edição de textos e manipulação de arquivos;
3. Prática de [montagens eletrônicas](#) para a parte experimental;
4. Noções de programação em qualquer linguagem ou conhecimentos de lógica de programação.

A linguagem C

Neste treinamento utilizaremos a linguagem C para programar os PIC's, e não o assembler.

A principal vantagem do uso da linguagem C esta no fato de que a crescente complexidade dos microcontroladores vai tornando a programação em assembler cada vez mais difícil, dificultando para o projetista a mudança de modelos, como, por exemplo, a migração, na linha microchip, da linha 16F para a linha 18F.

Com o compilador C, as constantes mudanças de arquitetura interna do chip, das instruções, e dos algoritmos de desenvolvimento de software, passa a ser muito mais simples, pois a recompilação de cada rotina ou função especial (por exemplo, a programação dos registros internos para uso do conversor analógico-digital), que com certeza é diferente nas linhas 16F e 18F, passa a ser transparente para o desenvolvedor.

Desta forma, ao invés de consumir tempo reescrevendo rotinas ou todo um programa, o projetista apenas vai revisar as funções do seu programa para ajustar-se aos periféricos do novo modelo, dedicando seu tempo a escrever as funções baseadas em lógica, e não perdendo-se em detalhes de bits e bancos de memória.

Método de estudo

Neste treinamento não vamos estudar na forma tradicional.

Vamos primeiro ver a linguagem C básica, sem se preocupar com sua aplicação nos pics, apenas analisando os detalhes para compilar os programas mínimos apresentados.

Em seguida, veremos o ambiente de desenvolvimento Mplab, da Microchip, e o compilador C da CCS (versão demo), numa apresentação gráfica de como se usa a ferramenta.

Depois estudaremos os principais periféricos das famílias 16F e 18F, baseados nos modelos 16F877, 17F877A e 18F458.

Após este estudo voltaremos ao estudo da linguagem C, agora já detalhando os comandos desenvolvidos para os PIC's.

Nesta parte do treinamento dedicaremos nosso maior tempo, pois já poderemos estudar e utilizar simultaneamente o pic, o compilador C, o ambiente Mplab em compilação e sobretudo em simulação.

Teremos então vários exemplos completos, com esquema e software detalhado.

Para enriquecer ainda mais este trabalho, temos arquivos para download com todos os programas já digitados e prontos para serem executados, com a versão demo do compilador PCWH, com o Mplab e com o software de gravação IC-PROG, e os esquemas individuais de cada experiência e de um gravador para os pic's.

II - A linguagem C básica

II.1 - Iniciação à linguagem C

A principal vantagem de se usar linguagens de alto nível (no nosso caso a linguagem C) esta na menor interação do projetista com o hardware, no que diz respeito ao controle do mesmo (ajuste de bancos de registradores, seqüências de inicialização, etc...).

Como exemplo vamos ver a seqüência de escrita na EEPROM de dados do 18F458 em assembler:

```
MOVLW      EE_ADDRESS
MOVWF      EEADR
MOVLW      EE_DATA
MOVWF      EEDATA
BCF        EECON1, EEPGD
BCF        EECON1, CFGS
BSF        EECON1, WREN
BCF        INTCON, GIE
MOVLW      55h
MOVWF      EECON2
MOVLW      0AAh
MOVWF      EECON2
BSF        EECON1, WR
BSF        INTCON, GIE
```

e em C:

```
write_eeprom ( EE_ADDRESS , EE_DATA );
```

II.2 - Algumas regras comuns para a programação em 'C'

- Use letras minúsculas em todo o programa (recomendado)
- Todas as funções e variáveis devem ser declaradas
- Palavras reservadas não podem ser usadas como identificadores de variáveis e funções
- Sempre escreva comentários em seus programas

II.3 - Modelo básico de um programa em C

Quatro elementos estão presentes em um programa C:

```
Comentários
Diretivas de compilação
Definições de dados
Blocos com instruções e funções
```

Modelo básico (desconsiderando ser um programa para pic's):

```

#include <...>                (Diretivas de compilação)

// comentário ocupando uma linha

/* comentários entre ‘/ *’ e ‘* /’      podem ocupar mais de
uma linha */

int8 i, j;                    (Variáveis de 8 bits)
float Tempo;                  (Variável de ponto flutuante)

void main()
{
    instruções do programa principal
}

void delay()
{
    instruções da função (rotina) delay
}

```

II.4 - Comentários

Comentários são informações que você, durante a escrita do programa fonte (*), vai inserindo e que permitem a você (programador) e a outros entenderem o significado do que esta sendo feito.

É boa prática comentar o máximo possível de linhas, e até mesmo incluir grandes blocos de comentários, explicando o porque do que esta sendo feito, pois após um certo tempo, nem mesmo o criador do programa lembrará de tudo o que estava pensando no momento da escrita.

O compilador ignora tudo que estiver definido como comentário.

(*) O programa fonte em C deve ter terminação “.C”, por exemplo: teste.c

Existem dois tipos de comentários:

- Comentários que ocupam apenas 1 linha

Este tipo de comentário é iniciado com duas barras conjuntas: //

Neste caso, tudo que estiver após as duas barras será ignorado pelo compilador, até o final da linha.

Exemplo:

```
x = x + 2;           // soma 2 à variável x
```