



UNIVERSIDADE DO ESTADO DE MINAS GERAIS

Docente: Rildo Afonso de Almeida

Microprocessadores & Microcontroladores



5 - Programação C

5.1 Linguagem de Alto Nível

“C” é uma linguagem de programação de alto nível largamente utilizada e com uma extensa e diversificada literatura sobre a mesma.

5 - Programação C

5.1 Linguagem de Alto Nível

Uma das principais características da linguagem C é a sua operação através de funções que podem ser chamadas a partir de uma função principal, denominada **main()**. Ela deve existir em qualquer programa escrito nessa linguagem (uso obrigatório).

5 - Programação C

5.1 Linguagem de Alto Nível

O uso de linguagens de alto nível na programação de microcontroladores se deve à grande complexidade dos programas escritos em Assembly; o volume de código é muito maior em Assembly do que em linguagens de alto nível, o que torna muito difícil a depuração do mesmo quando escrito em Assembly



5 - Programação C

5.1 Linguagem de Alto Nível

De qualquer forma, é muito importante o conhecimento do *hardware interno* do PIC, bem como dos seus registradores e funções especiais, para que se tire o máximo proveito dos mesmos com a programação em linguagem C.

5 - Programação C

5.2 Principais estruturas da linguagem C

A linguagem C, como a maioria das linguagens de programação, possui algumas estruturas necessárias para se escrever programas. Estudaremos as principais.

5 - Programação C

5.2 Principais estruturas da linguagem C

5.2.1 Variáveis, atribuições e comparações

Em C, há os seguintes tipos básicos de variáveis:

- **Char** – guarda um caractere.
- **Int** – guarda um número inteiro.

5 - Programação C

5.2 Principais estruturas da linguagem C

5.2.1 Variáveis, atribuições e comparações

- **Float** – guarda um número real com precisão simples.
- **Double** – guarda um número real com precisão dupla.
- **Void** – tipo vazio.

5 - Programação C

5.2 Principais estruturas da linguagem C

5.2.1 Variáveis, atribuições e comparações

A declaração de variáveis e atribuição de valores para as mesmas é feita como a seguir:

```
int evento;  
char corrida;  
float tempo;  
evento = 5;  
corrida = 'C';  
tempo = 27.25;
```

5 - Programação C

5.2 Principais estruturas da linguagem C

5.2.1 Variáveis, atribuições e comparações

Para efetuar comparações, utilizam-se os seguintes operadores:

- > maior que.
- >= maior ou igual.
- < menor que.



5 - Programação C

5.2 Principais estruturas da linguagem C

5.2.1 Variáveis, atribuições e comparações

Para efetuar comparações, utilizam-se os seguintes operadores:

- `<=` menor ou igual.
- `==` igual.
- `!=` diferente.

5 - Programação C

5.2 Principais estruturas da linguagem C

5.2.2 Estruturas de controle

Segundo Pereira (2003), a principal estrutura de controle é o comando “*if*” (*Se*), o qual testa se uma condição é verdadeira e, então, executa um bloco de comandos.

5 - Programação C

5.2 Principais estruturas da linguagem C

5.2.2 Estruturas de controle

Caso a condição do “*if*” não seja verdadeira, é possível executar um outro bloco de comandos com o comando “**else**” (**Alem disso**), como no exemplo a seguir:

5 - Programação C

5.2 Principais estruturas da linguagem C

5.2.2 Estruturas de controle

```
void main(void){  
int a;  
a = 5;  
if (a > 0){  
printf("a' é positivo");  
} else if (a == 0){  
printf("a' é nulo");  
} else {  
printf("a' é negativo");  
}}  

```



5 - Programação C

5.2 Principais estruturas da linguagem C

5.2.2 Estruturas de controle

Printf

A linguagem C define essa função que podem ser usada para escrever e ler valores.

Os valores são escritos em um chamado “fluxo de saída” que normalmente corresponde a um dispositivo referido como tela ou console.

Os valores são lidos de um chamado “fluxo de entrada” que normalmente corresponde a um dispositivo referido como teclado.

5 - Programação C

5.2 Principais estruturas da linguagem C

5.2.2 Estruturas de controle

O sinal de abertura de chave “{” é utilizado para delimitar o início da função e o sinal de fechamento da chave “}” indica o final da função.

Na realidade, as chaves delimitam o que chamamos de bloco de programa, ou bloco de código.



5 - Programação C

5.2 Principais estruturas da linguagem C

5.2.3 Estruturas de Repetição

Em C, há as seguintes estruturas de repetição:

For (Para)

While (Enquanto)

Do While (“Faça - Enquanto”)

5 - Programação C

5.2 Principais estruturas da linguagem C

5.2.3 Estruturas de Repetição

For

O laço **for** (“Para”) é basicamente utilizada para laços finitos de contagem, normamente utilizando uma variável de controle de contagem.

5 - Programação C

5.2 Principais estruturas da linguagem C

5.2.3 Estruturas de Repetição

Em C, há as seguintes estruturas de repetição:

```
For (variável = valor_inicial; condicao_envolvendo_variavel;  
incremento/decremento de var)  
{  
comandos...  
}
```



5 - Programação C

5.2 Principais estruturas da linguagem C

5.2.3 Estruturas de Repetição

For (variável = valor_inicial; condicao_envolvendo_variavel;
incremento/decremento de var)

Exemplo

```
for(int i=0; i <= 10; i++) {  
    printf("i=",i);  
}
```

5 - Programação C

5.2 Principais estruturas da linguagem C

5.2.3 Estruturas de Repetição

While

O comando **While** (“Enquanto”) é utilizada para repetição de um determinado conjunto de instruções enquanto uma condição for verdadeira.



5 - Programação C

5.2 Principais estruturas da linguagem C

5.2.3 Estruturas de Repetição

While (condição)

```
{
```

```
comandos...
```

```
/*Lembre-se que os comandos devem, de alguma  
forma, alterar condição*/
```

```
}
```

5 - Programação C

5.2 Principais estruturas da linguagem C

5.2.3 Estruturas de Repetição

Exemplo

```
int i=1;
while(i <= 10){
    printf("i=",i);
    i++;
}
```

5 - Programação C

5.2 Principais estruturas da linguagem C

5.2.3 Estruturas de Repetição

Do-While

O comando **Do-While** (“Faça - Enquanto”) é muito similar à estrutura **while**, com a diferença de que aqui a condição é analisada no final do bloco de comando.



5 - Programação C

5.2 Principais estruturas da linguagem C

5.2.3 Estruturas de Repetição

```
do
{
comandos...
/*Lembre-se que os comandos devem, de alguma
forma, alterar condição*/
} while (condição);
```



5 - Programação C

5.2 Principais estruturas da linguagem C

5.2.3 Estruturas de Repetição

Exemplo

```
int i=1;  
do{  
    printf("i=",i);  
    i++;  
}while(i <= 10);
```



Bibliografia Básica

MIYADAIRA, A. N. *Microcontroladores PIC18: aprenda e programe em Linguagem C* Ed. Érica, 1ª Ed., 2009, São Paulo.

LUZ, C. E. S. *Programando Microcontroladores PIC em Linguagem C com base no PIC4520*. Ed. Ensino Profissional, 2011, São Paulo.

SILVA, R. A. *Programando Microcontroladores PIC*. Ed. Ensino Profissional, 2011, São Paulo.

PEREIRA, F. *Microcontroladores PIC – Programação em C*. Érica: São Paulo, 2003.



Bibliografia Complementar

TAUB, H.. Circuitos Digitais e Microprocessadores. McGraw Hill do Brasil, 1984.

ZILLER, Roberto M. *Microprocessadores: Conceitos Importantes*. Edição do Autor, 2000.

DALTRINI, Beatriz M., JINO, M., MAGALHÃES, L. P.. *Introdução à Computação Digital*. Makron Books, 1999.