



INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
CEARÁ

# Curso de Microcontroladores PIC

## Capítulo 4

*Prof. Joacillo Luz Dantas*

Departamento de Telemática – IFCE

# Modificadores de Tipo

- Além dos tipos de dados, pode-se utilizar
- Signed, unsigned, short e long.
- Signed pode representar números positivos e negativos.
  - a) signed int pode representar valores de -128 a 127.
  - b) int pode representar valores de 0 a 255.
- Long é utilizado para ampliar a magnitude de representação do tipo especificado.
  - a) long int terá tamanho de 16 bits
  - b) signed long int representa valores de -32768 a 32767.

# Declaração de Variáveis

- Tipo nome
- Int tempo;
- Uma variável pode ser declarada em pontos do programa
  - a) No corpo principal, fora de qualquer função. Nesse caso são chamadas variáveis globais. Podem ser acessadas de qualquer parte do programa.
  - b) Dentro de uma função. Nesse caso são chamadas de variáveis locais. Só podem ser acessadas dentro da função na qual foram declaradas.
  - c) Como parâmetro de uma função. São um tipo especial de variáveis locais. É um valor entregue a função para a execução de alguma tarefa.

# Constantes

- Constantes são valores fixos , numéricos ou não, e que não podem ser alterados pelo programa durante a execução.
- Qualquer tipo de dados pode ser utilizado como constante:
  - a) `Const val = 10`
  - b) `Const pi = 3,14`
  - c) `Const x = 'a'`
  - d) `Const t = "teste"`
- Valores decimais, binários , octal e hexadecimal
  - a) Decimal     99
  - b) Binario     0b01100011
  - c) Octal       0143
  - d) Hexadecimal 0x63

# Operadores (dados)

- Atribuição

- a)  $Y=10;$

- b)  $Y = x;$

- Aritméticos

Operador	Ação
+	Adição
-	Subtração
*	Multiplicação
/	Divisão
%	Resto de divisão
++	Incremento
--	Decremento

# Operadores ( dados)

- Relacionais: Utilizados para testes condicionais

Operador	Ação
>	Maior que
>=	Maior ou igual
<	Menor
<=	Menor ou igual
==	Igual
!=	Diferente

# Operadores ( dados)

- Lógicos ou booleanos: O resultado da operação é Verdadeiro ou Falso

Operadores	Ação
&&	And
	OU
!	NOT

- `X=10;`
- `If(x>5 && x<20) y = x;`

# Lógico bit a bit

Operador	Ação
&	AND
	OU
^	XOR
>>	Deslocamento a direita
<<	Deslocamento a esquerda
~	NOT

- ❑ **Int a,b;**
- ❑ **a=0b01100101**
- ❑ **b=a&0b00001111**
- ❑ **Ou seja b=0b00000101**



# Testes condicionais

## □ Comando IF

- If(condição) comando;  
Else comandoB;

- If (condição)  
{ comando1;  
.  
.  
comandoN; }

- Else { comandoA;  
.  
.  
comandoX}

# Testes Condicionais

## □ Comando Switch

- Switch( variável)  
  { case constante 1: comando1;  
                                  ...  
                                  break;  
  case constante 2 : comandoA;  
                                  ....  
                                  break;  
                                  .....  
  default: comandoZ;  
                                  ...  
  }

# Estrutura de Repetição

- Laço While: executa a instrução ou o bloco de instruções enquanto a condições de teste for verdadeira.
- While ( teste) instrução;
- While( teste)
- { grupo de instruções  
}
- While(1) ou while(true) : loop infinito

# Estrutura de Repetição

- O laço for : executa uma instrução ou bloco de instruções repetidamente
  - For( inicialização; condição do teste; incremento,decremento ou ajuste) instrução;
  - For( inicialização; condição do teste; incremento ou decremento)  
    { conjunto de instruções  
    }
- For(;;) : loop infinito
  - For(;;) instrução;
  - For(;;)  
    { conjunto de instruções  
    }( inicialização; condição do teste; incremento,decremento ou ajuste) instrução;

# Incremento e Decremento

- ++ : Incremento de 1
  - `Tempo = tempo + 1;` é igual a `tempo++;`
- -- : Decremento de 1
  - `Tempo = tempo - 1;` é igual a `tempo--;`
- ++ : Incremento de 1
  - `Tempo = tempo + 1` é igual a `tempo++`