



altus



**Ponto**



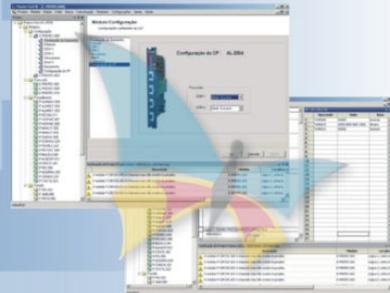
**Grano**



**Inversores de  
Frequência**



**FBs**



**MasterTool XE**



**Hadron**



**IHMs**

# Tutorial 119 Representação e operações com variáveis em ponto fixo e flutuante

Este documento é propriedade da ALTUS Sistemas de Informática S.A., não podendo ser reproduzido sem seu prévio consentimento.

# Sumário

---

<b>1</b>	<b>Descrição da Aplicação .....</b>	<b>3</b>
<b>2</b>	<b>Sistemas numéricos .....</b>	<b>3</b>
2.1	Número Decimal .....	3
2.2	Número Binário .....	3
2.3	Número Hexadecimal .....	3
<b>3</b>	<b>Conceitos Lógicos .....</b>	<b>4</b>
3.1	Bit (ponto).....	4
3.2	Nibble (quarteto).....	4
3.3	Byte (octeto).....	4
3.4	Word (palavra).....	4
3.5	Double Word (palavra dupla).....	5
<b>4</b>	<b>Descrição de variáveis ponto fixo.....</b>	<b>5</b>
4.1	Matemática em complemento de 2 .....	5
4.2	Faixas de representação .....	6
4.3	Ponto fixo com fração .....	6
<b>5</b>	<b>Descrição de variáveis ponto flutuante .....</b>	<b>6</b>
5.1	Faixas de representação .....	7
5.2	Limitações na utilização de ponto flutuante.....	7
5.3	Operações com ponto flutuante .....	7
5.4	Restrições de usos nas UCPs .....	8
<b>6</b>	<b>Referências.....</b>	<b>8</b>
<b>7</b>	<b>Revisões .....</b>	<b>8</b>

## 1 Descrição da Aplicação

---

Este documento descreve a representação e operação com variáveis em ponto fixo e flutuante.

As representações numéricas em sistemas computacionais, como é o caso dos CLPs, não é feita da mesma maneira que na matemática discreta. Os números nos CLPs são representados através de blocos de bits e a quantidade de números possíveis de serem representados depende desta quantidade de (bytes, words...).

A atividade de programação trabalha, basicamente, com duas classes principais de números: os números de ponto fixo e os números de ponto flutuante. Números de ponto fixo podem ser considerados números não fracionários (apesar de existirem representações de ponto fixo com frações) e números de ponto flutuante, que podem ser números fracionários e não fracionários.

## 2 Sistemas numéricos

---

Os CPs, como todos os computadores, somente conseguem manipular valores representados no sistema binário. Para dominar a programação dos CPs é fundamental o conhecimento dos vários sistemas numéricos.

### 2.1 Número Decimal

---

---

Dígitos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Base: 10

Cada algarismo, do número, é multiplicado pela potência de 10.

Exemplo:

$$n^{\circ} 456 = (4 \times 10^2) + (5 \times 10^1) + (6 \times 10^0) = 456_D$$

### 2.2 Número Binário

---

---

Dígitos: 0,1

Base: 2

Cada algarismo, do número, é multiplicado pela potência de 2

Exemplo:

$$n^{\circ} 10110 = (1 \times 2^4) + (0 \times 2^3) + (1 \times 2^2) + (1 \times 2^1) + (0 \times 2^0) = 22_D$$

### 2.3 Número Hexadecimal

---

---

Dígitos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F Base: 16

Cada algarismo, do número, é multiplicado pela potência de 16

Exemplo:

$$n^{\circ} 2B7 = (2 \times 16^2) + (11 \times 16^1) + (7 \times 16^0) = 695_D$$

### 3 Conceitos Lógicos

---

#### 3.1 Bit (ponto)

---

Bit (Binary DigiT) é a unidade para o sistema de numeração binário. Um bit é a unidade básica de informação e pode assumir o valor "0" ou "1".

Ex.: cada Entrada e/ou Saída digital é armazenada em um bit.

1 bit



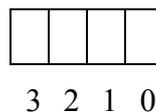
#### 3.2 Nibble (quarteto)

---

Nibble é a unidade formada por 4 bits consecutivos e pode assumir valores na faixa de 0 a 15.

Ex.: o nibble é a unidade básica dos números BCD.

1 nibble = 4 bits



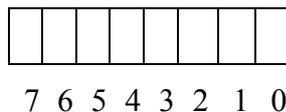
#### 3.3 Byte (octeto)

---

Byte é uma unidade constituída de 8 bits consecutivos e podem assumir valores na faixa de 0 a 255.

Ex.: as Entradas e Saídas digitais são agrupadas dentro de um byte.

1 byte = 2 nibbles = 8 bits



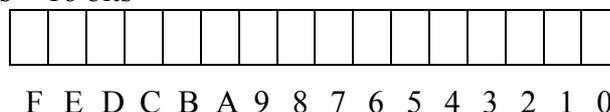
#### 3.4 Word (palavra)

---

Uma palavra é um conjunto de 2 bytes consecutivos e podem assumir valores na faixa de -32768 a +32767.

Ex.: valores numéricos, resultados de operações aritméticas, contagens, temporizações, ou entradas e saídas analógicas são armazenados em words.

1 word = 2 bytes = 4 nibbles = 16 bits





## 4.2 Faixas de representação

Na Tabela 4-1 são apresentados alguns formatos de tipos inteiros. Os tipos com a letra U na frente são números sem sinal, ou seja, não existe representação de números negativos, não possuindo sinal de magnitude. Os tipos sem a letra U possuem sinal de magnitude e representam uma faixa com números negativos e positivos.

	Quantidade de bits	Faixa
SINT	8	-128 a +127
INT	16	-32768 a 32767
DINT	32	-2E31 a (2E31 - 1)
LINT	64	-2E63 a (2E63 - 1)
USINT	8	0 a 255
UINT	16	0 a 65535
UDINT	32	0 a (2E32 - 1)
ULINT	64	0 a (2E64 - 1)

**Tabela 4-1. Quantidade de módulos de programa suportado**

Os formatos apresentados são os estabelecidos pela norma IEC61131.

## 4.3 Ponto fixo com fração

Alguns sistemas utilizam o ponto fixo para representar números fracionários, ou seja, os bits divididos em um grupo que representa a parte inteira e outra que representa a parte fracionária. Este formato não amplia a quantidade de números que pode ser representada, apenas permite uma representação fracionária com uma precisão igual ao número de bits disponíveis para a parte fracionária. A precisão é sempre fixa e não depende da magnitude da parte inteira do número representado. Esta representação é utilizada em alguns dispositivos eletrônicos, porém não é utilizado para aplicações de usuário em CLPs.

## 5 Descrição de variáveis ponto flutuante

A notação de ponto flutuante é utilizada para representar números fracionários de qualquer sinal. Outra vantagem da representação em ponto flutuante é que o intervalo de representação é bem maior. Na comparação com números inteiros de ponto fixo, se ambos possuírem a mesma quantidade de bits, a representação em ponto flutuante permitira representar uma faixa muito maior de números.

Números fracionários podem ser decompostos em duas partes: a parte inteira e a parte fracionária. O elemento que define a separação entre as partes é a vírgula. Na notação em ponto flutuante um número é dividido em três partes: a mantissa, o expoente e a base. A mantissa expressa o valor do número, a base indica em qual sistema de numeração ele é representado (binário, decimal, etc) e o expoente define a posição da vírgula. Por exemplo:  $1255,5E-1 = 12555 =$  mantissa;  $10 =$  base;  $-1 =$  expoente. Note que o expoente -1 determina que a vírgula seja levada uma casa para a esquerda.

Existem representações de ponto flutuante de 32 e 64 bits. No caso 32bits ocupam quatro bytes de memória (32 bits), armazenando o valor conforme a figura a seguir:



S – bit de sinal aritmético (0 – positivo, 1 – negativo)

O valor decimal de um ponto flutuante 32 bits é obtido pela seguinte expressão:

$$Valor = (-1)^S \times 2^{(Expoente-127)} \times 1, Mantissa$$

No caso de 64 bits que ocupam oito bytes de memória, o valor armazenando em 1 bit de sinal, 11 bits de expoente e 52 bits de mantissa.

## 5.1 Faixas de representação

A Tabela 5-1 apresenta a quantidade de bits, faixa de representação e precisão de dois tipos utilizando ponto flutuante.

	Quantidade de bits	Faixa	Precisão
REAL	32	$\pm 10E \pm 38$	1/2E23
LREAL	64	$\pm 10E \pm 308$	1/2E52

**Tabela 5-1. Quantidade de módulos de programa suportado**

Os formatos apresentados são os estabelecidos pela norma IEC61131. Ambos os tipos são os padrões da norma IEC 754. No caso do REAL ele é o que é definido pela norma como floating point single precision (ponto flutuante de precisão simples). Já o LREAL é definido pela norma como floating point double precision (ponto flutuante de precisão dupla)

## 5.2 Limitações na utilização de ponto flutuante

A definição de um maior número de bits na mantissa aumenta a precisão da fração representada, enquanto um maior número de bits de expoente aumenta a quantidade de números que podem ser representados. As representações de ponto flutuante nos CLPs apresentam um número fixo de bits para expoente e mantissa, conforme já apresentado em seções anteriores, definidos pela norma IEEE 754. Sendo assim devesse ter cuidado com os valores representados por um ponto flutuante e respeitar-se a precisão destes. Neste caso a precisão é função inversa da magnitude do número que está sendo representado. Para cada valor representado por um ponto flutuante a precisão deve ser calculada devendo-se utilizar esta informação para calcular se a perda de precisão irá impactar nas operações realizadas.

## 5.3 Operações com ponto flutuante

Tomando como exemplo um ponto flutuante de 32 bits, que possui uma mantissa de 23 bits é possível dizer que o número decimal máximo representável sem perda de precisão é um número de 23 bits. A partir deste ponto operações com magnitude inferior podem não impactar em alteração no resultado final da operação. Isso acontece, por exemplo, se estivermos trabalhando com um número de ordem de grandeza 10E35 e quisermos somar a ele um número com ordem de grandeza 10E1. O resultado da operação será o número original com maior ordem de grandeza, pois o outro possui um valor inferior a precisão do outro. Nestes casos é recomendado que se utilize para as operações variáveis de ponto fixo que trabalham com a precisão independente da magnitude. No caso de

necessidade de exibição em ponto flutuante esta operação deve ser feita no final da operação. Não é recomendada a utilização de ponto flutuante para operação de acumulação, como totalização de medições de grandezas físicas em que o valor máximo é maior que a precisão da variável utilizada.

## 5.4 Restrições de usos nas UCPs

---

Quando aplicável incluir quais os tipos de tipos inteiros não podem ser representados no tipo de UCP em questão. Ex: Os controladores DU350 e DU351 não suportam tipos inteiros de variáveis de 64 bits. Este tipo é o LREAL

Quando aplicável incluir quais os tipos de tipos inteiros não podem ser representados no tipo de UCP em questão. Ex: São eles: LWORD, LINT e ULINT.

## 6 Referências

---

- Para maiores informações, consultar os documentos:
  - Manual de Programação ST MasterTool Extended Edition MT8000 (MP399003);
  - Manual de Programação Ladder MasterTool Extended Edition MT8000 (MP399102);
  - Texto sobre ponto flutuante no Manual de Utilização PO3x47 UCPs Série Ponto;

## 7 Revisões

---

**Revisão:** A  
**Data:** 07/05/2009  
**Autoria:** Douglas Costa  
**Aprovação:** Marcelo Froener

**Natureza das alterações:**

- Primeira emissão do documento.