



**HI** tecnologia

**Indústria e Comércio Ltda**

---

## **Notas de Aplicação**

# **Utilização da função de troca de dados (bloco SWP) no ambiente SPDSW**

---

---

# **HI Tecnologia**

---

Documento de acesso público

---

**ENA.0031**

**Versão 1.00**

**novembro-2005**

---

## Apresentação

---

Esta nota de aplicação foi elaborada pela **HI Tecnologia Indústria e Comércio Ltda.** O departamento de suporte a clientes, está disponível, através do telefone **(19) 3252-6044** ou do e-mail [suporte@hitecnologia.com.br](mailto:suporte@hitecnologia.com.br), para esclarecimentos sobre as informações contidas neste documento, ou para dirimir quaisquer dúvidas a respeito de nossos produtos. Favor mencionar as informações a seguir para que possamos identificar os dados relativos a este documento.

ID da Nota de Aplicação: ENA.0031  
Versão Documento: 1.00

---

## HI Tecnologia Indústria e Comércio Ltda.

Endereço: Av. Dr. Armando de Sales Oliveira, 445

Cidade: Campinas – SP  
CEP: 13076-015

Fone: +55 (19) 3252-6044  
Fax: +55 (19) 3251-4253

E-mail: [hi@hitecnologia.com.br](mailto:hi@hitecnologia.com.br)

Web site: [www.hitecnologia.com.br](http://www.hitecnologia.com.br)

---



## Utilização da função de troca de dados (bloco SWP) no ambiente SPDSW

Referência: ENA.0031  
Arquivo : ENA0003100.doc

Revisão: 0  
Atualizado em: 29/11/2005

## Índice

|       |  |    |
|-------|--|----|
| 1.    | Introdução .....   | 4  |
| 2.    | Referências .....  | 4  |
| 3.    | Utilização do Bloco SWP para troca de dados .....                    | 4  |
| 3.1   | Funcionamento do Bloco SWP .....                                     | 4  |
| 3.2   | Seleção do Bloco SWP no SPDSW .....                                  | 6  |
| 4.    | Troca de dados .....   | 7  |
| 4.1   | Troca de dados entre operadores do mesmo tipo .....                  | 7  |
| 4.1.1 | Troca de dados entre memórias inteiras (tipo M) .....                | 7  |
| 4.1.2 | Troca de dados entre memórias reais (tipo D) .....                   | 8  |
| 4.2   | Troca de dados entre operadores diferentes .....                     | 9  |
| 4.2.1 | Troca de dados entre memórias inteira (tipo M) e real (tipo D) ..... | 9  |
| 4.2.2 | Troca de dados entre memórias real (tipo D) e inteira (tipo M) ..... | 10 |
| 5.    | Exemplo de utilização .....  | 11 |
|       | Controle do Documento .....  | 14 |
|       | Considerações gerais .....   | 14 |
|       | Responsabilidades pelo documento .....                               | 14 |



## Utilização da função de troca de dados (bloco SWP) no ambiente SPDSW

Tipo de Doc.: Notas de Aplicação  
Referência: ENA.0031

Revisão: 0  
Atualizado em: 29/11/2005

# 1. Introdução

---

Este documento destina-se a instruir os usuários dos controladores HI Tecnologia a utilizarem a função SWP (troca de dados), que permite trocar dados entre memórias (inteiras e reais) em suas aplicações. Esta troca de conteúdo entre memórias pode ser feita entre memórias de mesmo tipo (inteira com inteira ou real com real) ou entre tipos diferentes (inteira com real). Esta função está disponível para utilização no ambiente de programação SPDSW (versão 1.05.07 ou superior)

OBSERVAÇÃO: O aplicativo SPDSW encontra-se disponível para *download* gratuito em nosso *site*: [www.hitecnologia.com.br](http://www.hitecnologia.com.br)

# 2. Referências

---

Programas de Exemplo (em ambiente SPDSW):

- HIPJW024 – Utilização do bloco SWP para converter dados obtidos através de uma troca de dados via protocolo MODBUS-RTU com um medidor de grandezas elétricas.
- HIPJW025 – Utilização do bloco SWP com operações simples de troca de dados considerando variáveis do mesmo tipo e de diferentes tipos.

Todos os documentos referenciados acima estão disponíveis para *download* em nosso *site*: [www.hitecnologia.com.br](http://www.hitecnologia.com.br).

# 3. Utilização do Bloco SWP para troca de dados

---

Para acesso a função de troca de dados deve-se selecionar o bloco SWP no editor *ladder* do ambiente SPDSW. Vide item 3.2.

## 3.1 Funcionamento do Bloco SWP

---

O bloco SWP é composto por uma entrada E1, 2 parâmetros (P1 e P2) e uma saída S1, como ilustra a figura abaixo:

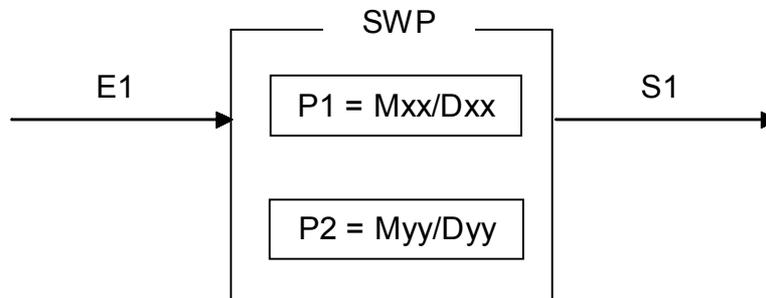


Figura 1 - Bloco "SWP"

Os elementos que compõem o bloco SWP são descritos abaixo:

- Parâmetro P1: Deve ser uma memória do tipo **Mxx** (inteira) ou **Dxx** (ponto flutuante) que indica o primeiro operador para a troca de dados.
- Parâmetro P2: Deve ser uma memória do tipo **Mxx** (inteira) ou **Dxx** (ponto flutuante) que indica o segundo operador para a troca de dados.
- Entrada E1: sinal de habilitação do bloco SWP, que quando:
  - Energizado - Bloco habilitado, executa troca de dados.
  - Desenergizado - Bloco desabilitado, não executa a função.
- Saída S1: cópia da entrada E1.

Seguem as seguintes considerações sobre a operação do bloco SWP:

1. Permite as seguintes operações de troca de dados:
  - a. Troca de dados entre duas memórias inteiras. Neste caso temos as seguintes configurações dos parâmetros : P1 = Mxx e P2 = Myy.
  - b. Troca de dados entre duas memórias de ponto flutuante. Neste caso temos as seguintes configurações dos parâmetros : P1 = Dxx e P2 = Dyy;
  - c. Troca de dados entre duas memórias inteiras e uma memória em ponto flutuante. Neste caso temos as seguintes configurações dos parâmetros : P1 = Mxx e P2 = Dyy ou então P1 = Dxx e P2 = Myy. Note-se que neste caso, utiliza-se duas memórias do tipo M para a operação de troca de dados, no caso, assume-se a memória subsequente a memória Mxx ou Myy especificada como parâmetro. Por exemplo, se especificadas as memórias M10 e D20 utilizam-se as memórias M10 e M11 para a execução da troca de dados com a memória D20.

- Note-se que este bloco SWP opera de acordo com o estado da entrada E1, e não apenas na transição de subida desta entrada de habilitação. Assim a cada ciclo do programa *ladder* em que se executa este bloco SWP, realiza-se a operação de troca de dados. Por exemplo, se executarmos este bloco SWP em dois ciclos consecutivos a operação de troca será inócua, pois no primeiro ciclo troca-se A com B, e no segundo ciclo de execução novamente troca-se A com B, retornando os valores originais de A e B. Assim, se desejar executar apenas uma troca simples de valores entre as variáveis A e B, deve-se habilitar a execução deste bloco SWP apenas em um ciclo de *scan*, por exemplo, utilizando um relé de pulso.

### 3.2 Seleção do Bloco SWP no SPDSW

Para inserir um bloco SWP em seu programa de aplicação *ladder* siga as instruções abaixo :

- Posicione o cursor no local desejado do programa *ladder*;
- Selecione a opção "**Movimento**" na palheta de comandos à esquerda da janela do editor *ladder*, conforme ilustrado abaixo.

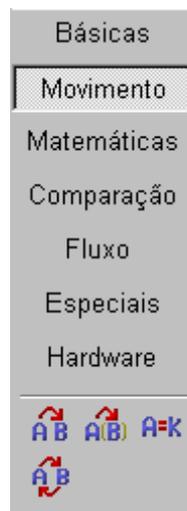


Figura 2 – Barra de ferramentas do editor Ladder, onde se encontra o bloco SWP.

- Selecione o ícone  associado ao bloco SWP. Estes passos inserirão um bloco SWP na posição do cursor no editor *ladder*.

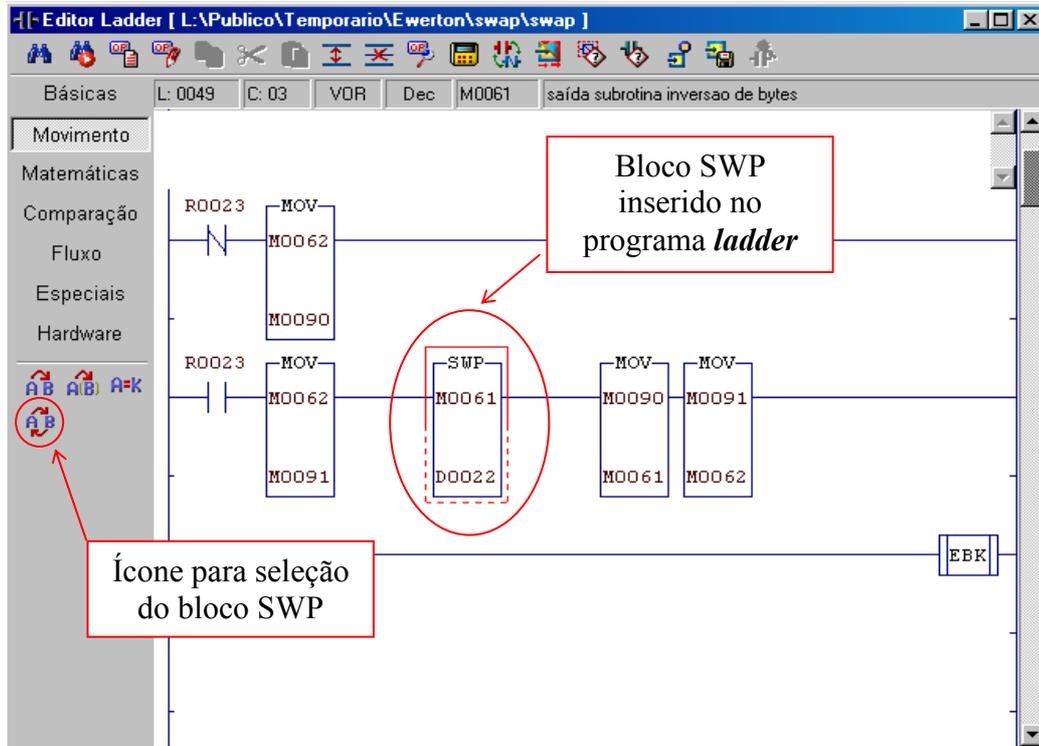


Figura 3 – Tela de programação do editor Ladder, com o bloco SWP inserido

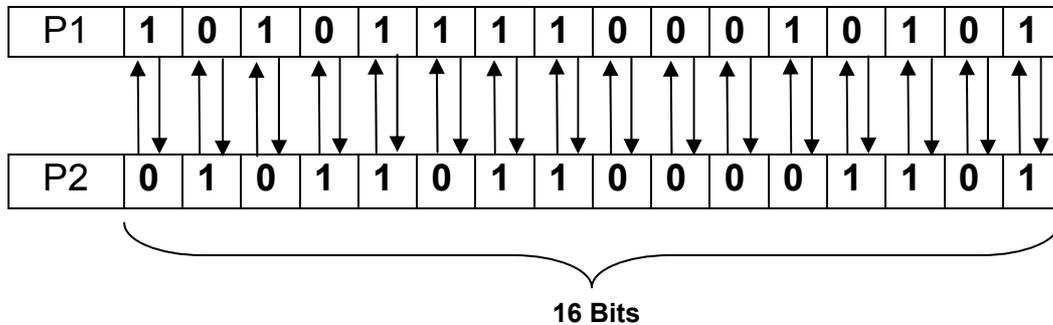
## 4. Troca de dados

### 4.1 Troca de dados entre operadores do mesmo tipo

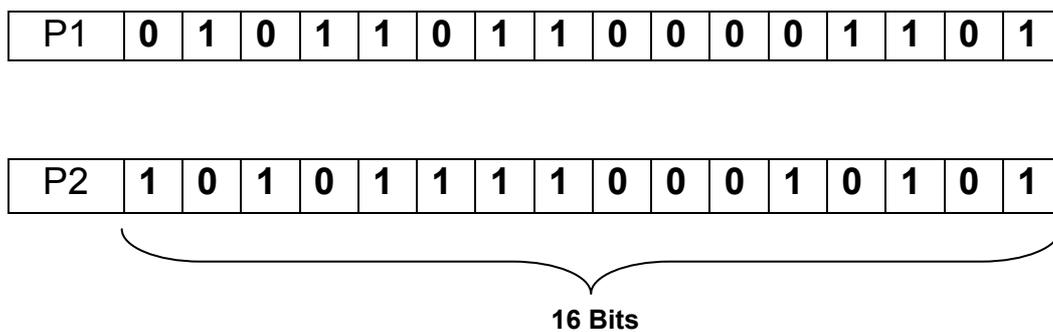
Quando os operadores são do mesmo tipo, como exemplo P1 e P2 sendo memória inteira **M**, ao energizar o bloco, os dados de P1 são transferidos para P2 e vice-versa. O processo é concretizado devido ao fato que na troca de dois parâmetros iguais o número de bits são os mesmos. As memórias inteiras **M** ocupam 1 **WORD** (16 bits) e as memórias reais **D** ocupam 2 **WORDS** (32 bits).

#### 4.1.1 Troca de dados entre memórias inteiras (tipo M)

Este exemplo mostra a troca de dados entre duas memórias inteiras tipo **M** (16 bits) usando o bloco SWP.

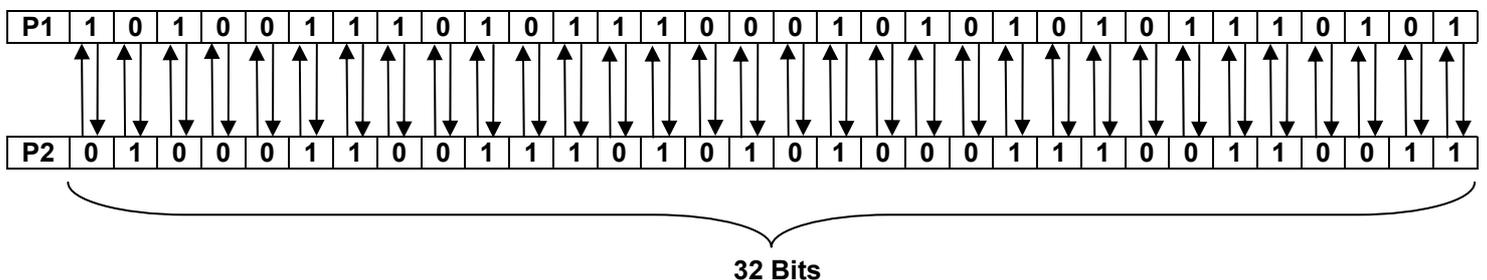


A figura a seguir apresenta o resultado da troca de dados após energizar o bloco SWP, observe que os dados que estavam na memória inteira **M** de P1 foram transferidos para a memória inteira **M** de P2. E os dados de P2 foram transferidos para P1.



#### 4.1.2 Troca de dados entre memórias reais (tipo D)

Este exemplo mostra a troca de dados entre duas memórias reais tipo **D** (32 bits) usando o bloco SWP.



A figura a seguir apresenta o resultado da troca de dados após energizar o bloco SWP, observe que os dados que estavam na memória real **D** de P1 foram transferidos para a memória real **D** de P2. E os dados de P2 foram transferidos para P1.

P1 0 1 0 0 0 0 1 1 0 0 1 1 1 0 1 0 1 0 1 0 0 0 1 1 1 0 0 1 1 0 0 1 1 0 0 1 1

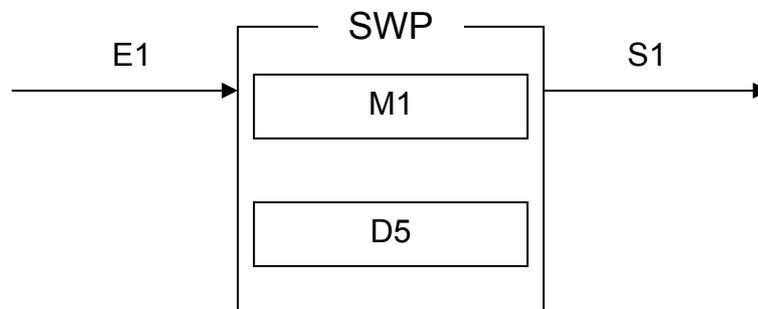
P2 1 0 1 0 0 1 1 1 0 1 0 1 1 1 0 0 0 1 0 1 0 1 0 1 0 1 1 1 0 1 0 1

32 Bits

## 4.2 Troca de dados entre operadores diferentes

### 4.2.1 Troca de dados entre memórias inteira (tipo M) e real (tipo D)

Quando P1 for uma memória inteira **M** e P2 for uma memória real **D**, será necessário usar uma memória **M+1** devido o fato da memória real **D** ser de 32 bits e memória inteira **M** ser de 16 bits. Ao realizar este processo, automaticamente o *ladder* usa a memória especificada em P1 mais uma seguinte. No exemplo abaixo, vamos nomear as memórias.



M2 (16 bits)

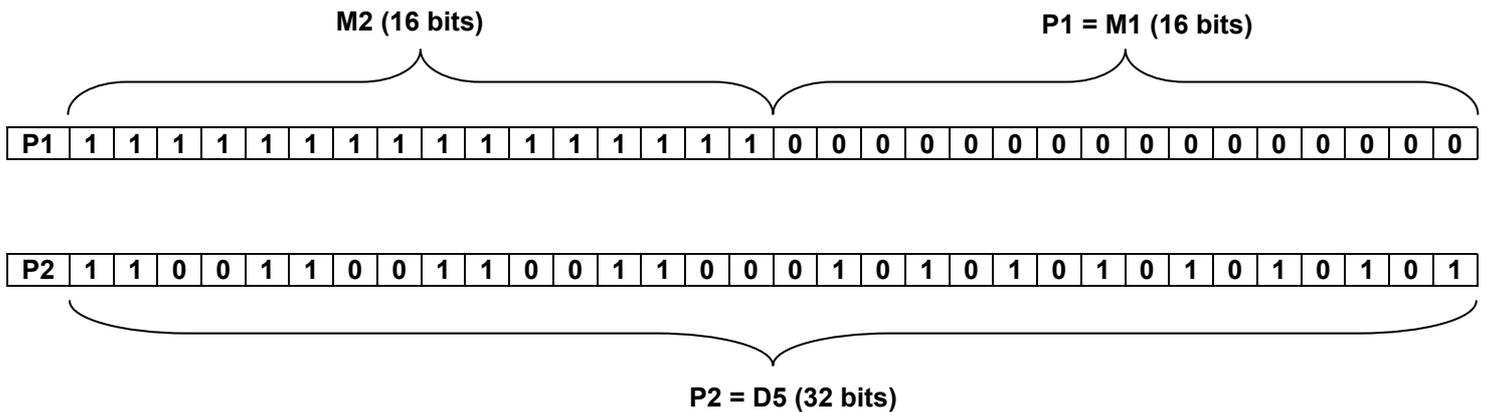
P1 = M1 (16 bits)

P1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1

P2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

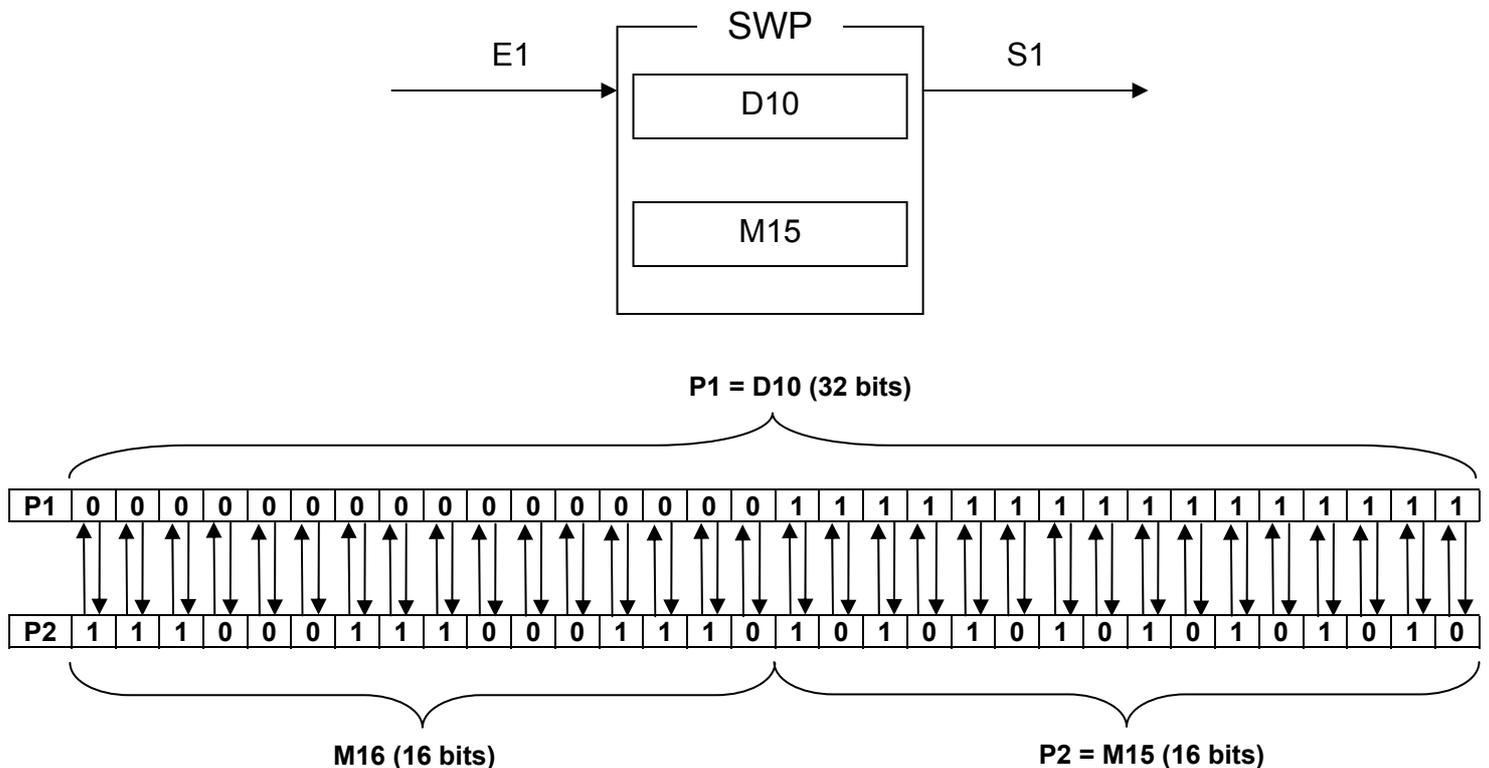
P2 = D5 (32 bits)

A figura a seguir apresenta o resultado da troca de dados após energizar o bloco SWP, observe que os dados que estavam nas memórias inteiras **M1** e **M2** foram transferidos para memória real **D5**. O valor de **D5** foi dividido em duas partes e distribuído nas memórias **M1** e **M2**, respectivamente.

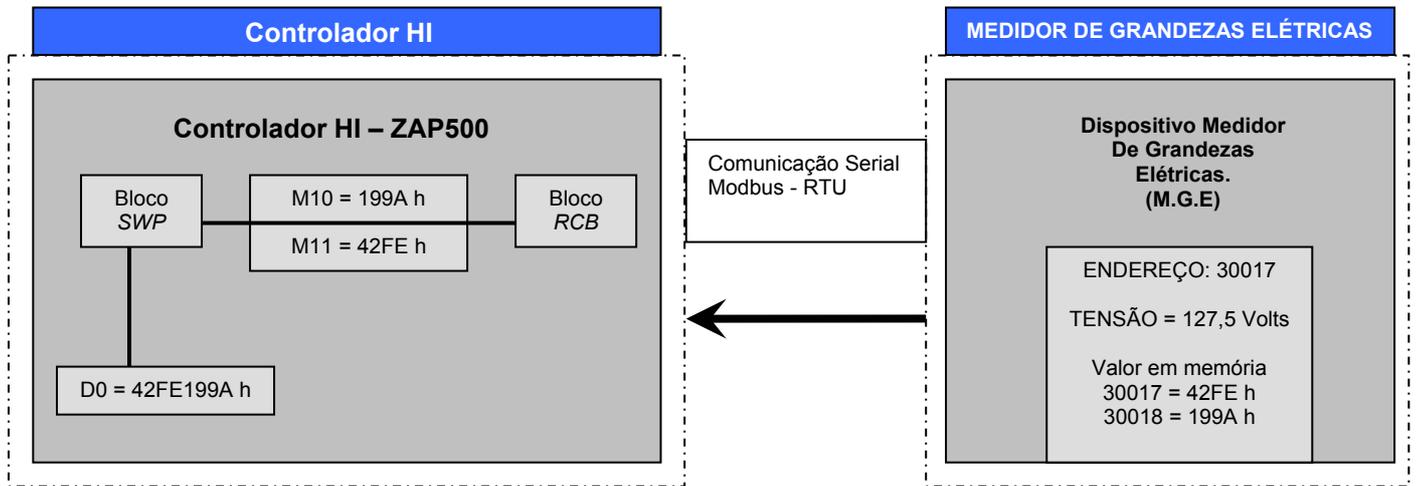


#### 4.2.2 Troca de dados entre memórias real (tipo D) e inteira (tipo M)

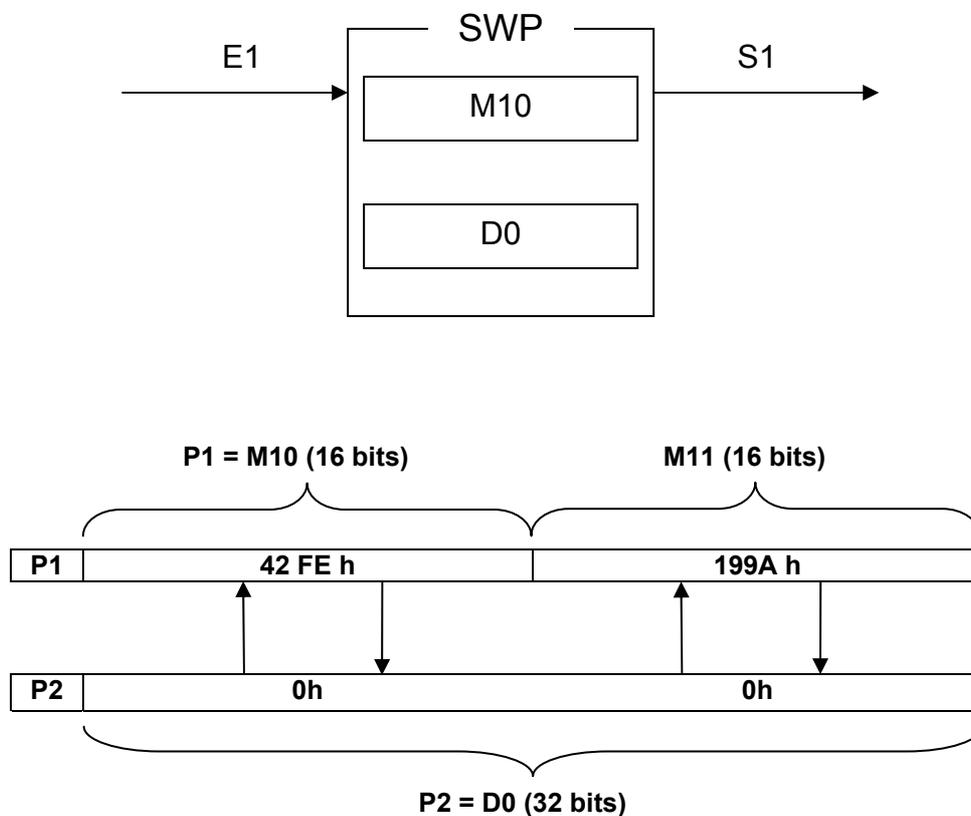
Quando P1 for uma memória real **D** e P2 uma memória inteira **M**, será necessário usar uma memória **M** seguinte **M+1** devido o fato de a memória **D** ser de 32 bits e memória **M** ser de 16 bits. Ao energizar o bloco pela entrada **E1**, a transferência dos dados de P1 para P2 a memória real **D** será dividida em duas memórias inteiras **M** para acoplar todos os dados. Siga o exemplo a seguir.



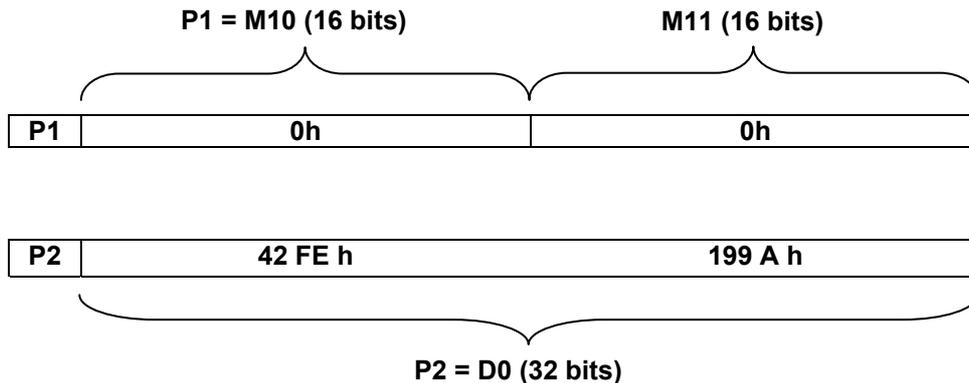




- O conteúdo dos endereços 30017 / 30018 que significam a tensão medida pelo dispositivo, são lidos pelo ZAP500 através do Bloco RCB e alocada nas memórias **M10 e M11** (devido à implementação do bloco no controlador).
- O bloco SWP é utilizado para compor uma memória real (**D0**), partindo-se de duas memórias inteiras (**M10 e M11**)



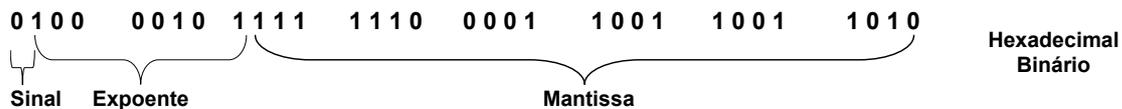
A figura a seguir apresenta o resultado da troca de dados após energizar o bloco SWP, observe que os dados que estavam nas memórias inteiras **M10** e **M11** foram transferidos para memória real **D0** e vice-versa.



O formato de representação de dados IEEE-754 32 bit float point é apresentado a seguir, exemplificando como é armazenado um valor real na memória do controlador.

D0 = 42FE 199A h (valor real representado em hexadecimal)

- Formatação do D0 no formato **IEEE-754 32 bit float point**



- Esta memória (D0) representa o valor de tensão obtida do Medidor de Grandezas Elétricas, que é igual a **127,05 Volts**.

No *website* da HI Tecnologia ([www.hitecnologia.com.br](http://www.hitecnologia.com.br)) podem ser obtidos os seguintes programas de exemplo:

- HIPJW024 – Utilização do bloco SWP para converter dados obtidos através de uma troca de dados via protocolo MODBUS-RTU com um medidor de grandezas elétricas
- HIPJW025 – Utilização do bloco SWP com operações simples de troca de dados considerando variáveis do mesmo tipo e diferentes tipos.



## Utilização da função de troca de dados (bloco SWP) no ambiente SPDSW

Tipo de Doc.: Notas de Aplicação  
Referência: ENA.0031

Revisão: 0  
Atualizado em: 29/11/2005

## Controle do Documento

---

### Considerações gerais

---

1. Este documento é dinâmico, estando sujeito a revisões, comentários e sugestões. Toda e qualquer sugestão para seu aprimoramento deve ser encaminhada ao departamento de suporte ao cliente da **HI Tecnologia**, especificado na “Apresentação” deste documento.
2. Os direitos autorais deste documento são de propriedade da **HI Tecnologia**.

### Responsabilidades pelo documento

---

|            | Data       | Responsável          |                          |
|------------|------------|----------------------|--------------------------|
| Elaboração | 02/09/2005 | Ewerton M. Gozzi     |                          |
| Revisão    | 29/09/2005 | Paulo C. M. Inazumi  | <i>Revisado em mídia</i> |
| Aprovação  | 29/11/2005 | Isaias M. C. Ribeiro | <i>Aprovado em mídia</i> |

#### Histórico de Revisões

| Data       | Rev | Descrição          |
|------------|-----|--------------------|
| 29/11/2005 | 0   | Documento original |