



HI tecnologia

Indústria e Comércio Ltda

Notas de Aplicação

**Utilização do Relógio de Tempo
Real - RTC**

HI Tecnologia

Documento de acesso publico

ENA.00007

Versão 1.01

abril-2005

Apresentação

Esta nota de aplicação foi elaborada pela **HI Tecnologia Indústria e Comércio Ltda.** Quaisquer dúvidas ou esclarecimentos sobre as informações contidas neste documento podem ser obtidas diretamente com o nosso departamento de suporte a clientes, através do telefone **(19) 2139-1700** ou do e-mail "suporte@hitecologia.com.br". Favor mencionar as informações a seguir para que possamos identificar os dados relativos a este documento.

ID da Nota de Aplicação: ENA.00007
Versão Documento: 1.01

HI Tecnologia Indústria e Comércio Ltda.

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

Cidade: Campinas – SP
CEP: 13076-015

Fone: +55 (19) 2139-1700
Fax: +55 (19) 2139-1710

E-mail: hi@hitecologia.com.br

Web site: www.hitecologia.com.br



Utilização do relógio de tempo real RTC

Referência: ENA.00007
Arquivo : ENA0000700.doc

Revisão: 1
Atualizado em: 18/04/2005

Índice

1.	Introdução.....	4
2.	O Relógio de Tempo Real (RTC).....	4
3.	Leitura dos Dados do Relógio de Tempo Real.....	4
3.1	Configuração dos parâmetros do RTC.....	4
3.1.1	Função AND:.....	5
3.1.2	Função OR:.....	6
3.1.3	Função deslocamento à direita (>>).....	6
3.2	Inserção de um bloco RTC no programa ladder.....	7
3.3	Filtragem do valor dos segundos.....	7
3.3.1	Exemplo de filtragem do valor dos segundos.....	7
3.4	Filtragem do valor dos minutos.....	8
3.4.1	Exemplo de filtragem do valor dos minutos.....	8
3.5	Filtragem do valor das horas.....	9
3.5.1	Exemplo de filtragem do valor das horas:.....	10
3.6	Filtragem do valor do dia da semana.....	10
3.6.1	Exemplo de filtragem do dia da semana.....	11
3.7	Filtragem do valor do dia do mês.....	12
3.7.1	Exemplo de filtragem do dia do mês.....	13
3.8	Filtragem do mês corrente.....	13
3.8.1	Exemplo de filtragem do mês corrente.....	14
3.9	Filtragem do ano corrente.....	14
3.9.1	Exemplo de filtragem do ano corrente.....	15
4.	Programação dos Dados do Relógio de Tempo Real.....	16
4.1	Selecionando o Bloco SCB no Ambiente SPDS.....	17
4.2	Função de Programação do RTC.....	19
4.3	Códigos de Retorno da Função de Programação do RTC.....	19
4.4	Exemplo de Utilização.....	20
5.	Utilização de RTC em mensagens da IHM.....	21
5.1	Programação para mensagens da IHM via SPDSW.....	21
5.2	Programação para mensagens da IHM via SPDS7.....	24
	Controle do Documento.....	26
	Considerações gerais.....	26
	Responsabilidades pelo documento.....	26



Utilização do relógio de tempo real RTC

Tipo de Doc.: Notas de Aplicação
Referência: ENA.00007

Revisão: 1
Atualizado em: 18/04/2005

1. Introdução

Este documento tem como objetivo fornecer informações sobre como utilizar o relógio de tempo real (RTC – Real Time Clock) presente nos controladores da HI Tecnologia.

O documento é dividido nas seguintes seções:

- Apresentação do Relógio de Tempo Real (RTC)
- Leitura dos dados do RTC
- Programação do RTC
- Utilização do RTC em mensagens da IHM

O documento foi concebido visando transmitir ao usuário as informações relativas ao RTC de uma forma bem didática, iniciando-se com uma rápida apresentação e fixando as informações por meio de exemplos práticos.

2. O Relógio de Tempo Real (RTC)

O Relógio de tempo real é um módulo que permite acessar a data e hora correntes, ou seja, é possível ter acesso à hora, minuto, segundo, dia do mês, mês e ano correntes, além de também informar o dia da semana, esses dados são atualizados mesmo com o controlador desligado, pois o RTC possui uma bateria interna.

As informações provenientes do RTC podem ser utilizadas no programa Ladder, sendo possível a sua utilização através de uma correta implementação nas lógicas do programa. Deve-se verificar a disponibilidade do RTC no controlador HI utilizado.

3. Leitura dos Dados do Relógio de Tempo Real

3.1 Configuração dos parâmetros do RTC

O relógio de tempo real pode ser acessado, no SPDS, através de um bloco RTC composto por três parâmetros, P1, P2 e P3. Para cada um destes parâmetros, devemos especificar uma memória inteira (Tipo M), que armazenará os resultados dos parâmetros.

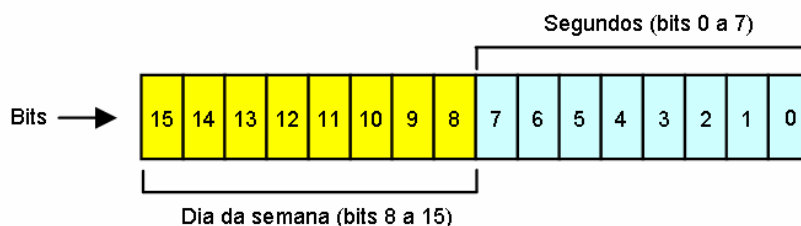
Uma memória do tipo M é composta por 16 bits, numerados de 0 a 15, sendo que o bit 0 é o bit menos significativo (LSB) e o bit 15 é o bit mais significativo (MSB).

Parâmetros	Bits	Significado
P1	Bit 0...7	Segundos
	Bits 8...15	Dia da semana
P2	Bit 0...7	Minutos
	Bit 8...15	Horas
P3	Bit 0...5	Dia do mês
	Bit 6...9	Mês
	Bit 10...15	Ano (off-set a partir de 1990)

Tabela 1 - Tabela com os parâmetros do RTC

Cada parâmetro é responsável por mais de um tipo de informação. Por exemplo: No parâmetro P1, estão contidas as informações de dia da semana e de segundos correntes. Na ilustração abaixo, veja como serão alocadas as informações na memória tipo M (inteira) que conterá o parâmetro P1:

Parâmetro P1:



Conforme podemos visualizar, os bits de 0 a 7 armazenam os segundos e os bits de 8 a 15 o dia da semana. Entretanto, para conseguirmos utilizar estas informações separadamente, precisaremos transportar as informações para memórias distintas, isto é, precisaremos filtrar as informações de dia da semana e de segundos e mover cada uma delas para memórias inteiras distintas, ou seja faremos um tratamento dos bits referentes ao dia da semana e enviaremos esta informação para uma memória M (inteira), e faremos a mesma coisa com os bits relativos aos segundos.

Para tratarmos os bits de uma memória, utilizaremos as funções AND, OR e a função deslocamento à direita (>>). A seguir para um melhor entendimento, explicaremos a seguir as funções AND, OR e deslocamento à direita (>>)

3.1.1 Função AND:

A Tabela 2 descreve o comportamento da função AND, sendo que A e B são as entradas e S é a saída da função. Em resumo, a saída S terá o valor lógico "1" apenas quando as duas entradas (A e B) possuírem o valor "1".

A	B	S
0	0	0
0	1	0
1	0	0
1	1	1

Tabela 2 - Tabela verdade da função AND

3.1.2 Função OR:

A *Tabela 3* descreve o comportamento da função OR, sendo que A e B são as entradas e S é a saída da função. Em resumo, a saída S terá o valor lógico "1" quando qualquer uma das duas entradas (A e B) possuir o valor "1".

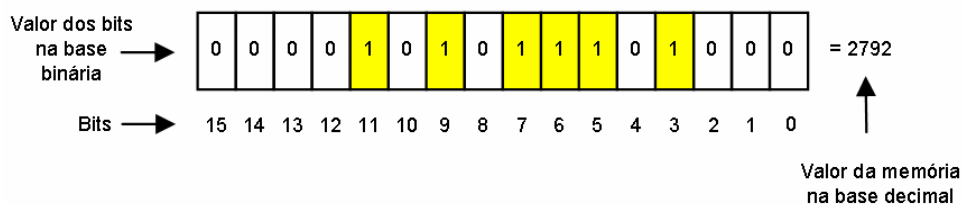
A	B	S
0	0	0
0	1	1
1	0	1
1	1	1

Tabela 3 - Tabela verdade da função OR

3.1.3 Função deslocamento à direita (>>)

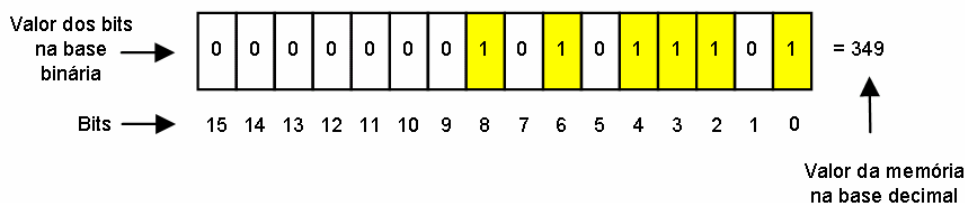
A função deslocamento à direita (>>) efetua o deslocamento dos bits de uma memória, sendo que o valor desta memória é deslocado na forma binária. A quantidade de posições que os bits serão deslocados é programável e faz parte dos parâmetros de chamada desta função. No exemplo abaixo, temos uma memória inteira (tipo M) cujo conteúdo na base decimal antes do deslocamento é igual a 2792.

Memória tipo M antes do deslocamento



Desejamos então efetuar o deslocamento de todos os bits 3 posições à direita:

Após o deslocamento de 3 bits à direita




Note que após o deslocamento, todos os bits estão deslocados 3 posições à direita. Os bits menos significativos foram descartados e os bits mais significativos que foram inseridos possuem o valor zero. O valor desta memória convertido para a base decimal passou a ser 349.

3.2 Inserção de um bloco RTC no programa ladder

Para explorarmos os exemplos a seguir, o primeiro passo é inserir no programa Ladder um bloco do relógio de tempo real (RTC).

Para inserir um bloco RTC no programa Ladder, siga o procedimento abaixo:

1. Abra o SPDSW e depois o editor Ladder.
2. Mova o cursor para a posição onde você deseja inserir o bloco RTC.
3. Com o mouse, clique no menu à esquerda do editor Ladder no menu “Especiais”
4. Dê um clique no botão . E em seguida digite M0000 para o parâmetro P1, M0001 para o parâmetro P2 e M0002 para o parâmetro P3. (Estas memórias foram utilizadas apenas para fins didáticos, podem ser utilizadas outras memórias tipo M desde que estejam em seqüência.)

Agora que já temos o bloco de RTC inserido no programa Ladder, podemos utilizar as informações que ele nos fornece. Como já discutimos anteriormente, todas as informações de horas, minutos, segundos, dia do mês, mês e ano estão condensadas em três parâmetros. Para extrair estas informações individualmente, será necessário um tratamento especial dos dados. Este tratamento será mostrado nos itens a seguir.

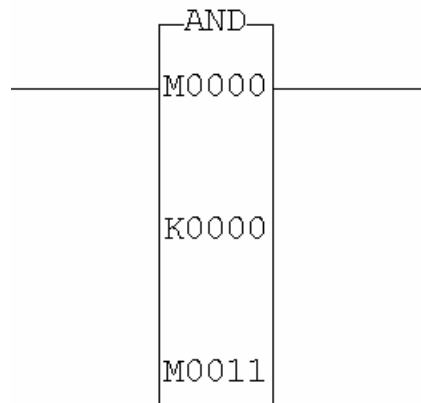
3.3 Filtragem do valor dos segundos

Conforme podemos visualizar na Tabela 1, os segundos estão alocados nos bits de 0 a 7 do parâmetro P1. Portanto, para transportar o valor dos segundos para uma memória qualquer, devemos efetuar os seguintes passos:

- 1 – Inserir um AND no programa Ladder
- 2 – Programar os três parâmetros do bloco AND, da seguinte forma:
 - Parâmetro P1: Especificar o número da memória que está sendo utilizada no parâmetro P1 do bloco RTC.
 - Parâmetro P2: Especificar uma constante K (constante inteira) cujo valor é FFh (na base hexadecimal) ou 11111111 (na base binária).
 - Parâmetro P3: Especificar uma memória inteira (tipo M) para onde será movido o valor dos segundos.
- 3 – O valor dos segundos estará disponível na memória especificada no parâmetro P3 do bloco AND.

3.3.1 Exemplo de filtragem do valor dos segundos

A seguir, temos um exemplo de lógica no SPDSW onde ocorre a filtragem dos valores dos segundos:



No exemplo acima, o primeiro parâmetro do bloco AND é a memória inteira M0000, que é a memória na qual está armazenado o primeiro parâmetro do bloco RTC. O segundo parâmetro é uma constante K0000 cujo valor é FFh (na base hexadecimal) que será utilizada na filtragem dos bits do parâmetro de 0 a 7. Já o terceiro parâmetro, que no exemplo é M0011, é a memória na qual será armazenado o valor dos segundos.

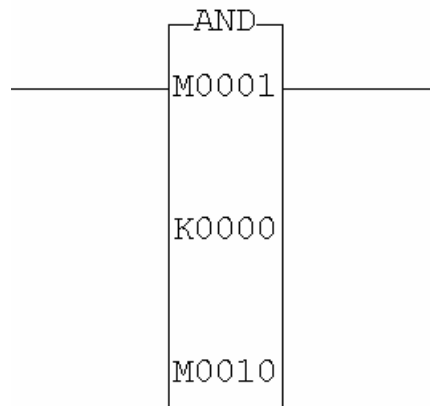
3.4 Filtragem do valor dos minutos

Conforme podemos visualizar na Tabela 1, os minutos estão alocados nos bits de 0 a 7 do parâmetro P2. Portanto, para transportar o valor dos minutos para uma memória qualquer, devemos efetuar os seguintes passos:

- 1 – Inserir um AND no programa Ladder
- 2 – Programar os três parâmetros do bloco AND, da seguinte forma:
 - Parâmetro P1: Especificar o número da memória que está sendo utilizada no parâmetro P2 do bloco RTC.
 - Parâmetro P2: Especificar uma constante K (constante inteira) cujo valor é FFh (na base hexadecimal) ou 11111111 (na base binária).
 - Parâmetro P3: Especificar uma memória inteira (tipo M) para onde será movido o valor dos minutos.
- 3 – O valor dos minutos estará disponível na memória especificada no parâmetro P3 do bloco AND.

3.4.1 Exemplo de filtragem do valor dos minutos

A seguir, temos um exemplo de lógica no SPDSW onde ocorre a filtragem dos valores dos minutos:



No exemplo acima, o primeiro parâmetro do bloco AND é a memória inteira M0001, que é a memória na qual está armazenado o segundo parâmetro do bloco RTC. O segundo parâmetro é uma constante K0000 cujo valor é FFh (na base hexadecimal) e que será utilizada na filtragem dos bits do parâmetro de 0 a 7. Já o terceiro parâmetro, que no exemplo é M0010, é a memória na qual será armazenado o valor dos minutos.

3.5 Filtragem do valor das horas

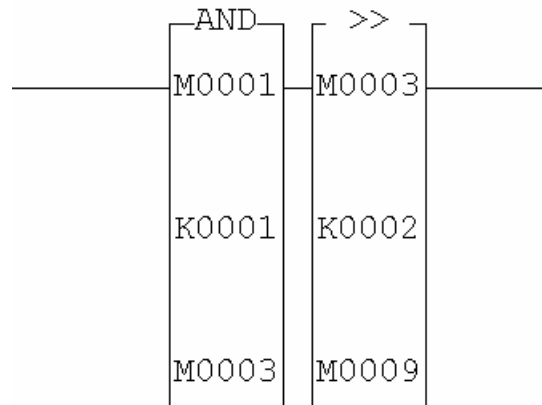
Conforme podemos visualizar na Tabela 1, as horas estão alocadas nos bits de 8 a 15 do parâmetro P2. Portanto, para transportar o valor das horas para uma memória qualquer, devemos efetuar os seguintes passos:

- 1 – Inserir um AND no programa Ladder
- 2 – Programar os três parâmetros do bloco AND, da seguinte forma:
 - Parâmetro P1: Especificar o número da memória que está sendo utilizada no parâmetro P2 do bloco RTC.
 - Parâmetro P2: Especificar uma constante K (constante inteira) cujo valor é FF00h (na base hexadecimal) ou 1111111100000000 (na base binária).
 - Parâmetro P3: Especificar uma memória inteira (tipo M) que receberá o valor filtrado dos bits 8 a 15 do parâmetro P2. Essa memória é auxiliar, pois o conteúdo dela sofrerá um deslocamento à direita.
- 3 – Inserir um bloco de deslocamento à direita (>>).
- 4 – Programar os três parâmetros do bloco de deslocamento à direita (>>) da seguinte forma:
 - Parâmetro P1: Especificar a memória que foi indicada no terceiro parâmetro do bloco AND para a filtragem do valor das horas. O conteúdo desta memória sofrerá um deslocamento à direita
 - Parâmetro P2: Determinar uma constante inteira (K), cujo valor vai especificar a quantidade de posições que os bits serão deslocados. O valor desta constante deve ser 8.
 - Parâmetro P3: Especificar uma memória inteira (tipo M), sendo que esta memória receberá o valor das horas.

Em resumo, os bits de 8 a 15 do parâmetro P2 serão filtrados e deslocados 8 casas à direita.

3.5.1 Exemplo de filtragem do valor das horas:

A seguir, temos um exemplo de aplicação para a filtragem de horas no SPDSW:



No exemplo acima, o primeiro parâmetro do bloco AND é a memória M0001, sendo que esta é a memória do segundo parâmetro do bloco RTC. O segundo parâmetro do bloco AND é uma constante inteira (tipo K), cujo valor é FF00h na base hexadecimal e que será utilizada na filtragem dos bits 8 a 15 do parâmetro P2 do RTC.

O terceiro parâmetro do bloco AND é a memória M0003, sendo que ela receberá o valor filtrado dos bits 8 a 15 e armazenará este valor para o deslocamento. Ao lado do bloco AND, temos o bloco de deslocamento à direita (>>). Este bloco é o responsável por deslocar os bits da memória M0003.

No primeiro parâmetro deste bloco, foi especificada a memória M0003, que é a origem dos dados que sofrerão deslocamento. No segundo parâmetro, foi especificada uma constante inteira (tipo K) que conterà a quantidade de posições que os bits da memória M0003 sofrerão o deslocamento à direita.

No terceiro parâmetro, determinou-se a memória M0009 para receber os dados referentes à hora do RTC.

Portanto, o bloco AND efetuou a filtragem dos bits e o bloco de deslocamento à direita (>>) deslocou os bits 8 casas à direita e disponibilizou este valor na memória M0009.

3.6 Filtragem do valor do dia da semana

O valor do dia da semana é numérico, sendo codificado conforme descrito abaixo :

- 1 - Domingo,
- 2 - Segunda-feira,
- 3 - Terça-feira,
- 4 - Quarta-feira,
- 5 - Quinta-feira,
- 6 - Sexta-feira,
- 7 - Sábado.

O dia da semana é atualizado após a mudança de dia, ou seja, na virada de 23:59:59 horas para 00:00:00 horas. Para inicializar o dia da semana corretamente, utilizar a opção de ajuste de data e hora no SPDSW, onde se permite configurar corretamente o dia da semana nos controladores HI (Menu “Controlador”, item “Programa Relógio/Calendário”)

O dia da semana está armazenado nos bits 8 a 15 do parâmetro P1, este valor deverá ser movido para uma memória auxiliar, filtrado e depois deslocado à direita 8 posições.

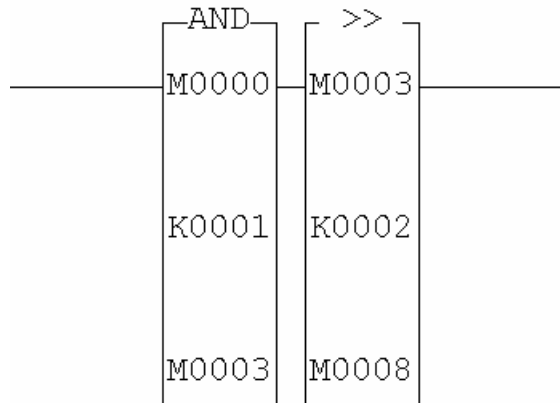
Para obter o valor do dia da semana, devemos executar os seguintes passos:

- 1 – Inserir um bloco AND no programa Ladder
- 2 – Programar os três parâmetros do bloco AND, da seguinte forma:
 - Parâmetro P1: Especificar o número da memória que está sendo utilizada no parâmetro P1 do bloco RTC.
 - Parâmetro P2: Especificar uma constante K (constante inteira) cujo valor é FF00h (na base hexadecimal) ou 1111111100000000 (na base binária).
 - Parâmetro P3: Especificar uma memória inteira (tipo M) que receberá o valor filtrado dos bits 8 a 15 do parâmetro P1. Essa memória é auxiliar, pois o conteúdo dela sofrerá um deslocamento à esquerda.
- 3 – Inserir um bloco de deslocamento à direita (>>).
- 4 – Programar os três parâmetros do bloco de deslocamento à direita (>>) da seguinte forma:
 - Parâmetro P1: Especificar a memória que foi indicada no terceiro parâmetro do bloco AND para a filtragem do valor do dia da semana. O conteúdo desta memória sofrerá um deslocamento à direita.
 - Parâmetro P2: Determinar uma constante inteira (K), cujo valor vai especificar a quantidade de posições que os bits serão deslocados. O valor desta constante deve ser 8.
 - Parâmetro P3: Especificar uma memória inteira (tipo M), sendo que esta memória receberá o valor do dia da semana.

Em resumo, os bits de 8 a 15 do parâmetro P1 serão filtrados e deslocados 8 casas à direita.

3.6.1 Exemplo de filtragem do dia da semana

A seguir, temos um exemplo de aplicação para a filtragem do dia da semana no SPDSW:



No exemplo acima, o primeiro parâmetro do bloco AND é a memória M0000, sendo que esta é a memória do primeiro parâmetro do bloco RTC. O segundo parâmetro do bloco AND é uma constante inteira (tipo K), cujo valor é FF00h na base hexadecimal e que será utilizada na filtragem dos bits 8 a 15 do parâmetro P1 do RTC.

O terceiro parâmetro do bloco AND é a memória M0003, sendo que ela receberá o valor filtrado dos bits 8 a 15 e armazenará este valor para o deslocamento. Ao lado do bloco AND, temos o bloco de deslocamento à direita (>>). Este bloco é o responsável por deslocar os bits da memória M0003.

No primeiro parâmetro deste bloco, foi especificada a memória M0003, que é a origem dos dados que sofrerão deslocamento. No segundo parâmetro, foi especificada uma constante inteira (tipo K) K0002 que conterá a quantidade de posições que os bits da memória M0003 sofrerão o deslocamento à direita. No terceiro parâmetro, determinou-se a memória M0008 para receber os dados referentes ao dia da semana do RTC.

Portanto, o bloco AND efetuou a filtragem dos bits e o bloco de deslocamento à direita (>>) deslocou os bits 8 casas à direita e disponibilizou este valor na memória M0008.

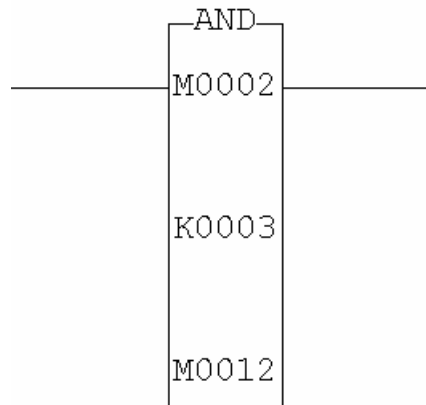
3.7 Filtragem do valor do dia do mês

Conforme podemos visualizar na Tabela 1, o dia do mês está alocado nos bits de 0 a 5 do parâmetro P3. Portanto, para transportar o valor do dia do mês para uma memória qualquer, devemos efetuar os seguintes passos:

- 1 – Inserir um AND no programa Ladder
- 2 – Programar os três parâmetros do bloco AND, da seguinte forma:
 - Parâmetro P1: Especificar o número da memória que está sendo utilizada no parâmetro P3 do bloco RTC.
 - Parâmetro P2: Especificar uma constante K (constante inteira) cujo valor é 3F (na base hexadecimal) ou 111111 (na base binária).
 - Parâmetro P3: Especificar uma memória inteira (tipo M) para onde será movido o valor do dia do mês.
- 3 – O valor do dia do mês estará disponível na memória especificada no parâmetro P3 do bloco AND.

3.7.1 Exemplo de filtragem do dia do mês

A seguir, temos um exemplo de lógica no SPDSW onde ocorre a filtragem do valor do dia do mês:



No exemplo acima, o primeiro parâmetro do bloco AND é a memória inteira M0002, que é a memória na qual está armazenado o terceiro parâmetro do bloco RTC. O segundo parâmetro é uma constante K0003 cujo valor é 3F (na base hexadecimal) e que será utilizada na filtragem dos bits 0 a 5 do parâmetro P3 do RTC. Já o terceiro parâmetro do bloco AND, que no exemplo é M0012, é a memória na qual será armazenado o valor do dia do mês.

3.8 Filtragem do mês corrente

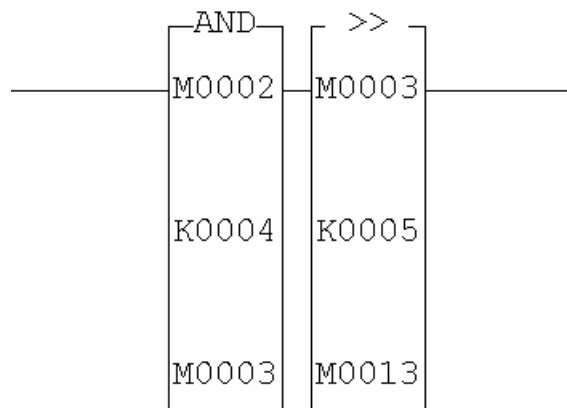
O mês está armazenado nos bits 6 a 9 do parâmetro P3. Este valor deverá ser movido para uma memória auxiliar, filtrado e depois deslocado à direita 6 posições. Para obter o valor do mês corrente, devemos executar os seguintes passos:

- 1 – Inserir um bloco AND no programa Ladder
- 2 – Programar os três parâmetros do bloco AND, da seguinte forma:
 - Parâmetro P1: Especificar o número da memória que está sendo utilizada no parâmetro P3 do bloco RTC.
 - Parâmetro P2: Especificar uma constante K (constante inteira) cujo valor é 3C0h (na base hexadecimal) ou 1111000000 (na base binária).
 - Parâmetro P3: Especificar uma memória inteira (tipo M) que receberá o valor filtrado dos bits 6 a 9 do parâmetro P3. Essa memória é auxiliar, pois o conteúdo dela sofrerá um deslocamento à direita.
- 3 – Inserir um bloco de deslocamento à direita (>>).
- 4 – Programar os três parâmetros do bloco de deslocamento à direita (>>) da seguinte forma:
 - Parâmetro P1: Especificar a memória que foi indicada no terceiro parâmetro do bloco AND para a filtragem do valor do mês. O conteúdo desta memória sofrerá um deslocamento à direita
 - Parâmetro P2: Determinar uma constante inteira (K), cujo valor vai especificar a quantidade de posições que os bits serão deslocados. O valor desta constante deve ser 6.
 - Parâmetro P3: Especificar uma memória inteira (tipo M), sendo que esta memória receberá o valor do mês.

Em resumo, os bits de 6 a 9 do parâmetro P3 serão filtrados e deslocados 6 casas à direita.

3.8.1 Exemplo de filtragem do mês corrente

A seguir, temos um exemplo de lógica no SPDSW onde ocorre a filtragem do valor do mês corrente:



No exemplo acima, o primeiro parâmetro do bloco AND é a memória M0002, sendo que esta é a memória do terceiro parâmetro do bloco RTC. O segundo parâmetro do bloco AND é uma constante inteira (tipo K), cujo valor é 3C0 na base hexadecimal e que será utilizada na filtragem dos bits 6 a 9 do parâmetro P3 do RTC.

O terceiro parâmetro do bloco AND é a memória M0003, sendo que ela receberá o valor filtrado dos bits 6 a 9 e armazenará este valor para o deslocamento. Ao lado do bloco AND, temos o bloco de deslocamento à direita (>>). Este bloco é o responsável por deslocar os bits da memória M0003.

No primeiro parâmetro deste bloco, foi especificada a memória M0003, que é a origem dos dados que sofrerão deslocamento. No segundo parâmetro, foi especificada uma constante inteira (tipo K) K0005, cujo valor é 6, e que contém a quantidade de posições que os bits da memória M0003 sofrerão o deslocamento à direita. No terceiro parâmetro, determinou-se a memória M0013 para receber os dados referentes ao mês corrente do RTC.

Portanto, o bloco AND efetuou a filtragem dos bits e o bloco de deslocamento à direita (>>) deslocou os bits 6 casas à direita e disponibilizou este valor na memória M0013.

3.9 Filtragem do ano corrente

O ano corrente está armazenado nos bits 10 a 15 do parâmetro P3 do bloco RTC. Este valor deverá ser movido para uma memória auxiliar, filtrado, deslocado à direita 10 posições e somado com o offset de 1990 anos. Para obter o valor do ano corrente, devemos executar os seguintes passos:

- 1 – Inserir um bloco AND no programa Ladder
- 2 – Programar os três parâmetros do bloco AND, da seguinte forma:
 - Parâmetro P1: Especificar o número da memória que está sendo utilizada no parâmetro P3 do bloco RTC.

- Parâmetro P2: Especificar uma constante K (constante inteira) cujo valor é FC00h (na base hexadecimal) ou 1111110000000000 (na base binária).
- Parâmetro P3: Especificar uma memória inteira (tipo M) que receberá o valor filtrado dos bits 10 a 15 do parâmetro P3. Essa memória é auxiliar, pois o conteúdo dela sofrerá um deslocamento à direita.

3 – Inserir um bloco de deslocamento à direita (>>).

4 – Programar os três parâmetros do bloco de deslocamento à direita (>>) da seguinte forma:

- Parâmetro P1: Especificar a memória que foi indicada no terceiro parâmetro do bloco AND para a filtragem do valor do ano. O conteúdo desta memória sofrerá um deslocamento à direita.
- Parâmetro P2: Determinar uma constante inteira (K), cujo valor vai especificar a quantidade de posições que os bits serão deslocados. O valor desta constante deve ser 10.
- Parâmetro P3: Especificar uma memória inteira (tipo M), sendo que esta memória receberá o valor do ano (sem o off-set de 1990).

5 – Inserir um bloco de soma (ADD).

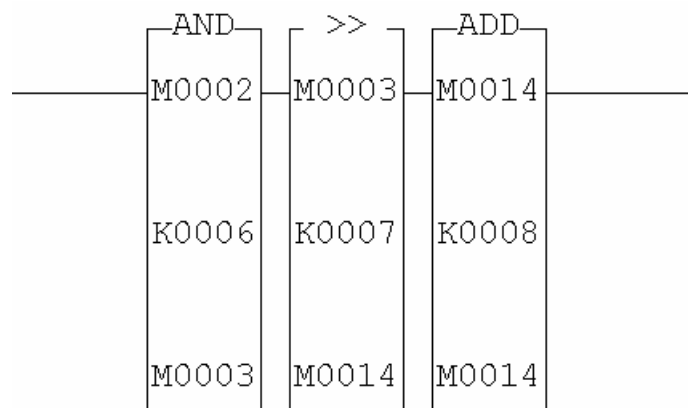
6 – Programar os três parâmetros do bloco somador da seguinte forma:

- Parâmetro P1: Especificar a memória do terceiro parâmetro (P3) do bloco de deslocamento à direita usado anteriormente.
- Parâmetro P2: Determinar uma constante inteira (K), cujo valor deve ser 1990.
- Parâmetro P3: Especificar uma memória inteira (tipo M), sendo que esta memória irá receber o valor do ano corretamente, isto é, vai receber a soma do valor do ano extraído do bloco RTC e somado com o valor 1990.

Em resumo, os bits de 10 a 15 do parâmetro P1 serão filtrados, deslocados 10 casas à direita e somados com o valor 1990.

3.9.1 Exemplo de filtragem do ano corrente

A seguir, temos um exemplo de lógica no SPDSW onde ocorre a filtragem dos valores do ano corrente:



No exemplo acima, o primeiro parâmetro do bloco AND é a memória M0002, sendo que esta é a memória do terceiro parâmetro do bloco RTC. O segundo parâmetro do bloco AND é uma constante inteira (tipo K), cujo valor é FC00h na base hexadecimal e que será utilizada na filtragem dos bits 10 a 15 do parâmetro P3 do RTC.

O terceiro parâmetro do bloco AND é a memória M0003, sendo que ela receberá o valor filtrado dos bits 10 a 15 e armazenará este valor para o deslocamento. Ao lado do bloco AND, temos o bloco de deslocamento à direita (>>). Este bloco é o responsável por deslocar os bits da memória M0003. No primeiro parâmetro deste bloco, foi especificada a memória M0003, que é a origem dos dados que sofrerão deslocamento.

No segundo parâmetro, foi especificada uma constante inteira (tipo K) K0007, cujo valor é 10, e que contém a quantidade de posições que os bits da memória M0003 sofrerão o deslocamento à direita. No terceiro parâmetro, determinou-se a memória M0014 para receber os dados referentes ao ano corrente do RTC (sem o off-set de 1990). Por fim, ao lado do bloco de deslocamento à direita temos o bloco somador, onde foi especificada a memória M0014 para o primeiro parâmetro. No segundo parâmetro foi especificada a constante inteira (tipo K) K0008, cujo valor é 1990, sendo que este valor é o valor de off-set.

No terceiro parâmetro, foi especificada a memória inteira (tipo M) M0014, que receberá o valor correto do ano (valor obtido do bloco RTC + 1990). Portanto, o bloco AND efetuou a filtragem dos bits, o bloco de deslocamento à direita (>>) deslocou os bits 10 casas à direita e o bloco somador fez a adição do valor deslocado com o valor 1990, e por fim disponibilizou este valor na memória M0014.

4. Programação dos Dados do Relógio de Tempo Real

Em aplicações desenvolvidas no ambiente SPDS deve-se utilizar o bloco SCB ("Bloco de Controle Padrão") para programar a data e a hora do RTC do controlador HI. O bloco SCB é composto por uma entrada E1, 4 parâmetros (P1, P2, P3 e P4) e uma saída S1, como ilustra a figura abaixo:

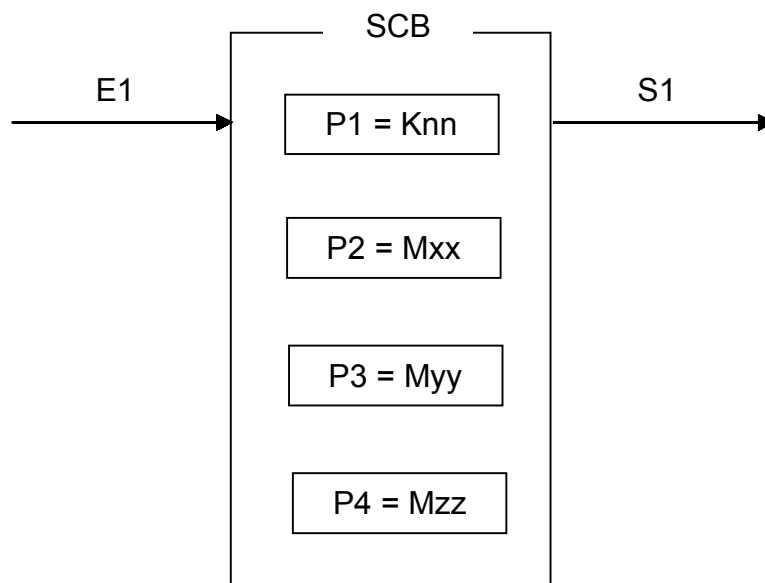


Figura - Função Ladder: Bloco "SCB"

Os elementos que compõem o bloco SCB são descritos abaixo:

- Parâmetro P1: Deve ser uma constante inteira do tipo **Knn**, onde o valor especificado para essa constante indica a função do bloco, devendo ser:
 - Igual a 65 para programação do RTC do controlador HI;
- Parâmetro P2: Deve ser uma memória inteira do tipo **Mxx**, a partir dessa deve existir uma seqüência de sete memórias consecutivas com parâmetros para a função. Exemplo: Sendo Mxx igual a M10, a seqüência será M10, M11, M12, M13, M14, M15 e M16.
- Parâmetro P3: Deve ser uma memória inteira do tipo **Myy**. No caso desta função de programação do RTC este parâmetro deve ser especificado, mas não é utilizado.
- Parâmetro P4: Deve ser uma memória inteira do tipo **Mzz**. Nesta memória Mzz sempre será retornado o código de retorno da execução de programação do RTC. Neste caso, se retornar um valor 0 (ZERO) indica função executada com sucesso, caso contrário indica o código de erro associado à execução da função selecionada.
- Entrada E1: sinal de habilitação do bloco SCB, onde:
 - Energizado - Bloco habilitado, executa programação do RTC.
 - Desenergizado - Bloco desabilitado, não executa nenhuma função.


Observação: Cabe ressaltar que na funcionalidade deste bloco SCB, o tratamento da função de programação do RTC somente é executado após a transição de subida desta entrada E1, devendo permanecer ativa até o término da execução da função, no caso indicado quando a saída S1 torna-se energizada.
- Saída S1: sinal de saída do bloco SCB, onde:
 - Energizado – Indicação de término da execução da função de programação do RTC.
 - Desenergizado – Bloco não habilitado, ou está habilitado, mas houve erro na execução da função de programação do RTC.

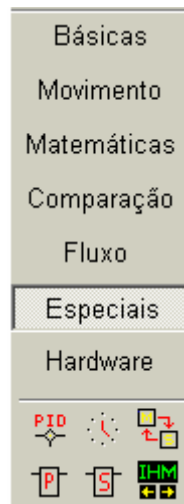
Esta funcionalidade de programação do RTC do controlador HI esta disponível a partir das seguintes versões de *firmwares* dos controladores HI:

- MCI02-QC: versão 2.03 ou superior
- ZAP-500 : versão 1.43 ou superior
- MIX-600: Sem suporte para RTC

4.1 Selecionando o Bloco SCB no Ambiente SPDS

Para inserir um bloco SCB em seu programa de aplicação ladder, deve-se proceder como descrito a seguir, dependendo do ambiente em que se estiver trabalhando:

- **Utilizando o SPDSW** : Posicione o cursor no local desejado do programa ladder, selecione a opção “Especiais” na palheta de comandos à esquerda da janela do editor ladder, e em seguida selecione o botão  .



Após a inserção do bloco SCB no programa *ladder*, deve-se configurar os parâmetros do mesmo.



Utilização do relógio de tempo real RTC

Tipo de Doc.: Notas de Aplicação
Referência: ENA.00007

Revisão: 1
Atualizado em: 18/04/2005

4.2 Função de Programação do RTC

Função

Programação do RTC

Descrição da Função

Executar a programação do RTC do controlador HI.

P1: Parâmetros de Entrada

Constante Knn Função do bloco SCB de acesso à programação do RTC.
Deve ser uma constante inteira (Knn) cujo valor deve ser 65.

P2: Parâmetros de Entrada

Memória xx + 0 Valor dos SEGUNDOS a serem programados no RTC (0..59)

Memória xx + 1 Valor dos MINUTOS a serem programados no RTC (0..59)

Memória xx + 2 Valor das HORAS a serem programadas no RTC (1..23)

Memória xx + 3 Valor do dia da SEMANA a ser programado no RTC (1..7), onde :
1-Dom, 2-Seg, 3-Ter, 4-Qar, 5-Quin, 6-Sex, 7-Sab

Memória xx + 4 Valor do DIA do mês a ser programado no RTC (1..31)

Memória xx + 5 Valor do MÊS a ser programado no RTC (1..12)

Memória xx + 6 Valor do ANO a ser programado no RTC. No caso, são válidos ano a partir de 1990.

P3: Parâmetros de Saída

Memória yy + 0 Parâmetro dummy não utilizado nesta função de programação do RTC

P4: Parâmetros de Saída

Memória zz + 0 Código de retorno da execução da função de programação do RTC, onde :
Igual a 0 Indica função executada com sucesso.
Diferente de 0 Indica condição de falha. Vide a lista de códigos de retorno

4.3 Códigos de Retorno da Função de Programação do RTC

Código de Retorno	Descrição
4	Controlador sem suporte de RTC disponível
5	Algum parâmetro de programação do RTC (semana, dia, mês, ano, hora, minuto, ou segundo) está fora da escala permitida.



Utilização do relógio de tempo real RTC

Tipo de Doc.: Notas de Aplicação
Referência: ENA.00007

Revisão: 1
Atualizado em: 18/04/2005

4.4 Exemplo de Utilização

Apresentamos a seguir exemplo de utilização da função de programação do RTC no controlador HI. No *website* da HI Tecnologia (www.hitecnologia.com.br) pode-se obter o seguinte programa de exemplo de utilização :

- HIPJW002 - programa de exemplo para **SPDSW** – Utilização do RTC nos controladores HI, com exemplo de leitura e programação do RTC.

Para uma melhor explanação sobre a função de programação do RTC mostra-se a seguir exemplo de utilização desta função via o bloco SCB. Neste caso, considera-se :

- um controlador HI, com suporte para RTC e com firmware que possua suporte para a respectiva programação do RTC.
- Programação da seguinte data e hora no RTC do controlador:

“Sexta-feira, 18/Abr/2005, as 12:10:00 horas”

Para realizar a programação do RTC no controlador HI deve-se :

1. Inserir um bloco SCB no programa ladder, considerando, como exemplo, o seguinte mapa de memórias:
 - P1 = K40 : Constante com valor 65, indicando função de programação do RTC
 - P2 = M50 : Buffer com os parâmetros de entrada
 - P3 = M60 : Buffer com os parâmetros de saída
 - P4 = M61 : Código de retorno da execução da função

Param.	Parâmetro	Valor	Significado do Valor do Parâmetro
P1:	K40	65	Função do bloco SCB para programação do RTC
P2:	M50	0	Valor dos SEGUNDOS a serem programados no RTC (0..59)
	M51	10	Valor dos MINUTOS a serem programados no RTC (0..59)
	M52	12	Valor das HORAS a serem programadas no RTC (1..23)
	M53	6	Valor do dia da SEMANA a ser programado no RTC (1..7)
	M54	18	Valor dos DIA do mês a ser programado no RTC (1..31)
	M55	4	Valor dos MES a ser programado no RTC (1..12)
	M56	2005	Valor do ANO a ser programado no RTC
P3:	M60		Parâmetro <i>dummy</i> , não utilizado nesta função de programação do RTC
P4:	M61		Código de retorno da execução da função

2. Programar as memórias M50..M56 com os valores necessários para a configuração da função SCB.
3. Habilitar a entrada (E1) do bloco SCB, para programar o RTC.
4. Aguardar a saída (S1) do bloco SCB torna-se ativa, indicando que programação foi executada com sucesso, caso contrário, a saída (S1) permanecerá inativa, porem um código de erro será armazenado na memória M61

5. Utilização de RTC em mensagens da IHM

O Relógio de tempo real permite disponibilizar informações sobre Data e Hora em mensagens de eventos para serem mostradas na IHM, ou para serem impressas em relatórios gerados em impressoras do tipo serial. Os formatos disponíveis para as informações de data e hora são os seguintes:

Formatos disponíveis	
dd/mm/aa	Data
hh:mm	Hora
dd/mm/aa hh:mm	Data e Hora

Tabela 4 – Formatos de apresentação da Data e/ou Hora


Ou seja, é possível mostrar em relatórios impressos ou em telas da IHM a informação de data e hora nos formatos mostrados na tabela anterior. **(Observação: Para maiores informações sobre a utilização de um impressora para gerar relatórios, consultar a nota de aplicação ENA.00013 – “Impressão matricial com os controladores HI”, disponível para download em nosso site: www.hitecnologia.com.br).**


Nesta seção, vamos orientar quanto à utilização de informações de data e hora em mensagens de IHM (Interface Homem-Máquina) da HI Tecnologia.

Em linhas gerais, a utilização de informações de data e de hora nas mensagens da IHM é simples. Em uma tela de mensagem de evento, basta inserir um “TAG” e nele fazer referência a uma das memórias determinadas como parâmetros do RTC. Em seguida, deve-se mudar o “Formato de apresentação” para que a informação seja mostrada em uma dentre as várias formas possíveis. **(Observação: Para maiores informações sobre “TAG’s” ou programação de mensagens na IHM, consultar a nota de aplicação ENA.00004 – “Programação da IHM no SPDSW”, disponível para download em nosso site: www.hitecnologia.com.br).**

Vamos mostrar mais detalhadamente como configurar uma mensagem de evento para mostrar informações de data e de hora. Nos passos a seguir, nós vamos mostrar como inserir um bloco do RTC, inserir uma mensagem de IHM e também como configurar o “TAG” para mostrar as informações de data e de hora nas mensagens da IHM.

5.1 Programação para mensagens da IHM via SPDSW

1. Abra o SPDSW e crie um projeto novo ou abra algum projeto já existente.
2. Abra o editor Ladder
3. Posicione o cursor célula desejada, dê um clique no menu “Especiais” e dê um clique sobre o botão  relógio de tempo real (RTC).

4. Para este caso utilize as memórias inteiras M0000 e M0001 e M0002, respectivamente no primeiro, segundo e terceiro parâmetros.
5. Uma vez inserido o bloco do relógio de tempo real, devemos agora inserir uma mensagem de evento. Posicione o cursor na ultima coluna da linha desejada que irá receber o bloco de mensagem de evento.
6. Dê um clique no menu “Especiais” e dê um clique no botão . Surgirá na tela o bloco de mensagem de evento, clique com o botão direito do mouse sobre o bloco de mensagem de evento e selecione propriedades onde poderemos especificar novo TAG e qual será o formato para ser mostrado na IHM, para isso clique com o botão direito dentro do editor de mensagem da IHM e selecione Novo Tag.
7. Quando inserimos um TAG, este deve fazer referência a uma memória inteira (tipo M), porém para cada formato de apresentação das informações de data e hora, o TAG deve fazer referência para a memória definida para o RTC sendo que cada parâmetro deste contém um tipo de informação.

A tabela abaixo nos orienta quanto às memórias dos parâmetros do TAG que devem ser referenciadas para cada formato de apresentação diferente:

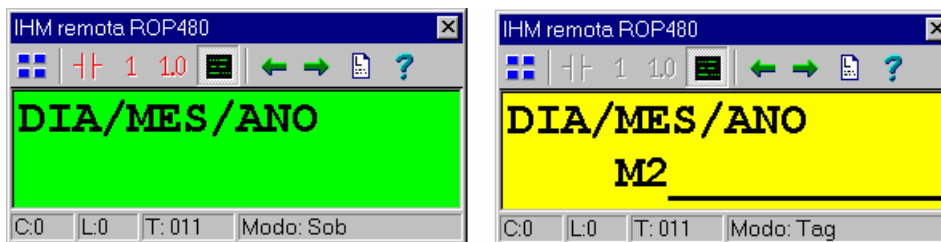
Formatos disponíveis		Parâmetro a ser referenciado	Memórias utilizadas no RTC neste exemplo
dd/mm/aa	Data	P3	M2
hh:mm	Hora	P2	M1
dd/mm/aa hh:mm	Data e Hora	P2	M1

Tabela 5 – Formatos disponíveis e os parâmetros associados a ele

Exemplos:

Vamos supor que queremos mostrar uma mensagem na tela da IHM contendo informações de data e hora. Existem três formatos de apresentação disponíveis, que estão listados na tabela acima.

Ex. 1 - Se quisermos mostrar a informação de data no formato “dd/mm/aa”, devemos criar um TAG que faça referência à memória que foi determinada para o parâmetro P3, que no nosso exemplo é M0002 e alterar o Formato de apresentação conforme configuração apresentada a seguir:





Tela 11, Var: 0

Tipo de dado associado
Memória inteira direta

Formato de apresentação
[dd/mm/aa] - Data

Id. 2

Linha 1

Coluna 4

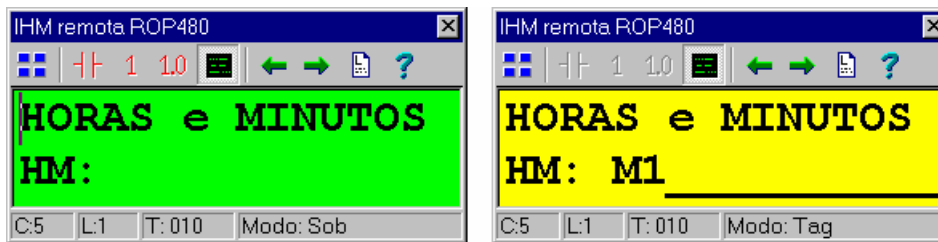
Campo 12

Refresh automático

Modo
 Supervisão
 Programação

Cancela Fecha

Ex. 2 – Se quisermos mostrar a informação de hora no formato “hh:mm”, devemos criar um TAG que faça referência à memória que foi determinada para o parâmetro P2, que no nosso exemplo é M0001 e alterar o Formato de apresentação conforme configuração apresentada a seguir:



Tela 10, Var: 0

Tipo de dado associado
Memória inteira direta

Formato de apresentação
[hh:mm] - Hora

Id. 1

Linha 1

Coluna 4

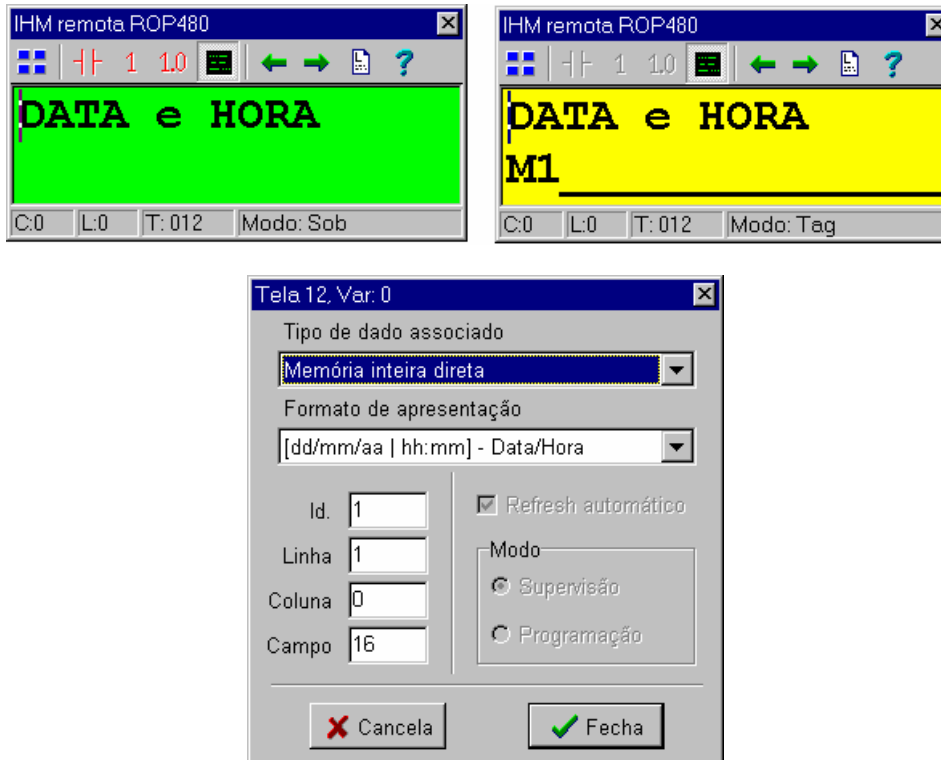
Campo 12

Refresh automático

Modo
 Supervisão
 Programação

Cancela Fecha

Ex. 3 – Se quisermos mostrar a informação de data e hora no formato “dd/mm/aa hh:mm”, devemos criar um TAG que faça referência à memória que foi determinada para o parâmetro P2, que no nosso exemplo é M0002 e alterar o Formato de apresentação conforme configuração apresentado a seguir:



Observação: Note que deve haver uma correspondência rigorosa entre a memória do parâmetro especificado e o formato de apresentação escolhido, do contrário poderá haver inexatidão nas informações mostradas na mensagem da IHM. Para não haver erro, siga as orientações da tabela anterior sobre os formatos de apresentação.

5.2 Programação para mensagens da IHM via SPDS7

1. Abra o SPDS7 e crie um projeto novo ou abra algum projeto já existente.
2. Abra o editor Ladder
3. Posicione o cursor na célula desejada com três linhas disponíveis, dê um clique no menu "Especiais" e dê um clique sobre o botão Relógio de Tempo Real (RTC).
4. Para este caso utilize as memórias inteiras M0000 e M0001 e M0002, respectivamente no primeiro, segundo e terceiro parâmetros.
5. Uma vez inserido o bloco do relógio de tempo real, devemos agora inserir uma mensagem de evento. Posicione o cursor na célula que irá receber o bloco de mensagem de evento.
6. Dê um clique no menu "Especiais" e dê um clique em Mensagens de Evento. Surgirá na tela uma mensagem de evento, onde poderemos especificar a variável a ser mostrada na IHM.
7. Posicione o cursor em cima da mensagem de evento e pressione as teclas ALT + T, veja a seguir os exemplos:

Exemplos:

Vamos supor que queremos mostrar uma mensagem na tela da IHM contendo informações de data e hora. Existem três formatos de apresentação disponíveis e que estão listados na tabela acima.

Ex. 1 - Se quisermos mostrar a informação de data no formato “dd/mm/aa”, devemos especificar a memória que faça referência ao parâmetro P3 e o formato de apresentação que queremos, para isso deve-se colocar o seguinte comando [**@F2,2**],

- @ - indica que queremos apresentar o conteúdo de uma variável;
- F2 - indica o formato de apresentação, que no caso será “dd/mm/aa”;
- 2 - indica o número da memória que foi especificada no parâmetro P3 que no nosso exemplo é M0002.

A seguir é apresentada a como deve ficar a programação da mensagem de evento:

```
Display do MCI
DIA/MES/ANO
@f2,2
```

Ex. 2 - Se quisermos mostrar a informação de data no formato “hh:mm”, devemos especificar a memória que faça referência ao parâmetro P2 e o formato de apresentação que queremos, para isso deve-se colocar o seguinte comando [**@F1,1**],

- @ - indica que queremos apresentar o conteúdo de uma variável;
- F1 - indica o formato de apresentação, que no caso será “hh:mm”;
- 1 - indica o número da memória que foi especificada no parâmetro P2 que no nosso exemplo é M0001.

A seguir é apresentada a como deve ficar a programação da mensagem de evento:

```
Display do MCI
HORAS e MINUTOS
HM: @F1,1
```

Ex. 3 - Se quisermos mostrar a informação de data no formato “dd/mm/aa hh:mm”, devemos especificar a memória que faça referência ao parâmetro P2 e o formato de apresentação que queremos, para isso deve-se colocar o seguinte comando [**@F3,1**],

- @ - indica que queremos apresentar o conteúdo de uma variável;
- F3 - indica o formato de apresentação, que no caso será “dd/mm/aa hh:mm”;
- 1 - indica o número da memória que foi especificada no parâmetro P2 que no nosso exemplo é M0001.

A seguir é apresentada a como deve ficar a programação da mensagem de evento:

```
Display do MCI
DATA e HORA
@F3,1
```



Utilização do relógio de tempo real RTC

Tipo de Doc.: Notas de Aplicação
Referência: ENA.00007

Revisão: 1
Atualizado em: 18/04/2005

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	21/10/2002	Ricardo L. Oliveira	
Revisão	16/04/2005	Paulo C. M. Inazumi	<i>Revisado em mídia</i>
Aprovação	18/04/2005	Isaías M. C. Ribeiro	<i>Aprovado em mídia</i>

Histórico de Revisões

18/04/2005	1	Acréscimo da função SCB de programação do RTC
01/07/2003	0	Documento original
Data	Rev	Descrição