



HI tecnologia

Indústria e Comércio Ltda

Notas de Aplicação

**Interface de Controladores HI
GII e GII-Duo com Dispositivos
via Protocolo ASCII**

HI Tecnologia

Documento de acesso público

ENA.00043

Versão 1.05

junho-2009

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.00043
Versão Documento: 1.05

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

Este documento abrange os seguintes Controladores nas plataformas especificadas abaixo.

Equipamentos	Família	Modelo	Plataformas			Abrangência
			GI	GII	GII_DUO	
Controladores	MCI02	MCI02	X			
		MCI02-QC	X			
	ZAP500	ZAP500	X			
		ZAP500-BX	X			
		ZAP500-BXH	X			
		ZTK500	X			
		ZTK501	X			
	ZAP900	ZAP900		X		√
		ZAP901		X		√
		eZAP900		X		√
		eZAP901		X		√
		ZAP900-BXH		X		√
		ZTK900		X		√
		eZTK900		X		√
	FLEX950	FLEX950-PLC		X		√
	P7C	P7C			X	√
IHMs	MMI600	MMI600		X		
		MMI601		X		
	MM650	MMI650		X		
	MMI800	MMI800		X		
	FLEX950	FLEX950-IHM		X		



Interface de Controladores HI GII e GII-Duo com Dispositivos via Protocolo ASCII

Referência: ENA.00043
Arquivo : ENA0004300.doc

Revisão: 5
Atualizado em: 22/06/2009

Índice

1.	Introdução	5
2.	Referências	5
3.	Interface ASCII com Controladores G-II e G-II Duo	6
3.1	Cabo de Comunicação	6
3.2	Configuração do Protocolo ASCII nos Controladores G-II e G-II Duo	7
3.2.1	Protocolo ASCII	7
3.2.2	Caractere de início de <i>frame</i> (SOF)	8
3.2.3	Caractere de fim de <i>frame</i> (EOF)	9
3.2.4	Parâmetros de comunicação	9
3.3	Quando utilizar o protocolo ASCII	10
3.3.1	Dispositivos que enviam informações em forma de Texto	10
3.3.2	Dispositivos que recebem informações em forma de Texto	10
3.3.3	Operação de <i>frames</i> não ASCII	11
4.	Funções de Comunicação via Protocolo ASCII	11
4.1	Descrição da Função SCB – Operações de Comunicação ASCII	12
4.2	Parâmetros do Bloco SCB	12
4.3	Operação do Bloco SCB	14
4.4	Selecionando o Bloco SCB no Ambiente SPDSW	14
4.5	Função Enviar Texto pela Serial do Controlador	16
4.5.1	Comentários	16
4.5.2	Exemplo	17
4.6	Função Receber Texto pela Serial do Controlador	18
4.6.1	Comentários	18
4.6.2	Exemplo	19
4.7	Função Enviar e Receber Texto pela Serial do Controlador	20
4.7.1	Comentários	20
4.7.2	Exemplo	21
4.8	Função Receber e Enviar Texto pela Serial do Controlador	22
4.8.1	Comentários	22
4.8.2	Exemplo	23
4.9	Função Programar o Estado da Linha de Controle RTS	24
4.9.1	Comentários	24
4.9.2	Exemplo	25
4.10	Códigos de Retorno do Bloco SCB	26
4.11	Tabela de Códigos de Retorno Associados à Comunicação	26
5.	ANEXO 1 – Tabela ASCII	29
	Controle do Documento	30
	Considerações gerais	30
	Responsabilidades pelo documento	30



1. Introdução

Este documento tem como objetivo fornecer informações sobre como trocar dados com dispositivos remotos que possuam protocolo ASCII, utilizando controladores geração G-II e G-II Duo da HI Tecnologia, como por exemplo, com os controladores ZAP-900, ZAP-901 e P7C.

Com esta funcionalidade, dispositivos remotos tais como leitores de código de barras, cartões magnéticos, ópticos, *buttons* podem ser conectados aos controladores da HI Tecnologia e os dados obtidos por estes dispositivos, armazenados, processados ou acessados por sistemas supervisórios.

O documento é dividido nas seguintes seções:

- Apresentação das funções de envio / recepção de *frames* texto;
- Apresentação das funções de gerência de comunicação nos Controladores HI, geração G-II e G-II Duo

O documento foi concebido visando transmitir ao usuário as informações relativas à utilização das funções de comunicação com uma rápida apresentação e fixando as informações por meio de exemplos práticos.

2. Referências

Notas de Aplicação:

- ENA.00022 Configuração dos Canais de Comunicação dos Controladores HI (formato PDF).
- ENA.00029 Interface Controlador HI com Leitor de Código de Barras [ZAP-500] (formato PDF).
- ENA.00042 Utilização de Variáveis do Tipo Texto nos Controladores G-II (formato PDF).

Programas de Exemplo (em ambiente SPDSW):

- HIPJW023 – Interface com leitor de código de barras (Controlador ZAP500)
- HIPJW031 – Utilização de movimentação de variáveis do tipo texto nos controladores HI, geração GII. Baseado em funções baseadas no bloco MOV.
- HIPJW032 – Utilização de operações sobre variáveis do tipo texto nos controladores HI, geração GII. Baseado em funções mapeadas no bloco SCB.
- HIPJW034 – Interface dos controladores HI geração GII com dispositivos via protocolo ASCII.

Controlador ZAP-900/901:

- PET.108.001 Folha de Especificação Técnica do ZAP-900/901 (formato PDF).

Controlador P7C:

- PMU 107.001 Manual do usuário P7C (formato PDF).

Todos os documentos e programas de exemplos referenciados acima estão disponíveis para “*download*” em nosso *site*: www.hitecnologia.com.br.

As funcionalidades para interface com dispositivos via protocolo ASCII estão disponíveis a partir das seguintes versões:

- Aplicativo SPDSW: Versão 2.5.00 ou superior
- Controlador ZAP 900: *Firmware* versão 1.7.13 ou superior
- Controlador ZAP 901: *Firmware* versão 1.7.13 ou superior
- Controlador P7C: *Firmware* versão 1.0.07 ou superior

3. Interface ASCII com Controladores G-II e G-II Duo

Normalmente, dispositivos que possuem protocolo ASCII, tais como leitores de código de barras, cartões magnéticos, etc., utilizam como meio de conexão uma interface serial RS232-C que pode ser conectada a qualquer uma das duas interfaces seriais disponíveis nos controladores HI – G-II e G-II Duo, conforme indicado na figura a seguir:



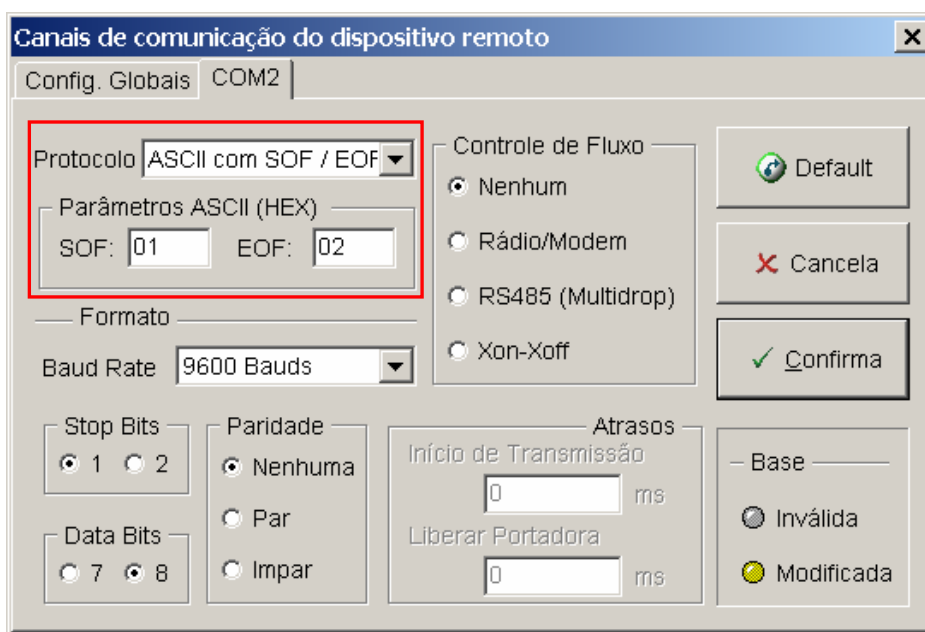
3.1 Cabo de Comunicação

Verifique na documentação fornecida juntamente com o seu dispositivo remoto a pinagem utilizada para disponibilizar os sinais da interface serial RS232-C. No [site da HI Tecnologia \(www.hitecnologia.com.br\)](http://www.hitecnologia.com.br) tem-se disponíveis a especificação dos pinos associados aos canais de comunicação de nossos controladores G-II e G-II Duo.

3.2 Configuração do Protocolo ASCII nos Controladores G-II e G-II Duo

A primeira etapa no processo de comunicação entre o dispositivo remoto e os controladores da HI Tecnologia é certificar-se que os parâmetros de comunicação dos dois equipamentos estão configurados de maneira compatível.

Para tanto, verifique o *setup* do dispositivo remoto a ser utilizado, e para a configuração da comunicação nos controladores HI, utilize o aplicativo SDPSW, selecione a opção “Ferramentas | Comunicação | Serial do Controlador”. Ao selecionar esta opção apresenta-se o seguinte formulário:



Neste formulário tem-se a possibilidade de configurar o tipo de protocolo ASCII e os demais parâmetros de configuração. Para tal selecione o canal serial do controlador a ser configurado (COM1, COM2 ou COM3, se disponível), e após o tipo de protocolo ASCII a ser utilizado. Os parâmetros de comunicação são descritos em maiores detalhes a seguir.

3.2.1 Protocolo ASCII

Na opção “Protocolo”, associada ao formulário do canal serial especificado, selecione o tipo de protocolo ASCII a ser utilizado. Neste caso, temos as seguintes opções:

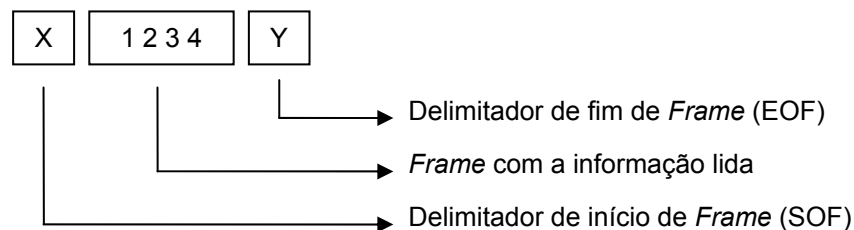


- ASCII: Protocolo ASCII configurado sem caractere de início de *frame* (SOF) e sem caractere de fim de *frame* (EOF)
- ASCII com SOF: Protocolo ASCII configurado com um caractere de início de *frame* (SOF) e sem caractere de fim de *frame* (EOF)
- ASCII com EOF: Protocolo ASCII configurado sem caractere de início de *frame* (SOF) e com um caractere de fim de *frame* (EOF)
- ASCII com SOF / EOF: Protocolo ASCII configurado com um caractere de início de *frame* (SOF) e um caractere de fim de *frame* (EOF)

Se o protocolo ASCII selecionado necessite de caracteres SOF e / ou EOF, os seus respectivos códigos também devem ser especificados, conforme descrito abaixo.

3.2.2 Caractere de início de *frame* (SOF)

O dispositivo remoto envia um *frame* texto para o controlador da HI Tecnologia, com a possibilidade de acrescentar um caractere delimitador de início de *frame*, conforme indicado, na figura a seguir:



Formato do *frame* 1234 enviado pelo dispositivo remoto

Os dispositivos remotos podem ou não enviar caracteres de início e fim de *frame*. Para operação correta da comunicação entre o dispositivo remoto e o controlador da HI Tecnologia é necessário que ambos estejam configurados da mesma forma. Verifique na documentação do dispositivo remoto, e se necessário, defina e programe um caractere para indicar o início de *frame*. Note que se este dispositivo for um leitor de código de barras, este caractere deve ser diferente dos caracteres a serem lidos pelo leitor, e também diferente do caractere delimitador de fim de *frame*, se existir.

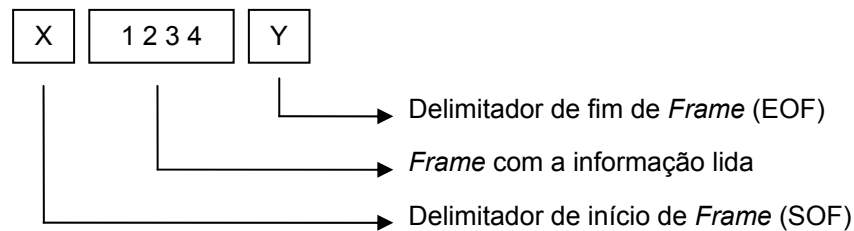
Exemplos:

- Se o caractere SOF for STX (*start of text*), deve-se especificar o valor 2 (hexadecimal) no campo SOF
- Se o caractere SOF for “#” deve-se especificar o valor 8D (hexadecimal) no campo SOF

Obs: Ver tabela ASCII no Anexo 1 deste documento.

3.2.3 Caractere de fim de *frame* (EOF)

De maneira semelhante, a possibilidade de especificar um caractere de início de *frame*, também existe a possibilidade de acrescentar um caractere delimitador de fim de *frame* conforme indicado na figura a seguir:



Formato do *frame* 1234 enviado pelo dispositivo remoto

Os dispositivos remotos podem ou não enviar caracteres de início e fim de *frame*. Para operação correta da comunicação entre o dispositivo remoto e o controlador da HI Tecnologia é necessário que ambos estejam configurados da mesma forma. Verifique na documentação do dispositivo remoto, e se necessário, defina e programe um caractere para indicar o fim de *frame*. Note que se este dispositivo for um leitor de código de barras, este caractere deve ser diferente dos caracteres a serem lidos pelo leitor, e também diferente do caractere delimitador de início de *frame*, se existir.

Exemplos:

- Se o caractere EOF for ETX (*end of text*) deve-se especificar o valor 3 (hexadecimal) no campo EOF
- Se o caractere EOF for CR (*carriage return*) deve-se especificar o valor D (hexadecimal) no campo EOF

Obs: Ver tabela ASCII no Anexo 1 deste documento.

3.2.4 Parâmetros de comunicação

Certifique-se que o valor selecionado do “*baud rate*” e do formato do caractere (tamanho, paridade e *stop bits*) para o dispositivo remoto e do controlador da HI Tecnologia são iguais. Os valores *default* destes parâmetros para os controladores da HI Tecnologia são:

- *Baud rate* : 9600
- *Data bits* : 8
- *Stop bits* : 1
- Paridade: sem paridade.

Verifique se o dispositivo remoto necessita de sinais de controle de fluxo para a troca de dados, por exemplo, de sinais de controle RTS, etc., e consulte a especificação do controlador HI utilizado, para identificar qual canal serial possui suporte para as linhas de controle necessárias para interfacear com o dispositivo remoto.



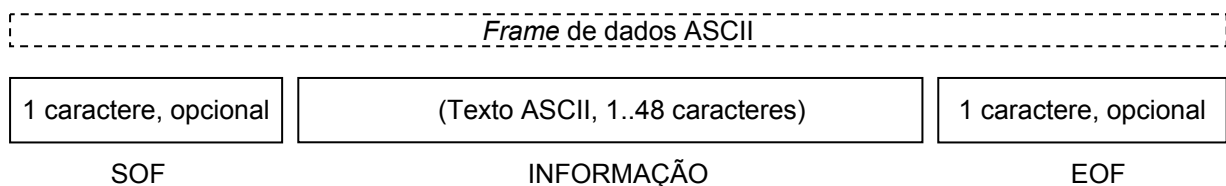
Para maiores informações sobre a configuração dos canais de comunicação dos controladores da HI Tecnologia consulte a Nota de Aplicação ENA.00022, disponível para “download” em nosso [site www.hitecnologia.com.br](http://www.hitecnologia.com.br).

3.3 Quando utilizar o protocolo ASCII

O protocolo ASCII é normalmente utilizado para interfacear dispositivos que recebem ou enviam informações em forma de texto. A necessidade de seleção do protocolo ASCII para o respectivo canal de comunicação é dependente do tipo de recurso de comunicação a ser utilizado, conforme detalhado a seguir.

3.3.1 Dispositivos que enviam informações em forma de Texto

Estes dispositivos obtêm informações do processo, formatam estes dados como um Texto e enviam através de canais de comunicação para serem tratados. Como exemplo podemos citar, leitores de código de barra, *scanners*, leitores de cartões magnéticos, leitores biométricos etc. Dependendo do equipamento utilizado, o texto enviado pode ser delimitado com caracteres especiais no início e/ou no final do mesmo. Desta forma o formato geral do *frame* é o seguinte:



Para receber *frames* com este formato, é necessário que o controlador utilizado, tenha a sua porta de comunicação, configurada para operação com protocolo ASCII.

A especificação dos caracteres SOF e EOF, bem como a habilitação dos mesmos é realizada na configuração do canal de operação associado. Quando habilitada operação com SOF e/ou EOF estes caracteres são filtrados pelo *driver* de recepção, não sendo armazenados na variável texto Xnnn associada.

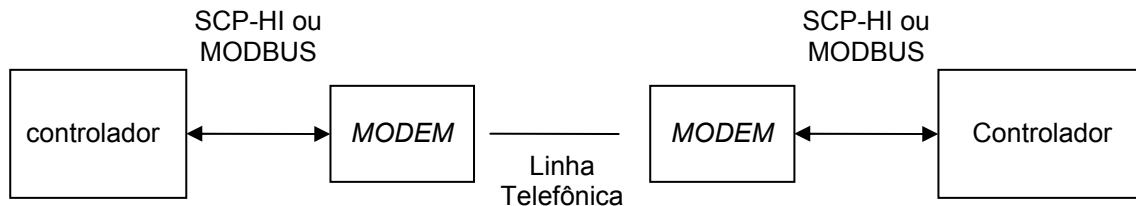
3.3.2 Dispositivos que recebem informações em forma de Texto

Estes dispositivos aguardam a recepção de *frames* de Textos pelo canal de comunicação. Como exemplo podemos citar, impressoras, *modems*, painéis de *displays* de *Led's* e etc. Neste caso, o *frame* de informação a ser enviado deve ser criado em uma variável de texto (Xnnnn) qualquer do equipamento GII.

A utilização ou não de caracteres de início de *frame* (SOF) ou final de *frame* (EOF) é definida pelo protocolo do equipamento que será conectado e, quando necessária, devem ser inseridos no texto antes do envio pelo canal de comunicação, mesmo que estejam especificados caracteres SOF e EOF no tipo de protocolo ASCII selecionado.

Note-se então que para enviarmos *frames* ASCII para um dispositivo remoto, os eventuais caracteres de início de *frame* (SOF) e de final de *frame* (EOF), mesmo que configurados no tipo de protocolo ASCII no controlador GII, NÃO são automaticamente enviados na transmissão do *frame* ASCII, pois estes caracteres são válidos apenas para a recepção de *frames* ASCII. Quando for necessário enviar *frames* com estes caracteres SOF e/ou EOF estes devem ser inseridos na variável texto a ser enviada.

É importante notar que, para o envio de um *frame* ASCII qualquer, não é necessário que o canal de comunicação associado esteja configurado como ASCII. A função SCB de transmissão de dados ASCII esta operacional independentemente do tipo de protocolo configurado para o canal de comunicação. Desta forma, pode-se, por exemplo, utilizar um canal serial do equipamento, configurado no protocolo SCP-HI (ou Modbus-RTU) para enviar informações de conexão para um *modem* e posteriormente se comunicar com o protocolo configurado, conforme ilustrado a seguir:



O bloco SCB de envio de *frames* ASCII pode ser utilizado para configurar e enviar comandos para o *modem* associado em qualquer dos controladores.

3.3.3 Operação de frames não ASCII

Todas as aplicações que utilizam comunicação nos exemplos citados anteriormente montam os *frames* de informação utilizando caracteres ASCII padrão. Entretanto o *frame* recebido ou enviado pelas funções de comunicação não necessita necessariamente de ser ASCII. Qualquer seqüência de valores (*bytes*) pode ser montada e transmitida pelo canal de comunicação bem como, recebida pelo mesmo.

Em outras palavras, o protocolo ASCII permite envio e recepção de um conjunto de *bytes* em qualquer formato. A única restrição imposta pelo *driver* é que a quantidade de dados recebida ou enviada em cada transação de comunicação (envio ou recepção) não seja superior a 48 *bytes*, devido a limitação do tamanho das variáveis "X" do tipo texto.

4. Funções de Comunicação via Protocolo ASCII

Para troca de dados com dispositivos remotos via protocolo ASCII, utilizaremos um conjunto de funções do controlador através do bloco de controle SCB. Este bloco provê uma interface padronizada com funções que permitem enviar e receber dados para o dispositivo remoto. Os dados recebidos do dispositivo podem ser transferidos para memórias do controlador da HI Tecnologia, e posteriormente acessados pelo programa de aplicação corrente.

Neste caso temos as seguintes funções disponíveis:

- Enviar *Frame* Texto pela Serial do Controlador
- Receber *Frame* Texto pela Serial do Controlador
- Programar o Estado da Linha de Controle RTS do Controlador

A seguir, realiza-se a apresentação mais detalhada de cada uma destas funções de comunicação.

4.1 Descrição da Função SCB – Operações de Comunicação ASCII

Em aplicações desenvolvidas no ambiente SPDSW deve-se utilizar o bloco SCB (“Bloco de Controle Padrão”) para acesso as funções de comunicação ASCII nos controlador HI geração G-II e G-II Duo. O bloco SCB é composto por:

- 1 entrada E1
- 4 parâmetros (P1, P2, P3 e P4), e
- 1 saída S1

A figura abaixo ilustra um bloco SCB no programa de aplicação (*ladder*):

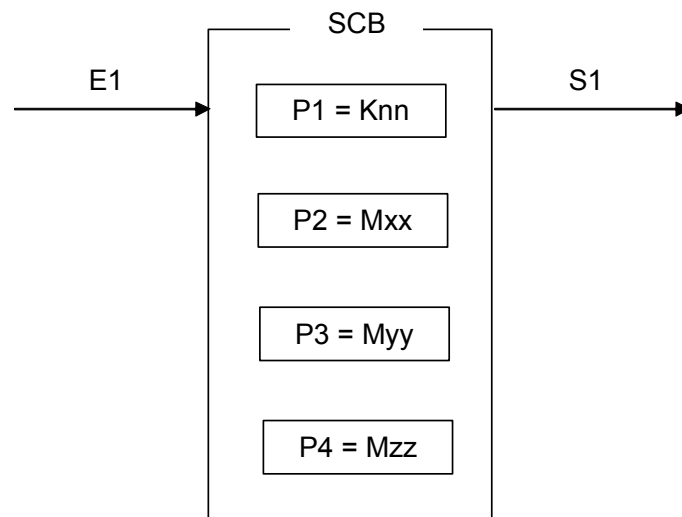


Figura - Bloco "SCB"

A seguir descrevem-se os parâmetros associados ao bloco SCB, bem como a sua respectiva operação.

4.2 Parâmetros do 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 240 para Enviar *Frame* Texto pela Serial do Controlador
 - Igual a 241 para Receber *Frame* Texto pela Serial do Controlador
 - Igual a 231 para Programar o Estado da Linha de Controle RTS do Controlador

- Parâmetro P2: Deve ser uma memória inteira do tipo **Mnn**, a partir dessa deve existir uma seqüência de memórias consecutivas com parâmetros de entrada associados à função selecionada. A quantidade de parâmetros é dependente de cada função que se deseja utilizar. Exemplo: Sendo Mxx igual a M10, a seqüência será M10, M11, M12,, etc, até a quantidade de parâmetros necessários a função especificada.
- Parâmetro P3: Deve ser uma memória inteira do tipo **Mxx**, a partir dessa deve existir uma seqüência de memórias consecutivas com parâmetros de saída associados à função selecionada. A quantidade de parâmetros é dependente de cada função que se deseja utilizar Exemplo: Sendo Mxx igual a M20, a seqüência será M20, M21, M22,, etc, até a quantidade de dados necessários a cada função especificada.
- Parâmetro P4: Deve ser uma memória inteira do tipo **Myy**, Nesta memória será retornado o código de retorno da execução da função especificada. 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.
 - Desenergizado - Bloco desabilitado, não executa nenhuma função.

Observação: Cabe ressaltar que na funcionalidade deste bloco SCB, o tratamento das operações sobre estas funções de comunicação, somente é executado a cada **transição de ativação** desta entrada E1, devendo permanecer ativa até o término da execução da função indicado quando a saída S1 torna-se energizada.
- Saída S1: sinal de saída do bloco SCB, onde:
 - Energizado - Término da execução da função de comunicação especificada.
 - Desenergizado – Bloco não habilitado, ou está habilitado, mas ainda não concluiu a execução da função de comunicação especificada.

Observação: Ao término da execução de função de comunicação deve-se testar o respectivo código de retorno referente ao resultado da execução da função. Para tal, vide especificação do parâmetro P4 deste bloco SCB.



4.3 Operação do Bloco SCB

Para utilização deste bloco SCB, deve-se seguir a seguinte seqüência de operação:

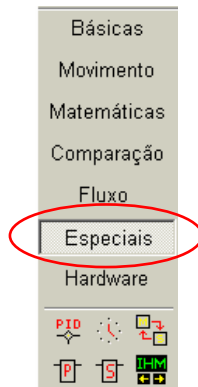
1. Preencher os parâmetros P1, P2, P3 e P4 de acordo com cada função de comunicação a ser utilizada, ou seja, para cada função existe um conjunto de parâmetros específicos necessários para a correta execução da função.
2. Para ativar a função de comunicação especificada do tipo texto deve-se gerar uma transição de subida na entrada E1. Esta entrada E1 deve permanecer ativa até o término da execução da função, indicado quando a saída S1 torna-se energizada.
3. Aguardar o término da execução da função selecionada, indicada quando a saída S1 torna-se ativa.
4. Após o término da execução da função deve-se analisar o código de retorno referente à execução da função retornado na memória especificada no parâmetro P4. Em linhas gerais, se o código de retorno for 0 (ZERO) indica que a função foi executada com sucesso, caso contrário indica o código de erro associado à execução da função selecionada. Se a função foi executada com sucesso, e se a mesma possuir parâmetros de resposta, pode-se analisá-los nas memórias especificadas no parâmetro P3 deste bloco SCB.
5. Recomenda-se desabilitar a entrada E1 do bloco SCB, de modo que em sua próxima ativação seja gerada uma transição de subida, tal qual descrito no passo 2 acima

4.4 Selecionando o Bloco SCB no Ambiente SPDSW

No ambiente SPDSW, para inserir um bloco SCB em seu programa de aplicação (*ladder*), deve-se proceder como descrito a seguir:

- 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.

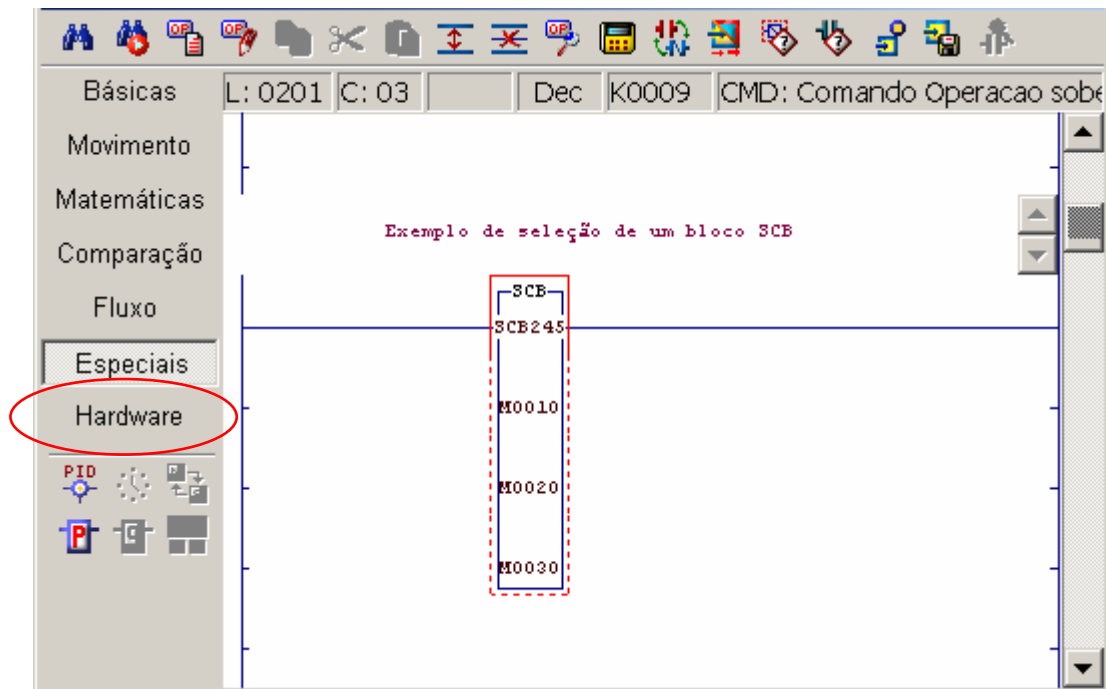


Figura – Exemplo de Edição de um Bloco "SCB" no ambiente SPDSW



4.5 Função Enviar Texto pela Serial do Controlador

Função

Enviar Texto

Plataforma de Controlador

Esta função está disponível para controladores das famílias **GII** e **GII-Duo** da HI Tecnologia.

Descrição da Função

Enviar Texto pela Serial do Controlador da HI Tecnologia

P1: Parâmetros de Entrada

Constante Knn Função do bloco SCB para Enviar Texto pela Serial do Controlador
Deve ser uma constante inteira (Knn) cujo valor deve ser 240

P2: Parâmetros de Entrada

Memória xx + 0 Identificador do canal de comunicação do controlador da HI Tecnologia utilizado para enviar o *frame* texto para o dispositivo remoto.

O código associado ao canal de comunicação depende de cada modelo de controlador da HI Tecnologia [veja OBS. B]

Memória xx + 1 Identificador da variável Texto (X0..Xn) cujo conteúdo corresponde ao *frame* texto a ser enviado pela porta serial do controlador para o dispositivo remoto

P3: Parâmetros de Saída

Memória yy + 0 Sem parâmetros de saída, parâmetro “*dummy*”.

P4: Parâmetros de Saída

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

4.5.1 Comentários

[OBS. A] A operação especificada neste bloco SCB somente é realizada a cada transição da entrada E1 deste bloco SCB. Assim, a cada operação desejada deve-se gerar uma transição de subida nesta entrada E1 em sua respectiva aplicação *ladder*.

[OBS. B] O código associado ao canal de comunicação depende do controlador a ser utilizado para realizar a comunicação ASCII. Por exemplo :

- ZAP900 => 0:COM1 ou Ethernet, 1:COM2, 2:COM3 (se disponíveis)
- P7C => -1:Ethernet, 0:COM1, 1:COM2, 2:COM3 (se disponíveis)

4.5.2 Exemplo

Este exemplo ilustra como enviar o conteúdo de uma variável do tipo texto através da porta de comunicação COM1 do controlador. Neste exemplo, utilizaremos a variável texto X0. Para realizar esta função de comunicação deve-se:

1. Inserir um bloco SCB no programa *ladder*, considerando, como exemplo, o seguinte mapa de memórias:
 - P1 = K0 : Constante com valor 240, indicando função de envio de um *frame* texto
 - P2 = M0 : *Buffer* com os parâmetros de entrada
 - P3 = M10 : *Buffer* com os parâmetros de saída
 - P4 = M11 : Código de retorno da execução da função

Param.	Parâmetro	Valor	Significado do Valor do Parâmetro
P1:	K0	240	Função do bloco SCB de envio de um <i>frame</i> texto
P2:	M0	0	Identificador do canal de comunicação, utilizado para enviar o <i>frame</i> texto, no caso, selecionamos a COM1 do controlador ZAP900, então devemos especificar o valor numérico 0 (zero)
	M1	0	Identificador da variável Texto (X0..Xn) cujo conteúdo corresponde ao <i>frame</i> texto a ser enviado pela porta serial do controlador, no caso, a variável X0, então deve-se especificar o valor 0 (zero).
P3:	M10		Sem parâmetro de saída, parâmetro não utilizado (<i>dummy</i>).
P4:	M11		Código de retorno da execução da função

2. Programar as memórias M0..M1 com os valores necessários para a configuração da função SCB;
3. Habilitar a entrada (**E1**) do bloco SCB, para programar e habilitar a função SCB;
4. Aguardar a saída (**S1**) do bloco SCB tornar-se ativa, indicando que a operação foi executada;
5. Analisar o a memória M11 para verificar se houve erro ou não na execução da função. Se o valor for o (ZERO) indica função executada com sucesso. Caso contrario, deve-se analisar o código de erro especificado em M11 e realizar o seu devido tratamento;
6. Para uma nova execução da função SCB, deve-se gerar um novo pulso na entrada (**E1**) do bloco.



4.6 Função Receber Texto pela Serial do Controlador

Função

Receber Texto

Plataforma de Controlador

Esta função está disponível para controladores das famílias **GII** e **GII-Duo** da HI Tecnologia.

Descrição da Função

Receber Texto pela Serial do Controlador da HI Tecnologia

P1: Parâmetros de Entrada

Constante Knn Função do bloco SCB para Receber Texto pela Serial do Controlador
Deve ser uma constante inteira (Knn) cujo valor deve ser 241

P2: Parâmetros de Entrada

Memória xx + 0 Identificador do canal de comunicação do controlador utilizado para receber o *frame* texto do dispositivo remoto

O código associado ao canal de comunicação depende de cada modelo de controlador da HI Tecnologia [veja OBS. B]

P3: Parâmetros de Saída

Memória yy + 0 Identificador da variável Texto (X0..Xn) onde será armazenado o *frame* texto recebido do dispositivo remoto através do canal de comunicação do controlador.

P4: Parâmetros de Saída

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

4.6.1 Comentários

[OBS. A] A operação especificada neste bloco SCB somente é realizada a cada transição da entrada E1 deste bloco SCB. Assim, a cada operação desejada deve-se gerar uma transição de subida nesta entrada E1 em sua respectiva aplicação *ladder*.

[OBS. B] O código associado ao canal de comunicação depende do controlador a ser utilizado para realizar a comunicação ASCII. Por exemplo :

- ZAP900 => 0:COM1 ou Ethernet, 1:COM2, 2:COM3 (se disponíveis)
- P7C => -1:Ethernet, 0:COM1, 1:COM2, 2:COM3 (se disponíveis)

4.6.2 Exemplo

Este exemplo ilustra como receber, através da porta de comunicação COM1 do controlador, um *frame* texto de um dispositivo remoto, armazenando o conteúdo recebido em uma variável do tipo texto. Neste exemplo, utilizaremos a variável texto X1 para armazenar o texto recebido. Para realizar esta função de comunicação deve-se:

1. Inserir um bloco SCB no programa *ladder*, considerando, como exemplo, o seguinte mapa de memórias:
 - P1 = K0: Constante com valor 241, indicando função de envio de um *frame* texto
 - P2 = M0: *Buffer* com os parâmetros de entrada
 - P3 = M10: *Buffer* com os parâmetros de saída
 - P4 = M11: Código de retorno da execução da função

Param.	Parâmetro	Valor	Significado do Valor do Parâmetro
P1:	K0	241	Função do bloco SCB de recepção de um <i>frame</i> texto
P2:	M0	0	Identificador do canal de comunicação, utilizado para receber o <i>frame</i> texto, no caso, selecionamos a COM1 do controlador ZAP900, então devemos especificar o valor numérico 0 (zero)
P3:	M10	1	Identificador da variável Texto (X0..Xn) onde será armazenado o conteúdo do <i>frame</i> texto recebido do dispositivo remoto, no caso, a variável X1, então deve-se especificar o valor 1
P4:	M11		Código de retorno da execução da função

2. Programar a memória M0 e M10 com o valor necessário para a configuração da função SCB;
3. Habilitar a entrada (**E1**) do bloco SCB, para programar e habilitar a função SCB;
4. Aguardar a saída (**S1**) do bloco SCB tornar-se ativa, indicando que a operação foi executada;
5. Analisar o a memória M11 para verificar se houve erro ou não na execução da função. Se o valor for o (ZERO) indica função executada com sucesso, e assim pode-se processar o conteúdo da variável X1 com os caracteres recebidos. Caso contrario, deve-se analisar o código de erro especificado em M11 e realizar o seu devido tratamento;
6. Para uma nova execução da função SCB, deve-se gerar um novo pulso na entrada (**E1**) do bloco.



4.7 Função Enviar e Receber Texto pela Serial do Controlador

Função

Enviar e Receber Texto

Plataforma de Controlador

Esta função está disponível para controladores das famílias **GII** e **GII-Duo** da HI Tecnologia.

Descrição da Função

Primeiro envia texto pela Serial do Controlador da HI Tecnologia e posteriormente fica aguardando a recepção de um texto de resposta.

P1: Parâmetros de Entrada

Constante Knn Função do bloco SCB para Enviar e Receber Texto pela Serial do Controlador Deve ser uma constante inteira (Knn) cujo valor deve ser 242

P2: Parâmetros de Entrada

Memória xx + 0 Identificador do canal de comunicação do controlador utilizado para enviar e receber o *frame* texto para o dispositivo remoto.

O código associado ao canal de comunicação depende de cada modelo de controlador da HI Tecnologia [veja OBS. B]

Memória xx + 1 Identificador da variável Texto (X0..Xn) cujo conteúdo corresponde ao *frame* texto a ser enviado pelo canal de comunicação do controlador para o dispositivo remoto

P3: Parâmetros de Saída

Memória yy + 0 Identificador da variável Texto (X0..Xn) onde será armazenado o *frame* texto recebido do dispositivo remoto através do canal de comunicação do controlador. Este *frame* texto é recebido como resposta do dispositivo remoto após o envio do *frame* texto especificado no parâmetro de entrada "Memória xx + 1"

P4: Parâmetros de Saída

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

4.7.1 Comentários

[OBS. A] A operação especificada neste bloco SCB somente é realizada a cada transição da entrada E1 deste bloco SCB. Assim, a cada operação desejada deve-se gerar uma transição de subida nesta entrada E1 em sua respectiva aplicação *ladder*.

[OBS. B] O código associado ao canal de comunicação depende do controlador a ser utilizado para realizar a comunicação ASCII. Por exemplo :



- ZAP900 => 0:COM1 ou Ethernet, 1:COM2, 2:COM3 (se disponíveis)
- P7C => -1:Ethernet, 0:COM1, 1:COM2, 2:COM3 (se disponíveis)

4.7.2 Exemplo

Este exemplo ilustra como enviar o conteúdo de uma variável do tipo texto através da porta de comunicação COM1 do controlador, e automaticamente ficar aguardando um texto de resposta do respectivo dispositivo remoto. Neste exemplo, utilizaremos o conteúdo da variável texto X0 do controlador da HI Tecnologia para ser enviada para o dispositivo remoto, e armazenaremos o texto de resposta recebido do respectivo dispositivo remoto na variável texto X2 do controlador da HI Tecnologia. Para realizar esta função de comunicação deve-se:

1. Inserir um bloco SCB no programa *ladder*, considerando, como exemplo, o seguinte mapa de memórias:
 - P1 = K0 : Constante com valor 242, indicando função de envio de um *frame* texto
 - P2 = M0 : *Buffer* com os parâmetros de entrada
 - P3 = M10 : *Buffer* com os parâmetros de saída
 - P4 = M11 : Código de retorno da execução da função

Param.	Parâmetro	Valor	Significado do Valor do Parâmetro
P1:	K0	242	Função do bloco SCB para envio e recepção de um <i>frame</i> texto
P2:	M0	0	Identificador do canal de comunicação, utilizado para enviar e receber o <i>frame</i> texto, no caso, selecionamos a COM1 do controlador ZAP900, então devemos especificar o valor numérico 0 (zero)
	M1	0	Identificador da variável Texto (X0..Xn) cujo conteúdo corresponde ao <i>frame</i> texto a ser enviado pela porta serial do controlador, no caso, a variável X0, então deve-se especificar o valor 0 (zero).
P3:	M10	2	Identificador da variável Texto (X0..Xn) onde será armazenado o conteúdo do <i>frame</i> texto recebido do dispositivo remoto, no caso, a variável X2, então deve-se especificar o valor 2
P4:	M11		Código de retorno da execução da função de comunicação

2. Programar as memórias M0, M1 e M10 com os valores necessários para a configuração da função SCB;
3. Habilitar a entrada (**E1**) do bloco SCB, para programar e habilitar a função SCB;
4. Aguardar a saída (**S1**) do bloco SCB tornar-se ativa, indicando que a operação foi executada;
5. Analisar o a memória M11 para verificar se houve erro ou não na execução da função. Se o valor for o (ZERO) indica função executada com sucesso. Caso contrario, deve-se analisar o código de erro especificado em M11 e realizar o seu devido tratamento;
6. Para uma nova execução da função SCB, deve-se gerar um novo pulso na entrada (**E1**) do bloco.



4.8 Função Receber e Enviar Texto pela Serial do Controlador

Função

Receber e Enviar Texto

Plataforma de Controlador

Esta função está disponível para controladores das famílias **GII** e **GII-Duo** da HI Tecnologia.

Descrição da Função

Primeiro aguarda a recepção de um *frame* texto do dispositivo remoto através da Serial do Controlador da HI Tecnologia, e posteriormente envia um outro *frame* texto de resposta.

P1: Parâmetros de Entrada

Constante Knn Função do bloco SCB para Receber e Enviar Texto pela Serial do Controlador
Deve ser uma constante inteira (Knn) cujo valor deve ser 243

P2: Parâmetros de Entrada

Memória xx + 0 Identificador do canal de comunicação do controlador utilizado para receber e enviar *frame* texto para o dispositivo remoto.

O código associado ao canal de comunicação depende de cada modelo de controlador da HI Tecnologia [veja OBS. B]

Memória xx + 1 Identificador da variável Texto (X0..Xn) cujo conteúdo corresponde ao *frame* texto a ser enviado pelo canal de comunicação do controlador para o dispositivo remoto, após a recepção do *frame* texto do mesmo.

P3: Parâmetros de Saída

Memória yy + 0 Identificador da variável Texto (X0..Xn) onde será armazenado o *frame* texto recebido do dispositivo remoto através do canal de comunicação do controlador.

P4: Parâmetros de Saída

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

4.8.1 Comentários

[OBS. A] A operação especificada neste bloco SCB somente é realizada a cada transição da entrada E1 deste bloco SCB. Assim, a cada operação desejada deve-se gerar uma transição de subida nesta entrada E1 em sua respectiva aplicação *ladder*.

[OBS. B] O código associado ao canal de comunicação depende do controlador a ser utilizado para realizar a comunicação ASCII. Por exemplo :

- ZAP900 => 0:COM1 ou Ethernet, 1:COM2, 2:COM3 (se disponíveis)



- P7C => -1:Ethernet, 0:COM1, 1:COM2, 2:COM3 (se disponíveis)

4.8.2 Exemplo

Este exemplo ilustra como receber um texto do dispositivo remoto através da porta de comunicação COM1 do controlador, e automaticamente enviar um texto de resposta para o respectivo dispositivo remoto.

Neste exemplo, utilizaremos a variável texto X2 do controlador da HI Tecnologia para armazenar o texto recebido do respectivo dispositivo remoto, e posteriormente utilizaremos o conteúdo da variável texto X0 do controlador da HI Tecnologia para ser enviada como resposta para o dispositivo remoto. Para realizar esta função de comunicação deve-se:

1. Inserir um bloco SCB no programa *ladder*, considerando, como exemplo, o seguinte mapa de memórias:
 - P1 = K0 : Constante com valor 243, indicando função de envio de um *frame* texto
 - P2 = M0 : *Buffer* com os parâmetros de entrada
 - P3 = M10 : *Buffer* com os parâmetros de saída
 - P4 = M11 : Código de retorno da execução da função

Param.	Parâmetro	Valor	Significado do Valor do Parâmetro
P1:	K0	243	Função do bloco SCB para recepção e envio de um <i>frame</i> texto
P2:	M0	0	Identificador do canal de comunicação, utilizado para receber e enviar o <i>frame</i> texto, no caso, selecionamos a COM1 do controlador ZAP900, então devemos especificar o valor numérico 0 (zero)
	M1	0	Identificador da variável Texto (X0..Xn) cujo conteúdo corresponde ao <i>frame</i> texto a ser enviado pela porta serial do controlador, no caso, a variável X0, então deve-se especificar o valor 0 (zero).
P3:	M10	2	Identificador da variável Texto (X0..Xn) onde será armazenado o conteúdo do <i>frame</i> texto recebido do dispositivo remoto, no caso, a variável X2, então deve-se especificar o valor 2
P4:	M11		Código de retorno da execução da função de comunicação

2. Programar as memórias M0, M1 e M10 com os valores necessários para a configuração da função SCB;
3. Habilitar a entrada (**E1**) do bloco SCB, para programar e habilitar a função SCB;
4. Aguardar a saída (**S1**) do bloco SCB tornar-se ativa, indicando que a operação foi executada;
5. Analisar o a memória M11 para verificar se houve erro ou não na execução da função. Se o valor for o (ZERO) indica função executada com sucesso. Caso contrario, deve-se analisar o código de erro especificado em M11 e realizar o seu devido tratamento;
6. Para uma nova execução da função SCB, deve-se gerar um novo pulso na entrada (**E1**) do bloco.



4.9 Função Programar o Estado da Linha de Controle RTS

Função

Programa Linha RTS

Plataforma de Controlador

Esta função está disponível para controladores da família **GII** da HI Tecnologia.

Descrição da Função

Realizar a programação do estado da linha de controle RTS de um canal serial de um controlador da HI Tecnologia.

P1: Parâmetros de Entrada

Constante Knn Função do bloco SCB de funções de gerência de comunicação, Deve ser uma constante inteira (Knn) cujo valor deve ser 230.

P2: Parâmetros de Entrada

Memória xx + 0 Código do sub-comando associado à função de comunicação. No caso, a função para programação da linha de controle RTS corresponde ao código numérico 1

Memória xx + 1 Identificador do canal de comunicação do controlador na qual será programado o estado da linha de controle RTS.

O código associado ao canal de comunicação depende de cada modelo de controlador da HI Tecnologia [veja OBS. B]

Note que o canal serial selecionado deve possuir suporte para a linha de controle RTS. Verifique a especificação do controlador da HI Tecnologia utilizado para saber quais canais de comunicação possuem este sinal RTS disponível.

Memória xx + 2 Estado a ser programado o sinal RTS, onde:
. valor igual a zero : Programa sinal RTS INATIVO (*OFF*)
. qualquer valor diferente de zero : Programa sinal RTS ATIVO (*ON*)

P3: Parâmetros de Saída

Memória yy + 0 Sem parâmetros de saída, parâmetro “dummy”.

P4: Parâmetros de Saída

Memória zz + 0 Código de retorno da execução da operação sobre texto, onde :
Igual a 0 Indica operação executada com sucesso.
Diferente de 0 Indica condição de falha. Vide a lista de códigos de retorno

4.9.1 Comentários

[OBS. A] A operação especificada neste bloco SCB somente é realizada a cada transição da entrada E1 deste bloco SCB. Assim, a cada operação desejada deve-se gerar uma transição de subida nesta entrada E1 em sua respectiva aplicação *ladder*.

[OBS. B] O código associado ao canal de comunicação depende do controlador a ser utilizado para realizar a comunicação ASCII. Por exemplo :

- ZAP900 => 0:COM1 ou Ethernet, 1:COM2, 2:COM3 (se disponíveis)
- P7C => -1:Ethernet, 0:COM1, 1:COM2, 2:COM3 (se disponíveis)

4.9.2 Exemplo

Este exemplo ilustra como programar a linha de controle RTS do canal de comunicação COM1 para o estado LIGADO. Para realizar esta operação deve-se:

1. Inserir um bloco SCB no programa *ladder*, considerando, como exemplo, o seguinte mapa de memórias:
 - P1 = K0 : Constante com valor 230, indicando função de função de gerência de comunicação
 - P2 = M0 : *Buffer* com os parâmetros de entrada
 - P3 = M10 : *Buffer* com os parâmetros de saída
 - P4 = M11 : Código de retorno da execução da função

Param.	Parâmetro	Valor	Significado do Valor do Parâmetro
P1:	K0	230	Função do bloco SCB função de gerência de comunicação
P2:	M0	1	Código do sub-comando de comunicação associado a programação da linha RTS, no caso, código numérico 1
	M1	0	Identificador do canal de comunicação a ser programado o estado da linha RTS, no caso, selecionamos a COM1 do controlador ZAP900, então devemos especificar o valor numérico 0 (zero)
	M2	1	Estado a programar o sinal RTS, no caso, desejamos ativar o sinal RTS, então devemos especificar qualquer valor diferente de zero, por exemplo, o valor numérico 1
P3:	M10		Sem parâmetro de saída
P4:	M11		Código de retorno da execução da função

2. Programar as memórias M0..M2 com os valores necessários para a configuração da função SCB;
3. Habilitar a entrada (E1) do bloco SCB, para programar e habilitar a função SCB;
4. Aguardar a saída (S1) do bloco SCB tornar-se ativa, indicando que a operação foi executada;
5. Analisar o a memória M11 para verificar se houve erro ou não na execução da função. Se o valor for o (ZERO) indica função executada com sucesso. Caso contrario, deve-se analisar o código de erro especificado em M11 e realizar o seu devido tratamento;
6. Para uma nova execução da função SCB, deve-se gerar um novo pulso na entrada (E1) do bloco.

4.10 Códigos de Retorno do Bloco SCB

Após a execução de uma função SCB para as funções de comunicação descritas neste documento, deve-se verificar no parâmetro P4 do referido bloco SCB o respectivo código de retorno resultante da execução da função. De maneira geral, o código de retorno 0 (ZERO) indica função executada com sucesso, caso contrário o valor numérico retornado corresponde a um código de erro detectado durante a execução da função.

Código de Retorno	Descrição do Código de Retorno
12	Canal de comunicação não está configurado com protocolo de comunicação ASCII
24	<i>Driver</i> de comunicação está "ocupado" com outro <i>frame</i> para transmissão. Já existe outro <i>frame</i> em processo de transmissão.
27	Variável texto a ser enviada como <i>frame</i> texto é nula, não possui caracteres
74	Canal de comunicação especificado não disponível no controlador
206	Variável do tipo texto especificada como parâmetro, não definida na aplicação. Por exemplo: a variável texto que contém o <i>frame</i> texto a ser enviado, ou a variável texto especificada para armazenar o valor do <i>frame</i> texto recebido. Por exemplo, configurou-se 10 variáveis do tipo texto na aplicação, e em alguma função especificou-se uma variável com identificador maior que 9, por exemplo especificar variável X10, X11, X12, etc.. são variáveis inválidas.

4.11 Tabela de Códigos de Retorno Associados à Comunicação

Código de Retorno	Descrição do Código de Retorno
64	Falha na criação dos semáforos de comunicação
65	<i>Buffer</i> de transmissão cheio
66	Falha no acesso aos dados solicitados
67	<i>Baud rate</i> inválido
68	<i>Overflow</i> no <i>buffer</i> de transmissão
69	<i>Overrun</i> no <i>buffer</i> de recepção
70	Erro genérico de comunicação
71	Canal de comunicação aberto (ativo)

72	Canal de comunicação fechado (inativo)
73	Canal inativo
74	Identificador do canal inválido
75	Identificador da estação inválido
76	Condição de <i>break</i> detectada
77	Tamanho do <i>frame</i> inválido
78	<i>Timeout</i> na transmissão de dados
79	<i>Timeout</i> na recepção de dados
80	<i>Timeout</i> na resposta de equipamento externo (<i>Modem</i>)
81	Erro de protocolo
82	Erro de <i>overrun</i>
83	Erro de paridade
84	Erro de <i>framing</i>
85	Erro de CRC
86	Dado inválido no protocolo
87	Comunicação interrompida
88	<i>Flag</i> de início de <i>frame</i> inválido
89	<i>Flag</i> de fim de <i>frame</i> inválido
90	Identificador do numero da estação do <i>frame</i> de resposta inválido
91	Comunicação temporariamente em pausa, esta ocupada ou impossibilitada de responder
92	Falha de inicialização no processo de comunicação
93	<i>Overflow</i> no campo de dados do <i>frame</i> de comunicação
94	<i>Timeout</i> na recepção de <i>frame</i>
95	2 <i>stop bits</i> não suportado pelo ambiente
96	Paridade não suportada pelo ambiente
97	Controle de fluxo não suportado pelo ambiente
98	Operação <i>multidrop</i> não suportado pelo ambiente
99	Canal não suportado pelo ambiente
100	Comando do <i>modem</i> executado com Sucesso
101	<i>Modem</i> conectado
102	<i>Ring</i>
103	<i>No Carrier</i>
104	Erro na execução do comando
106	<i>No Dial Tone</i>
107	<i>Busy</i>



108	<i>No Answer</i>
110	Conectado em 2400
111	<i>Ringin</i>
112	Código de resposta do <i>modem</i> verbal ou <i>ECHO ON</i>

5. ANEXO 1 – Tabela ASCII

Visando facilitar a consulta dos códigos de caracteres ASCII, segue tabela:

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	:	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Source: www.LookupTables.com



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	07/08/2006	Paulo C. M. Inazumi	
Revisão	22/06/2009	Wendel L. Bonfá / Paulo C. M. Inazumi	<i>Revisado em mídia</i>
Aprovação	22/06/2009	Isaias M. C. Ribeiro	<i>Aprovado em mídia</i>

Histórico de Revisões

22/06/2009	5	Inclusão das funções de envio e recepção ASCII simultâneas (funções 242 e 243)
07/04/2009	4	Acrescentadas referências ao controlador P7C
13/07/2007	3	Revisão do item 3.3.2
17/10/2006	2	Correção no código ASCII dos caracteres STX e ETX
04/09/2006	1	Inserção do parágrafo de “Necessidade do Protocolo ASCII”
14/08/2006	0	Documento original
Data	Rev	Descrição