



HI tecnologia

Indústria e Comércio Ltda

Notas de Software

Driver serial SCP para LINUX

HI Tecnologia

Documento de acesso publico

Apresentação

Este documento descreve como implementar comunicação serial com a família de controladores HI em plataformas LINUX, utilizando a biblioteca de comunicação fornecida pela HI Tecnologia.

Esta nota de software 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 Software: PNS.0003
Versão Documento: 1.00

HI Tecnologia Indústria e Comércio Ltda.

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

Cidade: Campinas – SP
CEP: 13090-150

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

E-mail: hi@hitecologia.com.br

Web site: www.hitecologia.com.br



Referência: PNS.0003
Arquivo : PNS0000300.doc

Revisão: 0
Atualizado em: 12/12/2002

Índice

1.	Introdução.....	4
1.1	Informação Copyright.....	4
1.2	Disclaimer	4
1.3	Novas versões	4
1.4	Realimentação	4
2.	Utilizando a Biblioteca	4
2.1	Definição de condicional	5
2.2	Funções da Biblioteca.....	5
	SCPOpenPort	5
	SCPclosePort:	6
	ResetDriver:	6
	SCPVersionRevision:.....	7
	ConfigComChannel:.....	7
	SCPCheckConnection:	8
	SCPCheckConnection:	8
	SCPReadData:.....	8
	SCPWriteData:.....	9
	ModemResetOperation:	10
	ModemCheckOperation:	10
	ModemConnect:.....	11
	ModemDisconnect:	11
3.	Configuração da porta serial	12
3.1	Sintaxe	12
3.2	Sessão PC_PORT_SESSION.....	13
3.3	Sessão PC_SERVER_SESSION	13
4.	Tratamento de Exceções.....	14
5.	Exemplo de comunicação serial.....	15
6.	Outras fontes de Informação.....	16
	Controle do Documento	17
	Considerações gerais	17
	Responsabilidades pelo documento.....	17



Driver serial SCP para LINUX

Tipo de Doc.: Notas de Software
Referência: PNS.0003

Revisão: 0
Atualizado em: 12/12/2002

1. Introdução

Os controladores da HI Tecnologia tradicionalmente possuem suporte para supervisão em ambiente Windows (dlls e bibliotecas). A crescente demanda pelo sistema operacional LINUX determinou o desenvolvimento de interfaces de comunicação serial apropriadas também para esse sistema operacional. Este documento descreve o resultado deste desenvolvimento através da apresentação das funções desenvolvidas e de um exemplo de utilização.

1.1 Informação Copyright

Este documento é propriedade da HI Tecnologia © 2002, sendo distribuído de acordo com os termos apresentados a seguir. Este documento pode ser distribuído no seu todo, ou em partes, em qualquer meio físico ou eletrônico, desde que os direitos de copyright sejam mantidos em todas as cópias.

1.2 Disclaimer

A utilização dos conceitos, exemplos e outros elementos deste documento é responsabilidade exclusiva do usuário.

1.3 Novas versões

Novas versões são continuamente liberadas no *site* da HI Tecnologia.

1.4 Realimentação

Realimentações são bem vindas. Por favor, envie seus comentários para suporte@hitecnologia.com.br

2. Utilizando a Biblioteca

A biblioteca de comunicação serial é distribuída com os seguintes elementos:

Tabela 1 FILE

Arquivo	Descrição
HiScpSta.a	Contém a biblioteca estática que deve ser ligada ao seu aplicativo
ScpLibFunc.h	Inclui com a declaração da interface da biblioteca
hiDefines.h	Inclui com os defines de constantes e exceções da HI Tecnologia
cmRetCodeh	Códigos de erro associados à comunicação serial
ScpSpec.psf	Arquivo de configuração da porta serial



Driver serial SCP para LINUX

Tipo de Doc.: Notas de Software
Referência: PNS.0003

Revisão: 0
Atualizado em: 12/12/2002

2.1 Definição de condicional

A biblioteca de comunicação serial possui interface multi plataforma, dessa forma, o mesmo aplicativo desenvolvido no WINDOWS pode ser executada no ambiente LINUX sem necessidade de alterações no código relativas a utilização da biblioteca.

Para alcançar esse objetivo foi utilizado um *#define* para cada sistema operacional. Assim para execução nas plataformas windows você deve definir o condicional "WINDOWS" e para execução em ambiente linux você deve definir o condicional "LINUX".

No caso específico do sistema operacional LINUX, essa definição pode ser alcançada da seguinte forma:

- compilador gcc: utilizar opção *-DLINUX*
- compilador Kylix Borland: definir o condicional "LINUX" no menu Projects->Options, tab Directories/Conditionals da interface IDE.

2.2 Funções da Biblioteca

SCPOpenPort:

SINOPSE

```
#include <hiDefines.h>  
#include <ScpLibFunc.h>  
int SCPOpenPort (char *PSFFilename);
```

DESCRIÇÃO

Abre a porta de comunicação serial utilizando o arquivo de configuração *PSFileName* para programar os atributos da porta serial.

O arquivo *ScpSpec.psf* distribuído junto com a biblioteca deve ser utilizado como referência para a construção de um novo arquivo de configuração apropriado para sua aplicação.

RETORNO

Retorna 0 em caso de sucesso e código de erro em caso de falha.

ERRO

O detalhamento do código de erro retornado é apresentado no arquivo *cmRetCode.h* presente na distribuição da biblioteca.

VEJA TAMBÉM

SCPClosePort



Driver serial SCP para LINUX

Tipo de Doc.: Notas de Software
Referência: PNS.0003

Revisão: 0
Atualizado em: 12/12/2002

SCPClosePort:

SINOPSE

```
#include <hiDefines.h>  
#include <ScpLibFunc.h>  
int SCPClosePort(void);
```

DESCRIÇÃO

Fecha a porta de comunicação retornando sua configuração para os valores originais anteriores a abertura com a função *SCPOpenPort*.

RETORNO

Retorna 0 em caso de sucesso e código de erro em caso de falha.

ERRO

O detalhamento do código de erro retornado é apresentado no arquivo *cmRetCode.h* presente na distribuição da biblioteca.

VEJA TAMBÉM

SCPOpenPort

ResetDriver:

SINOPSE

```
#include <hiDefines.h>  
#include <ScpLibFunc.h>  
int ResetDriver(void);
```

DESCRIÇÃO

Reinicia a porta de comunicação reprogramando os mesmos parâmetros fornecidos na função de abertura da porta.

RETORNO

Retorna 0 em caso de sucesso e código de erro em caso de falha.

ERRO

O detalhamento do código de erro retornado é apresentado no arquivo *cmRetCode.h* presente na distribuição da biblioteca.

VEJA TAMBÉM

SCPOpenPort ; *SCPClosePort*



Driver serial SCP para LINUX

Tipo de Doc.: Notas de Software
Referência: PNS.0003

Revisão: 0
Atualizado em: 12/12/2002

SCPVersionRevision:

SINOPSE

```
#include <hiDefines.h>
#include <ScpLibFunc.h>
int SCPVersionRevision(int &Version, int &Revision);
```

DESCRIÇÃO

Obtém a versão e a revisão da biblioteca de comunicação utilizada.

Os parâmetros de entrada *Version* e *Revision* retornam o valor atual da versão e revisão respectivamente.

RETORNO

Retorna 0 em caso de sucesso e código de erro em caso de falha.

ERRO

O detalhamento do código de erro retornado é apresentado no arquivo *cmRetCode.h* presente na distribuição da biblioteca.

ConfigComChannel:

SINOPSE

```
#include <hiDefines.h>
#include <ScpLibFunc.h>
int ConfigComChannel(char* FileName);
```

DESCRIÇÃO

Reconfigura a porta de comunicação serial utilizando o arquivo de configuração *PSFileName* para programar os atributos da porta serial. O arquivo *ScpSpec.psf* distribuído junto a biblioteca deve ser utilizado como referência para a construção de um novo arquivo de configuração apropriado para sua aplicação.

Note que para o perfeito funcionamento desta função é necessário que a porta de comunicação tenha sido previamente aberta através da função *SCPOpenPort*.

RETORNO

Retorna 0 em caso de sucesso e código de erro em caso de falha.

ERRO

O detalhamento do código de erro retornado é apresentado no arquivo *cmRetCode.h* presente na distribuição da biblioteca.

VEJA TAMBÉM

SCPOpenPort



Driver serial SCP para LINUX

Tipo de Doc.: Notas de Software
Referência: PNS.0003

Revisão: 0
Atualizado em: 12/12/2002

SCPCheckConnection:

SINOPSE

```
#include <hiDefines.h>  
#include <ScpLibFunc.h>  
int SCPCheckConnection(int Station);
```

DESCRIÇÃO

Verifica se a conexão com a porta serial está operacional.

RETORNO

Retorna 0 em caso da porta está operando normalmente e código de erro em caso de falha.

ERRO

O detalhamento do código de erro retornado é apresentado no arquivo cmRetCode.h presente na distribuição da biblioteca.

SCPCheckConnection:

SINOPSE

```
#include <hiDefines.h>  
#include <ScpLibFunc.h>  
int SCPCheckConnection(int Station);
```

DESCRIÇÃO

Verifica se a conexão com a porta serial está operacional.

RETORNO

Retorna 0 em caso da porta está operando normalmente e código de erro em caso de falha.

ERRO

O detalhamento do código de erro retornado é apresentado no arquivo cmRetCode.h presente na distribuição da biblioteca.

SCPReadData:

SINOPSE

```
#include <hiDefines.h>  
#include <ScpLibFunc.h>  
int SCPReadData(int Station, int TypeVar, int VarIni, int VarQuant, byte *DataBuf);
```

DESCRIÇÃO

Realiza a leitura de variáveis dos controladores da HI Tecnologia.

Para efetivar essa operação os seguintes parâmetros são necessários:



Driver serial SCP para LINUX

Tipo de Doc.: Notas de Software
Referência: PNS.0003

Revisão: 0
Atualizado em: 12/12/2002

Station: determina o endereço da estação sendo acessada

TypeVar: define o tipo de dado da variável sendo acessada (R - contato auxiliar, M - Memória inteira, D - Memória Ponto Flutuante). O arquivo *ScpLibFunc.h* define as seguintes constantes que devem ser utilizadas na definição deste parâmetro:

- R_TYPE // define tipo de variável R
- M_TYPE // define tipo de variável M
- D_TYPE // define tipo de variável D

VarIni: define o endereço inicial da variável sendo acessada.

VarQuant: define a quantidade de posições sequenciais a partir de *VarIni* que devem ser processadas pelo comando.

DataBuf: ponteiro para buffer de dados que receberá o resultado da operação de leitura.

RETORNO

Retorna 0 em caso de sucesso na operação e código de erro em caso de falha.

ERRO

O detalhamento do código de erro retornado é apresentado no arquivo *cmRetCode.h* presente na distribuição da biblioteca.

VEJA TAMBÉM

SCPWriteData

SCPWriteData:

SINOPSE

```
#include <hiDefines.h>
#include <ScpLibFunc.h>
int SCPWriteData(int Station, int TypeVar, int VarIni, int VarQuant, byte *DataBuf);
```

DESCRIÇÃO

Realiza a escrita em variáveis dos controladores da HI Tecnologia. Para efetivar essa operação os seguintes parâmetros são necessários:

Station: determina o endereço da estação sendo acessada

TypeVar: define o tipo de dado da variável sendo acessada (R - contato auxiliar, M - Memória inteira, D - Memória Ponto Flutuante). O arquivo *ScpLibFunc.h* define as seguintes constantes que devem ser utilizadas na definição deste parâmetro:

- R_TYPE // define tipo de variável R
- M_TYPE // define tipo de variável M



Driver serial SCP para LINUX

Tipo de Doc.: Notas de Software
Referência: PNS.0003

Revisão: 0
Atualizado em: 12/12/2002

- `D_TYPE` // define tipo de variável D

VarIni: define o endereço inicial da variável sendo acessada.

VarQuant: define a quantidade de posições sequenciais a partir de **VarIni** que devem ser processadas pelo comando.

DataBuf: ponteiro para buffer de dados que contém a informação a ser enviada para o controlador.

RETORNO

Retorna 0 em caso de sucesso na operação e código de erro em caso de falha.

ERRO

O detalhamento do código de erro retornado é apresentado no arquivo *cmRetCode.h* presente na distribuição da biblioteca.

VEJA TAMBÉM

SCPReadData

ModemResetOperation:

SINOPSE

```
#include <hiDefines.h>
#include <ScpLibFunc.h>
int ModemResetOperation(void);
```

DESCRIÇÃO

Envia comando de *reset* (ATZ) para o modem conectado à porta serial.

RETORNO

Retorna 0 em caso de sucesso na operação e código de erro em caso de falha.

ERRO

O detalhamento do código de erro retornado é apresentado no arquivo *cmRetCode.h* presente na distribuição da biblioteca.

ModemCheckOperation:

SINOPSE

```
#include <hiDefines.h>
#include <ScpLibFunc.h>
int ModemCheckOperation(void);
```



Driver serial SCP para LINUX

Tipo de Doc.: Notas de Software
Referência: PNS.0003

Revisão: 0
Atualizado em: 12/12/2002

DESCRIÇÃO

Envia o comando AT para o modem e verifica a resposta do mesmo para detectar se existe um modem operacional conectado na porta de comunicação.

RETORNO

Retorna 0 em caso de sucesso na operação e código de erro em caso de falha.

ERRO

O detalhamento do código de erro retornado é apresentado no arquivo `cmRetCode.h` presente na distribuição da biblioteca.

ModemConnect:

SINOPSE

```
#include <hiDefines.h>
#include <ScpLibFunc.h>
int ModemConnect(char* TelNumber);
```

DESCRIÇÃO

Ativa operação do modem através de conexão com o número de telefone fornecido como parâmetro (*TelNumber*). Podem ser acrescentados prefixos ao telefone para configuração da conexão (ex. T32526044 para discagem via tom).

RETORNO

Retorna 0 em caso de sucesso na operação e código de erro em caso de falha.

ERRO

O detalhamento do código de erro retornado é apresentado no arquivo `cmRetCode.h` presente na distribuição da biblioteca.

VEJA TAMBÉM

ModemDisconnect

ModemDisconnect:

SINOPSE

```
#include <hiDefines.h>
#include <ScpLibFunc.h>
int ModemDisconnect(void)
```



Driver serial SCP para LINUX

Tipo de Doc.: Notas de Software
Referência: PNS.0003

Revisão: 0
Atualizado em: 12/12/2002

DESCRIÇÃO

Envia comando de desconexão para o modem conectado à porta de comunicação.

RETORNO

Retorna 0 em caso de sucesso na operação e código de erro em caso de falha.

ERRO

O detalhamento do código de erro retornado é apresentado no arquivo *cmRetCode.h* presente na distribuição da biblioteca.

VEJA TAMBÉM

ModemConnect

3. Configuração da porta serial

Os atributos de operação da porta de comunicação serial (*baud rate*, *stop bits*, paridade, etc) são definidos através de arquivo de configuração analisado em tempo de execução do aplicativo. Para efetivar essa operação a função da biblioteca *ScpOpenPort*, que recebe como parâmetro o nome do arquivo de configuração, realiza a leitura do arquivo e a programação dos atributos da porta serial.

O arquivo de configuração possui sintaxe própria que deve ser respeitada no caso de alterações na configuração. Os próximos tópicos descrevem as sessões /sintaxe que compõe o arquivo de configuração.

3.1 Sintaxe

O arquivo de configuração está dividido em sessões e cada sessão possui um conjunto de atributos. A sintaxe para definição de sessão é apresentada abaixo:

[<nome_da_sessão>]

As seguintes sessões são definidas no arquivo de configuração serial:

- *PC_PORT_SESSION*: configuração da porta serial
- *PC_SERVER_SESSION*: configuração de comunicação via rede local (*sockets*)

O formato padrão de uma linha de configuração é apresentado abaixo:

<atributo>=<valor>

onde:

<atributo>: identificador do atributo da porta serial

<valor>: valor atual do atributo

Comentários podem ser inseridos no arquivo de configuração, desde que não contenham cadeias de caracteres aderentes aos padrões definidos para sessão e atributo. Como forma de padronização, foi adotado que as linhas que contém comentários devem iniciar pelo caractere '#'.



Driver serial SCP para LINUX

Tipo de Doc.: Notas de Software
Referência: PNS.0003

Revisão: 0
Atualizado em: 12/12/2002

3.2 Sessão PC_PORT_SESSION

A sessão PC_PORT_SESSION define os atributos de configuração da porta serial. A Tabela 2 sessão PC_PORT_SESSION apresenta detalhes desses atributos.

Tabela 2 sessão PC_PORT_SESSION

Arquivo	Descrição
pcpsPort	Define porta de comunicação [1... 8].
pcpsBaud	Define baud rate [300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200]
pcpsParity	Define paridade [0: Nenhuma, 1: Impar, 2: Par, 3:Marca, 4:Espaço].
pcpsStopBits	Define stop bits [1: 1 Stop Bits, 2 Stop Bits].
pcpsDataBits	define data bits [5: 5 Bits, 6: 6 Bits, 7: 7 Bits, 8: 8 Bits].
pcpsRS485Suport	Define RS485 Control [0: Desativado, 1: Ativado].
pcpsDataRadioSuport	Define suporte para comunicação via Data-rádio [0: Desativado, 1: Ativado].
pcpsExtraBytes	Numero de bytes adicionais no final do frame para temporizarão.
pcpsRTSExtTime	Tempo adicional de manutenção da portadora na transmissão.
pcpsCarrierOffTime	Tempo de espera para retirada da portadora do radio remoto.
pcpsModemSuport	Suporte para comunicação via modem [0: Desativado, 1: Ativado].
pcpsAutoConnect	Conexão automática [0: Desativado, 1: Ativado].
pcpsPhone	Define numero de telefone <i>default</i> para conexão.
pcpsModemPort	Define porta de comunicação associada ao modem [(1..8) [Válido quando]ModemSuport=1].
pcpsDebugActive	Define geração do arquivo de depuração [0: Desativado, 1: Ativado].
pcpsFailFramesToReset	Define número de frames consecutivos com falha para <i>reset</i> de driver.
pcpsModemCheckString	String de teste do modem.
pcpsModemResetString	String de reset do modem.
pcpsModemInitString	String de inicialização do modem.
pcpsModemDialString	String de chamada modem.
pcpsModemAnswerString	String de atendimento de chamada do modem.
pcpsModemExitString	String de término de conexão do modem.
pcpsModemHangupString	String de encerramento de chamada do modem.

3.3 Sessão PC_SERVER_SESSION

A sessão PC_SERVER_SESSION define os atributos de configuração do servidor de comunicação via rede local. . A Tabela 3 sessão PC_SERVER_SESSION apresenta detalhes desses atributos.

Tabela 3 sessão PC_SERVER_SESSION

Arquivo	Descrição
M_USE_SCP_SERVER	Flag de acesso à porta de comunicação via servidor SCP Se 0=acesso direto ao canal de comunicação local da máquina 1=acesso aos serviços de comunicação via servidor SCP
M_SCP_SERVER_CLASS	Classe do servidor de comunicação SCP
M_SCP_SERVER_NAME	Nome do servidor de comunicação SCP
M_SCP_SERVER_FILE	Localização do servidor de comunicação SCP Se <i>path</i> não for especificado, o <i>path</i> relativo ao diretório da aplicação
M_SERVER_HOST_NAME	Nome da máquina aonde reside o servidor SCP. Se for especificado endereço IP, este nome e desconsiderado.
M_SERVER_IPADD	Endereço IP da máquina onde reside o servidor SCP. Se M_SERVER_HOST_NAME e M_SERVER_IPADD não forem especificados considera comunicação local (na mesma máquina).
M_SERVER_PORT	Porta Base de comunicação.



Driver serial SCP para LINUX

Tipo de Doc.: Notas de Software
Referência: PNS.0003

Revisão: 0
Atualizado em: 12/12/2002

4. Tratamento de Exceções

O paradigma de orientação a objetos utiliza exceções para modelar situações de falha no sistema.

Como forma de permitir o maior alcance na utilização da biblioteca de suporte à comunicação serial da HI Tecnologia, o uso da estratégia de exceções foi minimizado. Dessa forma, todo o tratamento de erros é realizado através do retorno de informações diretamente na chamada da função.

Entretanto, o próprio sistema operacional gera exceções gerando situações onde um tratamento via exceções torna-se inevitável.

Para contemplar essas situações, a interface de comunicação da HI Tecnologia disponibiliza uma classe de exceções que permite identificar a causa da exceção, seja ela de responsabilidade do SO ou da própria interface. Essa classe é denominada *EScp*, como pode ser observado no código abaixo:

```
class EScp: public Exception
{
private:

public:
int HiErrCode; // codigo de falha gerado pela HI
int SoErrCode; // codigo de falha gerado pelo SO

__fastcall EScp(int iHiErrCode, int iSoErrCode):Exception(iHiErrCode),
HiErrCode(iHiErrCode),
SoErrCode(iSoErrCode){};
};
```

Exemplo de utilização dessa classe pode ser observado no programa demo disponibilizado no pacote de distribuição desta biblioteca.



Driver serial SCP para LINUX

Tipo de Doc.: Notas de Software
Referência: PNS.0003

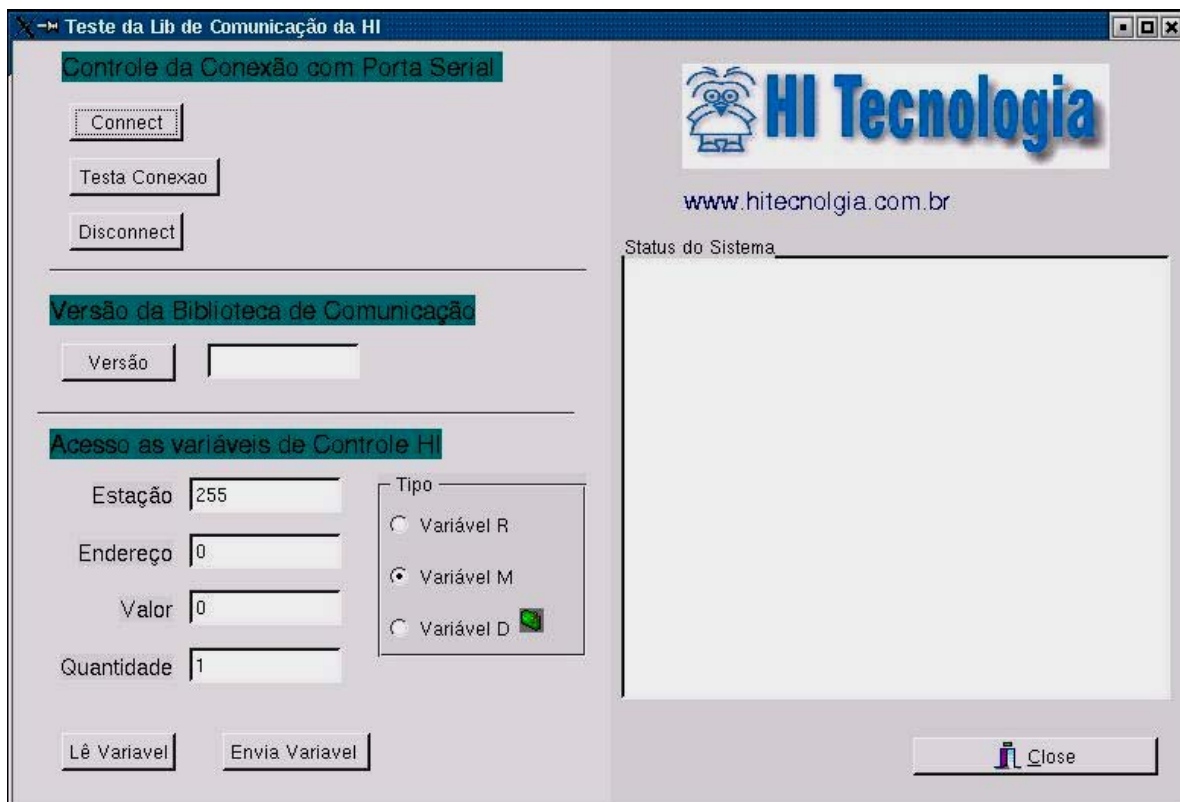
Revisão: 0
Atualizado em: 12/12/2002

5. Exemplo de comunicação serial

Um exemplo de utilização da biblioteca HiScpSta.a foi incluído na liberação do pacote de comunicação com controladores da HI Tecnologia em plataformas Linux.

Esse exemplo foi desenvolvido utilizando *Borland Kylix version 3.0 Enterprise Edition*.

A tela principal do demo da utilização da biblioteca é apresentada na figura abaixo.



De forma resumida, as seguintes opções foram disponibilizadas:

- **Connect:** Esta opção realiza a conexão com a porta serial de acordo com os parâmetros fornecidos no arquivo de configuração *ScpSpec.psf*. O resultado da execução do comando é direcionado para a tela "Status do Sistema".

Função utilizada: *SCPOpenPort*

- **Testa Conexão:** Esta opção verifica se a conexão com a porta serial esta operacional. O resultado da execução do comando é direcionado para a tela "Status do Sistema".

Função utilizada: *SCPCheckConnection*

- **Versão:** identifica a versão e o release da biblioteca. O resultado da execução do comando é direcionado o campo de edição presente no lado direito do botão "Versão".



Driver serial SCP para LINUX

Tipo de Doc.: Notas de Software
Referência: PNS.0003

Revisão: 0
Atualizado em: 12/12/2002

Função utilizada: SCPVersionRevision

- *Lê Variável:* Realiza a leitura de variáveis do controlador da HI Tecnologia considerando os parâmetros definidos pelos campos: Estação, Endereço, Valor, Quantidade. O resultado da execução do comando é direcionado para a tela "Status do Sistema".

Função utilizada: SCPReadData

- *Envia Variável:* Envia dados para o controlador da HI Tecnologia considerando os parâmetros definidos pelos campos: Estação, Endereço, Valor, Quantidade. O resultado da execução do comando é direcionado para a tela "Status do Sistema".

Função utilizada: SCPWriteData

- *Termina:* Encerra a execução do demo.

Função utilizada: SCPClosePort

6. Outras fontes de Informação

A descrição detalhada da família de controladores da HI Tecnologia, bem como utilitários, *drivers* e exemplos de utilização podem ser encontrados em nosso *site* (www.hitecnologia.com.br).

Descrições de como utilizar o sistema Operacional LINUX para desenvolver aplicativos podem ser facilmente obtidas na *Internet*. Em particular temos:

- Visão geral do sistema operacional Linux: www.linux.com
- Documentação Geral Linux: <http://www.tldp.org>
- HOWTO *compiler & linker*: <http://www.tldp.org/HOWTO/GCC-HOWTO/index.html>
- HOWTO serial Programming: <http://www.tldp.org/HOWTO/Serial-Programming-HOWTO/index.html>
- Serial Programming Guide for POSIX Operating Systems : <http://www.easysw.com/~mike/serial>



Driver serial SCP para LINUX

Tipo de Doc.: Notas de Software
Referência: PNS.0003

Revisão: 0
Atualizado em: 12/12/2002

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	06/12/2002	Helio Azevedo	
Revisão	12/12/2002	Helio J. Almeida Jr	<i>Revisado em mídia</i>
Aprovação	12/12/2002	Helio J. Almeida Jr.	<i>Aprovado em mídia</i>

Histórico de Revisões

12/12/2002	0	Documento original
Data	Rev	Descrição