



**HI** tecnologia

**Indústria e Comércio Ltda**

---

Notas de Software

Driver SCP-HI para  
Comunicação Serial em Linux

---

---

**HI Tecnologia**

---

Documento de acesso público

# Apresentação

---

Esta nota de software foi elaborada pela HI Tecnologia Indústria e Comércio Ltda. O departamento de suporte a clientes, está disponível, através do telefone (19) 2139-1700 ou do e-mail [suporte@hitecnologia.com.br](mailto:suporte@hitecnologia.com.br), para esclarecimentos sobre as informações contidas neste documento, ou para dirimir quaisquer dúvidas a respeito de nossos produtos. Favor mencionar as informações a seguir para que possamos identificar os dados relativos a este documento.

ID da Nota de Software: PNS.00024  
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@hitecnologia.com.br](mailto:hi@hitecnologia.com.br)

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

---



**Referência:** Erro! Fonte de referência não encontrada.  
**Arquivo:** PNS0002400.docx

**Revisão:** 0  
**Atualizado em:** Erro! Fonte

## Índice

1.	Introdução .....	4
1.1	Informação Copyright.....	4
1.2	Disclaimer.....	4
1.3	Sugestões .....	4
2.	Referências.....	5
2.1	Notas de Aplicação .....	5
3.	Protocolo SCP-HI.....	5
4.	Distribuição da biblioteca de comunicação .....	6
4.1	Pre-requisitos do sistema operacional Linux .....	6
5.	Lista de funções da Biblioteca Dinâmica.....	7
5.1	Descrição das funções.....	7
5.2	SCPOpenPort.....	8
5.3	SCPclosePort .....	9
5.4	SCPVersionRevision.....	10
5.5	SCPCheckConnection .....	11
5.6	SCPReadData.....	12
5.7	SCPWriteData .....	14
6.	Arquivo de configuração dos parâmetros de comunicação .....	16
6.1	Parâmetros default de operação do driver.....	16
7.	Códigos de retorno .....	17
7.1	Códigos de falha retornados pelo controlador .....	17
7.2	Códigos de falha associadas à configuração de comunicação.....	19
7.3	Códigos de falha associadas à interface de modem .....	21
7.4	Códigos de falha associadas à gerência de threads de comunicação.....	21
7.5	Códigos de falha associadas à transmissão e recepção de frames.....	22
	Anexo 1 - Exemplo de Utilização.....	23
	Controle do Documento.....	25
	Considerações gerais .....	25
	Responsabilidades pelo documento.....	25



# Driver SCP-HI para Comunicação Serial em Linux

Tipo de Doc.: Notas de Software  
Referência: PNS.00024

Revisão: 1  
Atualizado em: 07/05/2009

## 1. Introdução

---

Os controladores da HI Tecnologia possuem recursos para comunicação através de interfaces RS232-C, RS485 e Ethernet. Estes canais de comunicação são utilizados para programação, depuração e supervisão das informações contidas no programa de aplicação do controlador.

Utilizando um protocolo desenvolvido pela HI Tecnologia, denominado SCP-HI, é possível acessar a base de dados da aplicação nos controladores, permitindo programação e supervisão de parâmetros e dados do programa.

Para facilitar a utilização deste recurso de comunicação por desenvolvedores de software foi criada uma biblioteca dinâmica (.SO) para comunicação SERIAL para plataformas LINUX. Esta biblioteca implementa o protocolo SCP-HI provendo as funcionalidades básicas para acesso as variáveis dos controladores da HI Tecnologia.

Este documento apresenta as funções desta biblioteca dinâmica definindo funcionalidades, parâmetros, e sintaxe para sua correta utilização.

### 1.1 Informação Copyright

---

Este documento é propriedade da HI Tecnologia © 2009, 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 Sugestões

---

Sugestões são bem vindas. Por favor, envie seus comentários para [suporte@hitecnologia.com.br](mailto:suporte@hitecnologia.com.br)



## Driver SCP-HI para Comunicação Serial em Linux

Tipo de Doc.: Notas de Software  
Referência: PNS.00024

Revisão: 1  
Atualizado em: 07/05/2009

## 2. Referências

---

Seguem algumas referências para suporte na utilização dos produtos da HI Tecnologia associadas a protocolos de comunicação.

### 2.1 Notas de Aplicação

---

- ENA.00008 Controladores HI com Protocolo MODBUS (\*1)
- ENA.00022 Configuração dos canais de comunicação dos Controladores HI (\*1)
- ENA.00030 Integrando controladores em rede Ethernet com protocolo UDP
- ENA.00043 Interface dos Controladores HI-GII com Dispositivos via Protocolo ASCII
- ENA00044 Protocolos de Comunicação nos Equipamentos G-II
- ENA.00045 Comunicação remota com controladores HI-GII

(\*1) – Estes documentos referem-se a recursos disponíveis nos equipamentos anteriores e não se aplicam integralmente aos equipamentos classificados com G-II.

Todos os documentos referenciados acima estão disponíveis para “download” em nosso site:

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

## 3. Protocolo SCP-HI

---

A linha de controladores industriais da HI Tecnologia permite acesso a sua base de dados através de um protocolo de comunicação denominado SCP-HI. Este protocolo possui as seguintes especificações básicas:

- Protocolo do tipo mestre-escravo com acknowledge (order / replay);
- Permite comunicação ponto a ponto, em rede do tipo multidrop e em rede do tipo anel (denominado ponto a ponto estendido);
- Cada estação pode operar como mestre ou escravo na comunicação (não simultaneamente);
- Permite endereçamento de até 254 estações distintas;
- Utiliza comunicação assíncrona, podendo operar com taxas de 1200 a 38400 bauds (default = 9600 bauds) dependendo do modelo de controlador;



## Driver SCP-HI para Comunicação Serial em Linux

Tipo de Doc.: Notas de Software  
Referência: PNS.00024

Revisão: 1  
Atualizado em: 07/05/2009

### 4. Distribuição da biblioteca de comunicação

---

A biblioteca dinâmica de comunicação serial SCP-HI é distribuída com os seguintes arquivos:

Arquivo	Descrição
libScpHISerialLinux.so	Biblioteca de comunicação serial para Linux
ScpSpec.psf	Exemplo de arquivo de especificação dos parâmetros de configuração utilizados pela biblioteca de comunicação serial

Tabela 1 – Arquivos da Biblioteca Dinâmica

Arquivo	Descrição
PNS.00024	Nota de software documentando a interface com a biblioteca de comunicação (este documento)

Tabela 2 – Arquivos de Documentação da Biblioteca Dinâmica

#### 4.1 Pré-requisitos do sistema operacional Linux

---

A versão mais recente da biblioteca libScpHISerialLinux.só foi compilada com o compilador g++ (GCC) versão 4.2.4. Recomendamos que aplicações clientes da biblioteca libScpHISerialLinux.só utilizem a mesma versão do compilador ou tomem precauções de compatibilidade.



## Driver SCP-HI para Comunicação Serial em Linux

Tipo de Doc.: Notas de Software  
Referência: PNS.00024

Revisão: 1  
Atualizado em: 07/05/2009

### 5. Lista de funções da Biblioteca Dinâmica

---

A tabela a seguir apresenta a lista das funções disponíveis na Biblioteca Dinâmica para utilização pelo usuário.

Nro	Nome	Categoria
1	SCPOpenPort	Controle / Informação do driver
2	SCPClosePort	Controle / Informação do driver
3	SCPVersionRevision	Controle / Informação do driver
4	SCPCheckConnection	Comunicação
5	SCPReadData	Comunicação
6	SCPWriteData	Comunicação

Tabela 3 – Lista de Funções da Biblioteca Dinâmica

#### 5.1 Descrição das funções

---

Este capítulo identifica e define a interface para utilização das funções disponíveis na Biblioteca Dinâmica. Cada função é documentada com os seguintes itens:

- Identificação do Nome da função
- Protótipo da função em C/C++
- Descrição da funcionalidade associada
- Especificação e descrição dos parâmetros de entrada (quando aplicável)
- Especificação e descrição dos parâmetros de saída (quando aplicável)
- Descrição do retorno da função (quando aplicável)
- Notas adicionais quando aplicável



## Driver SCP-HI para Comunicação Serial em Linux

Tipo de Doc.: Notas de Software  
Referência: PNS.00024

Revisão: 1  
Atualizado em: 07/05/2009

### 5.2 SCPOpenPort

Função 01

SCPOpenPort

Protótipo da função em C/C++

```
int SCPOpenPort ( char* CfgFileName )
```

Descrição

Abre o canal de comunicação serial, alocando e configurando a porta serial especificada. Obtém do arquivo "CfgFileName" os parâmetros de configuração necessários para programar a porta serial do computador. Após a execução desta função com sucesso o driver está apto para trocar frames com o controlador conectado.

Terminado o processo de comunicação, utilize a função SCPClosePort para liberar o canal de comunicação do computador.

Parâmetros de Entrada

CfgFileName Nome do arquivo de configuração da comunicação a ser utilizado pelo driver.

Parâmetros de Saída

Não se aplica

Código de Retorno

= 0 (SUCCESS) Indica função executada com sucesso.  
<> 0 Indica condição de falha. Para maiores informações sobre os códigos de retorno consultar ENA.00090.

Notas O arquivo CfgFileName é um arquivo tipo texto permitindo sua abertura com qualquer editor ASCII. Uma vez aberto o arquivo é possível alterar os parâmetros de comunicação de acordo com a aplicação que o driver se destina.





## Driver SCP-HI para Comunicação Serial em Linux

Tipo de Doc.: Notas de Software  
Referência: PNS.00024

Revisão: 1  
Atualizado em: 07/05/2009

### 5.3 SCPClosePort

Função 02

SCPClosePort

Protótipo da função em C/C++

```
int SCPClosePort ( void )
```

Descrição

Fecha a porta de comunicação utilizada pelo driver, disponibilizando este recurso novamente para o sistema operacional.

Parâmetros de Entrada

Não se aplica

Parâmetros de Saída

Não se aplica

Código de Retorno

= 0 (SUCCESS) Indica função executada com sucesso.  
<> 0 Indica condição de falha. Para maiores informações sobre os códigos de retorno consultar ENA.00090.

Notas Quando a biblioteca é liberada, esta fecha automaticamente o driver de comunicação caso este esteja aberto.



## Driver SCP-HI para Comunicação Serial em Linux

Tipo de Doc.: Notas de Software  
Referência: PNS.00024

Revisão: 1  
Atualizado em: 07/05/2009

### 5.4 SCPVersionRevision

Função 03

SCPVersionRevision

Protótipo da função em C/C++

```
int SCPVersionRevision ( int &Versao, int &Revisao )
```

Descrição

Obtêm do driver a versão e revisão corrente do mesmo. O formato geral da versão e revisão utilizados é:

V.R onde:

V = Número da versão

R = Número da revisão

Exemplo: Quando retornados os valores Versão = 2 e Revisão = 0 a versão do driver será: 2.0

Parâmetros de Entrada

Versao	Variável para retornar o número da versão do driver
Revisao	Variável para retornar o número da revisão do driver

Parâmetros de Saída

Versao	Número da versão do driver
Revisao	Número da revisão do driver

Código de Retorno

= 0 (SUCCESS)	Indica função executada com sucesso.
<> 0	Indica condição de falha. Para maiores informações sobre os códigos de retorno consultar ENA.00090.

Notas



## Driver SCP-HI para Comunicação Serial em Linux

Tipo de Doc.: Notas de Software  
Referência: PNS.00024

Revisão: 1  
Atualizado em: 07/05/2009

### 5.5 SCPCheckConnection

Função 04

SCPCheckConnecti on

Protótipo da função em C/C++

```
int SCPCheckConnetion ( int Ctrl Number)
```

Descrição

Envia um frame de conexão para o controlador conectado a porta serial aberta pelo driver. Esta função permite detectar a presença de controlador e avaliar se o link de comunicação esta operacional. Neste comando não existe troca efetiva de dados com o controlador.

Parâmetros de Entrada

Ctrl Number      Número de identificação do controlador a ser conectado. Este número pode estar entre 1 e 255 e deve compatível com o numero previamente programado no controlador. O valor default deste parâmetro para os controladores HI é igual a 1. O valor 255 é definido como endereço global.

Parâmetros de Saída

Não se aplica

Código de Retorno

= 0 (SUCCESS)      Indica função executada com sucesso.  
<> 0                Indica condição de falha. Para maiores informações sobre os códigos de retorno consultar ENA.00090.

Notas      Caso o link de conexão com o controlador não esteja operacional, este processo de conexão aguarda um tempo de timeout específico para o controle desta função.



## 5.6 SCPReadData

### Função 05

### SCPReadData

#### Protótipo da função em C/C++

```
int SCPReadData ( int CtrlNumber, int TypeVar, int VarIni, int VarQuant, void* DataBuf )
```

#### Descrição

Solicita ao controlador os valores das variáveis especificadas na função. O usuário deve especificar o tipo de variável (R, M ou D), identificar o número da variável inicial e a quantidade de variáveis a serem obtidas a partir desta. Os valores das variáveis solicitadas são transferidos para o buffer de dados fornecido.

É de responsabilidade do usuário prover o buffer de recepção com o espaço necessário para receber todos os dados solicitados. Cada tipo de dado requer um espaço distinto no buffer de recepção conforme descrito na tabela a seguir:

<u>Tipo da Variável</u>	<u>Tipo associado na linguagem C</u>	<u>Tamanho</u>
R	unsigned char	1 byte (8 bits)
M	short int	2 bytes (16 bits)
D	float	4 bytes (32 bits)

O tamanho máximo de um frame de dados no protocolo SCP-HI é de 249 bytes. Isto define a quantidade máxima de variáveis de cada tipo que pode ser obtida do controlador em um único comando. Quando a quantidade de variáveis requisitadas ultrapassar este limite, a função divide o buffer de recepção, solicitando tantos frames de comunicação quanto forem necessários para obter todos os dados solicitados.

#### Parâmetros de Entrada

**Ctrl Number** Número de identificação do controlador a ser conectado. Este número pode estar entre 1 e 255 e deve ser compatível com o número previamente programado no controlador. O valor default deste parâmetro para os controladores HI é igual a 1. O valor 255 é definido como endereço global.

**TypeVar** Identifica o tipo de variável a ser solicitada do controlador conforme tabela a seguir:

<u>TypeVar</u>	<u>Tipo de variável</u>	<u>ID</u>	<u>Tamanho</u>	<u>Faixa de valores</u>	
				Min	Max
0	Contato auxiliar	R	1 byte	OFF(0)	ON(255)
1	Memórias inteiras	M	2 bytes	-32768	+32767
2	Memórias reais	D	4 bytes	-10E-38	+10E+38

**VarIni** Identifica o número da variável inicial a ser obtida. Todas as variáveis da base de dados dos controladores HI começam em zero (0) e vão até o número máximo de variáveis do tipo - 1. O número de variáveis de cada tipo depende do tipo de firmware carregado no controlador e, no caso de firmware de PLC este número é alocado dinamicamente em função do programa de aplicação criado. Caso seja especificada uma variável não definida no controlador, será retornado pela função um código de falha indicando esta situação.



## Driver SCP-HI para Comunicação Serial em Linux

Tipo de Doc.: Notas de Software  
Referência: PNS.00024

Revisão: 1  
Atualizado em: 07/05/2009

**VarQuant** Especifica o número de variáveis a serem obtidas a partir de `VarIni`. Sendo assim, se `VarQuant` é igual a 1, apenas `VarIni` é lida do controlador. Se `VarQuant` é igual a 2 serão lidas do controlador o conteúdo de `VarIni` e `VarIni+1`, e assim sucessivamente. Caso a quantidade especificada ultrapasse o número de variáveis definidas no controlador, será retornado pela função um código de falha indicando esta situação.

### Parâmetros de Saída

**DataBuf** Buffer do usuário onde serão salvos os valores obtidos pelo driver de comunicação. É de responsabilidade de o usuário prover o buffer com o espaço necessário para receber todos os dados solicitados. Não é realizado nenhum teste de consistência do tamanho do buffer. O tamanho mínimo do buffer deverá ser:

Tamanho mínimo do buffer de dados = `VarQuant` \* Tamanho de `VarType`

### Código de Retorno

= 0 (SUCCESS) Indica função executada com sucesso.  
<> 0 Indica condição de falha. Para maiores informações sobre os códigos de retorno consultar ENA.00090.

Notas



## 5.7 SCPWriteData

Função 06

SCPWriteData

Protótipo da função em C/C++

```
int SCPWriteData ( int CtrlNumber, int TypeVar, int VarIni, int VarQuant, void* DataBuf )
```

Descrição

Transfere para o controlador o conteúdo do buffer de variáveis especificadas na função. O usuário deve especificar o tipo de variável (R, M ou D), identificar o número da variável inicial e a quantidade de variáveis a serem transferidas a partir desta.

Cada tipo de variável requer um espaço distinto no buffer de transmissão conforme descrito na tabela a seguir:

<u>Tipo da Variável</u>	<u>Tipo associado na linguagem C</u>	<u>Tamanho</u>
R	unsigned char	1 byte (8 bits)
M	short int	2 bytes (16 bits)
D	float	4 bytes (32 bits)

O tamanho máximo de um frame de dados no protocolo SCP-HI é de 249 bytes. Isto define a quantidade máxima de variáveis de cada tipo que pode ser transferida para o controlador em um único comando. Quando a quantidade de variáveis a ser enviada ultrapassar este limite, a função divide o buffer de transmissão, enviando tantos frames de comunicação quanto forem necessários para transferir todos os dados especificados.

Parâmetros de Entrada

**CtrlNumber** Número de identificação do controlador a ser conectado. Este número pode estar entre 1 e 255 e deve ser compatível com o número previamente programado no controlador. O valor default deste parâmetro para os controladores HI é igual a 1. O valor 255 é definido como endereço global.

**TypeVar** Identifica o tipo de variável a ser transferida para o controlador conforme tabela a seguir:

<u>TypeVar</u>	<u>Tipo de variável</u>	<u>ID</u>	<u>Tamanho</u>	<u>Faixa de valores</u>	
				Min	Max
0	Contato auxiliar	R	1 byte	OFF(0)	ON(255)
1	Memórias inteiras	M	2 bytes	-32768	+32767
2	Memórias reais	D	4 bytes	-10E-38	+10E+38



## Driver SCP-HI para Comunicação Serial em Linux

Tipo de Doc.: Notas de Software  
Referência: PNS.00024

Revisão: 1  
Atualizado em: 07/05/2009

VarIni	Identifica o número da variável inicial a ser transferida. Todas as variáveis da base de dados dos controladores HI começam em zero (0) e vão até o número máximo de variáveis do tipo – 1. O número de variáveis de cada tipo depende do tipo de firmware carregado no controlador e, no caso de firmware de PLC este número é alocado dinamicamente ou estaticamente em função do programa de aplicação criado. Caso seja especificada uma variável não definida no controlador, será retornado pela função um código de falha indicando esta situação.
VarQuant	Especifica o número de variáveis a serem transferidas a partir de VarIni. Sendo assim, se VarQuant é igual a 1, apenas VarIni é enviada para o controlador. Se VarQuant é igual a 2 serão transferidas para do controlador o conteúdo de VarIni e VarIni+1, e assim sucessivamente. Caso a quantidade de variáveis especificada ultrapasse o número de variáveis definido no controlador, será retornado pela função um código de falha indicando esta situação.
DataBuf	Buffer do usuário com o vetor de valores do tipo especificado, a serem transferidos para o controlador.

### Parâmetros de Saída

Não se aplica

### Código de Retorno

= 0 (SUCCESS)	Indica função executada com sucesso.
<> 0	Indica condição de falha. Para maiores informações sobre os códigos de retorno consultar ENA.00090.

Notas



## Driver SCP-HI para Comunicação Serial em Linux

Tipo de Doc.: Notas de Software  
Referência: PNS.00024

Revisão: 1  
Atualizado em: 07/05/2009

## 6. Arquivo de configuração dos parâmetros de comunicação

Os principais parâmetros de configuração do driver de comunicação SCP-HI estão especificados em um arquivo de configuração denominado ScpSpec.psf. Este arquivo é do tipo texto podendo ser aberto por qualquer editor ASCII.

Não utilize editores formatados para abrir este arquivo (ex. Word), pois os mesmos podem inserir caracteres de controle no arquivo ao salvá-lo que poderão causar falhas na interpretação dos dados pelo driver. No arquivo existe uma lista de parâmetros com a identificação de cada um.

### 6.1 Parâmetros default de operação do driver

```
#-----#
# Configuracao da Porta Serial de Comunicacao do Computador      #
#-----#

[PC_PORT_SESSION]

# Porta de comunicacao COM -> 1..8
pcpsPort=1

# Baud rate-> 300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200
pcpsBaud=9600

# Paridade-> 0: Nenhuma, 1: Impar, 2: Par, 3:Marca, 4:Espaço
pcpsParity=0

# Stop Bits-> 1: 1 Stop Bit, 2: 2 Stop Bits
pcpsStopBits=1

# Data Bits-> 5: 5 Bits, 6: 6 Bits, 7: 7 Bits, 8: 8 Bits
pcpsDataBits=8

# RS485 Control-> 0: Desativado, 1: Ativado
pcpsRS485Suport=0

# Suporte para comunicacao via Data-rádio-> 0: Desativado, 1: Ativado
pcpsDataRadioSuport=0

# Bytes adicionais no final do frame p/ temporização
pcpsExtraBytes=1

# Tempo adicional de manutenção da portadora na transmissão
pcpsRTSExtTime=0

# Tempo de espera para retirada da portadora do radio remoto
pcpsCarrierOffTime=50

# Nro de frames consecutivos com falha para reset de driver
pcpsFailFramesToReset=50
```





## Driver SCP-HI para Comunicação Serial em Linux

Tipo de Doc.: Notas de Software  
Referência: PNS.00024

Revisão: 1  
Atualizado em: 07/05/2009

### 7. Códigos de retorno

---

---

Os códigos de retorno possíveis de serem obtidos pelo protocolo de comunicação: Códigos de Retorno do Controlador e Códigos de Retorno de Aplicativos estão disponíveis para consulta na ENA.00090

- ENA.00090 Lista de Códigos de Retorno de Funções.

Obs.: Este documento encontra-se disponível para *download* em nosso *site*:

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



## Driver SCP-HI para Comunicação Serial em Linux

Tipo de Doc.: Notas de Software  
Referência: PNS.00024

Revisão: 1  
Atualizado em: 07/05/2009

### Anexo 1 - Exemplo de Utilização

---

No diretório de distribuição da biblioteca existe o subdiretório `example`, contendo 2 exemplos básicos de utilização da biblioteca de ligação dinâmica. Neste diretório há os seguintes arquivos:

<code>compile.sh</code>	Arquivo Shell script com a invocação do compilador GCC para compilar os dois exemplos.
<code>example.c</code>	Código-fonte em C.
<code>example_c</code>	Executável correspondendo ao aplicativo exemplo em C gerado na compilação.
<code>run_c.sh</code>	Arquivo Shell script com a execução do aplicativo de teste <code>example_c</code> gerado pela compilação.
<code>example.cc</code>	Código-fonte em C++.
<code>example_cpp</code>	Executável correspondendo ao aplicativo exemplo em C++ gerado na compilação.
<code>run_cpp.sh</code>	Arquivo Shell script com a execução do aplicativo de teste <code>example_c</code> gerado pela compilação.

Para testar o aplicativo exemplo:

1. Copie a biblioteca `libScpHISerialLinux.so` e o arquivo de configuração `ScpSpec.psf` para o diretório onde estão os exemplos.
2. Faça a compilação dos fontes pelo comando: `./compile.sh`
3. Os arquivos `example_c` e `example_cpp` dever ser gerados como resultado do item 2
4. Para executar o programa em C, use o comando: `./run_c.sh`
5. Para executar o programa em C++, use o comando: `./run_cpp.sh`

Estes exemplos fazem a carga dinâmica da biblioteca, importam duas funções, abrem o arquivo de configuração e imprime o número versão da biblioteca. A seguir incluímos os códigos do fluxo de compilação do arquivo C.



## Driver SCP-HI para Comunicação Serial em Linux

Tipo de Doc.: Notas de Software  
Referência: PNS.00024

Revisão: 1  
Atualizado em: 07/05/2009

### compile.sh

```
g++ -o ./example_cpp example.cc -l/usr/include -L. -lScpHISerialLinux -ldl  
gcc -o ./example_c example.c -l/usr/include -L. -lScpHISerialLinux -ldl
```

### run\_c.sh

```
LD_LIBRARY_PATH=. ./example_c
```

### example.c

```
#include <stdio.h>  
#include <dlfcn.h>  
  
int main(void) {  
    const char *dlsym_error = dlerror();  
    int ret;  
    printf("C dlopen demo\n\n");  
    void* handle = dlopen("./libScpHISerialLinux.so", RTLD_LAZY);  
    if (!handle) {  
        printf("Cannot open library");  
        return 1;  
    }  
  
    dlerror();  
    typedef int (*SCPOpenPort_t)(char*);  
    SCPOpenPort_t ScpOpenPort = (SCPOpenPort_t) dlsym(handle, "SCPOpenPort");  
  
    if (dlsym_error) {  
        printf("Cannot load symbol 'SCPOpenPort': %s", dlsym_error);  
        dlclose(handle);  
        return 1;  
    }  
    ret = SCPOpenPort("ScpSpec.psf");  
    printf("Result %d\n", ret);  
  
    dlerror();  
    typedef int (*SCPVersionRevision_t)(int*, int*);  
    SCPVersionRevision_t SCPVersionRevision = (SCPVersionRevision_t) dlsym(handle, "SCPVersionRevision");  
    if (dlsym_error) {  
        printf("Cannot load symbol 'SCPVersionRevision': %s", dlsym_error);  
        dlclose(handle);  
        return 1;  
    }  
  
    int version, revision;  
    ret = SCPVersionRevision(&version, &revision);  
    printf("RetCode %d. Version %d. Revision %d.\n", ret, version, revision);  
  
    dlclose(handle);  
}
```



## Driver SCP-HI para Comunicação Serial em Linux

Tipo de Doc.: Notas de Software  
Referência: PNS.00024

Revisão: 1  
Atualizado em: 07/05/2009

## 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	29/04/2009	Paulo C. M. Inazumi	
Revisão	07/05/2009	Rodrigo D. A. Serra	<i>Revisado em mídia</i>
Aprovação	07/05/2009	Helio J. Almeida Jr.	<i>Aprovado em mídia</i>

#### Histórico de Revisões

Data	Rev	Descrição
07/05/2009	1	Apêndice com exemplo de utilização da biblioteca
30/04/2009	0	Documento original