

LIBSWEDA

BIBLIOTECA LINUX PARA IMPRESSORA FISCAL

**ST100 E ST1000
(PROTOCOLO STX)**

VERSÃO 1.0

Abril 2006

ESTE MANUAL TEM CARÁTER TÉCNICO- INFORMATIVO, SENDO PROPRIEDADE DA
SWEDA INFORMÁTICA LTDA (SWEDA).

TODOS OS DIREITOS RESERVADOS.

É PERMITIDA A REPRODUÇÃO DESTE DOCUMENTO, DESDE QUE PRESERVADA A
INTEGRIDADE DO SEU CONTEÚDO, COM AUTORIZAÇÃO DA SWEDA.

TODOS OS CUIDADOS FORAM TOMADOS NO PREPARO DESTE MANUAL.

ENTRETANTO A SWEDA NÃO ASSUME NENHUMA RESPONSABILIDADE POR ERROS,
IMPRECISÕES OU OMISSÕES EM SEU CONTEÚDO, ASSIM COMO POR QUAISQUER TIPOS DE
DANOS, PERDAS OU LUCROS CESSANTES DECORRENTES DO USO DA INFORMAÇÃO AQUI
CONTIDA.

AS INFORMAÇÕES CONTIDAS NESTE MANUAL ESTÃO SUJEITAS A ALTERAÇÃO SEM AVISO
PRÉVIO.

ÍNDICE

1	OBJETIVOS	4
2	DESCRIÇÃO DO CONTEÚDO.....	4
3	COMPILANDO APLICATIVOS PARA UTILIZAR A BIBLIOTECA	5
4	FUNÇÕES	5
4.1.1	<i>int SWCOM_ConfigurarInterface(void (*func)(int tamanho)).....</i>	5
4.1.2	<i>int SWCOM_AbrirUART(int id, int nbps)</i>	5
4.1.3	<i>int SWCOM_DetectarECF(void).....</i>	5
4.1.4	<i>int SWCOM_EnviarComando(char* comando).....</i>	6
4.1.5	<i>int SWCOM_EnviarDados(char* comando, int tamanho).....</i>	6
4.1.6	<i>int SWCOM_EnviarByte(char dado, int fd).....</i>	6
4.1.7	<i>int SWCOM_ObterRetorno(char* retorno, int extensao, int tempoEspera).....</i>	6
4.1.8	<i>void SWCOM_DestruirInterface(void).....</i>	6

1 OBJETIVOS

Este documento descreve os passos necessários para utilização da Biblioteca Linux no desenvolvimento de aplicações para a família de impressoras fiscais SWEDA IF ST-100 e ST-1000.

2 DESCRIÇÃO DO CONTEÚDO

A interface da biblioteca permite comunicação com a impressora fiscal utilizando as portas padrão (COM1 a COM4, mapeadas pelo sistema operacional nos dispositivos /dev/ttyS0 a /dev/ttyS3) e eventuais portas adicionais (no máximo 2) cujos dispositivos deverão estar listados no arquivo texto ports.ini (conforme exemplo abaixo):

```
/dev/ttyS10  
/dev/ttyS11
```

A primeira linha será mapeada na COMX e a segunda na COMY.

O pacote é composto dos seguintes binários:

- **libsweda.a** - Biblioteca de ligação estática com funções para acesso à impressora fiscal;
- **teste-abrir** - Binário que exercita as funções de abertura de porta e envio de bytes (em modo raw) lidos a partir de um arquivo texto;
Executado da seguinte forma:

```
./teste-abrir Numero_COM Velocidade Arquivo  
Numero_COM = 0->COM1 1->COM2 2->COM3 3->COM4 4->ComX 5->ComY  
Velocidade = 0->9600 1->19200 2->38400 3->57600 4->115200  
Arquivo = arquivo no diretório local  
O arquivo será lido (um byte por linha)  
ComX e ComY são lidos a partir do arquivo ports.ini - primeira e segunda linhas
```

- **teste-enviar** – Binário para exercitar as funções de criação de instância de interface (a partir de configuração existente) e envio de dados/comandos utilizando o protocolo de comunicação da sweda;
Executado da seguinte forma:

```
./teste-enviar "String a ser transmitida"
```

- **teste-receber** – Binário para exercitar as funções de criação de instância de interface (a partir de configuração existente) e recepção de dados;
Executado da seguinte forma:

```
./teste-receber
```

- **teste-detectar** – Binário para realizar a detecção da impressora fiscal nas portas COM1 a COM4 (/dev/ttyS0 a /dev/ttyS3) e nas portas listadas no arquivo ports.ini

Executado da seguinte forma:

```
./teste-detectar
```

- **teste-criar-destruir** – Binário para exercitar criação e encerramento de instância da interface (a partir de configuração existente).

Executado da seguinte forma:

```
./teste-criar-destruir
```

Os aplicativos teste-enviar, teste-receber, teste-criar-destruir operam com informações coletadas pelo aplicativo teste-detectar (que armazena as configurações de detecção de impressora no arquivo SWCOM.ini e lê a configuração de portas adicionais do arquivo ports.ini).

3 COMPILANDO APLICATIVOS PARA UTILIZAR A BIBLIOTECA

Com base no projeto exemplo, o usuário poderá desenvolver seus aplicativos de forma a utilizar da melhor forma os recursos da biblioteca.

- **libsweda.a** – Biblioteca de ligação estática
- **biblinux.h** – Arquivo de cabeçalho com assinatura das funções públicas da biblioteca
- **Makefile** – Arquivo de projeto exemplo
- **.H e .C** – Arquivos C e H do projeto exemplo

4 FUNÇÕES

4.1.1 **int SWCOM_ConfigurarInterface(void (*func)(int tamanho))**

Realiza a configuração da interface com a biblioteca e inicia os objetos de envio e recepção de dados pela porta serial. Esta função executa leitura no arquivo SWCOM.ini para obter informações de porta onde está conectada a impressora e velocidade de comunicação.

É esperado um ponteiro para função do tipo *void callback_func(int tamanho)* se for desejada a utilização de callback de recepção ou valor nulo (0) caso não se deseje esta forma de interação.

A função retorna valor 1 em caso de sucesso ou 0 em caso de falha

4.1.2 **int SWCOM_AbrirUART(int id, int nbps)**

Realiza abertura de porta de comunicação em uma determinada velocidade.

É esperado um inteiro com identificação da porta com um dos seguintes valores:

- 0 -> COM1 (/dev/ttyS0)
- 1 -> COM2 (/dev/ttyS1)
- 2 -> COM3 (/dev/ttyS2)
- 3 -> COM4 (/dev/ttyS3)
- 4 -> COMX (primeira porta serial adicional – opcional – lida a partir da primeira linha do arquivo ports.ini)
- 5 -> COMY (segunda porta serial adicional – opcional – lida a partir da segunda linha do arquivo ports.ini)

E um inteiro com velocidade de comunicação desejada:

- 0 -> 9600bps
- 1 -> 19200bps
- 2 -> 38400bps
- 3 -> 57600bps
- 4 -> 115200bps

A função retorna valor 1 em caso de sucesso ou 0 em caso de falha

4.1.3 **int SWCOM_DetectarECF(void)**

Realiza a detecção do ECF, efetuando busca nas seguintes portas:

- COM1 (/dev/ttyS0)
- COM2 (/dev/ttyS1)
- COM3 (/dev/ttyS2)
- COM4 (/dev/ttyS3)
- COMX (primeira porta serial adicional – opcional – lida a partir da primeira linha do arquivo ports.ini)
- COMY (segunda porta serial adicional – opcional – lida a partir da segunda linha do arquivo ports.ini)

A busca em cada porta será feita utilizando as seguintes velocidades:

- 9600bps
- 19200bps
- 38400bps
- 57600bps
- 115200bps

A procura é interrompida (com sucesso) assim que for obtida uma resposta de sucesso (ACK) ao comando de avanço de papel.

Caso a detecção seja concluída com sucesso, os parâmetros são registrados no arquivo SWCOM.ini e retornado valor 1. Em caso contrário, será retornado valor 0.

4.1.4 int SWCOM_EnviarComando(char* comando)

Envia um comando fiscal para a impressora por meio de interface (configurada anteriormente ao envio).

Recebe um comando em string terminada por zero, realiza encapsulamento por meio do protocolo STX/ETX . A função realiza controle de confirmação do comando enviado (ACK) ou falha de transmissão (NAK ou timeout) e eventualmente poderá efetuar retransmissão do comando.

A função é bloqueante e retorna 1 em caso de sucesso ou 0 em caso de falha.

4.1.5 int SWCOM_EnviarDados(char* comando, int tamanho)

Envia dados para a impressora por meio de interface (configurada anteriormente ao envio).

Recebe dado a ser enviado e tamanho, realiza encapsulamento por meio do protocolo STX/ETX . A função realiza controle de confirmação do dado enviado (ACK) ou falha de transmissão (NAK ou timeout) e eventualmente poderá efetuar retransmissão do dado.

A função é bloqueante e retorna 1 em caso de sucesso ou 0 em caso de falha.

4.1.6 int SWCOM_EnviarByte(char dado, int fd)

Envia um byte por meio de uma porta de comunicação aberta anteriormente.

Recebe byte a ser enviado e descritor de arquivo para o dispositivo da porta serial.

Retorna 1 em caso de sucesso e 0 em caso de falha.

4.1.7 int SWCOM_ObterRetorno(char* retorno, int extensao, int tempoEspera)

Obtem retorno (com pacote enviado pela impressora).

Recebe ponteiro para buffer de recepção, extensão do buffer e tempo máximo de espera pela disponibilidade dos dados para leitura. O buffer deverá ter tamanho suficiente para hospedar o maior pacote que possa ser enviado pela impressora.

Em caso de sucesso, o buffer conterá os dados e será retornado valor 1. Em caso de falha a função retornará valor 0. A função é bloqueante.

4.1.8 void SWCOM_DestruirInterface(void)

Desativa a instância da interface e elimina os objetos de comunicação com a porta serial