

ELGIN

Driver Linker SAT II

Manual do Usuário



Versão 1.0.3 Setembro/2016

CONTROLE DE VERSÕES

Versão	Data	Observações
1.0.0	20/04/2016	Versão inicial
1.0.1	05/08/2016	Inclusão do procedimento para instalação em Linux
1.0.2	23/08/2016	Atualizações no monitor, ilustração e diretório de instalação
1.0.3	09/09/2016	Revisão Geral

DEFINIÇÕES E SIGLAS

Termo	Notas Técnicas incorporadas
AC	Aplicativo Comercial – aplicativo para emissão de Cupons Fiscais de Venda e de Cancelamento
CFe	Cupom Fiscal Eletrônico de Venda ou de Cancelamento
PDV	Equipamento do Contribuinte onde está conectado o Linker SAT II
Projeto SAT-CFe	Conjunto de especificações técnicas definidas pelos documentos de Especificação de Requisitos (ER) e Manual de Orientação (MO) do Projeto SAT CF-e definidas a partir das disposições do Ajuste Sinief nº 11 de 24 de setembro de 2010.
SAT	Sistema de Transmissão e Autenticação de documentos fiscais eletrônicos
SEFAZ-SP	Secretaria da Fazenda do Estado de São Paulo

ÍNDICE

1	INTRODUÇÃO	5
1.1	Componentes do Driver Linker SAT II	5
2	PONTO DE VENDA	6
2.1	Aplicativo Comercial	7
2.2	Driver Linker SAT II	7
2.3	Driver de comunicação Linker SAT II	8
3	INSTALAÇÃO	8
3.1	Sistema Operacional Windows	8
3.2	Sistema Operacional Linux	10
4	Comunicador	12
5	Biblioteca de Funções	12
5.1	Função AtivarSAT	12
5.2	Função EnviarDadosVenda	14
5.3	Função CancelarUltimaVenda	14
5.4	Função ConsultarSAT	15
5.5	Função TesteFimAFim	15
5.6	Função ConsultarStatusOperacional	16
5.7	Função ConsultarNumeroSessao	16
5.8	Função ConfigurarInterfaceDeRede	17
5.9	Função AssociarAssinatura	17
5.10	Função AtualizarSoftwareSAT	18
5.11	Função ExtrairLogs	18
5.12	Função BloquearSAT	19
5.13	Função DesbloquearSAT	19
5.14	Função TrocarCodigoDeAtivacao	20
6	PROTÓTIPOS DAS FUNÇÕES	21

1 INTRODUÇÃO

Para que o Aplicativo Comercial possa se conectar ao Linker SAT II e acessar suas funções, um conjunto de componentes de software é fornecido para prover a interface de comunicação.

Este conjunto de componentes de software é chamado de Driver Linker SAT II. O Driver Linker SAT II é fornecido juntamente com o SAT e está disponível no sítio da ELGIN na forma de um programa instalador.

Este documento destina-se ao pessoal técnico responsável pela instalação do PDV e/ou desenvolvimento de Aplicativos Comerciais e descreve o Driver Linker SAT II, sua instalação, seus componentes e as funções disponíveis.

1.1 Componentes do Driver Linker SAT II

O Driver Linker SAT II é composto pelos seguintes componentes:

- Biblioteca de Funções
- Comunicador

2 PONTO DE VENDA

O Ponto de Venda (PDV) é o equipamento do contribuinte responsável pela execução do Aplicativo Comercial para a realização das operações fiscais. É nesse equipamento que o Linker SAT II deverá estar conectado via serial sobre USB. Para que seja possível a comunicação entre o Aplicativo Comercial e o Linker SAT II é necessário que o Driver de Comunicação USB e o Driver Linker SAT II estejam instalados corretamente.

Para o funcionamento do PDV são necessários os seguintes componentes:

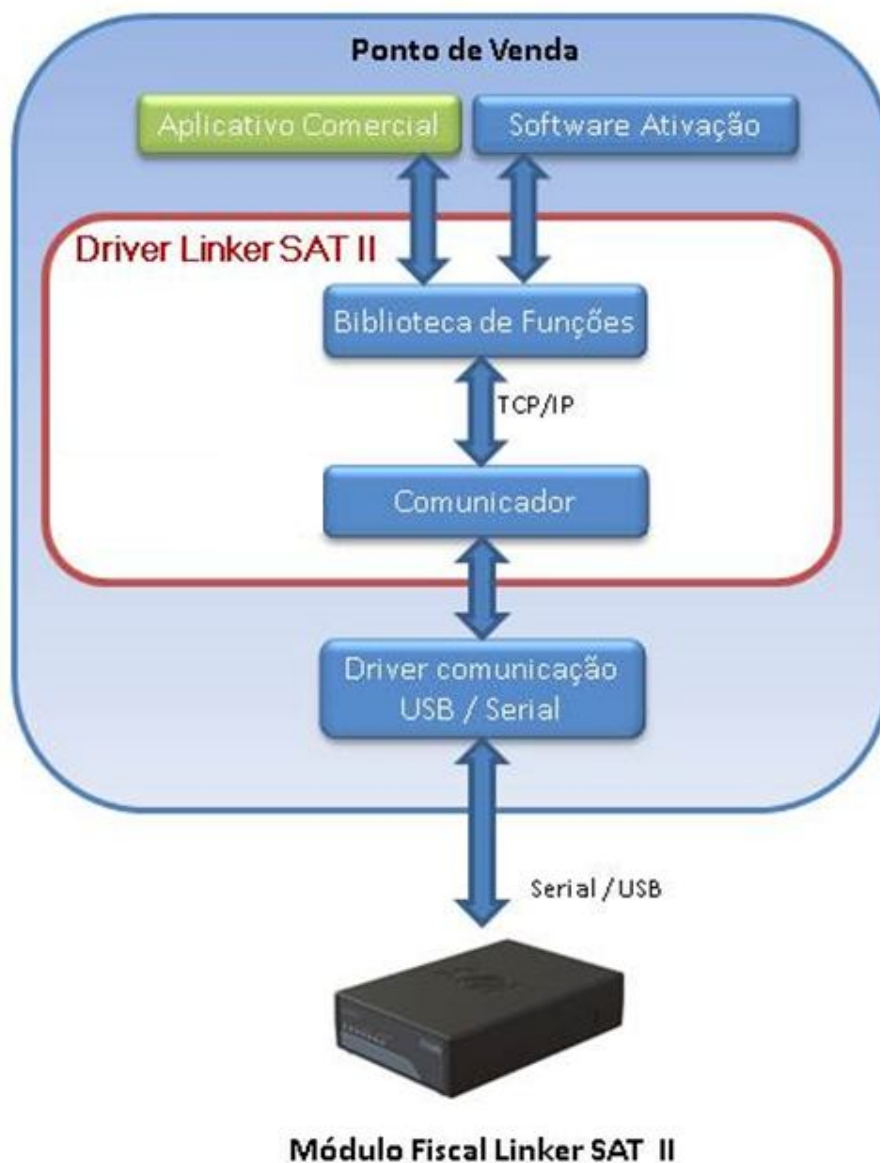


Figura 1 - Componentes de Software do Ponto de Venda

2.1 Aplicativo Comercial

Aplicação responsável por enviar os dados de venda ou de cancelamento para que o Linker SAT II possa gerar o respectivo Cupom Fiscal Eletrônico, além de permitir a execução de outras funcionalidades.

Poderá ser utilizado qualquer Aplicativo Comercial compatível com os padrões do SAT-CFe.

2.2 Driver Linker SAT II

Para facilitar a instalação e atualização dos componentes de software disponibilizados pela ELGIN, eles foram empacotados em uma única instalação chamada Driver Linker SAT II.

Os componentes do Driver Linker SAT II são:

Componente	Descrição
Biblioteca de Funções	<p>Biblioteca de funções (arquivo com extensão “.DLL” no Windows ou “.SO” no Linux) que disponibiliza para as aplicações (AC/Software de Ativação) todas as funcionalidades necessárias para a operação do Linker SAT II padronizando assim toda a comunicação.</p> <p>Em ambiente Windows 32 e 64 bits a biblioteca é chamada de sat.dll e se encontra na pasta: C:\Program Files (x86)\ELGIN\Driver Linker SAT II\Biblioteca de funções</p> <p>Esse caminho é definido nas variáveis de ambiente do Windows durante a instalação do Driver Linker SAT II.</p>
Comunicador	É uma aplicação executada como um serviço responsável por realizar a comunicação com o SAT.

O Driver Linker SAT II é compatível com os seguintes sistemas operacionais:

- Windows XP, Vista, 7, 8, 8.1 e 10 (32 e 64 bits)
- Linux OpenSUSE 12. 2e 12.3, Debian 7.3 e 8.1, e Fedora 21 e 22 (versões 32 e 64 bits)

2.3 Driver de comunicação Linker SAT II

Driver responsável por permitir a conversão do dispositivo USB, ao qual o Linker SAT II está conectado, em interface serial conforme o sistema operacional do PDV e disponibilizá-la para uso pelo serviço Comunicador.

Exemplos de portas seriais: COM11 (Windows) ou ttyS1 (Linux).

3 INSTALAÇÃO

O programa instalador do Driver Linker SAT II instalará automaticamente todos os seus componentes e configurará o sistema operacional do PDV para executar os serviços e aplicações necessárias.

Antes da sua instalação deve-se:

1. Instalar o Driver de Comunicação Linker SAT II.
2. Confirmar a existência de uma nova porta serial disponível quando o Linker SAT II está conectado.

3.1 Sistema Operacional Windows

A seguir são descritos os passos para a instalação do Driver Linker SAT II para o Sistema Operacional Windows:

1. Execute o arquivo de instalação;
2. Caso o Windows exiba uma janela perguntando “Deseja permitir que o programa de um fornecedor desconhecido faça alterações nesse computador”, confirme a execução.
3. Será exibida a tela de boas vindas, clique no botão “**Avançar**”:



Figura 2 - Tela Inicial do Instalador

4. Clique em instalar para iniciar a instalação:

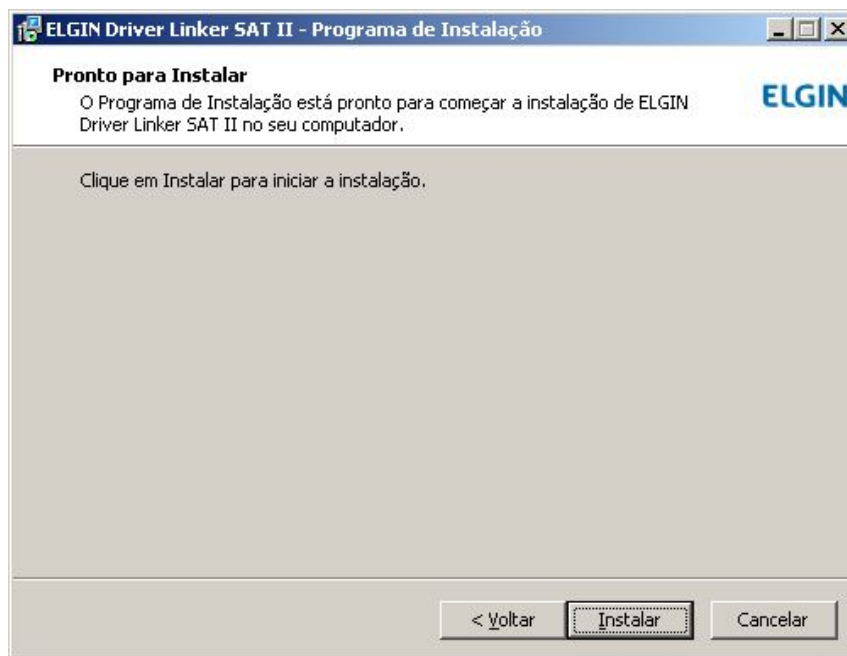


Figura 3 - Pronto para iniciar a instalação

Neste momento todos os softwares para uso do SAT foram instalados:

- A Biblioteca de Funções foi instalada
- O Comunicador já está em execução.

Observações:

1. De acordo com as configurações de segurança e atualizações do Windows em uso, algumas janelas podem não ser exibidas por serem automaticamente aceitas pelo sistema, neste caso ignore o passo em questão.

3.2 Sistema Operacional Linux

A seguir são descritos os passos para a instalação do Driver Linker SAT II para o Sistema Operacional Linux:

1. Acessar o terminal do Linux como root;
2. Descompactar o arquivo instalador-sp-elgin-driver-linux-32.tar.gz (ou instalador-sp-elgin-driver-linux-64.tar.gz) executando o comando `tar -vzxf nome do arquivo`;
3. Dentro da pasta, execute o script `instala_driver.sh` como mostra a Figura 4.

```
driver-linux-64# ./instala_driver.sh
Instalacao Driver Linker SAT II 01.03.45

Instalacao Comunicador Linker SAT II 01.03.01.037
Instalacao Biblioteca Linker SAT II 01.03.01.023
Criando diretorios
Copiando arquivos
Configurando porta serial
Configurando Servico de Comunicacao
Habilitando o Comunicador
ln -s '/lib/systemd/system/ccco-ser.service' '/etc/systemd/system/multi-user.target.wants/ccco-ser.service'
Iniciando o Comunicador
Status do Comunicador
ccco-ser.service - Comunicador - Componente de Comunicacao
Loaded: loaded (/lib/systemd/system/ccco-ser.service; enabled)
Active: active (running) since Tue, 24 May 2016 18:24:47 -0300; 26ms a
go
Main PID: 5811 (runccco-ser.sh)
CGroup: name=systemd:/system/ccco-ser.service
        5811 /bin/sh - /opt/elgin/ccco/bin/runccco-ser.sh
        5812 ./Comunicador

May 24 18:24:47 debian7964bit runccco-ser.sh[5811]: INICIANDO Comunicador
May 24 18:24:47 debian7964bit runccco-ser.sh[5811]: LD_LIBRARY_PATH=
May 24 18:24:47 debian7964bit runccco-ser.sh[5811]: PATH=/usr/local/sbin:/usr/...
May 24 18:24:47 debian7964bit runccco-ser.sh[5811]: [LogDebug] Instalando capt...
```

Figura 4 - Resultado da instalação com sucesso

Os arquivos foram instalados na pasta `/opt/elgin`.

Os componentes da instalação são:

Arquivo	Função
Comunicador	Aplicativo do Comunicador
libs*.so.*	Bibliotecas auxiliares do Driver Linker SAT
libdllsat.so.*	Biblioteca do SAT
instala_driver.sh	Script de instalação
remove_driver.sh	Script de remoção
cco-ser.ini	Configuração do cco-ser
cco-ser.service	Descritor do serviço
80-tty-elgin.rules	Configuração das seriais
cco-mon.desktop	Arquivo para startup

4 Comunicador

O Comunicador do Driver Linker SAT II é uma aplicação executada como um serviço é responsável por realizar a comunicação com o Linker SAT II através da porta serial emulada sobre uma porta USB.

Sem este serviço em execução a comunicação com o SAT não será possível.

Ele desempenha as seguintes funções:

- Identifica automaticamente quando um SAT é conectado ou desconectado do PDV;
- Recebe as funções chamadas via Biblioteca de Funções do Driver Linker SAT II e realiza a comunicação com SAT;

5 Biblioteca de Funções

A Biblioteca de Funções do Driver Linker SAT II disponibiliza todas as funções estabelecidas no **Projeto SAT-CFe**.

As descrições das funções, seus parâmetros e retornos estão nos documentos de Especificação de Requisitos (ER) em: <http://www.fazenda.sp.gov.br/sat>.

Segue uma descrição das funções da Biblioteca de Funções e respectivos parâmetros de entrada:

5.1 Função AtivarSAT

Função utilizada no processo de ativação do SAT e serve para informar:

- Qual será tipo de certificado usado na ativação: AC-SAT ou ICP-Brasil
- Código de Ativação
- CNPJ do Contribuinte
- Unidade Federativa

Protótipo da função:

```
char* AtivarSAT(  
    int iNumSessao,  
    int iSubCmd,  
    char* pcCodAtivacao,  
    char* pcCnpj,  
    int icF  
)
```

Parâmetros:

iNumSessao: Número da sessão.

iSubCmd:	Tipo de Certificado: AC-SAT ou ICP-BRASIL
pcCodAtivacao:	Senha definida pelo contribuinte.
pcCnpj:	CNPJ do contribuinte, somente números.
icUF:	Código do Estado. São Paulo: 35

5.2 Função EnviarDadosVenda

Envia os dados de uma venda para que o SAT possa gerar um CFe de venda.

Protótipo da função:

```
char* EnviarDadosVenda(  
    int iNumSessao,  
    char* pcCodAtivacao,  
    char* pcDadosVenda  
)
```

Parâmetros:

iNumSessao:	Número da sessão.
pcCodAtivacao:	Senha definida pelo contribuinte.
pcDadosVenda:	XML com os dados de venda.

5.3 Função CancelarUltimaVenda

Envia os dados de um cancelamento de venda para que o SAT gere um CFe de cancelamento de venda.

Protótipo da função:

```
char* CancelarUltimaVenda (  
    int iNumSessao,  
    char* pcCodAtivacao,  
    char* pcChaveCfe,  
    char* cDadosCanc  
)
```

Parâmetros:

iNumSessao:	Número da sessão.
pcCodAtivacao:	Senha definida pelo contribuinte.
pcChaveCfe:	Chave de acesso do CFe a ser cancelado.
pcDadosCanc:	XML com os dados de cancelamento.

5.4 Função ConsultarSAT

Função utilizada para verificar se há comunicação com o SAT.

Protótipo da função:

```
char* ConsultarSAT (  
    int iNumSessao  
)
```

Parâmetros:

iNumSessao: Número da sessão.

5.5 Função TesteFimAFim

Função utilizada para realizar a emissão de um CFe de teste e verificar se os processos de emissão de CFes e de comunicação com os servidores da SEFAZ-SP estão funcionando.

Protótipo da função:

```
char* TesteFimAFim (  
    int iNumSessao,  
    char* pcCodAtivacao,  
    char* pcDadosVenda  
)
```

Parâmetros:

iNumSessao: Número da sessão.

pcCodAtivacao: Senha definida pelo contribuinte.

pcDadosVenda: XML com os dados da venda de teste.

5.6 Função ConsultarStatusOperacional

Função utilizada para obter informações sobre o estado de operação do SAT.

Protótipo da função:

```
char* ConsultarStatusOperacional (  
    int iNumSessao,  
    char* pcCodAtivacao  
)
```

Parâmetros:

iNumSessao: Número da sessão.
pcCodAtivacao: Senha definida pelo contribuinte.

5.7 Função ConsultarNumeroSessao

Função utilizada para obter novamente os dados retornados na última sessão.

Protótipo da função:

```
char* ConsultarNumeroSessao (  
    int iNumSessao,  
    char* pcCodAtivacao,  
    int iNumSessaoConsultada  
)
```

Parâmetros:

iNumSessao: Número da sessão.
pcCodAtivacao: Senha definida pelo contribuinte.
iNumSessaoConsultada: Número da sessão a ser consultada.

5.8 Função ConfigurarInterfaceDeRede

Função utilizada para configurar a comunicação do SAT com a Internet.

Protótipo da função:

```
char* ConfigurarInterfaceDeRede (  
    int iNumSessao,  
    char* pcCodAtivacao,  
    char* pcDadosConfiguracao  
)
```

Parâmetros:

iNumSessao:	Número da sessão.
pcCodAtivacao:	Senha definida pelo contribuinte.
pcDadosConfiguracao:	XML com os dados de configuração das interfaces do SAT.

5.9 Função AssociarAssinatura

Função utilizada para vincular a assinatura do Aplicativo Comercial com o Contribuinte. Utilizada na ativação do SAT ou quando for trocado o fornecedor do Aplicativo Comercial.

Protótipo da função:

```
char* AssociarAssinatura (  
    int iNumSessao,  
    char* pcCodAtivacao,  
    char* pcCnpj,  
    char* pcAssinaturaCnpps  
)
```

Parâmetros:

iNumSessao:	Número da sessão.
pcCodAtivacao:	Senha definida pelo contribuinte.
pcCnpj:	CNPJ da empresa desenvolvedora do Aplicativo Comercial + CNPJ do

Contribuinte.

pcAssinaturaCnpj: Assinatura digital conjunto “CNPJ Software House” + “CNPJ do Contribuinte”.

5.10 Função AtualizarSoftwareSAT

Função utilizada para atualizar o Software Básico do SAT.

Protótipo da função:

```
char* AtualizarSoftwareSAT (  
    int iNumSessao,  
    char* pcCodAtivacao  
)
```

Parâmetros:

iNumSessao: Número da sessão.
pcCodAtivacao: Senha definida pelo contribuinte.

5.11 Função ExtrairLogs

Função utilizada para obter o log de operação do SAT.

Protótipo da função:

```
char* ExtrairLogs (  
    int iNumSessao,  
    char* pcCodAtivacao  
)
```

Parâmetros:

iNumSessao: Número da sessão.
pcCodAtivacao: Senha definida pelo contribuinte.

5.12 Função BloquearSAT

Função utilizada para bloquear as funções fiscais do SAT.

Protótipo da função:

```
char* BloquearSAT (  
    int iNumSessao,  
    char* pcCodAtivacao  
)
```

Parâmetros:

iNumSessao: Número da sessão.
pcCodAtivacao: Senha definida pelo contribuinte.

5.13 Função DesbloquearSAT

Função utilizada para desbloquear as funções fiscais do SAT.

Protótipo da função:

```
char* DesbloquearSAT (  
    int iNumSessao,  
    char* pcCodAtivacao  
)
```

Parâmetros:

iNumSessao: Número da sessão.
pcCodAtivacao: Senha definida pelo contribuinte.

5.14 Função TrocarCodigoDeAtivacao

Função utilizada para trocar o código de ativação escolhido pelo Contribuinte na ativação do SAT.

Protótipo da função:

```
char* TrocarCodigoDeAtivacao (  
    int iNumSessao,  
    char* pcCodAtivacao,  
    int iOpcao,  
    char* pcNovoCod,  
    char* pcConfirmaNovoCod  
)
```

Parâmetros:

iNumSessao:	Número da sessão.
pcCodAtivacao:	Senha definida pelo contribuinte.
iOpcao:	Informa se está sendo usado o código de emergência ou não.
pcNovoCod:	Novo código de ativação.
pcConfirmaNovoCod:	Confirmação do novo código de ativação.

6 PROTÓTIPOS DAS FUNÇÕES

Abaixo segue a lista dos protótipos da Biblioteca de Funções:

```
char* __STDCALL AtivarSAT(int iNumSessao, int iSubCmd, const char*  
pcCodAtivacao, const char* pcCnpj, int iUF );
```

```
char* __STDCALL ComunicarCertificadoICPBRASIL(int iNumSessao, const  
char* pcCodAtivacao, const char* pcCertificado );
```

```
char* __STDCALL EnviarDadosVenda(int iNumSessao, const char*  
pcCodAtivacao, const char* pcDadosVenda );
```

```
char* __STDCALL CancelarUltimaVenda(int iNumSessao, const char*  
pcCodAtivacao, const char* pcChaveCfe, const char* pcDadosCanc );
```

```
char* __STDCALL ConsultarSAT(int iNumSessao );
```

```
char* __STDCALL TesteFimAFim(int iNumSessao, const char*  
pcCodAtivacao, const char* pcDadosVenda );
```

```
char* __STDCALL ConsultarStatusOperacional(int iNumSessao, const  
char* pcCodAtivacao );
```

```
char* __STDCALL ConsultarNumeroSessao (int iNumSessao, const char*  
pcCodAtivacao, int iNumSessaoConsultada );
```

```
char* __STDCALL ConfigurarInterfaceDeRede (int iNumSessao, const  
char* pcCodAtivacao, const char* pcDadosConfiguracao );
```

```
char* __STDCALL AssociarAssinatura (int iNumSessao, const char*  
pcCodAtivacao, const char* pcCnpj, const char* pcAssinaturaCnpj );
```

```
char* __STDCALL AtualizarSoftwareSAT(int iNumSessao, const char*  
pcCodAtivacao );
```

```
char* __STDCALL ExtrairLogs(int iNumSessao, const char* pcCodAtivacao  
);
```

```
char* _STD_CALL BloquearSAT (int iNumSessao, const char*  
pcCodAtivacao );
```

```
char* _STD_CALL DesbloquearSAT (int iNumSessao, const char*  
pcCodAtivacao );
```

```
char* _STD_CALL TrocarCodigoDeAtivacao(int iNumSessao, const char*  
pcCodAtivacao, int iOpcao, const char* pcNovoCod, const char*  
pcConfirmaNovoCod );
```

As seguintes funções “dummy” foram acrescentadas à Biblioteca de Funções para efeito de compatibilidade com os Aplicativos Comerciais de mercado, porém não têm efeito prático. Ou seja, permite que o AC encontre apenas função na Biblioteca de Funções sem que erros ocorram durante a sua execução. De qualquer maneira, não afetam, por exemplo, a configuração da serial de comunicação com o SAT.

```
char* _STD_CALL Base64ToAscii(void);
```

```
int _STD_CALL AbreSerialSAT(int commPort, int baud, int nBits, int  
paridade, int nStops);
```

```
void _STD_CALL CloseSerial(void);
```

```
int _STD_CALL GeraNumeroSessao(void);
```