

Especificação sk_access V1.99.97
libsk_access.so V1.99.9b



Descrição:

sk_access.dll(Windows) / libsk_access.so(Linux) são bibliotecas de funções desenvolvidas pela Smak Teclados com o objetivo de facilitar o trabalho daqueles que desenvolvem softwares para os produtos SMAK. Este documento destina-se a desenvolvedores e contém as informações técnicas necessárias ao uso das bibliotecas.

As bibliotecas Windows e Linux disponibilizam as mesmas funções, facilitando o desenvolvimento de aplicações multiplataforma.

Rev. 1.99a

Índice

Histórico de alterações.....	3
Alterações da sk_access.dll V1.99.97 em relação à V1.99.94.....	3
Alterações da libsk_access.so V1.99.9b em relação à V1.99.7a.....	3
Quem deve ler este manual.....	4
Descrição da sk_access.....	4
Princípio de funcionamento.....	4
Importação da lib.....	5
Plug and Play.....	6
Variáveis de ambiente.....	7
Demonstração e testes das funções da biblioteca Smak.....	10
Instalando a lib.....	10
Windows.....	10
Linux.....	10
Testes.....	11
Utilizando a biblioteca da SMAK em seu ambiente de desenvolvimento.....	11
Convenção de chamada.....	11
Funções exportadas pela lib.....	12
Nota sobre a função GetOperatingSystem:.....	15
Nota sobre a função Redirect:.....	15
Nota sobre a função Send_Disp_Ctrl:.....	15
Notas sobre versões de Firmware do SKO-44.....	16

Histórico de alterações:

Revisão 1.99a (07-07-2020):

- Acrescentadas dados sobre lib Llinux

Revisão 1.99 (23-09-2019):

- Acrescido nome e local de criação do arquivo de log.
- Variáveis de ambiente grafadas em Maiúsculo.
- Localização das variáveis de ambiente no SO Windows.

Revisão 1.98 (24-02-2018) :

- Atualização info sobre Select_Interface, Set_Interface, Get_Current_interface.
- Tabela de caracteres do display LCD
- Acrescentado: Quem deve ler este manual.

Alterações da sk_access.dll V1.99.97 em relação à V1.99.94:

- Redirect sem teclado conectado fixa interface VCP.
- Retirado teste do timer de 15useg.
- Refeito teste da controladora 8042.
- Reconhecimento dos teclados por Vid e Pid.
- Ignoradas as variáveis de ambiente SK_MAXCOM e SK_USETSC
- Criadas variáveis de ambiente SK_NOP52, SK_NOHID, SK_NOCP, SK_2303, SK_FIRST_USB
- Conversor serial PL2303 não é mais reconhecido por padrão(Default).
- Alterado de 5 para 3 o valor gerado para a variável de ambiente SK_8042 no caso de controladora defeituosa.
- Melhoradas informações de log.
- Solução de Bugs

Alterações da libsk_access.so V1.99.9b em relação à V1.99.7a

- Redirect sem teclado conectado fixa interface VCP.
- Criadas funções Set_Beep_On e Set_Beep_Off para compatibilizar com dll do Windows.
- Reconhecimento dos teclados por Vid e Pid.
- Ignoradas as variáveis de ambiente SK_MAXCOM e SK_USETSC.
- Criadas variáveis de ambiente SK_NOP52, SK_NOHID, SK_NOCP, SK_2303.
- Conversor serial PL2303 não é mais reconhecido por padrão(Default).
- SK_FIRST_USB default=1, não reconhece, por padrão, teclados nas ttySx (COMs Legadas).
- Cria a variável de ambiente SK_8042=5 automaticamente no caso de controladora defeituosa.
- Melhoradas informações de log.
- Solução de Bugs.

Quem deve ler este manual:

Programadores que desejam se comunicar com um equipamento da Smak e procuram um driver/API/DLL. A `sk_access.dll/libsk_access.so` fornece funções que podem tornar mais rápido e fácil o desenvolvimento do programa aplicativo, pois já trata de várias questões de programação relativas ao relacionamento do hardware do PC com o equipamento Smak. Neste manual estão as descrições de todas as funções disponibilizadas pela dll/lib.

Programadores que desejam se comunicar diretamente com o hardware devem consultar a especificação do referido equipamento, no caso, por exemplo, do SKO-44 PS2, consultar **Especificacao_sko44_PS2.pdf**

Descrição da sk_access:

Para simplificar e agilizar o desenvolvimento de uma aplicação a SMAK desenvolveu a `sk_access` que é uma DLL(Dynamic Link Library) no Windows "`sk_access.dll`" e um SO(Shared Object) no Linux "`libsk_access.so`" que disponibiliza uma API(Application Programming Interface) com funções que permitem flexibilidade no acesso aos equipamentos possibilitando o gerenciamento do display assim como da interface de comunicação.

A dll/lib segue os padrões de passagem de parâmetros(Calling convention) para cada sistema operacional.

Por motivos de simplificação, vamos nos referir a DLL(Windows)/SO(Linux) simplesmente como lib.

Princípio de funcionamento:

A lib é única e pode ser usada para se comunicar com qualquer modelo de teclado Smak (HID, PS2, Serial e USB-VCP) .

A lib não possui uma função Open para iniciar suas operações, qualquer função da lib quando é invocada, verifica o status de conexão com um teclado e em caso negativo, executa os procedimentos adequados para estabelecer automaticamente esta conexão em qualquer interface disponível. A lib entretanto possui uma função de finalização **Free_Sk_Access**, que deve ser chamada para liberar recursos e evitar "vazamentos de memória".

É possível definir uma interface não automática usando a função `Set_interface` que força a interface de comunicação mesmo sem um teclado conectado e possibilita inclusive a coexistência de teclados com interfaces diferentes.

Após ser estabelecida comunicação com uma interface, só será reconhecida outra, após um reboot/reload da lib ou de um comando **Reset_Interface**.

O comando **Reset_Interface** pode ser executado a partir da própria lib ou chamando o programa "`reset_itf`" na linha de comando.

Importação da lib:

Executando os procedimentos abaixo, todas as funções da lib estarão disponíveis para uso.

Do VB: Está disponibilizado o arquivo **sk_access.vb6** com o “Declare” de todas as funções, podendo ser usado Ctrl C + Ctrl V e facilitar o trabalho de digitação.

Do Delphi/Lazarus: Para auxiliar a importação estática, está disponibilizado o arquivo **sk_access.inc**.

Somente acrescentar a diretiva de compilação **{ \$I sk_access.inc }**

Do C++:**-Windows:**

Para auxiliar a importação implícita*, está disponibilizado o arquivo de header **sk_access.h** e o arquivo para linkagem implícita **sk_access.lib**.

Acrescentar o arquivo **“sk_access.lib”** na lista de linkagem.

Acrescentar o Header:

#include "sk_access.h"

e chamar as funções:

Loadsk_access(); //Para carregar os nomes das funções

e

Freesk_access(); //Para liberar os recursos da lib.

-Linux:

Para auxiliar a importação implícita, está disponibilizado o arquivo de header **sk_access.h**.

Acrescentar a lib **“libsk_access.so”** na lista de linkagem.

Acrescentar o Header:

#include "sk_access.h".

E todas as funções da lib estarão disponíveis para uso.

**No Windows a linkagem implícita (estática) com sk_access.lib aponta para uma dll Stub que faz o carregamento explícito da sk_access.dll, a manobra é necessária porque a sk_access.dll não está codificada em MSC++ e não gera o arquivo .lib necessário para a linkagem.*

Plug and Play:

A biblioteca oferece uma plataforma unificada de desenvolvimento, de forma que um aplicativo feito utilizando suas funções, pode acessar indistintamente os teclados nas interfaces PS2, serial RS-232 / USB e HID sem modificação do código fonte do aplicativo do usuário.

Quando qualquer função da biblioteca é chamada, ele executa a função `Select_Interface` (Seleção automática).

A função `Select_Interface` procura por um teclado na interface HID, interface VCP, última COM legada selecionada, interface PS2 e depois outras COM's legadas (1..SK_FIRST_USB-1).

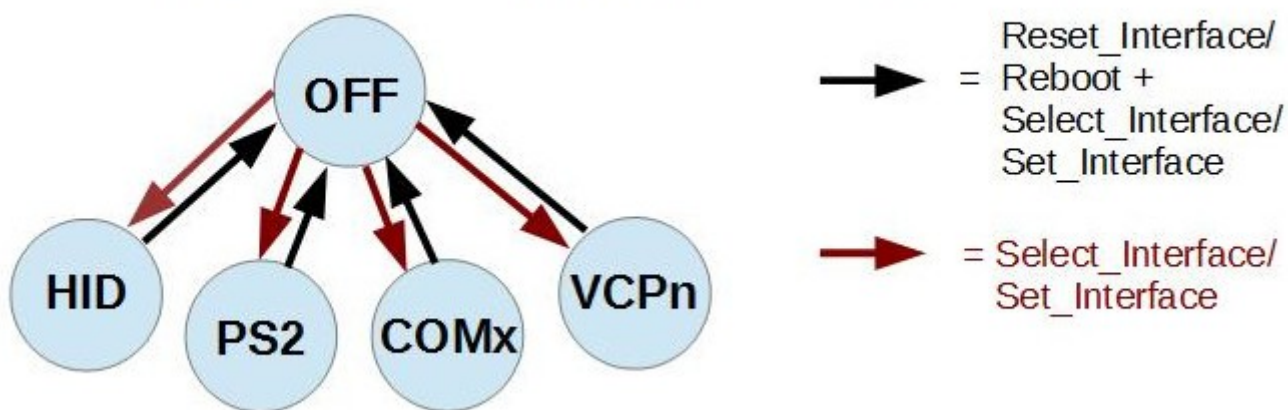
O valor de `SK_FIRST_USB` pode ser alterado como explicado mais adiante.

Se o teclado é encontrado em alguma destas interfaces ele é travada e a partir daí todas as operações são formatadas para esta interface até o reinício da lib.

Reiniciar a lib significa reiniciar o aplicativo que importou a lib.

Para poder trocar de interface sem reinício da lib, por exemplo depois de usar um teclado PS2 usar um teclado HID é necessário:

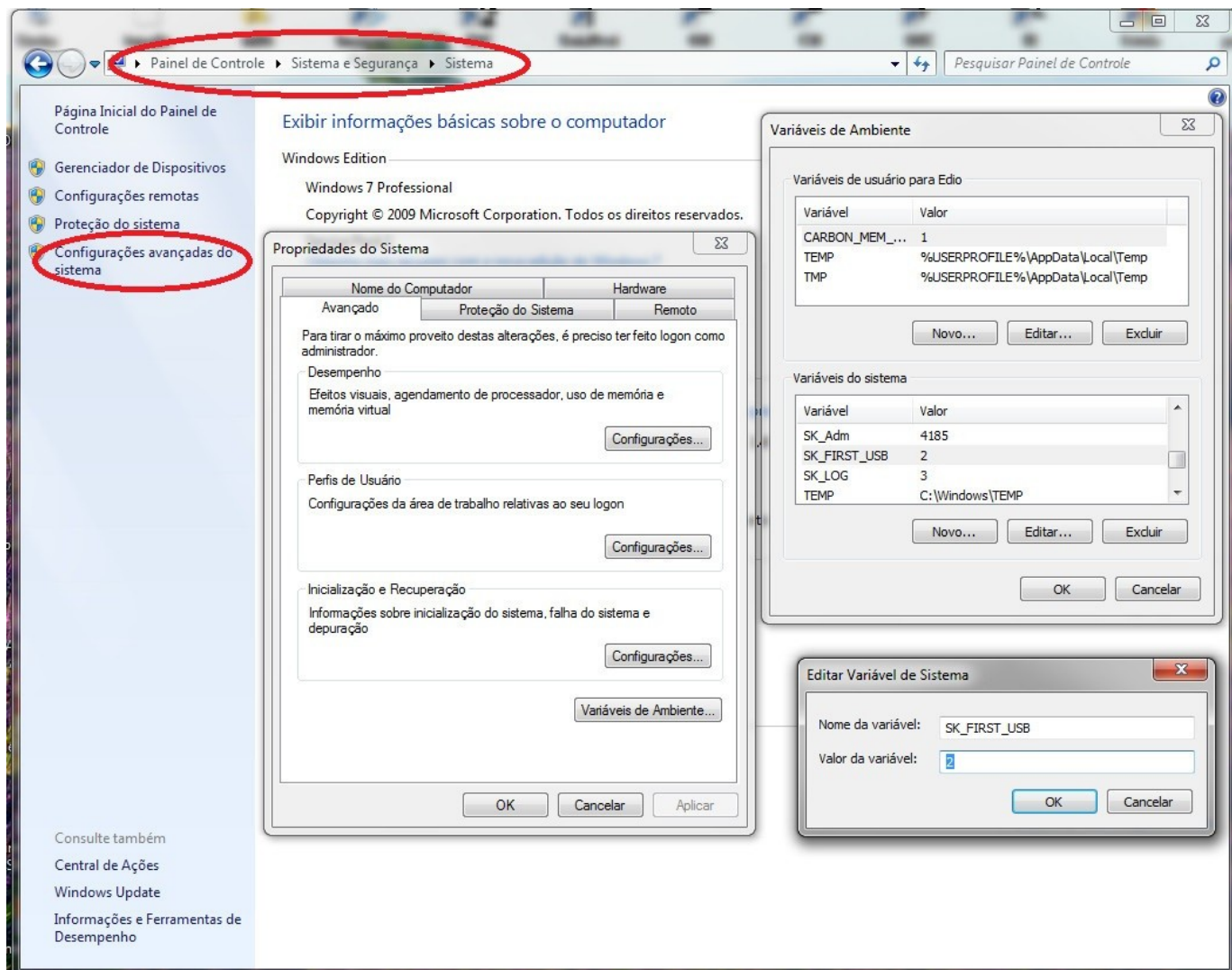
- a) Chamar a função **Reset_interface** da lib, ou
- b) Chamar o programa "**reset_itf.exe**" na linha de comando, ou

Diagrama de estados para troca de interfaces

Variáveis de ambiente:

A variável de ambiente **SK_FIRST_USB** redefine a primeira porta COM VCP acessada pela lib. Para criar/editar esta variável, no Windows, deve-se acessar:

(Win7) Painel de controle -> Sistema e Segurança -> Sistema -> Configurações Avançadas do sistema -> Variáveis de ambiente -> Novo/Editar



No Linux para criar a variável (por exemplo com o valor 5), colocar o comando
`export SK_FIRST_USB=5`

dentro arquivo .profile no diretório \$HOME

Este comando executado no prompt não será permanente e só valerá para sessão corrente.

Além da variável de ambiente SK_FIRST_USB, existem também as variáveis:
SK_8042 , SK_LOG, SK_ECO, SK_NOP2, SK_NOHID, SK_NOCP e SK_2303.

Se a variável não foi criada, seu valor é lido como 0 pela lib

SK_FIRST_USB define qual é o primeiro dispositivo considerado VCP/COM virtual. Dispositivos COM: com número abaixo de **SK_FIRST_USB** são consideradas COMs legadas. No Linux **SK_FIRST_USB** não pode ser maior do que 50.

Quando lib está procurando por uma interface, após varrer as interface HID, VCP e PS2, ela executa “um Pooling / uma comunicação” em cada COM legada para localizar um teclado conectado. Fazer **SK_FIRST_USB=1**(valor default) é o equivalente a dizer que não existem COMs legadas e portanto desativa esta busca.

Como o valor padrão/default da variável **SK_FIRST_USB=1** então para ativar as COMs legadas é necessário alterar este valor.

No Linux:

-Se **SK_FIRST_USB = 1** (valor default), uma tentativa de acesso à COM1: vai abrir /dev/ttyUSB0.

-Se **SK_FIRST_USB = 2**, uma tentativa de acesso à COM2 vai abrir /dev/ttyUSB0 e uma tentativa de acesso à COM1 vai abrir /dev/ttyS0.

-Uma tentativa de abrir COM50 vai abrir /dev/ttyACM0.

SK_8042 controla a forma com que a lib interage com a controladora de teclado PS2 e pode assumir valores de 0 a 5 e 10.

A controladora do ChipSet “Super IO F71869” da Fintech, usado nas MainBoard: IPX2500E1, também conhecida como RC8100 entre outras, pode apresentar um problema que atrapalha o envio de dados para o teclado causando perda de dados ou travamento da controladora. Existe a função **Unlock_Interface**, que pode ter sucesso no destravamento.

A lib faz um teste automático da controladora e gera **SK_8042=10** se ela está OK ou (**SK_8042=3** Windows / **SK_8042=5** Linux) para controladora que tem problemas.

SK_8042 pode ser alterado caso ainda haja problemas de comunicação com o teclado PS2 (teclado não mostra Texto no display, teclado não programa, etc), devem ser testados os valores 1..5 (começando pelo valor 1)para descobrir qual melhor se adapta à controladora em questão.

Se **SK_8042=5** é o valor escolhido, então um mouse PS2 não poderá ser usado em conjunto com o teclado neste computador.

Se a controladora tem problema e **SK_8042=5** não funciona, as outras opções dependem também do sistema de temporização do PC.

O PC pode se referenciar em três tipos de timer **TSC**, **HPET** e **ACPI**, normalmente o sistema usa o menos preciso que é o TSC, para forçar o uso de um timer melhor, proceder:

No WINxp: colocar a opção /usepmtimer no arquivo c:/boot.ini.

[operating systems]

```
multi(0)disk(0)rdisk(0)partition(1)\WINDOWS="Microsoft Windows XP Home Edition"  
/noexecute=optin /fastdetect /usepmtimer
```

No WiN7...: Digitar no prompt

```
C:\Windows\system32>bcdedit /set USEPLATFORMCLOCK on
```

No Linux: Digitar no prompt

```
# echo "acpi_pm" > /sys/devices/system/clocksource/clocksource0/current_clocksource
```

SK_LOG define o nível de registro do LOG, pode assumir valores de 0..9, quanto maior o valor, mais detalhado é o LOG gerado.

A lib usa o nível de LOG que for maior, entre o valor da variável **SK_LOG** e o valor estabelecido pela chamada a função **Log_On**.

Se **SK_LOG** é definido, o arquivo **sk_access_log** será gerado no mesmo sub-diretório do aplicativo que importou a lib.

SK_ECO Permite o uso do ECO da lib com linguagens que usam UNICODE.

Se **SK_ECO** = 0 → não UNICODE.

Se **SK_ECO** =1 ou 2 → UNICODE. Verificar qual funciona com a sua linguagem.

SK_NOP52 se for criada com qualquer valor diferente de 0(zero), **desabilita** a busca por teclados PS2.

SK_NOHID se for criada com qualquer valor diferente de 0(zero), **desabilita** a busca por teclados HID.

SK_NOCP se for criada com qualquer valor diferente de 0(zero), **desabilita** a busca por teclados Seriais Legados e VCP.

SK_2303 se for criada com qualquer valor diferente de 0(zero), **habilita** a busca por teclados Seriais VCP que usam o componente Prolific PL2303.

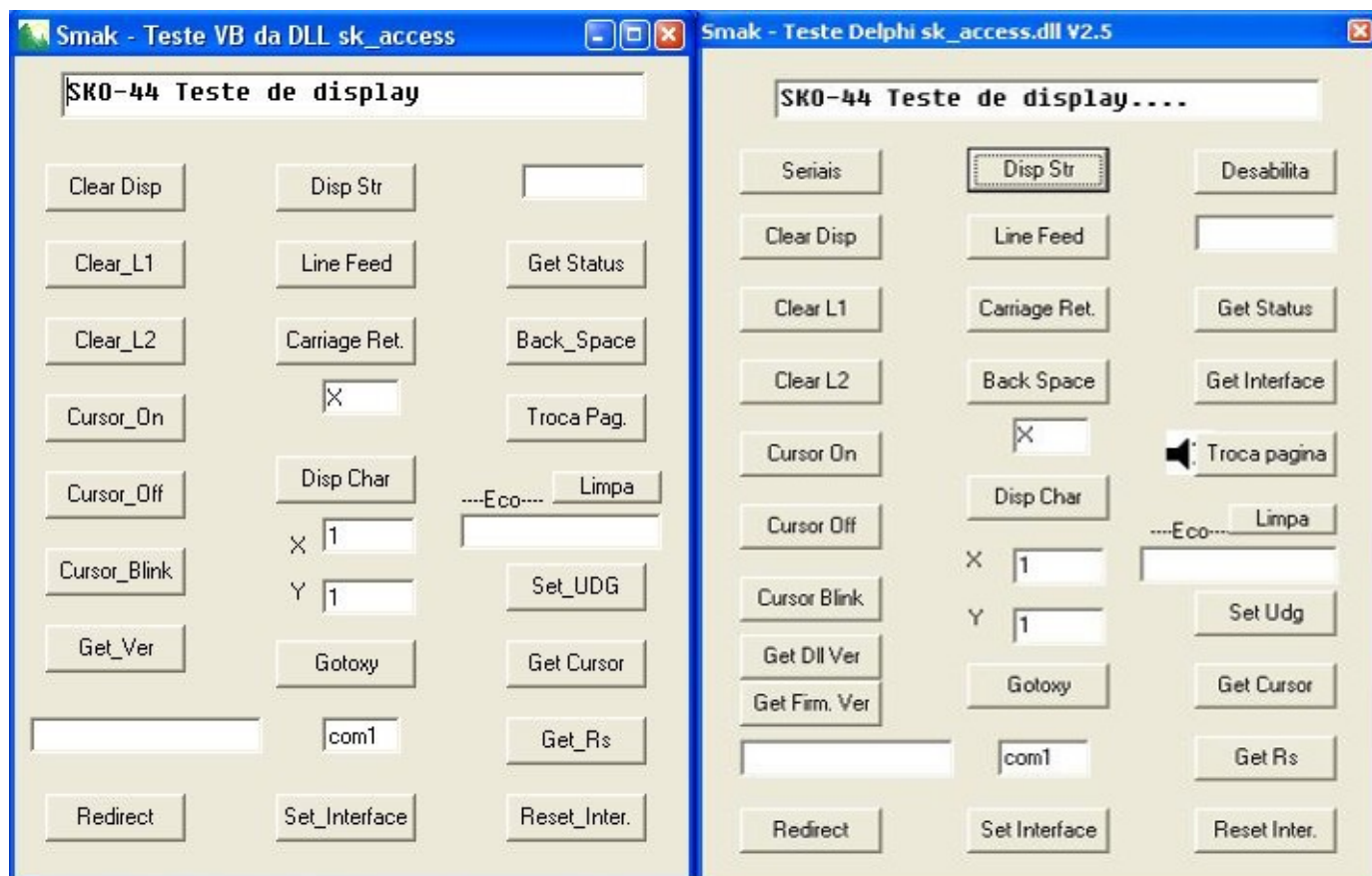
As variáveis de ambiente são armazenadas no registro do Windows em:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Environment

Demonstração e testes das funções da biblioteca Smak:

O utilitário de teste (Windows) disponibilizado em Delphi7 e VB6 (contendo inclusive o código fonte) foi desenvolvido especificamente como um exemplo para verificação da operação das funções exportadas pela biblioteca. A figura a seguir ilustra sua aparência no Windows.

Cada botão tem o nome da função sendo testada.



Instalando a lib:

Windows:

A instalação da biblioteca consiste em obter o pacote de software SKO_Tool_BoxVxxx.exe disponível no site www.smak.com.br, executá-lo e responder às perguntas de instalação.

Linux:

Instalação em Linux Kernel 2.6.xx

Copiar libsk_access.so para \usr\lib

Testes:

Conecte ao computador um teclado SKO-44 em seguida execute o utilitário de testes e teste alguns comandos.

Exemplos de testes:

- Teste da função DISP: Digite um texto na caixa de texto superior e aperte o botão *“Disp Str”*. O texto digitado deverá aparecer no display do SKO-44.
- Teste da função CLEAR_L1 e CLEAR_L2: Aperte os botões *“Clear_L1”* e *“Clear_L2”* para limpar as linhas do display.

Para saber sobre todos os testes consulte a informação na tabela no final deste documento, a qual contém a descrição das funções exportadas pela biblioteca.

Utilizando a biblioteca da SMAK em seu ambiente de desenvolvimento:

Uma vez que você conseguiu verificar o correto funcionamento do utilitário de testes, isto significa que *“sk_access.dll”* / *“libsk_access.so”* está corretamente disponível para uso no sistema. É interessante lembrar que em caso de dúvidas sobre o uso das funções da biblioteca, o profissional desenvolvedor poderá consultar o código fonte do aplicativo de testes disponibilizado em VB e Delphi e Cpp.

Convenção de chamada:

Nas funções abaixo, substituir *_mode* pela convenção de chamada (Calling Convention) apropriada.

A lib segue a convenção padrão do sistema operacional sendo:

_mode = *_stdcall* (ou *WINAPI*) em Windows.
_mode = *_cdecl* em Linux.

“char” representa um tipo que ocupa um byte (UTF-8).

Funções exportadas pela lib:

Veja a seguir a tabela com as funções e procedimentos disponíveis na biblioteca Smak.

PROTÓTIPO	DESCRIÇÃO	WIN	LINUX
void _mode Back_Space();	Back space do cursor	X	X
int _mode Buffer_In_Data();	Retorna número de dados no buffer de entrada		X
void _mode Carriage_Return();	Carriage return	X	X
void _mode Clear_Disp();	Apaga o display	X	X
void _mode Clear_L1();	Apaga a linha 1 do display	X	X
void _mode Clear_L2();	Apaga a linha 2 do display	X	X
void _mode Cursor_Blink();	Piscar o cursor	X	X
void _mode Cursor_Off();	Desligar o cursor no display	X	X
void _mode Cursor_On();	Ligar o cursor	X	X
void _mode Disable_Keyboard();	Desativa varredura do teclado	X	X
void _mode Disp_Char(char data);	Exibe um caracter no display	X	X
void _mode Disp(char *data);	Exibe uma string de até 80 caracteres.	X	X
void _mode Enable_Keyboard();	Ativa varredura do Teclado	X	X
void _mode Free_Sk_Access();	Remove hooks e libera memória	X	X
bool _mode Get_Com_Type();	Retorna tipo da porta COM: TRUE = VCP, FALSE = Legada	X	X
void _mode Get_Coms(char *Coms);	Retorna uma string com todas as COM's disponíveis	X	
int _mode Get_Current_Interface();	-1 = nenhuma interface 0 = PS2 , 1..n = COM1..COMn, 100=HID	X	X
char _mode Get_Cursor(char *x_pos, char *y_pos);	Lê coordenadas em x_pos e y_pos (home = 1,1), Função retorna posição absoluta (0 a 79) ou Time-out = 255.	X	X
void _mode Get_Data	'= Get_Data_Rs	X	X
int _mode Get_Data_Rs (char *buffer, int size);	Transfere para "buffer" um máximo de "size" caracteres lidos da porta serial. Retorna o número de caracteres efetivamente lidos.	X	X
void _mode Get_Dll_Version(char *ver);	Devolve na variável ver, a versão da DLL. "char *ver" deve ser maior que 20 caracteres.	X	X
void _mode Get_Firmware_Version (char *ver);	Retorna (HID)=v.vvv-c.ccc ou (PS2)=v.vvv ou (RS)=v.vvv, "char *ver" deve ser maior que 25 caracteres.	X	X
void _mode Get_Scan_Id(char *id);	Le Scan_Id gravado com diretiva \$Scan no arquivo .smk	X	
void _mode Get_Serial_Number (char *ser);	Retorna string com número de série. "char *ser" deve ser maior que 20 caracteres.	X	X
char _mode Get_Status();	Status da última operação, Retorna: 0 = OK ; 1 = erro e FFh = time-out.	X	X

PROTÓTIPO	DESCRIÇÃO	WIN	LINUX
int _mode GetOperatingSystem();	Identifica sistema operacional, Ver Nota	X	X
void _mode Gotoxy(char x , char y);	Posiciona em Linha Y, Coluna X, Home = (1,1).	X	X
void _mode Keyb_Reset();	Reseta o teclado.	X	X
void _mode Line_Feed();	Line-feed no display	X	X
void _mode Log_Off();	Desliga registro de LOG	X	X
void _mode Log_On(int dbg_level);	Será criado o arquivo sk_access_log com registro das atividades da DLL, dbg_level (1..9) controla detalhamento do log.	X	X
void _mode Redirect();	Captura a porta serial e redireciona para o buffer do teclado. Ver nota	X	X
void _mode Reset_Interface();	Esquece as informações de Interface.	X	X
void _mode Scroll_Enable_Off();	Tecla Scroll-lock Ativa/Desativa teclado serial	X	
void _mode Scroll_Enable_On();		X	
int _mode Select_Interface();	Procura por um teclado SKO44 conectado, Retorna: -1=ñ encontrado, 0= PS2, 1..99=COM(1..99), 100=HID	X	X
void _mode Send_Data(char data);	Envia um byte para o teclado	X	X
void _mode Send_Disp_Ctrl(char data);	Envia comando direto para o display. Ver nota	X	X
void _mode Set_Beep_Off();	Desliga Beep do teclado	X	X
void _mode Set_Beep_On();	Liga Beep do teclado	X	X
void _mode Set_Eco_Off();	Desliga eco	X	X
void _mode Set_Eco_On();	Liga eco	X	X
int _mode Set_Interface(char *interf);	Fixa Interface, parâmetro ='HID', 'VCP', 'PS2', 'COM1'.. 'COMn'. Retorna: -1=ñ encontrado, 0= PS2, 1..99=COM(1..99), 100=HID	X	X
void _mode Set_Page0();	Ativa pagina 0 de scancode.	X	X
void _mode Set_Page1();	Ativa pagina 1 de scancode.	X	X
void _mode Set_Psw_Off();	Desliga eco de senha	X	X
void _mode Set_Psw_On();	Liga eco de senha = *	X	X

PROTÓTIPO	DESCRIÇÃO	WIN	LINUX
void _mode Set_Udg (char caracter, char def0, char def1, char def2, char def3, char def4, char def5, char def6, char def7);	UDG (User Defined Graphics) Permite o desenho de até oito caracteres especiais que ficam armazenados nas posições 0 a 7 da tabela interna do display. Na procedure, o parâmetro “ <i>caracter</i> ” representa o código a ser atribuído ao desenho. “ <i>def 0</i> ” a “ <i>def 7</i> ” definem os elementos ativos da matriz 5x7 do caracter no display. Veja abaixo um exemplo dos valores de “ <i>caracter</i> ” e de “ <i>def 0 a 7</i> ” para formar a letra 'R' na posição 01h da tabela interna	X	X
	caracter=01h		
	def1 = x x x 1 1 1 1 0 = 1Eh		
	def2 = x x x 1 0 0 0 1 = 11h		
	def3 = x x x 1 0 0 0 1 = 11h		
	def4 = x x x 1 1 1 1 0 = 1Eh		
	def5 = x x x 1 0 1 0 0 = 14h		
	def6 = x x x 1 0 0 1 0 = 12h		
	def7 = x x x 1 0 0 0 1 = 11h		
Char _mode Unlock_Interface();	Tentar destravar controladora PS2	X	X
void _mode Update_Leds_Off();	Não envia atualização dos LED's para a porta auxiliar do teclado serial	X	
void _mode Update_Leds_On();	Envia atualização dos LED's para a porta auxiliar do teclado serial	X	

Nota sobre a função GetOperatingSystem:

Retorno da função:

WINDOWS:

-1=Desconhecido;	0=Win95;
1=Win98;	2=Win98se;
3=WinME;	4=WinNT;
5=Win2000;	6=XP;
7=XPpro64;	8=Server2003;
9=HomeServer;	10 = Server2003R2;
11= Vista;	12 = Server2008;
13 = Server2008R2;	14 = Win7;
15 = Win8;	16 = Win8.1
17 = Win10	

LINUX:

22, 24 ,26 para Kernel 2.2, 2.4, 2.6.

Nota sobre a função Redirect:

A função Redirect serve como driver para os teclados Smak(sem display) seriais Legados e VCP, possibilitando que os teclados sejam usados normalmente como se fossem teclados HID.

Nota sobre a função Send_Disp_Ctrl:

O controle do display do teclado pode ser acessado diretamente através do comando "Send_Disp_Ctrl", a tabela abaixo mostra como deve ser composto um comando.

Aqui estão apresentados todos os comandos aceitos pelo display, alguns destes comandos podem desconfigurar o display tornando-se necessário um reboot (liga/desliga) do teclado.

INSTRUÇÃO	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	DESCRIÇÃO
Clear Disp	0	0	0	0	0	0	0	1	Apaga Display e coloca cursor em (1,1)
Home	0	0	0	0	0	0	1	*	Posiciona cursor em (1,1)
									Retorna mensagem deslocada
Mode	0	0	0	0	0	1	I/D	S	I/D = sentido de deslocamento do cursor
									S = Desloca ou não a mensagem
Control	0	0	0	0	1	D	C	B	D = Ativa/desativa display
									C = Ativa/desativa cursor
									B = Ativa/desativa intermitência do cursor
Shift	0	0	0	1	S/C	R/L	*	*	S/C = Movimento do cursor
									R/L = Sentido de deslocamento da mensagem
INTERFACE	0	0	1	DL	N	F	*	*	DL = Número de bits da interface
									N = Número de linhas do display
									F = Tipo de formatação do caracter
CG Ram	0	1	Endereço						Apona endereço corrente da memoria gráfica
DD Ram	1	Endereço							Apona endereço corrente da memoria de dados

Ex.: Para apagar o display enviar 0x01

Para colocar o cursor em home enviar 0x02

Upper 4bit Lower 4bit		LLLL	LLH	LLHL	LLHH	LHLL	LHLH	LHHL	LHHH	HLLL	HLLH	HLHL	HLHH	HHLL	HHLH	HHHL	HHHH
LLLL	CG RAM (1)																
LLH	(2)																
LLHL	(3)																
LLHH	(4)																
LHLL	(5)																
LHLH	(6)																
LHHL	(7)																
LHHH	(8)																
HLLL	(1)																
HLLH	(2)																
HLHL	(3)																
HLHH	(4)																
HHLL	(5)																
HHLH	(6)																
HHHL	(7)																
HHHH	(8)																

Notas sobre versões de Firmware do SKO-44:

Teclados PS2:

Versões < 1.93d (anterior a Agosto-2007)

Não respondem a Get_Status. E não são reconhecidos pela busca automática.

Deve ser usado explicitamente Set_Interface("PS2");

Versões < 1.96 (anterior a Dezembro-2008)

Não respondem a Get_Firmware_Version

Versões entre 1.95 e 1.96 travam se for usada a função Get_Firmware_Version

Teclados Seriais Legados/VCP:

Versões ente 1.92 e 1.93 (anterior a Setembro-2007) travam se for usada a função

Get_Firmware_Version

Teclados HID:

Sem ressalvas.