

Arquivos de configuração SMK



Foto: SKO-44

Teclado de automação, Tecnologia Óptica
com display LCD e leitor magnético.

Descrição:

A programação dos teclados SMAK é executada à partir de um arquivo de texto contendo as informações necessárias para configurar o teclado. O arquivo de configuração pode facilmente ser criado ou modificado utilizando qualquer editor de textos simples e deve receber a extensão ".smk". Este documento descreve as regras de sintaxe de arquivos SMK e outros detalhes referentes a configuração dos teclados programáveis SMAK.

Rev. 1.95

Sumário

Histórico de alterações deste documento.....	3
Quem deve ler este manual.....	4
Convenções usadas neste documento.....	4
Extensão do arquivo de configuração.....	4
Produtos aplicáveis.....	4
Como são programados os teclados SMAK.....	5
Compatibilidade de Firmware.....	6
Criando um arquivo SMK.....	6
Elementos e regras de sintaxe.....	8
Definições de Teclas.....	8
Teclas com Shift.....	10
Teclas com Typematic.....	11
Tecla PAGE.....	12
Palavras reservadas.....	12
Definição de Strings.....	14
Definições em Hexa.....	15
Diretivas.....	16
Diretivas específicas para leitores de cartão magnético.....	18
- Diretivas para delimitadores preestabelecidas.....	19
- diretivas para especificar delimitadores customizados.....	19
Exemplo de um arquivo SMK com uma página de código de 44 teclas.....	21
Exemplo de um arquivo SMK com duas páginas de código de 44 teclas.....	23
Exemplo configuração para um teclado SERIAL tipo VT100 (ASCII).....	26
A1 - Tabelas de definições de teclas afetadas pelas diretivas &Abnt e &Inglês.....	28
A2 - Tabelas de Scancodes Inglês / Abnt2.....	29
A3 - Tabela ASCII internacional.....	30
A4 - Tabela Scancodes PS2emul.....	31
A5 - Tabela de conversão AT - XT.....	32
A6 - Folha para planejamento de Layout de Configuração.....	33

Histórico de alterações deste documento:Revisão 1.95 (2019) :

- Revisão de textos.
- Acrescentada informação sobre string Hexa.

Revisão 1.94 (--) :

- Acrescentada a definição (s){};
- Acrescentada Quem deve ler este manual

Revisão 1.93 (09-05-2014) :

- Revisão de textos.
- Acrescida interface HID.
- Acrescido InstallDriver para Win7 e 8.
- Alterado texto relativo aos prefixos \$, #, t e b

Quem deve ler este manual:

Quem deseja entender a sintaxe do arquivo de configuração de teclados Smak, os arquivos com extensão .smk.

O arquivo de configuração possibilita configurar todas as características dos teclados programáveis em qualquer das interfaces disponíveis (PS2, USB-VCP, USB-HID, Serial e Ethernet).

O arquivo .smk configura por exemplo, os códigos enviados pelas teclas, os delimitadores do leitor de cartão magnético e a mensagem de saudação do display.

O arquivo .smk é interpretado pelo aplicativo "Upload_Win.exe" informações em "MANUAL_UPLOADWIN.PDF".

Quem não deseja entender a sintaxe do arquivo .smk pode usar o aplicativo gráfico "Smk_Wizard.exe" informações em "MANUAL_SMK_WIZARD.PDF".

Convenções usadas neste documento:

Textos exibidos entre os sinais "<" e ">" significam teclas existentes em teclados de PC.

exemplos:

<CTRL> : significa a tecla CONTROL do teclado.

<CTRL>+<X> : significa manter a tecla "CONTROL" pressionada e em seguida pressionar a tecla "X".

Nota: o sinal de adição entre duas ou mais teclas indica que as teclas anteriores são mantidas pressionadas enquanto pressionamos a seguinte. Para simbolizar a tecla "+" usaremos a representação <+>.

Extensão do arquivo de configuração:

A programação dos teclados SKO44 é baseada na interpretação de um arquivo de texto que deve receber a extensão ".smk". por ex. "teste1.smk".

Produtos aplicáveis:

O assunto tratado neste documento refere-se aos seguintes produtos:

- SKO44 - interface PS/2.
- SKO44 - interface serial a partir do Firmware v1.91
- SKO44 - interface USB VCP.
- SKO44 - interface USB HID.
- SKO44 - interface Ethernet.

Como são programados os teclados SMAK :

O utilitário "*upload_win.exe*" (Windows) interpreta o conteúdo do arquivo SMK e programa o teclado do modo desejado.(ver [guia_programação.pdf](#)).

As tabelas a seguir ilustram mais detalhadamente todo o software requerido, que se encontra dentro do pacote *Sko_Tool_BoxVxx.exe* que pode ser obtido via download no site www.smak.com.br.

WINDOWS-XP:

Arquivo	Descrição
inpout32.dll	DLL para uso no Windows-XP
sk_access.dll	DLL requerida para uso no Windows
uploadwin.exe	Utilitário que faz a interpretação e carga do arquivo SMK

WINDOWS-7 a 10:

Arquivo	Descrição
inpout32.dll	DLL para uso no Windows-XP
sk_access.dll	DLL requerida para uso no Windows
uploadwin.exe	Utilitário que faz a interpretação e carga do arquivo SMK
InstallDriver.exe	Instala driver de sistema para acesso a porta PS2

LINUX:

Arquivo	Descrição
libsk_access.so	biblioteca para uso em Linux

Nota: Os downloads disponibilizados no site www.smak.com.br contém também informações adicionais sobre a instalação e uso de cada software.

O arquivo de configuração SMK

Compatibilidade de Firmware:

O aplicativo Upload_Win Localiza, identifica e programa os teclados Smak de acordo com a versão do Firmware do teclado e a interface selecionada, entretanto, teclados PS2 com versão de Firmware anterior à 1.93c (Agosto-2007) necessitam que seja habilitada Ferramentas → Opções → Dumb_Mode.

Atenção para que o arquivo .smk indique a mesma interface do teclado que se deseja programar, senão será emitida uma mensagem “Teclado não encontrado”.

Criando um arquivo SMK :

Usando o editor interno de Upload_Win ou um editor de textos simples semelhante ao Notepad, podemos iniciar a criação de um arquivo de configuração o qual deverá obrigatoriamente ter a extensão ".smk". O objetivo é especificar as configurações de teclas a serem programadas, além de outras características configuráveis do teclado programável como por exemplo a mensagem de saudação a ser exibida no Display.

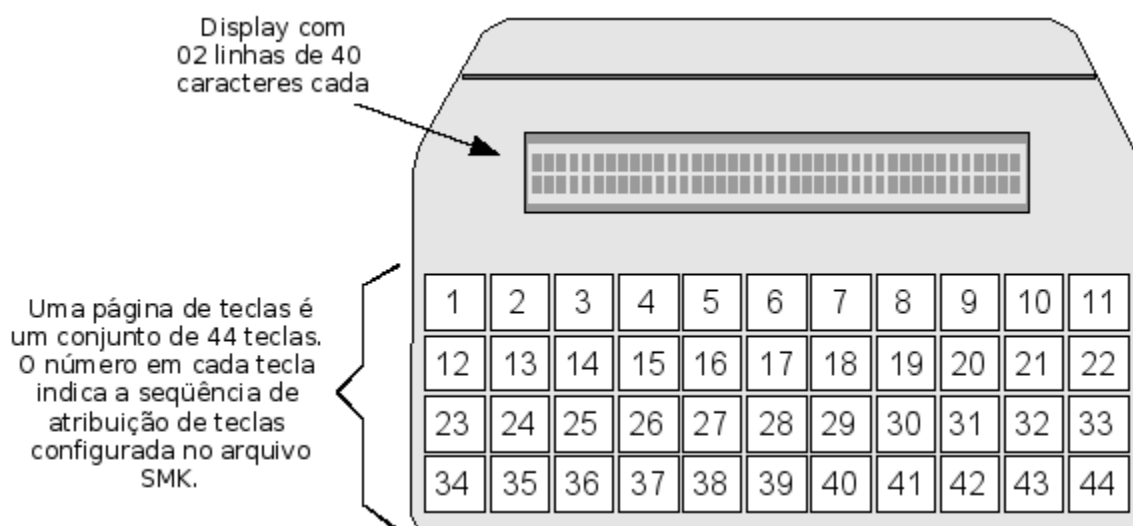
O arquivo SMK deve obedecer a uma sintaxe especial. Por exemplo, o início e fim do arquivo são determinados por rótulos entre colchetes " [] ".

No início do arquivo, o rótulo especifica o produto ao qual se destina, no nosso caso [SKO44]. O rótulo de final é sempre [FIM].

Os códigos são atribuídos a cada tecla através de uma expressão de definição de tecla. Durante a transferência das configurações ao teclado as definições vão sendo lidas do arquivo SMK e configuradas sequencialmente de forma que a 1ª expressão de definição encontrada vai ser atribuída a tecla 1, a 2ª vai para a tecla 2, a 3ª para a tecla 3, e assim por diante até que as 44 teclas tenham sido programadas.

Nota: As definições são associadas a cada tecla no teclado seguindo a ordem da esquerda para a direita e de cima para baixo. Veja nos exemplos no fim deste documento que para facilitar a compreensão agrupamos as definições de teclas em 4 grupos de 11 teclas, correspondendo às 4 filas de teclas do teclado.

Conforme podemos observar na figura a seguir, o SKO44 é um teclado de 44 teclas dispostas em 4 linhas de 11 teclas.



Sintaxe do arquivo SMK

O SKO-44 permite a programação de até 88 teclas, ou seja, até duas páginas de teclas.

Neste caso podemos criar um arquivo SMK para:

- a) configurar um conjunto de 44 teclas (uma página de teclas); ou,
- b) configurar dois conjuntos distintos de 44 teclas (duas páginas de teclas).

Ao escolher a opção b, deve-se também planejar como serão acessadas as duas páginas de códigos configuradas. A solução mais simples seria reservar uma tecla para executar a mudança de página. A tecla programada para este fim deve ocupar a mesma posição nas duas páginas de código. outra solução seria a mudança de página via software, o que permite dispensar a configuração de uma tecla de mudança de página. Consulte a documentação da DLL "*sk_access.dll*" para maiores informações sobre a mudança de página via software.

Elementos e regras de sintaxe:

Comentários: São anotações úteis que você pode fazer em seu arquivo SMK, devem ser escritos entre chaves. São ignorados pelo interpretador não tendo nenhuma outra função.
exemplo: { por estar entre chaves, isto seria ignorado dentro de um arquivo SMK}

Rótulos: A primeira e a última linha válida do arquivo começa com um rótulo entre colchetes "[]" que identifica respectivamente o início e o fim do arquivo de configuração. O rótulo de início válido é [SKO44] e o rótulo de finalização é sempre [FIM]. Não deve haver espaços dentro dos colchetes.

Diretivas: Iniciam sempre com "&" e servem para definir algumas características pré estabelecidas. Todas as diretivas são detalhadas adiante neste documento.

Definições de teclas: São as expressões que vão definir o código associado a cada tecla. As definições de teclas seguem o formato **(Prefixo)Código** onde: *(Prefixo)* representa um modificador de tecla e *Código* representa a tecla a ser enviada. Todas as definições de teclas são detalhadas nas próximas páginas.

Nota: A sintaxe do arquivo SMK pode ser "case-sensitive" ou "case-insensitive" (Faz diferenciação entre letras Maiúsculas e minúsculas ou não), dependendo da diretiva de interface utilizada.

Definições de Teclas:

O formato geral de uma expressão de definição de teclas é: **(PREFIXO) CÓDIGO**

(PREFIXO): É um elemento obrigatório, serve para representar um modificador de tecla. Caso não se queira usar nenhum modificador de tecla deve-se usar "()".
Exemplo: ()a

CÓDIGO: Um símbolo, letra, dígito ou palavra reservada para representar uma tecla. No ex. Acima a letra "a" representa o código da tecla <A> e define o Scancode daquela tecla.

É possível usar letras maiúsculas, minúsculas, dígitos, símbolos como " - = [~] / ; \ , . * e também algumas palavras reservadas para produzir as teclas que não podem ser representadas por um único símbolo.

A tabela a seguir resume o formato para definições de teclas:

DEFINIÇÕES DE TECLAS		Suportado pela diretiva	
(PREFIXO)	DESCRIÇÃO	&PS2 &PS2_EMUL &HID &USB &TCP_IP	&ASCII &TCP_IP2
()X	Envia uma tecla sem nenhum modificador. X pode ser um caracter, dígito, símbolo ou palavra reservada destinada a simbolizar uma tecla.	SIM	SIM
(a)X	define uma tecla p/ gerar: <ALT> + <X>	SIM	NÃO
(c)X	define uma tecla p/ gerar: <CTRL> + <X>	SIM	NÃO
(s)X	define uma tecla p/ gerar: <SHIFT> + <X>	SIM	NÃO
(b)X	Conforme o padrão da interface em uso, ativa ou desativa a geração do código BREAK da tecla. *1 Desativa a geração do BREAK, gerando apenas o MAKE. *2 Ativa o envio de um código BREAK formado pelo MAKE do caracter ASCII com o bit 7 = "1". Por ex. (b)A vai gerar: Make[0x41] e Break[0xC1]	SIM *1	SIM *2
(\$)XXX...	Configura a tecla para gerar uma string de até 64 caracteres.	SIM	SIM
(t)X	Ativa TYPEMATIC (auto repetição) da tecla	SIM	SIM
(#)HH	Envia o código Hexa HH como Make e F0h HH como Break	SIM	SIM
Notas:			
1. Além disso é possível fazer qualquer combinação ente a, c, s, b. Por ex.: (as)X, (ca)X, (cas)X			
2.Os prefixos \$,# e t só podem ser usados em conjunto com o prefixo b			
3. De acordo com o tipo de teclado (PS/2, Serial ou USB) há variações das possibilidades de programação, por ex.: um teclado serial tipo VT100 (configurado pela diretiva &ASCII) está limitado aos 128 símbolos da tabela ASCII padrão. A tentativa de programar um símbolo não suportado fará UploadWin emitir um erro de sintaxe.			

É importante notar que as diretivas &PS2, &PS2_EMUL, &TCP_IP, &TCP_IP2, &ASCII, &HID e &USB determinam o contexto de funcionamento do teclado. Ao interpretar o arquivo SMK, UploadWin determina que tipo de teclado deve ser programado, se não encontrar um teclado na interface especificada, será emitida uma mensagem de erro.

Para maiores detalhes, veja a tabela de diretivas.

Com as diretivas &PS2, &PS2_EMUL, &HID, &USB e &TCP_IP as definições do arquivo .smk são “case-insensitive”, ou seja, não diferencia letras maiúsculas de minúsculas.

Com as diretivas &ASCII e &TCP_IP2 as definições são “case-sensitive” Diferenciando letras maiúsculas de minúsculas.

Veja os exemplos a seguir:

Definindo uma tecla com diretiva &PS2, &PS2_EMUL, &HID, &USB ou &TCP_IP.

()A : Configura o código correspondente a tecla <A>

()a : idem ao anterior, pois trata-se do mesmo Scancode (mesma tecla).

Definindo uma tecla com diretiva &ASCII e &TCP_IP2.

()A : Configura o código ASCII correspondente ao símbolo "A"

()a : Configura o código ASCII correspondente ao símbolo "a"

Definindo uma tecla com modificador "Shift" em um arquivo com diretiva &PS2, &PS2_EMUL, &HID, &USB ou &TCP_IP.

(s)a ou (s)A : configura o código correspondente a <SHIFT>+<A>

Em um teclado SERIAL usando a diretiva &ASCII não é possível a definição acima:

(s)X : *** VAI ACUSAR ERRO ***

Nas diretivas &ASCII e &TCP_IP2 os prefixos s,a,c não são aceitos veja quadro "Definição de Teclas"

Teclas com Shift:

Os símbolos “ , ! , @ , # , \$, % , & , * , (,) , _ , + , ` , { , ^ , } , | , < , > , : , ; , ? ” estão nas teclas ‘ , 1 , 2 , 3 , 4 , 5 , 7 , 8 , 9 , 0 , - , = , ' , [, ~ ,] , \ , , , . , ; , /

Nos teclados comuns (PS2 ou USB) esses símbolos são acessados com a ajuda da tecla <SHIFT>.

Para gerar esses símbolos com as diretivas &PS2, &PS2_EMUL, &HID, &USB ou &TCP_IP. é necessário o uso do prefixo (s), pois nessas diretivas <1> e <!> tal como <2> e <@> são consideradas a mesma tecla, Exemplo:

()5 ou ()% gera a digitação de <5> e é mostrado no console 5.

(s)1 ou (s)! gera a digitação de <Shift>+<1> e é mostrado no console !

(s)9 ou (s)(gera a digitação de <Shift>+<9> e é mostrado no console (

(s)[gera a digitação de <Shift>+<[> e é mostrado no console {

(s){ gera erro porque { é usado para definir um comentário, para que a definição seja aceita deve ser escrita como (s){ {

Para gerar esses símbolos com as diretivas &ASCII e &TCP_IP2 o uso do prefixo não é necessário, Exemplo:

()1 gera “1”

()! gera “!”

Teclas com Typematic:

As teclas com prefixo (t) Typematic, geram auto repetição, ou seja, depois de pressionada, a tecla fica gerando o seu código repetidamente a intervalos regulares até que seja solta.

Windows x Linux:

As máquinas Windows usam a característica de auto repetição do teclado, sendo inclusive possível em “Painel de controle → Teclado” programar essa taxa de repetição. Essa programação é repassada ao teclado, entretanto o SKO-44 possui taxa de repetição fixa e essa programação é ignorada.

As máquinas Linux ignoram as repetições enviadas pelo teclado, o Linux reconhece somente quando a tecla é pressionada e quando a mesma é solta. A repetição de teclas no Linux é gerada pelo próprio sistema, quando a taxa de repetição é alterada o sistema é reprogramado mas o teclado não.

Essa diferença entre os sistemas ocasiona a seguinte situação:

No Linux o prefixo (t) é irrelevante, pois todas as teclas vão ter auto repetição gerada pelo sistema.

No Windows somente as teclas com prefixo (t) terão auto repetição entretanto a taxa de repetição é fixa, pois o teclado não aceita reprogramação.

Obs. O prefixo (t) só pode ser usado em conjunto com o prefixo b.

Tecla PAGE:

A tecla PAGE é usada para selecionar entre as 2 páginas de teclas programadas no teclado, portanto só deve ser usada em configurações com duas páginas (Diretiva &page2)

Quando utilizada deve ser programada na mesma posição de tecla nas duas páginas.

Uma configuração com duas páginas sem tecla PAGE, só poderá mudar de página por um comando do Host(PC).

A tecla PAGE pode operar de duas maneiras: com trava e sem trava.

Com trava significa que a cada vez que pressionamos a tecla PAGE ela troca a página ativa. No modo sem trava, a tecla PAGE seleciona a segunda página somente enquanto está pressionada voltando a ativar a primeira página quando é solta.

Em configurações &PS2, &PS2_EMUL, &HID, &USB ou &TCP_IP:

()PAGE Define uma tecla PAGE sem trava.

(b)PAGE Define uma tecla PAGE com trava.

Em configurações &ASCII e &TCP_IP2:

()PAGE Define uma tecla PAGE com trava.

(b)PAGE Define uma tecla PAGE sem trava.

Palavras reservadas :

Conforme já foi mencionado, é necessário o uso de algumas palavras especiais para a definição daquelas teclas que não podem ser definidas com um único símbolo. Por ex. para programar uma tecla <ENTER> a definição de tecla é: **()ENTER**. A tabela a seguir contém todas as palavras reservadas, e sua validade conforme o tipo de configuração.

TABELA DE PALAVRAS RESERVADAS

PALAVRA RESERVADA	SIGNIFICADO	Suportado pela diretiva	
		&PS2 &PS2_EMUL &HID &USB &TCP_IP	&ASCII &TCP_IP2
F1 à F12	representa as teclas <F1> à <F12>	Sim	Não
ESC	tecla <ESC>	Sim	Sim
TAB	tecla <TAB>	Sim	Sim
CAPS	tecla <CapsLock>	Sim	Não
SCRL	tecla <ScrollLock>	Sim	Não
NUM	tecla <NumLock>	Sim	Não
CTRL	tecla <CTRL> esquerda	Sim	Não
CTRL_R	tecla <CTRL> direita	Sim	Não
ALT	tecla <ALT> esquerda	Sim	Não
ALT_R	tecla <ALT> direita	Sim	Não
SHIFT	tecla <SHIFT> esquerda (não há representação para tecla shift direita)	Sim	Não
ENTER	tecla <ENTER> central (nao na representação para <ENTER> do Pad Numérico)	Sim	Sim
BS	tecla <Backspace>	Sim	Sim
SPC	barra de espaço	Sim	Sim
WIN_L	tecla <Win> da esquerda	Sim	Não
WIN_R	tecla <Win> da direita	Sim	Não
WIN_P	tecla <Wmenu> (cx. propriedades)	Sim	Não
INS	tecla <INS>	Sim	Não
DEL	tecla 	Sim	Não
HOME	tecla <HOME>	Sim	Não
END	tecla <END>	Sim	Não
PGUP	tecla <PGUP>	Sim	Não
PGDN	tecla <PGDN>	Sim	Não
GRAVE	tecla do acento grave	Sim	Sim
APOST	tecla do apóstrofe	Sim	Sim
AGUDO	tecla do acento agudo (p/ uso com a diretiva &ABNT)	Sim	Não
CEDILHA	a tecla <Ç> para a diretiva &ABNT	Sim	Não
UP	tecla <SETA P/ CIMA>	Sim	Não

(continuação da tabela de palavras reservadas)

DOWN	tecla <SETA P/ BAIXO>	Sim	Não
LEFT <-	tecla <SETA P/ ESQUERDA> outra forma de representar	Sim	Não
RIGHT ->	tecla <SETA P/ DIREITA> outra forma de representar	Sim	Não
P0 à P9	teclas <0> à <9> do Pad numérico	Sim	Não
P/	tecla </> do Pad numérico	Sim	Não
P+	tecla <+> do Pad numérico	Sim	Não
P-	tecla <-> do Pad numérico	Sim	Não
P*	tecla <*> do Pad numérico	Sim	Não
PDEL	tecla do Pad numérico	Sim	Não
P.	tecla <.> do Pad numérico (&ABNT)	Sim	Não
00	criar uma tecla de duplo zero	Sim	Sim
000	criar uma tecla de triplo zero	Sim	Sim
NULL	criar uma tecla nula(não gera código)	Sim	Sim
PAGE	cria uma tecla de mudança de página	Sim	Sim

Definição de Strings:

A tecla definida como: (\$)teste

Quando pressionada enviará o texto “teste” para o PC.

Se for necessário que o texto seja finalizado com <ENTER>; a definição:

(\$)testeENTER

Não funcionará, será enviado o texto “testeENTER” para o PC.

Para se obter o efeito desejado, <ENTER> dentro de uma string tem que ser programada como ^m, então:

(\$)teste^m

Enviará ao PC o texto “teste” seguido de <ENTER>.

Da mesma maneira que ^m significa <ENTER>, outras teclas também podem ser codificadas de forma semelhante quando dentro de uma string, a saber:

^1 =<F1>, ^2 =<F2>, ^3 =<F3> ... ^9 =<F9>, ^0 =<F10>.

^H =<BS>, ^I =<TAB>, ^M =<ENTER>. (H,I ou M podem ser h,i ou m)

Portanto a definição:

(\$)^31^m

Enviará <F3><1><ENTER> para o PC.

Os códigos que podem ser enviados, ainda dependem da diretiva de interface utilizada.

Uma definição de string pode conter até 63 caracteres.

Definições em Hexa:

A tecla definida como:

(#)1C

Quando pressionada enviará o código 0x1C 0xF0 0x1C (MAKE E BREAK de 1C) ao computador, se a diretiva de interface for &ps2 o computador interpretará isso como sendo a tecla A. Veja apêndice 2 “Tabela de Scancodes inglês /Abnt2”.

A tecla definida como:

(#)41

Quando pressionada enviará o código 0x41 ao computador, se a diretiva de interface for &ascii será interpretado como 'A', Veja Apêndice 3 “Tabela ASCII internacional”.

Como visto acima, o prefixo # respeita o padrão da diretiva de interface selecionada (teclas com ou sem BREAK).

O prefixo b, e somente o prefixo b, pode ser usado em conjunto com o prefixo #, alterando o comportamento do BREAK.

Ex.: (#b)1C, Na interface PS2, enviará somente 0x1C, o código de BREAK (0xF0 0x1C) não. Códigos >= 0xE0 precisam ser precedidos de 00 (zero zero), por exemplo

(#)00 E5 Envia 0xE5 ao computador.

Todos os códigos precedidos por 00 (zero zero) independentemente da interface e do prefixo b, não enviam BREAK, ou seja, enviam sempre somente o código após o 00(zero zero).

O prefixo (#) aceita mais de um código (String Hexa),

Ex. (#)1B 3A 1C 42 5A

Para a diretiva &ps2 envia <S><M><A><K><ENTER> (teclas com MAKE e BREAK), no pressionamento da tecla, a soltura da tecla com string Hexa não produz código.

EX.(#b)00 E1 14 77 00 E1 00 F0 14 00 F0 77

Para a diretiva &ps2 envia <PAUSE>. Como o prefixo b foi usado, os códigos não geram BREAK. Veja apêndice 2 “Tabela de Scancodes inglês /Abnt2”, para o código da tecla <PAUSE>.

Códigos com valor acima de 0xE0, não precedidos de 00, serão reavaliados e se transformarão, na interface PS2, em:

0xE0 = Seta p/ cima	0xE1 = Seta p/ esquerda	0xE2 = Seta p/ baixo
0xE3 = Seta p/ direita	0xE4 = Ctrl direito	0xE5 = Alt direito
0xE6 = Win esquerda	0xE7 = Win direita	0xE8 = Win menu
0xE9 = Insert	0xEA = Delete	0xEB = Home
0xEC = End	0xED = Pg Up	0xEE = Pg Down
0xEF = / do pad		

Diretivas :

As diretivas permitem configurar certas características do teclado SKO44.

As diretivas &PS2 , &PS2_EMUL , &HID , &USB , &TCP_IP , &ASCII e &TCP_IP2 definem a interface e o Scancode a ser usado pelo teclado, sendo:

DIRETIVA	INTERFACE	SCAN-CODE
&PS2	PS2	PS2
&HID	USB-HID	PS2
&PS2_EMUL	RS232	PS2_EMUL
&USB	RS VIRTUAL	PS2_EMUL
&TCP_IP	ETHERNET	PS2_EMUL
&ASCII	RS232	ASCII
&ASCII	RS VIRTUAL	ASCII
&TCP_IP2	ETHERNET	ASCII

Isto quer dizer que o conjunto de diretivas disponíveis vai depender do tipo de teclado(interface) que se está utilizando. Por exemplo: as diretivas &ABNT e &INGLES só se destinam a teclados &PS/2.

Na interface PS/2, dependendo do layout usado ser &INGLES ou &ABNT, certas teclas geram Scancodes diferentes como no caso das definições abaixo:

No modo &INGLES:

- () ; : os códigos Make/Break configurados são: 4C / F0-4C
- : SKO44 em um PC configurado para Inglês a tecla produzirá ";"
- : SKO44 em um PC configurado para abnt2 a tecla produzirá "ç"

No modo &ABNT:

- () ; : os códigos Make/Break configurados são: 4A / F0-4A
- : SKO44 em um PC configurado para abnt2 a tecla produzirá ";"
- : SKO44 em um PC configurado para Inglês a tecla produzirá "/"

Nota: Veja no Apêndice A1(pg 28) as teclas afetadas pelo uso das diretivas &ABNT e &INGLES.

TABELA DE DIRETIVAS:

DESCRIÇÃO DAS DIRETIVAS				
DIRETIVAS DE TIPO DE INTERFACE: Essas devem ser as primeiras diretivas em um arquivo SMK, seu propósito é especificar o tipo de teclado a ser programado.				
&PS2	A diretiva &PS2 é o default, ou seja, no caso de ser um teclado PS2 será opcional o seu uso. No protocolo PS/2 as teclas são configuradas com os códigos de teclas Make/Break (ver Apêndice A2: Tabela de Scancodes).			
&HID	A diretiva &HID se destina a teclados conectados na interface USB-HID e o teclado é programado com scancode PS2			
&ASCII &TCP_IP2	A diretiva ASCII e TCP_IP2 destinam-se à teclados SERIAIS no protocolo VT100, atribuindo às teclas os códigos da tabela ASCII (ver Apêndice A3: Tabela ASCII).			
&PS2_EMUL &USB &TCP_IP	Essas diretivas programam o teclado de acordo com a tabela de SCAN CODE PS2_Emul. O comando Redirect da Sk_access.dll e o programa Redirect.exe tratam esses teclados que são seriais ou de rede de forma que possam ser lidos como um teclado PS2 normal			
A seguir apresentamos outras diretivas, dependendo da interface em uso algumas tornam-se desnecessárias, não sendo portanto suportadas.			Suportado pela diretiva	
DIRETIVA	DESCRIÇÃO	&PS2	&ASCII &TCP_IP2	&PS2_EMUL &HID &USB &TCP_IP
&ABNT	Esta diretiva faz com que os códigos de teclas (scancodes) sejam enviados de acordo com o que seria enviado por um teclado padrão Abnt2.	Sim	Não	Sim
&INGLES	Esta diretiva faz com que os códigos de teclas sejam enviados de acordo com o que seria enviado por um teclado padrão Inglês.	Sim	Não	Não
NOTA: O propósito das diretivas &ABNT e &INGLES é permitir que o usuário utilize o teclado programável de forma mais transparente possível utilizando o layout de teclado padrão de seu computador, seja o inglês ou o brasileiro. Quando usadas, devem ser posicionadas no arquivo SMK antes da primeira definição de teclas. Caso omitidas, será assumido por padrão o layout ABNT. Veja no Apêndice: Tabela de teclas afetadas pelo uso das diretivas ABNT e INGLES.				
&GREET String	"String" representa uma mensagem de saudação de até 80 caracteres a ser exibida no display ao ligar o teclado. O tamanho do display é 2x40 (duas linhas de 40 caracteres), mensagens com mais de 40 caracteres são quebradas em duas linhas.	Sim	Sim	Sim
&PAGE2	Indica o início de definição de uma segunda página do teclado contendo logo a seguir a configuração de outras 44 teclas. Geralmente, esta diretiva é usada em conjunto com a instrução ()PAGE, a qual define uma tecla para a mudança de página.	Sim	Sim	Sim

(continuação da tabela de diretivas)

&DELAY	Delay colocado entre dados provenientes de leitura de cartão magnético ou de teclas configuradas como String	Sim	Sim	Sim
&SCAN Scan_id	"Scan_id" especifica uma string de identificação de até 07 caracteres para o arquivo SMK. É exibido no display durante a inicialização do teclado, o que ocorre sempre que o teclado é ligado. Permite ao cliente saber o arquivo SMK utilizado.	Sim	Sim	Sim
&CAPS_ON &CAPS_OFF &NUM_ON &NUM_OFF &SCRL_ON &SCRL_OFF	Estado inicial do Teclado: Estas diretivas permitem configurar o estado inicial de CAPS_LOCK, NUM_LOCK e SCROLL_LOCK ao ligar o teclado.	Sim	Não	Sim

Diretivas específicas para leitores de cartão magnético:

Delimitadores são sequências de caracteres destinadas a separar campos de dados lidos das trilhas magnéticas.

Os delimitadores podem ser de início, fim ou de separação de campos de dados, sua configuração depende dos requisitos da aplicação cliente a qual se destina o SKO44.

Nota: As diretivas de configuração do leitor magnético são válidas apenas para a interface PS/2 e HID.

- Diretivas para delimitadores preestabelecidas:

Estas são usadas para estabelecer configurações padrões, aquelas mais comumente utilizadas. São diretivas simples, que trazem embutidos os delimitadores de início, fim e de separador, para cada trilha a que se destinam e determinam sempre uma codificação seguindo o padrão INGLÊS.

Tabela das diretivas: &M1D0, &M2D0, &M2D1 e &M2D2							
Diretivas p/ delimitadores padrões	Trilha Afetada	Delimitadores estabelecidos				em caso de erro de leitura envia:	
		Inicial		Separador	Final		
		Inglês	Abnt2			Inglês	Abnt2
&M1D0	trilha 1	%		não tem	? Enter	: Enter	Delimitador_inicial F Delimitador_final
&M2D0	trilha 2	^A ;	^A ç	=	? ^B Enter	: ^B Enter	
&M2D1	trilha 2	;	ç	=	? Enter	: Enter	
&M2D2	trilha 2	['	=]	[
NOTAS:							
1. ^ A significa o código correspondente a <CTRL>+<A> (idem para ^B)							
2. Independentemente de estar ou não utilizando a diretiva &ABNT2 no arquivo SMK as diretivas acima utilizam sempre scancodes do padrão INGLES.							
3. Os delimitadores configurados por &M1D0 e &M2D0 são os padrões de fábrica programados nos teclados SKO-44 com versões de firmware anteriores à v.1.94.							

Na tabela anterior, as diferenças entre as colunas Inglês e Abnt2 se referem apenas ao caractere exibido de acordo com o layout que estiver configurado no PC, e não sofrem efeito da utilização das diretivas &INGLES ou &ABNT, assumindo sempre o padrão Inglês.

- diretivas para especificar delimitadores customizados:

Se a aplicação do cliente exigir delimitadores específicos diferentes dos acima, será necessário utilizar as diretivas &MR, &M2M, &M1S, &M2S, &M1E e &M2E, as quais permitem flexibilidade total para a configuração de delimitadores, inclusive obedecendo as diretivas &INGLES ou &ABNT.

DIRETIVA	DESCRIÇÃO
&MR Cod_err	Estabelece uma string que é um código de erro a ser enviado quando houver erro de leitura em qualquer uma das trilhas magnéticas. Caso ocorra um erro de leitura, vai enviar o delimitador inicial configurado, a string "Cod_err" e o delimitador final. Cod_err pode ocupar até 04 caracteres minúsculos ou dígitos, porém ao usar símbolos que requeiram a geração de shift, o tamanho de cod_err fica reduzido pela metade.

(continuação da tabela da página anterior)

DIRETIVA	DESCRIÇÃO
&M2M Separ	" <i>Separ</i> " representa um caracter separador de campos para uso na trilha 2, Exemplo: &M2M + (configura o sinal de soma como separador). Obs: não há separadores para trilha 1.
&M1S Delim_inic &M2S Delim_inic	As diretivas &M1S e &M2S são semelhantes entre si, sendo que a primeira é para a trilha 1 enquanto que a outra é para a trilha 2. Permite configurar um delimitador inicial para a respectiva trilha. " <i>Delim_inic</i> " representa a combinação de um ou mais caracteres como um delimitador inicial. Pode ser usado letras, dígitos, alguns símbolos como : ; ! # \$ % & () + = [] ? e também qualquer caracter precedida por '^' significando <CTRL>+Caracter. OBS: Use ^M para configurar <ENTER> no delimitador
&M1E Delim_final &M2E Delim_final	As diretivas &M1E e &M2E são semelhantes entre si sendo que a primeira é para a trilha 1 enquanto que a outra é para a trilha 2. Permite ao usuário configurar um delimitador final para a respectiva trilha. " <i>Delim_final</i> " representa caracteres aceitos como delimitadores. Sua formação é semelhante à das diretivas &M1S e &M2S.

Exemplos de uso das diretivas &MR, &M1S , &M2S , &M1E , &M2E :

&MR er : Especifica "er" como código de erro de leitura.
 &M2M = : Estabelece "=" como separador da trilha2.
 &M1S (: Especifica "(" como delimitador inicial da trilha 1.
 &M1E)^M : Especifica ")ENTER" como delimitador final da trilha 1.
 &M2S [: Especifica "[" como delimitador inicial da trilha 2.
 &M2E] : Especifica "]" como delimitador final da trilha 2.

Nota: Ao configurar delimitadores customizados lembre-se que diferentemente dos anteriores, estes podem ser afetados pelas diretivas &ABNT ou &INGLES (para saber se os delimitadores configurados são ou não afetados, veja no apêndice a tabela de teclas afetadas pelo uso das diretivas &ABNT e &INGLES).

{Definição da primeira linha de teclas do SK044}

{Definimos as 11 teclas da linha começando da esquerda para a direita}

()a

()b

()c

()d

()e

()f

()g

()h

()7

()8

()9

{Definição da segunda linha}

()i

()j

()k

()l

()m

()n

()o

()p

()4

()5

()6

{Terceira linha}

()bs

()q

()r

()s

()t

()u

()v

()w

()1

()2

()3

{Quarta e última linha de teclas}

()pgup

()enter

()x

```
()y
()z
()spc
().
();
()0
()f1
()f2
```

```
{Indica final de definição de teclas}
```

```
[FIM]
```

```
{Note que definimos uma única página de 44 teclas}
```

Exemplo de um arquivo SMK com duas páginas de código de 44 teclas :

```
-----
{ Configuracao de um teclado PS/2 }
{ com duas páginas de teclas      }
[sko44]
&ingles

        {
            Régua para o posicionamento da mensagem no Display
        {.....|.....|.....|^.....|.....|.....|.....}
&greet          ** SMAK TECLADOS **                ** SMAK TECLADOS **

{delimitadores customizados para o leitor trilha 2}
{ATENCAO: DIRETIVAS DE LEITORES ACEITAS APENAS A PARTIR DO FIRMWARE 1.94 DO SK0-44}
{delimitador de inicio "*" }
&m2s *

{delimitador final "<ENTER>" }
&m2e *^M

{Um código para de erro de leitura "er"}
&mr er

{delimitador separador "+" }
&m2m +

{Um delay de 3ms para intervalo entre caracteres quando enviados pelo SK044}
&delay 3
```

{inicio da primeira pagina do teclado}

()a
()
()b
()
()c
()
()d
()
()e
()
()f
()
()g
()
()h
()
()7
()
()8
()
()9

{segunda linha}

()i
()
()j
()
()k
()
()l
()
()m
()
()n
()
()o
()
()p
()
()4
()
()5
()
()6

{Terceira linha}

()bs
()
()q
()
()r
()
()s
()
()t
()
()u
()
()v
()
()w
()
()1
()
()2
()
()3

{Quarta linha}

()p9
()
()enter
()
()x
()
()y

()z
()spc
()0
()00
()000
()f1
()page

{Diretiva indica o inicio da configuração de uma segunda página}
&page2

{primeira linha da segunda página}

()1
()2
()3
()4
()5
()6
()7
()8
()9
()0
()bs

{segunda linha}

()a
()b
()c
()d
()e
()f
()g
()h
()i
()j
()k

{terceira linha}

()l
()m
()n
()o
()p
()q
()r

```
()s  
()t  
()u  
()v
```

```
{quarta linha}
```

```
()w  
()x  
()y  
()z  
(s)=  
()-  
()null  
()spc  
()null  
()enter  
()page
```

```
[fim]
```

Exemplo configuração para um teclado SERIAL tipo VT100 (ASCII):

```
{ SMK PARA UM TECLADO SERIAL }
```

```
[sko44]
```

```
&ascii
```

```
{ A diretiva &ascii configura um teclado serial tipo VT100 }  
{ Atenção: somente teclados seriais com firmware superior a v 1.91 }  
{ podem ser programados via UploadWin }  
{ Para a diretiva ASCII, as definições de teclas são case-sensitive }
```

```
{ Letras maiúsculas }
```

```
()A  
()B  
()C  
()D  
()E  
()F  
()G  
()H  
()7  
()8  
()9
```

{segunda linha}

()I
()J
()K
()L
()M
()N
()O
()P
()4
()5
()6

{Terceira linha}

()Q
()R
()S
()T
()U
()V
()W
()X
()1
()2
()3

NOTA:

O teste de funcionamento de um teclado serial depende do software cliente, o qual deve ser capaz de receber os dados da porta serial. Entretanto, é possível alguns testes rápidos usando o programa exemplo de uso e testes da DLL (Delphi ou VB). Este encontra-se disponível para download no site da Smak. Ao executar o utilitário veja os botões "Get_Rs" e "Redirect":

{Quarta linha}

()Y
()Z
()+
()-
().
()0
()00
(\$)Uma_String
()bs
()NULL
()ENTER

Get_Rs: Esta função lê os dados recebidos e buferizados de teclados seriais programados com a diretiva &ASCII. No Aplicativo de Exemplo (Delphi ou VB) captura as últimas 03 teclas apertadas no teclado serial e envia para a caixa de textos superior do próprio utilitário de testes.

Redirect: Esta função redireciona os dados recebidos na porta serial para o buffer de teclado, qualquer aplicativo, como por exemplo o bloco de Notas ou mesmo um prompt do DOS, vai processar os dados como se fossem de um teclado PS2. Destina-se a teclados seriais programados para simular scan-codes (diretiva "&PS2_EMUL"), mas funciona parcialmente com teclados seriais programados com a diretiva &ASCII.

[fim]

APÊNDICE

A1 - Tabelas de definições de teclas afetadas pelas diretivas &Abnt e &Inglês:

Devido às diferenças entre os mapas de teclados INGLES e ABNT, dependendo da diretiva especificada no arquivo SMK certas representações de teclas terão interpretação diferente, abaixo as definições de teclas afetadas por estas diretivas.

Definição de tecla	Comportamento conforme o uso das diretivas &ABNT ou &INGLES
() [()] () ; () ~ () /	Geram scancodes diferentes conforme a diretiva utilizada.
() GRAVE () APOST	Palavras reservadas que geram scancodes diferentes conforme a diretiva utilizada
() AGUDO () CEDILHA () P .	Palavras reservadas para uso exclusivo com a diretiva &ABNT (não funcionam com a diretiva &INGLES)

A2 - Tabelas de Scancodes Inglês / Abnt2:

SCAN CODE PS2 ABNT/Inglês																
			Após E0													
	00	10	10	20	20	30	30	40	40	50	50	60	60	70	70	80
00														o Ins	Ins	
01	F9	Alt_L	Alt_R	C		N		, <		/ ?		\		Del ,	Del	
02		Shift_L		X		B		K		~ ^	' "			2 ↓	↓	
03	F5			D		H		I						5		F7
04	F3	Ctrl_L	Ctrl_R	E		G		O		' `	[{			6 →	→	Prtsc²
05	F1	Q		4 \$		Y		o)		= +				8 ↑	↑	
06	F2	1 !		3 #		6 ^		9 (BS		Esc		
07	F12				Win_R		Power							Num Lock		
08										Caps				F11		
09	F10			Spc				. >		Shift_R		1 End	End	+		
0A	F8	Z		V		M		; : / ?	/	Enter	Enter_P			3 PgDn	PgDn	
0B	F6	S		F		J		L		[{] }	4 ←	←	-		
0C	F4	A		T		U		Ç ; :				7 Home	Home	*	Prtsc¹	
0D	Tab	W		R		7 &		P] }	\	.		9 PgUp	PgUp	
0E	' "	~	2 @		5 %		8 *	- _			Wake			Scroll	Pause¹	
0F			Win_L		Win_P		Sleep									

Prtsc¹ = Prtsc = Shift + Prtsc = Ctrl + PrtscPrtsc² = Alt + PrtscPrtsc³ = E0 - 12 - E0 - 7C / E0 - F0 - 7C - E0 - F0 - 12Pause¹ = Ctrl + PausePause² = E1 - 14 - 77 - E1 - F0 - 14 - F0 - 77

= Scan Code Inglês

= Código depende de shift e num-lock

Se Num-lock = on e Shift=off então Make = E0 - 12 - Make da tabela

Break = Break da tabela - E0 - F0 - 12

Código válido para
Shift = off e Num-lock=off
ou
Shift = on e Num-lock=on

A3 - Tabela ASCII internacional: *Tabela ASCII (usada com as diretivas &ASCII e &TCP_IP2)*

SCAN CODE ASCII								
MSB LSB	00	10	20	30	40	50	60	70
00	NUL	DLE	SPACE	0	@	P	`	p
01	SOH	DC1	!	1	A	Q	a	q
02	STX	DC2	"	2	B	R	b	r
03	ETX	DC3	#	3	C	S	c	s
04	EOT	DC4	\$	4	D	T	d	t
05	ENQ	NAK	%	5	E	U	e	u
06	ACK	SYN	&	6	F	V	f	v
07	BEL	ETB	'	7	G	W	g	w
08	BS	CAN	(8	H	X	h	x
09	TAB	EM)	9	I	Y	i	y
0A	LF	SUB	*	:	J	Z	j	z
0B	VT	ESC	+	;	K	[k	{
0C	FF	FS	,	<	L	\	l	
0D	CR	GS	-	=	M]	m	}
0E	SO	RS	.	>	N	^	n	~
0F	SI	US	/	?	O	_	o	DEL

A4 - Tabela Scancodes PS2emul: Tabela PS2emul (usada na configuração com as diretivas &PS2emul, &USB, &TCP_IP)

SCAN CODE PS2_Emul									
MSB LSB	00	10	20	30	40	50	60	70 E0	F0
00	NULL	ALT_L	SPC	o)	P-	P	AGUDO GRAVE ` ~	ALT_R	PAGE
01	CAPS_ON	SHIFT_R	P0	1 !	A	Q	F1	INS	DUPLO o
02	CAPS_OFF	SHIFT_L	P1	2 @	B	R	F2	END	TRIPLO o
03	NUM_ON		P2	3 #	C	S	F3	↓	
04	NUM_OFF		P3	4 \$	D	T	F4	PG_DN	
05	SCRL_ON	NACK	P4	5 %	E	U	F5	←	
06	ACK	CTRL_L	P5	6 "	F	V	F6	CTRL_R	
07	SCRL_OFF		P6	7 &	G	W	F7	→	
08	BS		P7	8 *	H	X	F8	HOME	
09	TAB		P8	9 (I	Y	F9	↑	
0A		CEDILHA Ç	P9		J	Z	F10	PG_UP	
0B		ESC	APOST ' "	; :	K	[{	F11	P/	
0C	~ ^		, <	P*	L	\	F12	PRT_SCR	
0D	ENTER	P. (ABNT)	- _	= +	M] }	WIN_L	P_ENTER	
0E	CAPS	SCRL	. >	P+	N	SET_AUX	WIN_R	PAUSE	
0F		NUM	/ ?	P DEL	O	SET_MAIN	WIN_P	DEL	

A5 - Tabela de conversão AT - XT

Tabela de conversão AT → XT																
	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	FF	43	41	3F	3D	3B	3C	58	64	44	42	40	3E	0F	29	
10	65	38	2A	52	1D	10	02	1C	0E	53	2C	1F	1E	11	03	5B
20	67	2E	2D	20	12	05	04	5C	68	39	2F	21	14	13	06	5D
30	4F	31	30	23	22	15	07	5E	6A	50	32	24	16	08	09	5F
40	40	33	25	17	18	0B	0A	60	47	34	35	26	27	19	0C	56
50	7E	73	28	4D	1A	0D	62	6E	3A	36	1C	1B	48	2B	63	01
60	0D	56	45	57	4E	51	0E	4A	37	4F	49	4B	47	7E	7F	6F
70	52	53	50	4C	4D	48	01	45	57	4E	51	4A	37	49	46	54
80				41	54											
90																
A0											AA	AB				
B0																
C0																
D0																
E0	E0	E1														

