8 TECHNICAL INFORMATION

This chapter contains important technical information for programmers. It describes the memory layout of BASIC on the P2000C and gives notes on the system's treatment of ASCII codes.

8.1 MEMORY LAYOUT

8.1.1 Terminology

The memory layout is described here by hexadecimal addresses and their contents. The following terms are explained first:

Term	Meaning
Jump	A three-byte sequence which directs control to a routine. The first byte (=C3H) is the jump command and the second two bytes contain the hexadecimal address of the routine.
Pointer	A two-byte hexadecimal address.
Flag	A single byte which sets a system mode off or on.

Technical Information

High Memory: The last accessible address for BASIC. This can also be set by the CLEAR command. The default value is (start of CP/M)-1 (see below). Machine code subroutines should be situated above the high memory address (e.g., Screen Handler and KSAM80).

(KKKKH) Beginning of KSAM80 (if loaded)
For 61 K Conf. ..CC00
For 62 K Conf. ..D000
For 63 K Conf. ..D400

(SSSSH) Beginning of Screen Handler (if loaded)
For 61 K Conf. ..D800
For 62 K Conf. ..DC00
For 63 K Conf. ..E000

Beginning of BDOS
For 61 K Conf. ..E006
For 62 K Conf. ..E406
For 63 K Conf. ..E806

Beginning of CBIOS
For 61 K Conf. ..EE00
For 62 K Conf. ..F200
For 63 K Conf. ..F600

```
PRINT ESC$;"Y";CHR$(10+&H20);CHR$(20+&H20);"HELLO"
```

- This ESCAPE sequence will result in the word 'HELLO' being printed at line 11, column 21. It is called a Cursor Control Function. Note that it can be simplified in your program by replacing it with the following sequence of commands:

```
10 DEF FNX$(X,Y)=CHR$(&H1B)+"Y"+CHR$(X+&H20)+CHR$(Y+&H20)
```

- Later on in the program, the cursor can be controlled by using the following statement:

```
20 PRINT FNX$(10,20);"HELLO"
```

8.3 THE KEYBOARD

The codes generated by the keys on the P2000C keyboard may be set up and saved on disk by the 'Keyboard Table Editing Option' of the configuration program. This is fully documented in the CP/M Reference Guide, but it is also a self-explanatory program, and you should readily be able to set the codes to your requirements, using the translation tables contained in this manual, (this chapter pages 8-10).

8.3.1 Keyboard Control Sequences

The key:

combined with different keys generates a control sequence; the different sequences are shown overleaf.

Technical Information

POINTS TO NOTE ABOUT ASCII CODES

The PRINT USING [#filenumber] command uses special format statements (refer to the BASIC Reference Manual, Section 3.78). The ASCII backslash (\) is used to format strings. This character does not exist in all national versions because of the national ISO code restrictions (only 14 characters). The backslash must therefore be represented by the same character at the same code position in your national version.

Some national keyboard versions do not have the numeric symbol '#'. The character '£ '(pound sign) must be used instead. For example:

```
A#=1.456778
```

must be represented as follows:

```
A£ =1.456778
```

If you are writing a program for international application, be aware of the characters that will change from version to version.

Technical Information

HEX:	23	24	27	40	5B	5C	5D	5E	5F	60	7B	7C	7D	7E
ASCII/INT.	#	$	'	@	[	\	]	^	_	`	{	|	}	~
UK (special)	£	$	'	@	#	\	|	^	_	`	{	|	}	~
D/A	#	$	'	§	Ä	Ö	Ü	^	_	`	ä	ö	ü	ß
F/B	£	$	'	à	°	ç	§	^	_	`	é	ù	è	¨
I	£	$	'	§	°	ç	é	^	_	ù	à	ò	è	ì
E	#	$	'	@	[	Ñ	]	^	_	`	{	ñ	}	¨
S/SF	#	¤	'	É	Ä	Ö	Å	^	_	é	ä	ö	å	ü
DK/N	#	$	'	@	Æ	Ø	Å	^	_	`	æ	ø	å	~
P	£	$	'	@	Ã	Ç	Õ	^	_	`	ã	ç	õ	~
CH	£	$	'	ç	à	é	è	^	_	`	ä	ö	ü	¨
UK/NL	£	$	'	@	[	\	]	^	_	`	{	|	}	~

Fig. 8.2 National Code Table

Access to the Video Terminal

9 ACCESS TO THE VIDEO TERMINAL

9.1 SCREEN SPECIFICATIONS

8-bit code (national versions) 24 lines/80 characters

Character Mode

- Bi-directional scrolling
- Split-screen capability (partial scroll)
- Three Attribute modes:
  Manual — Read and write data from/to attribute page (normal memory access).
  Auto duplicate: The read attribute data will be duplicated.
  Block mode: For "block moves" such as scrolling, the attribute page will be scrolled automatically.
- Attributes: Underline, Invert, Blink, 4 Intensity levels) and all combinations
- Adjustable TABs
- Teletext graphics
- Text and Attributes back-transfer from screen possible.

High Resolution Graphics Mode

- 2 selectable modes
  512 x 252 dots (no attributes)
  256 x 252 dots (3 intensity levels + background)

Each dot addressable
Simple vector handling:
- In Cartesian and Polar co-ordinates.
- Combination with character mode
  (characters: 21 lines/64 characters).

9.2.2 Escape Sequences - Set Attribute

SET ATTRIBUTE = ESC,0,b
where b=attribute byte
and 0=numeric zero

An attribute can be set at any time and is valid until a new attribute is selected.

Attributes: Underline - UL
Blink - BL
Invert - INV
4 Intensities
res - reserved

Attribute byte:

	res	in1	UL	INV	res	res	BL	in2
bit:	7	6	5	4	3	2	1	0

Intensities	in1	in2
Quarter bright	0	0
Bold	0	1
Normal	1	0
Half bright	1	1

For example, the ESCape sequence ESC,0,99 (or ESC,0,&H63) would cause screen characters to be produced:

- half brightness
- underlined
- blinking

Note: To execute ESCape sequences it is necessary to use the CHR$(27) code.
The above example could be included in a BASIC program in the following way:

```
PRINT CHR$(27)+"0"+CHR$(&H63)
        or
PRINT CHR$(27)+"0"+CHR$(99)
```

Access to the Video Terminal

Send Text from Cursor Position	ESC,$,nn nn=number of characters	$=24H/36
Send Attributes of Text from Cursor	ESC,%,nn	%=25H/37
Load User Program	ESC,p (in INTEL HEX format)	p=70H/112
End of INTEL HEX Format (exits loader, normal operation)	ESC,:	:=3AH/58
Execute User Program	ESC,x	x=78H/120
Load New Keyboard Table	ESC,@	@=40H/64
Load New Screen Table	ESC,!	!=21H/33
Define Caps Lock Key (k=key,nn=upper limit (Hex))	ESC,+,k,nn	+=2BH/43

9.2.4 Description of Control Codes

Cursor home:	New cursor position is column 1/ row 1
Cursor forward:	Column + 1.
Cursor down:	New line. Scroll if last line or beginning of a locked area.
Cursor up:	One line up. If 1st line then new position is the bottom line.
Cursor back:	Column - 1.
Backspace:	Same as cursor back.
TAB:	Cursor to next TAB position, default every eighth column.
Clear screen:	Erase the whole screen, cursor at home position.
CR:	Cursor at column 1 in current line.
End of page:	New cursor position is column 80 and row 24
Reset terminal:	Initialise hardware and software. After RESET allow 500ms before sending data to terminal.
Lock keyboard:	All keyboard inputs are ignored.
Unlock keyboard:	Keyboard entries are re-enabled.
Reset:	Re-initializes the hardware and software.
Set attribute:	A new attribute is used until the next "Set attribute" command.
Cursor address:	Absolute cursor address with an offset of 20H (ESC,Y,20H,20H is the "home position").

Access to the Video Terminal

Back TAB:	New cursor position is previous TAB position.
Cursor visible:	Display cursor.
Cursor invisible:	Do not display cursor.
Start Teletext graphic:	All characters between 20H & 3FH, and 60H & 7FH are interpreted as teletext characters.
End Teletext graphic:	Normal character mode.
Lock area:	From cursor line, n lines will be locked. (This area will not be scrolled by Cursor up and down, only by explicit "Scroll up (down)" command.
Unlock area:	The cursor position area is unlocked.
Unlock all:	Normal screen status.
Send status:	12 bytes terminal status information will be sent (see STATUS INFORMATION).
Send text:	nn characters from cursor position will be sent back.
Send attribute:	nn attribute bytes from cursor position will be sent.
Load user PGM:	Starts INTEL HEX FORMAT loader for down-loading a machine code program in INTEL HEX format.
End loader:	Exits the loader, enables normal operation.
Execute PGM:	Calls a previously loaded program.
Load new keyboard table:	A new keyboard table (national version) will be downloaded. It consists of 4 sub-tables (NORMAL, SHIFT, SUPER SHIFT, SUPER SHIFT-SHIFT) (see STANDARD KEY TABLE).
Load new screen table:	A new screen translation table will be loaded.
Define Caps Lock Key:	The key K is the new caps lock key. There is no default key.

Access to the Video Terminal

9.2.6 Description of Graphic Commands

Start mode 1:	Start 256 x 252 resolution mode with 3 intensity levels. The change of this level is done by "Set attribute" command.
Start mode 2:	Start the 512 x 252 resolution mode.
Start character mode:	Exit high resolution graphic mode 1 or 2.

After a start graphic mode command the text on the screen is not cleared but will be re-arranged as 21 lines of 64 characters. The graphic screen will be cleared and the internal cursor and origin set to zero. The start character mode command (exit graphics mode) will clear the text buffer and graphics screen and set the internal cursor and origin field to zero.

CARTESIAN COORDINATES

Clear dot:	Erase pixel at screen position xy.
Set dot:	Set a pixel at xy.
Move to:	Set the internal cursor field to xy.
Draw to:	Draw a line from internal cursor field to xy, and set the internal cursor field to xy.
Clear to:	Same as "Draw to" but the line is erased.

POLAR COORDINATES

Set origin:	Set the internal origin field to xy.
Clear dot:	Erase a pixel at A (angle), abs (absolute value) according to the origin.
Set dot:	Same, but set the pixel.
Move to:	Set the internal cursor field to the calculated (using A abs, origin) value xy.
Draw to:	Draw a line to the calculated value xy and set internal cursor field to xy.

Note: Any combination of POLAR and CARTESIAN coordinates is possible!

8.1.2 Layout

Address	Contents
0000H	Jump to warm boot (Note that the command sequence 'M = 0: CALL M' has the same effect as the command 'SYSTEM')
0005H	Jump to CP/M (Refer to the CP/M Manuals for more information on CP/M)
0100H	Begin of TPA (Transient Program Area) Start of BASIC Interpreter The TPA is used by the BASIC Interpreter in the following way (starting at 100H): Administration Area: internal BASIC flags and pointers are set in this area. BASIC Interpreter: The administration area and interpreter require approximately 24K memory space. File Buffer Area: The size of this area depends on how many file buffers were allocated on initialization. You can evaluate the size of this area with the VALPTR(#1) command. BASIC Program Area: A BASIC program is written into this area. Variables: Numeric variables are allocated in this area. String pointers are also set here (for more information, see the BASIC Reference manual). Array Variables: Arrays are written into this area. String Values: BASIC dynamically allocates strings in this area. To reorganise this memory space, use the command FRE(""). Stack: The BASIC stack is used for GOSUB, FOR..NEXT and expression evaluation. The stack size can be set by the CLEAR command. The default size is 512 bytes, which is sufficient for normal application programs.

8.2 Access to the Video Terminal in BASIC

The best demonstration of how to use the Video Terminal is to execute and analyse the Screen Handler Activator demonstration program (see Chapter 6). Information is sent to the screen using the BASIC 'PRINT [USING]' command (see the BASIC Reference Manual, Section 3.78). All commands are sent to the screen as ESCAPE sequences. It is therefore recommended that you use the following constant in all of your programs:

```
ESC$=CHR$(&H1B)
```

You can then use this constant to control screen output. The following are examples of ESCAPE sequences to control screen output:

```
PRINT ESC$;CHR(48);"P"      - This will cause every character
                              after the constant to be printed
                              in inverse video.

PRINT ESC$;CHR$(48);        - This will cause every character
CHR$(&H40)                    after the constant to be printed
                              normally.

PRINT ESC$;"c"              - This will make the cursor invisible
```

CTRL A	Enters BASIC EDIT mode for the current printed line.
CTRL C	Terminates program execution.
CTRL R	Redisplays the current printed line.
CTRL S	Suspends program execution until any other key is pressed.
CTRL X	Deletes the current printed line.

8.3.2 ASCII Codes

Each ASCII character is represented by one of the 7 bit codes shown in Table 1 (there are 128 possible 7 bit codes). Fourteen of these positions are know as 'national ISO code' positions. In each national version, these positions represent different characters. The national codes are shown in Table 2.

Internally, the P2000C uses an 8 bit code (=256 positions, see Table 3).

Technical Information

	0	1	2	3	4	5	6	7
0	NUL	DLE	SP	0		P		p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3		3	C	S	c	s
4	EOT	DC4		4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB		7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K		k	
C	FF	FS	,	<	L		l	
D	CR	GS	-	=	M		m	
E	SO	RS	.	>	N		n	
F	SI	US	/	?	O		o	↵

Fig. 8.1 7 Bit ASCII Code Table

Empty spaces represent national ISO code positions.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	DLE	SP	0	@	P	`	p	▗	▄	◇	°	≠	–	Ω	ë
1	SOH	DC1	!	1	A	Q	a	q	▖	▙	¡	±	Á	á	Æ	æ
2	STX	DC2	"	2	B	R	b	r	▘	▛	¢	²	´	î	Å	å
3	ETX	DC3	#	3	C	S	c	s	▀	▀	£	³	À	à	a	û
4	EOT	DC4	$	4	D	T	d	t	▚	▙	Õ	x	Â	â	Ô	ô
5	ENQ	NAK	%	5	E	U	e	u	▌	▙	¥	µ	Ä	ä	Ö	ö
6	ACK	SYN	&	6	F	V	f	v	▞	▟	¨	¶	É	é	Ó	ó
7	BEL	ETB	'	7	G	W	g	w	▛	▟	§	õ	È	è	Ò	ò
8	BS	CAN	(	8	H	X	h	x	▜	▟	¤	÷	Ê	ê	Ú	ú
9	HT	EM	)	9	I	Y	i	y	▜	▟	⌜	⌝	⊢	⊣	ø	ø
A	LF	SUB	*	:	J	Z	j	z	▐	▟	⌞	⌟	⊤	⊥	+	ı
B	VT	ESC	+	;	K	[	k	{	▛	▟	«	»	Í	í	Ω	ß
C	FF	FS	,	<	L	\	l	|	▜	■	←	¼	Ì	ì	Ù	ù
D	CR	GS	-	=	M	]	m	}	▚	▙	↑	½	Ï	ï	Ü	ü
E	SO	RS	.	>	N	^	n	~	▟	▟	→	¾	Ç	ç	Ã	ã
F	SI	US	/	?	O	_	o	↵	▛	■	↓	¿	Ñ	ñ	-	·

Fig. 8.3 Standard P2000 8 Bit Code Table

Access to the Video Terminal

9.2 SCREEN AND SCREEN CODES

The screen is controlled by both single codes and ESCape sequences.

9.2.1 Single Codes

A C T I O N	Code	
Cursor Home	SOH	01H
Cursor Forward	ACK	06H
Cursor Down	LF	0AH
Cursor Up	SUB	1AH
Cursor Back	NAK	15H
Bell-Beep	BEL	07H
Backspace (same as Cursor Back)	BS	08H
TAB	TAB	09H
Clear Screen	FF	0CH
CR	CR	0DH
End of Page (column 80, row 24)	EOT	04H
Reset Terminal	CAN	18H
CAPS LOCK	SI	0FH
Lock Keyboard	EM	19H
Unlock Keyboard	STX	02H

9.2.3 Escape Sequences - Screen Control

A C T I O N		Code	ASCII
Cursor Addressing		ESC,Y,r,c r=row max = 24 c=column max = 80	Y=59H/89
Erase to End of Line		ESC,K	K=4BH/75
Erase to End of Screen		ESC,k	k=6BH/107
Scroll Up one Line		ESC,S	S=53H/83
Scroll Down one Line		ESC,T	T=54H/84
Set TAB at Cursor Position		ESC,I	I=49H/73
Clear TAB at Cursor Position		ESC,G	G=47H/71
Clear all TABs		ESC,g	g=67H/103
Insert Line		ESC,L	L=4CH/76
Delete Line		ESC,l	l=6CH/108
Insert Character at Cursor Position	ON	ESC,Q	Q=51H/81
	OFF	ESC,R	R=52H/82
Delete Character at Cursor Position		ESC,P	P=50H/80
Insert Character Wrap-around	ON	ESC,N	N=4EH/78
	OFF	ESC,R	R=52H/82
Delete Character Wrap-around		ESC,O	O=4FH/79
Back TAB		ESC,i	i=69H/105
Cursor Visible		ESC,C	C=43H/67
Cursor Invisible		ESC,c	
Start Teletext Graphic		ESC,1	1=31H/49
End Teletext Graphic		ESC,2	2=32H/50
Lock Area for Scrolling		ESC,A,n n=number of lines	A=41H/65
Unlock Area from cursor		ESC,a	a=61H/97
Unlock all Areas		ESC,u	u=75H/117
Send Status		ESC,?	?=3FH/63

Access to the Video Terminal

Erase to end of line:	Clear all characters including cursor position to column 80.
Erase to end of screen:	Same as "Erase to end of line" to column 80, row 24.
Scroll up 1 line:	Scroll up the whole screen or area and clear last line.
Scroll down 1 line	Same as above, but scroll down.
Set TAB:	Sets a new TAB position at cursor position.
Clear TAB:	A TAB position is removed at cursor position.
Clear all TABs:	Removes all TABs.
Insert line:	Scroll down the lines from cursor line + 1, and clear cursor line.
Delete line:	Scroll up the lines from cursor line +1 to cursor line.
Insert ON:	The next character will be inserted at cursor position, the last character in the line will be lost.
Insert wrap-around:	Same as "Insert" but the last character of the screen is lost
Insert OFF:	Exit insert mode, normal overwrite.
Delete character:	Delete character at cursor position, the last character in the line will be blank.
Delete character wraparound:	Same as "Delete character" but last character of screen will be blank.

9.2.5 Graphic Control

ACTION	Code	ASCII
Start high resolution mode 1 (256 x 252)	ESC,5	5=35H/53
Start high resolution mode 2 (512 x 252)	ESC,3	3=33H/51
Start character mode (end graphics)	ESC,4	4=34H/52
CARTESIAN CO-ORDINATES:		
Clear dot	ESC,d,xy	d=64H/100
Set dot	ESC,D,xy	D=44H/68
Move to	ESC,m,xy	m=6DH/109
Draw to	ESC,M,xy	M=4DH/77
Clear to	ESC,v,xy	v=76H/118
POLAR CO-ORDINATES:		
Set origin	ESC,z,xy	z=7AH/122
Move to	ESC,y,Aabs	y=79H/121
Draw to	ESC,U,Aabs	U=55H/85
Clear to	ESC,w,Aabs	w=77H/119
Set dot	ESC,F,Aabs	F=46H/70
Clear dot	ESC,f,Aabs	f=66H/102
Receive picture	ESC,r,xy,nn nn=number of bytes	r=72H/114
Send picture	ESC,t,xy,nn	t=74H/116

Note: xy = co-ordinates: 2 bytes in mode 1
3 bytes in mode 2
(low 'x' byte first, i.e., x(low), x(high), y

Aabs = A = angle ALPHA (2 bytes)
abs = absolute value (2 bytes)

For sending and receiving picture function, a byte represents the contents of the video RAM (like a dump).

Co-ordinates: x is the horizontal co-ordinate, y the vertical co-ordinate x=0, y=0 is bottom leftmost dot on the screen.

Angle ALPHA in steps of one degree (0 to 360)

9.3 STATUS INFORMATION

When the 'send status' code sequence ESC, ? (1BH,3FH) is sent to the terminal, the terminal status information is returned in the form of a 12 byte string, as shown below:

Byte Number		Contents
1		cursor position - column
2		cursor position - row
3		character at cursor position
4		status flag
	Bit:	
	0	1 = graphics on
	1	1 = graphic mode 2
	2	1 = teletext on
	3	1 = insert mode on
	4	1 = insert wraparound on
	5	1 = keyboard is locked
	6	reserved
	7	reserved
		internal graphic cursor field
5,6		x co-ordinate
7		y co-ordinate
8,9		free space pointer (beginning of RAM area for user program)
10,11,12		reserved for future use

The twelve bytes must be read via a normal 'keyboard read' instruction.

System Errors

A SYSTEM ERRORS

There are a number of errors that may occur when you are using the BASIC Interpreter Development System. They are described in the following sections of this appendix.

A.1 Disk Errors

The format of a disk error message is as follows:

BDOS ERR ON drive name : error

where

drive name — is the name of the drive on which the error occurred.

error — is one of the following error messages:

BAD SECTOR — This indicates* that the disk may be physically damaged, that the disk controller may be defective, or that the disk has not been correctly inserted. Open the drive flap and check to see if the disk is positioned correctly, and that the disk is of the correct format. This is a fatal error and the data that has been changed before the occurrence of the error and after the last back-up was made, will probably have been lost. To escape from the error, type a carriage return to ignore the sector (this will probably harm your data), or press the reset button to reboot the system (this will erase the contents of the memory).

SELECT — This error will occur if you address an invalid drive name. Press the reset button and the system will reboot. The contents of the memory will be erased.

System Errors

READ ONLY — This error will occur if you did not program a RESET (see BASIC Reference Manual command section) after inserting a new disk in one of your drive units, or if a write protect tab is on the disk. Type CTRL/C and the system will reboot. The contents of the memory will be erased.

Note - Because of the possible occurrence of the above errors, take care of your disks, follow the instructions on the back of the disk cover. Use the recommended back-up procedures (see Appendix D) as often as possible.

A.2 Printer Error

If you try to use the printer when it is not ready for use, the following error message will occur:

IGNORE PRINTER WITH ESC

The printer is not ready for use when it is not turned on, when there is no paper, or when the ribbon is out. If, having prepared the printer for use, you type any key but (ESC), the system will re-execute the specified procedure. If you do not have a printer, press the (ESC). From then on, the printer will be ignored every time you wish to print, and to enable printing, you must reset the system.

A.3 BASIC Errors

Refer to the BASIC Reference Manual for a list of errors which may occur in the BASIC Environment.

Files on the System Disk

B FILES ON THE PRODUCT DISK

The following is a list of files present on the product disk:

MBASIC	.COM	- Microsoft BASIC Interpreter Rev.5.22
CONFIG	.BAS	- Configuration program
VOLORG	.BAS	- Disk utility
VA	.COM	- Machine code routines for VOLORG
SH	.L61	- Screen Handler machine code routine for 61K configuration
SH	.L62	- Screen Handler machine code routine for 62K configuration
SH	.L63	- Screen Handler machine code routine for 63K configuration
SHACT	.BAS	- Screen Handler Activator demonstration program
SHACT	.MES	- Message file for demonstration program
KSM80	.L61	- KSAM 80 for 61K configuration
KSM80	.L62	- KSAM 80 for 62K configuration
KSM80	.L63	- KSAM 80 for 63K configuration
LKSAMUT	.BAS	- Loader for KSAM 80 Utility
KSAMUT	.COM	- KSAM 80 Utility program
KSAM-0	.BAS	- Program to evaluate KSAM80 load address
KSAM-1	.BAS	
.		
KSAM-14	.BAS	- KSAM80 demonstration programs (see chapter 7)

Example Error Message Definition

C EXAMPLE ERROR MESSAGE DEFINITION

The following is an example error message definition

```
10 CLEAR,&HCAFF,512        ' Beware not to erase other program
   parts !
20 PRINT CHR$(12)
30 ' The actual message will be displayed on line 23
   (ESC Y 22H 00H) and
32 ' inverse (ESC 0 P). A beep will also be activated (07).
40 A$="PRINTER ERROR.  CR  TO RETRY,  SPACE  TO IGNORE:"
50 A$=CHR$(7)+CHR$(&H1B)+"Y"+CHR$(&H20+22)+CHR$(&H20)
   +CHR$(&H1B)+"0P"+A$+CHR$(&H1B)+"0"+CHR$(&H40)
55 'Set the video write parameters
60 A$=CHR$(5)+CHR$(&H31)+CHR$(0)+CHR$(LEN(A$))+A$
65 ' POKE into area reserved for the message and set the
   pointer at 1EH
70 FOR I=1 TO LEN(A$):POKE &HCAFF+I,ASC(MID$(A$,I,1)):NEXT
80 POKE &H1E,&H0
90 POKE &H1F,&HCB
100 LPRINT                 ' Test
```

D GUIDELINES FOR DISK HANDLING

This appendix provides guidelines for the handling of disks and the procedures to be used for making disk back-ups. Most of the guidelines and techniques described will already be familiar to you, if you have developed your own software before. Nevertheless, we recommended that you read this appendix in order to minimize disk handling problems.

The information given here should be a part of the operating instructions passed on to the user of the software you have developed.

The appendix is divided up into four sections, disk handling, general information, the back-up procedure, and using the back-up procedure to handle errors.

D.1 Care of Disks

Disks are made of flexible plastic with a sensitive magnetic surface. Do not remove the disk casing or touch any open magnetized surfaces.

A disk should always be put back into its cover after it has been removed from a drive unit. Store it in a place free from magnetic disturbances (note: do not lay a disk on the P2000C for this reason). It is recommended that you use a disk box for storage purposes.

Temperature tolerance: 10° C - 52° C SL08
(50° F - 125° F) SL08

Humidity: 20% - 80% (no condensation) SL08

Only correctly specified disks which have been pre-formatted or formatted with the UTIL program can be used; others could damage the drive unit and the guarantee will be forfeited. If write errors occur during the first use, restart the procedure (check for write protection labels on the destination disk, incorrect insertion).

If the write procedure is still unsuccessful after retrying, the exact configuration and conditions should be stated in any reclamation.

D.2 General Information

- We recommend that you keep a computer logbook (see attachment).
- Disks should be inserted only after turning on the computer, and should be removed before it is turned off.
- Before the computer is turned off, make sure you have completed your program correctly. If not, data may be destroyed.
- After the end of a program has been prompted, disks should be removed from the drives and returned to their covers. The machine can then be turned off.
- A back-up should be made of every disk containing data, which had been altered during a work session. This is the only way you will be able to correct errors which might occur during a later session.
- Correct use of the COMPUTER LOGBOOK will enable you to record not only which programs and disks were in use, but also information on abnormal occurences and back-up procedures.
- Error detection can only be supported if a correct and exact logbook is available.

Guidelines for Disk Handling

D.3 The Back-Up Procedure

The following is known as the minimal back-up procedure. If you are using files which involve a very long processing procedure, we recommend making an extra (second generation) set, or even a periodical back-up (third generation).

There should be at least two generations (at different processing levels) for each data disk. These two generations should be marked with differently named or coloured labels.

D.3.1 Back-Up Procedure

The purpose of backing-up disks is to protect you from losing all your work if a disk error occurs. Every disk you work with should be copied after it has been updated, in order to ensure this protection. As an added form of security, a third disk should be used in the back-up procedure, giving you double protection.

When you first copy your main disk (known as the work disk), simply copy it to the second disk. When you next update your work disk, this time copy it to the third disk. If any mistakes occur during this copy, you will still have the second disk left intact.

From then on, whenever you make a back-up of an updated work disk, copy it to the disk which has the oldest copy; thus if there are any errors in that copy, you will still have a more up-to-date copy to work from. An example of this procedure is given in the table on the next page.

The back-up procedure should only be done with an authorized COPY DISK procedure (to be found in the VOLORG program - or under CP/M with the UTIL program). The WORK disk (Source) should always be inserted in drive unit 1 and the destination (COPY or Generation 2) disk in drive unit 2.

Guidelines for Disk Handling

Example of minimal back-up procedure:

RED (R)=WORK disk BLUE (B)=SAVE disk YELLOW(Y)=COPY disk

				STATUS	ACTION	CONTINUE
1	START	R	B	RED and BLUE identical		
2	PROGRAM			RED changed by program BLUE old level	Copy	3
3	COPY DISK	R	Y	Run OK Target destroyed Source destroyed	Copy Exchange YELLOW Copy	4 3 6
4	COPY DISK	R	B	Run OK Target destroyed Source destroyed	Ready for next session Exchange BLUE Exchange RED	1 4 5
5	COPY DISK	Y	R	Run OK Target destroyed Source destroyed	Copy Exchange RED Copy	4 5 6
6	COPY DISK	B	R	Run OK Target destroyed Source destroyed	Setback and re-enter all data since last backup Exchange RED Start from first session again	6

Exchange means that a new Disk has to be used.

Guidelines for Disk Handling

D.4 Using the Back-Up Procedure to Handle Errors

Abnormal program terminations are to be noted in the "COMPUTER LOGBOOK". If the reason is also known, it should be stated.

If an abnormal termination occurs, please follow the procedure described in the example in the previous section.

Never use back-up disks directly in any program session. After locating and solving a hardware or software problem, the technician and/or programmer will execute a test session (involving a test disk) to check if the problem has been solved.

Guidelines for Disk Handling

COMPUTER LOGBOOK

DATE	PROGRAM	DISK/COLOUR	OPERATOR	NOTES

Index

INDEX

	chapter	page
ADD	7.3.3	7-26
ADD INVERSION	7.3.4	7-33
Allocation	7.1.1	7-2
Alternate Files	7.1.4	7-4
Alternate Keys	7.1.3	7-4
Alternate Key Commands	7.3.4	7-27
Autostart	2.1	2-1
Booting	2.1	2-1
BUFFER Areas	7.5.1	7-41
CLOSE	7.3.1	7-11
Command Mode	3.1.10	3-12
COMPACT	7.4.3	7-37
Compare Disks	3.1.9	3-11
Configuration Program	4	4-1
COPY COMPACT	7.4.3	7-39
Copy Disk	3.1.4	3-6
Copy File	3.1.3	3-5
COPY REORGANIZE	7.4.3	7-39
CP/M	8.2	8-2
CREATE	7.3.1	7-12
CREATE ALTERNATE	7.3.4	7-28
Current Record Pointer (CRP)	7.5.2	7-42
Data Set	7.1.1	7-2
DELETE	7.3.3	7-22
Delete Disk	3.1.6	3-8
Delete File	3.15	3-7
DELETE INVERSION	7.3.4	7-34
Disk Directory	3.1.2	3-4
Disk Information	5.1	5-1
ERASE	7.4.3	7-37
Escape Sequences	9.2.2	9-3

Index

	chapter	page
File Commands	7.3.1	7-10
File Control Block	7.1.5	7-6
	7.5	7-40
File Management	7.1	7-1
File Name	3.1	3-2
File Size	3.1.8	3-10
Five-Inch Disks	5.1	5-1
Header	7.1.1	7-2
High-Resolution Graphic Mode	9.1	9-1
Key Blocks	7.1.1	7-2
Keyboard	8.3	8-5
Key Set	7.1.1	7-2
Keys	7.1.3	7-3
KSAFCB	7.1.5	7-6
KSAM 80 Files	7.1.1	7-2
KSAMUT	7.4	7-35
KSAMUT Function Specifications	7.4.3	7-36
LOAD	7.3.3	7-25
Logical Records	7.1.2	7-3
LOGICAL RECORD AREA	7.5.1	7-41
Memory Layout	8.1	8-1
OPEN	7.3.1	7-13
Primary Keys	7.1.3	7-4
Random Access Commands	7.3.3	7-19
Record Access	7.1	7-1
READ APPROXIMATE	7.3.3	7-24
READ BY ALTERNATE KEY	7.3.4	7-29
READ CURRENT	7.3.2	7-16

P2000 C

Index

	chapter	page
READ FIRST	7.3.4	7-30
READ NEXT	7.3.2	7-17
READ NEXT HOLDING	7.3.4	7-31
READ NTH RECORD	7.3.3	7-23
READ PREVIOUS	7.3.2	7-15
READ RANDOM	7.3.3	7-20
Screen Codes	9.2	9-1
Screen Handler	6	6-1
Screen Handler Activation	6	6-1
Screen Specifications	9.1	9-1
Sequential Access Commands	7.3.2	7-14
SHACT	6	6-1
STATUS	7.4.3	7-38
Status Information	9.3	9-10
UPDATE	7.3.3	7-21
VERIFY INVERSION	7.3.4	7-32
Video Terminal	9	9-1
VOLORG	3	3-1

Manual Status Control Form

BASIC INTERPRETER OPERATOR MANUAL

12NC: 5103 993 21121

This issue comprises the following updates:

no updates

Manual Comment Form

BASIC INTERPRETER OPERATOR MANUAL

12 NC: 5103 993 21121

Including Update(s)__ __ __

Originator:

Name .....
Address.....
.....
.....
.....

Comment (if possible, add a copy of the page(s) affected by the comment, marked with the proposed changes).

Please return this form to: Österr. Philips Ind. GesmbH
MAG-PERCO / MARKETING SUPPORT
A-1100 Wien, Triesterstr. 64,
A U S T R I A