

## PARAMETERS

<b>a</b>	=	a numeric expression or value to be converted into a string.	
<b>a\$</b>	=	a string holding numeric value or expression to be converted to a numeric variable or number.	
<b>addr</b>	=	address	(0-65536)
<b>ang</b>	=	angle (in radians)	
<b>b</b>	=	binary number	
<b>c</b>	=	condition	
<b>cn</b>	=	colour number	(0 to 7)
<b>col</b>	=	column	(0 to 31)
<b>d</b>	=	duration	(1=1 second)
<b>e</b>	=	expression	
<b>f</b>	=	device number	<b>f = 0</b> keyboard <b>f = 1</b> bottom two lines of screen <b>f = 2</b> screen lines 0 to 21 <b>f = 3</b> printer
<b>i</b>	=	integer	
<b>inc</b>	=	increment	
<b>item</b>	=	print items	
<b>len</b>	=	length	
<b>l\$</b>	=	string literal of one graphics character e.g. "P"	
<b>lin</b>	=	line	(0 to 21)
<b>ln</b>	=	line number	
<b>n,j,k</b>	=	number	
<b>p</b>	=	pitch	(0 to 69)
<b>rn</b>	=	row number	(0 to 7)
<b>r</b>	=	row	
<b>rad</b>	=	radius	
<b>sadd</b>	=	start address	(0 to 65536)
<b>st</b>	=	stream number	(0 to 15)
<b>v</b>	=	numeric variable	
<b>v\$</b>	=	string variable/literal string	



£1.99

# PHOENIX COMPUTER CRIB CARD

## SPECTRUM

**KEYWORDS**

**OPERATING COMMANDS**

**GRAPHIC AND SOUND COMMANDS**

**COLOUR COMMANDS**

**DATA COMMANDS**

**INPUT/OUTPUT COMMANDS**

**MICRODRIVE COMMANDS**



## THE KEYBOARD

KEYWORD	CURSORS	KEY
ABS	E	G
ACS	E	SHIFT W
AND	K,L,C	SHIFT Y
ASN	E	SHIFT Q
AT	K,L,C	SHIFT I
ATN	E	SHIFT E
ATTR	E	SHIFT L
BEEP	E	SHIFT Z
BIN	E	B
BORDER	K	B
BRIGHT	E	SHIFT B
CAPS LOCK	K,L	C/SH 2
CAT	E	SHIFT 9
CHR\$	E	U
CIRCLE	E	SHIFT H
CLEAR	K	X
CLOSE #	E	SHIFT 5
CLS	K	V
CODE	K	I
CONTINUE	K	C
COPY	K	Z
COS	E	W
DATA	E	D
DEF FN	E	SHIFT 1
DELETE	C,G	0
DELETE	K,L,C,G	C/SH 0
DIM	K	D
DRAW	K	W
EDIT	K,L,C	C/SH 1
ERASE	E	SHIFT 7
EXP	E	X
FLASH	E	SHIFT V
FN	E	SHIFT 2
FOR	K	F
FORMAT	E	SHIFT 0
GO SUB	K	H
GO TO	K	G
GRAPHIC CURSOR	K,L,C	C/SH 9
IF	K	U
IN	E	SHIFT I
INK	E	SHIFT X
INKEY\$	E	N
INPUT	K	I
INT	E	R
INVERSE	E	SHIFT M
LEN	E	K
LET	E	L
LINE	K	SHIFT 3
LIST	K	K
LLIST	E	V
LN	E	Z
LOAD	K	J
LPRINT	E	C
MERGE	E	SHIFT T
MOVE	E	SHIFT 6
NEW	K	A
NEXT	K	N
NOT	K,L,C	SHIFT S
OPEN #	E	SHIFT 4
OR	K,L,C	SHIFT U
OUT	E	SHIFT O
OVER	E	SHIFT N
PAPER	E	SHIFT C
PAUSE	K	M
PEEK	E	O



KEYWORD	SYNTAX	DEFINITION
<b>OPERATING COMMANDS</b>		
<b>CLEAR</b>	<b>CLEAR [n]</b>	<b>Clears variables</b> <b>[changes position of</b> <b>RAMTOP]</b>
<b>CONTINUE</b>	<b>CONTINUE</b>	<b>Continues execution of</b> <b>program after a BREAK</b> <b>or STOP command</b>
<b>EDIT</b>	<b>EDIT</b>	<b>Edits line indicated by</b> <b>the &gt; cursor</b>
<b>LOAD</b>	<b>LOAD ""</b>	<b>Loads first program</b> <b>encountered on the</b> <b>tape into memory</b>
	<b>LOAD v\$</b>	<b>Loads program called</b> <b>v\$ into memory</b>
	<b>LOAD v\$ CODE</b> <b>[[sadd],[len]]</b>	<b>Loads v\$. Loads len</b> <b>bytes into memory</b> <b>starting at address sadd</b>
	<b>LOAD v\$ DATA</b> <b>v[\$]()</b>	<b>Loads string or numeric</b> <b>array into memory</b>
	<b>LOAD v\$ SCREEN\$</b>	<b>Loads screen display v\$</b> <b>into memory</b>
<b>MERGE</b>	<b>MERGE v\$</b>	<b>Loads and merges v\$</b> <b>with the program</b> <b>currently residing in</b> <b>memory</b>
<b>NEW</b>	<b>NEW</b>	<b>Clears program and</b> <b>variables</b>
<b>RUN</b>	<b>RUN [ln]</b>	<b>Executes program</b> <b>[from ln]</b>
<b>SAVE</b>	<b>SAVE v\$</b>	<b>Saves the program</b> <b>called v\$ on tape</b>
	<b>SAVE v\$ CODE</b> <b>sadd,len</b>	<b>Saves v\$ on tape. Saves</b> <b>len bytes starting at</b> <b>address sadd</b>
	<b>SAVE v\$ DATA</b> <b>v[\$]()</b>	<b>Saves string or</b> <b>numeric array</b>
	<b>SAVE v\$ SCREEN\$</b>	<b>Saves screen display v\$</b> <b>on tape</b>
<b>STOP</b>	<b>STOP</b>	<b>Stops execution of a</b> <b>program</b>
<b>VERIFY</b>	<b>VERIFY v\$</b>	<b>Verifies that saving v\$</b> <b>on tape was successful</b>
	<b>VERIFY v\$ CODE</b> <b>sadd,len</b>	<b>Verifies v\$. Verifies that</b> <b>len bytes have been</b> <b>saved</b>
	<b>VERIFY v\$ DATA</b>	<b>Verifies that the array</b>



KEYWORD	SYNTAX	DEFINITION
<b>NUMBER FUNCTIONS</b>		
<b>ABS</b>	<b>ABS n</b>	Returns absolute value of n
<b>BIN</b>	<b>BIN b</b>	Returns decimal value of b. b cannot be a variable
<b>DEF FN</b>	<b>DEF FNv(v,v,...)</b>	Defines user-defined function
<b>EXP</b>	<b>EXP n</b>	Returns $e \wedge n$
<b>FN</b>	<b>FNv(v,v,...)</b>	Calls user-defined function
<b>INT</b>	<b>INT n</b>	Truncates n to return an integer value
<b>LN</b>	<b>LN n</b>	Returns the natural logarithm of n
<b>PI</b>	<b>PI</b>	3.1415927
<b>RANDOMIZE</b>	<b>RANDOMIZE [n]</b>	Random number seed value n=0 to 65535
<b>RND</b>	<b>RND</b>	Returns a pseudo-random number
<b>SGN</b>	<b>SGN n</b>	Returns signum of n
<b>SQR</b>	<b>SQR n</b>	Returns square root of n

### TRIGONOMETRIC FUNCTIONS

<b>ACS</b>	<b>ACS n</b>	Returns arc-cosine of n
<b>ASN</b>	<b>ASN n</b>	Returns arc-sine of n
<b>ATN</b>	<b>ATN n</b>	Returns arc-tangent of n
<b>COS</b>	<b>COS n</b>	Returns cosine of n
<b>SIN</b>	<b>SIN n</b>	Returns sine of n
<b>TAN</b>	<b>TAN n</b>	Returns tangent of n

### ARITHMETIC OPERATORS

symbol	operation	priority
+	addition	4



KEYWORD	SYNTAX	DEFINITION
<b>LOGICAL OPERATORS</b>		
<b>AND</b>	<b>c AND c</b>	Combines relations only returning True (1) if both conditions are True. j AND k=j if k<>0 or 0 if k=0 a\$ AND k=a\$ if k<>0 or "" if k=0
<b>IF</b>	<b>IF c THEN action</b>	A decision structure. The condition must be met (True) before the THEN statement is executed.
<b>NOT</b>	<b>NOT e</b>	Returns the logical inverse of an expression. NOT e=0 if e<>0 or 1 if e=0
<b>OR</b>	<b>c OR c</b>	Combines relations returning True (1) if either of its conditions are True.
<b>THEN</b>	<b>IF c THEN action</b>	The action is only executed if the condition is met

<b>LOOPS</b>		
<b>FOR</b>	<b>FOR v=n TO n1</b>	A loop structure. Calls subroutine starting at line ln, storing return address Transfers control of program to line ln Terminates loop when v=n1 (end value of the loop counter) Returns control to the line following the GOSUB statement Determines the increment value. If omitted inc=1
<b>GOSUB</b>	<b>GOSUB ln</b>	
<b>GOTO</b>	<b>GOTO ln</b>	
<b>NEXT</b>	<b>NEXT v</b>	
<b>RETURN</b>	<b>RETURN</b>	
<b>STEP</b>	<b>FOR v=n TO n1 [STEP inc]</b>	



**KEYWORD****SYNTAX****DEFINITION****I/O INSTRUCTIONS**

<b>AT</b>	<b>PRINT AT</b> <b>lin,col[;item]</b>	Moves print position to the point specified by lin,col
<b>CLS</b>	<b>CLS</b>	Clears the screen
<b>COPY</b>	<b>COPY</b>	Lists a COPY of the screen on the printer
<b>IN</b>	<b>IN addr</b>	Returns the byte read from the I/O port (addr).
<b>INKEY\$</b>	<b>V\$=INKEY\$</b>	Returns key-press. Does not wait for input
<b>INPUT</b>	<b>INPUT v,v\$,...</b>	<b>INPUT</b> numeric or string data
	<b>INPUT LINE v\$</b>	Allows a string to be <b>INPUT</b> without quotes
<b>INPUT #</b>	<b>INPUT # f;v,v\$,...</b>	Inputs string or numeric data
<b>INPUT #</b>	<b>INPUT # f; LINE v\$</b>	Allows the input of string data without quotes to device specified by f
<b>LINE</b>	<b>INPUT LINE v\$</b>	Allows a string to be <b>INPUT</b> without quotes
	<b>SAVE v\$ LINE ln</b>	<b>SAVEs</b> a program storing the line number from which it runs automatically when loaded
<b>LIST</b>	<b>LIST [ln]</b>	Lists program to the screen
<b>LIST #</b>	<b>LIST # f[,ln]</b>	Lists program to device specified by f
<b>LLIST</b>	<b>LLIST [ln]</b>	Lists program on the printer
<b>LPRINT</b>	<b>LPRINT v\$,v,...</b>	Prints argument on the printer
<b>OUT</b>	<b>OUT addr,n</b>	Writes value n to I/O port addr
<b>PRINT</b>	<b>PRINT [item]</b> <b>PRINT AT lin,col[;</b> <b>item]</b> <b>PRINT TAB col[;</b> <b>item]</b>	Prints to the screen Prints item at the point specified by lin,col Prints item at the point specified by col
<b>PRINT #</b>	<b>PRINT # f[[;item],</b> <b>item]</b>	Prints to device f



KEYWORD	SYNTAX	DEFINITION
<b>MICRODRIVE COMMANDS</b>		
<b>FORMAT</b>	<b>FORMAT "m";n;v\$</b>	Formats cartridge in drive n and names it v\$
	<b>FORMAT "n";n</b>	Sets network station number to n
	<b>FORMAT "t";n</b>	Sets baud rate for RS232 interface to n
	<b>FORMAT "b";n</b>	Sets baud rate for RS232 interface to n
<b>INKEY\$#</b>	<b>INKEY\$# st</b>	Returns a single character from stream st, of "" if no character is available
<b>MOVE</b>	<b>MOVE source TO des</b>	Moves data from source to des (destination)
<b>OPEN#</b>	<b>OPEN# st,n</b>	Enables I/O to channel n of stream st

## GRAPHICS AND COLOUR COMMANDS

<b>ATTR</b>	<b>ATTR (lin,col)</b>	Returns the colour attribute at the point lin,col.
<b>BORDER</b>	<b>BORDER cn</b>	Makes border the colour cn
<b>BRIGHT</b>	<b>BRIGHT z</b>	Brightness control, z=0 normal, z=1 bright, z=8 transparent
<b>CIRCLE</b>	<b>CIRCLE x,y,rad</b>	Draws a circle with centre x,y and radius rad
<b>DRAW</b>	<b>DRAW x,y[,ang]</b>	Draws line from previous point to relative x,y co-ordinates [turning through angle ang]
<b>FLASH</b>	<b>FLASH z</b>	Controls flashing, z=0 normal, z=1 flash, z=8 no change
<b>INK</b>	<b>INK cn</b>	Makes the ink the colour cn, cn=8 no change



KEYWORD	SYNTAX	DEFINITION
<b>DATA COMMANDS</b>		
<b>DATA</b>	<b>DATA v,v,....</b>	Stores data items: numeric
	<b>DATA v\$,v\$,...</b>	Stores data items: string
	<b>DATA v,v\$,...</b>	Stores data items: both numeric and string
<b>DIM</b>	<b>DIM v(n[,n..])</b>	Dimensions numeric array
	<b>DIM v\$(n[,n,..])</b>	Dimensions string array
<b>LET</b>	<b>LET v=n</b>	Defines numeric variable
	<b>LET v\$=v\$</b>	Defines string variable
<b>READ</b>	<b>READ v[,v,.]</b>	Reads numeric data into v
	<b>READ v\$[,v\$,...]</b>	Reads string data into v\$
	<b>READ v[\$](n)</b>	Reads string/numeric data into the array
<b>RESTORE</b>	<b>RESTORE [ln]</b>	Restores data. Moves pointer to the start of data [or line ln]

### USEFUL POKES & PEEKS

<b>POKE 23562,3</b>	Key repeat
<b>POKE 23561,n</b>	n/50 of a second delay before key repeat
<b>POKE 23609,n</b>	louder click, n=1 to 15
<b>POKE 23609,n</b>	louder beep, n=15 to 255
<b>POKE 23609,0</b>	disables to normal click
<b>POKE 23692,0</b>	auto scroll
<b>POKE 23659,n</b>	stops printing on bottom n lines, n=2 to 25
<b>POKE 23659,1</b>	print on 23 lines
<b>POKE 23659,0</b>	print on 24 lines
<b>POKE (PEEK 23635+256*PEEK 23636),n</b>	Protects a program, n=40 to 63 will change previous lines to a symbol



## ATTRIBUTE FILE

Each character display location is complemented by an attribute location which determines display mode (i.e. INK, PAPER, BRIGHT etc). The start address for the attribute is 23528 and the end address 23295. By POKEing the appropriate value into the required location the display features of a given character can be set. The codes below set paper and ink. For example:

**POKE 23295,2**

Will set INK to red and PAPER to black. For BRIGHT simply add 64 to any code:

**POKE 23295,68**

and to create the FLASH add a further 128:

**POKE 23295,196**

The intersection point in the table below establishes the initial INK/PAPER code:

INK COLOUR	PAPER COLOUR							
	BLACK	BLUE	RED	MAGENTA	GREEN	CYAN	YELLOW	WHITE
BLACK	0	8	16	24	32	40	48	56
BLUE	1	9	17	25	33	41	49	57
RED	2	10	18	26	34	42	50	58
MAGENTA	3	11	19	27	35	43	51	59
GREEN	4	12	20	28	36	44	52	60
CYAN	5	13	21	29	37	45	53	61
YELLOW	6	14	22	30	38	46	54	62
WHITE	7	15	23	31	39	47	55	63

## DISPLAY FILE

The full Spectrum screen makes use of 6144 locations for character display and, since each character requires eight locations, individual symbols can be displayed in a total of 768 print positions (6144/8). In order to position a character on the 24 x 32 screen we require the start address for the position in question. The table which follows provides the start address for the first and last character position of each of the twenty-four rows:



## ERROR MESSAGES

These appear at the bottom of the screen whenever the computer stops executing some BASIC, and explain why it stopped.

The Error Messages have a code number or letter for you to refer to.

CODE	MEANING	PROBABLE REASONS
1	<b>NEXT without FOR</b> The control variable does not exist (it has not been set up by a FOR statement), but there is an ordinary variable with the same name.	<b>NEXT</b>
2	<b>Variable not found</b> For a simple variable this will happen if the variable is used before it has been defined by a LET, READ or INPUT statement loaded from tape or set up in a FOR statement.	<b>Any</b>
3	<b>Subscript wrong</b> A subscript is beyond the dimension of the array, or there are the wrong number of subscripts.	<b>Subscripted variables, Substrings</b>
4	<b>Out of memory</b> There is not enough room in the computer for what you are trying to do.	<b>LET, INPUT, FOR, DIM, GO SUB, LOAD, MERGE. Sometimes during expression evaluation</b>
5	<b>Out of screen</b> An INPUT statement has tried to generate more than 23 lines in the lower half of the screen. Also occurs with PRINT AT 22,...	<b>INPUT, PRINT AT</b>
6	<b>Number too big</b>	<b>Any arithmetic</b>
7	<b>RETURN without GO SUB</b> There has been one more RETURN than there were GO SUBs.	<b>RETURN</b>
9	<b>STOP statement</b>	<b>STOP</b>



## ERROR MESSAGES

These appear at the bottom of the screen whenever the computer stops executing some BASIC, and explain why it stopped.

The Error Messages have a code number or letter for you to refer to.

CODE	MEANING	PROBABLE REASONS
E	Out of DATA You have tried to READ past the end of the DATA list.	READ
F	Invalid file name SAVE with name empty or longer than 10 characters	SAVE
G	No room for line There is not enough room left in memory to accommodate the new program line.	Entering a line into the program
H	STOP in INPUT Some INPUT data started with STOP, or INPUT LINE was pressed.	INPUT
I	FOR without NEXT There was a FOR loop to be executed and the corresponding NEXT statement could not be found.	FOR
J	Invalid I/O device	Microdrive, etc, operations
K	Invalid colour The number specified is not an appropriate value.	INK, PAPER, BORDER, FLASH, BRIGHT, INVERSE, OVER; also after one of the corresponding control characters
L	BREAK into program BREAK pressed, this is detected between two statements.	Any
M	RAMTOP no good The number specified for RAMTOP is either too big or too small.	CLEAR; possibly in RUN



£1.99

# PHOENIX COMPUTER CRIB CARD

## SPECTRUM

**KEYWORDS**

**OPERATING COMMANDS**

**GRAPHIC AND SOUND COMMANDS**

**COLOUR COMMANDS**

**DATA COMMANDS**

**INPUT/OUTPUT COMMANDS**

**MICRODRIVE COMMANDS**