N Wilhelm

MITS™
DISK-EXTENDED
BASIC
PROGRAMMING REFERENCE BOOK
JULY, 1978

PCC PERTEC COMPUTER CORPORATION

Dec 266-2290

INITIALIZATION DIALOG

MEMORY SIZE?
-To use all of memory, type <return>.
-To limit memory, type <number of bytes>.

LINE PRINTER?
-Centronics printer, enter C.
-Qume printer, enter Q.
-No printer, enter C.

NUMBER OF FILES?
-Enter maximum number of disk files to be open at one time

NUMBER OF RANDOM FILES? (300/25 Series BASIC only
-Enter the maximum number of random disk files to be ope
at one time (must be less than or equal to NUMBE
OF FILES).

ENTER DAY?

ENTER MONTH?

ENTER YEAR?
-Enter day, month, and year.

SPECIAL CHARACTERS [c indicates control]

Ac	Edits line being entered or last line typed
Cc	Interrupts program execution Returns to BASIC command level (types OK)
Hc	Erases last character typed (backspace)
c	Tab. Used for program formatting (8 columns)
Oc	Toggle output switch
Qc	Resume from pause switch
Sc	Pause switch
Uc	Erases line being typed
Xc	Same as Uc
return>	Ends every line typed in
linefeed>	Used to break a logical line into physical lines
rubout>	Erases last character typed
	Current line for EDIT, RENUM, DELETE, LIST, LLIST commands
O or &	Octal constant
H	Hexadecimal constant
	Separates statements typed on the same line
	Equivalent to PRINT (Note: L? is not equivalent to LPRINT)

ariable Type Declaration

String	0 to 255 characters
Integer	—32768 to 32767
Single precision	7.1 digit floating point
Double precision	16.8 digit floating point

COMMANDS

Statement	Syntax/Function	Example
CLEAR	CLEAR [n] Clear program variables	CLEAR 100
CONT	CONT Continue program execution	CONT
DELETE	DELETE [[start line][-[end line]]] Delete line[s] in a program	DELETE 20-25
EDIT	EDIT n Edit a program line. See EDIT MODE subcommands.	EDIT 110
FILES	FILES [drive number] List files in disk directory	FILES 1
LIST	LIST [[start line][-[end line]]] List program lines at terminal	LIST 100-1000
LLIST	LLIST [[start line][-[end line]]] List program lines to printer	LLIST 50-250
LOAD	LOAD [[, disk number], R] Load a program file. R option means RUN.	LOAD"INVEN"
MERGE	MERGE filename [, disk number] Merge program on disk with one in memory. File must have been SAVEd in ASCII mode.	MERGE"SUB1"
NEW	NEW Delete current program and variables.	NEW
RENUM	RENUM [first number[,first line[,increment]]] Renumber program lines	RENUM 100,,100
RUN	RUN [line number] Run a program [from line number]	RUN RUN 50
	RUN filename [, disk number] Run a program.	RUN"TEST"
SAVE	SAVE filename [[,disk number],A] Save the program in memory with name "filename." A saves program in ASCII.	SAVE"MYPROG"
TROFF	TROFF Turn Trace off	TROFF
TRON	TRON Turn Trace on	TRON
WIDTH	WIDTH n Set terminal carriage width. Default is 72. Must be between 15 and 255.	WIDTH 80

EDIT MODE SUBCOMMANDS

Command	Function
A	Restart EDIT at the start of the line
nCc	Change n character[s]. . .
nD	Delete n character[s] at the current position
E	End editing and save changes but don't type the rest of the line
H string <escape>	Delete the rest of the line and Insert string
I string <escape>	Insert string at current position
nKc	Kill all characters from current position up to the nth occurrence of c
L	Print the rest of the line and go to the start of the line
Q	Quit and cancel all changes
nSc	Search for nth occurrence of c from current position
Xstring <escape>	Go to the end of the line and insert string
<rubout> or <backspace>	In Insert mode, deletes previous character; else, backs up buffer pointer
<space>	Advances buffer pointer
<return>	End editing and save changes

Note: A string argument can be terminated with <return>. The effect is equivalent to string <escape> <return>.

PROGRAM STATEMENTS (Except I/O)

Statement	Syntax/Function	Example
DEF	DEF FNx [argument list] = expression Define an arithmetic or string function	DEF FNA (X,Y) = SQR (X*X + Y*Y)
	DEF USRn = address Define the entry address for the nth assembly language subroutine	DEF USR3 = &2000
DEFDBL	DEFDBL letter [-letter] Define double precision default variable names	DEFDBL A.
DEFINT	DEFINT letter[-letter] Define integer default variable names	DEFINT I-N
DEFSNG	DEFSNG letter[-letter] Define single precision default variable names	DEFSNG X-Z
DEFSTR	DEFSTR letter[-letter] Define string default variable names	DEFSTR S-V
DIM	DIM variable [size1 [,size2...]]... Allocate space for arrays. Arrays may be dimensioned dynamically.	DIM A(3) B$(10,2,3) DIM Z (2*1)

PROGRAM STATEMENTS (Cont'd)

END	END Stop program and return to BASIC command level	END
ERASE	ERASE variable [,variable]... Release space previously reserved for an array	ERASE A,B$
ERROR	ERROR code Generate error of code (see table). May call user ON ERROR routine or force BASIC to handle error.	ERROR 17
FOR	FOR variable = expression TO expression [STEP expression] Used with NEXT statement to repeat a sequence of program lines. The variable is incremented by the value of STEP.	FOR I = 1 to 5 STEP .5...
GOSUB	GOSUB line number Call a BASIC subroutine by branching to the specified line number. See RETURN.	GOSUB 210
GOTO	GOTO line number Branch to specified line number	GOTO 90
IF	IF expression THEN statement [ELSE statement...] The relation X < Y is tested. If true, the THEN clause is executed. If false, the ELSE clause is executed.	IF X < Y THEN Y = X ELSE Y = A
LET	[LET] variable = expression Assign a value to a variable	LET X = I + 5
NEXT	NEXT variable [, variable].. Delimits the end of a FOR loop	NEXT I
ON ERROR GOTO	ON ERROR GOTO line number Enables error trap subroutine beginning at specified line. If line number = 0, disables error trapping. If line number = 0 inside error trap routine, forces BASIC to handle error.	ON ERROR GOTO 1000
ON... GOSUB	ON expression GOSUB line[,line]... GOSUBs to statement specified by expression. [If J + 1 = 1, to 20; if J + 1 = 2, to 20; if J + 1 = 3, to 40]	ON J + 1 GOSUB 20,20,40
ON... GOTO	ON expression GOTO line[,line]... Branches to statement specified by I [To 20 if I = 1, to 30 if I = 2]	ON I GOTO 20,30
OUT	OUT port, byte Puts byte 16 + 1 to output port 41	OUT 41, 16 + 1
POKE	POKE location, byte Puts byte specified into memory location specified	POKE &23100, 255

PROGRAM STATEMENTS (Cont'd)

REM	REM any text Allows user to insert comments in program Note: ":" does not terminate a REM statement.	REM comment
RESTORE	RESTORE Resets DATA pointer so that DATA statements may be re-read	RESTORE
RESUME	RESUME or RESUME 0 Returns from ON...ERROR routine to statement that caused error	RESUME
	RESUME NEXT Returns to statement after the one that caused the error	RESUME NEXT
	RESUME line Returns to the specified line	RESUME 100
RETURN	RETURN Return from subroutine to statement following last GOSUB performed	RETURN
STOP	STOP Stop program execution, print BREAK message, and return to COMMAND mode	STOP
SWAP	SWAP variable, variable Exchanges values of two variables. (The variables must be of the same type.)	SWAP A$,B$
WAIT	WAIT port, mask [,select] Waits for specified condition on a hardware port	WAIT 21,1

PRINT USING Format Field Specifiers

Numeric Specifier	Possible Digits	Field Characters	Definition	Example
#	1	1	Numeric field	####
.	0	1	Decimal point	#.#
+	0	1	Print leading or trailing sign. Positive numbers will have " + ", negative numbers will have "—".	+## ###+
—	0	1	Trailing sign-prints "—" if negative, otherwise blank.	##.##—
**	2	2	Leading asterisk fill	**###.##
$$	1	2	Floating dollar sign. $ is placed in front of the leading digit.	$$##.##
**$	2	3	Asterisk fill and floating dollar sign	**$#.##

PRINT USING (Cont'd)

	1	1	Use comma every three digits (left of decimal point only)	##,###.##
↑↑↑↑	0	4	Exponential format. Number is aligned so leading digit is non-zero.	#.##↑↑↑↑
String				
!			Single character	!
\<spaces>\			2 + numbers of spaces character field	\ \

INPUT/OUTPUT STATEMENTS

Statement	Syntax/Function	Example
CLOSE	CLOSE [f_1[,f_2...]] Closes disk files. If no argument, all open files are closed.	CLOSE 6
DATA	DATA item [,item...] Specifies data to be used in a READ statement	DATA2.3 "PLUS",4
FIELD	FIELD [#] f, count AS string variable [,count AS string variable]... Define fields in a random file buffer	FIELD#1, 3 AS A$
GET	GET [#] f [,record number] Read a random record from a data file	GET#1, 17*1
INPUT	INPUT["prompt string literal";] variable [,variable]... Read data from the terminal	INPUT"VALUES"; A,B
	INPUT #f, variable[,variable]... Read data from a disk file	INPUT #2,A,B
KILL	KILL filename [,disk number] Delete a disk file	KILL "INVEN"
LINE INPUT	LINE INPUT ["prompt string literal";] string-variable Read an entire line from the terminal	LINE INPUT A$
	LINE INPUT #f, string-variable Read an entire line from a disk file	LINE INPUT#2,B$
LSET	LSET field variable = expression Store data in random file buffer, left justified.	LSET A$ = "JOHN JONES"
OPEN	OPEN mode-string, [#] file number,filename [,disk number] Open a disk file. Mode string must be one of: I = input file O = output file R = random file	OPEN"O",#1, "output"
PRINT	PRINT [USING format string;] exp [,exp]... Print data at the terminal using the format specified. See table for format characters.	PRINT USING "!"; A$,B$

INPUT/OUTPUT STATEMENTS (Cont'd)

	PRINT #f, [USING format string;] exp [,exp]... Write data to a disk file	PRINT #4,A,B
	LPRINT [USING format;] variable [,variable]... Write data to a line printer	LPRINT A,B
PUT	PUT [#]f[,record number] Writes data from a random buffer to a data file	PUT #3,4
READ	READ variable [,variable]... Reads data into specified variables from a DATA statement	READ I,X,A$
RSET	RSET field variable = expression Puts data into a random file buffer, right justified	RSET B$ = "CORRECT"

OPERATORS (in order of precedence)

1.	()	Expressions in parentheses
2.	↑	Exponentiation
3.	*,/	Multiplication, division
4.	\	Integer division
5.	MOD	Modulus arithmetic
6.	+,—	Addition, subtraction
7.	<	Less than
	<=	Less than or equal to
	>	Greater than
	>=	Greater than or equal to
	=	Equal to
	<>	Not equal to
8.	NOT	Logical negation
9.	AND	Logical disjunction
10.	OR	Logical conjunction
11.	XOR	Logical exclusive OR
12.	EQV	Logical equivalence
13.	IMP	Logical implication

ARITHMETIC FUNCTIONS

Statement	Function	Example
ABS	Absolute value of expression	Y = ABS(A + B)
ATN	Arctangent of the expression [in radians]	PRINT ATN (A)
CDBL	Convert the expression to a double precision number	A = CDBL(Y)
CINT	Convert the expression to an integer	B = CINT(B)
COS	Cosine of the expression [expression in radians]	A = COS(3)
CSNG	Convert the expression to a single precision number	C = CSNG(X)
ERL	Error line number	PRINT ERL
ERR	Error code number	PRINT ERR
EXP	Raises the constant e to the power of the variable	B = EXP(C)

DSKF(DRIVE) RETURNS FREE SPACE ON DRIVE

ARITHMETIC FUNCTIONS (Cont'd)

FIX	Returns truncated integer of expression	J = FIX(A/B)
FRE	Gives memory free space not used by BASIC	PRINT FRE(0)
INP	Inputs a byte from an Input port	PRINT INP(21)
INT	Evaluates the expression for the largest integer contained	C = INT(X + 3)
LOG	Gives the natural logarithm of the expression	D = LOG(Y — 2)
LPOS	Carriage position of line printer	LPOS(3)
PEEK	Reads a byte from memory at address specified	PRINT PEEK (&2000)
POS	Carriage position of terminal	POS(5)
RND	Generates a random number. Parameter <0 seed new sequence 0 return previous random number >0 return new random number	E = RND(1)
SGN	1 if expression >0 0 if expression = 0 —1 if expression <0	B = SGN(X + Y)
SIN	Sine of the expression [expression in radians]	B = SIN(A)
SPC	Used in PRINT statement to print spaces	PRINT SPC(5)
SQR	Square root of the expression	C = SQR(D)
TAB	Used only in PRINT statement. Tabs carriage to specified position	PRINT TAB(20)
TAN	Tangent of the expression [expression in radians]	D = TAN(3,14)
USRn	Calls the user's machine language subroutine with the specified argument. See DEFUSR	PRINT USR (&27000)
VARPTR	Returns address of variable in memory	I = VARPTR(X)

STRING FUNCTIONS

Statement	Function	Example
ASC	Returns the ASCII value of the first character of a string	PRINT ASC(A$)
CHR$	Returns a one-character string whose character has the ASCII code of exp	PRINT CHR$(48)
FRE	Returns remaining string free space	PRINT FRE(A$)
HEX$	Converts a number to a hexadecimal string	H$ = HEX$(100)
INSTR	Returns the first position of the first occurrence of string 2 in string 1 starting at position exp	INSTR(A$,":") INSTR(X$,Y$)

STRING FUNCTIONS (Cont'd)

LEFT$	Returns leftmost length characters of the string expression	B$ = LEFT$ (X$,8)
LEN	Returns the length of a string	PRINT LEN (B$)
MID$	Returns characters from the middle of the string starting at the position specified to the end of the string or for length characters	A$ = MID$ (X$,5,10)
	MID$ may also be used to assign a substring inside a string	MID$ (A$,3,2) = "TO"
OCT$	Converts a number to an octal string	O$ = OCT$ (100)
RIGHT$	Returns rightmost length characters of the string expression	C$ = RIGHT$ (X$,8)
SPACE$	Returns a string of exp spaces	S$ = SPACE$ (20)
STR$	Converts a numeric expression to a string	PRINT STR$(35)
STRING$	Returns a string count long containing first character of string	X$ = STRING$ (100,"A")
STRING$	Returns a string count long containing characters with numeric value exp	Y$ = STRING$ (100,42)
VAL	Converts the string representation of a number to its numeric value	PRINT VAL ("3.1")

INPUT/OUTPUT FUNCTIONS

Name	Function	Syntax
CVD	Converts an 8-character string to a double precision number	CVD (8-character string)
CVI	Converts a 2-character string to an integer	CVI (2-character string)
CVS	Converts a 4-character string to a single precision number	CVS (4-character string)
EOF	Returns true [—1] if a sequential file is positioned at its end	EOF (file number)
LOC	Next record number to read [random file]. Number of records read or written [sequential file].	LOC (file number)
LOF	Returns number of records written in current extent of random file	LOF (file number)
MKD$	Converts a double precision number to an 8-character string	MKD$ (double precision value)
MKI$	Converts an integer to a 2-character string	MKI$ (integer value)
MKS$	Converts a single precision number to a 4-character string	MKS$ (single precision value)

BASIC ERROR CODES

Code	Error
1	NEXT without FOR
2	Syntax error
3	RETURN without GOSUB
4	Out of data
5	Illegal function call
6	Overflow
7	Out of memory
8	Undefined line
9	Subscript out of range
10	Redimensioned array
11	Division by zero
12	Illegal direct
13	Type mismatch
14	Out of string space
15	String too long
16	String formula too complex
17	Can't continue
18	Undefined user function
19	No RESUME
20	RESUME without error
21	Unprintable error
22	Missing operand
23	Line buffer overflow
Disk Errors	
50	Field overflow
51	Internal error
52	Bad file number
53	File not found
54	Bad file mode
55	File already open
56	Disk not mounted
57	Disk I/O error
58	File already exists
59	Set to non-disk string
60	Disk already mounted
61	Disk full
62	Input past end
63	Bad record number
64	Bad file name
65	Mode-mismatch
66	Direct statement in file
67	Too many files
68	Out of random blocks
69	File link error

HARD DISK ERROR CODES

Code	Error
01	Undefined system error
05	Invalid mode
07	Unable to allocate buffer
09	Invalid drive number
0A	Attempt to write read-only volume
13	Volume already mounted
15	Invalid drive number in mount
17	Drive not mounted
1F	File name not found
21	Attempt to open too many files
23	Out of index entry blocks
27	Invalid file number
29	File number not opened
2F	Attempt to open read-only file in write mode
31	Not enough space on volume
33	File name already on volume
35	Insufficient directory space for new file
4F	Unable to update directory
55	Attempt to read past end of file
57	Attempt to allocate to allocated group
59	Attempt to access invalid logical page
5B	No free bit found during allocation
5D	Attempt to write/kill read-only file
5F	Unable to update directory entry
61	Unable to update volume descriptor
63	Invalid drive number
67	Invalid index pointer
69	Invalid seek mode parameter
6F	Error in closing file during dismount
71	Cannot kill open file
73	Buffer pool write error
97	No response — timed out
99	Drive not ready
9B	Header CRC error
9D	Wrong CRC
9F	Data CRC error
A1	Data verification error
A3	Write protected
A5	Transient error

DOUBLE DENSITY FLOPPY DISK ERROR CODES

Code	Error
01	Illegal driver opcode
02	Data CRC error
03	Header CRC error
04	DMA timeout error
05	Diskette format failure
06	Write protection error
07	Seek error
41	Illegal opcode to file manager
42	Bad unit number
43	Unit already mounted
44	Unit not mounted
45	File already exists
46	Directory full
47	Logical unit full
48	No such file
49	Illegal operation on protected file
4A	Bad file number
4B	File already open
4C	No free file control blocks
4D	Bad file mode
4E	File not open
4F	Input past end
50	Input of record never written
51	Bad record number

DECIMAL-HEX-ASCII TABLE

D	H	ASCII
0	00	NUL
1	01	SOH
2	02	STX
3	03	ETX
4	04	EOT
5	05	ENQ
6	06	ACK
7	07	BEL
8	08	BS
9	09	HT
10	0A	LF
11	0B	VT
12	0C	FF
13	0D	CR
14	0E	SO
15	0F	SI
16	10	DLE
17	11	DC1 [X-ON]
18	12	DC2 [TAPE]
19	13	DC3 [X-OFF]
20	14	DC4
21	15	NAK
22	16	SYN
23	17	ETB
24	18	CAN
25	19	EM
26	1A	SUB
27	1B	ESC
28	1C	FS
29	1D	GS
30	1E	RS
31	1F	US
32	20	SP
33	21	!
34	22	"
35	23	#
36	24	$
37	25	%
38	26	&
39	27	/
40	28	[
41	29	]
42	2A	*
43	2B	+
44	2C	'
45	2D	-
46	2E	.
47	2F	/
48	30	0
49	31	1
50	32	2
51	33	3
52	34	4
53	35	5
54	36	6
55	37	7
56	38	8
57	39	9
58	3A	:
59	3B	;
60	3C	<
61	3D	=
62	3E	>
63	3F	?

D	H	ASCII
64	40	
65	41	A
66	42	B
67	43	C
68	44	D
69	45	E
70	46	F
71	47	G
72	48	H
73	49	I
74	4A	J
75	4B	K
76	4C	L
77	4D	M
78	4E	N
79	4F	O
80	50	P
81	51	Q
82	52	R
83	53	S
84	54	T
85	55	U
86	56	V
87	57	W
88	58	X
89	59	Y
90	5A	Z
91	5B	[
92	5C	/
93	5D	]
94	5E	
95	5F	—
96	60	/
97	61	a
98	62	b
99	63	c
100	64	d
101	65	e
102	66	f
103	67	g
104	68	h
105	69	i
106	6A	j
107	6B	k
108	6C	l
109	6D	m
110	6E	n
111	6F	o
112	70	p
113	71	q
114	72	r
115	73	s
116	74	t
117	75	u
118	76	v
119	77	w
120	78	x
121	79	y
122	7A	z
123	7B	{
124	7C	|
125	7D	} [ALT MODE]
126	7E	
127	7F	DEL [RUB OUT]

PERTEC
COMPUTER
CORPORATION
MICROSYSTEMS DIVISION

20630 Nordhoff Street, Chatsworth, California 91311
Phone (213) 998-1800 / TWX 910-494-2788

PERTEC
COMPUTER
CORPORATION
BUSINESS SYSTEMS DIVISION

International Marketing Headquarters
17112 Armstrong Avenue, Irvine, California 92714
Phone (714) 540-8340 / TWX (910) 595-1912

78519 (7-78)