

## CP/M Commands

### Built-in commands

#### DIR

Lists the names of all files on the currently logged in drive.

#### DIR dr:

Lists the names of all files on the specified drive.

#### DIR dr:afn

Lists the names of all files satisfying the specified ambiguous file name (afn) on the display screen.

#### ERA dr:ufn or ERA dr:afn

Erases a specific file or all files that satisfy a specific ambiguous file name from the specified disk device

#### REN dr:ufn1=ufn2

Changes the name of a file which is on the specified drive from ufn2 to ufn1.

#### SAVE n drain

Saves the contents of the specified number of pages of the transient program *area* to a disk device under the specified file name (Each page consists of 256 bytes of memory).

#### TYPE dr:ufn

Displays the contents of the specified ASCII file (a file stored in ASCII codes) in the specified disk device.

#### USER n

Allows different users to specify their own logical directories on the same disk device. n is the user number (an integer from 0 to 15) which specifies a logical directory.

### Transient commands

#### CONFIG

The CONFIG command sets the system environment of the PX-8 and determines various default settings.

#### FILINK

The FILINK program transfers files between PX-8 and another computer (such as another PX-8, QX-10, etc.) through the RS-232C communication line.

## PIP

Activates the PIP commande

**PIP d:=s:**

<b>afn</b>
<b>ufn</b>

Copies the specified file(s) from drive s to drive d.

**PIP**

<b>LST:</b>	<b>=s:ufn</b>
<b>PUN:</b>	
<b>CON:</b>	

Transfers the specified file from drive S to the specified device.

**PIP d:ufn = RDR:**

Transfers data from RDR: to the specified file.

**PIP d:newname. typ=s1:ufn1, s2:ufn2 ...**

Concatenates the specified files to generate a new file.

## PIP parameters

### [B] (Block)

when specified, PIP performs block mode transfer.

### [Dn] (Delete)

When specified, PIP deletes characters exceeding column n of each line.

### [E] (Echo)

When specified, data transferred is also output to the console.

### [F] (Form feed)

When specified, form feed characters (OCH) are removed from data transferred.

### [Gn]

When specified, data can be transferred from another user area.

### [H] (Hex format)

When specified, PIP checks to confirm that data transferred is in Intel HEX format. (If not, operation is terminated.)

### [I] (Ignore null)

When specified, PIP ignores null records (00H) and checks to confirm that data transferred is Intel HEX format. (If not, operation is terminated.)

### [L] (Lower case)

When **specified**, all upper case characters are converted to lower case characters.

### [N] (Line number)

When specified, line numbers are added to the beginning of each line. Specifying "N2" adds zeros to the beginning of each line number.

### [O] (Object files)

When specified, PIP ignores the physical end of file code (1AH) during concatenation and transfer. This makes it possible to transfer files other than ASCII files.

### [Pn] (Page eject)

When specified, PIP makes a page eject every n lines. When n is 1 or omitted, 60 is assumed.

### [Qstring ^Z] (Quit)

When specified, PIP quits transfer when the specified string is detected.

### [R] (Read)

Makes it possible to transfer SYS files. (The [W] parameter is set automatically when [R] is specified.)

### [Sstring ^Z] (Start)

When specified, PIP searches the data to be transferred for the specified string, then starts transfer from the point at which it is detected.

### [Tn] (Tab)

When specified, the tab width for transfer is set to n columns.

### [U] (upper case)

When specified, all lower case characters are converted to upper case characters.

### [V] (Verify)

When specified, each file is verified as it is copied.

**[W] (Write In R/O)**

When specified, the R/O attribute is ignored for destination files.

**[Z] (Zero parity)**

When specified, parity bits in data received are reset to zero.

**STAT**

Displays the amount of free space on the disk in the currently logged-in drive and other drives which have been at least once used, as well as their R/W attribute.

**STAT dr:**

Displays the amount of free space on the disk in the specified drive.

**STAT dr:filename.typ**

Displays the size and attributes of the specified file.

**STAT dr:filename.typ \$\$**

Displays the size and attributes of the specified file in detail.

**STAT dr:filename.typ \$R/O**

Sets the specified file to read only

**STAT dr:filename.typ \$R/W**

Makes it possible to read or write the specified file.

**STAT dr:filename.typ \$SYS**

Sets the SYS attribute for the specified file.

**STAT dr:Filename.typ \$DIR**

Sets the DIR attribute for the specified file.

**(Device related formats)**

**STAT DEV:**

Displays the current physical-to-logical device assignments (that is, the contents of IOBYTE).

**STAT VAL:**

Displays instructions for specifying the operand of the STAT command.

**STAT DSK:**

Displays the status of the currently logged-in drive and other drives which have been used at least once.

**STAT USR:**

Displays the current user number and user numbers which have active files on the current disk.

**STAT dr: = R/O**

Sets the specified drive to read only. The read only state remains effective until a cold or warm start is made.

**STAT logical: = physical:**

Assigns the specified physical device to the specified logical device.

**SUBMIT filename parameters**

Executes the commands in command procedure file "filename.SUB" using parameters.

**TERM**

The TERM program makes it possible to connect the PX-8 to a host computer through a RS-232C communication line for use as a terminal.

**XSUB**

Extends the function of SUBMIT.

## BASIC Commands and statements

### ABS

- FORMAT** ABS(X)  
**PURPOSE** Returns the absolute value of expression X,  
**EXAMPLE** A= ABS(- 10)

### ALARM

- FORMAT** ALARM [<date>, <time>, <string> [.W]]  
**PURPOSE** Specifies the alarm or wake time.  
**EXAMPLE** ALARM "01/31/84", "10:00:00", "Staff meeting"  
ALARM " \* \*\* \* \*\* \* " , "08:00:00";  
"A:MORNING" + chr\$(13), W

### ALARM\$

- FORMAT** ALARM\$ [<function>]  
**PURPOSE** Returns information about the ALARM setting"  
**EXAMPLE** PRINT ALARM\$(1)

### ASC

- FORMAT** ASC(X\$)  
**PURPOSE** Returns the numeric value which is the ASCII code for the first character of string X\$.  
**EXAMPLE** A = ASC("A")

### ATN

- FORMAT** ATN(X)  
**PURPOSE** Returns the arc tangent in radians of X.  
**EXAMPLE** A = ATN(X)

### AUTO

- FORMAT** AUTO [<line number>][, [<increment>]]  
**PURPOSE** Initiates automatic generation of program line numbers.  
**EXAMPLE** AUTO  
AUTO 100, 50

### AUTO START

- FORMAT** AUTO START <auto start string>  
**PURPOSE** Sets the auto start string.  
**EXAMPLE** AUTO START"A:MELODY"+CHR\$(13)

### BEEP

- FORMAT** BEEP [<duration>]  
**PURPOSE** Sounds the PX-8's built-in speaker.  
**EXAMPLE** BEEP ON  
BEEP OFF  
BEEP 100

### CALL

- FORMAT** CALL <variable name> [( <argument list> )]  
**PURPOSE** Starts execution of a machine language subroutine.  
**EXAMPLE** CALL J(X)

### CDBL

- FORMAT** CDBL(X)  
**PURPOSE** Converts numeric expression X to a double Precision number.  
**EXAMPLE** A# = CDBL(X!)

### CHAIN

- FORMAT** CHAIN [MERGE] <file name>[, [<Line number exp>][, ALL][, DELETE<range>]]  
**PURPOSE** Calls the BASIC program designated by <file name > and passes variables to it from the calins program.  
**EXAMPLE** CHAIN "SAMPLE"  
C14AIN "A.SAMPLEZ3" ,,ALL  
CHAIN MERGE "SUB",100

### CHR\$

- FORMAT** CHR\$(J)  
**PURPOSE** Returns the character whose ASCII code equals the value of integer expression j.  
**EXAMPLE** A\$ = CHR\$(65)

## CINT

**FORMAT** CINT(X)

**PURPOSE** Rounds the decimal portion of numeric expression X to the nearest whole number and returns the equivalent integer value.

**EXAMPLE** A% = CINT(5.6)

## CLEAR

**FORMAT** CLEAR[[<dummy>][,<upper memory limit>][,<stack area size>]]

**PURPOSE** Clears all numeric and string variables. When options are specified, also reserves an area in memory for machine language programs and sets the stack area size.

**EXAMPLE** 10 CLEAR ,&HC400

## CLOSE

**FORMAT** CLOSE[[#]<file number>[,[#]<file number ... >]]

**PURPOSE** Terminates access to files.

**EXAMPLE** CLOSE # 3

## CLS

**FORMAT** CLS

**PURPOSE** Clears the currently selected virtual screen.

**EXAMPLE** CLS

## COMMON

**FORMAT** COMMON<list of variables>

**PURPOSE** Passes variables to a CHAINED program.

**EXAMPLE** COMMON B, A\$( )

## CONT

**FORMAT** CONT

**PURPOSE** Resumes execution of a program interrupted by STOP, END or the STOP key.

**EXAMPLE** CONT

## COPY

**FORMAT** COPY

**PURPOSE** OutPuts the contents of the LCD creen to the printer.

**EXAMPLE** COPY

## COS

**FORMAT** COS(X)

**PURPOSE** Returns the cosine of anfle X, where X is in radians

**EXAMPLE** A# = COS(1.570796326794897)

## CSNG(X)

**FORMAT** CSNG(X)

**PURPOSE** Converts numeric expression X to a single precision number

**EXAMPLE** A! = CSNG(16%)

## CSRLIN

**FORMAT** CSRLIN [« function »]

**PURPOSE** Returns the vertical character coordinate of the cursor on the virtual screen or the vertical Position of the first line of the screen window in the vertical screen.

**EXAMPLE** A% = CSRLIN

## CVI/CVS/CVD

**FORMAT** CVI (<2-byte string>

CVS (<4-byte string>

CVD (<8-byte string>

**PURPOSE** Converts ASCII representations of BCD code to numeric values.

**EXAMPLE** PRINT CVI (CHRS(5)+CHR\$(0)

PRINT CVS (CHRS(0) + CHR\$(0)

+CHR\$(32)+CHR\$(131)»

PRINT CVD (CHR\$(0) + CHR\$(0)

+CHR\$(0) + CHR\$(0)

+CHR\$(0) + CHR\$(0)

+CHR\$(32)+CHR\$(131)

## DATA

**FORMAT** DATA < list of constants >

**PURPOSE** Stores numeric and string constants which are substituted into variables by the READ statement.

**EXAMPLE** DATA PX,S,EPSON

## DATE\$

**FORMAT** As a statement

DATE\$ = " <MM>/<DD>/<YY>"

As a variable

X\$ = DATE\$

**PURPOSE** As a statement, sets the date of the PX-8's calendar clock. As a variable, returns the date of the PX-8's built-in clock.

**EXAMPLE** DATE\$ = "01/28/84"  
AS = DATE\$

## DAY

**FORMAT** As a statement

DAY = <W>

As a variable

X% = DAY

**PURPOSE** As a statement, sets the day of the week of the PX-8's built-in clock. As a variable, returns the day of the week from the PX-8's built-in clock.

**EXAMPLE** DAY=6  
A% = DAY

## DEF FN

**FORMAT** DEF FN < name > (<parameter list>)  
= < function definition >

**PURPOSE** defines and names user-written functions.

**EXAMPLE** DEF FNA(X,Y)=X\*3/(Y+2)

## DEFINT/SGN/DBL/STR

**FORMAT** DEF INT <range(s) Of letters>

DEF SNG <range(s) Of letters>

DEF DBL <range(s) Of letters>

DEF STR <range(s) of letters>

**PURPOSE** Declares type of variables as integer, single precision, double precision, or string.

**EXAMPLE** DEFINT I-N, W-Z

## DEF USR

**FORMAT** DEF USR [<digit>]=<integer expression>

**PURPOSE** Used to specify the starting address of user-written machine language subroutines.

**EXAMPLE** DEF USR1=&HC000

## DELETE

**FORMAT** DELETE [<line number 1>][-<line number 2>]

**PURPOSE** Deletes specified lines from a BASIC program.

**EXAMPLE** DELETE 40  
DELETE 40-100  
DELETE -40

## DIM

**FORMAT** DIM <list of subscripted variables>

**PURPOSE** Specifies the maximum range of array subscripts and allocates space for storage of array variables.

**EXAMPLE** DIM G%(25), F%(25)

## DSKF

**FORMAT** DSKF «drive name»

**PURPOSE** Returns the number of kilobytes of free space in specified disk device.

**EXAMPLE** A% = DSKF("A: ")

## EDIT

**FORMAT** EDIT <line number> .

**PURPOSE** Enters the EDIT mode at the specified line.

**EXAMPLE** EDIT 40

## END

**FORMAT** END

**PURPOSE** Terminates program execution, closes all files, and returns BASIC to the command level.

**EXAMPLE** END



## GET

**FORMAT** GET [ #]< file number >[,< record number>]  
**PURPOSE** Reads a record from a random disk file.  
**EXAMPLE** GET # 1, X

## GOSUB ... RETURN

**FORMAT** GOSUB < line number >  
:  
RETURN

**PURPOSE** Used for branching to and returning from subroutines.  
GOSUB < line number >

:  
**EXAMPLE** RETURN

## GOTO or GO TO

**FORMAT** GOTO < line number >  
GO TO <line number>

**PURPOSE** Transfers program execution to the program line specified by <line number>.

**EXAMPLE** GOTO 200

## HEX\$

**FORMAT** HEX\$(X)

**PURPOSE** Returns a character string representing the hexadecimal value of X.

**EXAMPLE** PRINT HEX\$(44323)

## IF..THEN[...ELSE]/IF...GOTO

**FORMAT** IF <logical expression>

THEN	<statement>	[ELSE	<statement>	]
	<line No.>		<line No.>	
GOTO	<line No.>			

**PURPOSE** Changes the flow of program execution according to the result of a logical expression.

**EXAMPLE** IF A=B THEN PRINT "A=B" ELSE  
PRINT "A<>B"

## INKEY\$

**FORMAT** INKEYS

**PURPOSE** Checks the keyboard buffer and returns one character (or a null string if no key has been pressed).

**EXAMPLE** AS=INKEY\$

## INP

**FORMAT** INP (J)

**PURPOSE** Returns one byte of data from machine port J.

**EXAMPLE** A=INP(176)

## INPUT

**FORMAT**

INPUT [;] [<prompt string> ; ] <list of variables>

**PURPOSE** Allows values to be substituted into variables from the keyboard during program execution

**EXAMPLE** INPUT "NAME";AS

## INPUT#

**FORMAT** INPUT # <file number>,<variable list>

**PURPOSE** Reads data into variables from a sequential file.

**EXAMPLE** INPUT# 1, AS, BS, CS

## INPUT\$

**FORMAT** INPUT\$(X[,#]<file number>)

**PURPOSE** Reads a string of X characters from the keyboard buffer or file opened under <file number>.

**EXAMPLE** AS = INPUT\$(1)  
AS = INPUT\$(10, #1)

## INSTR

**FORMAT** INSTR(J,IXS,Y\$)

**PURPOSE** Searches for string Y\$ in string X\$ and returns a number indicating the position at which it was found.

**EXAMPLE** A = INSTR(X\$, "ABC")



## INT

**FORMAT** INT(X)

**PURPOSE** Subtracts the decimal portion of X from X and returns the integer value which is the result.

**EXAMPLE** A=INT(-B/3)

## KEY

**FORMAT** <n>, <X\$>

**PURPOSE** Defines the functions of programmable function keys.

**EXAMPLE** KEY 2, "LIST" + CHR\$(13)

## KEY LIST/KEY LLIST

**FORMAT** KEY LIST  
KEY LLIST

**PURPOSE** Outputs a list of the programmable function key definitions to the display or printer.

**EXAMPLE** KEY LIST

## KILL

**FORMAT** KILL <file descriptor>

**PURPOSE** Deletes the disk device file specified by <file descriptor>.

**EXAMPLE** KILL "FILE3.BAS"  
KILL "D:SAMPLE1.BAS"

## LEFT\$

**FORMAT** LEFT\$(X\$, i)

**PURPOSE** Returns a string of J characters from the left end of string X\$.

**EXAMPLE** A\$=LEM(X\$, 5)

## LEN

**FORMAT** LEN(X\$)

**PURPOSE** Returns the number of characters in string X\$.

**EXAMPLE** A=LEN(XS)

## LET

**FORMAT** [LET] <variable> = <expression>

**PURPOSE** Assigns the value of an expression to a variable.

**EXAMPLE** PI = 3.14159  
LET PI = 3.14159

## LINE

**FORMAT** LINE [[STEP] (X1, Y1)-[STEP](X2, Y2)[, [<function code>][, (B[F])][, <line style>]]]

**PURPOSE** Draws a straight line between two specified points.

**EXAMPLE** LINE (0,0)-(500,300)  
LINE -STEP (20,20)  
LINE (25,25)-(500,200),4,,&HAAA

## LINE INPUT

**FORMAT** LINE INPUT[;J<prompt string>:] <string variable>

**PURPOSE** Assigns character strings entered from the keyboard during program execution to variables.

**EXAMPLE** LINE INPUT "ENTER NAME (LAST, FIRST)";A\$

## LINE INPUT#

**FORMAT** LINE INPUT# <file number>, <string variable>

**PURPOSE** Reads lines of data into variables from a sequential disk file.

**EXAMPLE** LINE INPUT#1, A\$

## LIST

**FORMAT** LIST[\*][<line number>][-<line number>]  
LIST[\*][<file descriptor>][<line number>][-<line number>]

**PURPOSE** Lists BASIC program lines on the display or printer.

**EXAMPLE** LIST  
LIST -50  
LIST 50-  
LIST 50-200  
LIST "LPT0:"

## LLIST

- FORMAT** LLIST [ \*] [< line number >] [-]  
[< line number >]  
**PURPOSE** Outputs a program list to the printer.  
**EXAMPLE** LLIST 1000 - 2000

## LOAD

- FORMAT** LOAD < file descriptor > [,R]  
**PURPOSE** Loads a program into inemory.  
**EXAMPLE** LOAD "LNINPT"  
LOAD "B:LNINPT.BAS"

## LOC

- FORMAT** LOC (<file number >)  
**PURPOSE** Returns the random access file record number following that used by the last GFT or PUT statement, or the number of file sectors read/written since a sequential file was opened.  
**EXAMPLE** A = LOC(1)

## LOCATE

- FORMAT** LOCATE [<X>][, [<Y>][, <cursor switch>]  
**PURPOSE** Moves the cursor to specified virtual screen coordinates.  
**EXAMPLE** LOCATE 1,1,0

## LOF

- FORMAT** LOF (<file number>)  
**PURPOSE** Returns the size of a file in sectors.  
**EXAMPLE** A = LOF(1)

## LOG

- FORMAT** LOG(X)  
**PURPOSE** Returns the natural logarithm of X.  
**EXAMPLE** PRINT LOG(2.7812818)

## LOGIN

- FORMAT** LOGIN < program area no.> [,R]  
**PURPOSE** Switches to the specified BASIC program area. If R is specified, executes the program in that area.  
**EXAMPLE** LOGIN 2  
LOGIN 3,R

## LPOS

- FORMAT** LPOS (X)  
**PURPOSE** Returns the current position of the buffer pointer in the printer output buffer. (X is a dummy argument.)  
**EXAMPLE** A = LPOS(X)

## LPRINT

- FORMAT** LPRINT [<list of expressions>]  
**PURPOSE** Outputs data to a printer connected to the PX-8.  
**EXAMPLE** LPRINT "EPSON PX-8"

## LPRINT USING

- FORMAT** LPRINT USING <format string>;<list of expressions>  
**PURPOSE** Outputs data to a printer in a specific format.  
**EXAMPLE** LPRINT USING "# # # #";A;B

## LSET/RSET

- FORMAT** LSET <string variable> = <string expression >  
RSET <string variable> = <string expression >  
**PURPOSE** Prepares character data for storage in a random access file by moving it into a random file buffer.  
**EXAMPLE** LSET A\$= B\$

## MENU

- FORMAT** MENU  
**PURPOSE** Returns BASIC to the BASIC program.  
**EXAMPLE** MENU

## MERGE

**FORMAT** MERGE <file descriptor >

**PURPOSE** Merges a program from a file (disk device, disk image RAM or COM0:) with the program currently in memory.

**EXAMPLE** MERGE "TEST1"  
MERGE "COM0:"

## MID\$

**FORMAT** As a statement  
MID\$ (<string expl>,n[,m]) = < string exp2>  
As a function  
MID\$ (X\$,J[,K])

**PURPOSE** As a statement, replaces characters from position n in <string expl > with the first m characters of < string exp2 >. As a function, returns K characters from the middle of X\$, starting with character J.

**EXAMPLE** MID\$ (A\$,5,7) = B\$  
A\$ = MID\$("ABCDEFG",3,3)

## MKI\$/MKSS\$/MKD\$

**FORMAT** MKI\$ (<integer expression>)  
MKSS\$ (<single precision expression >)  
MKD\$ (<double precision expression>)

**PURPOSE** Converts numeric values to strings for storage in random access files.

**EXAMPLE** A\$ = MKI\$(X%)  
A\$ = MKSS\$(X!)  
A\$ = MKD\$(X#)

## MOUNT

**FORMAT** MOUNT

**PURPOSE** Reads the microcassette tape directory into memory and prepares the microcassette drive for access as a disk device.

**EXAMPLE** MOUNT

## NAME

**FORMAT** NAME < old file name > AS < new file name >

**PURPOSE** Changes the name of a disk device file.

**EXAMPLE** NAME "SAMPLE1.BAS"  
AS "SAMPLE2.BAS"

## NEW

**FORMAT** NEW

**PURPOSE** Deletes the program in the currently selected memory area and clears all variables.

**EXAMPLE** NEW

## OCT\$

**FORMAT** OCT\$ (X)

**PURPOSE** Returns a character string representing the octal value of X.

**EXAMPLE** A\$ = OCT\$(9999)

## ON ERROR GOTO

**FORMAT** ON ERROR GOTO [<line number>]

**PURPOSE** Causes program execution to branch to the first line of an error handling subroutine when an error occurs.

**EXAMPLE** ON ERROR GOTO 1000

## ON...GOSUB/ON...GOTO

**FORMAT** ON < numeric expression> GOTO  
<list of line numbers>  
ON < numeric expression> GOSUB  
<list of line numbers>

**PURPOSE** Branches to one of several specified program line numbers depending on the value returned for <expression>.

**EXAMPLE** ON A GOSUB 100,200,500,1000

## OPEN

**FORMAT** OPEN "<mode>",<#><file number>,  
<file descriptor>,<record length>

**PURPOSE** Opens a disk file or other device for input or output.

**EXAMPLE** OPEN"O", #1,"CLIENTS.DAT"

## OPTION BASE

**FORMAT** OPTION BASE  $\left| \begin{array}{l} 0 \\ 1 \end{array} \right|$

**PURPOSE** Declares the minimum value for array subscripts.

**EXAMPLE** OPTION BASE 1

## OPTION COUNTRY

- FORMAT** OPTION COUNTRY < character string >  
**PURPOSE** Specifies the international character set to be used for keyboard input/output, LCD display, and output to the printer,  
**EXAMPLE** OPTION COUNTRY "U"  
OPTION COUNTRY "england"

## OPTION CURRENCY

- FORMAT** OPTON CURRENCY <string expression>  
**PURPOSE** Changes the currency symbol.  
**EXAMPLE** OPTION CURRENCY "@"

## OUT

- FORMAT** OUT < integer expression 1 >, < integer expression 2>  
**PURPOSE** Outputs the value of < integer expression 2 > to the machine output port specified in <integer expression 1>.

## PCOPY

- FORMAT** PCOPY < program area no.>  
**PURPOSE** Copies the program in the current program area to another program area.  
**EXAMPLE** PCOPY 3

## PEEK

- FORMAT** PEEK (J)  
**PURPOSE** Returns one byte of data from memory location J.  
**EXAMPLE** A = PEEK (&HE00)

## POINT

- FORMAT** POINT (horizontal position, vertical position)  
**PURPOSE** Returns the setting of the display dot at the specific graphic screen location.  
**EXAMPLE** PRINT POINT (10,10)

## POKE

- FORMAT** POKE < integer expression 1 >, <integer expression 2>  
**PURPOSE** Writes the data byte specified by <integer expression 2 > to the inemory address specified by < integer expression 1 >.  
**EXAMPLE** POKE &HC001,A

## POS

- FORMAT** POS (<file no.>)  
**PURPOSE** Returns the current position of the buffer pointer in the file output buffer.  
**EXAMPLE** PRINT POS (1)

## POWER

- FORMAT** POWER OFF[,RESUME]  
POWER [<duration>]  
POWER CONT  
**PURPOSE** Turns off the power or sets the auto power-off function.  
**EXAMPLE** POWER OFF, RESUME  
POWER 10  
POWER CONT

## PRESET

- FORMAT** PRESET [STEP](X,Y)[,<fonction code>]  
**PURPOSE** Resets the dot at the specified graphic display coordinates,  
**EXAMPLE** PRESET (X,Y)  
PRESET STEP (10,10),I

## PRINT

- FORMAT** PRINT [ < list of expression > ]  
**PURPOSE** Outputs data to the LCD screen  
**EXAMPLE** PRINT "Name is ";A\$  
PRINT X,Y

## PRINT USING

**FORMAT** PRINT USING " < format string > ", < list of expressions >

**PURPOSE** Outputs data to the LCD screen in the format specified in <"format string">.

**EXAMPLE** PRINT USING "\ \";A\$;B\$;C\$  
PRINT USING "####.##";A;B;C

## PRINT#

**FORMAT** PRINT # <file number >,  
[ < list of expressions > ]

**PURPOSE** Writes data to a sequential file.

**EXAMPLE** PRINT # 1,A\$;"",B\$

## PRINT# USING

**FORMAT** PRINT # <file number>,USING <format string>;  
<list of expression>

**PURPOSE** Writes data to a sequential file in a specific format.

**EXAMPLE** PRINT # 1 USING "####.##";A;B;C

## PSET

**FORMAT** PSET [STEP](X,Y)[,<function code>]

**PURPOSE** Sets the dot at the specified graphic coordinates.

**EXAMPLE** PSET (A,B)  
PSET STEP (5, - 5),4

## PUT

**FORMAT** PUT [#] < file number >[,<record number>]

**PURPOSE** Writes a data record to a random access file.

**EXAMPLE** PUT#1,X

## RANDOMIZE

**FORMAT** RANDOMIZE [ < expression > ]

**PURPOSE** Reinitializes the random number generator.

**EXAMPLE** RANDOMIZE  
RANDOMIZE VAL (RIGHT\$(TIMES\$,2))

## REAL

**FORMAT** READ < list of variables >

**PURPOSE** Reads values from DATA statements and substitutes them into variables.

**EXAMPLE** READ A\$,B\$,C\$

## REM

**FORMAT** REM < remark >  
' < remark >

**PURPOSE** Used to insert explanatory remarks into a program.

**EXAMPLE** ' REGRESSION ROUTINE

## REMOVE

**FORMAT** REMOVE

**PURPOSE** Writes the microcassette directory to tape and disables read and write access to the microcassette drive.

**EXAMPLE** REMOVE

## RENUM

**FORMAT** RENUM [[<new line number>],[<old line number>],[<increment>]]

**PURPOSE** Renumbers the lines of programs.

**EXAMPLE** RENUM  
RENUM 300,50  
RENUM 1000,900,20

## RESET

**FORMAT** RESET

**PURPOSE** Resets the READ ONLY condition after a floppy disk in a disk drive has been replaced.

**EXAMPLE** RESET

## RESTORE

**FORMAT** RESTORE [<line number>]

**PURPOSE** Resets the pointer which keeps track of the last item read from DATA statements.

**EXAMPLE** RESTORE  
RESTORE 1000

## RESUME

**FORMAT** RESUME  
RESUME 0  
RESUME NEXT  
RESUME <line number>

**PURPOSE** Used to continue program execution after completion of an error processing routine.

**EXAMPLE** RESUME 100

## RIGHT\$

**FORMAT** RIGHT\$(X\$,J)

**PURPOSE** Returns J characters from the right end of string X\$.

**EXAMPLE** A\$=RIGHT\$("abcdefg",3)

## RND

**FORMAT** RND(X)

**PURPOSE** Returns a random number between 0 and 1

**EXAMPLE** A=RND

## RUN

**FORMAT** RUN [ < line number > ]  
RUN < file descriptor > [,R]

**PURPOSE** Starts execution of a program.

**EXAMPLE** RUN 300  
RUN "B:SAMPLE",R

## SAVE

**FORMAT** SAVE <file descriptor >[,A]  
SAVE <file descriptor >[,P]

**PURPOSE** Saves the program in memory to a disk file or the RS-232C interface.

**EXAMPLE** SAVE"ADDRESS.DAT"  
SAVE"COM0:"

## SCREEN

**FORMAT** SCREEN ( < mode > ]  
[, [< virtual screen > ]  
[, [ < function key switch > ]  
[, [<boundary character>]  
[WIDTH [<no. columns>]  
[, [<no. lines 1 >][,<no. lines 2>]]]]]]

**PURPOSE** Sets the screen mode and screen parameters.

**EXAMPLE** SCREEN 2,1,0,"E" WIDTH 20,20,20

## SCREEN

**FORMAT** SCREEN (<horizontal position >, < vertical position>)

**PURPOSE** Returns the ASCII code corresponding to the character at the specified screen location.

**EXAMPLE** A = SCREEN(5,5)

## SGN

**FORMAT** SGN (X)

**PURPOSE** Returns the sign of X.

**EXAMPLE** A = SGN(X)

## SIN

**FORMAT** SIN(X)

**PURPOSE** Returns the sine of X.

**EXAMPLE** A=SIN(X)

## SOUND

**FORMAT** SOUND <pitch>,<duration>

**PURPOSE** Outputs a tone of the specified pitch and duration from the speaker.

**EXAMPLE** SOUND 1000,100

## SPACES\$

**FORMAT** SPACES(J)

**PURPOSE** Returns a string of spaces of a specified length.

**EXAMPLE** A\$="AAA"+SPACES(10)+ "CCC"

## SPC

**FORMAT** SPC(J)

**PURPOSE** Outputs a string of J spaces to the display or printer.

**EXAMPLE** PRINT SPC(10);A\$

## SQR

**FORMAT** SQR(X)

**PURPOSE** Returns the square root of X.

**EXAMPLE** PRINT SQR(2 #)

## STAT

**FORMAT** STAT 1 < program area no. > 1  
STAT ALL

**PURPOSE** Displays the status of BASIC program areas. STAT

**EXAMPLE** STAT 1  
STAT ALL

## STOP

**FORMAT** STOP

**PURPOSE** Terminates program execution and returns BASIC to the command level.

**EXAMPLE** STOP

## STOP KEY

**FORMAT** STOP KEY | ON |  
                  | OFF |

**PURPOSE** Disables and reenables the **STOP** key.

**EXAMPLE** STOP KEY ON  
STOP KEY OFF

## STR\$

**FORMAT** STR\$(X)

**PURPOSE** Returns a string representation of the value of X.

**EXAMPLE** AS=STR\$(123)

## STRING\$

**FORMAT** STRING\$(J,K)  
STRING\$(J,X\$)

**PURPOSE** Returns a string of J characters.

**EXAMPLE** PRINT STRING\$(10,65)  
PRINT STRING\$(10,"A")

## SWAP

**FORMAT** SWAP < variable 1 >, < variable 2 >

**PURPOSE** Exchanges the values of the variables specified in <variable 1 > and <variable 2 >

**EXAMPLE** SWAP A\$,B\$

## SYSTEM

**FORMAT** SYSTEM

**PURPOSE** Terminates BASIC operation and returns control to the CP/M operatins system.

**EXAMPLE** SYSTEM

## TAB

**FORMAT** TAB(J)

**PURPOSE** Moves the cursor (print head) to character position J on the display screen (print line).

**EXAMPLE** PRINT TAB(10);"ABC"  
LPRINT TAB(10);"ABC"

## TAN

**FORMAT** TAN(X)

**PURPOSE** Returns the tangent of X

**EXAMPLE** A = TAN(3.1416/4)

## TAPCNT

**FORMAT** As a statement  
TAPCNT=J  
As a variable  
J=TAPCNT

**PURPOSE** Sets or reads the PX-8's microcassette drive counter.

**EXAMPLE** TAPCNT=0  
PRINT TAPCNT

## TIMES

**FORMAT** As a statement  
TIMES = " <HH>:<MM>:<SS> "  
As a variable  
X\$ = TIMES

**PURPOSE** Sets or reads the PX-8's built-in clock.

**EXAMPLE** TIMES="15:35:00"  
PRINT TIMES

## TITLE

**FORMAT** TITLE[<program area name>][,P]

**PURPOSE** Sets the name and edit attribute of the currently selected BASIC program area.

**EXAMPLE** TITLE"FINAL",P  
TITLE""

## TRON/TROFF

**FORMAT** TRON  
TROFF

**PURPOSE** Starts or stops the trace mode of program execution.

**EXAMPLE** TRON  
TROFF

## USR

**FORMAT** USR[ <digit > ](argument)

**PURPOSE** Calls a machine language subroutine defined by a DEF USR statement.

**EXAMPLE** A = USR0(B)

## VAL

**FORMAT** VAL(X\$)

**PURPOSE** Converts a string composed of numeric ASCII characters into a numeric value.

**EXAMPLE** A=VAL("123")

## VARPTR

**FORMAT** VARPTR(<variable name>)  
VARPTR(# <file number>)

**PURPOSE** Returns the address of the specified variable or file buffer.

**EXAMPLE** PRINT HEX\$(VARPTR(A))  
PRINT HEX\$(VARPTR(1))

## WAIT

**FORMAT** WAIT <port number >,J[,K]

**PURPOSE** Suspends program execution until a specified bit pattern is developed at the specified machine input port.

## WHILE ...WEND

**FORMAT** WHILE <expression>  
:  
[ <loop statements > ]  
WEND

**PURPOSE** Repeats the series of instructions included between WHILE and WEND as long as the result of the specified expression is TRUE.

**EXAMPLE** WHILE X ≤ 100  
:  
WEND

## WIDTH

**FORMAT** WIDTH [ <no. of columns > ][, [ <no. of lines 1 >]][, <no. of lines 2 >]]  
WIDTH <file descriptor>,<no. of columns>  
WIDTH LPRINT <no. of columns >

**PURPOSE** Sets the column width of the virtual screens or other specified device or file.

**EXAMPLE** WIDTH 20,20,20  
WIDTH "LPT0:",80  
WIDTH #1,80  
WIDTH LPRINT 40

## WIND

**FORMAT** WIND[ | <counter value> | ]  
ON  
OFF

**PURPOSE** Turns the microcassette drive motor on or off, winds the tape until the specified counter value is reached, or rewinds the tape to the beginning and resets the counter to 0.

**EXAMPLE** WIND  
WIND ON  
WIND OFF  
WIND 3000



## **WRITE**

**FORMAT** WRITE [<list of expressions>]

**PURPOSE** Displays data on the LCD screen.

**EXAMPLE** WRITE A\$,B\$,C\$

## **WRITE#**

**FORMAT** WRITE# < file number >,< list of expressions >

**PURPOSE** Writes data to a sequential disk file using the format of the WRITE statement.

**EXAMPLE** WRITE#1,A\$,B\$