

# Pocket Computer

## Model PC-2

### Quick Reference Card

Catalog Number 26-3601

**Note:** Shaded information applied to the PC-2 Printer/  
Cassette Interface (26-3605) only.

----	----	----	----
----	----	----	----
----	----	----	----
0	1	2	3
----	----	----	----
----	----	----	----
----	----	----	----
4	5	6	7
----	----	----	----
----	----	----	----
8	9	A	B
----	----	----	----
----	----	----	----
----	----	----	----
C	D	E	F

For example, to display a solid line, type:

GPRINT "7F"

RADIO SHACK, A DIVISION OF TANDY CORPORATION

U.S.A. FORT WORTH, TEXAS 76102  
CANADA BARRIE, ONTARIO L4M 4W5

TANDY CORPORATION

AUSTRALIA  
280 316 VICTORIA ROAD  
RYDALMERE, N.S.W. 2116

BELGIUM  
PARC INDUSTRIEL DE NANINNE  
5140 NANINNE

U.K.  
BILSTON ROAD WEDNESBURY  
WEST MIDLANDS WS10 7JN

# Statements and Functions

Argument ranges are indicated below by special letters:

**x**: numeric expression between +2047 and -2048

**y**: numeric expression between +2047 and -2048

**n**: (-9.999999999 E-99, +9.999999999 E99)

**c**: (0-255)

**str**: string argument

**var**: variable name

**ABS n** Computes absolute value. Abbreviations: AB.  
Y = ABS X

**ACS n** Computes arccosine. Abbreviations: AC.  
A = ACS .102

**AND** Logical comparison.  
IF X= 9 AND Y= 14 THEN 100

**AREAD var** Assigns Display contents to variable.  
Must be used with a Definable Key label. Abbreviations: A, AR, ARE, AREA.  
"A" AREAD X

**ARUN** Automatic execution on power-up. Must be the first instruction in the first line in a program. Abbreviations: ARU.  
ARUN ARU.

**ASC str** Returns ASCII code of first character in string.  
A = ASC "ASHER"

**ASN n** Computes arcsine. Abbreviations: AS.  
A = ASN (X/3)

**ATN n** Computes arctangent. Abbreviations: AT.  
Y = ATN 45

**BEEP switch** Turns tone feature on or off.  
BEEP ON BEEP OFF

**BEEP number, frequency, duration** Sounds tone for specified *number* (0-65535) of times, *frequency* (0-255), and *duration* (0-65279) of each tone. Abbreviations: B.  
BEEP 25,45,14 B.150

**CALL address, var** Calls the machine-language routine stored at specified entry address and will use specified optional variable values to execute. Abbreviations: CA, CAL.  
CALL &2000 CALL 35423.A

**CHAIN -1 "filename", line number** Allows execution of a segmented program. Program statement only. Abbreviations: CHA, CHAI.  
100:CHAIN"PR0G1",30  
100:CHAIN"PR0G2",A

**CHR\$ c** Converts ASCII decimal code form 0 through 127 to equivalent character string. Abbreviations: CH, CHR.  
P#=CHR\$ T

**CLEAR** Clears all data, resets variables to zero, and erases DIMensioned arrays. Strings are not set to zero but to null. Abbreviations: CL, CLE, CLEA, CLEAR

**CLOAD -1 "filename"** Loads program from cassette. Abbreviations: CLO, CLOA.  
CLOAD CLOAD"PR0G"

**CLOAD? -1 "filename"** Compares program on cassette to resident program. Abbreviations: CLO.? CLOA.?  
CLOAD? CLOAD?"PR0G"

**CLS** Erases Display.  
CLS

**COLOR pen** Rotates pen holder to specified pen position (0-3). Abbreviations: COL, COLO.  
COLOR3

**CONT** Continues execution after BREAK or STOP. Abbreviations: C, CO, CON.  
CONT

**COS n** Computes cosine.  
Y = COS X

**CSAVE -1 "filename"** Saves program on cassette. Abbreviations: CS, CSA, CSAV.  
CSAVE "PR0G"

**CSIZE size** Sets the printing character size (1-9). Abbreviations: CSI, CSIZ.  
CSIZE 3

**CURSOR position** Display printing will start at one of the 26 (0-25) Display positions specified by *position*. Abbreviations: CU, CUR, CURS, CURSO.  
CURSOR 13 CUR. 20

**DATA expression** Stores data to be accessed by a READ statement. Abbreviations: DA, DAT.  
DATA "LINCOLN, A.",1861,"ILLINDIS"

**DEG** Converts degree, minutes, and seconds to decimal degrees.  
DEG 32.2513

**DEGREE** Sets Angle Calculation to degrees. Abbreviations: DE, DEG, DEGR, DEGRE, DEGREE DEG.

**DIM** Dimensions one or more arrays. Abbreviations: D, DI.  
DIM X\$(4) DIM X(5)

**DMS** Converts decimal degrees into degrees, minutes, and seconds. Abbreviations: DM, DMS 32.2513

**EXP n** Computes natural antilog. ( $e^n$ ) Abbreviations: EX.  
Y = EXP X

**END** Ends program execution. Abbreviations: E, EN, END

**FOR . . TO . . STEP/NEXT** Opens program loop. Counter variable range -32768 through +32767. Abbreviations: F, FO, STE, N, NE, NEX.  
FOR I = 1 TO 100 STEP 10...NEXT I

**GCURSOR position** Selects Display start position (0-155). Abbreviations: GCU, GCUR, GCURS, GCURSOR 100 GCUR. 75

**GLCURSOR (x, y)** Positions Printer pen. Abbreviations: GL, GLC, GLCU, GLCUR, GLCURS, GLCURSO, GLCURSOR(100,100)

**GOTO** Transfers program control to the specified line. Abbreviations: G, GO, GOT.  
GOTO 180 GOTO A GO. "A" GOT. A#

**GOSUB** Transfers program control to the specified line. Abbreviations: GOS, GOSU.  
GOSUB 750 GOS. A GOSU. "A" GOSUB A#

**GPRINT pattern delimiter pattern delimiter . . .** Sets graphic dots on the Display. *pattern* is between 0-127; *delimiter* is a comma (blank space) or semi-colon (no space). Abbreviations: GP, GPR, GPRI, GPRINT 10,100 GPRI. &F:&2C GP."7F4949497F"

**GRAD** Sets Angle Calculation to gradients.  
GRAD

**GRAPH** Enters GRAPH Mode either immediately or from a program line. Abbreviation: GRAP.  
GRAPH 10:GRAPH

**IF ... THEN** Tests conditional expression.

Abbreviations: T. TH. THE.

IF P = Q THEN 200

**INKEY\$** Gets keyboard character if available.

Abbreviations: INK. INKE. INKEY.

A\$=INKEY\$

**INPUT** Inputs data from keyboard. Abbreviations: I.

IN. INP. INPU.

INPUT "WHAT IS THE SCORE":S

IN. "NAME":N\$

**INPUT# (-1) "filename", variable, ...** Transfers data from cassette to memory. Abbreviations: I.# IN.#

INP.# INPU.#

IN.#, "PROG" 100:INPUT#"PROG",A,B

**INT n** Returns largest whole number not greater than n.

Y = INT X

**LCURSOR position** Positions pen. (TEXT Mode only.) Abbreviations: LC. LCU. LCUR. LCURS.

LCURSO.

LCURSOR 6

**LEFT\$ str, c** Returns left portion of string.

Abbreviations: LEF. LEFT.

A\$=LEFT\$ "BENJAMIN",3

**LEN str** Returns the number of characters in a string.

X=LEN SEN\$

**LET** Assigns value to variable (optional).

Abbreviations: LE.

LET X = 10

**LF length** Line feed. Reverse paper movement cannot be greater than 10.24 cm.

LF 8 LF A\$

**LINE (x1, y1)-(x2, y2)-... (x7, x7), line style, color,**

**B** Draws a line (or lines). (x1, y1) is optional; if omitted the current startpoint is used.

Abbreviation: LIN.

LINE(50,50)-(100,100),3,1,B

LINE -(50,75)-(125,100),1,1

**LIST line** Lists first program line or specified line.

Abbreviations: L. LI. LIS.

LIST 100 L.75 L.

**LLIST startline, endline** Lists program lines to Printer. Abbreviation: LL. LLI. LLIS.

LLIST LLIST600 LLIST600,900

**LN n** Computes natural logarithm (base e).

Y=LN X

**LOCK** Locks the PC-2 in current Operation Mode.

Abbreviations: LOC.

LOCK LOC.

**LOG n** Computes logarithm to base 10.

Abbreviations: LO.

Y=LOG X

**LPRINT item delimiter item delimiter ...** Prints an item. Abbreviation: LP. LPR. LPRI. LPRIN.

LPRINT"FF",66 LPRINT A\$:B:C

**MEM** Finds amount of free memory.

Abbreviations: M. ME.

MEM M.

**MERGE -1 "filename"** Merges (appends) cassette program with resident program.

Abbreviations: MER. MERG.

MERGE"PROG"

**MID\$ (str, position, length)** Takes a character(s) beginning at position of the specified string.

Abbreviations: MI. MID.

PRINT MID\$(A\$,3,3)

**NEW** Erases current program from memory and clears variables.

NEW

**NEW0** Erases current program and resets the Computer.

NEW0

**NOT** Logical comparison.

IF NOT THEN 400

**ON ERROR GOTO** Sets up an error-handling routine.

Abbreviations: O. ER. ERR. ERRO.

ON ERROR GOTO 210

**ON ... GOSUB** Multi-way branch to specified sub-routines. Abbreviations: O. GOS. GOSU.

ON Y GOSUB 50, 100, 150, 200

**ON ... GOTO** Multi-way branch to specified lines.

Abbreviations: O. G. GO. GOT.

ON X GOTO 190, 200, 210

**OR** Logical comparison.

IF X < > 10 OR Y < > 10 THEN 200

**PAUSE** Print message and continue execution.

Abbreviations: PA. PAU. PAUS.

PAUSE

**PEEK address** Gets value in specified address from Primary memory buffer.

PEEK 34223 PEEK &2C00

**PEEK# address** Gets value in specified address from Alternate memory buffer.

Abbreviations: PE. PEE.

PEEK# 53990 PEEK# &A45

**PI  $\pi$**  Returns value of PI (3.141592654).

A=PI/3

**POINT position** Test dot pattern of specified column (0-155) on Display. Abbreviations: POI. POIN.

A = POINT 100

**POKE address, value** Puts value (0-255) into specified memory address (0-65535) of Primary memory buffer.

POKE 3422,22

**POKE# address, value** Puts value (0-255) into specified memory address (0-65535) of Alternate memory buffer. Abbreviations: PO. POK.

POKE# 3422,22 PO. &5C.81

**PRINT# (-1) "filename"; variable, ...** Transfers data from memory to cassette. Abbreviations: P.#

PR.# PRI.#

10:PR.#"PROG" P.#"PROG",A,B,C

PRI.# PRINT# PRINT#"PROG"

**PRINT** Prints an item or list of items on the Display.

Abbreviations: P. PR. PRI. PRIN.

PRINT A\$ PR. 100 PRI. A

**PRINT USING** Formats strings and numbers for printing. The width of a numeric field must always be one more than the width of the data.

# Formats numbers.

PRINT USING "####";66.2

\* Specifies Asterisk Fill of the specified positions of a numbering field which do not contain data.

PRINT USING "\*\*\*\*#";Y

• Decimal point.

PRINT USING "####.###";58.76

, Displays a comma to the left of every third digit left of the decimal point. An extra # is required for each comma.

PRINT USING "#####.###";246813

^ Exponential format. Displays numbers in scientific notation.

PRINT USING "###.^";3.14

+ Prints a + sign in the first position if the specified number is positive number; a minus sign (-) if the specified number is negative. (0 is assumed to be positive.)

```
PRINT USING "+###";66.2
PRINT USING "-###";-74.1
```

& Specifies a character field.

```
PRINT USING "#####";"JACKSON"
```

**RADIAN** Sets Angle Calculation to radians.

Abbreviations: RAD. RADI. RADIA.  
RADIAN RAD.

**RANDOM** Reseeds the random number generator.

Abbreviations: RA. RAN. RAND. RANDO.  
RANDOM

**READ** Reads value(s) from a DATA statement.

Abbreviations: REA.  
READ T READ T\$ REA. NM\$,AGE

**:REM** Remark; instructs the PC-2 to ignore the rest of the line.

:REM

**RESTORE** Resets data pointer to the first item in the data line. Abbreviations: RES. REST. RESTO.

```
RESTORE RESTORE 100 RESTORE "A"
```

**RETURN** Returns from the subroutine to the next statement after GOSUB. Abbreviations: RE. RET.

```
RETU. RETUR.
RETURN
```

**RIGHT\$ (str, length)** Returns the right portion of string. Abbreviations: RI. RIG. RIGH. RIGHT.

```
ZIP$=RIGHT$(AD$,5)
```

**RLINE (x1, y1)-(x2, y2)- . . . (x7, y7), line style, color, B** Draws line from relative origin. (x1,y1) is optional; if omitted, the current start point is used. Abbreviation: RL.

```
RLINE (50,50)-(100,100),1,1,B
RLINE-(200,200)
```

**RMT switch** Disables/enables remote switch. (REM 1 only.) Abbreviations: RM. OF. RMT OF. RM.O.

```
RMT OFF RMT ON
```

**RND n** Generates a pseudo-random number between 1 and n if n > 1, or between 0 and 1 if n = 0.

Abbreviations: RN.  
Y=RND(100)

**ROTATE direction** Sets printing *direction* (0-3) on printer.

```
ROTATE2
```

**RUN** Executes program from beginning or from a specified line number or program label.

Abbreviations: R. RU.

```
RUN R.100
```

**SGN n** Returns a value indicating sign of number: -1, 0, 1, if n is negative, zero, positive.

Abbreviations: SG.  
X = SGN(A\*B)

**SIN n** Computes sine. Abbreviations: SI.

```
Y = SIN X
```

**SORGN** Sets origin on Printer.

Abbreviations: SO. SOR. SORG.  
SORGN

**SQR n**  $\sqrt{\quad}$  Computes square root.

Abbreviations: SQ.  
Y = SQR(A+B)

**STATUS number** Checks the current memory status. number returns 0 = program steps available; 1 = program steps used; 2 = address + 1 of location of end of program; 3 = address of the end of variable storage area; 4 = line number of program when execution was halted. Abbreviations: STA. STAT.

```
STATU.
STATUS0 STATUS 1
```

**STOP** Stops program execution. Abbreviations: S. ST. STO.

```
STOP
```

**STR\$ n** Converts a numeric expression to a string. Abbreviations: STR.

```
S$ = STR$ X
```

**TAB position** Positions pen.

```
LPRINT TAB 6;7 LPRINT TAB 5;"PROGRAM"
```

**TAN n** Computes tangent. Abbreviations: TA.

```
X = TAN Y
```

**TEST** Printer self-test. Abbreviation: TE. TES.

```
TEST
```

**TEXT** Enters Printer TEXT Mode. Abbreviation:

```
TEX.
TEXT 10:TEXT
```

**TIME month day hour . minutes seconds**

Sets or returns the current time in 24-hour time.

Abbreviations: TI. TIM.

```
TIME=123014.3030 TIME TIM.
```

**TROFF** Turn program trace off.

Abbreviations: TROF.  
TROFF TROF.

**TRON** Turn program trace on. Abbreviations: TR.

```
TRO.
TRON TRO.
```

**UNLOCK** Unlocks the PC-2 operation Mode after it has been locked. Abbreviations: UN. UNL. UNLO.

```
UNLOC.
UNLOCK UN.
```

**VAL str** Converts a string to a number.

Abbreviations: V. VA.  
U=VAL "100 DOLLARS"

**WAIT delay** Specifies duration of PRINT. *delay* can be from 0 to 65535. If *delay* is not specified, you must press **[ENTER]** to go to next program step.

Abbreviations: W. WA. WAI.  
WAIT 100 W.500 WA. 10 WAI.

## Error Codes

Error Code	Explanation
1	Syntax Error. Incorrectly typed statement.
2	NEXT statement without a FOR.
4	READ statement without a DATA.
5	Array variable already exists.
6	Array specified without first DIMensioning it.
7	Illegal variable name.
8	DIMensioned array has more than two levels.
9	Array subscript exceeds size of array specified in DIM statement.
10	Out of memory.
11	Program line does not exist.
12	Incorrect format for PRINT USING statement.
13	Program exceeds program-memory capacity or Function Key specification exceeds Function Key memory capacity.
14	Buffer space exceeded or FOR statement nested too deeply and stack capacity has been exceeded.

Error Code	Explanation
15	GOSUB nested to deeply and stack area has been exceeded or the string buffer size has been exceeded by the character strings while parsing an expression.
16	Specified value is greater than 1 E100 or less than -1 E-100 or the hexadecimal value is greater than 65535 decimal.
17	Data type is inappropriate for calculation expression.
18	Number of arguments is inappropriate for expression.
19	Specified numeric value is outside permitted range.
20	There is not a left parentheses following @ or @\$ when fixed memory array variables were specified.
21	Required variable is not in the expression.
22	There is not enough memory available to load the program that is loading.
23	TIME is incorrectly typed in.
26	Command cannot be executed in the current mode.
27	There is not a program which corresponds to the specified label.
28	INPUT or AREAD statements have been used as variables or a command has been inserted within quotation marks.
30	Line number is greater than 65535.
32	Graphic cursor is between Columns 152-155 during execution of input commands. The input code cannot be displayed.
34	Specified optional device is not attached.
35	The optional device specified in the PRINT# or INPUT# expression is not consistent or the specified optional device cannot handle input/output commands according to the given syntax.
36	Inappropriate PRINT USING format.
37	Calculation results are greater than 9.999999999 E99
38	Division by zero.
39	An illogical calculation has been attempted.
40	Inappropriate specification for expression.
41	CSAVE and CLOAD have been specified for the ROM area.
42	Cassette file data is too large for memory.
43	Data which is being verified with CLOAD? does not match file format.

Error Code	Explanation
44	Checksum error.
70	Pen has exceeded or reached the limit of the coordinate range (-2048, +2047).
71	Paper has receeded or reached the reverse line feed limit of 10.24 cm.
72	Value given is inappropriate for the value of TAB or LCURSOR.
73	Wrong Printer Mode (GRAPH or TEXT) for command issued.
74	Number of commas in LINE or RLINE is too large.
76	Results cannot be printed on a single line in TEXT Mode with LPRINT.
78	Pens are in the process of being changed or a low battery condition exists.
79	Color signal has not been given.
80	Low battery.
177-181	Program has overwritten the data area.
224-241	Incorrect input data during the execution of an INPUT or AREAD command.

**CHECK 6**  
NEW0? : CHECK6 Printer is not fully charged.

## Operators

^	Exponentiation
-, +	Unary negative, positive
*, /	Multiplication, division
+, -	Addition and concatenation, subtraction
<, =, >, <=, >=, <>	Relational tests
NOT	
AND	
OR	

## Order of Mathematical Operations

- ( ) Any values enclosed in parentheses will be evaluated first.
- Retrieval of values from variables (PI, MEM, TIME, etc.) are considered.
- Trigonometric functions (such as SIN, COS, TAN, etc.) will be evaluated next.
- Exponentiation (^) is evaluated next.
- Logical Arithmetic operations (+, -) are performed next.
- Multiplication (\*) and Division (/) are next.
- Addition and subtraction is then performed.
- Comparison operators (<, >, =, >=, <=, <>) are evaluated.
- Logical operators (AND, OR, NOT) are the last to be evaluated.

Operations of equal precedence are evaluated left to right.

## Graphic Characters

Use GPRINT with the appropriate two-digit hexadecimal codes to generate the following graphic characters.

The seven rows are divided into a lower group of three rows and an upper group of four rows. Each group is numbered, from top to bottom, by powers of two.

Because the lower group has only three rows, the range of allowable digits for this group will be from 0-7.

Of the two hexadecimal digits required, the first digit will represent the lower group and the second digit will represent the upper group.

Note that the lower group must be specified before the upper group.