Canon

X-07

HAND HELD COMPUTER

BASIC REFERENCE CARD

BEEP

Function: Outputs sound from the speaker.
Format: BEEP <tone>, <duration>
Example: BEEP 100, 2

CIRCLE

Function: Draws a circle.
Format: CIRCLE [STEP] (<x coordinate>, <y coordinate>), <radius>
Example: CIRCLE (98, 16), 8

CLEAR

Function: Initializes variables, and defines the memory area to be used.
Format: CLEAR [<string area> [, <upper limit of user area>]]
Example: CLEAR 100, 500

CLOAD

Function: Loads a file from a cassette tape.
Format: CLOAD ["<file name>"]
Example: CLOAD "PROG1"

CLOAD?

Function: Confirms a file on a cassette tape.
Format: CLOAD? ["<file name>"]
Example: CLOAD? "PROG1"

CLS

Function: Clears the LCD.
Format: CLS
Example: CLS

CONSOLE

Function: Sets the status of the console.
Format: CONSOLE [<scroll start line>] [, [<number of scroll lines>] [, [<key display switch>] [, [<click sound switch>] [, [<key repeat switch>]]]]]
Example: CONSOLE 1, 2, 1, 0, 1

CONSOLE @

Function: Turns on or off the alarm speaker, initializes the user defined key characters and sets the keyboard mode.
Format: CONSOLE @ [[<alarm switch>] [, [<definition>], [, <key≠mode>]]]
Example: CONSOLE @ ,,3

CONT

Function: Resumes program execution after it has been stopped.
Format: CONT
Example: CONT

CSAVE

Function: Saves the program in the text area of memory onto a cassette tape.
Format: CSAVE "<file name>"
Example: CSAVE "PROG1"

DATA

Function: Specifys the values to be read by the READ statement.
Format: DATA <constant> [,< constant> ...]
Example: DATA 1, "X-07", "CANON COMPUTER", 3

DEFFN

Function: Defines a function.
Format: DEFFN <name> [(<argument> [, <argument> ...])] = <function definition>
Example: DEFFNZ (X, Y) = X*2 + Y*3 + 6

DEFINT/SGN/DBL/STR

Function: Declares variable types.
Format: DEF <type> <range(s) of letters>
Example: DEFINT I-N, X

DELETE

Function: Deletes a file from the RAM file.
Format: DELETE "<file name>" [, "<file type>"]
Example: DELETE "SAMPLE", "P"

DIM

Function: Declares an array or arrays.
Format: DIM <variable name> (<maximum value of subscript> [, <maximum value of subscript> ...]) [, <variable name> (<maximum value of subscript> [, <maximum value of subscript> ...]) ...
Example: DIM A$ (100), B (18, 16)

DIR

Function: Displays the contents of the RAM file.
Format: DIR [# <file number>]
Example: DIR # 1

END

Function: Terminates program execution.
Format: END
Example: END

ERASE

Function: Erases an array or arrays.
Format: ERASE <array variable name> [, <array variable name> ...]
Example: ERASE A, B$

ERROR

Function: Simulates the occurence of an error.
Format: ERROR <error number>
Example: ERROR 200

EXEC

Function: Executes a machine language program.
Format: EXEC <start address>
Example: EXEC &H3000

FOR TO STEP ~ NEXT

Function: Repeatedly executes a series of instructions.
Format: FOR <variable> = <initial value> TO <final value> [STEP <increment>]

•
•

NEXT
Example: FOR I = 1 TO 10: NEXT

FSET

Function: Sets the size of the RAM file.
Format: FSET <file area size>
Example: FSET 1024

GOSUB ~ RETURN

Function: Executes a subroutine.
Format: GOSUB <line number>
Example: GOSUB 1000

GOTO

Function: Changes the flow of program execution.
Format: GOTO <line number>
Example: GOTO 2000

IF THEN ELSE

Function: Decides the condition specified by a logical expression.
Format: IF <logical expression> | THEN | <statement> | | <line number> | | GOTO <line number> | [ELSE | <statement> | <line number> |]
Format: IF A = 0 THEN 100 ELSE A = 0:GOTO 200

INIT

Function: Sets a device ready for operation.
Format: INIT # <file number>, "<file descriptor>"[, <file size> | <baud rate> | [, "<file type>" | "<mode>" |]]
Example: INIT #2, "FILE 1", 100, "D"

INPUT

Function: Inputs data to variables.
Format: INPUT ["<prompt string>";] <variable> [, <variable> ...]
Example: INPUT "YOUR NAME"; N$

INPUT

Function: Reads data from a file, and assigns them to variables.
Format: INPUT # <file number>, <variable> [, <variable> ...]
Example: INPUT #1, A, B$

LET

Function: Assigns a value to a variable.
Format: [LET] <variable> = <expression>
Example: LET P1 = 3.1415926545898

LINE

Function: Draws a line.
Format: LINE [STEP] [(<X coordinate>, <Y coordinate>)]-[STEP] (<x coordinate>, <y coordinate>)
Example: LINE (0, 0)-(119,31)

LINE INPUT

Function: Inputs an entire line.
Format: LINE INPUT ["<prompt string>";] <string variable>
Example: LINE INPUT "WHAT IS YOUR NAME"; N$

LINE INPUT

Function: Inputs an entire line from a file.
Format: LINE INPUT # <file number>, <string variable>
Example: LINE INPUT #1, N$

LIST

Function: Displays a list of the program in the text area of memory.
Format: LIST [<start line number>] [-[<final line number>]]
Example: LIST 100-200

LIST @

Function: Displays, line by line, the list of the program in the text area of memory.
Format: LIST @[<start line number>][-[<final line number>]]
Example: LIST @100-200

LIST

Function: Outputs list of the program in the text area of memory to a file.
Format: LIST [@]# <file number>, [<start line number>] [-[<final line number>]]
Example: LIST #1, 100-

LLIST

Function: Outputs list of the program in the text area of memory to a printer.
Format: LLIST [<start line number>] [-[<final line number>]]
Example: LLIST

LOAD

Function: Loads a program from a file into the text area of memory.
Format: Load "<file descriptor>"[, | <file size> | <baud rate> |
[, | "<file type>" |]]
"<mode>"
Example: LOAD "OPT:", 2400, "B"

LOAD?

Function: Checks the contents of a program file.
Format: LOAD? "<file descriptor>"[, | <file size> | <baud rate> |
[, | "<file type>" |]]
"<mode>"
Example: LOAD? "COM:"

LOCATE

Function: Moves the cursor.
Format: LOCATE <horizontal position>, <vertical position>
Example: LOCATE 10, 0

LPRINT

Function: Outputs data to the graphic printer.
Format: LPRINT [{[<character size>][, <color>]}] [,] [USING "<format string>";] [<expression> [, <expression> ...]]
Example: LPRINT [N, 0] "CANON"
LPRINT [2], USING "# # #"; A, B
LPRINT U$

MOTOR

Function: Turns on the motor of a cassette recorder.
Format: MOTOR [| ON |]
| OFF |
Example: MOTOR ON

NEW

Function: Erases the program in the text area of memory.
Format: NEW
Example: NEW

NEXT

Function: Indicates the end of a loop.
Format: NEXT [<variable> [,<variable> ...]]
Example: NEXT I, J

OFF

Function: Turns off the power of the X-07.
Format: OFF [1 | 2]
Example: OFF

ON ERROR GOTO

Function: Defines branch destination line in case of error occurence.
Format: ON ERRO GOTO <line number>
Example: ON ERROR GOTO 100

ON ~ GOSUB/ON ~ GOTO

Function: Branches to one of several lines depending on the value of an expression.
Format: ON <expression> GOSUB | GOTO <line number$_1$> [, <line number$_2$> ...]
Example: ON I GOSUB 100, 130, 230, 250

OUT

Function: Outputs data to the output port.
Format: OUT <port address>, <expression>
Example: OUT &H7F, &H84

OUT

Function: Outputs a value to a file.
Format: OUT # <file number>, <expression>
Example: OUT #1, 18

POKE

Function: Writes a numeric value into memory.
Format: POKE <address>, <expression>
Example: POKE &H0C00, &HC9

PRESET

Function: Erases a dot.
Format: PRESET [STEP] (<x coordinate>, <y coordinate>)
Example: PRESET (40, 25)

PRINT

Function: Outputs data to the LCD.
Format: PRINT [<expression> [, | ; <expression> ...]
Example: PRINT "CA"; "NO", "N", 3•2
? "CANON"

PRINT USING

Function: Outputs string or numeric constants according to a specified format.
Format: PRINT USING "<format string>"; [<expression> [, <expression>...]
Example: PRINT USING "# # # "; A

PRINT

Function: Outputs data to a file.
Format: PRINT # <file number>, [USING "<format string>";] [<expression> [, | ; <expression> ...]]
Example: PRINT # 1, "CANON"

PSET

Function: Draws a dot.
Format: PSET [STEP] (<x coordinate>, <y coordinate>)
Example: PSET (40, 25)

READ

Function: Reads constants from a DATA statement and assigns them to variables.
Function: READ <variable> [, <varaible> ...]
Example: READ A, I, M$

REM

Function: Inserts remarks.
Format: REM [<remark>]
Example: REM ::**COMP**::
'CANON

RESTORE

Function: Reuses a set of data.
Format: RESTORE [<line number>]
Example: RESTORE 1000

RESUME

Function: Resumes program execution after an error recovery procedure.
Format: RESUME [| NEXT |]
| <line number> |
Example: RESUME 100

RETURN

Function: Indicates the end of a subroutine.
Format: RETURN
Example: RETURN

RUN

Function: Executes the program in the text area of memory.
Format: RUN [<line number>]
Example: RUN 1000

RUN "file descriptor"

Function: Executes a program in the RAM file.
Format: RUN "[RAM:] <file name>" [, <file type>"]
Example: RUN "RAM: FILE1"

SAVE

Function: Saves the program in the text area of memory into a file.
Format: SAVE "<file descriptor>" [, | <file size> |
| <baud rate> |
[, | "<file type>" |]]
| "<mode>" |
Example: SAVE "OPT:", 1200 "B"

SLEEP

Function: Retains displays, variables, etc. when the power of the X-07 is turned off.
Format: SLEEP
Example: SLEEP

STOP

Function: Terminates program execution.
Format: STOP
Example: STOP

TROFF

Function: Terminates the trace mode.
Format: TROFF
Example: TROFF

TRON

Function: Displays the line being executed.
Format: TRON [# <file number>]
Example: TRON #1

Functions

ABS

Function: Returns the absolute value of a numeric expression.
Format: ABS (<numeric expression>)
Example: PRINT ABS (−1.8)

ALM$

Function: Sets the alarm.
Format: ALM$ = "[<year>], [<month>], [<day>], [<day of week>], [<hours>], [<minutes>]"
Example: ALM$ = "1983//, &H13, 8: 15"

ASC

Function: Returns the character code of the first character of a string expression.
Format: ASC (<string expression>)
Example: PRINT ASC ("A")

ATN

Function: Returns the arctangent value of a numeric expression in radians.
Format: ATN (<numeric expression>)
Example: A = ATN (0.5)

CDBL

Function: Converts the value of a numeric expression into a double-precision real number-type constant.
Format: CDBL (<numeric expression>)
Example: A# = CDBL (B!)

CHR$

Function: Returns the character specified by a character code.
Format: CHR$ (<character code>)
Example: PRINT CHR$ (&H41)

CINT

Function: Converts the value of a numeric expression into an integer-type constant.
Format: CINT (<numeric expression>)
Example: A% = CINT (B#)

COS

Function: Returns the consine value of a numeric expression in radians.
Format: COS (<numeric expression>)
Example: A = COS (3.1415926535898/3)

CSNG

Function: Converts the value of a numeric expression into a single-precision real number-type constant.
Format: CSNG (<numeric expression>)
Example: A! = CSNG (B#)

CSRLIN

Function: Returns the vertical position of the cursor on the LCD.
Format: CSRLIN
Example: Y = CSRLIN

DATE$

Function: Sets the calendar.
Format: DATE$ = "[<year>]/[<month>]/[<day>]"
Example: DATE$ = "1983/8/22"

ERL

Function: Returns the line number of the program line in which an error occurred.
Format: ERL
Example: L = ERL

ERR

Function: Returns the error code of the cause of an error.
Format: ERR
Example: C = ERR

EXP

Function: Returns the exponent of the value of a numeric expression.
Format: EXP (<numeric expression>)
Example: E = EXP (1)

FIX

Function: Returns the integer part of a numeric expression.
Format: FIX (<numeric expression>)
Example: A = FIX (B)

FONT$

Function: Specifys an user defined character.
Format: FONT$ (<character code>) = "<variable$_1$>,..., <variable$_8$>"
Example: FONT$ (128) = "56, 56, 16, 124, 144, 40, 68, 132"

FRE

Function: Returns the unused area of memory in number of bytes.
Format: FRE (<expression>)
Example: PRINT FRE (A$)

HEX$

Function: Returns the string which represents the hexadecimal value of a numeric expression.
Format: HEX$ (<numeric expression>)
Example: PRINT HEX$ (255)

INKEY$

Function: Returns the character of the key being depressed.
Format: INKEY$
Example: A$ = INKEY$

INP

Function: Returns data from a file or a port address.
Format: INP (# <file name>)
INP (<port address>)
Example: A = INP (&H84)

INSTR

Function: Searches for occurence of string 2 in string 1, and when it is found, returns the position of occurence.
Format: INSTR ([<numeric expression>,] <string 1>, <string 2>)
Example: N = INSTR (B$, "N")

Functions

INT

Function: Returns the largest integer which is not greater than the numeric expression.
Format: INT (<numeric expression>)
Example: A = INT (B)

KEY$

Function: Defines the contents of a user defined key.
Format: KEY$ (<key number>) = "<string expression>"
Example: KEY$ (3) = "RUNRUN" + CHR$ (13)

LEFT$

Function: Returns a string consisting of left most characters from a string.
Format: LEFT$ (<string>, <length>)
Example: PRINT LEFT$ ("CANON", 3)

LEN

Function: Returns the length of a string.
Format: LEN (<string>)
Example: N = LEN (A$)

LOG

Function: Returns the natural logarithm of a numeric expression.
Format: LOG (<expression>)
Example: A = LOG (B)

MID$

Function: Returns a string consisting of specifed length of characters from a string.
Format: MID$ (<string>, <expression 1> [, <expression 2>])
Example: A$ = MID$ (B$, 3, 2)

PEEK

Function: Returns the contents of memory.
Function: PEEK (address)
Example: A = PEEK (&H0)

POINT

Function: Checks for a dot.
Format: POINT [STEP] (<x coordinate>, <y coordinate>)
Example: A = POINT (110, 10)

POS

Function: Returns the horizontal position of the cursor.
Format: POS (<expression>)
Example: X = POS (0)

RIGHT$

Function: Returns a string of specifed length, consisting of right most characters of a string.
Format: RIGHT$ (<string>, <length>)
Example: PRINT RIGHT$ ("CANON", 3)

RND

Function: Generates random numbers.
Format: RND (<numeric expression>)
Example: A = RND (1)

SCREEN

Function: Returns the character code of a character displayed on the LCD.
Format: SCREEN (<horizontal position>, <vertical position>)
Example: PRINT SCREEN (X, Y)

Functions

SGN

Function: Returns a value according to the value of a numeric expression.
Format: SGN (<numeric expression>)
Example: PRINT SGN (– 1.2)

SIN

Function: Returns the sine value of a numeric expression in radians.
Format: SIN (<numeric expression>)
Example: A = SIN (3.1415926535898/3)

SNS

Function: Inputs data from an input device.
Format: SNS (# <file number> [, <expression>])
Example: A = SNS (#1, &HFF)

SQR

Function: Returns the square root value of a numeric expression.
Format: SQR (<numeric expression>)
Example: A = SQR (2)

START$

Function: Sets a program as the start program.
Format: START$ = <string expression>
Example: START$ = "RUN" + CHR$ (13)

STICK

Function: Returns the status of the cursor Keys.
Format: STICK (<expression>)
Example: A = STICK (0)

STR$

Function: Returns a string representation of a numeric value.
Format: STR$ (<numeric expression>)
Example: A$ + STR$ (123)

STRIG

Function: Returns the status of the space key and the user defined key 6.
Format: STRIG (<expression>)
Example: A = STRIG (0)

STRING$

Function: Returns a string of a specified length.
Format: STRING$ (<numeric expression$_1$>, | <string expression> | <numeric expression$_2$> |)
Example: PRINT STRING$ (5, "A")

TAB

Function: Sets a tab.
Format: TAB (<numeric expression>)
Example: PRINT TAB (8); "CANON"

TAN

Function: Returns the tangent value of a numeric expression in radians.
Format: TAN (<numeric expression>)
Example: A = TAN (3.1415926535898)

TIME$

Function: Sets the hours, minutes, and seconds of the timer.
Format: TIME$ = "[<hours>]:[<minutes>]:[<seconds>]"
Example: TIME$ = "23:53:25"

Functions

TKEY

Function: Returns the status of a key.
Format: TKEY (<string>)
Example: A = TKEY ("A")

USR

Function: Calls a machine language subroutine.
Format: USR (<start address>, <argument>)
Example: A = USR (&H0C00", B)

VAL

Function: Returns the numeric value of a string expression.
Format: VAL (<string expression>)
Example: A = VAL ("123")

VARPTR

Function: Returns the address of a stored variable.
Format: VARPTR (<variable>)
Example: A = VARPTR (B)

Graphic instructions

Instruction	Meaning·Format·Function
A	ALL INITIALIZE The pen is lifted from the paper and returned to the left corner. This point becomes the start point, and the X07-BASIC returns to the text mode.
C	COLOR CHANGE Cn (n = 0...black, 1...blue, 2...green, 3...red) Select the pen according to the color code.
D	DRAW D x_1, y_1, ... xn, yn Starting from the curent pen position, lines are drawn by connecting each points successively.
F	NEW LINE Carriage return and line feed are executed only when alpha rotate (Q) is 0.
H	HOME The pen is lifted from the paper and returned to the start point
I	INITIALIZE The current pen position becomes the start point.
J	RELATIVE DRAW J Δx_1, Δy_1,...Δxn, Δyn Starting from the current pen position, lines are drawn one by one to the relative point (xn, yn)
L	LINE TYPE Lp (p = 0 ~ 15) Specifys the pitch of dotted line: 0 = solid line, 1 ~ 15 = dotted line.
M	MOVE M x, y The pen is lifted and moved to (x, y).
P	PRINT P c_1 c_2 c_3 ... cn (n < 256) Prints characters in the graphic mode.
Q	ALPHA ROTATE Qn (n = 0 ~ 3) Changes the angle of characters.
R	RELATIVE MOVE R Δx, Δy Moves the pen by (Δx, Δy) from the current pen position.
S	SCALE SET Sn (n = 0 ~ 15) Changes the character size.

Character code table

Decimal form	Hexa decimal form	Charac-ter	Decimal form	Hexa decimal form	Charac-ter	Decimal form	Hexa decimal form	Charac-ter	Decimal form	Hexa decimal form	Charac-ter
0	00	Refer to the control code table	32	20		64	40	@	96	60	`
1	01		33	21	!	65	41	A	97	61	a
2	02		34	22	"	66	42	B	98	62	b
3	03		35	23	#	67	43	C	99	63	c
4	04		36	24	$	68	44	D	100	64	d
5	05		37	25	%	69	45	E	101	65	e
6	06		38	26	&	70	46	F	102	66	f
7	07		39	27	'	71	47	G	103	67	g
8	08		40	28	(	72	48	H	104	68	h
9	09		41	29	)	73	49	I	105	69	i
10	0A		42	2A	*	74	4A	J	106	6A	j
11	0B		43	2B	+	75	4B	K	107	6B	k
12	0C		44	2C	,	76	4C	L	108	6C	l
13	0D		45	2D	-	77	4D	M	109	6D	m
14	0E		46	2E	.	78	4E	N	110	6E	n
15	0F		47	2F		79	4F	O	111	6F	o
16	10		48	30	0	80	50	P	112	70	p
17	11		49	31	1	81	51	Q	113	71	q
18	12		50	32	2	82	52	R	114	72	r
19	13		51	33	3	83	53	S	115	73	s
20	14		52	34	4	84	54	T	116	74	t
21	15		53	35	5	85	55	U	117	75	u
22	16		54	36	6	86	56	V	118	76	v
23	17		55	37	7	87	57	W	119	77	w
24	18		56	38	8	88	58	X	120	78	x
25	19		57	39	9	89	59	Y	121	79	y
26	1A		58	3A	:	90	5A	Z	122	7A	z
27	1B		59	3B	;	91	5B	[	123	7B	{
28	1C		60	3C	<	92	5C	¥	124	7C	|
29	1D		61	3D	=	93	5D	]	125	7D	}
30	1E		62	3E	>	94	5E	^	126	7E	~
31	1F		63	3F	?	95	5F	_	127	7F	←

Device table

Device name	Meaning	Output	Input	1st parameter	2nd paramater
CON:	Console	○	○	—	—
KBD:	Keyboard		○	—	—
COM:	Serial I/O (RS-232C)	○	○	Baud rate: 100 ~ 8000 Default: 4800	ACIA mode: A ~ H Default: B
OPT:	Optical coupler	○	○	Baud rate: 100 ~ 2400 Default: 1200	ACIA mode: A ~ H Default: B
GPR:	Graphic printer	○		—	—
LPT:	Centronics-type printer	○		—	—
PRT:	Thermal printer	○		300 baud, fixed	—
CASI:	Cassette recorder input		○	1200 baud, fixed	Mode B, fixed
CASO:	Cassette recorder output	○		1200 baud, fixed	Mode B, fixed
RAM:	RAM file	○	○	Size: number of bytes	Data type: A ~ Z Default: D

Character code table

Decimal form	Hexa-decimal form	Charac-ter	Decimal form	Hexa-decimal form	Charac-ter	Decimal form	Hexa-decimal form	Charac-ter	Decimal form	Hexa-decimal form	Charac-ter
128	80	♠	160	A0	√	192	C0	タ	224	E0	ô
129	81	♥	161	A1	◦	193	C1	チ	225	E1	ö
130	82	♣	162	A2	⌈	194	C2	ツ	226	E2	ò
131	83	♦	163	A3	⌋	195	C3	テ	227	E3	ŷ
132	84	○	164	A4	、	196	C4	ト	228	E4	ç
133	85	●	165	A5	＋	197	C5	ナ	229	E5	ç
134	86	Ä	166	A6	ヲ	198	C6	ニ	230	E6	Ñ
135	87	Å	167	A7	ァ	199	C7	ヌ	231	E7	ñ
136	88	ä	168	A8	ィ	200	C8	ネ	232	E8	Γ
137	89	à	169	A9	ゥ	201	C9	ノ	233	E9	Σ
138	8A	â	170	AA	ェ	202	CA	ハ	234	EA	Π
139	8B	á	171	AB	ォ	203	CB	ヒ	235	EB	Ω
140	8C	å	172	AC	ャ	204	CC	フ	236	EC	α
141	8D	a	173	AD	ュ	205	CD	ヘ	237	ED	β
142	8E	Ï	174	AE	ョ	206	CE	ホ	238	EE	γ
143	8F	ï	175	AF	ッ	207	CF	マ	239	EF	∂
144	90	ì	176	B0	ー	208	D0	ミ	240	F0	ε
145	91	î	177	B1	ア	209	D1	ム	241	F1	δ
146	92	í	178	B2	イ	210	D2	メ	242	F2	θ
147	93	Ü	179	B3	ウ	211	D3	モ	243	F3	χ
148	94	ü	180	B4	エ	212	D4	ヤ	244	F4	λ
149	95	ù	181	B5	オ	213	D5	ユ	245	F5	μ
150	96	û	182	B6	カ	214	D6	ヨ	246	F6	ρ
151	97	ú	183	B7	キ	215	D7	ラ	247	F7	π
152	98	Ë	184	B8	ク	216	D8	リ	248	F8	τ
153	99	ë	185	B9	ケ	217	D9	ル	249	F9	φ
154	9A	è	186	BA	コ	218	DA	レ	250	FA	χ
155	9B	ë	187	BB	サ	219	DB	ロ	251	FB	ω
156	9C	é	188	BC	シ	220	DC	ワ	252	FC	ν
157	9D	Ö	189	BD	ス	221	DD	ン	253	FD	ξ
158	9E	ö	190	BE	セ	222	DE	゛	254	FE	ϕ
159	9F	ò	191	BF	ソ	223	DF	°	255	FF	÷

Printer control code table

Code	Function
CHR$(8)	BS (Back space)
CHR$(10)	LF (Line Feed)
CHR$(11)	LU (Line Up)
CHR$(13)	CR (Carriage Return)
CHR$(17)	DC1 (Text Mode)
CHR$(18)	DC2 (Graphic Mode)

ACIA mode table

Mode	A	B	C	D	E	F	G	H	Operation
EP	0	0	0	0	1	1	1	1	Parity 1 = Even 0 = Odd
PEN	0	0	1	1	0	0	1	1	Parity 1 = Enable 0 = Disable
CBL	0	1	0	1	0	1	0	1	Bit length 1 = 8 bits 0 = 7 bits

Control code table

Code	Function	Corresponding key
0	Null string	
1		CTRL + A
2	Returns to the top of the previous word.	CTRL + B
3	Stops execution.	CTRL + C , BREAK
4		CTRL + D
5	Erases the following line.	CTRL + E
6	Moves to the top of the next word.	CTRL + F
7	Outputs sound from the speaker.	CTRL + G
8	Erases the previous character.	CTRL + H
9	Tab.	CTRL + I
10	Line feed.	CTRL + J
11	Returns the cursor to the home position.	CTRL + K , HOME
12	Clears the LCD, and returns the cursor to the home position.	CTRL + L , CLR
13	Carriage return, and one line input.	CTRL + M , RETURN
14		CTRL + N
15		CTRL + O
16	Erase everything after the cursor.	CTRL + P
17		CTRL + Q
18	Inputs one space.	CTRL + R , INS
19	Temporarily stops command or program execution.	CTRL + S
20		CTRL + T
21	Erases the current cursor line, and returns the cursor to the beginning of the line.	CTRL + U
22	Erases the character at the cursor position.	CTRL + V , DEL
23		CTRL + W
24	Moves the cursor to the end of the line.	CTRL + X
25		CTRL + Y
26		CTRL + Z
27		
28	Moves the cursor to the right by one position.	▶
29	Moves the cursor to the left by one position.	◀
30	Moves the cursor up by one line.	▲
31	Moves the cursor down by one line.	▼

Error message table

11	/0	**Division by Zero** Division by 0 has been executed.
25	BF	**Bad File Mode** The structure of a file is incorrect.
9	BS	**Bad Subscript** The subscript of an array is incorrect.
17	CN	**Can't Continue** Cannot resume program execution.
10	DD	**Duplicate Definition** An array or user defined function is defined twice.
5	FC	**Illegal Function Call** A statement or function is called incorrectly.
12	ID	**Illegal Direct** A statement has been input in the direct mode even when it is not allowed.
22	IO	**Device I/O Error** An error during input from or output to a device.
27	IR	**Illegal Run** Tried to direct RUN a file which has incorrect contents.
15	LS	**String too Long** A string is too long.
23	MO	**Missing Operand** A necessary parameter is missing.
24	NE	**File not Exist** The specified file does not exist.
1	NF	**NEXT without FOR** The FOR statement is missing.
26	NO	**File not Open** An undefined file number is used.
19	NR	**No RESUME** The RESUME statement is missing in an error recovery subroutine.
4	OD	**Out of Data** Data to be read by the READ statement is not supplied.
7	OM	**Out of Memory** Not enough memory.
14	OS	**Out of String Space** The string space is too small.
6	OV	**Overflow** Numeric values exceed the allowed range.
3	RG	**RETURN without GOSUB** The RETURN statement has been executed before the GOSUB statement.
20	RW	**RESUME without Error** The RESUME statement has been executed without the occurence of an error.
2	SN	**Syntax Error** Incorrect grammar.
16	ST	**String Formular too Complex** A string expression is too complex.
13	TM	**Type Mismatch** The variable type is wrong.
18	UF	**Undefined User Function** Called on a undefined function.
8	UL	**Undefined Line Number** A line number is specified incorrectly.
—	UE	**Undefined Error** An error with an error code other than the above error code has occured.