Interpreted, Compiled and Extended BASIC

Quick Reference Card

FLUKE®

NOTE

The **REF** column refers to the Reference number used in the Reference Section for the Basic Keyword.
The quantities X, Y, N%, A$, and B$ used in the following examples are the arguments received by the statement, function or command.
I.M.C. = Immediate Mode Command
† = Interpreted BASIC only
* = Compiled and Extended BASIC only
** = Extended BASIC only

REF	KEYWORD	ACTION
1	ABS(X) Function	Returns |X| or the absolute value of its argument.
69	AND Operator	Logical (Boolean) AND operation.
2	Arithmetic Operators	Unary operators, addition, subtraction, multiplication, division, and exponentiation.
3	Array Variables	Describes the format and usage of array variables.
4	ASCII(A$) Function	Returns the value of the ASCII character as an integer.
5	ASH(A%,C%) Function	Signed arithmetic shift C% of integer A%.
6	ASSIGN Function	Assign a device as system device (SY0:).
7	ASSIGNMENT Operator	Using "=" to assign values.
8	ATN(X) Function	Returns arctan(X) in radians.
9	BREAK Statement	Send break signal on RS-232 port.
10	CALL Statement	Execute CBASIC, XBASIC, FORTRAN, or Assembly subroutine.
130a*	CASE Statement	Branch test statement for SELECT Statement.
11	CHR$(X%) Function	Yields a one character string equal to the ASCII equivalent of X%.
12	CLEAR Statement	Send IEEE-488 device clear signal to device or devices.
13	CLOSE Statement	Closes a channel previously opened to a port, device, or file.
14	CMDFILE Function	Logical function: is a command file active?
15	CMDLINE$ System Function	Read the current FDOS command line.
16	COM Statement	Reserve variables and arrays in common area for chained programs.
17	CONFIG Statement	Configure or unconfigure instrument for parallel poll.
18+	CONT TO I.M.C.	Resume program execution at linenumber from breakpoint caused by STOP, STOP ON, CONT TO, or ⟨CTRL⟩/C.
19	COPY Statement	Copy a file.
20	COS(X) Function	Returns cosine(X) for X in radians.
21	CPOS(X%,Y%) Function	Returns the string needed to position the cursor at the given X,Y coordinates.
117	DATA Statement	Define sequence of data items to be read by READ.
150	DATE$ System Variable	Returns date as DD-MMM-YY.
22	DEF FN Statement	Define a function.

P/N 718155 4/85

REF	KEYWORD	ACTION
23+	DELETE I.M.C.	Delete program lines in memory.
24	DIM Statement	Dimension main memory array or virtual array.
25	DIR Statement	Print directory of file structured device (like FUP /L).
26	DISABLE Statement	Disable all interrupts (except ON ERROR, ON CTRL/C).
27	DISABLE CMDFILE Stmt	Suspend execution of command file.
28	DUPL$(A$,N%) Function	Return string A$ duplicated N% times.
29	EDIR Statement	Print extended directory of file structured device (like FUP/E)
30+	EDIT I.M.C.	Enter the EDIT mode.
36a*	ELSIF Statement	Conditional ELSE clause of Extended IF Statement.
31	ENABLE Statement	Enable interrupts after a DISABLE Statement.
32	ENABLE CMDFILE Stmt	Resume command file execution (see DISABLE CMDFILE).
33	END Statement	The last program statement, causes exit to Ready prompt.
36a*	ENDIF Statement	Closes Extended IF Statement.
69a*	ENDLOOP Statement	Closes LOOP statement.
130a*	ENDSELECT	Closes SELECT statement
170*	ENDWHILE Statement	Closes WHILE statement.
150	ERL System Variable	Returns line number of last error.
150	ERR System Variable	Returns error number of last error.
150*	ERR$ System Variable	Returns name of module where last error occurred.
34	EXEC Statement	Chain to machine language (.FD2) or command file (.CMD).
35	EXIT	Exit BASIC, return to FDOS or SHELL.
35a**	EXPORT Statement	Declares a set of variables and arrays as global variables.
36	EXP(X) Function	Returns natural log base e raised to power X.
36a*	EXTENDED IF Statement	Multi-line IF Statement.
37	FLEN Variable	Returns the length of a file.
38	Floating-Point Variables	Describes floating-point (real) variables, constants, and data.
39	FOR and NEXT Statement	Sets up a program loop.
40	GOSUB Statement	Subroutine call — goto linenumber, do it, come back.
41	GOTO Statement	Branch to linenumber, don't return.
42	IF-GOTO Statement	IF condition GOTO linenumber.
42	IF-THEN Statement	IF condition THEN do this.
42	IF-THEN-ELSE Statement	IF condition THEN do this ELSE do this.
42a**	IMPORT Statement	Declares a set of variables and arrays defined in another program unit as global variables.
43	INCHAR(X%) Function	Read single character from open channel or console.
44	INCOUNT(X%) Function	Number of characters/lines available from serial device.
45	INIT Statement	Initialize IEEE-488 port.
46	INPUT Statement	Get data from user.
47	INPUT #n Statement	Data input from channel.
48	INPUT Statement	IEEE-488 Bus input.
49	INPUT LINE Statement	Get string data from user.
50	INPUT LINE #n Statement	Get string data from I/O channel.
51	INPUT LINE @ Statement	Get string data from IEEE-488 Bus.
52	INPUT LINE WBYTE Stmt	IEEE-488 Bus input with prior Bus message.
53	INPUT WBYTE Statement	Send IEEE-488 Bus message prior to getting data item.
54	INSTR(N%,A$,B$) Function	Returns the first character position of a substring B$ within string A$ starting at N%.
55	INT(X) Function	Return the greatest integer ≤ X.
56	INTEGER Variables	Describes the format and usage of integer variables, constants, and data.
57	KEY System Variable	Returns touch-sense "key" last pressed.
58	KILL Statement	Remove, erase this file from device.
59	LCASE$(A$) Function	Convert string to lower case.
59a*	LEAVE Statement	Leave a structured loop.
60	LEFT(A$,N%) Function	Return the leftmost N% characters of A$.
61	LEN(A$) Function	Return the number of characters in A$.
62	LET Statement	Assignment operator — LET X = 5.
63+	LINK Statement	Load an object file into system RAM.
64+	LIST I.M.C.	LIST program lines.
65	LN(X) Function	Return the natural log (log e) of X.
66	LOCAL Statement	Reset IEEE-488 Bus instruments to local state.
67	LOCKOUT Statement	Implement local lockout on the IEEE-488 Bus.
68	LOG(X) Function	Return the log (log 10) of X.
69	Logical Operators	Describes the AND, OR, XOR, and NOT operators.
69a*	LOOP-ENDLOOP Statement	Loop until GOTO, LEAVE, or interrupt.
70	LSH(A%,C%) Function	Unsigned logical integer shift (see ASH).
71	MATH Functions	Describes the mathematical functions.
72	MEM System Variable	Returns amount of available memory in bytes.
73	MID(A$,S%,N%) Function	Return substring of A$ given starting position S% and number of characters N%.

REF	KEYWORD	ACTION
74	MOD(X,Y) Function	Return remainder of division from division operation.
39	NEXT Statement	Closes FOR-NEXT loop.
69	NOT Operator	Logical (Boolean) NOT operation.
75	NUM$(X) Function	Convert X to a string as PRINT or PRINT USING would output it.
76	OFF CTRL/C Statement	Disable ⟨CTRL⟩/C interrupt processing.
77	OFF #n Statement	Disable interrupt processing from channel #n.
78	OFF CLOCK Statement	Disable clock interrupts.
79	OFF ERROR Statement	Disable action of previous ON ERROR GOTO statement.
80	OFF INTERVAL Statement	Disable interval interrupts.
81	OFF KEY Statement	Disable interrupt from Touch Sensitive Overlay.
82	OFF PORT Statement	Disable port status change interrupts.
83	OFF PPOL Statement	Disable parallel poll interrupt processing.
84	OFF PPORT Statement	Disable PPORT status checking.
85	OFF SRQ Statement	Disable service request interrupt processing.
86†	OLD Statement	Load a program into memory from mass storage, do not run.
87	ON CLOCK Statement	Create interrupt at specific time of day.
88	ON CTRL/C Statement	Enable ⟨CTRL⟩/C interrupt processing.
89	ON ERROR Statement	Enable interrupt processing on error condition.
90	ON #n Statement	Enable interrupt processing from channel #n.
91	ON...GOSUB Statement	Multi-way GOSUB Statement.
91	ON...GOTO Statement	Multi-way GOTO Statement.
92	ON INTERVAL Statement	Create interrupt at interval (SET INTERVAL).
93	ON KEY Statement	Enable interrupt from Touch Sensitive Overlay.
94	ON PORT Statement	Enable port status change interrupts.
95	ON PPOL Statement	Enable parallel poll interrupt processing.
96	ON PPORT Statement	GOTO from parallel port interrupt.
97	ON SRQ Statement	Enable service request interrupt processing.
97a*	ON..SUBRET Statement	Subroutine return on interrupt.
98	OPEN Statement	OPEN a channel to a file, device, or port.
99	OPEN Statement	OPEN a channel to a virtual array.
69	OR Operator	Logical (Boolean) OR operation.
100	PACK Statement	Pack a file structured device.
101	PASSCONTROL Statement	Give up IEEE-488 control to another controller.
102	PI User Constant	π = 3.14159265358979.
102a	PORT Expression	Direct address to IEEE-488 port.
103	PORTSTATUS(X%) Function	Determine status of IEEE-488 interface port.
104	PPL Function	Parallel poll Bus port and return result.
105	PRINT Statement	PRINT something (usually on the controller's display).
106	PRINT @ Statement	Output something to the IEEE-488 Bus.
107	PRINT #n Statement	PRINT something to a channel.
108	PRINT USING Statement	PRINT something USING this format.
109	PROTECT Statement	Set delete protection for a file.
110	QDIR Statement	Quick directory listing of file-structured device (like FUP/Q).
111	RAD$(X%,B%) Function	Integer X% to base B% conversion.
112	RANDOMIZE Statement	Shake up the random number generator.
113	RBIN Statement	Receive binary data from IEEE-488 instruments.
114	RBIN WBYTE Statement	Output data specified by WBYTE, then perform RBIN.
115	RBYTE Statement	Read fixed-length byte from IEEE-488 instruments.
116	RBYTE WBYTE Statement	Output data specified by WBYTE, then perform RBYTE.
117	READ, DATA, and RESTORE Statement	READ the data from the DATA statements, RESTORE (resume) reading of data from the DATA statement.
118	Relational Operators	Describes =, >, <, <>, <=, and >= operators.
119	REM Statement	REMark, comment. Comment your programs.
120	REMOTE Statement	Set IEEE-488 Remote Enable line to true.
121†	REN Immediate Mode Cmd	Renumber the program in memory.
122	RENAME Statement	Rename a file.
123*	REPEAT-UNTIL	Loop with condition test at UNTIL.
124†	RESAVE Statement	Non-interactive SAVE.
124†	RESAVEL Statement	Non-interactive SAVEL.
117	RESTORE Statement	Resume or re-read DATA statements(s).
125	RESUME Statement	Resume program operation after interrupt.
126	RETURN Statement	Subroutine return: go back to calling program after GOSUB.
127	RIGHT(A$,N%) Function	Return a substring of A$ starting at N% to the end of A$.
128	RND Function	Return a pseudorandom number.
129	RUN I.M.C.	Run a BASIC program.
129	RUN Statement	Chain to another BASIC program.

REF	KEYWORD	ACTION
130†	SAVE Command	SAVE program in file on device, ask before clobbering old file.
130†	SAVEL Statement	SAVE program in lexical form, ask before clobbering old file.
130a*	SELECT Statement	n-way branch statement. (CASE)
131	SET CLOCK Statement	Set clock interrupt (ON CLOCK...GOTO).
132	SET CMDLINE$ Statement	Set the FDOS command line.
133	SET DATE Statement	Set the date in the system clock.
134	SET ECHO Statement	Puts the keyboard into normal ("line") mode.
135	SET INTERVAL Statement	Set interval interrupt (ON INTERVAL...GOTO).
136	SET NOECHO Statement	Puts the keyboard into editing ("character") mode.
137	SET SHELL Statement	Use this program when exit occurs, rather than FDOS.
138	SET SRQ Statement	Service request from controller-in-charge.
139	SET TIME Statement	Set the system clock.
140	SGN(X) Function	Return the sign of X (-1, 0, 1).
141	SIN(X) Function	Return sine(X) for X in radians.
142	SPACE$(N%) Function	Return a string of N% space characters.
143	SPL(N%) Function	Return status byte of instrument by serial poll.
144	SQR(X) Function	Return the square root of X.
144a+	STEP I.M.C.	Single Step program execution
145	STIME$ System Variable	Returns the current time in 24 hour format (HH:MM:SS).
146	String Comparisons	Describes the usage of relational operators on strings.
147	Strings	Describes string constants, variables, and data.
148	STOP Statement	STOP program execution.
149	STOP ON Statement	STOP program execution on this line number.
149a*	SUB Statement	Define a true subroutine.
149b*	SUBEND Statement	Marks the end of a true subroutine.
149c*	SUBRET Statement	Return from a true subroutine.
150	System Variables	Describes the various system variables.
151	TAB(N%) Function	Move the cursor or printhead to the next tab position.
152	TAN(X) Function	Returns tangent(X) for X in radians.
153	TERM Statement	Specify terminating character for input data (IEEE-488).
150	TIME System Variable	Returns time in milliseconds since midnight.
150	TIME$ System Variable	Returns time as HH:MM in 24 hour format.
154	TIMEOUT Statement	Limit wait time for response to IEEE-488 Bus request.
154A	TRACE OFF Statement	Disable trace facility
155	TRACE Statement	Line number and variable trace facility.
156	TRIG Functions	Describes trigonometric functions.
157	TRIG Statement	Address specified devices as listeners (IEEE-488).
158	TRIM Statement	Trim trailing null characters from virtual array strings.
159	UCASE$(A$) Function	Convert string to upper-case.
160†	UNLINK Statement	Forget previously LINKed .obj file
161	UNPROTECT Statement	Remove delete protection from a file.
123*	UNTIL Statement	Closes REPEAT loop.
162	VAL(A$) Function	Convert numeric string A$ to floating-point.
163	WAIT Statement	Suspend program execution for specified time period.
164	WAIT Statement	Suspend program execution until interrupt occurs.
164	WAIT FOR TIME Statement	Uninterruptible WAIT statement.
165	WAIT FOR KEY Statement	Suspend program execution for TSO input.
166	WAIT FOR PPOL	Suspend program execution for parallel poll on IEEE bus.
167	WAIT FOR SRQ	Suspend program execution for service request on IEEE bus.
168	WBIN Statement	Output single or double precision data (IEEE-488).
169	WBYTE Statement	Send integer array as Bus message to designated port.
170*	WHILE-ENDWHILE Statement	Loop with condition test at WHILE statement
69	XOR Operator	Logical (Boolean) exclusive OR operation.

REF	SYMBOL	ACTION	REF	SYMBOL	ACTION
56	N%	Integer variable or data.	2	*	Multiplication operator (X = Y * 5).
147	A$	String variable or data.	2	/	Division operator (X = Y / 2).
3	A$(X,Y)	String array (2 dimensional).	2		Exponentiate (X 2 = X * X).
3	A%(X,Y)	Integer array (2 dimensional).	118	<	Relational less than operator.
3	A(X,Y)	Floating-point variable array.	118	<=	Relational less than or equal to operator.
7	=	Assignment operator (let X = Y).	118	>	Relational greater than operator.
118	=	Relational equality operator.	118	>=	Relational greater than or equal to operator.
2	+	Addition operator (X = Y + 3).			
146	+	String concatenation operator.	118	<>	Relational not equal operator.
2	-	Subtraction operator (X = Y -3).	119	!	Denotes comment following "!" symbol.

ERROR CODES

TYPE	NUMBER	LEVEL	ERROR
OVERFLOW	0		F Memory overflow
	1		F Virtual array file >64K bytes long, or >64K elements (BC)
	2		F Virtual array file too small for arrays
SYSTEM	100	F	BASIC interpreter or Runtime system internal error
	101	F	Incompatible Lexical file or Compiled BASIC program
COMMAND	200	F	Immediate mode error
	201	F	Cannot CONTinue
	202	F	STEP outside break mode
I/O	300	R	Device not ready
	301	R	Disk write protected
	302	R	Illegal channel number specified
	303	R	Channel already in use
	304	R	Invalid device name OR device not present
	305	R	File not found on device
	306	R	No room on device
	307	R	Read/write past end of file
	308	R	Channel not open
	309	R	RS-232 channel input queue overflow
	310	R	Input line too long
	311	R	Disk read error
	312	R	Illegal filename syntax
	313	F	Random access to sequential file
	314	F	Sequential access to random file
	315	F	Virtual array assigned to sequential device
	317	R	Illegal directory on device
	318	R	Read (write) from (to) output (input) file
	319	R	ON channel device not RS-232
	320	F	Object file error
	321	R	Device directory full
	322	R	Illegal operation for device
	323	R	File delete protected
	324	R	Can't RENAME file
	325	R	File medium swapped
	326	R	Can't load — too little memory
	327	R	Illegal image file format
	328	R	Command line too long
	329	R	RS-232 port number out of range
	330	R	Parallel port number out of range
INSTRUMENT BUS CONTROL	400	R	Illegal IEEE-488 port number
	401	R	Illegal IEEE-488 device address
	402	R	Illegal IEEE-488 secondary device address
	403	R	Incomplete IEEE-488 handshake
	404	R	Too many ports designated for IEEE-488 function
	405	R	No devices attached to IEEE-488 port
	406	R	No IEEE-488 ports available
	407	R	IEEE-488 port specified is unavailable
	408	R	IEEE-488 port timeout
	409	R	Illegal WBYTE data
	410	R	Parallel poll bit number out of range
	411	R	Parallel poll bit sense not 0 or 1
	412	R	IEEE-488 timeout limit out of range
	413	R	TERM string longer than one character
	414	R	1722 No IEEE-488 driver in System
	415	W	SET SRQ status byte value out of range
	416	R	Illegal IEEE-488 operation for current port state
SYNTAX	500	F	Unrecognized statement
	501	F	Illegal char terminating statement
	502	F	Illegal subscript (0)
	503	F	Mismatched parens
	504	F	Illegal LET
	505	F	Illegal IF
	506	F	Illegal line number
	507	F	Illegal PRINT
	508	F	Illegal format for PRINT or NUM$()
	509	F	Illegal INPUT statement
	510	F	Illegal array dimension size
	511	F	Badly formed define
	512	F	Illegal FOR statement
	513	F	FOR without NEXT
	514	F	NEXT w/o FOR (jump back into "for" loop)
	515	F	Unmatched quotes

ERROR CODES (cont.)

TYPE	NUMBER	LEVEL	ERROR
SYNTAX (cont)	516	F	Ill-formed expression
	517	F	Bad OPEN statement
	518	F	Bad CLOSE statement
	519	F	IEEE-488 syntax error
	520	F	Initial COM at illegal point in program
	521	F	Not a well-structured statement
	522	F	Illegal variable name
	523	F	ON statement syntax error
	524	F	OFF statement syntax error
	525	F	TRACE syntax error
	526	F	Illegal file size in open
	527	F	RENumber parameter error
	528	F	RENumber syntax error
	529	F	ELSE without IF
	530	F	NEXT syntax error
	531	F	INPUT WBYTE requires IEEE-488 input
	532	F	Illegal subrange descriptor
	533	F	WBYTE/RBYTE data not integer type
	534	F	Can't specify column for WBYTE/RBYTE subrange
	535	F	Can't use undimensioned variable for WBYTE/RBYTE I/O
	536	F	Virtual array illegal for WBYTE/RBYTE
	537	F	2-dimensional array illegal with WBYTE/RBYTE I/O
	538	F	Illegal CONFIG statement
	539	F	Illegal RBYTE syntax
	540	F	RBYTE increment <= 0
	541	F	Illegal RBYTE cycle length
	542	F	Illegal WBYTE clause syntax
	543	F	WBIN/RBIN precision error
	544	F	WAIT statement syntax error
	545	F	Illegal CALL statement
	546	F	Virtual array parameter illegal
	547	F	Parameter syntax error
	548	F	Illegal SET statement syntax or option
	549	F	Require filename for SAVE
	550	F	Illegal RENAME statement syntax
MATH	600	F	Illegal mode mixing
	601	R	Arithmetic overflow
	602	R	Arithmetic underflow
	603	R	Divide by zero
	604	R	Square root arg < 0
	605	R	Exponent too large
	606	R	Log argument < = 0
	607	R	Trig function argument too large
	608	R	Illegal arg(s) for power operator
	609	F	Illegal floating point opcode
	610	F	Unimplemented floating operation attempted
TRANSFER	700	F	Illegal GOTO or GOSUB
	701	F	RETURN without GOSUB
	702	F	RESUME outside interrupt handler
	703	F	CALL to undefined FN
	704	R	ON expr GOTO selector out of range
	705	F	CALL to undefined subroutine
	706	F	Parameter count mismatch for CALL
	707	R	Illegal time/date value
	708	R	Timer value uninitialized for "ON TIME"
INPUT	800	R	Out of DATA in READ
	801	W	Too much data entered for INPUT
	802	W	Too little data entered for INPUT
	803	W	Illegal char for INPUT or VAL()
	804	F	Bad format in data statement
VARIABLE	900	F	Access to undefined variable
	901	W	Redimension of array
	902	R	Subscript out of range
	903	F	COM of variable which is already defined
	904	W	String too long for virtual array field
	905	F	Incompatible COM declaration
	906	F	'DIM' within nested interrupt handler
	907	F	Bad XOP 1 call
	908	F	Illegal array parameter (memory vs. virtual)
	909	F	Illegal conformal dim parameter

F = Fatal R = Recoverable W = Warning