

Index	Additional	Description
-----	-----	-----
0	[2 bytes]	Start of a program line / line number. If an empty line could be a END IF or REPeat or REMark

***** Operators *****

1	=	Equal (integer)	See 51, 52, and 53 for ==
2	=	Equal (float)	
3	=	Equal (string)	
4	<>	Not equal (integer)	
5	<>	Not equal (float)	
6	<>	Not equal (string)	
7	<	Less than (integer)	
8	<	Less than (float)	
9	<	Less than (string)	
10	>	Greater than (integer)	
11	>	Greater than (float)	
12	>	Greater than (string)	
13	<=	Less than or equal (integer)	
14	<=	Less than or equal (float)	
15	<=	Less than or equal (string)	
16	>=	Greater than or equal (integer)	
17	>=	Greater than or equal (float)	
18	>=	Greater than or equal (string)	
19	+	Add (integer)	
20	+	Add (float)	
21	-	Subtract (integer)	
22	-	Subtract (float)	
23	*	Multiply (integer)	
24	*	Multiply (float)	
25	/	Divide (integer)	May be the same as DIV
26	/	Divide (float)	

27	&	Join strings	
28	&&	Bitwise AND	
29		Bitwise OR	
30	^^	Bitwise XOR	
40	OR	As in IF (a OR b)	
41	AND	As in IF (a AND b)	
42	XOR		
43	NOT	(Integer)	
44	MOD		
45	DIV	Divide (integer)	
46	NOT	(float)	
47	INSTR		
48	^	Raise to a power (float)	
51	==	Almost equals (integer)	May be the same as = (integer)
52	==	Almost equals (float)	May be the same as = (ON)
53	==	Almost equals (string)	

***** Actual values *****

55	[2 bytes]	An actual integer to put on stack
56	[6 bytes]	An actual floating point to put on stack
57	[undefined]	An actual string to put on stack
58		A zero to put on stack (integer)

***** Normal variables *****

59	[2 bytes]	Get a variable (integer)
60	[2 bytes]	Get a variable (float)
61	[2 bytes]	Get a variable (string). Also get an array element If preceded by 0,0 means the whole string otherwise it's a substring. e.g. 1,5 means a\$(1 TO 5)
62	[2 bytes]	Assign a variable (integer)
63	[2 bytes]	Assign a variable (float)
64	[2 bytes]	Assign a variable (string)

***** Arrays *****

65	[4 bytes]	DIMention a integer array (1 or more elements) First word is no of
66	[4 bytes]	DIMention a float array (1 or more elements) elements - 1
67	[2 bytes]	DIMention a string array (1 element)
68	[4 bytes]	DIMention a string array (2 or more elements) 1 st word is no of elements - 2
69	[2 bytes]	Get an array element (integer) multiple element
70	[2 bytes]	Get an array element (float) multiple element Get an array element (string) See 61
71	[2 bytes]	Assign a numeric array element (integer)
72	[2 bytes]	Assign an array element (float)
73	[2 bytes]	Assign an array element (string)
74	[2 bytes]	Assign a substring of an array element (string)

***** Stack manipulation *****

75	Covert a string variable on stack to an actual string
76	Convert integer on stack to a float
77	Convert a float to an integer
78	Convert an integer on stack to a string
79	Convert to a negative (integer)
80	Convert to a negative (float)
81	Move a float onto the main stack
82	Move a float from the main stack
83	Convert FP ASCII on stack to a float
84	Convert float variable to ASCII for PRINT/INPUT
85	Duplicate integer on top of the stack onto the stack (part of Procedure parameter passing)
86	Move an integer onto the main stack
87	Move an integer from the main stack
88	Convert a decimal ASCII string to an integer (long?)

***** PEEK/POKE *****

90	PEEK
91	PEEK_W
92	PEEK_L
93	POKE
94	POKE_W
95	POKE_L

***** Keyword table commands *****

96 Preceeds actual parameters of a command

97 [2 bytes] Keyword table entry (procedure)
 [undefined] Parameter bytes

98 [2bytes] Keyword table entry (function)
 [undefined] Parameter bytes

***** Procedures and Functions *****

100 [2 bytes] Call a Proc/Fun, also GOSUB
 101 [2 bytes] Local parameter for proc/fun (integer)
 102 [2 bytes] Local parameter for proc/fun (float)
 103 [2 bytes] Local parameter for proc/fun ??? string

109 RETurn/END DEF

***** PRINT *****

110 PRINT
 111 , (comma) In PRINT/INPUT print spaces to the next tab
 112 Newline in PRINT/INPUT - On it's own means PRINT#x
 113 TO In PRINT/INPUT
 118 ! (exclamation) in PRINT/INPUT, print a space

***** INPUT *****

120 INPUT (integer)
 121 INPUT (float)
 122 INPUT (string)

***** FOR loops *****

130 [2 bytes] | Set on offset to \$0001
 131 [2 bytes] | Set offset to float on the stack
 132 [2 bytes] | Set an offset to \$0002
 133 [6 bytes]+ First word is an offset to next program position, After END FOR
 [24 bytes] Second word is number of bytes to skip over
 134 [4 bytes] Set loop variable, First word is a pointer to 133
 Second word is variable pointer
 135 [2 bytes] END FOR & NEXT Word is pointer to 134
 Also NEXT in FOR and REPEAT loops in SuperCharge V2.00

***** IF..THEN *****

140 [2 bytes] IF/THEN

***** SELECT ON *****

145 [2 bytes] ON First word is a pointer to start of code to do
Followed by a GO TO to the start of the next test

146 = (ON) (float)

147 TO (ON)

148 = (ON) (integer) same as index 1?

149 = (ON) (string) same as index 3?

= REMAINDER is handled by inner loop

***** Various functions *****

150 CODE()

151 CHR\$()

152 LEN()

153 RESPR()

154 FILL\$()

155 EOF for embedded DATA statements

156 EOF() channels

157 [4 bytes] DIMN First word is the array
Second word is dimension number in the array

158 [2 bytes] DIMN Without dimension number

***** Various commands *****

160 [2 bytes] GOTO watch out for Def Proc/Fun & REPEAT & IF/THEN/ELSE

161 STOP also NEW

162 READ integer

163 READ float

164 READ string

165 [2 bytes] RESTORE

166 CLEAR

167 [4 bytes + (2*number of destinations)+4] ON..GOTO
1st word - Number of destinations
2nd word - pointer to start of destination offsets
3rd word onwards... list of word sized destination offsets
last 2 words are an end marker

168 [6 bytes + (2*number of destinations)+4] ON..GOSUB
1st word - Number of destinations
2nd word A6 offset to next program line
3rd word - pointer to start of destination offsets
4th word onwards... list of word sized destination offsets
last 2 words are an end marker

***** Channels *****

180	Check channel is open	(These may be the wrong way round)
181	Check if a channel is a window	
183	Colour stipples (double and triple)	

***** Program initialization *****

190		Something to with procedure parameter passing (string)
193	[4 bytes]	Variable initialization - String & Arrays (all)
194	[14 bytes]	Used in BASIC program initialization of some sort
196	[2 bytes]	Variable initialization - Float
197	[2 bytes]	Variable initialization - Integer
199	[6 bytes]	Used in BASIC program initialization of some sort