

## Variables

The following is a list of the mnemonics used for BASICs variables used throughout this document, together with their address and function.

DEGRAG (201 hex., 513 decimal)

Bit 0 of this byte is set if OPTION ANGLE DEGREES is in effect.

SYMBTYPE (202 hex., 514 decimal)

This byte contains the type of symbol of the last item read by the system call GETITEM. See section 6 (System Calls) for more details.

SYMBLEN (203 hex., 515 decimal)

This byte contains the length of the last item read by the system call GETITEM. See section 6 (System Calls) for more details.

SIGNIS (204 hex., 516 decimal)

This byte contains the type of the last sign read by the the system call GETITEM. See section 6 (System Calls) for more details.

STMTEND (206 hex., 513 decimal)

When a statement's execution routine is called, this byte is set to zero, and most extension statements will leave it as zero. This causes BASIC to check for and execute a multi-statement line when the execution routine returns. If it is set to one, then BASIC will check instead for an end of line, and give an error if not found. If set to two, then BASIC will not do any checking but will simply start executing at the line following the current line. If set to three, then BASIC will start executing the current line. The latter is the most common alternative value, since it is quite reasonable for a statement to change the current line (ie. change the variable START to point to a different line. An example of this is GOTO, which sets START to point to the line specified by the line number).

No statement should set STMTEND to a value other than those described above.

CURCHAN (209 hex., 521 decimal)

This byte contains the current I/O channel. Often this will be zero, but may be changed for example with PRINT \$c.

CURPROG (20A hex., 522 decimal)

This byte contains the number of the currently paged in and executing / edited program. It should not be changed by any extensions.

PTR (214 hex., 532 decimal)

This pointer points to the next item in the line currently being executed. It is advanced by the GETITEM system call.

START (216 hex., 534 decimal)

This pointer points to the start of the line currently being executed.

LN\* (218 hex., 536 decimal)

This 16-bit variable contains an integer/line number after a call to the GETITEM system routine, and at a few other times. A value of 0 indicated that an invalid line number / integer number was read.

TEXT (21A hex., 538 decimal)

This pointer points to the start of the current program in RAM. For program 0 this will be in page 0, but for other programs will be at the start of page 1.

CURLOC (21C hex., 540 decimal)

This pointer points to the next free byte within an ALLOCATED space not yet used by CODE ie. the current location counter or program counter.

EXTOP (21E hex., 542 decimal)

This pointer points to the first byte after the top-most BASIC extension, ie. the first byte of any ALLOCATED memory.

MCTOP (220 hex., 544 decimal)

This pointer points to the first byte after the user's ALLOCATED area. This will be the start of program 0, if there. When program 0 is made the current program, TEXT is set to the value in MCTOP.

STKTOP (226 hex., 550 decimal)

This pointer points to the base of BASICs stack, which may move. This is not the Z80 stack.

STXPTR (228 hex., 552 decimal)

This points to the last byte used by BASIC on its stack, and grows downwards in memory. The value in STXPTR should not be saved whilst BASIC is called, since the stack may move.

KEYPTRS (232 hex., 562 decimal)

This pointer points to the first table in the chain of pointers to keyword descriptors. It is not advisable to point this anywhere else.

VARPTR (214 hex., 564 decimal)

This pointer points to the next free byte in the symbol table. It must always be at least 256 bytes below STKPTR.

VARBASE (236 hex., 566 decimal)

This pointer points to an entry in the symbol table, all entries with a higher address being local to the innermost invoked user-defined function.

OUCHPOS (242 hex., 578 decimal)

When CURCHAN above has a value of 255, then the system call OUTCHAR and any call that uses it write the characters to memory instead of to a channel. The OUCHPOS pointer points to the byte after the last character written.

BUFF1 (248 hex., 584 decimal)

This 255-byte buffer is used to hold the input line during tokenisation and execution from immediate mode. For statements that are only executable from a program, it may be used as a general scratch area.

SYMB (347 hex., 839 decimal)

This pointer points to a position in BUFF1 above during tokenising, and is used when, for example, a keyword's tokenisation routine needs to convert a number into a line number.

BUFF2 (349 hex., 841 decimal)

This 253-byte area may generally be used as a scratch area.

MSGBUFF (446 hex., 1094 decimal)

This 80-byte buffer is used by the system call MSG to hold the error message corresponding to the code it is given.

VALUE (595 hex., 1429 decimal)

Floating-point numbers are put into this 9-byte buffer by the system calls GETITEM and READNUMBER.

CHR (59E hex., 1438 decimal)

This 20-byte buffer holds the ASCII representation of a number after conversion by the system call CFTOS.

EASHTAB (DCB hex., 3531 decimal)

This is the start of the 32-entry hash table, consisting of pointers to chains of symbol table entries.

X (EGB hex., 3595 decimal)

This is a 7-byte buffer used during the calculation of transcendental functions and by system calls such as XASSIGN.

Y (E12 hex., 3602 decimal)

As above.

SEGTAB (E54 hex., 3668 decimal)

This marks the end of variables and start of BASICs segment allocation table, which varies in size according to how many working RAM segments are in the system. It is always large enough to contain all segments in the system.