# System variables

The bytes in memory from 23552 to 23733 are set aside for specific uses by the system. You can peek them to find out various things about the system, and some of them can be usefully poked. They are listed here with their uses.

The abbreviations in column 1 have the following meanings:

|  |  |
| --- | --- |
| X | The variables should not be poked because the system might crash. |
| N | Poking the variable will have no lasting effect |

The number in column 1 is the number of bytes in the variable. For two bytes, the first one is the less significant byte the reverse of what you might expect.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Address | Name | Notes | Contents |
| 5C00 | 23552 | KSTATE | N8 | Used in reading the keyboard. |
| 5C08 | 23560 | LAST K | N1 | Stores newly pressed key. |
| 5C09 | 23561 | REPDEL | 1 | Time (in 50ths of a second in 60ths of a second in N. America) that a key must be held down before it repeats. This starts off at 35, but you can POKE in other values. |
| 5C0A | 23562 | REPPER | 1 | Delay (in 50ths of a second in 60ths of a second in N. America) between successive repeats of a key held down: initially 5. |
| 5C0B | 23563 | DEFADD | N2 | Address of arguments of user defined function if one is being evaluated; otherwise 0. |
| 5C0D | 23565 | K DATA | N1 | Stores 2nd byte of colour controls entered from keyboard . |
| 5C0E | 23566 | TVDATA | N2 | Stores bytes of colour, AT and TAB controls going to television. |
| 5C10 | 23568 | STRMS | X38 | Addresses of channels attached to streams. |
| 5C36 | 23606 | CHARS | 2 | 256 less than address of character set (which starts with space and carries on to the copyright symbol). Normally in ROM, but you can set up your own in RAM and make CHARS point to it. |
| 5C38 | 23608 | RASP | 1 | Length of warning buzz.  RAM-DONE : $40 |
| 5C39 | 23609 | PIP | 1 | Length of keyboard click.  RAM-DONE : $00 |
| 5C3A | 23610 | ERR NR | 1 | 1 less than the report code. Starts off at 255 (for 1) so PEEK 23610 gives 255. |
| 5C3B | 23611 | FLAGS | X1 | Various flags to control the BASIC system. |
| 5C3C | 23612 | TV FLAG | X1 | Flags associated with the television. |
| 5C3D | 23613 | ERR SP | X2 | Address of item on machine stack to be used as error return.  NMI\_VECT : RAMTOP – 3 = SP - 2 |
| 5C3F | 23615 | LIST SP | N2 | Address of return address from automatic listing. |
| 5C41 | 23617 | MODE | N1 | Specifies K, L, C. E or G cursor. |
| 5C42 | 23618 | NEWPPC | 2 | Line to be jumped to. |
| 5C44 | 23620 | NSPPC | 1 | Statement number in line to be jumped to. Poking first NEWPPC and then NSPPC forces a jump to a specified statement in a line. |
| 5C45 | 23621 | PPC | 2 | Line number of statement currently being executed. |
| 5C47 | 23623 | SUBPPC | 1 | Number within line of statement being executed. |
| 5C48 | 23624 | BORDCR | 1 | Border colour \* 8; also contains the attributes normally used for the lower half of the screen. |
| 5C49 | 23625 | E PPC | 2 | Number of current line (with program cursor). |
| 5C4B | 23627 | VARS | X2 | Address of variables. |
| 5C4D | 23629 | DEST | N2 | Address of variable in assignment. |
| 5C4F | 23631 | CHANS | X2 | Address of channel data.  NMI\_VECT : $5CB6, initially following the system variables |
| 5C51 | 23633 | CURCHL | X2 | Address of information currently being used for input and output. |
| 5C53 | 23635 | PROG | X2 | Address of BASIC program. |
| 5C55 | 23637 | NXTLIN | X2 | Address of next line in program. |
| 5C57 | 23639 | DATADD | X2 | Address of terminator of last DATA item. |
| 5C59 | 23641 | E LINE | X2 | Address of command being typed in. |
| 5C5B | 23643 | K CUR | 2 | Address of cursor. |
| 5C5D | 23645 | CH ADD | X2 | Address of the next character to be interpreted: the character after the argument of PEEK, or the NEWLINE at the end of a POKE statement. |
| 5C5F | 23647 | X PTR | 2 | Address of the character after the ? marker. |
| 5C61 | 23649 | WORKSP | X2 | Address of temporary work space. |
| 5C63 | 23651 | STKBOT | X2 | Address of bottom of calculator stack. |
| 5C65 | 23653 | STKEND | X2 | Address of start of spare space. |
| 5C67 | 23655 | BREG | N1 | Calculator's b register. |
| 5C68 | 23656 | MEM | N2 | Address of area used for calculator's memory. (Usually MEMBOT, but not always.) |
| 5C6A | 23658 | FLAGS2 | 1 | More flags. |
| 5C6B | 23659 | DF SZ | X1 | The number of lines (including one blank line) in the lower part of the screen. |
| 5C6C | 23660 | S TOP | 2 | The number of the top program line in automatic listings. |
| 5C6E | 23662 | OLDPPC | 2 | Line number to which CONTINUE jumps. |
| 5C70 | 23664 | OSPCC | 1 | Number within line of statement to which CONTINUE jumps. |
| 5C71 | 23665 | FLAGX | N1 | Various flags. |
| 5C72 | 23666 | STRLEN | N2 | Length of string type destination in assignment. |
| 5C74 | 23668 | T ADDR | N2 | Address of next item in syntax table (very unlikely to be useful). |
| 5C76 | 23670 | SEED | 2 | The seed for RND. This is the variable that is set by RANDOMIZE. |
| 5C78 | 23672 | FRAMES | 3 | 3 byte (least significant first), frame counter. Incremented every 20ms. See Chapter 18. |
| 5C7B | 23675 | UDG | 2 | Address of 1st user defined graphic You can change this for instance to save space by having fewer user defined graphics.  RAM-DONE : 21 user defined graphics, copy of the standard characters A – U at the end of RAM (168 bytes), address of start of A |
| 5C7D | 23677 | COORDS | 1 | x-coordinate of last point plotted. |
| 5C7E | 23678 |  | 1 | y-coordinate of last point plotted. |
| 5C7F | 23679 | P POSN | 1 | 33 column number of printer position |
| 5C80 | 23680 | PR CC | X2 | Full address of next position for LPRINT to print at (in ZX printer buffer). Legal values 5B00 - 5B1F. [Not used in 128K mode or when certain peripherals are attached] |
| 5C82 | 23682 | ECHO E | 2 | 33 column number and 24 line number (in lower half) of end of input buffer. |
| 5C84 | 23684 | DF CC | 2 | Address in display file of PRINT position. |
| 5C86 | 23686 | DFCCL | 2 | Like DF CC for lower part of screen. |
| 5C88 | 23688 | S POSN | X1 | 33 column number for PRINT position |
| 5C89 | 23689 |  | X1 | 24 line number for PRINT position. |
| 5C8A | 23690 | SPOSNL | X2 | Like S POSN for lower part |
| 5C8C | 23692 | SCR CT | 1 | Counts scrolls: it is always 1 more than the number of scrolls that will be done before stopping with scroll? If you keep poking this with a number bigger than 1 (say 255), the screen will scroll on and on without asking you. |
| 5C8D | 23693 | ATTR P | 1 | Permanent current colours, etc (as set up by colour statements). |
| 5C8E | 23694 | MASK P | 1 | Used for transparent colours, etc. Any bit that is 1 shows that the corresponding attribute bit is taken not from ATTR P, but from what is already on the screen. |
| 5C8F | 23695 | ATTR T | N1 | Temporary current colours, etc (as set up by colour items). |
| 5C90 | 23696 | MASK T | N1 | Like MASK P, but temporary. |
| 5C91 | 23697 | P FLAG | 1 | More flags. |
| 5C92 | 23698 | MEMBOT | N30 | Calculator's memory area; used to store numbers that cannot conveniently be put on the calculator stack. |
| 5CB0 | 23728 | NMIADD | 2 | This is the address of a user supplied NMI address which is read by the standard ROM when a peripheral activates the NMI. Probably intentionally disabled so that the effect is to perform a reset if both locations hold zero, but do nothing if the locations hold a non-zero value. Interface 1's with serial number greater than 87315 will initialize these locations to 0 and 80 to allow the RS232 "T" channel to use a variable line width. 23728 is the current print position and 23729 the width - default 80. |
| 5CB2 | 23730 | RAMTOP | 2 | Address of last byte of BASIC system area.  RAM-DONE : highest working RAM address ($FFFF) - 168 |
| 5CB4 | 23732 | P RAMT | 2 | Address of last byte of physical RAM.  RAM-DONE : highest working RAM address ($FFFF) |

# Init sequence

DI

I = 3F (to avoid snow effect)

All memory >= 4000 set to 0

(RAMTOP) = 3E : gosub end marker, no significance

SP = RAMPTOP – 1 : machine stack pointer

IM 1

IY = $5C3A (middle of table, enables to reach all system variables)

EI

# Memory Map

0000 3FFF ROM

4000 57FF Screen pixels

5800 5AFF Screen attributes

5B00 5BFF Printer buffer

5C00 5CBF System variables

5CC0 5CCA Channels

5CCB FF57 Available memory (between PROG and RAMTOP)

FF58 FFFF User defined graphics

There are two ways to run machine code on the Spectrum. You can reserve memory for your code and preserve all the system memory areas. That way you can drop back into BASIC after the machine code has completed. Or, you can use all the memory from &5B00 to &FFFF but the only way back to BASIC is with a reset or switching the Spectrum off and on. Games programmers tended to go for the latter option as it gave them the most available RAM.

If you do wish to reserve memory, then issue the command CLEAR *n*, where *n* is the last byte you wish to be available to BASIC. So, if you want your machine code to sit at address &8000 in the memory map, then issue the command:

|  |  |
| --- | --- |
| 1 | CLEAR 32767 |

We want the last byte of memory available to basic to be just before our code, so (&8000 – 1) is 32767 in decimal. Assemble and load your code into address 32768 and run.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Adres lub zmienna** | **Adres zmiennej** | **Pocz. wartość zmiennej** | **Obszar** | **Uwagi** |
| 0 |  |  | ROM | Adresy w zapisie dziesiętnym |
| 14446 |  |  | ROM | "Dziura" - obszar wypełniony wartościami 255 |
| 15616 |  |  | ROM | generator znaków |
| 16384 |  |  | Pamięć ekranu |  |
| 22528 |  |  | Atrybuty |  |
| 23296 |  |  | Bufor drukarki |  |
| 23552 |  |  | Zmienne systemowe | Dokładny opis w osobnej tabelce |
| 23734 |  |  | Mapa microdrive'u |  |
| CHANS | 23631 | 23734 | Informacja o kanałach | Znacznik końca: 80h |
| PROG | 23635 | 23755 | Program w BASIC-u | Początek systemu BASIC |
| VARS | 23627 |  | Zmienne BASIC-a | Znacznik końca: 80h |
| E\_LINE | 23641 |  | Obszar (bufor) edytora BASIC-a | Na końcu obszaru są 2 bajty: 0Dh i 80h |
| WORKSP | 23649 |  | Bufor instrukcji INPUT | Znacznik końca: 0Dh Za nim jest chwilowa przestrzeń pracy |
| STKBOT | 23651 |  | Stos kalkulatora |  |
| STKEND | 23653 |  | Wolny obszar pamięci |  |
| rejestr SP (wskaźnik stosu) |  |  | Stos maszynowy | Wykorzystwany przez mikroprocesor |
| ERRSP | 23613 |  | Stos instrukcji GO SUB | To jest ostatni obszar systemu BASIC |
| RAMTOP | 23730 | 65367 | Miejsce na kod maszynowy | Znacznik końca: 3Eh |
| UDG | 23675 | 65368 | Grafika definiowana | Np. na polskie znaki |
| P\_RAMT | 23732 | 65535 | Koniec pamięci |  |

A = 3F (last page in ROM)

B = 0

DE = FFFF

HL = FFFF

# Basic tokens

http://en.wikipedia.org/wiki/ZX\_Spectrum\_character\_set

## Codepage layout[[edit](http://en.wikipedia.org/w/index.php?title=ZX_Spectrum_character_set&action=edit&section=4" \o "Edit section: Codepage layout)]

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Spectrum Character Set**[[1]](http://en.wikipedia.org/wiki/ZX_Spectrum_character_set" \l "cite_note-man_a-1) | | | | | | | | | | | | | | | | | |
|  | **0x keypress** | **0x character** | **1x** | **2x** | **3x** | **4x** | **5x** | **6x** | **7x** | **8x** | **9x** | **Ax** | **Bx** | **Cx** | **Dx** | **Ex** | **Fx** |
| **x0** |  |  | INK |  | 0 | @ | P | £ | p | [ZXSpectrum80.svg](http://en.wikipedia.org/wiki/File:ZXSpectrum80.svg) | (A) | (Q) | **VAL** | **USR** | **FORMAT** | **LPRINT** | **LIST** |
| **x1** |  |  | PAPER | ! | 1 | A | Q | a | q | [ZXSpectrum81.svg](http://en.wikipedia.org/wiki/File:ZXSpectrum81.svg) | (B) | (R) | **LEN** | **STR$** | **MOVE** | **LLIST** | **LET** |
| **x2** |  |  | FLASH | " | 2 | B | R | b | r | [ZXSpectrum82.svg](http://en.wikipedia.org/wiki/File:ZXSpectrum82.svg) | (C) | (S) | **SIN** | **CHR$** | **ERASE** | **STOP** | **PAUSE** |
| **x3** |  |  | BRIGHT | # | 3 | C | S | c | s | [ZXSpectrum83.svg](http://en.wikipedia.org/wiki/File:ZXSpectrum83.svg) | (D) | (T)4 | **COS** | **NOT** | **OPEN #** | **READ** | **NEXT** |
| **x4** | true video |  | INVERSE | $ | 4 | D | T | d | t | [ZXSpectrum84.svg](http://en.wikipedia.org/wiki/File:ZXSpectrum84.svg) | (E) | (U)5 | **TAN** | **BIN** | **CLOSE #** | **DATA** | **POKE** |
| **x5** | inv video |  | OVER | % | 5 | E | U | e | u | [ZXSpectrum85.svg](http://en.wikipedia.org/wiki/File:ZXSpectrum85.svg) | (F) | **RND** | **ASN** | **OR** | **MERGE** | **RESTORE** | **PRINT** |
| **x6** | caps lock | comma | AT | & | 6 | F | V | f | v | [ZXSpectrum86.svg](http://en.wikipedia.org/wiki/File:ZXSpectrum86.svg) | (G) | **INKEY$** | **ACS** | **AND** | **VERIFY** | **NEW** | **PLOT** |
| **x7** | edit |  | TAB | ' | 7 | G | W | g | w | [ZXSpectrum87.svg](http://en.wikipedia.org/wiki/File:ZXSpectrum87.svg) | (H) | **PI** | **ATN** | **<=** | **BEEP** | **BORDER** | **RUN** |
| **x8** | left | left1 |  | ( | 8 | H | X | h | x | [ZXSpectrum88.svg](http://en.wikipedia.org/wiki/File:ZXSpectrum88.svg) | (I) | **FN** | **LN** | **>=** | **CIRCLE** | **CONTINUE** | **SAVE** |
| **x9** | right | right2 |  | ) | 9 | I | Y | i | y | [ZXSpectrum89.svg](http://en.wikipedia.org/wiki/File:ZXSpectrum89.svg) | (J) | **POINT** | **EXP** | **<>** | **INK** | **DIM** | **RANDOMIZE** |
| **xA** | down |  |  | \* | : | J | Z | j | z | [ZXSpectrum8a.svg](http://en.wikipedia.org/wiki/File:ZXSpectrum8a.svg) | (K) | **SCREEN$** | **INT** | **LINE** | **PAPER** | **REM** | **IF** |
| **xB** | up |  |  | + | ; | K | [ | k | { | [ZXSpectrum8b.svg](http://en.wikipedia.org/wiki/File:ZXSpectrum8b.svg) | (L) | **ATTR** | **SQR** | **THEN** | **FLASH** | **FOR** | **CLS** |
| **xC** | delete |  |  | , | < | L | \ | l | | | [ZXSpectrum8c.svg](http://en.wikipedia.org/wiki/File:ZXSpectrum8c.svg) | (M) | **AT** | **SGN** | **TO** | **BRIGHT** | **GO TO** | **DRAW** |
| **xD** | enter | enter |  | - | = | M | ] | m | } | [ZXSpectrum8d.svg](http://en.wikipedia.org/wiki/File:ZXSpectrum8d.svg) | (N) | **TAB** | **ABS** | **STEP** | **INVERSE** | **GO SUB** | **CLEAR** |
| **xE** | extend | number3 |  | . | > | N | ↑ | n | ~ | [ZXSpectrum8e.svg](http://en.wikipedia.org/wiki/File:ZXSpectrum8e.svg) | (O) | **VAL$** | **PEEK** | **DEF FN** | **OVER** | **INPUT** | **RETURN** |
| **xF** | graphics |  |  | / | ? | O | \_ | o | © | [ZXSpectrum8f.svg](http://en.wikipedia.org/wiki/File:ZXSpectrum8f.svg) | (P) | **CODE** | **IN** | **CAT** | **OUT** | **LOAD** | **COPY** |

(X) characters are User Definable Graphics  
1In the Standard ROM CHR$8 fails backing from line 1 to line zero, and backing off line zero.  
2In the Standard ROM CHR$9 does not actually move the text output position.  
3 Used in Basic programs as a inline marker prefixing a 5-byte floating point number. Is not a printable character or control code.  
4 **SPECTRUM** in 128K BASIC.  
5 **PLAY** in 128K BASIC.

## Printable characters[[edit](http://en.wikipedia.org/w/index.php?title=ZX_Spectrum_character_set&action=edit&section=1" \o "Edit section: Printable characters)]

The printable part of the Spectrum character set, 0x20–0x7F, is almost standard, except that 0x60 is the [pound sign](http://en.wikipedia.org/wiki/Pound_sign" \o "Pound sign) (£) instead of the [grave accent](http://en.wikipedia.org/wiki/Grave_accent" \o "Grave accent) ( ` ) and 0x7F is the [copyright sign](http://en.wikipedia.org/wiki/Copyright_sign" \o "Copyright sign) (©) instead of the [control code](http://en.wikipedia.org/wiki/Control_character" \o "Control character) [DEL](http://en.wikipedia.org/wiki/Delete_character" \o "Delete character). The pound sign was mapped to 0x60, and not 0x23 as in the British variant of ASCII ([ISO-646-GB](http://en.wikipedia.org/wiki/ISO_646" \o "ISO 646)), making both the pound sign and the [number sign](http://en.wikipedia.org/wiki/Number_sign" \o "Number sign) (#) available universally. Code 0x5E contains an [up-arrow](http://en.wikipedia.org/wiki/Arrow_(symbol)" \o "Arrow (symbol)) (↑) as in ASCII-1963 instead of the ASCII-1967 [caret](http://en.wikipedia.org/wiki/Caret" \o "Caret) (^); however, 0x5F has an [underscore](http://en.wikipedia.org/wiki/Underscore" \o "Underscore) (\_) and not a [left-arrow](http://en.wikipedia.org/wiki/Arrow_(symbol)" \o "Arrow (symbol)) (←).

Beyond 0x7F, the Spectrum character set uses the high-bit range, 0x80–0xFF, for special purposes. 0x80–0x8F contain block graphics. 0x90–0xA4 contain the User Defined Graphics (UDGs), which the user can customise with a few lines of [BASIC](http://en.wikipedia.org/wiki/Sinclair_BASIC). 0xA5–0xFF contain tokens (BASIC keywords represented as single characters): for example, pressing P at the beginning of a line would generate the code 0xF6, which would cause the BASIC keyword PRINT to display on the screen. Codes 0xC7–0xC9 are the mathematical operators <= (less-than-or-equal), >= (greater-than-or-equal) and <> (not-equal) respectively; unlike the relational operators of most other systems,[*[citation needed](http://en.wikipedia.org/wiki/Wikipedia:Citation_needed" \o "Wikipedia:Citation needed)*] these are characters in their own right and cannot be achieved by typing the two constituent symbols one after the other.

Mapping the printable Spectrum character set to [Unicode](http://en.wikipedia.org/wiki/Unicode) is possible, but fonts containing some of the block graphics characters are still not commonplace.[*[original research?](http://en.wikipedia.org/wiki/Wikipedia:No_original_research" \o "Wikipedia:No original research)*]

The default printable characters (32 (space) to 127 (copyright)) are stored at the end of the Spectrum's ROM at memory address 15616 (0x3D00) to 16383 (0x3FFF) and are referenced by the system variable CHARS which can be found at memory address 23606/7. The value in CHARS is actually 256 bytes lower than the first byte of the space character so that referencing a printable ASCII character does not need to consider the first 32 characters. As such, the CHARS value (by default) holds the address 15360 (0x3C00).[[2]](http://en.wikipedia.org/wiki/ZX_Spectrum_character_set" \l "cite_note-man_25-2)

The UDG characters (Gr-A to Gr-U) are stored at the end of the Spectrum's RAM at memory address 65368 (0xFF58) to 65535 (0xFFFF). As such, a POKE issued to this address range changes the UDG characters used in subsequent PRINT statements (though not any UDG characters already drawn to the screen). The USR keyword (when followed by a single quoted character) provides a quick method to reference these addresses from BASIC. As with the printable characters, the location of the UDG characters is stored in the system variable UDG.[[2]](http://en.wikipedia.org/wiki/ZX_Spectrum_character_set" \l "cite_note-man_25-2)

The final two UDG characters (Gr-T and Gr-U) are not available on the 128K Spectrums (except in the backward-compatible 48K mode), where they are replaced with two new BASIC keywords: SPECTRUM and PLAY. A side-effect of this is that some older games do not work properly, displaying the keywords SPECTRUM and PLAY instead of their intended graphics.

## Control codes[[edit](http://en.wikipedia.org/w/index.php?title=ZX_Spectrum_character_set&action=edit&section=2" \o "Edit section: Control codes)]

In the control codes area (the C0 range), the Spectrum uses its own proprietary controls, such as INK and PAPER to control foreground and background colour. The only similarity to ASCII is having cursor-left for [0x](http://en.wikipedia.org/wiki/Hexadecimal)08 (ASCII Back Space) and ENTER for 0x0D (ASCII [Carriage Return](http://en.wikipedia.org/wiki/Carriage_Return)), which also generates an automatic linefeed. Cursor-down 0x0A (ASCII Line Feed) can be simulated with 32 spaces printed with OVER 1 (transparent overprint) and cursor-up 0x0B (ASCII Vertical Tabulation) can be simulated with 32 backspaces. The system ROM has a fault which prevents cursor-right 0x09 (ASCII Horizontal Tabulation) from working.[*[citation needed](http://en.wikipedia.org/wiki/Wikipedia:Citation_needed" \o "Wikipedia:Citation needed)*][[3]](http://en.wikipedia.org/wiki/ZX_Spectrum_character_set#cite_note-3)

Control code 0x0e is used to indicate that a floating-point number follows, to accelerate text processing. In a [Sinclair BASIC](http://en.wikipedia.org/wiki/Sinclair_BASIC) program, ASCII numbers are followed by a 0x0E byte, and then a 5-byte representation of the number in binary floating point format. When listing the Basic program the LIST command skips past these 5 bytes, but when the program is being run the 5-byte representation is used and the text part is ignored. Some Spectrum programs used this behaviour to hide the real numbers from the user.[*[citation needed](http://en.wikipedia.org/wiki/Wikipedia:Citation_needed" \o "Wikipedia:Citation needed)*] For example, a BASIC line could contain the ASCII characters GOTO 10, followed by a 0x0e byte and the floating-point representation of 100. Anyone listing the program would see the number 10, but when executed the program would jump to line 100.

## Undefined codes[[edit](http://en.wikipedia.org/w/index.php?title=ZX_Spectrum_character_set&action=edit&section=3)]

Ranges 0x00–0x05, 0x07, 0x0A–0x0C, 0x0F and 0x17–0x1F are undefined.