SHARP USERS
CLUB

SOFTWARE MANUAL III

SHARP MZ-80K/B/A/700/800 COMPUTERS

AND THEIR BASICS

S.U.C. SOFTWARE MANUAL III - EDITORIAL - AUGUST 1990

When we first thought about producing SOFTWARE MANUAL III, our aim was to publish a Manual which would save the Editor and the Librarians the chores of running around getting photocopies of pages of out-of-date issues of the Club Magazine.

Since then the goalposts have moved, and we now find that the main demand for photocopies is from new members who have acquired second-hand Sharp Computers without Manuals, who are desperately looking for guidance on how to start up.

As a result SOFTWARE MANUAL III has turned into a summary of the behaviour of Sharp MZ-80K/B/A/700/800 computers and their Basics.

It is as comprehensive as we could make it in the number of pages at our disposal, and we hope it will serve the dual purpose of helping newcomers to the Sharp scene, and of acting as a reference book for the already-dedicated Sharp enthusiast.

Special thanks are due to John Ibberson, who master-minded the structure of the Section B of the Manual and proof-read most of it. Also to David Want, who seems to be the only person in the U.K. who runs an MZ-800 with Quickdisks AND 5.25" disks; without his help, the MZ-800 sections of this Manual could not have been written at all. And finally to Greg Chapman and John Edwards, who contributed many suggestions and improvements.

Maurice Hawes

SHARP USERS CLUB

CONTENTS

SECTION A - SHARP COMPUTERS

SECTION B - SHARP MZ BASICS 1980-84

SECTION C - MODIFIED BASICS, TOOLKITS, OTHER BASICS

APPENDIX A - CHARACTER CODES & BANK-SWITCHING

APPENDIX B - SHARP BASIC ODDITIES

FIRST STEPS ON YOUR SHARP COMPUTER

All the Sharp computers covered in this Manual have a built-in keyboard and a built-in tape deck, and the three earliest models (MZ-80K, MZ-80A and MZ-80B) also have a built-in video display unit. The two later models (MZ-700 and MZ-800) do not have a built-in video display unit; so if you have an MZ-700 or MZ-800, the first thing to do is connect it to a TV set. This requires a special coaxial lead which has a TV plug on one end, and a 'phono' plug on the other. The lead goes between the TV aerial socket and the 'RF' socket at the back of the computer; if you do not have such a lead, try Tandy's or any similar electronics shop. The TV set should be tuned to Channel 36 or thereabouts, and should be finally adjusted with the computer switched on.

When the computer is switched on, it automatically starts up a simple program on a chip inside the machine. On the MZ-80K/A/700 this program is called the 'Monitor', whereas on the MZ-80B/800 it is called the 'IPL' (Initial Program Loader). The name doesn't matter, as the main purpose of this simple programme in every case is to allow the user to load in a more useful program from tape (or from disk if disk drives are fitted). But before you attempt to load anything, you must understand that there are two types of program, one of which CANNOT be loaded by the 'Monitor' or 'IPL'.

The first type, which CAN be loaded, is written in what is called 'machine-code', and includes BASIC interpreters, most word-processors, programmer's utilities such as Assemblers and Disassemblers, and many games (such as SPACE INVADERS).

The second type, which CANNOT be loaded by the Monitor/IPL, includes all programs which are NOT written in machine code, such as programs written by a BASIC interpreter (or other interpreter such as PASCAL), text files from a word processor, or source code files produced by an Assembler or Disassembler.

If you are a beginner, do not be worried by all this. What it boils down to is, the Monitor/IPL CAN load BASIC or a machine-code word-processor or a machine-code game, but it CANNOT load a program written by BASIC, nor a word-processor text file.

Let us then assume that you wish to load a BASIC interpreter. You must have the appropriate program on tape or disk; if you do not have it, the SUC can supply it. The standard BASIC interpreter for each machine is as follows:-

MACHINE	TAPE BASIC	5.25" DISK BASIC	QUICKDISK BASIC
MZ-80K	SP-5025	SP-6015	NOT APPLICABLE
MZ-80B	SB-5510	SB-6510	NOT APPLICABLE
MZ-80A	SA-5510	SA-6510	NOT APPLICABLE
MZ-700	1Z-013B	2Z-009E	5Z-008
MZ-800	1Z-016	2Z-046	5Z-009

The procedure for loading BASIC is slightly different on each machine. In some cases you have to type in a word or sequence of letters, in some you type in just one letter, and in others the process is automatic as soon as you switch on. We now describe the procedures for each machine in turn, starting with the MZ-80K.

LOADING BASIC ON THE MZ-80K

The Monitor title (SP-1002) is on the top line of the screen. Below this is '*' followed by a flashing 'cursor' which shows that the Monitor is waiting for a keyboard entry. To load tape BASIC, insert the tape and type LOAD (CR). The machine will then tell you to press PLAY, and BASIC will load in about 3 minutes.

To load disk BASIC, put the disk in drive 1 and type FD(CR). The machine will ask BOOT (which) DRIVE ? Answer with '1' or CR. The drive LED will light up and BASIC will load in about 7 seconds.

LOADING BASIC ON THE MZ-80A OR MZ-700

The Monitor title is SA-1510 or 1Z-013A, but the process is the same as on the MZ-80K, except that the commands are single-letter. For tape, type L(CR); for disk, type F(CR). Then proceed as above.

If the MZ-700 is fitted with a QUICKDISK, the top line of the opening display will show 'MAKE READY QD' (i.e. put a disk in!). The QD Monitor title (9Z-503M) is on the next line, with the flashing cursor below. After inserting the disk you can do a 'directory' with QD(CR), or boot BASIC from the disk with QL(CR).

On the MZ-700 only, BASIC will be loaded automatically if a master disk (5.25" or QD) is in its drive when you switch on.

LOADING BASIC ON THE MZ-80B

The IPL program in the MZ-80B is different from the Monitors on the MZ-80K/A/700. No title is shown and loading is designed to be automatic at switch on; if you do not have disk drives, the IPL looks for a tape in the deck and loads it. If the tape deck is empty, IPL displays 'MAKE READY CMT' i.e. put a tape in!

If you have disk drives the machine will automatically try to 'boot' from drive 1. If there is no disk there, or if it is not a BASIC master, you are asked to press 'F' for floppy, or 'C' for cassette. Insert a BASIC master disk in drive 1, or a tape in the tape deck, and press the appropriate key. If you press 'F' you will be asked to specify the drive number (answer 1 or CR).

LOADING BASIC ON THE MZ-800

IPL on the MZ-800 is similar to IPL on the MZ-80B, except that the tape deck does not work automatically. If you do not have disk drives, IPL asks you to press 'C' for cassette or 'M' for Monitor (the MZ-800 has a Monitor AND an IPL). Insert the BASIC tape and press 'C'. The tape will load in about 5 minutes.

If you have disk drives (either 5.25" or Quickdisk), insert a BASIC master disk before you switch on (into drive 1 if you have more than one drive). The machine will then 'boot' automatically at switch-on. Sharp recommend this method of operation; other methods are possible but need not concern us here.

THE NEXT STEP

To learn more about your machine, turn over and read the rest of this section. To learn more about BASIC, turn to SECTION B.

THE TECHNICAL DETAILS OF SHARP COMPUTERS

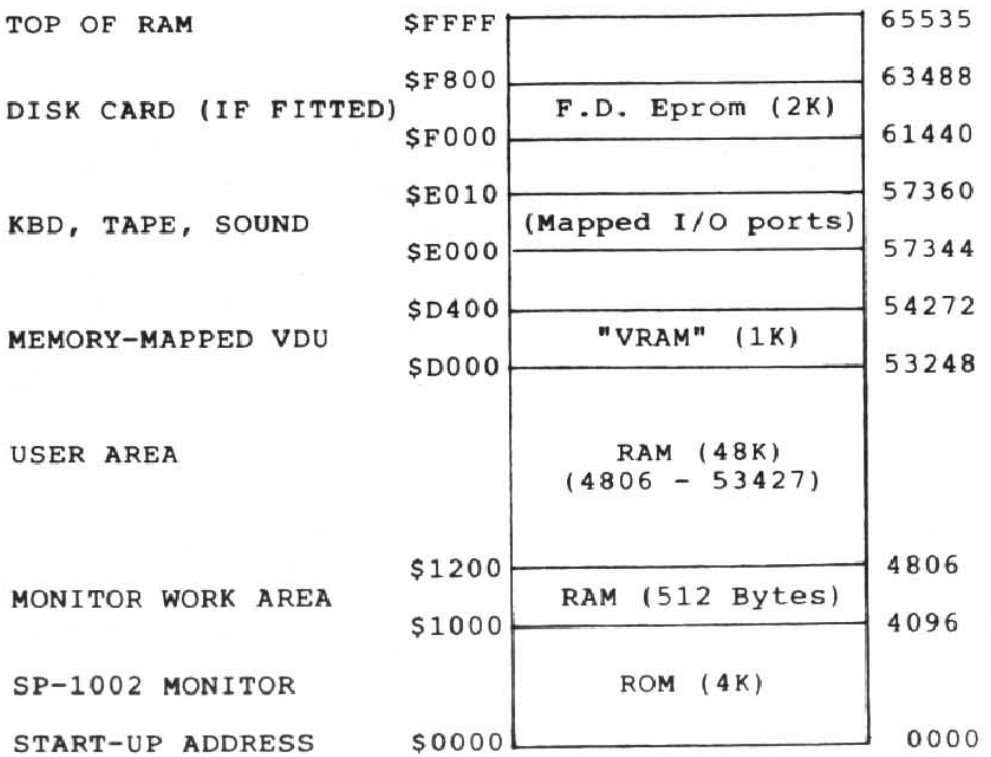
Sharp MZ-80K/B/A/700/800 computers all use the Zilog Z80 central processor (or the later, faster, Z80A). Both can address up to 64K of memory space, plus 256 'input' ports and 256 'output' ports.

In general, 'ports' are used to control keyboards, tape decks, disk drives, printers, joysticks, and memory-switching. They are hard-wired and controlled by the Monitor, and/or the program in use; detailed knowledge is not needed to start using your machine, but 'ports' will be mentioned later, as and when required.

The 'memory' of the computer is of more immediate interest; the user can read most of it, and write to most of it; and in some cases manipulate it, once he knows a few tricks. Therefore memory organisation and memory addresses are useful things to know about a computer, as they help you to understand how it works, and what its limitations are. We start with standard MZ-80K, in which the memory map is never switched and is relatively easy to understand.

THE MEMORY MAP OF THE STANDARD MZ-80K

Some people think of addresses in hexadecimal, and others think in decimal, so both are shown below:-



As you can see, not all the 64K addressing potential of the Z80 chip is utilised. But the addresses from $1000 up to $D400 are connected to RAM, and most of this is available to the user. The Monitor ROM program is permanently wired in at $0000-$0FFF, and takes control of the machine when it is switched on, because the Z80 always starts off by reading the instruction at address $0000.

THE MEMORY-MAPPED VISUAL DISPLAY ON THE MZ-80K

On most Sharp machines the VDU screen is memory-mapped into what is called the video display RAM (VRAM or DRAM for short). On the MZ-80K, one byte of VRAM is allocated to each character on the screen. The 40 x 25 screen can hold 1000 characters, so VRAM must be at least 1000 bytes; in fact $D000-$D3FF is 1024 bytes. The top row is mapped at $D000-$D027, the second row at $D028-$D04F, and so on. The top L.H. corner is always at $D000 (53248 decimal).

SHARP DISPLAY CODES

On the MZ-80K (and some other Sharp machines) a special table of codes is used to store characters in VRAM and display them on the screen; these codes are called DISPLAY codes, and the set of codes for the MZ-80K is shown in APPENDIX A-1. PLEASE NOTE that they are VERY different from the character codes used for all other purposes, which are called ASCII codes (see the next paragraph).

SHARP SO-CALLED 'ASCII' CODES

All computers use codes to represent numbers, letters (upper and lower-case), punctuation marks, arithmetic symbols, and other special characters. In the early days of computers, a set of codes evolved which is now virtually standard. It is called the 'ASCII' set, and uses 128 codes. The first 32 codes (0-31 or 00-1FH) are reserved as 'control codes', and are used to send non-printing messages to printers and the like. This leaves 96 codes (32-127, or 20H-7FH) for normal use, and these are just about sufficient.

Unfortunately, when designing the MZ-80K, Sharp decided to use an 'ASCII' set of their own. It uses the standard ASCII codes for for upper-case letters and most special symbols; but the control codes, and the codes for lower-case letters are very non-standard. The 'SHARP ASCII' set for the MZ-80K is shown in APPENDIX A-2.

USING THE MZ-80K MONITOR SP-1002

Essentially, the SP-1002 ROM Monitor controls the keyboard, the VDU, the tape deck, and the sound. At switch-on, it allows the user to key in the full range of keyboard characters (including lower-case via SHIFT/SML, and all graphics), and will display them on the VDU; but only five commands will provoke a response:-

LOAD(CR) - asks the user to PRESS PLAY, to load a machine-code program off tape. The program can be a Basic such as SP-5025, or an independent machine-code program.

FD(CR) - looks for a disk I/F card at $F000; if found, asks for a drive number, and if that drive is ready and a Master disk is in place, BASIC (SP-6015 or SP-6115) will load. Thereafter, BASIC may be used to load its own programs, or machine-code programs, off disk as required.

GOTO$XXXX(CR) - jumps to hexadecimal address $XXXX. NOTE that GOTO is fussy - the address MUST start with $, and this MUST follow GOTO with NO spaces!!

SG(CR) - turn the key 'bleep' ON
SS(CR) - turn the key 'bleep' OFF

THE LIMITATIONS OF THE MZ-80K MONITOR

The SS and SG commands are trivial, and GOTO is not much use when RAM is empty; so when you first switch on, all the Monitor can usefully do is load a program off tape or disk. If you want to examine memory, write to memory, modify memory, or save or load a section of memory on tape, you must load in a suitable program.

Another limitation of the SP-1002 Monitor-in-ROM is that its keyboard routine does not offer auto-repeating keys i.e. keys do not repeat if held down. If you want this facility, the program you are using must contain a 'repeating keyboard' routine.

OTHER LIMITATIONS OF THE STANDARD MZ-80K

The standard MZ-80K, with its fixed memory map and limited VRAM capacity, is not suitable for use with CP/M, which requires RAM to start at $0000, and an 80-column screen. But it is possible to modify the MZ-80K so that RAM may be switched into $0000-$0FFF, in place of the Monitor ROM; and it is possible to add extra VRAM to occupy some or all of the unused addresses at $D400-$DFFF, after which the screen circuits may be rejigged to offer an 80-column option. But these modifications are tricky and do not always work first time, because many MZ-80Ks contain chips which are too slow to respond to the signals required to drive an 80-column screen.

Sooner or later you, or the new program you are testing, will make a mistake and cause the MZ-80K to 'crash'. The keyboard may go dead, the screen may fill with rubbish, odd sound effects may be heard, or all three things may happen at once! All you can do is switch off, wait about ten seconds, and then switch on; but you have lost everything in RAM, and you must reload the program and start all over again - frustrating and time-consuming!

All other Sharp machines have a RESET button, which may be used to handle a 'crash' in a far less drastic manner; when RESET is pressed, control is passed to the Monitor at $0000, and from the Monitor you can try a jump into the crashed program. The procedure is not infallible, but it works about 90% of the time.

There is no RESET button on a standard MZ-80K, but it is easy to fit one, as Sharp provided suitable terminals inside the machine. Therefore we recommend that if your MZ-80K is not already fitted with a reset button, you should fit one as described below. You should then be able to recover from most 'crashes' with GOTO$XXXX.

FITTING A RESET SWITCH TO YOUR MZ-80K

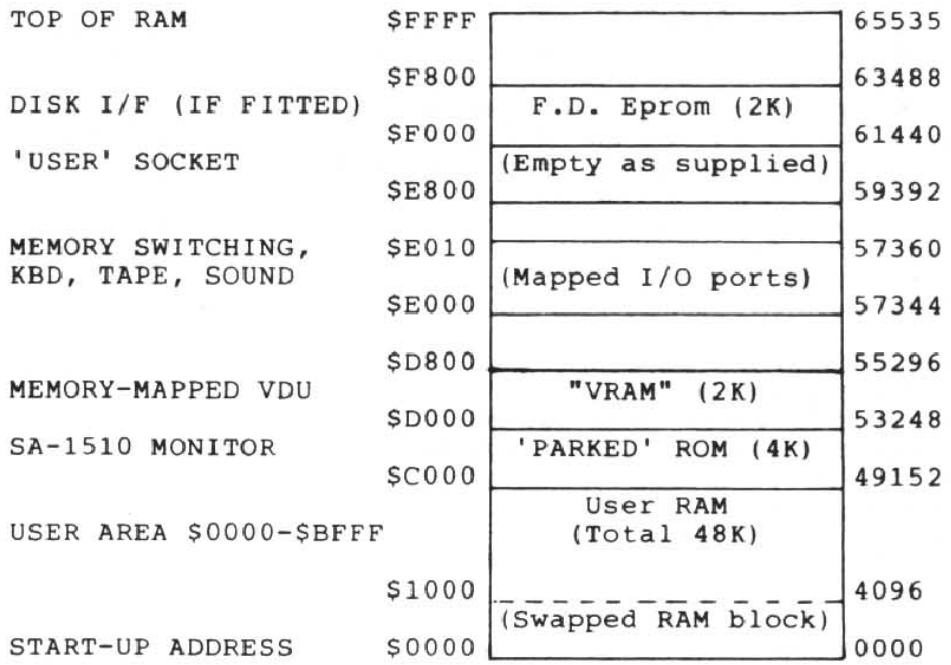
SWITCH OFF AND UNPLUG THE COMPUTER. Remove the screws underneath the edges of the top cover, which fix it to the base unit; then hinge back the top cover and prop it open. On the R.H. side of the main PCB, about 3" back, is a rectangle marked RESET SW, with six solder holes around it. Solder two wires into the two L.H. holes, and connect the free ends to a 'normally-open' push-button switch. Switch back on, LOAD any machine-code program, and then depress the button momentarily; the MZ-80K should reset to $0000 and thus to the SP-1002 Monitor. If all is well, SWITCH OFF AND UNPLUG, drill a correct-sized hole in the R.H. sideplate of the base unit, and mount the switch in it. The switch may then be reached without opening the ccver, but is protected from inadvertent operation.

THE MEMORY MAP OF THE MZ-80A

We now turn to the MZ-80A, which is a very similar machine to the MZ-80K, but with improvements which make it a more powerful machine. The memory map at start-up is very similar to that of the MZ-80K, but includes a double-size VRAM which can store 50 lines of 40 characters. The screen provides a 25-line 'window' on the VRAM, and may be scrolled around VRAM without losing characters. This sounds great, but it brings problems because the top L.H. corner of the screen wanders about in VRAM instead of staying at $D000 as on the MZ-80K. This upsets some programs, so Sharp fitted a software switch to set VRAM to a 25-line mode which emulates the MZ-80K. This causes chaos when dumping the screen to a printer!

Another innovation is the 'USER SOCKET' at $E800-$EFFF. This is NOT DOCUMENTED EXPLICITLY, but it is scanned by the ROM Monitor program, and appears in the Service Manual circuit diagrams. If it is filled by a user program in ROM, this program will be executed by the Monitor at switch-on (e.g. Sharpsoft's 'DYBUG' utility).

A more significant difference is that the SA-1510 Monitor ROM is not hard-wired at $0000-$0FFF, and may 'parked' at $C000-$CFFF by swapping it with the 4K of RAM which is normally there. The swap is performed by LD A,(E00CH) and gives the memory map below:-



THE ROM MONITOR IS WRONGLY LOCATED WHEN 'PARKED'. If you wish to call it, you MUST restore the map to normal with LD A,(E010H). The MZ-80A Owner's Manual hints (badly) at block copying the Monitor ROM into RAM and then swapping memory, to give a RAM Monitor at $0000-0FFF which may be modified by the user. It is also claimed, in a throw-away paragraph which looks like an afterthought, that

'(this configuration).....is especailly (sic) effective when the systems programs used start at address $0000 and when the systems programs utilized make active use of interrupt processing.'

THE 80-COLUMN SCREEN MODIFICATION FOR THE MZ-80A

It is unclear from the above-quoted piece of 'Japanese English' whether Sharp realised that the MZ-80A, with its RAM switched to start at $0000, and its 2048-byte screen VRAM, could run CP/M. If they did not, Sharp dealers certainly did; they spotted that the double-VRAM could store enough characters for an 80 x 25 screen, and a rush of 80-column kits appeared on the market. Almost at a stroke, the MZ-80A was turned into a serious business machine. The only snag with the modification was that it cut out the K VRAM option, which is required by some programs of MZ-80K origin.

In 1986 the S.U.C acquired the rights to Kuma's 80-column kit; it was the best of the bunch and not difficult to fit. The S.U.C. later designed a 'Dual' Monitor eprom. One half contains the original Sharp Monitor ROM, to run 40-column programs which are upset by the 80-column modification. The other half contains the 80-column Monitor ROM. The two halves are switched manually.

SHARP DISPLAY AND SHARP 'ASCII' CODES ON THE MZ-80A

Like the MZ-80K, the MZ-80A uses a special set of DISPLAY codes to store characters in VRAM. It is very similar BUT NOT IDENTICAL to the set of DISPLAY codes used on the MZ-80K, and is shown in APPENDIX A-3.

The MZ-80A also uses a non-standard 'SHARP ASCII' set. This is also very similar to the set used on the MZ-80K, and is shown in APPENDIX A-4. It is marginally better than the MZ-80K set, because it includes some standard ASCII characters which are missing from the MZ-80K set; but it is not as useful as it might be, because the re-instated characters are in non-standard locations!!

Furthermore, there is an 'bug' somewhere in the MZ-80A, which gives very weird results if you send SHARP ASCII characters 96-104 (60H-68H) to the screen via the Monitor. Instead of the characters shown in APPENDIX A-4, you get the following effects:-

```
CHR$(96)  - DELETE CHARACTER
CHR$(97)  - INSERT CHARACTER
CHR$(98)  - CHANGE TO ALPHA MODE (FROM GRAPHICS MODE)
CHR$(99)  - CHANGE TO GRAPHICS MODE (FROM ALPHA MODE)
CHR$(100) - NO EFFECT
CHR$(101) - REVERSE VIDEO
CHR$(102) - CARRIAGE RETURN
CHR$(103) - SCROLL DISPLAY UP ONE LINE
CHR$(104) - SCROLL DISPLAY DOWN ONE LINE
```

The effects are disconcerting to say the least, and CHR$(104) (or hexadecimal 68H in machine-code) is particularly nasty, as the scrolled-down line may be overprinted by the line which follows it; in some cases the scrolled-down line disappears altogether!

If the same characters are POKEd to VRAM, using DISPLAY codes 199-207 ($C7-$CF), they appear as expected, with no problems.

Similar strange effects, plus some undocumented graphics, occur if characters 0-31 (0-1FH) are sent to the screen via the Monitor. Six of these are cursor graphics characters, and their DISPLAY codes 193-198 ($C1-$C6) can be POKEd to VRAM with no problems.

USING THE MZ-80A MONITORS (40-COLUMN AND 80-COLUMN)

At first sight the original MZ-80A 40-column Monitor, SA-1510, is very similar to the MZ-80K Monitor. It allows full use of the keyboard, including lower-case via a SHIFT key, and graphics via the GRPH key; but only 4 literal commands provoke a response:-

L(CR) - PRESS PLAY and load tape program

F(CR) - look for disk I/F, ask for drive, boot master disk

JXXXX(CR) - jump to hexadecimal address XXXX. Note that spaces are NOT accepted, and a '$' sign is NOT required!

B(CR) - toggles key 'bleep' ON/OFF

Here the similarity with the MZ-80K ends. The keyboard input routine in SA-1510 offers repeating keys, and this facility is available to machine-code programs (such as Basic) via a simple call. The routine also recognises a few CTRL combinations:-

CTRL+A - SHIFT LOCK (i.e. switches keyboard to lower-case)
CTRL+D - ROLL DISPLAY UP ONE LINE
CTRL+E - ROLL DISPLAY DOWN ONE LINE
CTRL+Z - PRINT 'RIGHT ARROW' (PASCAL DELIMITER)
CTRL+@ - TOGGLE REVERSE VIDEO
CTRL+[- CLEAR SCREEN AND SET VRAM TO MODE 'K'
CTRL+] - CLEAR SCREEN AND SET VRAM TO MODE 'A'

The Kuma 80-column Monitor behaves as above, except for:-

CTRL+[- CLEAR SCREEN AND SET IT TO 80 COLUMNS
CTRL+] - CLEAR SCREEN AND SET IT TO 40 COLUMNS (VRAM MODE 'A')

THE LIMITATIONS OF THE MZ-80A

The keyboard of the MZ-80A has a nice typewriter-style layout plus a CTRL key and a numeric pad; but it is spoilt by the lack of four separate cursor keys. The scrolling double-VRAM is not a good idea in itself, but facilitates easy conversion to 80 columns, so we should be grateful. The 80-column modification does away with the K VRAM mode, needed for many programs which originated on the MZ-80K. SHARP ASCII characters 96-104 cannot be printed to the screen via the Monitor, and any attempt to do so creates problems.

If you use an 80-column machine, the 'Double Monitor' designed by the S.U.C. is recommended, as it is the only way of getting the machine into K VRAM mode. If you use the machine primarily for CP/M, the S.U.C can supply a program to reconfigure the keyboard with four separate cursor keys (and 1-byte control codes on the numberpad keys 0-9 if desired).

THE RESET KEY ON THE MZ-80A

The MZ-80A comes with a RESET key (in a recessed panel at the rear of the machine, next to the brightness and sound controls). When pressed momentarily, RESET will switch the memory map to its 'normal' state, with the Monitor ROM at $0000-$0FFF, and then restart the Monitor at $0000. From there, the user can restart a program with a warm jump, or load another machine-code program.

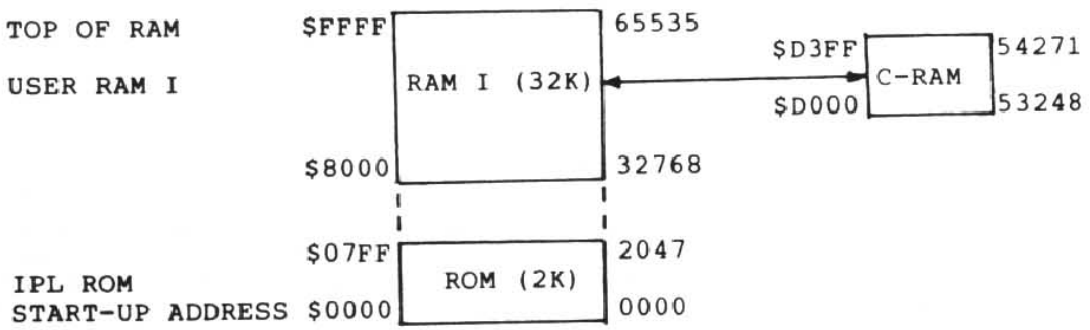
THE MEMORY MAP OF THE MZ-80B

We now turn to the MZ-80B. This is a more sophisticated machine than the MZ-80K/MZ-80A, and has a switchable 40/80 column screen. The U.K. version has a full 64K of user RAM (in two 32K blocks). In addition, there are two banks of VRAM, one for characters or text (2K C-RAM), and one for GRAPHICS (8K G-RAM). There is also a 2K BOOT ROM, called the IPL (Initial Program Loader). The various VRAMS, and the IPL ROM, are switched into and out of the main 64K memory space as required. A further 8K G-RAM is an optional extra.

The IPL is only used at start-up, when it is at $0000-$07FF in the 64K memory space. C-RAM is switched into the 64K memory space by setting BIT 7 of $E8. The G-RAM(S) represent a luxury as far as most applications go, and may not be used at all; but if they are, the program must take care to switch them in and out properly. This may be done from the MZ-80B Basics, as described in Section B of this Manual; or by outputting data to port $F4. For full details of all MZ-80B bank-switching, see APPENDIX A-6.

There are TWO reset buttons, with quite different functions, at the rear of the machine. The reset button nearest to the L.H. back corner of the VDU casing resets to $0000 without switching memory; in other words, memory remains intact. The other button resets to $0000 AND switches to the IPL mode shown below; this is equivalent to switch-on, and the contents of memory are effectively lost.

At switch-on, only the IPL, the C-RAM, and one of the two 32K blocks of RAM (called RAM I) are in use. The screen is in 40 cclumns, so C-RAM extends up to $D3FF:-

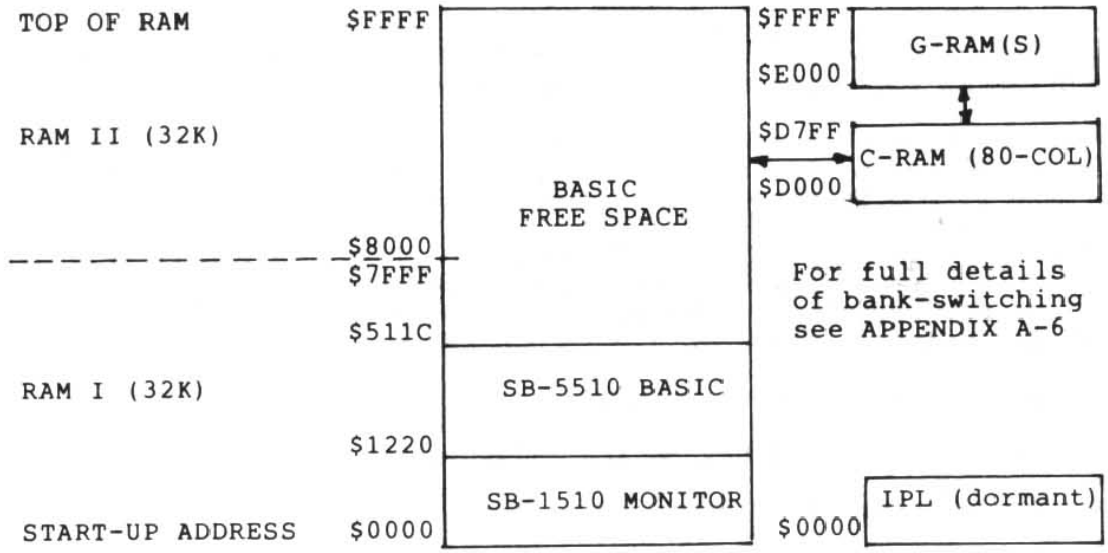


THE FUNCTION OF THE IPL

The IPL first looks for disk drive 1, and will automatically try to boot any disk there. If there is no drive 1, or if it is empty, or if the disk is not bootable, IPL displays a suitable message and asks you to press 'F' or 'C'. An 'F' response asks for a drive number, and then the disk in that drive is booted. A 'C' response opens the tape door and asks you to 'make ready CMT' (i.e. put a tape in). If a tape is already in, 'C' loads it automatically.

The IPL puts the loaded program at $8000 and upwards in RAM I; it then switches itself out, shifts the whole of RAM I down to $0000-$7FFF, switches in the empty RAM II at $8000-$FFFF, and then resets to $0000. Therefore any program loaded by IPL must start and execute at $0000; the 'Monitor-in-RAM' in MZ-80B Basics does just this, and CP/M is designed to execute at $0000 in any case.

THE MEMORY MAP WITH TAPE BASIC SB-5510, IN 80 COLUMNS



THE VISUAL DISPLAY ON THE MZ-80B

The C-RAM switch is transparent, and the character (text) screen appears to be permanently switched in unless the user tries to PEEK or POKE to locations $D000-$D7FF. It then becomes clear that these commands do NOT address themselves to C-RAM. In fact, the only way to put things on the text screen is to enter them in from the keyboard, or PRINT them to the screen in a program.

If a G-RAM has been programmed, its contents may be put on the VDU at the same time as text. So if two G-RAMs are fitted, it is possible to have three 'screens' displayed simultaneously. This can be very effective, and is especially striking when one of the screens is 'cleared' without affecting the other(s).

ASCII AND DISPLAY CODES ON THE MZ-80B

The MZ-80B ASCII set is shown in APPENDIX A-5. The lower half uses standard ASCII characters for codes 32-126 (20H-7EH); the 'odd man out' is character 127 (7FH). This makes it relatively easy to use a standard printer on the MZ-80B, and to transport text files between the MZ-80B and other machines.

The upper half of the MZ-80B ASCII set (128-255 or 80H-FFH) is devoted to a few graphics symbols, signs for '£', 'pi' and 'Yen', and characters 32-126 (20-7EH) in inverted video. There is no real standard for this part of the ASCII set (except possibly that used by IBM), and the characters are not employed in normal text files.

The MZ-80B codes below 20H are used for non-standard purposes; the most unusual is 31(1FH), which prints a chequered square. Most of the others simulate the action of special keys on the keyboard.

As it is not possible to POKE and PEEK to the screen, DISPLAY codes are not used on the MZ-80B.

USING THE MZ-80B MONITOR-IN-RAM, SB-1510

Strictly speaking, SB-1510 is not part of the MZ-80B, but part of Basic. However, as we have included Monitors in the start-up information for other computers, we include it here.

The first thing to note is that, because SB-1510 is in RAM, it may be altered by the user. Furthermore, as SB-1510 contains a command to modify memory, the alterations may be self-inflicted!

SB-1510 uses the full MZ-80B keyboard (lower-case with SHIFT or SHIFTLOCK, graphics with GRPH, and reverse-video with RVS). Most keys do NOT repeat, but the cursor keys repeat if used with SHIFT. Six single-letter commands are recognised:-

D - DUMP memory contents to screen

J - JUMP to specified hexadecimal address

L - LOAD program from tape

M - MODIFY memory

S - SAVE program to tape

V - VERIFY tape against memory

D responds with '$' prompts for (S)tart and (E)nd addresses, which must be in 4-figure hex. If the resultant display fills more than one screen, it may be held by depressing SPACE.

J responds with a '$' prompt for a hexadecimal address, which must be entered as 4 figures.

M also responds with a '$' prompt for a hexadecimal address; to alter the location, enter a 2-figure hexadecimal number and CR; M will then step on one location. Exit with BREAK.

L and V prompt for a filename and CR. The filename may have up to 16 characters, and spaces are allowed. A default filename of CR is accepted, and loads the first file found.

S prompts for a filename (default CR is accepted, but this procedure is not recommended!). S then prompts for START, END and JUMP addresses, in 4-figure hexadecimal. The start and end addresses define the area of memory to be saved, and the JUMP address is saved as the EXECUTE address of the file.

THE MZ-80B USER-DEFINABLE KEYS

The 10 blue keys above the main keyboard are user-definable, and are normally set by a program as it loads. They are very useful if set for CP/M programmes which employ obscure control sequences.

THE DISADVANTAGES OF THE MZ-80B

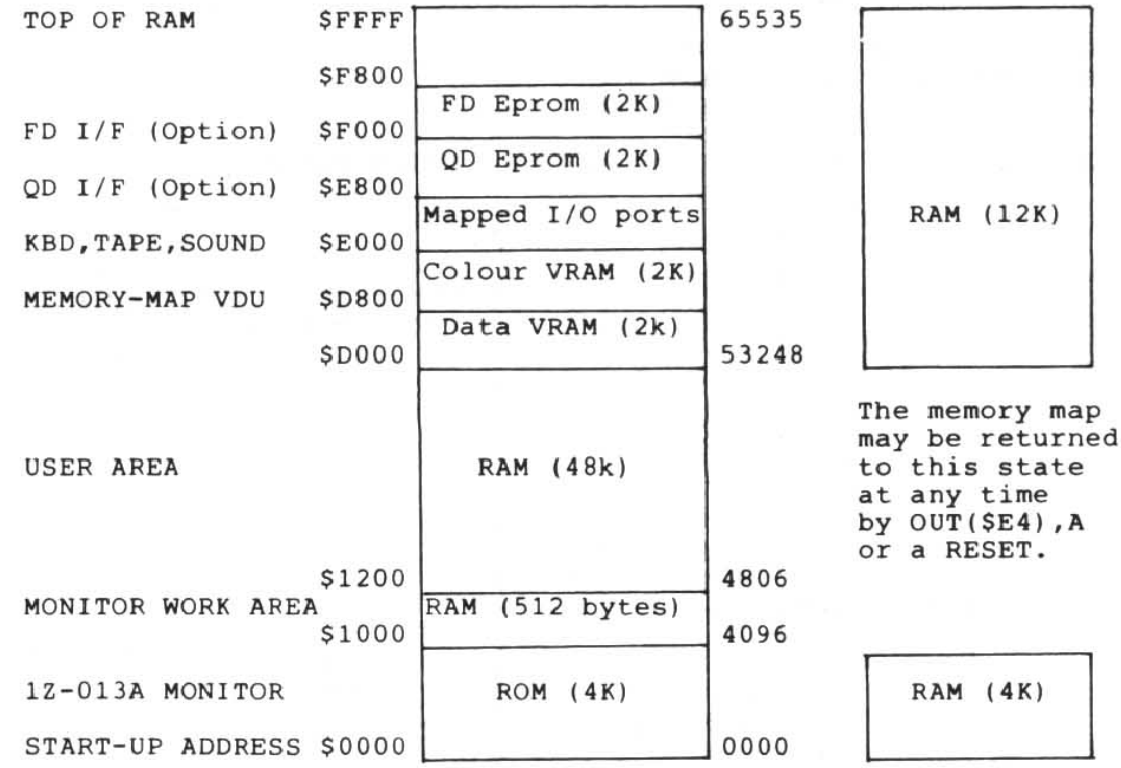
These are few. The non-repeating keyboard is a drawback, but as the Monitor is in RAM this is easily rectified, and the SUC have a program to do so. The special keys above the main keyboard have a tendency to wear out; they can be replaced 'with difficulty'.

THE MEMORY MAP OF THE MZ-700

We now come to the MZ-700. It has 64K of user RAM, and provides a coloured display on a colour TV, or on a separate RGB Monitor; Sharp did not provide a built-in VDU because a colour unit would have been too expensive. Nor did they provide an 80-column screen; presumably because a colour display requires a lot of video RAM, and 80 columns would have required twice as much!

In addition to the main 64K user RAM, there is a separate VRAM which holds all the data for the display and is duplicately mapped at $D000-$DFFF. The area $E000 - $FFFF is also duplicately mapped and is used by the control ports for keyboard, cassette and sound, and by the ROM interfaces of any optional peripherals in use. The whole of the duplicate area from $D000 up to $FFFF is switched in and out of the 64K memory map in one block.

There is also a bank-switched Monitor-in-ROM, which is mapped duplicately at $0000-$0FFF. When the MZ-700 is first switched on, this Monitor and the duplicate area $D000-$FFFF are both switched into the main 64K map, which is as follows:-



LOADING MACHINE-CODE PROGRAMS INTO THE MZ-700

The main purpose of the Monitor-in-ROM is to load machine-code programs such as Basic. It has other uses which will be discussed later. Programs can be loaded off tape with 'L', or off 5.25" disk with 'F'. If an MZ-700 Quickdisk is fitted, its Monitor (9Z-503M) automatically takes over and offers additional commands 'QL' and 'QD'. If floppy and Quickdisk drives are fitted simultaneously, the QD system can only be accessed by switching on (or resetting) with the 'M' or 'Q' key pressed ('Q' boots the QD automatically).

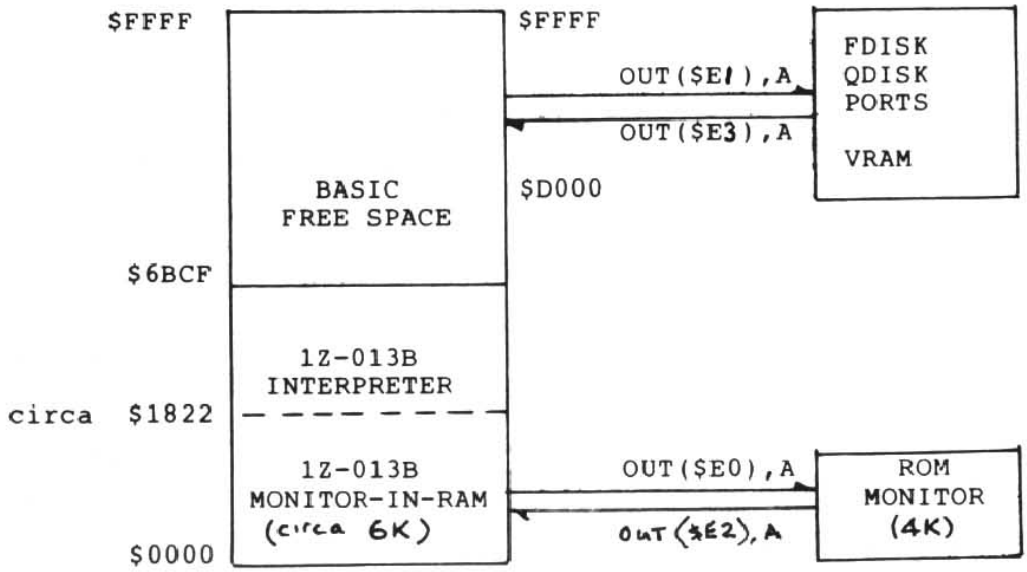
THE MEMORY MAP AFTER LOADING A PROGRAM

All machine-code programs load at $1200 or higher. If the program is meant to run under the ROM Monitor, it executes and runs where it loads and leaves the ROM switched in. In this case the memory map remains as at switch-on, except that if the program is specifically written for the MZ-700, it may contain commands to bank-switch the area $D000-$FFFF, in order to take advantage of the full 64K memory space.

If the program contains its own Monitor or operating system and is designed to run from an address below $1200, things are more complicated. The program still loads at $1200 initially, but then the disk-booting eprom (or the program's own built-in initialising code high in memory) will switch out the Monitor ROM at $0000 and replace it with 4K of RAM. Once this has been done, the program is copied down to its proper start location (usually $0000 but it can be some other address such as $0100). Finally the program executes at its 'proper' start address; as it does so, it may deliberately erase any traces of itself high in memory. When running, it leaves the ROM monitor permanently in 'limbo', and bank-switches the area $D000-$FFFF as required - the commands to do this are:-

```
OUT($E0),A to switch Monitor ROM OUT
OUT($E3),A to switch VRAM etc.  IN
OUT($E1),A to switch USER RAM   IN
```

A typical memory map after loading a Sharp Basic is:-



THE BASIC DIFFERENCES BETWEEN THE MZ-700 MONITORS

The MZ-700 Monitor-in-ROM occupies $0000-$0FFF, is standard, and is virtually 100% compatible with the MZ-80K Monitor-in-ROM i.e. most of its routines are at the same addresses as on the MZ-80K.

The MZ-700 Monitors-in-RAM are much larger programs, and are NOT standard i.e. each Basic has its own version, running up to about $1800. None of them are compatible with the MZ-80K Monitor-in-ROM.

USING THE MZ-700 MONITOR-IN-ROM

The keyboard is relatively primitive; it does NOT repeat, and there is no SHIFTLOCK. NOTE THE SERIOUS BUGS IN THE 'S' COMMAND:-

```
SHIFT/3(CR) i.e. '#'(CR) - restarts Basic (same as CTRL+RESET)
B(CR)       toggles the keyboard BLEEP on/off
JXXXX(CR)   JUMP to hexadecimal address XXXX
L(CR)       LOAD the first program found on a tape
MXXXX(CR)   MODIFY memory starting at XXXX; the byte there can be
            modified by entering 2 hex digits(CR), after which 'M'
            steps to the next location. Exit with SHIFT/BREAK.

P.....(CR)  sends commands or data to the PLOTTER/PRINTER ONLY:-

PABC prints the letters ABC          P&T prints a test pattern
P&S  text mode, 80 chars/line        P&L text mode, 40 chars/line
P&G  graphics mode                   P&C next pen colour

SXXXXYYYYEEEE(CR)  SAVE memory to tape. NOTE NO SPACES ALLOWED!!!
                   XXXXYYYYEEEE = START, END, EXECUTE addresses
```

S then prompts FILENAME?(SPACE). If you don't DELETE the SPACE it becomes the 1st char. of the name subsequently typed in. Worse still, if the name is more than 16 char. IT IS ACCEPTED WITHOUT AN ERROR MESSAGE and overwrites the 'FILESIZE' buffer!!

```
V(CR)       VERIFIES memory against first file found on tape
```

USING THE MZ-700 MONITORS-IN-RAM

There are two MZ-700 RAM Monitors - one for 1Z-013, and one for disk Basics 2Z-009 and 5Z-008. The latter require filenames in the form DEVICE:NAME (no quotes!) and thus can talk to non-default devices. Both versions give a repeating keyboard, normally in upper-case. SHIFT, GRPH and ALPHA work as expected, and there is a 'hidden' SHIFTLOCK (CTRL+E, cancelled by CTRL+F). The commands are all single-letter, but some require data to follow; this data may include spaces as shown below, but they are not essential:-

```
DXXXX YYYY(CR)      DUMP area XXXX YYYY to screen; hold with BREAK

FXXXX YYYY CD 00 12(CR)    FIND CD 00 12 in area specified
FXXXX YYYY ;C ;A ;T(CR)    FIND 'CAT' in area specified

GXXXX(CR)           CALL subroutine at specified hex address

LXXXX:FILENAME(CR)  LOAD program and put at address XXXX
(XXXX is optional; in disk Monitors FILENAME can be e.g. CMT:TEST)

MXXXX(CR)   MODIFY memory (works as in ROM Monitor, see above)
P(CR)       PRINTER toggle (applies only to D and F commands)
R(CR)       RETURN to calling program (usually BASIC)

SXXXX YYYY EEEE:FILENAME(CR) SAVE area XXXX YYYY, to load at XXXX,
execute at EEEE; in disk Monitors FILENAME can be e.g. CMT:TEST

TXXXX YYYY ZZZZ(CR)  TRANSFER (BLOCK COPY) XXXX YYYY to ZZZZ up
VFILENAME(CR)        VERIFY 'PROGAM' on tape against memory
```

THE MEMORY-MAPPED DISPLAY ON THE MZ-700

The VRAM stores can store 50 lines, each of 40 characters, at $D000-$D7FF, plus 50 lines of colour information at $D800-$DFFF. If you are using a monochrome display, the upper half of VRAM is not really needed, but the information there, if changed by colour commands, will affect the tones on the display.

The standard 25-line display acts as a 'window' on the 50 lines stored, and may be scrolled by SHIFT plus the UP or DOWN cursor key. However, the data is also scrolled in the VRAM, so that the top L.H. corner of the screen is ALWAYS stored at $D000.

The MZ-700 Manual states that the commands PEEK@ and POKE@ will address VRAM. This is wrong; the normal commands PEEK and POKE may be used to address VRAM, and the modified commands PEEK@ and POKE@ will address the USER RAM above $D000.

SHARP DISPLAY CODES ON THE MZ-700

MZ-700 DISPLAY codes are almost identical to the MZ-80A DISPLAY codes shown in APPENDIX A-3. The only difference is that code $F0 is a SPACE on the MZ-700.

SHARP ASCII CODES ON THE MZ-700

MZ-700 SHARP ASCII codes are almost identical to the MZ-80A SHARP ASCII codes shown in APPENDIX A-4. The only differences are that, on the MZ-700, code $6C is a diagonally hatched square and code $7F is the 'PRESS PLAY' symbol.

This means that, for virtually all practical purposes, a Sharp printer fitted with an MZ-80A character ROM will reproduce the MZ-700 character set satisfactorily.

THE MZ-700 USER-DEFINABLE KEYS

The 5 blue keys above the main keyboard are user-definable, and are normally set by a program as it loads. They may be changed by the user, with the Basic command DEF KEY (N) where N = 1-10.

USING PRINTERS WITH THE MZ-700 BUILT-IN INTERFACE

The MZ-700 was the first Sharp home computer to appear in the U.K. with a built-in printer interface; but it is designed for Sharp printers (and the Sharp plotter/printer), and cannot drive a standard 'Centronics' printer without modifications. Fortunately, the modifications are simple, see APPENDIX B-2 of this Manual.

THE DISADVANTAGES OF THE MZ-700

Considering its size, the MZ-700 is a remarkable machine in many ways, and has no serious 'bugs'. But given the built-in facility of a full 64K RAM, it is a pity that the original machines did not have an 80-column option, to run CP/M. However, the SUC can supply an 80-column kit at a low price, and have also modified the MZ-80A version of CP/M to run on the MZ-700. With 80 columns fitted, and with its printer interface modified to run a standard 'Centronics' printer, the MZ-700 should take on a new lease of life.

THE MEMORY MAP OF THE MZ-800

Finally we come to the MZ-800, the last Sharp 'home' computer to be sold in the U.K. It resembles the MZ-700 externally, and like that machine it has 64K of user RAM, provides a coloured display on a colour TV or a separate RGB Monitor, and can run Quickdisks. There the similarity ends; the MZ-800 has a 40/80-column screen, bit-mapped graphics (like the MZ-80B only more so), a 3-channel music chip, and a built in printer interface which can be set to 'Centronics'. Last, but not least, the whole machine can be set to behave like an MZ-700, at the flick of a single switch!

In addition to the main 64K user RAM, there is a separate 16K VRAM which holds enough data for a 320 x 200 4-colour display; an optional EXTRA 16K VRAM may be added, to increase the resolution to 640 x 200 (or to increase the number of colours to 16). The two 16K VRAMS are memory-mapped, in parallel, at $8000-$BFFF.

There is also a 16K ROM chip, addressed as if it were 3 separate chips; an MZ-800 Monitor ROM/IPL combined (8K), an MZ-700 Monitor ROM (4K), and a Character Generator (4K). The MZ-800 ROM/IPL is memory-mapped at $E010-$FFFF; the MZ-700 Monitor is memory-mapped at $0000-$0FFF; and the CG ROM is memory-mapped at $1000-$1FFF.

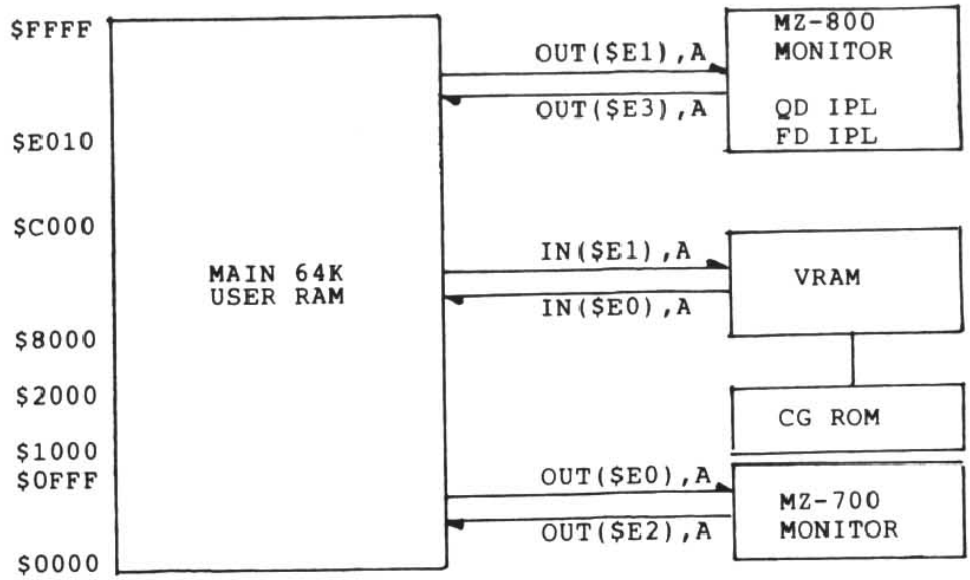
A small but important point to note is that the combined Monitor and IPL section of the 16K ROM is mapped from $E010-$FFFF and not from $E000-$FFFF. This trick leaves room for memory-mapped I/O ports at $E000-$E00F when the machine it in its MZ-700 Mode. (In the MZ-800 mode, I/O ports are addressed direct).

This combined Monitor and IPL is most interesting. It contains a traditional Sharp 'Monitor' (9Z-504M), AND disk-booting routines for floppy disks AND Quickdisks. In this respect (as in very many others) the MZ-800 is closely related to the MZ-80B, which also has its disk-booting routines in its IPL. Therefore, disk cards on the MZ-80B and the MZ-800 do NOT carry boot eproms. On the other Sharp machines covered by this Manual the disk-booting routines are in ROM on the disk interface cards. One result of all this is that disk cards for the MZ-700 and MZ-800 are not interchangeable.

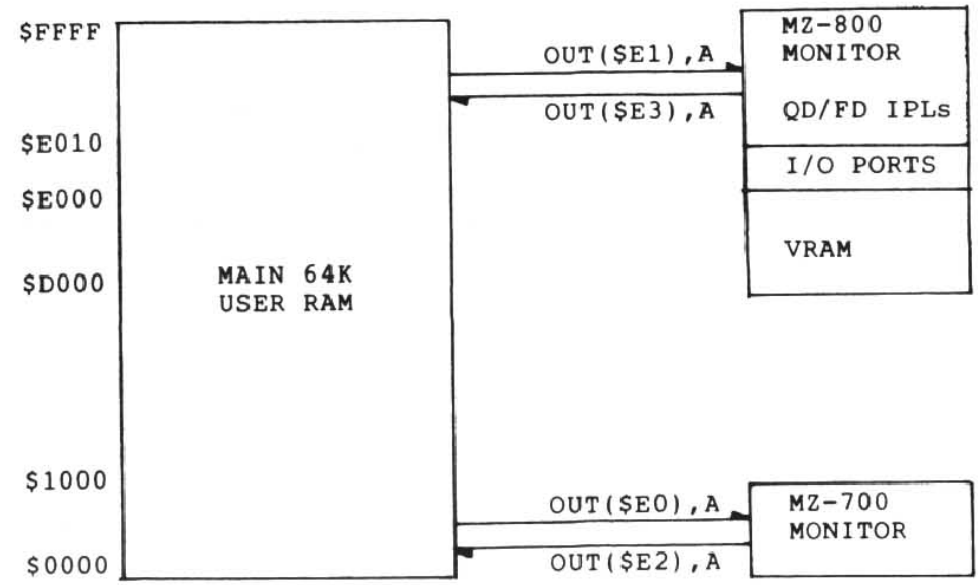
All these extra bits of RAM and ROM are bank-switched in and out of the main 64K memory map as required. The resulting permutations and combinations (22 in all) are brilliantly explained on 3 pages of the MZ-800 SERVICE Manual (Part No.00ZMZ800////E). Any MZ-800 owner who hasn't got this publication is advised to get it at once from TEEGA AGENCIES, Byerden Mill, Martin Street Burnley BB10 1SH.

Obviously it is impossible to reproduce all or even most of the maps here, but on the next page we reproduce our own versions of the two most important maps - the MZ-800 in its 'normal' mode, and the MZ-800 in its 'MZ-700' mode.

THE MEMORY MAP OF THE MZ-800 IN ITS 'NORMAL' MODE'



THE MEMORY MAP OF THE MZ-800 IN ITS 'MZ-700' MODE



In both cases, all the blocks of Monitor ROM, VRAM, IPLs etc. can be switched INTO the 64K memory space with OUT($E4),A. The MZ-700 'mode' is not exactly the same as an MZ-700, because it uses the 800 Monitor/IPL ROM to control disk drives.

As on the MZ-700, both types of disk drive may be connected at the same time. To get the second card into the bottom slot of the MZ-800 I/O bus, open the case and remove both I/O connectors by gently rearranging the wiring. Once the I/O connectors are outside the case, both cards may be inserted. Their ports do NOT clash.

THE FUNCTION OF THE IPL PROGRAM

At switch-on the MZ-800 IPL controls program loading. If there are no disk drives, it offers two options; 'C' for cassette, and 'M' for Monitor. To load a machine-code tape program, insert the tape and press 'C'. To enter the ROM Monitor (9Z-504M) press 'M'.

If there is one type of disk drive fitted (5.25" or QD) and the card is in the TOP slot (i.e the one normally accessible), this drive boots automatically if it contains a disk at switch-on, and no message appears. If the drive is left empty, the IPL offers a third option covering the type of disk drive connected i.e. it will offer options 'F', 'C' and 'M', or options 'Q', 'C' and 'M'. To load a program off disk, insert the disk and press 'F' or 'Q'.

If both types of disk drive are fitted, the one that is connected in the top slot takes preference as far as automatic booting is concerned. If no disks are in the drive at switch-on, then all four options are offered i.e. 'F', 'Q', 'C' and 'M'.

THE VIDEO DISPLAY ON THE MZ-800

The standard MZ-800 can display 320 x 200 dots in 4 colours, or 640 x 200 dots in 2 colours. The optional extra 16K VRAM enhances the display to 320 x 200 dots in 16 colours, or 640 x 200 dots in 4 colours. COLOURS are selected by storing them in coded PALETTES. There can be 40 or 80 columns, and there are 4 display 'MODES', each of which allows the appropriate number of palettes:-

MODE	RESOLUTION	COLUMNS	PALETTES	PALETTE CODES
1	320 X 200	40	4	0-3
2	320 X 200	40	16	0-15
3	640 X 200	80	2	0-1
4	640 x 200	80	4	0-3

COLOUR CODES are used to store colours in palettes; these are also in the range 0-15, colours 8-15 being 'light' e.g.'Lblue':-

(0)black blue red magenta green cyan yellow white(7)
(8)grey Lblue Lred Lmagenta Lgreen Lcyan Lyellow Lwhite(15)

On a standard machine (1 x 16K VRAM), MODES 2 and 4 may NOT be used, and the initial settings of MODES 1 and 3 are:-

```
MODE 1: Palette 0 = Colour 0 (black)
        Palette 1 = Colour 1 (blue)
        Palette 2 = Colour 2 (red)
        Palette 3 = colour 15 (light white)

MODE 3: Palette 0 = background colour 0 (black)
        Palette 1 = foreground colour 15 (light white)
```

MODE is changed by INIT"CRT:Mn", and a palette is 'refilled' by PALp,c. Any INIT"CRT:Mn" is usually followed by PALp,c commands, because a MODE change on its own restores the initial settings:-

```
INIT"CRT:M3":PAL0,14:PAL1,4  (set 80-col, green on Lyellow)
```

In initial MODE 2, COLOURS 0-15 are in PALETTES 0-15, which are grouped into 4 BLOCKS of 4. See Owner's Manual for more detail.

USING THE MZ-800 MONITOR-IN-ROM

This has no key-repeat and no SHIFTLOCK. It gains the commands QD, QL, & QS when a QD unit is connected. Note that in D, G, J, and M, SPACES are NOT permitted BEFORE or BETWEEN addresses:-

```
CTRL/RESET  restarts Basic
B(CR)       toggles the keyboard bleep on/off
DXXXXYYYY(CR) Dump specified area to screen; hold with SPACE bar
GXXXX(CR)   CALL subroutine at hexadecimal address XXXX
JXXXX(CR)   JUMP to hexadecimal address XXXX
L(CR)       LOAD the first machine-code program on tape
MXXXX(CR)   MODIFY memory starting at XXXX; the byte there can be
            modified by entering 2 hex digits(CR), after which 'M'
            steps to the next location. Exit with SHIFT/BREAK.

QD(CR)      display QUICKDISK DIRECTORY
QL(CR)      LOAD the first programme on Quickdisk
QS(CR)      SAVE program on QUICKDISK (prompts as for 'S' below)

S(CR)       SAVE program on tape; prompts separately for FILENAME,
            TOP ADDRESS (i.e. START!), END and EXECUTE addresses.

VFILENAME(CR)       VERIFIES memory against 'FILENAME' on tape
```

USING THE MZ-800 MONITORS-IN-RAM

The 3 Basics (tape, FD, QD) share a common syntax, so the 3 RAM Monitors all behave in the same way i.e. filenames are required in DEVICE:NAME format (without quotes), but the default DEVICE may be omitted. The keyboard auto-repeats and is normally in upper-case; SHIFT, GRPH and ALPHA all work as designed and there is a 'hidden' SHIFTLOCK (CTRL+E, cancelled by CTRL+F). Single-letter commands are used, but most of them require data to follow; this data may include spaces as shown below, but the spaces are not essential:-

```
DXXXX YYYY(CR)  DUMP specified area to screen; hold with BREAK

FXXXX YYYY CD 00 12(CR)     FIND CD 00 12 in area specified
FXXXX YYYY ;C ;A ;T(CR)     FIND 'CAT' in area specified

GXXXX(CR)   CALL subroutine at specified hex address

LXXXX:DEVICE:FILENAME(CR)   LOAD 'FILENAME' from DEVICE, and store
                            at XXXX. The load address XXXX and DEVICE:
                            are optional, and FILENAME may be omitted if
                            CMT: is already the default device.

MXXXX(CR)   MODIFY memory (works as in ROM Monitor, see above)
P(CR)       PRINTER toggle (applies only to D and F commands)
R(CR)       RETURN to calling program (usually BASIC)

SXXXX YYYY EEEE:DEVICE:FILENAME(CR)     SAVE memory area XXXX-YYYY
                            to DEVICE:, with name 'FILENAME' and with
                            LOAD address XXXX and execute address EEEE.
                            Options and omissions as for the L command.

TXXXX YYYY ZZZZ(CR)  TRANSFER (BLOCK COPY) XXXX YYYY to ZZZZ up

VPROGRAM(CR)        VERIFY 'PROGAM' on tape against memory
```

SHARP DISPLAY CODES ON THE MZ-800

As on the MZ-80B, PEEK and POKE address themselves to the main 64K USER RAM, and they cannot address VRAM. Therefore DISPLAY CODES are NOT used on the MZ-800.

SHARP ASCII CODES ON THE MZ-800

MZ-800 SHARP ASCII codes are identical to those on the MZ-700 and these, as already explained, are with one exception the same as the SHARP ASCII codes for the MZ-80A (see APPENDIX A-4).

This means that, for virtually all practical purposes, a Sharp printer fitted with an MZ-80A character ROM will reproduce the MZ-800 character set satisfactorily.

THE MZ-800 USER-DEFINABLE KEYS

The 5 blue keys above the main keyboard are user-definable, and are normally set by a program as it loads. They may be changed by the user, with the Basic command DEF KEY (N) where N = 1-10.

USING PRINTERS WITH THE MZ-800 BUILT-IN INTERFACE

The MZ-800 has a built-in 'Centronics' printer interface as an option. However, switching to 'Centronics' does NOT guarantee that your printer gets the correct signals, and it does NOT switch in a code-changing ROM, so the required SHARP/STANDARD ASCII conversion of control and lower-case codes must be done by software. The problems are explained hilariously in the MZ-800 SERVICE MANUAL:-

> 'When the MZ-800 dedicated printer is used, there may be such a case that proper operation is not attained due to different printing characters and control codes. It must also be noted that all MZ-800 characters cannot be printed. Besides, connection with a Centronics compatible printer may not be permitted hardware-wise, sometimes.'

MZ-800 PCP/M copes easily with this situation; its SETUP.COM program allows the user to specify Sharp or Standard ASCII character codes to the printer, and to send $0D or ($0D+$0A) at the end of a line. As a matter of interest, this and all other versions of CP/M for Sharp machines work internally in standard ASCII. If you have a Sharp printer, they convert standard ASCII to Sharp ASCII before sending stuff for hard copy!

Problems do arise in the MZ-800 Basics, which contain printer routines intended for Sharp printers. In the not too distant future the SUC hope to supply 'Centronics' programs for all MZ-800 Basics, similar to the programs which have already been written for all the standard Basics on the earlier machines.

END OF MAIN SECTION A (APPENDICES ARE AT END OF MANUAL)

The Development of Sharp MZ Basics 1980 - 1984

By John Ibberson and Maurice Hawes

INTRODUCTION AND SCOPE

The main aim of this section is to provide a single, comparative source of information covering all the 'standard' Sharp Basics on MZ-80K/B/A/700/800 home computers. By 'standard' we mean the 8 s.f. Basics supplied free with new machines and disk drives, the special SB-6511 Basic supplied with MZ-80B Serial I/O cards, and the 16 s.f. 'double precision' disk Basics offered by Sharp (U.K) as optional extras for the MZ-80K and MZ-80B only.

In other words, we have not included Basics such as SP-5060, SP-7010, Hu-Basic, SBM-Superbasic, K&P DBasic etc., which arrived in the U.K. from sources overseas; nor have we mentioned the various toolkits which were written by independent U.K. software houses to cover the inadequacies of many early Sharp Basics. All these 'other' Basics and Basic toolkits will be dealt with in section C of this Manual. To clarify this point, here is a chronological list of the Basics that are dealt with in this section:-

MZ-80K Tape: SP-5025 (early 1980)
MZ-80K Disk: SP-6015 (8 s.f.)
SP-6115 (16 s.f. 'double precision')

MZ-80B Tape: SB-5510 (early 1981)
MZ-80B Disk: SB-6510 (8 s.f.); SB-6511 (SB-6510 plus Serial I/O)
SB-6610 (16 s.f. 'double precision')

MZ-80A Tape: SA-5510 (late 1981)
MZ-80A Disk: SA-6510 (early 1982)

MZ-700 Tape: 1Z-013 (1983)
MZ-700 Disk: 2Z-009 (5.25"); 5Z-008 (Quickdisk)

MZ-800 Tape: 1Z-016 (1984)
MZ-800 Disk: 2Z-046 (5.25"); 5Z-009 (Quickdisk)

An important secondary aim is to provide the bare essentials of an instruction Manual for any one of the Sharp Basics listed above, since these Manuals are no longer available.

THE MAIN ADVANTAGES OF SHARP BASICS

All the above Sharp Basics have full-screen editing, and tolerate spaces between keywords but do not insist on them, except in a few special cases in the later Basics (AND, OR, MOD, NOT, XOR). In addition, variables are NOT lost when a programme is edited; you may therefore BREAK out of a programme, edit a line, and then jump back into the revised programme with all variables intact, by using a command such as GOTO NN or RUN NN (where NN = line number). These features make the editing and debugging of Sharp Basic programmes much easier than on many other machines. Furthermore, the arithmetic accuracy of 8 s.f. in all the 'free' Basics is two decimal places more than on most home computers of the same vintage.

THE MAIN DISADVANTAGES OF SHARP BASICS

Most of these Sharp Basics (i.e. all except those for the MZ-80B) use a VERY idiosyncratic, non-sequential set of 'Sharp ASCII' codes for lower-case letters. In addition, the Basics for the MZ-80K/B/A are hopeless at string comparisons, which are impossible on the MZ-80K, and are done with length-based priority on the MZ-80B/MZ-80A.

The MZ-80K/B/A Basics also lack a RENUMBER command. Whatever the purists may say, this is a serious disadvantage in practice.

The later Basics (MZ-700/800) include RENUM, and perform string comparisons on a letter-by-letter basis; but even in those Basics, any attempt at a lower-case sort is thwarted by the non-sequential lower-case 'Sharp ASCII' codes!! These peculiar lower-case codes also make life difficult if you wish to use a standard ASCII printer on the MZ-80K/A/700/800, or if you wish to transfer a programme between any one of those machines, and the MZ-80B.

TOKENISATION OF SHARP BASIC KEYWORDS

All Sharp Basics store their prorammes in memory, and on tape and disk, in a condensed form. This form is achieved by 'tokenising' the keywords in a programme, so that each is represented by one or two bytes in the range $80-$FF. Unfortunately, Sharp used one set of tokens for the MZ-80K Basics, another set of tokens for the MZ-80A/B Basics, and a third set of tokens for the MZ-700/800 Basics. As a result, the transfer of a programme from one Sharp Basic to another often requires the conversion of keyword tokens; and although this is not particularly difficult, Sharp rarely provided this facility.

TOKEN CONVERSION

Token conversion is not required when transferring a programme from one Basic to another on the same machine; nor is it required when transferring programes between the MZ-80A and the MZ-80B; nor between the MZ-700 and the MZ-800. In all other cases conversion is required. Other factors may also come into play, as discussed below.

COMPATIBILITY BETWEEN DIFFERENT BASICS ON THE SAME MACHINE

There is complete compatibility between the tape Basic and the disk Basic(s) on a given machine. This is because the tape Basic always uses a subset of the keywords and tokens in the disk Basic(s), and because all the disk Basics include commands for loading and saving programmes on tape.

The same general situation exists when a machine has more than one 'standard' disk Basic; such Basics always share a large common set of keywords/tokens, and this, in theory, should make programmes easily transferrable. However, on the MZ-80K, the two disk Basics SP-6015 and SP-6115 use DIFFERENT DISK DIRECTORY LAYOUTS; this means they cannot read one another's (otherwise largely compatible) programmes off disk, so transfers must be done via tape (!!). On the MZ-80B, the three disk Basics CAN read one another's programme files with no problems. On the MZ-700 and MZ-800, tape may be used to transfer programmes from 5.25" disk Basic to Quickdisk Basic.

COMPATIBILITY BETWEEN BASICS ON DIFFERENT MACHINES

The problems due to different token sets, discussed above, are compounded by several other factors. The MZ-80B uses a higher tape Baud rate than the other machines; the MZ-80K uses a different disk format; and the 'ASCII' set of the MZ-80B, being standard, is completely different from the 'ASCII' sets of the other machines!

Token-converting programmes have been written for all likely situations, but only three of them will be discussed here, to cover the principal areas of difficulty. Note, however, that NO token-converting programme can deal with machine-specific PEEKs, POKEs and USRs etc., so these must be checked and changed manually!!

Given the common tape format used on the MZ-80K/A/700/800, their Basics can read one anothers' programme tapes, but in most cases will not convert any tokens. However, MZ-700 tape Basic 1Z-013 will automatically convert MZ-80K tape Basic tokens to their MZ-700 (and MZ-800) counterparts. To convert MZ-80K programmes to run on the MZ-80A, the most convenient method is to use SA-5510 + BAS MOD, described in section C of this Manual.

Given the common disk format used by the MZ-80A/B/700/800 disk Basics it is theoretically possible for all these Basics to read one another's programmes; but in practice, this ideal is not achieved. However, all the MZ-80A Basics use subsets of the MZ-80B Basic keywords; so an MZ-80A programme saved on disk will load and run under an MZ-80B disk Basic; but an MZ-80B programme saved on disk will only run under MZ-80A disk Basic if it avoids keywords unique to the MZ-80B. This MZ-80A/B compatibility is also compromised if a programme contains messages or REMs in lower-case. MZ-700 disk Basic programmes may be run on the MZ-800 in its MZ-700 mode.

THE SPECIAL ROLE OF MZ-700 K&P DISK BASIC

Although it is really outside the scope of this section, we feel that we must mention the MZ-700 disk Basic from Kirsten & Partners in West Germany, which uses the Sharp MZ-80A/B/700/800 disk format, and has an automatic token-converter for MZ-80A/B Basic programmes. Therefore 'K&P DBASIC' (as it is called) provides a crucial 'missing link', facilitating the transfer of MZ-80A/B programmes to the MZ-700/800. It is described fully in section C of this Manual.

VARIANTS, 'QUIRKS' and 'BUGS' IN SHARP BASICS

Many Sharp Basics have variants with the same name and/or part number, and in some there are minor 'quirks' or even 'bugs'. These aberrations are summarised below (for details see Appendix B-1):-

SP-5025	at least 3 versions exist, and they all have peek protect, auto-run, and anti-list flags, and a 'bug' in TI$.
SP-6015	has a peek protect flag and a dormant 'bug' in PRINT/P TAB
SP-6115	has 2 arithmetic 'bugs' (the same as in SB-6110)
SA-6510	at least 3 versions exist
SB-5510	at least 2 versions exist
SB-6610	has 2 arithmetic 'bugs' (same as in SP-6115)
1Z-013	has a line length 'bug'

BASIC 'COLD STARTS', 'WARM STARTS', and TESTING

When a Sharp Basic is first loaded, it executes what is known as a 'cold start'. This clears RAM from the end of the interpreter to the top of the program area, sets essential pointers, displays the opening titles and the free (program) area, and then says 'READY'.

In the Basics for the MZ-80K/B/A, the size of the free area is actually measured during a 'cold start' and is then printed on the screen. But in MZ-700/800 Basics the free area is worked out in advance, and the figure is coded permanently into the 'cold start' screen announcements. The free area may be checked by doing PRINT SIZE as soon as Basic has loaded; in some Basics (e.g. MZ-700 tape Basic 1Z-013) the 'announced' free area is in error!

If you leave Basic for any reason you can often (but not always) restart it by jumping to its 'cold start' address; however, it is more useful to restart it by jumping to its 'warm start' address. A 'warm start' cuts out initialising routines and announcements, and leaves any Basic program (and its data) intact so that it may be reentered as if you had never left, with 'RUN nn' or 'GOTO nn'.

To load a machine-code utility high in memory, set a suitable LIMIT, then RESET to the Monitor and load the utility from tape, as this allows a subsequent 'warm' restart. Disk users should note that, if the utility is loaded off disk with 'RUN', the current 'Basic' program is erased and a 'warm' restart is IMPOSSIBLE.

The 'cold start' and 'warm start' addresses for various Basics are shown below. Note that with MZ-700/800 Basics a 'cold restart' must be done from tape or disk as it uses bank-switching routines and screen messages which are not stored permanently in RAM. To compensate, a 'warm restart' is very easy - just do CTRL + RESET:-

MACHINE	BASIC	COLD START JUMP ADDRESS	WARM START JUMP ADDRESS
MZ-80K	SP-5025	$1200	$124B *
MZ-80K	SP-6015	$21FA	$2245
MZ-80K	SP-6115	$1200	$1247
MZ-80A	SA-5510	$1200	$1250
MZ-80A	SA-6510	$12A0	$1300
MZ-80B	SB-5510	$1220	$1280
MZ-80B	SB-6510	$1220	$1280
MZ-80B	SB-6511	$1220	$1280
MZ-80B	SB-6610	$1220	$1280
MZ-700	1Z-013	RELOAD BASIC	CTRL+RESET **
MZ-700	2Z-009	RELOAD BASIC	CTRL+RESET **
MZ-700	5Z-008	RELOAD BASIC	CTRL+RESET **
MZ-800	1Z-016	RELOAD BASIC	CTRL+RESET
MZ-800	2Z-046	RELOAD BASIC	CTRL+RESET
MZ-800	5Z-009	RELOAD BASIC	CTRL+RESET

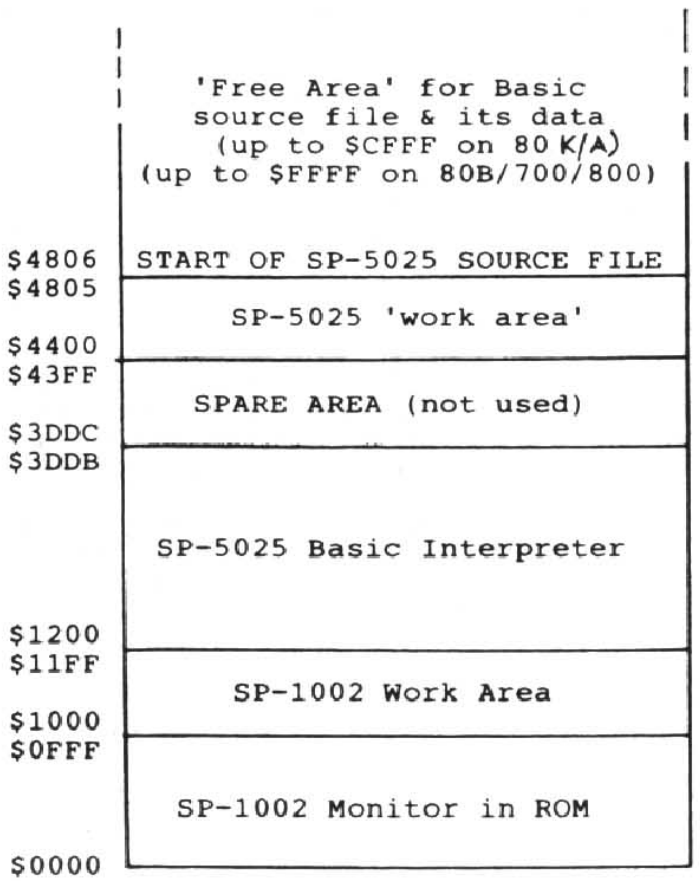
* $124D in the version that comes on an SP-6115 master disk!!

** On the MZ-700 a Basic 'warm start' is also possible, from the MZ-700 ROM monitor, by keying-in SHIFT/3 (i.e. '=') then CR.

'FREE' AND 'SPARE' AREAS, and START OF PROGRAM FILES

The 'free area' in a particular Basic is the area (ABOVE the end of the interpreter) which is available for a Basic program and its data. In the MZ-80K and MZ-80A, various bits of hardware (such as the VDU, keyboard, tape deck, disk drives) are addressed at $D000 upwards, so the 'free area' stops at $CFFF. But in the MZ-80B and the MZ-700/800 these hardware items are banked-switched out when not in use, so the 'free area' runs up to $FFFF (or close to it).

However, the early Sharp Basics contain at least one unused area BELOW the 'free' area. In the MZ-80K/B/A Basics, this 'spare' area lies between the end of the interpreter code proper, and the area being used by the interpreter as its 'work space'. In this context, the first MZ-80K Basic (SP-5025) was typical, and a diagram is worth a thousand words:-



In the above case the area $3DDC - $43FF is SPARE, and may be used for permananent additions to the interpreter. The original SP-5025 Basic for the MZ-80K lacked many commands which today would be considered essential, and many software houses wrote toolkits for it (see Section C of this manual). They were all able to claim that they improved SP-5025 'without increasing its size'. This apparently magic trick was in fact only possible because they put the extra code in the SPARE area!!

The 'SPARE AREAS' areas in all the relevant Sharp basics are listed below. The MZ-80A and MZ-80B Basics, like those for the MZ-80K, have a large 'spare area' between the interpreter proper and its work area; and the MZ-80A Basics have another, smaller, 'spare area' in the interpreter itself, between PAGE/P and COPY/P. We think that this area was reserved for IMAGE/P, which is located between PAGE/P and COPY/P in the MZ-80B Basics. However, it seems that, after a struggle, Sharp gave up trying to implement IMAGE/P on the MZ-80A; the evidence for this is that the codes in the tape and disk versions differ, and neither is implemented.

MZ-700 tape Basic 1Z-013 contains a large spare area at the end of the keyword table. All subsequent MZ-700/800 Basics (MZ-700 disk Basic 2Z-009E, MZ-700 Quickdisk Basic, and all MZ-800 Basics) do not have significant spare areas; this may be because they are notable amongst Sharp Basics for having all the commands one might expect, and do not need expansion.

The Basic source file starts at some address higher than the actual end of the interpreter code; this is to allow space for the interpreter buffers and pointers:-

MACHINE	BASIC	SPARE AREA START	SPARE AREA FINISH	SOURCE FILE START
MZ-80K	SP-5025	$3DDC *	$43FF	$4806
MZ-80K	SP-6015	$560D	$5FFF	$652C
MZ-80K	SP-6115	$561F	$567F	$5F0D
MZ-80A	SA-5510	$4232	$48FF	$505C
(? Unused	IMAGE/P)	$32FC	$33AA	
MZ-80A	SA-6510	$57F2 **	$5FFF	$675C
(? Unused	IMAGE/P)	$4560	$4623	
MZ-80B	SB-5510	$496A	$49BF	$511C
MZ-80B	SB-6510	$5B1A	$5CFF	$675C
MZ-80B	SB-6511	$68AB	$6AFF	$725C
MZ-80B	SB-6610	$5B1A	$5CFF	$68DD
MZ-700	1Z-013	$2CA9	$30A8	$6BCF
MZ-700	2Z-009	***	***	$85FE
MZ-700	5Z-008	***	***	$85FF
MZ-800	1Z-016	***	***	$A3FA
MZ-800	2Z-046	***	***	$A473
MZ-800	5Z-009	***	***	$A471

* $3DDC is for SP-5025 on tape. For the SP-5025's on MZ-80K master disks, read $3DEC (SP-6015), or $3ED6 (SP-6115).

** $57F2 is for the original MAR.15,'82 version of SA-6510; for the later versions this value must be raised to $5802.

*** MZ-700 disk Basic 2Z-009 does have two useful (if small) spare areas at ($17A6-$17FF) and ($3FDF - $3FFF). To the best of our knowledge, the remaining Basics in this group do not contain significant spare areas. However, most Basics contain bits of unused code, and you can always shorten a long-winded message. If spare areas come to light in any of these Basics, they will be published in the Magazine, and you can fill in the gaps.

DATA FILING IN SHARP BASICS

DATA FILES ON TAPE

Data may be filed on tape from any of the standard tape or disk Basics; but it is a slow process, principally because the data is recorded sequentially, in blocks of 256 bytes, and the tape is stopped and restarted between blocks. This makes data filing a very tedious job in any tape Basic; but for other more technical reasons, the MZ-80K Basics are slower than the later Basics.

Some special MZ-80K Basics and Basic toolkits, discussed in Section C of this Manual, include patches which speed up data filing without making the files unreadable by standard Basics. These modified data-filing routines are as reliable as the standard routines, and we recommend them; but they will not be discussed further in this section.

For some reason which has never been explained, tape data files have an element of unreliability in them, and it is quite common to find that a tape data file will not load at the first attempt, but will eventually load if you keep on trying. We think that the problem may be due to voltage fluctuations caused by the repeated stopping and starting of the tape motor between blocks of data; but we have never been able to find time to conduct tests to find out whether this is the cause, or not.

Because tape data files are saved in 'blocks', it is not easy to copy them. Most copying programmes, if asked to copy a tape data file, will copy only the first 'block' and then stop, thinking that the end of the file has been reached. Unfortunately, the otherwise admirable S.U.C. 'SUPERTAPE' program acts in this way.

DATA FILES ON DISK

All Sharp disk Basics can save data on disk. The process is much faster than with tape, and virtually 100% reliable. Disk data files can be handled on a 'sequential' basis, similar to tape data files. However, with such files, all the data must be read in to examine one item; and if any item needs changing, the whole file has to be resaved under a new name. With the large files which are practicable on disks, 'sequential' methods can be cumbersome.

It is therefore more usual to construct disk data files on a completely different basis, known as 'random-access'. In such files, it is possible to read in ONE data item, alter it, and resave it in the correct place on the disk without affecting the rest of the file. This is much more convenient, and much faster, than loading and resaving a complete 'sequential' file. But the coding for 'random-access' filing is, as you might expect, more complicated than for 'sequential' filing; the computer has to be told which item in the file is to be read or written, and the program has to keep a record, to enable this to be done correctly.

(Examples of data-filing routines follow)

EXAMPLES OF TAPE AND DISK DATA FILING ROUTINES

All data filing routines involve special commands, combined in a particular way to produce the desired result. Furthermore, in disk Basics, it is possible and frequently desirable to handle several files 'in parallel' in the same program. The best way to explain how data-filing commands do all these different jobs is to quote extended examples involving all the necessary keywords, and such examples are given below. Individual keywords are explained fully in the alphabetic list which follows these introductory notes.

1. DATA FILING IN MZ-80K/B/A/700 TAPE BASICS

1a) Writing to a new file

```
10 WOPEN "Filename"
20 PRINT/T N: REM N = number of array items to be saved
30 FOR NC = 1 TO N
40 PRINT/T A$(NC)
50 NEXT
60 PRINT/T X,Y,Z
70 CLOSE
```

1b) Reading from an existing file

```
10 ROPEN "Filename"
20 INPUT/T N: REM find out how many array items are in the file!
30 FOR NC = 1 TO N
40 INPUT/T A$(NC)
50 NEXT
60 INPUT/T X,Y,Z
70 CLOSE
```

2. TAPE DATA FILING IN MZ-80K/B/A DISK BASICS

The same as in 1a and 1b above, EXCEPT that ROPEN, WOPEN & CLOSE must be changed to ROPEN/T, WOPEN/T and CLOSE/T.

3. TAPE DATA FILING IN MZ-700 DISK BASICS AND ALL MZ-800 BASICS

3a) Writing to a new file

```
10 WOPEN#1, "CMT:Filename"
20 PRINT#1, N: REM N = number of array items to be saved
30 FOR NC = 1 TO N
40 PRINT#1, A$(NC)
50 NEXT
60 PRINT#1, X,Y,Z
70 CLOSE#1
```

3b) Reading from an existing file

```
10 ROPEN#2, "CMT:Filename"
20 INPUT#2, N: REM find out how many array items are in the file!
30 FOR NC = 1 TO N
40 INPUT#2, A$(NC)
50 NEXT
60 INPUT#2, X,Y,Z
70 CLOSE#2
```

4. SEQUENTIAL FILING TO THE DEFAULT DRIVE IN ALL DISK BASICS

4a) Writing to a new file

```
10 WOPEN#5, "Filename"
20 PRINT#5, N: REM N = number of array items to be saved
30 FOR NC = 1 TO N
40 PRINT#5, A$(NC)
50 NEXT
60 PRINT#5, X,Y,Z
70 CLOSE#5
```

4b) Reading from an existing file

```
10 ROPEN#8, "Filename"
20 INPUT#8, N: REM find out how many array items are in the file!
30 FOR NC = 1 TO N
40 INPUT#8, A$(NC)
50 NEXT
60 INPUT#8, X,Y,Z
70 CLOSE#8
```

5. RANDOM-ACCESS FILING TO DEFAULT DRIVE (5.25" DISK BASICS ONLY)

5a) Writing to a new file

```
10 XOPEN#57, "Filename"
40 PRINT#57(REC), A$,B$,X,Y,Z: REM (REC) = record number
70 CLOSE#57
```

5b) Reading from an existing file

```
10 XOPEN#26, "Filename"
20 INPUT#26(REC), A$,B$,X,Y,Z,: REM (REC) = record number
70 CLOSE#26
```

6. FILING TO A NON-DEFAULT DRIVE OR DEVICE

To file to a different drive or device, its specification must be included in the ROPEN, WOPEN or XOPEN statement, for example:-

In MZ-80K/B/A disk Basics: `10 ROPEN#74, FD1 "Filename"`

In MZ-700 disk/800 Basics: `10 WOPEN#75, "QD:Filename"`
or: `10 XOPEN#76, "FD1:Filename"`

7. A NOTE ON LOGICAL FILE NUMBERING

In MZ-80K/B/A/700 tape Basics only one file may be opened at a time; the 'open' file must be closed, before another is 'opened'. This also applies when using MZ-80K/B/A disk Basics to handle tape data files. In all these cases 'logical' numbers do not apply.

When handling tape data files under MZ-700 disk Basics or ANY MZ-800 Basic, or when handling disk data files under ANY disk Basic, it is possible to have up to 10 files open at any one time, using logical numbers in the range 1-127. Such 'logical' file numbers are prefixed by a '#' sign, and must be followed by a ',' separator where necessary (see the various examples above).

ARITHMETIC, COMPARISONS, AND SORTING

All Sharp Basics include the four standard arithmetic operators, +, -, *, and /, plus the 'up arrow' for exponentiation. Brackets are allowed, and stuff within brackets is evaluated first, with the normal order of priorities, which is:-

Exponentiation, Negation, *, /, +, -

In addition, all the MZ-800 Basics implement MOD, and '\' for integer division. In priority, these come between '/' and '+'.

In all Sharp Basics, '+' simulates logical (relational) OR, and '*' simulates logical (relational) AND. The full keywords OR and AND, which also allow bitwise comparisons, only appear in MZ-700 and MZ-800 Basics. The bitwise operators NOT and XOR are missing from 1Z-013, but appear in all other MZ-700/800 Basics.

For some reason the keywords MOD, OR, AND, NOT and XOR must have SPACES BEFORE AND AFTER (otherwise they are treated as variables).

All Sharp Basics include the comparison operators =, <>, ><, >=, =>, <=, and =< . But in all the 8 s.f. MZ-80K Basics, operators which include '<' and '>' only recognise NUMERIC variables; as a result, in these Basics, it is IMPOSSIBLE to sort strings.

In SP-6115, and all Basics for the MZ-80A/B, the operators '<' and '>' recognise string variables; but the comparison is done on a 'length before content' basis. Thus, for example, a 5-character string is always 'greater' than a 4-character string, whatever their respective contents. Therefore, in these Basics, the first step towards a correct 'alphabetic' sort is to pad all strings out to the same length, with spaces. This is sufficient to give a correct upper-case sort on all 3 machines, and a correct lower-case sort on the MZ-80B. But a lower-case sort on the MZ-80K/A has problems, due to the the weird Sharp lower-case 'ASCII' codes.

In the standard Sharp Basics for the MZ-700 and the MZ-800, the operators '<' and '>' recognise strings, AND compare them on a character-by-character basis. Therefore, in these Basics, correct alphabetic upper-case string sorting is straightforward; but a lower-case sort is still confused by the lower-case 'ASCII' codes.

In all cases, correct lower-case or even mixed-case sorts are possible by subterfuge, the simplest of which is to treat upper- and lower-case versions of the same letter as 'equal' during a sort. If this is done, a correct mixed-case sort may be got on the MZ-700 and MZ-800, without any further tricks. On the MZ-80A/B, it is also necessary to pad out all strings to the same length.

Fortunately, there are plenty of Basic toolkits which overcome the worst problems on the early machines. For MZ-80K Basics, there are several toolkits which allow string comparisons to be made (but those which do comparisons on a length before content basis are to be avoided). The 'BAS MOD' toolkit for SA-5510 offers alphabetic upper-case sorting on the MZ-80K and MZ-80A, without the need for padding. The 'XPATCH' toolkit for SA-5510 also offers this, and can also (optionally) be made to treat upper-case and lower-case as equal; but 'X-PATCH' only works on the MZ-80A. All these toolkits are described further in Section C of this Manual.

DIFFERENCES IN FLOATING-POINT ARITHMETIC

The 8 s.f. MZ-80K/B/A Basics use a Sharp-originated F.P. package with a magnitude range of 9.2E+18 down to 2.7E-20. The later 8 s.f. MZ-700/800 Basics revert to a more conventional F.P. package with the wider, normal magnitude range of 1.70E+38 down to 2.9E-39 (for more on this topic see PCW Jan.'85 pp.166-173). The Sharp 'double-precision' disk Basics (available for the MZ-80K/B only) use BCD techniques to give 16 s.f. with a magnitude range from 10E+78 down to 1E-48. These 16 s.f. packages are good at everyday calculations, and also offer PRINT USING, but they both contain two nastly little bugs which can affect scientific calculations; fortunately, these bugs are easily trapped (for full details see Appendix B-1).

SPECIAL PROBLEMS WITH ASCII PRINTERS

Sharp computers are designed to be used with Sharp-protocol printers, and all except the MZ-800 need special hardware (or wiring changes) to drive a standard-protocol ('Centronics parallel') printer. The MZ-800 has a built-in switch for this purpose.

We have already mentioned the non-standard lower-case character codes used on the MZ-80K/A/700/800; to use an 'ASCII' printer on one of these computers, special software or firmware is required to convert these codes from Sharp 'ASCII' to standard 'ASCII'. Sharp printers also use very non-standard control codes, so those parts of a Basic interpreter which are written specifically for such printers must be changed if you wish to run an ASCII printer. In addition, the CHR$ command in SP-5025, and in SP-6015, demands special attention. All these 'ASCII', control code and CHR$ changes are detailed in Appendix B-2.

BACKING-UP SHARP BASIC MASTER TAPES AND DISKS

Sharp used various strategies to discourage the copying of master tapes and disks; members of the S.U.C. have no problems here, as any tape Basic can be copied with SUPERTAPE 2 or CLUB COPY, and any disk Basic can be copied by its own disk-copying programme, once that programme has been modified to SUC instructions (see Appendix B-3).

NOTABLE CASES OF NON-STANDARD SYNTAX

In Sharp Basics, 'PRINT/P' and 'LIST/P' are used for hard-copy output (equivalent to 'LPRINT' and 'LLIST' in most other Basics).

'DELETE' in all the 5.25" disk Basics is used to delete files off a disk; but in many later Basics it also deletes programme lines.

'INIT' is used in MZ-700 5.25" Disk Basic to set the parameters of an (optional) RS-232 card; but 'INIT' in MZ-700 Quickdisk Basic is used to format a Quickdisk(!); and in ALL MZ-800 Basics, 'INIT' is used to set up ALL external peripherals, optional or otherwise!!

'SWAP' in Sharp floppy-disk Basics is used to switch between the current programme module and another module on disk. There is NO command in Sharp Basics for exchanging variables during a sort, so dummy variables must be used.

ERRORS AND OMISSIONS IN SHARP OWNER'S MANUALS

In their day, these Sharp computers were deservedly popular for their flexibility and reliability; but Sharp failed to do themselves justice in many of the Owner's Manuals. The early Manuals were written in very bad English; nearly every Manual omits at least one command; and some later Manuals omit several commands:-

The SP-5025 Manual does not mention INP# and OUT#

The SB-6510 Manual does not mention CHARACTER$

The SA-5510 & SA-6510 Manuals list CHARACTER$ but do not explain it!

The 1Z-013 Manual gives the wrong explanations for PEEK@ and POKE@; and it omits AND, BOOT, CLS, HEX$, JOY, OR, TROFF and TRON

The 2Z-009 Manual does not even mention PEEK@ and POKE@, omits the eight 1Z-013 keywords listed above, and also omits BEEP, CSRH, CSRV, EDIT, ELSE, FRAC, NOT, SPACE$, and XOR

The 1Z-016 Manual omits BEEP, EDIT, FRAC, HEX$, MOD and SPACE$

The 2Z-046 Manual omits the six 1Z-016 keywords listed above, and also omits ALL, CCOLOR, CRESET, and CSET.

The Quickdisk Basic Manuals on the MZ-700/800 cross-refer to the Manuals for 2Z-009 and 1Z-016 respectively, and in doing so pick up their errors. One notable result of all this confusion is that the MZ-700 disk Basics, and all the MZ-800 Basics, implement both of the keywords SPACE$ and SPC, which do the same job!!

THE MAIN KEYWORD INDEX

The following pages start with an index to ALL the keywords in ALL these Sharp Basics; this index lists the Basics in which each keyword is to be found, gives related keywords, and page references for the detailed explanations.

The index is followed by lists of detailed explanations of the keywords used in Sharp home computers; the first, main list excludes the hi-res graphics keywords in the MZ-80B/800 Basics, the plotter/printer keywords in the MZ-700/800 Basics, and the special 'Serial I/O' keywords in SB-6511. These are covered by supplementary smaller lists which follow.

Each entry in these lists explains the purpose of the keyword, and gives at least one example to illustrate the syntax of its use.

Appendices B-1 to B-3 deal with the various problem points mentioned briefly in the preceding pages of this section.

*****

KEYWORD	BASIC(S)	PAGE	XREF
REW	80B ONLY	50	FAST
RIGHT$	ALL	50	LEFT$,MID$
RLINE	700/800 (P/P ONLY)	61	RMOVE
RMOVE	700/800 (P/P ONLY)	61	MOVE,PMOVE
RND	ALL	50	
ROPEN	TAPE BASICS EXCEPT 800	51	WOPEN,CLOSE
ROPEN#n	ALL DISK and 800 TAPE	51	WOPEN#n,CLOSE
ROPEN/T	K/A/B DISK BASICS ONLY	51	WOPEN/T,CLOSE/T
RSMODE	SB-6511 ONLY	62	INIT (7/800)
RUN	ALL	51	STOP,END
SAVE	ALL	51	LOAD
SEARCH	700 DISK & ALL 800	51	SEARCH/P
SEARCH/P	700 DISK & ALL 800	51	SEARCH
SET	ALL	52	RESET
SGN	ALL	52	ABS
SIN	ALL EXCEPT D.P.	51	COS,TAN,ATN
SIZE	ALL	51	LIMIT
SKIP	700 (P/P ONLY)	61	PSKIP (800PP)
SOUND	800 ONLY	51	MUSIC, NOISE
SPACE$	80A/80B/800 & 700 DISK	52	SPC
SPC	80K/700/800 ONLY	52	SPACE$
SQR	ALL	52	
STEP	ALL	52	FOR,NEXT
STICK	800 ONLY (JOYSTICK)	53	STRIG
STOP	ALL	53	END,CONT
STR$	ALL	53	VAL
STRIG	800 ONLY (JOYSTICK)	53	STICK
STRING$	80A/80B ONLY	53	SPACE$,SPC
SWAP	ALL 5.25" DISK BASICS	53	CHAIN
SYMBOL	MZ-800 ONLY (HI-RES)	57	
TAB	ALL	53	SPACE$,SPC
TAN	ALL EXCEPT D.P.	53	ATN,COS,SIN
TEMPO	ALL	53	MUSIC
TEST	700 (P/P ONLY)	61	PTEST (800PP)
THEN	ALL	53	IF, ELSE
TI$	ALL	54	
TROFF	700/800 ONLY	54	TRON,TRON/P
TRON	700/800 ONLY	54	TRON/P,TROFF
TRON/P	700/800 ONLY	54	TRON,TROFF
UNLOCK	ALL 5.25" DISK BASICS	54	LOCK
USR	ALL	54	SEE PAGE 54
VAL	ALL	54	STR$
VERIFY	NOT IN SP-6015 & 5Z-009!	55	
WAIT	800 ONLY	55	
WOPEN	TAPE BASICS EXCEPT 800	55	ROPEN,CLOSE
WOPEN#n	ALL DISK and 800 TAPE	55	ROPEN#n,CLOSE#
WOPEN/T	K/A/B DISK BASICS ONLY	55	ROPEN/T,CLOSE/T
XOPEN#n	ALL 5.25" DISK BASICS	55	CLOSE#n,CLOSE
XOR	700 DISK & ALL 800	55	AND, OR, NOT

ALPHABETICAL LIST OF PRINCIPAL KEYWORDS

ABS returns the value of a numeric expression WITHOUT sign
e.g. PRINT ABS(-35) displays the number 35

ALL an undocumented command in MZ-800 Basics; use not known

AND Boolean logical operator, relational and bitwise:-

(Relational) IF (X>3) AND (Y>5) THEN END
(Bitwise) PRINT 255 AND 6 returns the value 6

Relational AND may be simulated by '*' in ALL Basics
Bitwise AND requires SPACES BEFORE AND AFTER IT, and
is ONLY available in MZ-800 Basics.

APPEND or APPEND/T (D-P +SB-6511) appends a named Basic program file to the Basic program file already in memory

e.g. APPEND FDn @V "Diskfile" appends a disk file
or APPEND/T "Tapefile" appends a tape file

N.B. the 'appended' program must have higher line numbers than the program already in memory

ASC returns the ASCII code of the FIRST letter of a string
e.g. PRINT ASC("BCD") displays 66 (decimal)

ATN returns the arctangent (in radians) of a numeric expression
e.g. PRINT ATN(1) displays 0.78539816 (radians)

AUTO sets up auto line-numbering for program input; AUTO with no parameters starts at line 10 and increments in steps of 10, but this may be changed by adding suitable parameters
e.g. AUTO 75,5 begins at line 75 & increments in steps of 5

BEEP sends a short signal to the built-in speaker

BOOT puts the machine into the same state as when it is first switched on (in Basics for MZ-80B, MZ-700 and MZ-800 ONLY)

BYE jumps to a Monitor, leaving memory intact:-

On the MZ-80K, to the ROM Monitor SP-1002
On the MZ-700/800, to the current RAM Monitor
(Equivalent command on the MZ-80A/B is MON)

CHAIN passes execution from the current program in RAM to another program loaded off disk. VARIABLES and ARRAYS are preserved. This is a PROGRAM EQUIVALENT of GOTO e.g.:-
CHAIN FDn @V "Filename" (K/A/B disk Basics)
CHAIN "FDn:Filename" (700/800 5.25" disk Basics)
CHAIN "QD:Filename" (700/800 Quickdisk Basics)

CHANGE reverses MZ-80B keyboard case i.e. toggles upper case to lower case, or vice versa

CHARACTER$ returns the character at the given screen location, in the form of a 1-character string e.g. A$ = CHARACTER$(X,Y)
The range of X is 0-39 in 40 columns, 0-79 in 80 columns
The range of Y is 0-24

CHR$ returns a 1-character string with the given ASCII code
e.g. PRINT CHR$(82) prints 'R' (character 82 decimal)

In all Sharp Basics BUT PLEASE NOTE:-

In SP-5025, CHR$(0-31),(34) & (44) return as null strings
In SP-6015, CHR$(0-31) return as null strings

In the MZ-80A Basics, CHR$(0-31) and (96-104), sent to the SCREEN, produce undocumented 'funny' effects/characters

See Appendix B-2 for improvements to CHR$ in SP-5025/6015
See Appendix B-1 for details of MZ-80A 'funny' effects

CLOSE in a TAPE Basic, closes the OPEN data file
in a DISK Basic, closes ALL disk data files

Variation in all tape Basics EXCEPT MZ-800
CLOSE "Name" to close a specific file

Variations in all disk Basics and MZ-800 tape Basic:-
CLOSE£n to close a specific file

Variations in K/A/B disk Basics:-
CLOSE/T to close all tape files
CLOSE/T "Name" to close a specific tape file

CLR clears all variables (i.e. numeric variables are set to 0 strings are emptied, and arrays destroyed)

N.B. In some Sharp Basics, CLR also clears the interpreter stack; if used in a GOSUB, this will upset the RETURN

CLS clears the screen and 'Homes' the cursor; the following alternatives are available via the PRINT command:-

On the MZ-80K/A/700, PRINT" C "
On the MZ-80A and 700 ONLY, PRINT CHR$(22)
On the MZ-80B ONLY, PRINT CHR$(6)

COLOR in the MZ-700 Basics is used to set the foreground and background colours at a specific screen location.

In the MZ-800 Basics it is used differently, to set the palette code and the mode of superimposition:-

(MZ-700) COLOR x,y,f,b sets screen location x,y to foreground colour f on background colour b

(MZ-800) COLOR p,m sets palette code p, mode m

CONSOLE is used on the MZ-700/800 to set the 'scrolling area' of the screen; on the MZ-80B, it may also be used to set the screen width, and to switch normal/reverse video:-.

(MZ-80B) CONSOLE S4,14,C80,R sets scrolling area to lines 4-14, 80-columns, reverse video. To revert to a full 40-column 'normal' screen, use CONSOLE S0,25,C40,N

(MZ-700) CONSOLE 4,10,5,30 sets the scrolling area to lines 4-14, columns 5-35

(MZ-800) CONSOLE 4,10 sets scrolling area to lines 4-14

N.B. (MZ-80B) PRINT CHR$(6) clears the scrolling area only
(MZ-700) PRINT " C " clears the scrolling area only
(MZ-800) CLS command ALWAYS clears the WHOLE screen

CONT is designed to resume a program which has been halted by BREAK or STOP. In addition, all MZ-80K Manuals state that CONT will resume after END. Later Manuals do not state this but it DOES happen in all MZ-80A/B Basics, and in MZ-700 Basics 1Z-013 and K&P DBasic. In MZ-700 disk Basic 2Z-009E, and in all QD and MZ-800 Basics, CONT after END gives ERROR

COPY/P is a screen dump facility...

COPY/P1 copies the TEXT screen to printer (MZ-80A/B)
COPY/P2 copies GRAPHIC screen 1 to printer (MZ-80B only)
COPY/P3 copies GRAPHIC screen 2 to printer (MZ-80B only)
COPY/P4 copies GRAPHIC screens 1+2 to printer (MZ-80B only)
N.B. See also HCOPY in MZ-800 Basics

COS returns the cosine of a numeric expression in radians
e.g. PRINT COS(3.1415927) will display -1

CSRH system variable which holds the horizontal position of the cursor (range 0-39 with 40-columns, 0-79 with 80-columns)

CSRV system variable which holds the vertical position of the cursor (range 0-24)

CURSOR positions the cursor according to the value of the X and Y co-ordinates which follow e.g.:-
CURSOR 5,10: PRINT "Hello"
will print the word 'Hello' at column 5, row 10

DATA defines a program line containing READ variables
e.g. 1000 DATA 2,4,6,8
or 1000 DATA No,Yes,"Yes please",Maybe
(Strings only require double quotes if spaces are included)

DEF FN defines a mathematical function
e.g. DEF FNA(X)=(2*X+6)
where A is the function label, 26 are allowed (i.e. A-Z)
and X is the variable (only 1 allowed)
On the K/A/B/700, nesting is allowed up to 5 levels
On the MZ-800, nesting is not allowed at all

DEF KEY defines a FUNCTION KEY e.g. DEF KEY(3)="GOTO 200"

On the MZ-80B a CR symbol (" ") may be added by GRPH and SFTLOCK together; and multi-statements are permitted e.g. DEF KEY (4) LOAD!RUN (which will list as LOAD:RUN)

On the MZ-700/800, a CR may be added as CHR$(13)

DEFAULT defines the default device name
e.g. DEFAULT "CMT:" (or RAM:/LPT:/RS1:/RS2:/QD:etc)

DELETE as a Basic keyword, may have two different purposes:-

DELETE 50-100 will delete all lines in the range 50-100

DELETE FDn aV "EXAMPLE" (K/A/B disk Basics) or
DELETE "FDn:EXAMPLE" (700/800 5.25" disk Basics)
will delete the file "EXAMPLE" from a 5.25" disk

DELETE (lines) is in MZ-700/800, and D.P. Basics
DELETE (files) is in all 5.25" Disk Basics

N.B. Quickdisk 'DELETE' is not a Basic keyword
It is a machine-code program which copies a Quickdisk into memory, erases files in RAM as required, and then rewrites the remaining files to the disk

DIM sets up a numeric or string ARRAY; the elements of the array are numbered from 0 upwards, so DIM A(100) sets a 1-D numeric array with 101 elements, numbered 0-100. Similarly, DIM A$(10,5) sets a 2-D string array with 11 x 6 elements

In the 8 s.f. K/B/A Basics, all arrays are limited to 256; in the 16 s.f. D-P disk Basics 1-D arrays are limited only by free memory, but 2-D arrays are still limited to 256

In 700/800 Basics, arrays are limited only by free memory

DIR lists the contents of the disk on the default drive
i.e. the drive which LAST had its directory listed
The default drive at boot-up is Drive 1, which may be changed by specifying a different drive e.g:-
(In MZ-80K disk Basics) DIR2
(In all other disk Basics) DIR FD2
Should the list fill more than one screen, DIR waits until you press CR before displaying the remainder

DIR/P EXACTLY the same as DIR but outputs the information to PRINTER. In SP-6115 ONLY, DIR/P also lists file sizes (in sectors)

EDIT displays the specified line for editing
e.g. EDIT 100 displays line 100
or EDIT alone displays default line 10

ELSE IF..THEN..ELSE is found in 700 disk and all 800 Basics
e.g. IF X=3 THEN 200:ELSE 300

END denotes the LOGICAL end of a program.
Terminates the program, closes all OPEN files,
and returns to command mode with 'Ready'

All MZ-80K Manuals state that CONT will resume after END.
Later Manuals do not state this, but it DOES happen in
all A/B Basics and in S-Basic/K&P DBasic on the MZ-700.

In disk Basic 2Z-009E on the MZ-700, and in all Quickdisk
and MZ-800 Basics, CONT after END gives ERROR

EOF used to divert program if end of file is detected
e.g. IF EOF(£2) THEN 2000

ERL diverts program if ANY error occurs in given line
e.g. IF ERL 300 THEN 2000

ERN used to divert program if given error type occurs
e.g. IF ERN 1 THEN 2000

EXP returns the value of 'e' to the POWER X
e.g. PRINT EXP(1) prints 2.7182818

FAST winds the tape cassette FAST FORWARDS
(MZ-80B ONLY, same as pressing the 'FF' key)

FOR starts a FOR...NEXT loop (with optional STEP n)

e.g. FOR J=33 TO 255:PRINT CHR$(J);:NEXT J
will print the character set of your computer

STEP (if used) may be +ve or -ve
e.g. FOR L=0 TO 40 STEP 2
or FOR M=40 TO 0 STEP -2

FRAC returns the fractional part of a numeric expression
e.g. PRINT FRAC(5.25) displays 0.25

GET scans the KBD for a keypress, does not wait, but
e.g. 200 GET A$:IF A$="" THEN 200
will loop around line 200 UNTIL a key is pressed; the
variable may be numeric, in which case the syntax is
e.g. 200 GET A:IF A=0 then 200

GOSUB calls a subroutine by its line number; the sub-routine
MUST end with a RETURN, which will cause the program to
resume at the line FOLLOWING the original GOSUB

e.g. 200 GOSUB 800
210 REM This line will be executed after RETURN

GOTO transfers program control to specified line

e.g. 200 GOTO 800

GO TO (with space) is allowed, EXCEPT in 2Z-009 & 800 BASICS!!

GRAPH followed by appropriate parameters, controls the graphics screen(s) on the MZ-80B (GRAPH 1 is fitted as standard, GRAPH 2 requires an optional extra RAM card). To write to a screen it must be OPEN FOR INPUT; to view a screen it must be OPEN FOR OUTPUT; the commands are as follows:-

GRAPH I1 (letter I then 1) sets GRAPH 1 to INPUT mode
GRAPH I2 sets GRAPH 2 to INPUT mode
GRAPH O1 (letter O then 1) sets GRAPH 1 to OUTPUT mode
GRAPH O2 sets GRAPH 2 to OUTPUT mode
GRAPH O12 sets both GRAPH 1 & 2 to output mode
GRAPH O0 Resets output mode i.e. NEITHER screen displayed
GRAPH C clears the graphic area currently in INPUT mode
GRAPH F FILLS the graphic area currently in INPUT mode

Multiple statements are allowed e.g. GRAPH O0,I2,C,I1,C,O1

HCOPY copies the screen to the printer
(MZ-800 Basics ONLY and does NOT work to PLOTTER/PRINTER!)

HEX$ converts a decimal expression to hexadecimal
e.g. PRINT HEX$(1234) displays 4D2

IF starts a program branch; usually followed by THEN
e.g. IF X=4 THEN GOTO/GOSUB/PRINT/STOP etc.

In two cases THEN may be omitted i.e.:-
IF..GOTO (linenumber) and IF..GOSUB (linenumber)

IF..THEN..ELSE is not allowed in K/A/B Basics or 700 tape Basic; but it is implemented in other 700/800 Basics

INIT has multiple strange uses on the MZ-700/800:-

(1) INIT "RS1:a,b,c" sets serial port (in 2Z-009 only)
(2) INIT "QD:" formats a Quickdisk!! (in 5Z-008 only)
(3) INIT "RAM:/LPT:/RS1:/RS2:/CRT: (in all 800 Basics)

The syntax in cases (1) & (3) is very complicated, and is given in full in the appropriate Owner's Manual

INP# reads a Z80 PORT and returns the setting in a variable
e.g. INP#X,A reads port X and stores the setting in A
In Basics SP-5025, 1Z-013, K&P 700DBasic

INP@ reads a Z80 PORT and returns the setting in a variable
e.g. INP@X,A reads port X and stores the setting in A
In all Basics except SP-5025, 1Z-013, K&P 700DBasic

INPUT prompts for keyboard input with '?' and flashing cursor; the type of input expected (numeric or string) is governed by the following variable i.e. INPUT A or INPUT A$

'?' may be changed to a more explicit user prompt by
e.g. INPUT "HOW MANY ? ";A or INPUT "Name please..";A$

N.B. TEXT input to a NUMERIC variable will generate ERROR

INPUT#n reads SEQUENTIAL data from a file previously opened for reading by ROPEN#n, into the specified buffer
e.g. INPUT#1, A$,B$,A,B

INPUT#n( ) reads RANDOM data from a file previously opened for reading by XOPEN#n, into the specified buffer

e.g. INPUT#1(nn),A$,B$,A,B (where nn is the record number)

INPUT/T reads data from a file previously opened for reading by ROPEN or ROPEN/T, into the tape data buffer
e.g. INPUT/T A$,B$,A,B

INT returns the WHOLE NUMBER part of a numeric expression
e.g. PRINT INT(5.25) displays the value 5

JOY(n) returns an attribute of the MZ-700 Joystick
(N.B. centre of screen = 127,127)

JOY(0) returns the X-position of Joystick 1 (0-255)
JOY(1) returns the Y-position of Joystick 1 (0-255)
JOY(2) returns the X-position of Joystick 2 (0-255)
JOY(3) returns the Y-position of Joystick 2 (0-255)

JOY(4) reads switch 1 on Joystick 1 (0=OFF, -1=ON)
JOY(5) reads switch 2 on Joystick 1 (0=OFF, -1=ON)
JOY(6) reads switch 1 on Joystick 2 (0=OFF, -1=ON)
JOY(7) reads switch 2 on Joystick 2 (0=OFF, -1=ON)

KEY LIST lists the MZ-700/800 function keys as they are currently defined (cf. KLIST on MZ-80B)

KILL# kills the specified file buffer(s)
e.g. KILL#1 kills file buffer #1
or KILL (alone) kills ALL file buffers and closes all open files

KLIST lists the MZ-80B function keys as they are currently defined (cf. KEY LIST on MZ-700/800)

LABEL defines a label, for use by GOTO and GOSUB

```
e.g. 20 GOSUB "PRINTNAME""
     ...
     ...
     1000 LABEL "PRINTNAME"
     1010 PRINT"John Smith"
     1020 RETURN
```

LEFT$ returns the leftmost characters of a string
e.g. A$=LEFT$(B$,2)
returns A$ as the leftmost TWO characters of B$

LEN returns the number of characters in a string
e.g. A=LEN(X$) or A$=LEN("This string")

LET assigns a value (numeric or string) to a variable
e.g. LET A=29 or LET A$="Message"

However, LET is optional, and it may be omitted e.g.:-
A=29 or A$="Message" will do just as well

LIMIT limits the memory area available to BASIC
e.g. LIMIT 49152 makes 49152 the upper limit of Basic

In all Basics except those for the MZ-80K
the address may be in HEX e.g. LIMIT $BFFF

The cold-start LIMIT can be restored by LIMIT MAX

LIST on its own, lists the complete program in memory
LIST 100- lists the program from line 100 onwards.
LIST 100-900 lists the program from line 100 to line 900

On all machines, listing is aborted by SHIFT/BREAK
In SA- and SB-Basics, and in 1Z-013 and K&P 700DBASIC,
listing may be halted by holding down the SPACE BAR
In SA-Basics only, listing is slowed down by pressing '?'
In 2Z-009E9 and 5Z-008, and in all MZ-800 Basics, listing
may be stopped/restarted by tapping the SPACE BAR

LIST/P acts as LIST, but sends the listing to the printer

LN returns the natural logarithm of a numeric
expression (i.e. the log to base 'e')
e.g. PRINT LN(10) will display 2.3025851

LOAD is used to load a file from tape or disk. Filenames may be up to 16 characters long and may contain spaces, but the exact syntax depends on the Basic being used:-

IN TAPE BASICS ON THE MZ-80K/A/B/700
LOAD "Filename" loads that BASIC file from TAPE
LOAD on its own loads the next BASIC file on the TAPE

IN TAPE BASIC 1Z-016 ON THE MZ-800
the DEVICE and the FILENAME are required e.g.
LOAD "CMT:Filename" loads the named BASIC file from TAPE
Possible device names are CMT:and RAM:, which may
be omitted if the device has been set by DEFAULT
LOAD is programmable (i.e. to load a machine-code file)

IN DISK BASICS ON THE MZ-80K/A/B and MZ-700 K&P DBASIC
LOAD FDn aV "FILENAME" loads that BASIC file from disk
LOAD/T "Filename" loads that BASIC file from TAPE
LOAD/T on its own loads the next BASIC file on a TAPE

IN ALL QUICKDISK BASICS and MZ-700 2Z-009E disk Basic
the DEVICE and the FILENAME are required e.g.
LOAD"FD1:Filename" loads 'Filename' from floppy drive 1
Possible device names are CMT:/FD1:/FD2:/QD:, which
may be omitted if the device has been set by DEFAULT
As with MZ-800 tape Basic 1Z-016, LOAD is programmable

LOCK flags a 5.25" disk file so that it cannot be erased or renamed. There are two slightly different syntaxes e.g.
In MZ-80K/A/B disk Basics: LOCK FD1 "Filename"
In MZ-700/800 5.25" disk Basics: LOCK "FD1:Filename"
In both cases the drive designation may be omitted if the file is on the default drive

LOG returns the logarithm of a numeric expression, to base 10
e.g. PRINT LOG(1000) will display 3

MERGE is used to read in a Basic program from tape or disk, and append it to the program already in RAM. If line numbers clash, the new lines overwrite the old:-

IN TAPE BASICS ON THE MZ-700
MERGE "Filename" appends the named Basic TAPE file
MERGE on its own appends the first BASIC file on a TAPE

IN TAPE BASIC 1Z-016 ON THE MZ-800
MERGE "CMT:Filename" appends the named Basic TAPE file

In MZ-700 and MZ-800 disk Basics
MERGE "FDn:Filename" (in 5.25" disk Basics)
MERGE "QD:Filename" (in Quickdisk Basics)

MID$ is a powerful string-manipulating function
e.g. A$=MID$(B$,5,2) returns A$ as a 2-character string, starting with the 5th character in B$

MOD returns the remainder from a division sum
e.g. PRINT 99 MOD 6 returns 3
MOD requires SPACES BEFORE AND AFTER IT, and is ONLY available in MZ-800 Basics

MON jumps to a Monitor, leaving memory intact.
On the MZ-80A, to ROM Monitor SA-1510
On the MZ-80B, to RAM Monitor SB-1510
(Equivalent command on MZ-80K/700/800 is BYE)

MUSIC plays a series of notes as specified by one or more string variable(s); each note is represented by a letter (A-G), and may be sharpened by a preceding "#" (flats are obtained by sharpening the note below). On the MZ80K/A/B/700, THREE octaves are available, starting with 'C' below middle 'C'; a preceding '-' denotes the bottom octave, and a preceding '+' denotes the top octave. The DURATION of a note is fixed by a suffix (0-9); 9 = semibreve, 7 = minim, 5 = crochet, 3 = quaver, 1 = semi-quaver, and 0 = demi-semi-quaver. A rest is denoted by R, suffixed by (0-9) to fix its length. Overall speed is set by the TEMPO command (1-7).

On the MZ-800, SEVEN 'basic' octaves are available, set by including the commands O0 through O6 in music strings; and two or three music strings may be played simultaneously by separating them with semi-colons.

Examples of MUSIC strings are given in Appendix B-4.

NEW erases the Basic program currently in memory, and clears all variables and arrays. NEW does NOT alter the current LIMIT (the MZ-80A Owner's Manuals are wrong on this point)

NEW ON (MZ-800 only) erases the program in memory TOGETHER WITH the plotter/printer routines (the space released by erasing the p/p routines is re-allocated for Basic program use)

NEXT increments the FOR/NEXT counter by STEP (+1 by default)

NOISE generates 'white noise'; the pitch and duration are specified by music strings, as with 'MUSIC'

NOT Boolean logical operator, bitwise negation
e.g. PRINT NOT 0 returns -1
NOT requires SPACES BEFORE AND AFTER IT

ON ERROR GOTO on error jumps to specified program line
e.g. ON ERROR GOTO 2000

ON...GOSUB/GOTO branches to one of a list of line numbers, depending on the integral value of the variable

e.g. 10 ON A GOTO/GOSUB 100,200,300,400
20 REM this is the next line of the program

If A=1 then the program jumps to 100; if A=2, it jumps to 200; and so on. If A>(items in list), execution skips to the next line; in the above example, this happens if A>4.

OR Boolean logical operator, relational and bitwise:-
(Relational) IF (X>3) OR (Y>5) THEN END
(Bitwise) PRINT 255 OR 6 returns the value 255

Relational OR may be simulated by '+' in ALL Basics
Bitwise OR requires SPACES BEFORE AND AFTER IT, and and is ONLY available in MZ-800 Basics

OUT# writes a numeric value (0-255) to the specified port
e.g. OUT#232,A writes the value of A to port 232
(In Basics SP-5025, 1Z-013, K&P 700DBasic ONLY)

OUT@ writes a numeric value (0-255) to the specified port
e.g. OUT@232,A writes the value of A to port 232
(In all Basics EXCEPT SP-5025, 1Z-013, K&P 700DBASIC)

PAGE sets the length of the printed page, in lines. The default value is 66, and may be changed to (say) 70 by PAGE 70
Subsequent 'form feeds' will be to the new page length
N.B. MZ-700/800 Basics, for PLOTTER/PRINTERS ONLY

PAGE/P sets the length of the printed page, in lines. The default value is 66, and may be changed to (say) 70 by PAGE/P 70
Subsequent 'form feeds' will be to the new page length
N.B. MZ-80A/B Basics, SHARP printers (or equivalent) ONLY

PAI returns a multiple of pai e.g. PAI(2) returns 6.2831853

PAL matches MZ-800 palette and color codes e.g. PAL p,c (For a full explanation see Appendix A of the 800 Owner's Manual)

PEEK returns the contents of the specified address as a decimal number 0-255 e.g. PRINT PEEK (53248) returns the 'Display' code of the character at $D000 in VRAM

N.B. the address may be a variable e.g. PRINT PEEK(AD); or, in all Basics except those for the MZ-80K, the address may be expressed in HEX e.g. PRINT PEEK ($D000)

PEEK@ (on the MZ-700 ONLY) peeks the SYSTEM RAM above $CFFF e.g. PRINT PEEK@($D000) returns the code at $D000 in S-RAM

POINT on the MZ-80B, acts as follows:-
returns 0 if the location is RESET in BOTH graphic areas.
returns 1 if the location is SET in graphic area 1 ONLY
returns 2 if the location is SET in graphic area 2 ONLY
returns 3 if the location is set in BOTH graphic areas

On the MZ-800, POINT(X,Y) returns the PALETTE CODE in force at the given location

POKE fills the specified RAM location (0-65535) with the value of a numeric expression (0-255) e.g. POKE 53248,16 puts a 'P' (display code 16) at the first location in VRAM

N.B. parameters may be variables, and in all Basics except MZ-80K Basics, one or both may be in HEX e.g. POKE $D000,16 or POKE 53248,$10 or POKE $D000,$10

POKE@ (on the MZ-700 ONLY) writes to the SYSTEM RAM above $CFFF e.g. POKE@ $D000,16 writes 16 to $D000 in S-RAM

PRINT displays numerical values or text on screen
e.g. PRINT A;A$,"HELLO" (note semicolon and comma)
The semicolon causes A$ to follow straight after A
The comma causes 'HELLO' to appear at the next TAB position
On the MZ-700, PRINT may include colour parameters f,b
On the MZ-800, PRINT may include a palette code p

PRINT USING sets up a 'mask', which formats numeric variables to the specified number of decimal places and aligns the decimal point to the specified position. The syntax is:-

```
PRINT USING "####.##";A
```
which displays A with the d.p. in the 5th column, to 2 d.p.

The 'mask' may be pre-defined, may include commas, and may be preceded or followed by appropriate messages
e.g. PRINT USING"UNIT PRICE $#,###.## PLUS VAT";A
or M$="UNIT PRICE $#,###.## PLUS VAT":PRINT USING M$;A

PRINT USING holds for 1 line and must be declared as needed
On the MZ-700/800 colour parameters may be included; for full explanations of these, see appropriate Owner's Manual

PRINT#n writes sequential data to a file previously opened for writing by WOPEN#n, into the specified buffer
e.g. PRINT#1, A$,B$,A,B

PRINT#n( ) writes RANDOM data to a disk file previously opened by XOPEN#n, into the specified buffer
e.g. PRINT#1(nn),A$,B$,A,B (where nn = record number)

PRINT/P is the same as PRINT, but sends output to the printer

PRINT/P USING is the same as PRINT USING but outputs to printer

PRINT/T writes sequential data to a TAPE file previously opened for writing by WOPEN or WOPEN/T
e.g. PRINT/T A$,B$,A,B

RAD converts a numeric expression from degrees to radians
e.g. PRINT RAD(180) displays the value 3.1415927

READ...DATA reads successive items of DATA into variables from a list held in separate DATA statements

```
e.g. 10 READ A$,B$,C$,D$
     20 DATA JOHN,THOMAS,JANE,MAURICE
     30 PRINT A$:PRINTB$:PRINTC$:PRINTD$
```

will read in the four Christian Names, and print them out on separate lines.

READ must have a corresponding DATA statement somewhere in the program. If an item of string data includes spaces, it MUST be in double quotes; otherwise quotes are not needed

The variable in READ must match the DATA being read i.e. string DATA will cause an error if READ expects a number

REM a non-executing command which causes Basic to ignore the rest of the current statement; normally used for messages

```
e.g. 399 REM The subroutine in 400 waits for a keypress
     400 GET A$:IF A$="" THEN 400: RETURN
```

RENAME is used in 5.25" disk Basics, to rename a file

```
e.g. RENAME FDn aV "Oldname","Newname" (K/A/B disk Basics)
or   RENAME "FDn:Oldname","Newname" (700/800 disk Basics)
```

N.B. you can't rename a LOCKED file!

RENUM renumbers some or all program lines; the syntax is:-

RENUM 1st new number, 1st line to be renumbered, increment

Default values are 10, existing 1st line, 10 e.g:-

```
RENUM (renumbers all lines, as 10,20,30...)
RENUM 100 (renumbers all lines, as 100,110,120...)
RENUM 200,30 (renumbers old line 30 onwards, as 200,210...)
RENUM 1,,1 (renumbers all lines, as 1, 2, 3...)
```

RESET turns off a dot or pixel at the specified position
e.g. RESET 20,12 turns on the dot/pixel at X=20, Y=12

On the MZ-80K/A/700, 'position' refers to 'pixels' on an 80 x 50 screen and the limits are X=(0-79) and Y=(0-49)

On the MZ-80B, 'position' refers to the 320 x 200 graphic area currently in INPUT mode, limits X=(0-319), Y=(0-199)

On the MZ-800 the screen limits are X=(0-639), Y=(0-199) but larger 'virtual' values of X and Y are permitted, from -16384 to +16383 in both cases

On the MZ-800 RESET may be used to CHANGE the colour of an existing dot, by inserting a new colour specification (in SQUARE brackets) in front of the X,Y values e.g.:-
RESET [3,0] 100,50 changes to Pallete Code 3, Mode 0
The default colour is that in the last COLOR statement

RESTORE allows DATA statements which have been read before, to be read again; in ALL Basics, RESTORE on its own will reset the READ pointer to the first DATA line in the program

In all Basics EXCEPT those for the MZ-80K, RESTORE may be followed by a line number (e.g. RESTORE 200); this will restore the READ pointer to the specified DATA line, rather than the first DATA line

MZ-80K Basics do not reject RESTORE nn as a SYNTAX ERROR; but the line number will be ignored, and the READ pointer will be reset to the first DATA line

RESUME causes execution to transfer to the specified program line
e.g. RESUME 2000 resumes execution at line 2000; normally used in combination with error trapping functions:-

e.g. 10 IF (ERN=24)*(ERL=200) then RESUME 300

RETURN marks the end of a SUB-ROUTINE, and causes program flow to jump to the line following the associated GOSUB

REW on the MZ-80B, causes the TAPE cassette to rewind
Equivalent to pressing the REW key

RIGHT$ returns the rightmost characters of a string
e.g. A$=RIGHT$(B$,2)
returns A$ as the rightmost TWO characters of B$

RND returns a pseudo-random number (i.e. one of a set) in the range 0.00000001 - 0.99999999:-

RND(0) or RND(-n) returns the first number of a new set

RND(+n) returns the next number of the existing set

INT(RND(X)*(M-N+1)+N) returns an integral random number between N and M

ROPEN opens the named TAPE file for READING sequential data
e.g. ROPEN "Filename" (K/A/B/700 Tape Basics ONLY)
(only ONE tape file may be OPEN at any one time)

ROPEN/T opens the named TAPE file for READING sequential data
e.g. ROPEN/T "Filename" (K/A/B disk Basics ONLY)
(only ONE tape file may be OPEN at any one time)

ROPEN#n opens logical file number 'n' for READING sequential data from a DISK file, or from a TAPE file in 2Z-009 and all MZ-800 Basics. Up to 10 files may be OPEN at any one time, and 'n' may have any value from 1 to 127:-

e.g. ROPEN#3, FDn aV, "NAME" (K/A/B disk Basics)
or ROPEN#25,"FDn:NAME" (700/800 5.25" disk Basics)
or ROPEN#70,"QD:NAME" (Quickdisk Basics)
or ROPEN#99,"CMT:NAME" (700 disk Basics/all 800 Basics)

RUN on its own, clears all variables and arrays and starts execution of the program in memory at the lowest linenumber

RUN followed by a linenumber starts program execution at the linenumber specified, leaving variables and arrays with their original values intact - very useful for testing

SAVE is used to save a file on tape or disk. Filenames may be up to 16 characters long and may contain spaces:-

IN TAPE BASICS ON THE MZ-80K/A/B/700
SAVE "Filename" saves that BASIC file on TAPE

IN TAPE BASIC 1Z-016 ON THE MZ-800
the DEVICE and the FILENAME are required e.g.
SAVE "CMT:Filename" saves a BASIC file on TAPE
Device names are CMT:/RAM:/DEFAULT device

IN DISK BASICS ON THE MZ-80K/A/B
SAVE FDn "Filename" saves that BASIC file on disk
(If FDn is omitted the default drive will be used)
SAVE/T "Filename" saves that BASIC file to TAPE

IN ALL MZ-700/800 DISK BASICS
the DEVICE and the FILENAME are required e.g.
SAVE"FD1:Filename" to save 'Filename' on floppy disk 1
Device names are CMT:/FD1:/FD2:/QD:/DEFAULT device

SEARCH searches the Basic program in memory, for lines containing the specified character string e.g.:-
SEARCH "ABC" display the line(s) containing ABC

SEARCH/P is the same as SEARCH, except that lines containing the search string are sent to the printer

SET turns on a dot or pixel at the specified position
e.g. SET 20,12 turns on the dot/pixel at X=20, Y=12

On the MZ-80K/A/700, 'position' refers to 'pixels' on an 80 x 50 screen and the limits are X=(0-79) and Y=(0-49)

On the MZ-80B, 'position' refers to the 320 x 200 graphic area currently in INPUT mode, limits X=(0-319), Y=(0-199)

On the MZ-800 the screen limits are X=(0-639), Y=(0-199) but larger 'virtual' values of X and Y are permitted, from -16384 to +16383 in both cases

On the MZ-700 a 3rd (optional) parameter specifies colour:-
SET X,Y,C sets the point X,Y to colour C (C=0-7, giving black(0),blue,red,purple,green,light blue,yellow,white(7)

On the MZ-800 a colour specification (in SQUARE brackets) may PRECEDE the X,Y values e.g.:-
SET [3,0] 100,50 sets Pallete Code 3, Mode 0
The default colour is that of the last COLOR statement

SGN returns the arithmetic SIGN of a numeric expression i.e.:-
if X > 0, SGN(X) returns +1
if X = 0, SGN(X) returns 0
if X < 0, SGN(X) returns -1

SIN returns the sine of a numeric expression in radians
e.g. PRINT SIN(1.5707963) displays 1

SIZE returns the number of free memory locations
PRINT SIZE will display the message 'nnnnn BYTES FREE'

SOUND generates a sound of specified pitch and duration
e.g. SOUND 24,100 generates middle 'C' for 1 second
The pitch range is 9 (octave 0 'A') to 83 (octave 6 'B')
The duration is specified in 1/100's of a second

SPACE$ assigns the given number of spaces to the named variable
e.g. B$ = SPACE$(12):PRINT B$...prints 12 spaces
or B$ = RIGHT$(SPACE$(7)+STR$(X),7)..right justifies X

SPC may be used to print the specified number of spaces from the current cursor position
e.g. PRINT SPC(12) prints 12 spaces

N.B. Some Basics have SPACE$, some have SPC, and some have BOTH!! From the above, note that SPACE$ is more powerful

SQR returns the POSITIVE SQUARE ROOT of a numeric expression
e.g. PRINT SQR(4) returns 2

STEP (+ve or -ve) is optional with FOR..NEXT

e.g. FOR L=0 TO 40 STEP 2 (positive step)
or FOR M=40 TO 0 STEP -4 (negative step)

STICK on the MZ-800, returns an integer (1-8) to indicate the state of the joystick lever or cursor control keys:-

STICK(0) returns the state of the cursor keys
STICK(1) returns the state of joystick 1
STICK(2) returns the state of joystick 2

Up, Right, Down and Left give 1,3,5,7 respectively
Intermediate positions give 2,4,6,8 and can be simulated on the cursor keys by pressing two keys at once e.g. pressing cursor up+right together gives 2

STOP interrupts a program... typing CONT (CR) will re-start it

STR$ converts the value of a NUMERIC expression to the corresponding STRING VARIABLE
e.g. STR$(3.142) returns the string "3.142"

STRIG on the MZ-800, returns an integer (0 = not pressed, 1 = pressed) to indicate the state of the joystick button or the space bar:-

STRIG(0) returns the state of the space bar
STRIG(1) returns the state of joystick 1 button
STRIG(2) returns the state of joystick 2 button

STRING$ constructs a string of consecutive characters
e.g. B$ = STRING$("A",6) makes B$ = "AAAAAA"

SWAP loads a program off disk and switches to it. Variables and arrays are preserved. When the swapped program ends execution of the original program resumes at the line after the SWAP. This is a PROGRAM EQUIVALENT of GOSUB
e.g. SWAP FDn aV "Filename" (K/A/B disk Basics)
or SWAP "FD1:Filename" (700/800 disk Basics)
N.B. NOT AVAILABLE in Quickdisk Basics!!

TAB moves the cursor to the specified X position

e.g. PRINT TAB (25);"Hello" moves the cursor along the current line to column 25 and prints the word 'Hello'

N.B. columns are numbered 0-39 (or 0-79 in 80-columns) Also note the semi-colon, which causes printing to occur at the cursor, rather than on the next line. If the cursor is already past the position specified, TAB is ignored

TAN returns the tangent of a numeric expression in radians
e.g. PRINT TAN(0.78539816) displays the value 1

TEMPO sets the speed at which MUSIC statements are played
e.g. TEMPO 4 sets medium speed (1 = slowest, 7 = fastest)

THEN must follow an IF statement
e.g. IF X<3 THEN STOP

TI$ stores the current machine 'TIME' as a six digit number
TI$ starts at 000000 when the computer is switched on
but may be set thereafter by TI$="HHMMSS"(CR)
PRINT TI$ displays the machine 'TIME' in 6-digit format

TROFF turns off the program TRACE facility which is
implemented by TRON or TRON/P

TRON turns on a program TRACE facility
After TRON, line numbers are printed on the screen
in square brackets, as the program executes

TRON/P is the same as TRON, but outputs to printer

UNLOCK unlocks a locked 5.25" disk file
e.g. UNLOCK FDn aV "Filename" (K/A/B disk Basics)
or UNLOCK "FD1:Filename" (700/800 disk Basics)

USR transfers control to a machine code sub-routine
e.g. USR(49152) transfers control to RAM location 49152
The machine-code routine at the specified address will
execute until a RETURN is encountered, when control
returns to the Basic program. In all Sharp Basics EXCEPT
SP-5025 the address may be specified in HEX e.g. USR($C000)

In all Basics EXCEPT SP-5025 for the MZ-80K, USR may
also be used to pass a predefined string variable:-

(In SP-6015) WOPEN#10, USR(nnnn):PRINT#10,A$:CLOSE#10

(In all others except 5025) USR($nnnn,$AA)

Prior to the execution of the machine-code routine
the address of the string variable is placed in DE
and its length is placed in BC

On the MZ-800, it is also possible to specify another
string variable to be loaded into DE and BC when the
machine-code routine returns to Basic:-

USR($nnnn,$AA,$ZZ)

On the MZ-800, USR will also pass the address of any
declared error-processing routine, into the IX register

VAL converts numeric data held as a string, to a proper
numeric variable
e.g. PRINT VAL("123.6")/2 will display the number 61.8

VERIFY compares the specified file ON TAPE with the program in memory, and reports on its validity

e.g. VERIFY"TESTFILE" (Most K/A/B Basics, 700 tape Basic)
or VERIFY"CMT:TESTFILE" (700 disk & most 800 Basics)

On all machines, if an error is detected, an error message of some kind will be displayed. On all except the MZ-800, successful verification prints the messages 'O.K.' and then 'Ready'. On the MZ-800, successful verification jumps straight to 'Ready'.

N.B. VERIFY is not available in SP-6015 and 5Z-009 !!

WAIT suspends program execution for the specified number of milliseconds
e.g. WAIT 100 suspends the program for 0.1 seconds

WOPEN opens a TAPE file for storing sequential data
e.g. WOPEN "Filename" (K/A/B/700 Tape Basics ONLY)
(Only ONE tape file may be OPEN at any one time)

WOPEN/T opens a TAPE file for storing sequential data
e.g. WOPEN/T "Filename" (K/A/B disk Basics ONLY)
(Only ONE tape file may be OPEN at any one time)

WOPEN#n opens logical file number 'n' for WRITING sequential data to a DISK file, or to a TAPE file in 2Z-009 and all MZ-800 Basics. Up to 10 files may be open at any one time, and 'n' may have any value from 1 to 127:-

e.g. WOPEN#3, FDn aV,"NAME" (K/A/B disk Basics)
or WOPEN#25,"FDn:NAME" (700/800 5.25" disk Basics)
or WOPEN#70,"QD:NAME" (Quickdisk Basics)
or WOPEN#99,"CMT:NAME" (700 disk Basics/all 800 Basics)

XOPEN#n opens a buffer for READING/WRITING to a RANDOM data file on DISK. Up to 10 files may be XOPEN at any one time, and 'n' may have any value from 1 to 127:-

e.g. XOPEN#20, FD1 a9, "Filename" (K/A/B disk Basics)
or XOPEN#30, "FDn:Filename" (700/800 5.25" disk Basics)

N.B. Random files only possible in 5.25" disk Basics

XOR Boolean logical operator, bitwise exclusive OR
e.g. PRINT 255 XOR 6 returns 249

Bitwise XOR requires SPACES BEFORE AND AFTER IT

****

MZ-80B/MZ-800 HI-RES COMMANDS

N.B. In many of the commands below, the first 2 parameters [p,m] give the COLOUR SPECIFICATION (p = palette code, m = superimpose mode); these do NOT apply on the MZ-80B. Note also that X and Y co-ordinates are restricted differently on the two machines:-

On the MZ-80B, screen limits are X=(0-319) and Y=(0-199)

On the MZ-800, screen limits are X=(0-639), Y=(0-199), but larger 'virtual' values are accepted i.e. X (or Y) = (-16384 to +16383)

In both cases (0,0) represents the top L.H. corner of the screen

****

BLINE draws a LINE between (X1,Y1), (X2,Y2), (X3,Y3) etc.

e.g. BLINE [p,m] 5,5,10,5,10,10,5,10

On the MZ-80B the line is 'black' i.e. BLINE may be used to 'UNDRAW' the 'white' lines produced by LINE. On the MZ-800, BLINE is used to CHANGE the colour of a line

BOX draws a box based on the co-ordinates for opposite corners; the co-ordinates may be followed by a PALETTE code

e.g. BOX [2,0]20,20,60,60,2

outlines a box in colour specification [2,0] and fills it in according to palette code 2 (if the palette code is omitted, only the outline is drawn)

CIRCLE draws all or part of a circle or ellipse. The syntax is:-

CIRCLE [p,m] X,Y,R,a,s,f,O

where X,Y are the coordinates of the centre, R is the radius, and 'a' is the aspect ratio if an elipse is required. These parameters may be followed by the starting and finishing angles 's' and 'f' (in radians) to limit the figure to an arc, in which case a final letter 'O', will draw the enclosing radii to form a 'fan'

N.B. CIRCLE on the 700 plotter/printer is similar but different!

IMAGE/P is the printer equivalent of PATTERN, in that it draws 8-bit(vertical) dot patterns on printer. The syntax is:-

IMAGE/P Z$

where Z$ is made up of letters or CHR$ codes as described for the PATTERN command below. Each letter or CHR$ code represents a VERTICAL 8-bit column (highest bit at the top). To 'TAB' the pattern on paper, Z$ must begin with the requisite No. of CHR$(0). In NORMAL PRINT mode the printer can print up to 480 vertical rows of dots, whilst in CONDENSED mode, up to 816 rows are possible

LINE draws a LINE between (X1,Y1), (X2,Y2), (X3,Y3) etc.

e.g. LINE [p,m] 5,5,10,5,10,10,5,10

N.B. LINE on the 700 plotter/printer is similar but different!

PAINT fills in an area on the screen with the colour specified by a PALETTE code. The area is specified by a co-ordinate within it and a surrounding boundary colour e.g.:-

```
10 CIRCLE[2]160,100,50:REM paint the boundary in code 2
20 PAINT[1]160,100,2:REM fill it in with code 1
```

PATTERN draws a dot pattern on the current graphic screen

e.g. PATTERN [p,m] 8,A$

In the above example, the '8' refers to the numbers of layers to be drawn (a negative value draws the pattern in layers from the top downwards, whilst a positive value draws it from the bottom upwards.)

Each character in A$ is an 8-bit binary representation of the POINTS to be lit (e.g. "N" = $4E = 01001110). A$ may be built up using letters (e.g."ABCDEF") or using CHR$ (e.g. CHR$($41)+CHR$($42) +.... etc.) Any number of characters may be used and will (in this example) print in ascending layers of 8, repeating across the page from left to right until all the data in A$ has been printed

POSH is a system variable which holds the HORIZONTAL position pointer in the (current) graphic screen.

POSITION sets the position pointer for graphics statements

e.g. POSITION X,Y (N.B. 0,0 = top L.H. corner of screen)

POSV is a system variable which holds the VERTICAL position pointer in the (current) graphic screen.

SYMBOL draws a graphics pattern, based on the specified character string, but magnified by given factors horizontally and vertically, and rotated through a specified angle

e.g. SYMBOL [p,m] X,Y,"STRING DATA",h,v,a

where X,Y = coordinates of upper left corner
"STRING DATA" = character pattern to be drawn
h = horizontal magnification 1-255 (default 1)
v = vertical magnification 1-255 (default 1)
a = angle of rotation about upper left corner and:-
a = 0 represents 0 degrees rotation (default)
a = 1 represents 90 degrees rotation
a = 2 represents 180 degrees rotation
a = 3 represents 270 degrees rotation

****

MZ-700/MZ-800 PLOTTER/PRINTER COMMANDS

NOTES

(1) The actions of the plotter/printer commands in MZ-700 Basics are identical to the actions of the plotter/printer commands in MZ-800 Basics; but in the MZ-800 Basics many of the names of the commands are changed by being prefixed with a 'P'. One could perhaps cope with this fairly easily, if it were not for the fact that, in addition, the names of some MZ-700 plotter/printer commands appear as SCREEN commands in MZ-800 Basics, but with different syntaxes and actions!! We hope this section will help to clear up the difficulties.

(2) For both plotter/printer systems, the Manuals state that the theoretical range of X is (-480 to +480), and the theoretical range of Y is (-999 to +999). However, it is also made clear that the practical limits of X depend on the position of the pen; thus for example, the allowable limits of X immediately after entering the 'graphics' mode are 0 and +480, whereas if the Y-AXIS is then moved to e.g. the centre of the paper with (P)MOVE 240,-240:HSET, then the allowable limits of X become -240 and +240.

In spite of the above general statements, the Manuals also state that, in the CIRCLE and PCIRCLE commands, X and Y both have the range (-999 to +999); and the range of the radius, R, is (0-999). We have not tested to see whether these limits are practicable.

****

AXIS (700/800) draws an X or Y axis, with scale marks. The syntax is:-

AXIS x,p,r

where x = integer specifying X or Y axis (X=0, Y=1)
p = integer specifying scale pitch (-999 to +999)
r = integer specifying no. of repetitions (1 - 255)

CIRCLE (700 only) draws all or part of a circle or polygon. The syntax is:-

CIRCLE X,Y,R,s,f,d

where X,Y are the coordinates of the centre, and R is the radius (0-999). These parameters may be followed by the starting and finishing angles 's' and 'f' (in degrees) to draw a partial figure, and by the step angle d (degrees), which is the angle subsided by the sides of a polygon. The lowest normal value of d is 0.2, which draws a near-perfect circle; if d = 0 then the enclosing radii are added to a part circle, to form a 'fan'

N.B. PCIRCLE on the 800 plotter/printer is identical!!
CIRCLE on the 800 VDU is similar but different!

GPRINT (700/800) prints the specified character, at the specified size and at an angle of 0, 90, 180, or 270 degrees. The syntax is:-

GPRINT n,a,X$

where n = integer indicating size (0-63, default 1)
a = integer indicating angle (0-3, default 0)
X$ = character or string to be printed

HSET (700/800) sets the current pen location as a new origin. A common use is to move the pen to the required origin for a new figure, and then set it by HSET e.g.:-

```
10 (P)MOVE 240,-240
20 HSET
```

LINE (700 Only) draws a LINE between points (X1,Y1), (X2,Y2), (X3,Y3) .....etc. An (optional) initial parameter %n specifies the type of line

e.g. LINE %2,5,5,10,5,10,10,5,10

draws a dotted-line box (%1 gives a solid line, %2 gives a dotted line, higher values of %n give dashed lines with dashes and breaks of increasing length)

N.B. PLINE on the MZ-800 plotter/printer is identical!!
LINE on the MZ-800 VDU is similar but different!

MODE (700 only) sets the plotter/printer to the specified text or graphics mode. The syntax is:-

MODE TN to set text mode, 40 characters per line
MODE TL to set text mode, 26 characters per line
MODE TS to set text mode, 80 characters per line

MODE GR to set graphics mode

N.B. PMODE on the MZ-800 is identical!!

MOVE (700 only) lifts the pen and moves it to the specified location

e.g. MOVE 240,240

N.B. PMOVE on the MZ-800 is identical!!

PAGE (700/800) sets the number of lines per printed page

e.g. PAGE 20 sets the form length to 20

N.B. To perform a form feed, PRINT/P CHR$($0F)

PCIRCLE (800 only) draws all or part of a circle or polygon. The syntax is:-

PCIRCLE X,Y,R,s,f,d

where X,Y are the coordinates of the centre, and R is the radius (0-999). These parameters may be followed by the starting and finishing angles 's' and 'f' (in degrees) to draw a partial figure, and by the step angle d (degrees), which is the angle subsided by the sides of a polygon. The lowest normal value of d is 0.2, which draws a near-perfect circle; if d = 0 then the enclosing radii are added to a part circle, to form a 'fan'

N.B. CIRCLE on the 700 plotter/printer is identical!!

PCOLOR (700/800) specifies the pen color to be used

e.g. PCOLOR n

The range of n is 0-3 (0=black, 1=blue, 2=green, 3=red)

PHOME (700/800) returns the pen to the origin

PLINE (800 Only) draws a LINE between points (X1,Y1), (X2,Y2), (X3,Y3) .....etc. An (optional) initial parameter %n specifies the type of line

e.g. PLINE %2,5,5,10,5,10,10,5,10

draws a dotted-line (n=1 gives a solid line, n=2 gives a dotted line, higher values give dashed lines with dashes and breaks of increasing length)

N.B. LINE on the MZ-700 plotter/printer is identical!!

PLOT OFF - see below, under PLOT ON

PLOT ON (700/800) makes it possible to use the plotter/printer as a display unit. PLOT ON is only valid if MODE is set to TN on the MZ-700, or any TEXT MODE on the MZ-800

On the MZ-800 only, PLOT ON automatically sets the number of characters per line on the printer to the same as on the screen i.e. 40 or 80

Any character not in the P/P's character set is printed as '.', and the pen color may be changed by CTRL+'G'

PLOT ON is cancelled by PLOT OFF

On the MZ-800 only, PLOT ON is also cancelled, automatically, by any INIT "CRT:...." command

PMODE (800 only) sets the plotter/printer to the specified text or graphics mode. The syntax is:-

PMODE TN to set text mode, 40 characters per line
PMODE TL to set text mode, 26 characters per line
PMODE TS to set text mode, 80 characters per line

PMODE GR to set in graphics mode

N.B. MODE on the MZ-700 is identical!!

PMOVE (800 only) lifts the pen and moves it to the specified location

e.g. PMOVE 240,240

N.B. MOVE on the MZ-700 is identical!!

PSKIP (800 only) feeds the paper forwards or backwards by the specified number of lines, maximum 20

e.g. PSKIP 12 feeds the paper 12 lines forwards
or PSKIP -6 feeds the paper 6 lines backwards

N.B. SKIP on the MZ-700 is identical!!

PTEST (800 only) causes the plotter/printer to print squares in black, blue, green and red, in that order, to test the colours, ink quality etc.

N.B. TEST on the MZ-700 is identical!!

RLINE (700/800) is the same as LINE on the MZ-700 and PLINE on the MZ-800, except that the co-ordinates are read as RELATIVE to the current pen location, instead of being read as absolute

RMOVE (700/800) lifts the pen and moves it RELATIVE to its current position

e.g. RMOVE -120,0 moves the pen 120 units vertically down

SKIP (700 only) feeds the paper forwards or backwards by the specified number of lines, maximum 20

e.g. SKIP 12 feeds the paper 12 lines forwards
or SKIP -6 feeds the paper 6 lines backwards

N.B. PSKIP on the MZ-800 is identical!!

TEST (700 only) causes the plotter/printer to print squares in black, blue, green and red, in that order, to test the colours, ink quality etc.

N.B. PTEST on the MZ-700 is identical!!

******

SPECIAL COMMANDS IN SHARP DISK BASIC SB-6511

EXTRA RS-232 COMMANDS (as in Manual for RS-232C I/F card)

RSMODE sets up the specified RS-232 channel; syntax:-

RSMODE a,Rb,Tc,Md,RXe

where:-

a = channel (A or B)
b = no of bits to receive (5,6,7 or 8)
c = no of bits to send (5,6,7 or 8)
d = parity and stop bits
69,70,71 = odd, none, even parity (1 stop bit)
73,74,75 = odd, none, even parity (1.5 stop bits)
77,78,79 = odd, none, even parity (2 stop bits)
e = receive inactive(0) or active (1)

Thus a valid command would be:- RSMODE A,R8,T8,M79,Rx1

RSO sends data via the specified channel e.g.:-

RSO A,T$

sends data (as previously stored in T$) via channel A

RSI receives data via the specified channel e.g.:-

RSI B,T$

receives data via channel B, and stores it as T$

THE EXTRA I/O COMMANDS FOR THE GP-IB GENERAL PURPOSE I/F CARD ARE ON THE NEXT PAGE

EXTRA I/O COMMANDS (as in Manual for GP-IB I/F card)

N.B. In some of the commands below, n1;..;n14 is used to indicate that up to 14 devices may be specified, with device numbers in the range 0-30; and V1,..,Vn is used to denote a list of data variables, which may be string, numeric or constant.

ICL = interface clear (sends 100 uS pulse to reset I/F card)

REN = enable 'Remote' mode (all devices)

LCL = set all devices to 'Local' mode

LCL n1;..;n14 = set selected devices to 'Local' mode

LLO = 'Local' lock-out (stops all devices from being set LOCAL)

DCL = clears all devices

DCL n1;..;n14 = clear selected devices

TRG n1;..;n14 = trigger selected devices

PCT n = pass control to device n

WRT n1;..;n14,V1,..,Vn = write data to selected devices

RED n,V1,..,Vn = read data from device n

WRT/V1,..,Vn = check that MZ-80B is a talker, then send data

RED/V1,..,Vn = check that MZ-80B is a listener, then receive data

CMDW(A$) = set 'B' as talker, and set specified device as listener

CMDR(A$) = set 'B' as listener, and set specified device as talker

ON SRQ (nn) = jump to linenumber nn after a service request

SPOL n,A = perform serial poll on device n, store status in A

PPC n,L = configure parallel poll to device n on data line L (1-8)

PPOL A = perform parallel poll, store status in A

PPU = unconfigure parallel poll

GPIBM n = sets the 'MZ-80B' as system and active controller, with device number n

EOIW b1[,b2] = set b1[,b2] as delimiter(s) for data transmission

EOIR b1[,b2] = set b1[,b2] as delimiter(s) for data reception

****

OTHER BASICS AND TOOLKITS FOR SHARP COMPUTERS

A) MODIFIED 'STANDARD' BASICS AND THEIR TOOLKITS ON THE MZ-80K

The need for better Basics is greatest on the MZ-80K; the 8 s.f. Basics lack key-repeat and string comparisons; and no Basic has RENUMBER. The first ideas came as 'Toolkits'; the best were 'BASIC EXTENSIONS', 'BASIC PLUS 2', 'SPEED BASIC' and 'KNIGHT COMMANDER'.

'B. EXTENSIONS' adds BREAK, TRACE, STEP, DELETE, RENUMBER, AUTO, SET, RESET, PRINT@, AND, OR, NOT and A$<>B$. 'B. PLUS 2' adds to 'B. EXTENSIONS' and gives key-repeat, a number pad, and LINK (i.e. APPEND). 'SPEED B.' adds AUTO, RENUMBER, DELETE, APPEND, TRACE, DUMP, PRINT@, AND, OR, NOT, and string comparisons on a length priority; it also has PRINT/S/A/N/O/B (see below), and fast DATA filing. 'K. COMMANDER' is similar to 'SPEED B.', but does string comparisons on a letter-by-letter basis, and has no fast filing.

BASIC PLUS 3 was developed from SP-5025 by Maurice Hawes, for his own use. He added 'BASIC EXTENSIONS', 'BASIC PLUS 2', bits from the others, and his own ideas. The story of 'BASIC PLUS 3' was published in PCW, March 1983 pp.178-181. BASIC PLUS 3 is now in the Club Library, and was reviewed in Vol.9 No.3. The syntax for RENUMBER is <RENUMBER mm-nn/new start,inc>, and any one or all of the parameters may be omitted; RENUMBER defaults to 1000,10.

There are 3 other improved Basics derived from SP-5025, all in the Library. They are SP-5025.K2, SP-5060VME & SP-5060VME/EPSON.

SP-5025.K2 was developed by Peter Tuffs, and was described in Vol.5 No.2 pp.27-31 and Vol.7 No.1 pp.19-30. It adds repeating keys, string comparisons, toggle on SML/CAP, an improved CHR$, DELete lines, LIST hold, PRINT@, PRINT/A, PRINT/N, PRINT/S, RENUM, RESTOR nn, and RETURNP. The RENUM syntax is <RENUM L1-L2:L3-L4>. L1 is the new first line number, L2 is the step, and L3-L4 defines the block to be renumbered; RENUM defaults to 10-10. The command PRINT/A sends PRINT and PRINT/P to screen AND printer; PRINT/N restores normality; and PRINT/S dumps the screen to the printer. RETURNP is RETURN AND POP, for jumping out of a subroutine.

SP-5060VME is covered in SOFTWARE MANUAL I; the 'EPSON' version prints Sharp graphics by bit-image. All versions of SP-5060 have one drawback; the 'work area' starts at $4700, not $4400 as in SP-5025, and this may upset m/c routines or sophisticated POKEs. Also, the SP-5060 'PEEK PROTECT' flag is moved, to $41A4 (16804).

Disk users have 2 choices; Knight's 'DISK COMMANDER' adds to SP-6015, giving key-repeat, a numpad, function keys, AUTO, DELETE, DUMP, RENUMBER, & TRACE (with STEP). The disk version of SP-5060, called SP-7011, is the other choice. Both are in the Library.

B) OTHER BASICS ON THE MZ-80K

In 1980, Crystal Research produced 'XTAL BASIC 2.2'. It ran on the MZ-80K and other machines, and was later developed to 'XTAL BASIC 3.1'. 'XTAL' Basics are NOT in the the Club Library.

HU-BASIC is a powerful Microsoft-style Basic, with 16 s.f. The tape version is described in SOFTWARE MANUAL I.; the disk version is called 'SBM SUPERBASIC', and was reviewed by Peter Tuffs in Vol.8 No.2, on pages 20-27. Both versions are in the Club Library.

C) MODIFIED STANDARD BASICS AND THEIR TOOLKITS ON THE MZ-80A

The tape 'KNIGHT COMMANDER' adds DELETE, DUMP, RENUMBER, and TRACE (see Vol.10 No.2 p.33). It is now in the Club Library.

The amazingly comprehensive (and memory-consuming) BAS MOD toolkit for SA-5510 adds about 50 commands, including the very useful KLOAD for loading MZ-80K Basic programs and converting them to suit the MZ80A. (There is also a version of BAS MOD which converts SA-5510 itself to run on the MZ-80K!!) The sheer size of BAS MOD is a drawback, but if you are interested the Editor can supply a photocopy of pp.14-15 of Vol.6 No.2 for £1; this lists all the added commands, with brief explanations. The full Manual fills 43 pages and costs £6 as a 'Special Request' item.

More realistically, 'XPATCH' for SA-5510 is an extremely useful toolkit, and the MZ-80A Sub-editor swears by it. It offers full string comparisons on a 'content before length' basis, which can be switched to ignore case; this is EXTREMELY useful, given the crazy Sharp lower-case ASCII set. 'XPATCH' also offers faster data filing and a LIST hold. New commands are CLS, PRINT@, RENUM, MOVE, DELETE, APPEND, FIND (strings) with selective DELETE, CHANGE, COMPRESS (removes REMs and unwanted spaces), XREF and XREF/P (to list the variables in a program, and where they occur). The syntax for RENUM is <RENUM mm-nn/new start,inc>, and any one or all of the parameters may be omitted. RENUM on its own defaults to 10,10.

Disk users of SA-6510 are less well served. We think there MIGHT be a 'DISK COMMANDER' for SA-6510, otherwise there is nothing!

D) OTHER BASICS ON THE MZ-80A

The tape versions of SP-5060 and HU-BASIC have been converted by members to run on the MZ-80A. The MZ-80K versions are covered in SOFTWARE MANUAL I; the MZ-80A versions are very similar.

Disk users who have CP/M can run MICROSOFT BASIC v5.1. Also, if you run MALLARD BASIC under CP/M on another machine, it is not difficult to convert it to run under CP/M on the MZ-80A.

Crystal Basics run on the MZ-80A (2.2 on tape, and 3.1 on disk under CP/M), but they are NOT in the Library.

E) MODIFIED STANDARD BASICS AND THEIR TOOLKITS ON THE MZ-80B

Knight's 'COMMANDER' programs are available for SB-5510 and for SB-6510. They add DELETE, DUMP, RENUMBER, TRACE (with STEP), and CLEAR SCREEN and HOME using Sharp cursor graphics 'C' and 'H'.

Josef Riha's 'special' version of SB-6511 was fully covered in Vol.8 No.3; this is still available as a 'back issue'.

F) OTHER BASICS ON THE MZ-80B

Disk users who have CP/M can run MICROSOFT BASIC v5.1. Also, if you run MALLARD BASIC under CP/M on another machine, it is not difficult to convert it to run under CP/M on the MZ-80B.

Crystal Basic 3.1 runs under CP/M, but it is NOT in the Library.

G) MODIFIED STANDARD BASICS AND THEIR TOOLKITS ON THE MZ-700

The MZ-700 Basics are satisfactory, but there is a cut-down version of BAS MOD for tape Basic 1Z-013. It adds REPEAT..UNTIL, WHILE..WEND, PUT, DOKE, DEEK, BEEP, TONE, QUIET, IF THEN ELSE, GOTO LABEL, and the logical operators AND, OR, NOT. This version of BAS MOD is available as a 'Special Request' item, price £2.

There are no toolkits for either of the 'standard' MZ-700 disk Basics 2Z-009 and 5Z-008; they are more comprehensive than tape Basic 1Z-013, and do not really need toolkits anyway. But there IS a most important alternative disk Basic from Germany, and as it is clearly developed from tape Basic 1Z-013, we deal with it here.

'K&P DBASIC 700' as it is called, is from Kirsten and Partners in West Germany. It was sold in the U.K. by Sharpsoft Ltd., on 3.5" single-sided disks, but was also available in West Germany on 5.25" floppy disks, and copies in this format have reached the U.K. This Basic has the same commands as 1Z-013 plus the usual Sharp disk-operating commands DIR, LOCK, UNLOCK, DELETE, CHAIN, SWAP, and RENAME, It also has MERGE, CSRH and CSRV.

'K&P DBASIC' may be run on an MZ-700 fitted with an Expansion Box, an MZ-80A floppy-disk interface card modified by changing one ROM chip on it, and any 'Shugart-compatible' 5.25" disk drives (including standard Sharp drives). Its one great advantage over the standard Sharp 5.25" disk Basic 2Z-009E is that it will load programs off an MZ-80A or MZ-80B disk and AUTOMATICALLY convert the tokens to MZ-700/800 tokens before it says 'Ready'.

This trick is possible because K&P DBASIC labels its OWN Basic source files as 'TYPE 5'. When it encounters a 'TYPE 2' file, it assumes that it is an MZ-80A/B Basic source file, and converts the tokens accordingly.

IZ-013 tape Basic also labels its own source files as 'TYPE 5'; but when it encounters a 'TYPE 2' file it assumes that it is an MZ-80K file, and converts the tokens on that basis.

Therefore, on an MZ-700 'armed' with 1Z-013 and 'K&P DBASIC', the user can load in programmes from ANY of the earlier Sharp machines! Very useful - but guess what happens if you try to load in a program from any other MZ-700 disk Basic, in which source files are labelled 'TYPE 2'.....

H) OTHER BASICS ON THE MZ-700

Tape users can run SP-5060VME and VME/EPSON, and HU-BASIC; both have been converted by members and are in the Library.

Disk users who have CP/M can run MICROSOFT BASIC v5.1. Also, if you run MALLARD BASIC under CP/M on another machine, it should be possible to convert it to run under CP/M on the MZ-700.

J) TOOLKITS AND OTHER BASICS ON THE MZ-800

The standard MZ-800 Basics are very powerful, especially on the graphics side, do not need toolkits. However, if you have disks and run Sharp's PCP/M, you can use Microsoft Basic.

DISPLAY CODES ON THE MZ-80K (IN DECIMAL)

Code	Symbol	Code	Symbol	Code	Symbol	Code	Symbol	Code	Symbol	Code	Symbol	Code	Symbol	Code	Symbol
0	SP	32	0	64	SP	96	7C	128	SP	160		192		224	
1	A	33	1	65		97	!	129	a	161		193		225	
2	B	34	2	66		98	"	130	b	162		194		226	
3	C	35	3	67		99	#	131	c	163		195		227	
4	D	36	4	68		100	$	132	d	164		196		228	
5	E	37	5	69		101	%	133	e	165		197		229	
6	F	38	6	70		102	&	134	f	166		198		230	
7	G	39	7	71		103	'	135	g	167		199		231	
8	H	40	8	72		104	(	136	h	168		200		232	
9	I	41	9	73		105	)	137	i	169		201		233	
10	J	42		74		106	+	138	j	170		202		234	
11	K	43		75		107	*	139	k	171	ü	203		235	
12	L	44		76		108	,	140	l	172	ö	204		236	
13	M	45	/	77		109	-	141	m	173	Ü	205		237	
14	N	46		78		110	.	142	n	174	Ä	206		238	
15	O	47		79		111	/	143	o	175	Ö	207		239	
16	P	48		80		112		144	p	176		208		240	SP
17	Q	49		81		113		145	q	177		209		241	
18	R	50		82		114		146	r	178		210		242	
19	S	51		83		115		147	s	179		211		243	
20	T	52		84		116		148	t	180		212		244	
21	U	53		85	@	117		149	u	181		213		245	
22	V	54		86		118		150	v	182		214		246	
23	W	55		87		119		151	w	183		215		247	
24	X	56		88		120		152	x	184		216		248	
25	Y	57		89		121		153	y	185		217		249	
26	Z	58		90		122		154	z	186		218		250	
27	€	59		91		123		155	ä	187		219	o	251	
28		60		92		124		156		188		220		252	
29		61		93		125		157		189		221		253	
30		62		94		126		158		190		222		254	
31		63		95		127		159		191		223		255	

Note: SP represents a space or blank.

The above diagram is reproduced by permission of SHARP CORPORATION

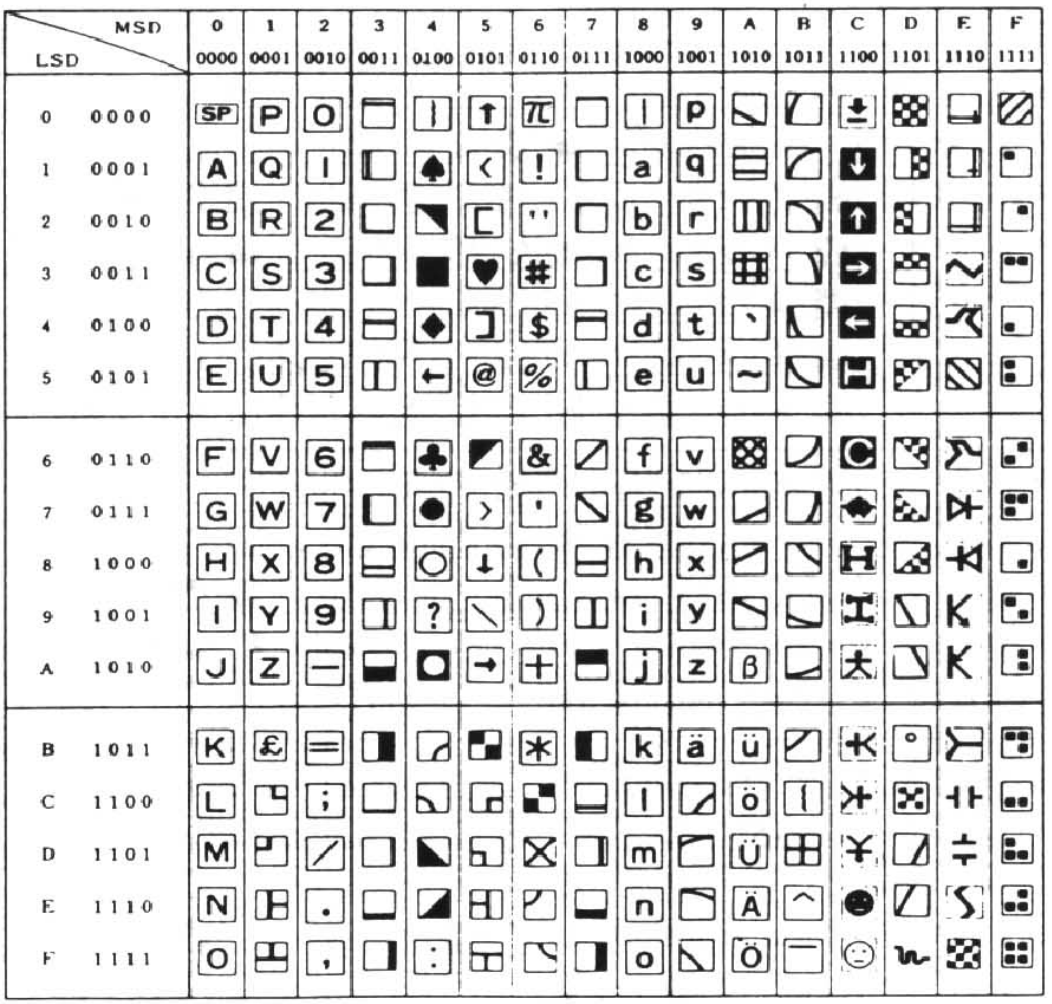
SHARP 'ASCII' CODES ON THE MZ-80K (IN DECIMAL)

Code	Symbol	Code	Symbol	Code	Symbol	Code	Symbol	Code	Symbol	Code	Symbol	Code	Symbol
32	SP	64	@	96		128	SP	160	q	192	SP	224	
33	!	65	A	97		129		161	a	193		225	
34	"	66	B	98		130		162	z	194		226	
35	#	67	C	99		131		163	w	195		227	
36	$	68	D	100		132		164	s	196		228	
37	%	69	E	101		133		165	u	197		229	
38	&	70	F	102		134		166	i	198		230	
39	'	71	G	103		135		167		199		231	
40	(	72	H	104		136		168	Ö	200		232	
41	)	73	I	105		137		169	k	201		233	
42	*	74	J	106		138		170	f	202		234	
43	+	75	K	107		139		171	v	203		235	
44	,	76	L	108		140		172		204		236	
45	-	77	M	109		141		173	ü	205		237	
46	.	78	N	110		142		174	ß	206		238	
47	/	79	O	111		143		175	j	207		239	
48	0	80	P	112		144		176	n	208		240	
49	1	81	Q	113		145		177		209		241	
50	2	82	R	114		146	e	178	Ü	210		242	
51	3	83	S	115		147		179	m	211		243	
52	4	84	T	116		148		180		212		244	
53	5	85	U	117		149		181		213		245	
54	6	86	V	118		150	t	182		214		246	
55	7	87	W	119		151	g	183	o	215		247	
56	8	88	X	120		152	h	184	l	216		248	
57	9	89	Y	121		153		185	Ä	217		249	
58	:	90	Z	122		154	b	186	ö	218		250	
59	;	91	[	123	o	155	x	187	ä	219		251	
60	<	92	¥	124		156	d	188		220		252	
61	=	93	]	125		157	r	189	y	221		253	
62	>	94	↑	126		158	p	190		222		254	
63	?	95	←	127		159	c	191		223		255	

Note: The code is based on the decimal system. SP represents a space.

SHARP DISPLAY CODES ON THE MZ-80A (IN HEXADECIMAL)

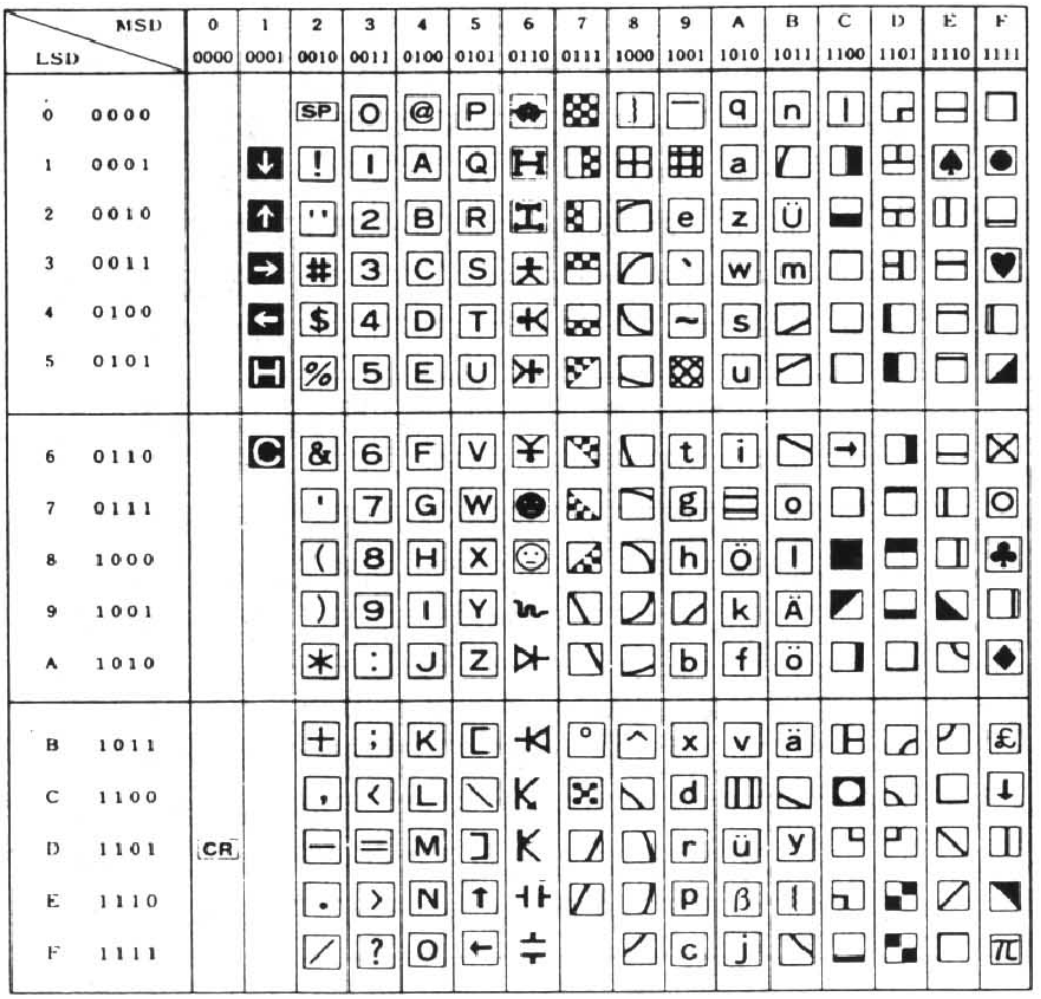
NOTE: This set of Sharp DISPLAY codes is also used on the MZ-700, except that on that machine the character at $F0 is a SPACE.



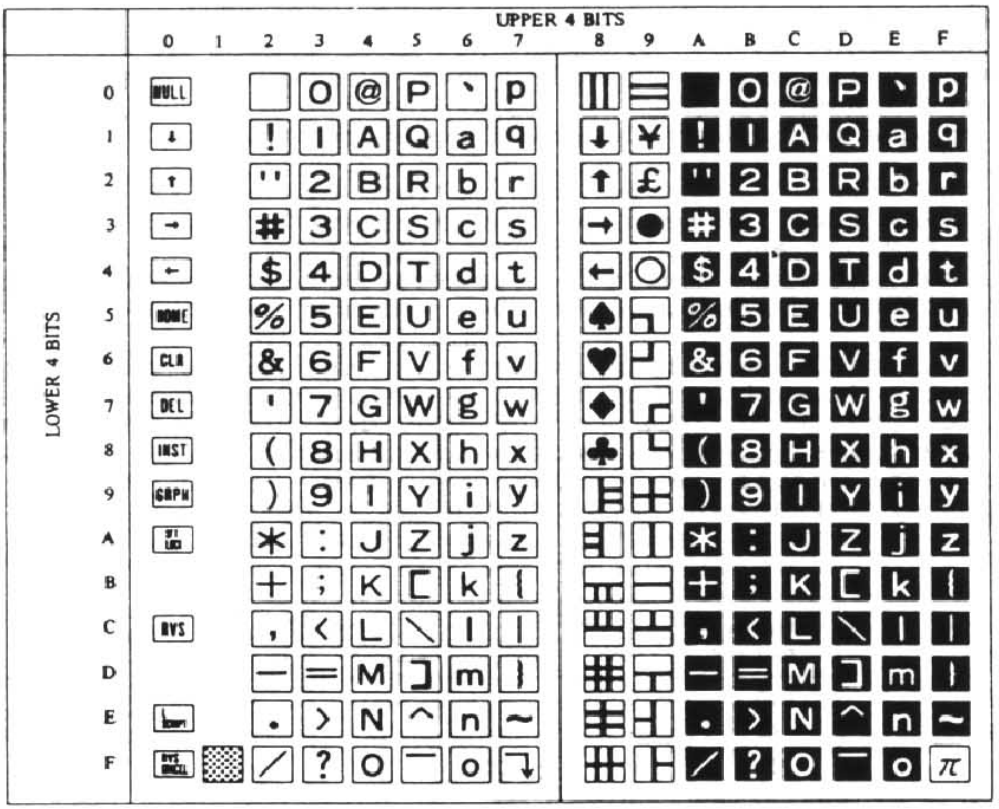
The above diagram is reproduced by permission of SHARP CORPORATION

SHARP 'ASCII' CODES ON THE MZ-80A (IN HEXADECIMAL)

NOTE: This set of Sharp 'ASCII' codes is also used on the MZ-700 and MZ-800, except that on both those machines the character at $6C is a diagonally-hatched square, and the character at $7F is the PRESS PLAY symbol.



SHARP 'ASCII' CODES ON THE MZ-80B (IN HEXADECIMAL)



The above diagram is reproduced by permission of SHARP CORPORATION

MEMORY BANK-SWITCHING ON THE MZ-80B

Memory bank-switching, like keyboard scanning, is more difficult to program on the MZ-80B than on the MZ-80K/A/700/800, because it is done at a lower level. In other words, individual bits of the relevant ports have to be set and reset directly, instead of just sending an OUT or IN command to a 'control' port and letting the computer do the thinking.

There are 3 items to be bank switched; the IPL ROM; the 'text' or 'character' VRAM (which we call C-RAM); and the one or two graphics VRAMs (which we call G-RAMs).

The switching of the IPL ROM does not need to be programmed by the user; the IPL is 'in' the main 64K memory space when the machine is switched on, or when the IPL RESET button is pressed; then, as soon as a program has been loaded at $8000, the IPL switches itself out, and remains out (unless the IPL reset button is pressed). However, for the record, the IPL is switched in by PORT C of the 8255, bit 3, and switched out by bit 1.

The character C-RAM is controlled by bit 7 of PORT $E8. To switch C-RAM into the main 64K space, SET bit 7 of $E8; and to switch C-RAM out and replace it by RAM, RESET bit 7 of $E8. In either case, this involves reading $E8 to see how it is set already, changing the bit, and then writing to $E8.

The graphics G-RAMs are switched at the same time as C-RAM, by the same bit (i.e. bit 7 of $E8). However, the G-RAMs are further controlled by PORT $F4, which decides the 'mode' in which the G-RAMs are placed i.e. whether one of them is 'in' or both of them are 'in'; and whether an individual GRAM, if it is 'in', is set for input or output, or both. The details are as follows:-

Output data to port $F4	V-RAM GRPH I		V-RAM GRPH II	
	Input	Output	Input	Output
00	O	×	×	×
01	×	×	O	×
02	O	O	×	×
03	×	O	O	×
0C	O	×	×	O
0D	×	×	O	O
0E	O	O	×	O
0F	×	O	O	O

Note Input O: V-RAM transfer enabled
× : V-RAM transfer disabled
Output O: shown on CRT display
× : not shown on CRT display

There is a further facility for switching the VRAMs in and out at $5000-$7FFF. This is rarely used; it is done by BIT 6 of $E8.

VARIANTS AND 'QUIRKS' IN SHARP BASICS 1980-84

A) 3 VERSIONS OF MZ-80K TAPE BASIC SP-5025

The 'standard' version of SP-5025 came with nearly all MZ-80K's sold in the U.K. (some early machines were supplied with SP-5010). In 1980, the New Bear Computer Store published a commented listing of 'standard' SP-5025. This listing was withdrawn within a few months, but your Chief Editor (more by luck than judgment) bought it whilst it was still available; £30 seemed a lot to part with in 1980, but it has proved a good investment because SP-5025 is the precursor of all the Basics reviewed in Section B of this Manual.

Sharp U.K. issued two other versions of SP-5025, which carry no external signs that they are different. One such 'rogue' version, on the master disk of SP-6015, differs from 'standard' SP-5025 in only one respect; the printer 'ON/OFF' check routine at $3C66 is the full version found in many other Sharp packages. It adds 10H bytes to the code near the end of the interpreter, and pushes up by 10H the few routines which follow it. The main consequences are that POUT is now at $3C87; and the interpreter ends at $3DEB.

The second 'rogue' version of SP-5025, issued on the master disk of SP-6115 (D.P. Basic) is a different kettle of fish. The code is rewritten using Z80 relative jumps, and new code is added in some places. Everything is relocated, including the 'secret' pokes and the 'warm start' address ($124D). It is possible to POKE and PEEK in hexadecimal, and there is a new (selectable) KBD input routine at $3D26, invoked by POKE $3D25,0 (1); this gives repeating cursor movement, delete and insert keys - very daring! This version of SP-5025 is slightly larger than the original, and ends at $3ED5.

All three versions have PEEK PROTECT, ANTI-LIST and AUTO-RUN flags; and the first two versions have a 'bug' in TI$. The relevant locations (in decimal) are as follows:-

Version	PEEK	ANTI-LIST	AUTO-RUN	TI$ 'BUG'
Standard	10167	10680	10682	7113
'SP-6015'	10167	10680	10682	7113
'SP-6115'	10033	10648	10650	corrected!

The default PEEK flag setting is '0' and means protect ON; any PEEK into RAM below $D000 then returns 32 (SPACE). The default LIST flag setting is '0' and means protect OFF i.e. programs can be listed. The default AUTO-RUN flag setting is '0' and means AUTO-RUN OFF i.e. saved programs will not AUTO-RUN. In all cases, to 'set' the flag, POKE it to 1. With all 3 flags set, any PEEK into RAM below $D000 will return the correct value; programs will not LIST; and saved programmes will AUTO-RUN. To correct TI$ in the 'standard' and 'SP-6015' versions, POKE 7113,87 (89).

The 'SP-6115' version of SP-5025 contains the TI$ correction and also sends $0C,$0A at the start of LIST/P, instead of $0F; these changes, the use of Z80 relevant jumps, hexadecimal PEEKS/POKES, and repeating keys all show that this version of SP-5025 is later than the others and was meant as an 'improvement'. Its 'proper' warm start is $124D; but it will also restart at $1274, without 'READY'. This happens in other versions, and may explain why $1274 is often championed as the 'correct' warm start for SP-5025!

VARIANTS AND QUIRKS IN SHARP BASICS 1980-84 (continued)

B) OTHER QUIRKS IN MZ-80K TAPE BASIC SP-5025

Apart form the PEEK, LIST and AUTO-RUN flags already mentioned, there is another strange 'quirk' in SP-5025 which makes it quite unlike any other Basic ever invented. The CHR$ function changes codes 0-31, 32, 34, and 44 to carriage returns (13, or 0DH).

POKE 10029,0 will modify CHR$ in SP-5025 so that it passes codes 0-31 unchanged (or POKE 9897,0 in the SP-6115 version of SP-5025); but this does not cure the quirk with 32, 34, and 44; to do this, the second half of the CHR$ routine must be completely rewritten. The code below is for 'standard' SP-5025, and changes CHR$ so that it will pass all codes 0-255 unchanged:-

$272C	LD BC, 0001	01 01 00	N.B. This code is
$272F	LD HL, 4400	21 00 44	not suitable for
$2732	PUSH HL	E5	the 'SP-6115'
$2733	LD (HL),E	73	version of SP-5025
$2734	INC HL	23	(nor for other non-
$2735	LD (HL),0DH	36 0D	standard versions
$2737	POP HL	E1	such as SP-5060).
$2738	CALL 18DF	CD DF 18	
$273B	JR 2756	18 19	

All 3 versions of SP-5025 contain a routine which sends control code(s) at the start of LIST/P. In 'standard' SP-5025 these are cut out by POKE 15477,201. In the SP-6015 version, POKE 15493,201; or the SP-6115 version, POKE 15299,201. The hexadecimal values for these addresses are $3C75, $3C85 and $$3BC3 respectively.

C) QUIRKS IN MZ-80K DISK BASIC SP-6015

SP-6015 still contains the PEEK PROTECT flag used in the earlier tape Basic SP-5025. Its location is $1F70 (we give it that way because all Sharp Basics EXCEPT the tape Basics on the MZ-80K can use hexadecimal numbers with PEEK and POKE). The default setting is '0', meaning PEEK protect ON. To set the flag, POKE $1F70,$1; after which a PEEK into RAM below $D000 returns the correct value. (For the faint-hearted, the decimal equivalent is POKE 8048,1).

Another definite 'quirk' in SP-6015, which will affect you if you modify the printer routine to run a standard 'Centronics' printer, arises because the PRINT/P TAB function sends 'cursor right' codes ($13) to the printer driver. There, they are picked out and converted to spaces ($20).

This is a dangerous procedure, because any replacement printer driver is not likely to respond correctly to $13; and in any case, no other Sharp Basic does things this way. Therefore, whether you intend to install a new printer routine or not, the best thing to do is to change PRINT/P TAB so that it sends $20 (SPACES) to the printer. This is done with POKE $3A79,$20 (or POKE 14969,32).

Finally, the CHR$ function in SP-6015 is still slightly odd, though not as odd as CHR$ in SP-5025. It handles CHR$(32), (34) and (44) correctly, but will not pass control codes i.e. codes below $20. To correct this, POKE $1EE3,0 (or POKE 7907,0). CHR$ will then pass all values 0-255 unchanged.

VARIANTS AND QUIRKS IN SHARP BASICS 1980-84 (continued)

D) ARITHMETIC 'BUG' IN THE DOUBLE-PRECISION BASICS

Both of the double-precision disk Basics (i.e. SP-6115 for the MZ-80K, and SP-6610 for the MZ-80B) suffer from a 'bug' in their common BCD-coded 16 d.p. floating-point package. The 'bug' shows up in at least two different sets of circumstances, and may do so in others which have not yet been documented. This is a pity, as these packages are otherwise very useful, and perform well in normal business calculations involving addition and subtraction.

The 'bug' surfaces most obviously in arithmetic division, if the first significant figures in the numerator represent a smaller number than the corresponding figures in the denominator e.g:-

PRINT 1/3 displays the answer 0.333333333333333

What is wrong here ? Not a lot, you may think - except that the answer is only given to 15 s.f. This in itself is not incorrect, but it indicates to the expert that something has gone wrong with the internal F.P. coding. This is confirmed by the appearance of more serious errors in other situations e.g:-

```
        PRINT 1/3E20             displays  3.33333333333333E-20
or      PRINT 1/3 - INT(1/3)     displays -0.333333333333333
```

In the first example, the answer is ten times too big, and in the second it is wrongly negative. But more significantly to the expert, it is only given to 15 s.f. in both cases. A pointer to the cause of the problem is that multiplying by 1 will produce a correct-looking result, but only to 15 s.f.:-

```
        PRINT 1/3E20*1           displays 3.33333333333333E-21
or      PRINT 1/3 - INT(1/3)*1   displays 0.333333333333333
```

It seems that the internal coding of the result has lost one place, but can be 'fiddled' by multiplying by 1. Clearly, this is not a reliable situation, and in the absence of a proper 'fix' for the bug, a better solution is required. This is to multiply the top and bottom of the fraction by such a factor that the top starts with a higher significant figure than the bottom e.g:-

```
        PRINT 9/27E20            displays  3.333333333333333E-21
and     PRINT 9/27 - INT(9/27)   displays  0.3333333333333333
```

Both of these answers are correct AND contain 16 significant figures, indicating that the internal coding is correct.

****

For the moment, you must write all double-precision programs to avoid arithmetic division in which the significant figures are 'bottom-heavy'. This is undoubtedly a tedious restriction, but it is not impossible, and is preferable to producing answers which are ten times to big, or negative when they should be positive!

Sharp were informed of this 'bug' many years ago, but we have never heard of a solution. If any member can find time to solve this one, he would make lots of friends!

VARIANTS AND QUIRKS IN SHARP BASICS 1980-84 (continued)

E) THREE VERSIONS OF MZ-80A DISK BASIC SA-6510

Sharp produced (at least) 3 versions of the disk Basic for the MZ-80A.They secretly date-stamped each version, so to start off, here is a program to identify the version you are using:-

```
10 PRINT " c ": FOR N=0 TO 10
20 PRINT CHR$(PEEK(22503+N));
30 NEXT:PRINT:PRINT:END
```

The versions identified so far are MAR.15,'82, AUG.20,'82, and NOV.18,'82. We think they are the ONLY date-stamped Basics, so if you know of any others (MZ-80A or otherwise) please tell us!!

The reasons for the AUG.20 changes were given in an official Sharp 'Technical Report' which was reprinted in Vol.7 of the Sharpsoft User Notes. Through the Japlish, it seems that the first SA-6510 was subject to ERROR 50 when accessing random files.

Before this report was brought to our attention, we had looked at the rather complicated AUG.20 changes and decided that they trapped syntax errors in AUTO RUN programs. We now have to concede that there are also code changes relating to disk routines.

In contrast, the NOV.18 modifications are simple, and definitely relate to disk drives; they just introduce extra calls to $3B61, in two places. This call certainly relates to disks, though we are not sure of its purpose. G. Chapman's report (in Vol.10 No.1 p.65) of ERROR 50 with sequential files may be relevant here.

To summarise, the changes are not extensive, and the relatively tiny changes in the code seem to be aimed at technical problems with disk drive timing. There may or may not be a subsidiary aim, to trap syntax errors in AUTO RUN programs. The real danger in all this is that the later versions of SA-6510 contain patches which occupy early bytes ($57F2-5801) in the area which is 'spare' in the original version. Therefore, if you use either of the later versions, a machine-code program which utilises $57F2-5801 may cause problems. In particular, early versions of 'DISK>CMT' assume that $57F2-5801 is 'spare', and do cause problems; but the latest version, DISK>CMT.A6', starts by ensuring that the area is safe to use, by restoring SA-6510 to its to its original MAR.15 coding.

F) TWO VERSIONS OF MZ-80B TAPE BASIC SB-5510

As reported on p.55 of Vol.8 No.3, it appears that there is a German version of SB-5510 in which most of the code sits 4 bytes higher than in the U.K. version. The German version does not seem to have reached here, but you should be aware of its existence.

G) LINE LENGTH CORRECTION WHEN COPYING MZ-700 TAPE BASIC 1Z-013

In the RAM Monitor section of 1Z-013, location $004F stores the maximum length of the keyboard input line. The default value $FF, used by Basic, is changed to $64 when you enter the RAM Monitor. Therefore, if you use this Monitor to copy Basic 1Z-013 from RAM, make sure that you POKE $004F,$FF before making the SAVE. This is NOT required when copy 1Z-013 from tape (e.g. with 'SUPERTAPE 2').

RUNNING STANDARD ASCII PRINTERS UNDER SHARP BASICS

Sharp went out of their way to make life difficult for anyone who wishes to run a standard 'Centronics-compatible' ASCII printer on a Sharp MZ computer. First of all, the electrical signals are not compatible - two of control signals required by a 'Centronics parallel' printer are inverted in the Sharp system. Secondly, whereas all modern 'Centronics parallel' printers use a standard or near-standard set of control codes (such as that for the Epson FX-80), the control codes used by Sharp printers are completely different and, in most cases, completely incompatible. Last but by no means least, MZ-80K/A/700/800 computers use a very non-standard ASCII set in which the codes for lower-case letters are scattered about in random and non-sequential locations between $92 and $BD, instead of being in one continuous block (a-z) at $61-$7A.

Nevertheless, on account of the high cost of Sharp peripherals in the early 80's, hardware suppliers such as Peterson Electronics and Mills-Harris Associates produced reasonably-priced units which solved the problems in the same way; the two rogue control signals were inverted, and the data codes could be altered or let through unchanged, in a selectable manner, via a switchable eprom.

Most such interfaces are no longer available, though Petersons will manufacture to special order, and Solo Software and the Sharp Users Club can still supply units for the MZ-700. But in any case such units do have a serious drawback; their performance is fixed at the design stage and cannot be altered by the user. Given the complicated nature of the system, a situation will arise, sooner or later, for which the interface was not designed. Then, the only solution is to switch the interface to 'transparent', to let all codes through unchanged, and tackle the problems in software.

In the long run it is beneficial if the interface is permanently set to 'transparent' and all your software is modified to suit it; in this way, you always know exactly what is happening, and it happens logically. If you rely on interface card special settings, you tend to treat the interface as a 'magic box', and are liable to learn next to nothing about what is actually happening, mainly because it is all very confusing anyway. This is especially true as far as control signals are concerned; Sharp software converts one set of crazy Sharp codes to another set of crazy Sharp codes, and the interface eprom then converts the second set of crazy Sharp codes to standard 'Centronics' codes (if it is lucky!).

We therefore recommend that, if you already have a switchable commercial 'Centronics' interface for your Sharp computer, you should use its 'transparent' setting all the time, and modify all your software to suit. You will eventually find that this is a far better way of working, enabling you to take full advantage of ALL the features of a modern 'Centronics-compatible' printer.

Whether you have such an interface or not, read on! If you use a Sharp interface card, we shall show you how to make simple changes to the card and to Sharp Basics, so that a 'standard' printer will run on your computer. If you have a non-switchable 'Centronics' interface which is 'transparent', the software changes alone may still be very useful. And if you have a switchable 'Centronics' interface, the software changes alone will enable you to leave the interface in its 'transparent' setting when using Sharp Basics.

RUNNING STANDARD PRINTERS UNDER SHARP BASICS (continued)

MODIFYING SHARP PRINTER HARDWARE TO SUIT A STANDARD PRINTER

The first thing to do is to cut two tracks in your printer I/F circuit, and reconnect them to produce inverted signals:-

MZ-80K I/F Card	IC2	Cut track to pin 3, connect to pin 2
	IC2	Cut track to pin 6, connect to pin 7
MZ-80A/B I/F Card	IC6	Cut track to pin 3, connect to pin 2
	IC6	Cut track to pin 6, connect to pin 7
MZ-700 Main PCB	IC4J	Cut track to pin 2, connect to pin 3
	IC4J	Cut track to pin 15, connect to pin 14
MZ-800	No mods. required - just set the switches!	

Next you must make up a new printer lead with a Sharp plug on one end (of a type to suit your board), and a Centronics plug on the other (to suit your printer):-

Pins 1-9	(Sharp)	straight through to	Pins 1-9 (Centronics)
Pin 10	(Sharp)	crossed over to	Pin 31 (Centronics)
Pin 11	(Sharp)	straight through to	Pin 11 (Centronics)
Pin 12	(Sharp)	crossed over to	Pin 32 (Centronics)
Pins 13-25	(Sharp)	'Signal grounds'	Pins 19-30 (Centronics)

It may be possible to leave off one or both of the crossed-over connections; 10-31 initialises the printer, and 12-32 is an error signal which varies from printer to printer. But note that with some printers 10-31 is ESSENTIAL, to keep pin 31 at +5V.

GETTING THE PRINTER TO WORK IN AN ELEMENTARY WAY

The 'standard' end-of-line character to a 'Centronics' printer is $0A. The 'EPSON FX-80' handbook explains that $0A prints the data in the printer buffer AND does a line feed; but if the buffer is empty or contains only SPACES, $0A just does a line feed. This system is used on the Sharp MZ-80B.

The FX-80 Handbook also explains that $0D just prints the data in the printer buffer, and that if the buffer is empty or all SPACES, nothing happens; except that if the printer is set to AUTO LINE FEED, then whatever has just happened will be followed by a line feed. This is the system used on the MZ-80K/A/700/800.

Many of us have been conditioned into thinking that the '$0D' system is the way to do things. But the '$0A' system is much more flexible, and compatible with CP/M. So we suggest that if you use an MZ-80K/A/700/800, you should modify Basic to send '$0A' at the end of a line, and set your printer to 'AUTO LF OFF'; you will then be in line with the MZ-80B, most other systems, and CP/M.

The next step is to cut out any Sharp test routines; these send characters such as 05, 06, 07, 08, 1B03 and 1B04 to the printer, to test for certain conditions. They do not work with standard printers, and may ring bells, or even stop printing altogether because they 'think' there is something wrong with the printer!

FINAL TUNING OF THE NEW PRINTER ROUTINE

After making the foregoing changes, a standard printer should respond correctly to CAPITAL letters, and print the correct line feeds etc. But it will NOT respond correctly to lower-case letters (except on the MZ-80B); and it will not respond correctly to control codes, because these are still being changed by Basic.

We can solve the two problems at once, if we replace the special Sharp subroutine which changes control codes, by a routine which DOES NOT change control codes, but DOES convert lower-case codes to standard. On the MZ-80B cutting out the control-code subroutine will suffice, because the lower-case codes are already standard.

Finally we must ensure that ALL printer output goes through the new routines (and, on the MZ-80K only, we must modify CHR$).

The changes are numerous, and differ for each of the many Sharp Basics, so Maurice Hawes is writing a series of programs on tape, liberally commented. If you RUN the programs the changes will be done for you, automatically; or you can read the programs and then do the changes yourself one by one, and watch what happens. At the moment the following programs are available from Maurice in Weymouth, just send a blank tape and return postage:-

MZ-80K	MZ-80B	MZ-80A	MZ-700
50XX CENTRONICS	5510B CENTRONICS	5510A CENTRONICS	S-BAS CENTRONICS
6015 CENTRONICS	5510B CENTRONICS	6510A CENTRONICS	K&P CENTRONICS
6115 CENTRONICS	6511B CENTRONICS		009E CENTRONICS
	6610B CENTRONICS		

Other programs will follow, for the newly-discovered 'SP-6115' variant of SP-5025, for MZ-700 Quickdisk Basic, and for all the MZ-800 Basics. Ring Maurice if in any doubt which version to use.

A word of advice; if you decide to delve into the MZ-700/800 Basic printer routines yourself, on tape or disk, note that 'POUT' is in the RAM Monitor. Furthermore, in 2Z-009E (and we suspect in all the Quickdisk and MZ-800 Basics) it is below $1000, and is thus invisible when the ROM monitor is invoked. So you will have to look at it via the RAM Monitor, or make a copy of it on tape (from the RAM Monitor) and transfer it to a different environment (e.g. CP/M) so that you can look at it with a disassembler.

TURNING A PIG'S EAR INTO A SILK PURSE

For those who have a Sharp P5/6 printer, there can be a new life for it at very little expense. No need to modify hardware or make a new lead; just plug an MZ-80B 'ROM' into the printer, and make the software changes! You then have the neo-Centronics facilities of the P6B at your fingertips (even the incremental line feeds required by WDPRO to do underlining!) for the cost of an eprom and some Club software. And then, when you decide to splash out on a new 'Centronics' printer, all your software will be ready, and all you will have to do is make the hardware modifications and the new printer lead. You will then be able to enter a new world with all the printer's facilities at your fingertips!

MAKING COPIES OF SHARP MASTER DISKS

Every Sharp Basic 'Master' Disk contains a disk-copying routine, but in the early Basics copying is either prevented altogether, or is restricted, if the source disk carries a bootable 'Basic'.

On the MZ-80K, the disk-copying programs on 'Master' disks of SP-6015 and SP-6115 are called 'DISKETTE COPY'. The two programs are very different, but both refuse to copy a 'Master'.

On the MZ-80A/MZ-80B all the disk-copying routines on 'Masters' of SA-6510, SB-6510, SB-6511 and SB-6610 are essentially the same, and form part of 'UTILITY'. There are minor differences in coding or location, but all versions copy a 'Master' disk. However, the copy is secretly marked as a 'Sub-master' and cannot be copied.

On the MZ-700/800 the disk-copying routine on Sharp 'Masters' of 2Z-009E, 5Z-008, 2Z-046 and 5Z-009 is part of a program called 'FDCOPY'. This routine will copy any disk, even a 'Master' disk, without changing it in any way, so there are no problems.

The situation on the MZ-80K is intolerable, as it forces you to use your one and only 'Master' disk all the time. The situation on the MZ-80A/B is a little better, but if anything goes wrong with your one and only master disk, your source of copies has gone!

Drastic action was called for and Sharp users, in this country and abroad, devised methods of getting round the problems. Some of these were ephemeral, but eventually all the relevant programs were permanently modified, and now appear on Club disks labelled as version 2. For the record, the changes are given below; they were done by direct disk-editing (there is no MZ-80B version of DISKEDIT, but MZ-80B disks may be diskedited on the MZ-80A).

CHANGES TO SHARP DISK-COPYING PROGRAMS TO MAKE v2.0

"K" DISKETTE COPY 2 (SP-6015)	$30C0 AND $30CB BOTH changed, from $77 to $00
"K" DISKETTE COPY 2 (SP-6115)	$12AE changed from $28 to $18
"B" UTILITY v2.0 (SB-6510/6610)	$13E3 changed from $20 to $18
"B" UTILITY v2.0 (SB-6511)	$13E5 changed from $20 to $18
"A" UTILITY v2.0 (SA-6510)	$1455 changed from $20 to $18

NOTE: the disk-copying programs with the various 3.5" and 5.25" versions of MZ-700 K&P DBASIC are all based on 'UTILITY', and thus require modification along the same lines. However, there are many variants, some of which are heavily rewritten and are renamed as 's.d.s FORMAT/BACKUP'. See Vol.9 No.3 p.52 for full details.

COPYING WITH ONLY ONE DISK DRIVE

The disk-copying sections of UTILITY and FDCOPY will copy from DRIVE 1 to DRIVE 1 - just enter "1" in response to both "DRIVE NO.-" prompts, and follow the instructions VERY CAREFULLY!

DISKETTE COPY does NOT offer a one-drive facility; if you have only one drive on your 'K', use John Edwards' 'SUPERDISK 1D'.

EXAMPLES OF SHARP MUSIC STRINGS

The first example comes from a program which is 'hidden' on every Master copy of Sharp MZ-80A Disk Basic SA-6510.

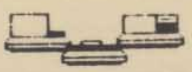
This copy was 'undeleted' by using J. Edwards' 'DISKEDIT' and it ran (rather fast!) at the first time of asking!!

```
5 REM From "The Well-tempered Klavier" by J.S. Bach
6 REM
7 REM Salvaged from the depths of an MZ-80A Master disk
8 REM
10 A$="C1EG+C+EG+C+E":B$="CDA+D+FA+D+F":C$="-BDG+D+FG+D+F"
20 D$="CEA+E+AA+E+A":E$="CD#FA+D#FA+D":F$="-BDG+D+GG+D+G"
30 G$="-BCEG+CEG+C":H$="-ACEG+CEG+C"
40 I$="-D-AD#F+CD#F+C":J$="-G-BDGBDGB":K$="-G-#AEG+#CEG+#C"
50 L$="-D-ADA+DDA+D":M$="-F-#GDFBDFB":N$="-E-GCG+CCG+C"
60 O$="-E-F-ACF-ACF":P$="-D-F-ACF-ACF":Q$="-G-D--G-BF-G-BF"
70 R$="-C-E-GCE-GCE":S$="-C-G-#ACE-#ACE":T$="-F":TT$="-F-ACE-ACE"
80 U$="-#F-C-AC#D-AC#D":V$="-#G-F-BCD-BCD":W$="-G-F-G-BD-G-BD"
90 X$="-G-E-GCE-GCE":Y$="-G-D-GCF-GCF":Z$="-G-D-G-BF-G-BF"
100 AA$="-G-#D-AC#F-AC#F":BB$="-G-E-GCG-GCG":CC$="-G-D-GCF-GCF"
110 DD$="-G-D-G-BF-G-BF":EE$="-G-C-G-#AE-G-#AE"
120 FF$="-F-C-F-ACFC-AC-A-F-A-F-D-F-D-C-BGB+D+F+DB+DBGBDFED-C0CEG+C5"
999 TEMPO 5
1000 MUSIC A$,A$,B$,B$,C$,C$,A$,A$,D$,D$
1001 MUSIC E$,E$,F$,F$,G$,G$,H$,H$,I$,I$,J$,J$
1002 MUSIC K$,K$,L$,L$,M$,M$,N$,N$,O$,O$
1003 MUSIC P$,P$,Q$,Q$,R$,R$,S$,S$,T$,TT$,T$,TT$
1004 MUSIC U$,U$,V$,V$,W$,W$,X$,X$,Y$,Y$,Z$,Z$
1005 MUSIC AA$,AA$,BB$,BB$,CC$,CC$,DD$,DD$,EE$,EE$,FF$
```

The second example comes from a program which was written by Maurice Hawes, to run on the MZ-800 which was loaned to the S.U.C. for the 1986 PCW Computer Show. It shows how the MZ-800 can play MUSIC in three parts.

```
10 M1$="O3D8D5A5G4R1G3F3E4R1D4R1D6#D3E3R3
20 M2$="O3C8O2B7O3C5E3D3C4R1O2B4R1B7O3C3R3
30 M3$="V12O2#F5G5#G5G5F5E4R1F5G4R1G4R1G7C3R3
40 MUSIC M1$;M2$;M3$
```

Would you believe, this program runs on earlier Sharp computers! But not in 3 simultaneous parts; that would be a miracle!!!



SHARP USERS
CLUB