INTRODUCTION

Thank you for buying this copy of Sharp Zen.If you have any questions re Zen please direct them,in writing,via our distributors.
Zen consists of an Editor,an Assembler and an Object Debugger. The editor lets you create and modify a source file consisting of statements in the Z80 Assembly Language.The assembler translates these statements into object code in memory. The debugger allows you to execute object code in a controlled manner.

The primary advantages of assembler programming over BASIC programming are flexibility,compactness and speed of execution.As an example consider the fact that using the BASIC SET x,y statement it takes about twenty seconds to fill the screen.

At assembler level it can be done in about twenty milli-seconds, a thousand to one improvement.

STARTING UP

(1) Insert Zen cassette,rewind if necessary.
(2) Type LOAD then press the (CR) key.
(3) Depress cassette recorder PLAY key.

After about ten seconds the SP1002 monitor will display the file name and commence loading Zen into memory.After a further twenty seconds the file will have been loaded and you will hear a beep as Zen takes control.

COMMAND LEVEL

Whenever the prompt ZEN is displayed you are at command level. You can select any one of the following commands:

Source edit

Z ... ZAP numeric
E ... ENTER
N ... NEW

Source position

D ... DOWN numeric
U ... UP numeric
T ... TARGET numeric
L ... LOCATE string
P ... PRINT numeric

Source global

A ... ASSEMBLE
H ... HOWBIG
K ... KILL
S ... SORT character

Cassette

R ... READ character
V ... VERIFY character
W ... WRITE character

Object

C ... COPY
F ... FILL
G ... GOTO numeric
I ... IN numeric
M ... MODIFY numeric
O ... OUT numeric
Q ... QUERY numeric
X ... XAMINE

The default command, i.e. just typing (CR), will clear the screen.

COMMAND SYNTAX

To select a command type in the first letter of the command name, followed
by a parameter if necessary, then press (CR). The type of parameter required
by a given command is shown in lower case in the above command table.

The types are:

    Character .... a single character.
    String .... a sequence of characters.
    Numeric ...... a decimal, hexadecimal or octal number. Decimal is the default
                   base. Hex numbers are 'H' postfixed, octal are 'O' postfixed.

All the commands which take a parameter will adopt a default value if you
don't actually supply one. All commands are explained in detail later on.

If Zen doesn't understand anything you've typed in it will display the error
message HUH? and beep at you (negative reinforcement).

USER INPUT

Whenever you are required to input from the keyboard, whether at command level
or anywhere else, the following rules apply:

    (1) Zen takes no action until you press (CR)
    (2) DELETE can be used to backspace
    (3) You cannot backspace past the start of your input
    (4) You cannot type beyond the end of the video line
    (5) CAP/SMALL can be used to change case
    (6) All other keys are interpreted as data characters

SOURCE EDIT COMMANDS

ZAP....... Deletes parameter lines from the source file commencing with
the current line.
  Z74(CR) would delete seventy-four lines
  Z(CR)would delete one line,the default value
Command cancels itself if it runs into the end of file,you get
a beep and EOF is displayed.Otherwise the new current line is
displayed.

ENTER..... This is the command which enters lines of text into the source
file.It is a continuous mode command. When you first enter this
command Zen will display a line number to show you your position
in the file.Type in a line of text terminated by (CR).
Zen will display the next number and so on. To exit from this
command type a full stop '.' at the beginning of a line.
You may enter text anywhere in the file.Text is entered at the
current with the old current line, and all following lines, being
moved downwards towards EOF.

NEW....... This command lets you modify the current line.The line number
and the line itself will be displayed and the cursor will be at
the end of the line.Change the line then type (CR) and the line
will be restored to the file.
If you are at EOF you will get a beep and the EOF message.

SOURCE POSITION COMMANDS

DOWN...... Moves you down the file parameter lines and makes that the new
current line.
  D13(CR) would move you down thirteen lines
  D(CR) would move down one line,the default value
Command cancels itself at EOF as in ZAP.

UP........ This command is the complement of DOWN.Cancels at SOF.

TARGET.... Moves you to parameter line,makes it current and displays it.
  T52(CR) would move you to the fifty-second line
  T(CR) moves you to SOF, default value
Cancels at EOF as in ZAP.

LOCATE.... Moves you to the first line containing the reference string.
  LBIT 7,(HL)(CR)
Would move you to the first line containg the string BIT 7,(HL)
makes it current and displays it.Cancels at EOF as in ZAP.
The file is searched downwards from the line AFTER the current
line,you'll see why when you've tried it a few times.
There are no restrictions on the string content.

PRINT..... Displays parameter lines of the file beginning with current.
Last line displayed is the new current line.
  P9(CR) would display nine lines
  P(CR) would display one line,default value
Cancels at EOF as in ZAP.

SOURCE GLOBAL COMMANDS

ASSEMBLE...Assembles the source file from the SOF to the END pseudo-op.
Upon entry you will be prompted for a list option,these are:

    V(CR) List to video
    E(CR) List to external device,typically a printer
    (CR) Default option generates no output.This is the fastest
          option and should be used until all source errors are
          eliminated.

HOWBIG.... Shows you how large the source file is.The SOF(Start of file),
EOF(End of file) addresses plus the address of the top of
memory are displayed in hex.

KILL...... Erases the source file completely, as in the BASIC NEW state-
ment. When the file is KILLed the EOF is equal to the SOF.
This is the state of the file when you first load Zen from
cassette.
It is possible to recover an accidentally KILLed file,see
later.

SORT...... Alphabetically sorts and lists the symbol table built up
during the previous assembly.Upon entry you will be prompted
for a list option,these are:
  V(CR) List to video
  E(CR) List to external device
Output is generated a page at a time (see ASSEMBLY OUTPUT).
You can restrict the sort process to part of the table by
entering a selector character with the command letter.
 SB(CR) would only list symbols beginning with the letter 'B'
The default parameter lists the whole table.

CASSETTE COMMANDS

Zen can READ,VERIFY or WRITE source or object files.Zen source files are an
extension of the existing Sharp file structure.All Sharp files have the same
format and differ only in the file type number, i.e.:

Object 1
BASIC source 2
BASIC data 3
Zen source 4

To select between object and source files you add a character parameter to the
command letter.The default command always applies to source files:

R(CR) READ source file   RO(CR) READ object file
V(CR) VERIFY source file   VO(CR) VERIFY object file
W(CR) WRITE source file   WO(CR) WRITE object file

Zen checks the file type as well as the name so there's no chance of picking
up the wrong type of file, even if the names are the same.

You can abort from any of these commands by pressing the BREAK key.

READ ... Upon entry you will be prompted for a file name.If you supply a file
          name then that file is searched for. If you default then Zen loads the
          first file, of the correct type, that it finds.

          Source files are always loaded at the EOF, allowing you to merge files
          together. After the file is loaded the new file size is displayed.

          Object files are loaded at the load address on the cassette.Object
          files are never auto-executed,instead the execution address is placed
          in the User Program Counter for use with the GOTO command.

VERIFY . This command works in the same way as READ but the files are never
          loaded,merely checked against memory.The VERIFY commands must be used
          IMMEDIATELY after a WRITE command.

WRITE .. If writing the source file you'll just be prompted for a name and
          then the command executes.
          With object files you have to supply more information.You'll be prompted
          for START,STOP,EXEC and LOAD addresses.
          Memory is written from START to STOP inclusive,STOP must be greater
          than START or you'll get an error message.
          EXEC defines the execution address of the file,you should bear in
          mind that SP1002 auto-executes any file with an execution address
          greater than 11FFH.
          The LOAD address defines where you want the file to load in memory.
          For example you might want to write from 4000H to 43FFH but have the
          file load back in at 1200H.

COPY .... Copies a block of memory from START address to STOP address
          inclusive to a DESTINATION address.You are prompted for all three
          parameters.

FILL .... Fills a block of memory from START address to STOP address
          inclusive with a DATA constant.You are prompted for all three
          parameters.

GOTO .... Transfers control to a user program for debugging.
              G4603H(CR) would transfer to that address
              G(CR) would transfer to the address in the User PC
          You will be prompted for a BREAKPOINT.Inputting an address para-
          meter will set a breakpoint at that address,the default response
          sets no breakpoint.A breakpoint cannot be set in ROM code, and it
          must be the first byte of an instruction.
          The command loads the Z80 with all the User registers and finally
          sets the Program Counter to transfer control.
          Zen uses a RST 38H breakpoint.When this is encountered in a user
          program the Z80 pushes it's PC onto the stack and vectors to 38H.
          Zen will have set a series of further vectors to direct control
          back to the TRAP handler.
          Trap saves the Z80 machine state in the User Registers and returns
          to command level.You may now use any command. To continue from the
          breakpoint use the G(CR) option .

IN ...... Reads one of the 256 Z80 I/O ports.
              I17(CR) would read port 17
          The data read is displayed in hex and binary.

MODIFY .. Lets you examine and alter memory contents.
              M513H(CR) would commence at that address
          The byte at that address is displayed in hex and Zen waits for a
          parameter from you.If you supply a parameter then it's stored at
          that address,if you default then Zen justs steps to the next address.
          MODIFY and QUERY share a common address default option.If you default
          on the address then operation commences at the last address displayed
          by either command.
          To quit the MODIFY command type a full stop '.' as a parameter.

OUT ..... Writes to one of the 256 Z80 I/O ports.
              OFEH(CR) would write to port 254
          You are prompted for a DATA parameter.

QUERY ... Displays a block of memory in hex and literal ASCII
              Q1200H(CR) would display the start of Zen
          Sixty-four bytes are displayed as eight rows of eight bytes

XAMINE .. Displays the User Registers in hex. The main registers are on the
          first line and the alternate on the second line.

Zen expects source statements to be constructed according to the syntax
defined in the ZILOG Z80 Assembly Language Programming Manual. Each line
of the source file is a statement divided conceptually into, at most, four
fields:

PICTURE:LD HL,(CURSOR); Pick up cursor

Label Operator Operand(s) Comment

We say conceptually divided because the components of a statement don't have
to be specifically positioned into fields. As long as you use the correct
seperators (spaces,commas,etc.) Zen accepts statements in free format.

Comments ..... Comments are ignored by the assembler. They are preceded by
               a semi-colon and are terminated by end of line.

Operators .... There are 74 generic operators (CALL,LD,JP,etc.). In addition
               there are the pseudo-ops described later.

Operands ..... The number of operands in a statement depends upon the operator.
               Examples:

               NOP .... No operand
               CP ..... One operand
               BIT .... Two operands
               JR ..... One or two (JR SYMBOL or JR Z,SYMBOL)

               Operands may be:

               Register names (A,B,HL,IY,etc.)
               Condition codes (Z,NZ,C,PO,etc.)
               Numbers

               Numbers
               Decimal,hex and octal numbers are accepted with decimal being
               the default base. Hex numbers are 'H' postfixed, octal 'O'.
               Numbers must begin with a digit, a leading zero may be necessary
               with some hex numbers.

               The assembler will generate the ordinal value of any character
               enclosed in single or double quotes.

               Symbols are explained in detail later, they can be used anywhere
               a literal number can be used.

               The program counter may be accessed by using the $ character

               All of the above numeric types may be elements of an expression.
               The infix math operators are:

               + Addition   / Division
               - Subtraction   & Logical AND
               * Multiplication   . Logical OR

               An expression may be used anywhere that a simple operand can be
               used. Expressions are evaluated strictly left to right with no
               precedence ordering. Arithmetic is 16bit unsigned integer and
               overflow will be ignored. Elements in an expression need not be
               delimited by seperators as the math operators are implied
               seperators.

ASSEMBLER SYNTAX

Labels ..... A label is a way of marking a statement.Each time you use a
JP,CALL etc you need a way of specifying the destination as
an operand.Assembly language allows you to use a symbolic
name as a label.

SYMBOLS

A symbol is a name with an associated value. The name is used rather than
explicitly stating the value.A symbols value is declared to the assembler
in one of two ways:

(1) By using the EQU pseudo-op (similar to LET in some respects).This allows
you to assign your own value to a symbol.

(2) By placing it at the start of a statement.The assembler will assign the
value of the program counter to the symbol.The symbol is being used as
a label.

Whichever method is used a symbol must be postfixed with a colon ':' when
declared.A symbol MUST begin with a letter but may contain letters or digits
after that.Zen allows symbols of any length but may not include the (non-ASCII)
lower case letters produced by the Sharp keyboard.

The register names and condition code names may not be used as symbols.Any
attempt to do so will result in an error message.

PSEUDO-OPS

These are additional operators which are not part of the Z80 instruction set
but are understood by the assembler.They are used in exactly the same way as
normal operators.

```
END    End assembly...... no operand
DS     Define Storage.... one operand
DW     Define Word....... one operand
DB     Define Byte....... variable operands
EQU    Equate............ one operand
ORG    Origin............ one operand
LOAD   Load memory....... one operand
```

END .... This operator must be used to terminate assembly.Failure to do so
will result in an error message and an incomplete assembly.

DS ..... Skips a number of object locations.Commonly used to reserve space
for a stack or text buffer where memory contents don't need to be
defined.

DW ..... Generates a word (two bytes) in the object file in reversed order
as required by the Z80 sixteen bit instructions.

DB ..... Generates the value of the operand(s) in the object file.Takes as
many operands as desired seperated by commas.
Each operand may be an expression but obviously no operand can have
a value greater than 255.
The program counter will be incremented after each location is filled
as if the operands were on seperate lines.
In addition to the usual operand types each operand may be of the
data type literal string.

MESSAGE:DB'THROWN INTO CONFUSION'

Strings may be of any length, but, unlike the single character literal
cannot form part of an expression.A string is formed in the same way
as a single character literal,by enclosing in single or double quotes.

ASSEMBLER SYNTAX
───────────────

EQU .... assigns a value to a symbol.There are some restrictions on how you
         can form the operand.The value of any symbols used in the operand
         expression must already be known to the assembler,forward referenced
         symbols will result in the UNDEFINED error message.

              CR:EQU BACKSPACE+5
              BACKSPACE:EQU CR-5

         This sequence is ILLEGAL because each symbol is defined in terms of
         the other.The 'no forward reference' rule is designed to prevent
         you ever making such a mistake.In practise you will probably never
         encounter such a situation as most EQUATES have simple operands.

ORG .... Defines the origin,or assembly address, of the object code
         This operator may be used as often as desired to produce sections of
         code at different locations.The same restrictions apply to the
         operand as for EQUATES.

LOAD ... Lets you load the generated object code straight into memory as it
         is produced.You may load anywhere in memory independant of the actual
         origin.If this operator is not used then no location outside Zen
         will be altered in any way.
         Note that use of a subsequent ORG operator turns the loading process
         off, each time you set a new ORG you are specifically required to
         re-establish the loading process.

ERROR HANDLING
──────────────

If the assembler encounters a source error the following will happen:

 Assembly terminates and an error message is displayed and you get beeped at.
 The incorrect line is displayed and becomes the editor current line.
 Zen returns to command level.

You may correct the error and assemble again.It is impossible to make a
syntax error which will damage Zen or the source file.Assembly runs at about
four thousand lines a minute so errors are found very quickly.Here are the
error messages:

UNDEFINED... You've used an undeclared symbol.
SYMBOL...... You have forgotten the symbol(as in EQU) or it is of zero length.
RESERVED.... You have used a reserved name for a symbol.
FULL........ The symbol table is full,see later.
DOUBLE SYMBOL... You have declared the same name more than once.
EOF......... You have forgotten END and have hit EOF.
ORG!........ No origin supplied
HUH?........ Assembler is completely baffled.
OPERAND..... Something wrong with an operand, e.g. LD A,256
                                                     BIT 9,B
                                                     LD(DE),C
                                                     INC AF

         Also included are relative jumps and indexing out of range.

         The assembler will catch all incorrect statements,don't be
         afraid to experiment.

ASSEMBLY OUTPUT

Zen supports two list devices, the 40chr video display and an 80chr external
ASCII device. Zen generates completely formatted output to either device.The
external device driver is user defined and fifty-seven bytes are left free
for this purpose (see Zen listing....EXTERN).
Zen generates listings a page at a time with a delay between pages.The page
length for the external device is set at sixty-two lines though this is easily
changed with the MODIFY command (see Zen listing.....PAGE).

The external device driver only has to recognise two control characters.These
are ASCII FORMFEED (OCH), which is issued at the start of every page, and
ASCII CARRIAGE RETURN(ODH), which is issued after every line.

Pause control

When assembly is in progress you may use the BREAK key to pause.Pressing this
key will hold listing at the END OF THE PAGE, allowing you to change paper on
a sheet feed printer.To restart listing press any key except 'Q', which
causes you to quit assembly and return to command level.
Note that if you have selected the null output list option then the BREAK
key causes an immediate return to command level.

Formatting

The symbol,operand and comment fields of a statement may be of indefinite
length.To prevent line overflow and an unsightly listing Zen will truncate
the source in these fields if necessary.The width of these fields is
extended when you choose the external option.
You may thus prepare source for the external device and feel confident that
it will never become incomprehensible on the video display.
The object field is never truncated in any way.

Sort output

Sort listing is identical to assembly listing. On the video option three
symbols are listed per line,on the external device this is changed to four
per line and the symbol name field is extended.

The accompanying listing and sort were both produced by Zen as was the
cassette.

ODDS and ENDS

Symbol Table

The symbol table is where symbols are stored with their (sixteen bit) value.
The ST is positioned immediately above Zen and underneath the source file.
Sooner or later you'll want to make it larger.As the Start of File Pointer is
also used as the end of table pointer all you have to do is move the source
file higher in memory (at the moment it's at 2500H).

KILL the file
Find the SOFP and EOFP using the listing
Change BOTH pointers to the new position using MODIFY
Save the new Zen on cassette using the WO command.

Saving Zen

When you save Zen on cassette you must be aware of the following:

The file must be KILLed
You must then ASSEMBLE to close the symbol table
You must write up to the VERY last byte shown in the listing

Memory Protection

The source file cannot grow indefinately.Sooner or later memory runs out
and things get tricky.There are three commands which can extend the source
file, these are ENTER,NEW and READ.
All these commands look ahead to see what will happen if they are executed.
If they will exceed the top of memory then you will get an error message of
the form:

    address MEMORY FULL

where address is a hex address showing where the file would have extended to
had the command executed fully.

Normally the top of RAM is used as the memory limit, however you can put
in a limit of your own. Find the word LIMIT in the listing. When this is
set to zero Zen will use the top of RAM as the limit. If you change it to
an actual value (i.e. non-zero) then Zen will let the source file grow up
to, and including, that address but no further.

Recovery

Earlier we mentioned that you can recover a KILLed file, here's how.When
you KILL the file all Zen does is set the EOFP equal to the SOFP, the text is
still there in memory. Find the last byte of the source file,which will be
an ASCII CR. Use the MODIFY command to change the EOFP to this address plus 1.
The source file will re-appear.

Good luck.