TRS-80

ZEN

A Z80 Editor, Assembler, Debugger—LEVEL II

AVALON SOFTWARE
of England

copyright 1979

INTRODUCTION

Thank you for buying this copy of ZEN, we feel sure you'll be satisfied with your purchase. If you have any questions regarding ZEN please direct them, in writing, via our distributors.

ZEN is an integrated system for the production of Z80 machine code. Source statements are entered with the Editor. The source file may be assembled at any time and then executed, under control, with the Debugger.

ZEN offers significant advantages over alternative systems for the TRS-80.

STARTING UP

ZEN is supplied on cassette in the standard LEVEL II object format, the filename is ZEN.
The cassette, together with the accompanying Listing and Symbol Sort, was produced by ZEN assembling it's own source file.

After loading the cassette use the SYSTEM command to enter ZEN. The entry point to ZEN is 17152 decimal or 4300H (Hexadecimal).

The video screen will clear and the command-loop prompt is displayed:

Z>

ZEN now wants a command

COMMAND SET

GLOBAL COMMANDS:

H ... HOWBIG
K ... KILL
S ... SORT   (selector parameter)
A ... ASSEMBLE

CASSETTE COMMANDS:

W ... WRITE   (file type parameter)
R ... READ   (file type parameter)
V ... VALID   (file type parameter)

SOURCE POINTER COMMANDS:

T ... TOP
B ... BASE
U ... UP   (iteration parameter)
D ... DOWN   (iteration parameter)
P ... PRINT   (iteration parameter)
L ... LOCATE (target string parameter)

SOURCE EDIT COMMANDS:

Z ... ZAP   (iteration parameter)
E ... ENTER
N ... NEW

DEBUGGER COMMANDS:

G ... GOTO
C ... COPY
F ... FILL
X ... XAMINE
M ... MODIFY (address parameter)
Q ... QUERY   (address parameter)

USER INPUT

All commands are available at all times.
All user input to ZEN is via an intermediate text buffer,no action is
taken until you hit the ENTER key.
All user input may be edited with the BACKSPACE (Left-arrow) key. While
other control keys may have a visible effect they are NOT RECOGNISED by ZEN.
User lines will be restricted to fifty-nine characters by ZEN.

Commands comprise the key letter and, in some cases, an optional parameter.
Where more than one parameter is required by a given command you will be
explicitly prompted for each parameter rather than entering them at command
level.

Numeric parameters may be Decimal(default),Hex('H' postfix) or Octal('O' post

The default command,ENTER on it's own, clears the video screen.

GLOBAL COMMANDS

H ... HOWBIG,displays the start and end addresses (SOF,EOF) of the source file in hexadecimal.

K ... KILL,erases the source file from memory.After a KILL the start and end addresses of the file are equal.This is also the state of the file when ZEN is initially entered.

S ... SORT,alphabetically sorts and displays the symbol table built during the previous assembly.You can add a selector letter to the command letter which will restrict display to symbols beginning with that letter.Upon entry you will be prompted for an output option, these are:

V ... VIDEO
E ... EXTERNAL

Output characteristics are identical to those of assembly LIST output,see ASSEMBLY OUTPUT for more detail.

A ... ASSEMBLE,will assemble the source file from start of file to the END pseudo-op.Upon entry you will be prompted for an output option, these are:

V ... VIDEO (List to video).
E ... EXTERNAL (List to external device).
C ... CASSETTE (Object to cassette in LEVEL II format).

The default option generates no output.This is the fastest mode and should be used until all source errors are eliminated.If you select the CASSETTE option you'll be further prompted for an EXECUTION ADDRESS and a FILENAME.

e
by ZEN.

eter.
be
ommand

'O' post

CASSETTE COMMANDS

The cassette drive is used to save and load two types of file,the
LEVEL II format OBJECT file and the ZEN format SOURCE file.As the
formats of the two types are completely different we suggest that
you don't mix them on the same cassette.
The three cassette commands apply to both types of file.The default
command, just the command letter, always refers to source files.
Adding the letter 'O' to the command letter causes the command to
apply to object files.
Object files are blocks of position dependant machine code with a
filename up to six characters and an execution address at the end
of the file.
Source files are blocks of position independant text with filenames
of up to thirty characters.

W ... WRITE,will write a file to cassette.In source mode the file is
simply the current memory source file.In object mode you'll be
prompted for a START ADDRESS,STOP ADDRESS and EXECUTION ADDRESS.
Data is written from START to STOP inclusive.
Both modes will prompt for a filename.

R ... READ,will read a file from cassette and load it into memory.Source
files are appended to the end of the current memory file allowing
a simple form of merging.Object files are stored at their load
address.
Both modes prompt for a filename then commence to search for that
file.A default response to the filename request causes ZEN to skip
the name check and pick up the first file it finds.
When the file is located the letter 'L' replaces the cursor.Any
checksum errors will cause an asterisk '*' to be displayed. No
asterisks means the file was read correctly.
After a source file has been read the new file size is displayed.
If the source file exceeds available memory the error message
MEMORY FULL is displayed and loading stops.

V ... VALID,checks that a file on cassette contains no errors.This
command is used in exactly the same way as READ but nothing is
loaded into memory thus allowing you to check the result of many
hours work immediately.It should become a habit to check files
as soon as you have dumped them.

SOURC

T ...
B ...
D ...

U ...
P ...

L ...

SOURC

Z ...
E ...

N ...

NOTE

NOTE

SOURCE POINTER COMMANDS

As the Z80 Assembly Language is entirely line orientated the Editor in ZEN is line,rather than character,orientated. ZEN always maintains an internal pointer to the CURRENT LINE in the source file.
The following commands all move the source pointer.

T ... TOP, pointer moves to start of file.

B ... BASE, pointer moves to end of file.

D ... DOWN, pointer moves down towards end of file (parameter) lines.
Example .... D37 moves pointer down thirty-seven lines.
The default parameter is assigned a value of one, so D is equivalent to D1.The four editor commands which take an iteration parameter all take the same default value.

U ... UP, pointer moves up (parameter) lines.

P ... PRINT, displays(parameter) lines on the video display. The last line displayed is the new current line.

L ... LOCATE, will find an arbitrary target string in the source file.
Example .... LBIT 7,(HL)
Moves the pointer to the first line containing that string.The file is searched downwards from the current line.If the string is not located the pointer is at EOF.
There are no restrictions on the string content.

SOURCE EDIT COMMANDS

Z ... ZAP, erases (parameter) lines from the file,starting with current.

E ... ENTER, enters text continuously into the file.Upon entry the line number is displayed as a prompt.
Type in a line of text,terminating with the (ENTER) key. The next line number will then be displayed and so on.To exit from the command simply key a full stop '.' as the first character of the line.
You can enter text anywhere in the file.Text is inserted at the current line with the old current line, and all following lines, moving downwards.

N ... NEW, lets you replace an old line with a new one.This command is just a more convenient way of performing a ZAP/ENTER sequence.

NOTE Although line numbers are displayed with most commands you are never required to actually enter them yourself.They are there solely as a positional guide and are computed dynamically by ZEN rather than being stored in the file along with the text.

NOTE Should you exceed available memory the MEMORY FULL error message will be displayed.The file will be in a safe state.

DEBUGGER COMMANDS

G ... GOTO, transfers control to a user program.You will be prompted for
   a transfer address first and then a BREAKPOINT address.Both of
   these parameters have default values.
   If you supply the default parameter for a transfer address then
   execution commences at the existing User Program Counter.
   The default response to the breakpoint request means that no
   breakpoint is set.
   If you supply a breakpoint address then the byte at that address is
   saved and a RST 38H instruction is inserted there.When the breakpoint
   is encountered control is transferred back to ZEN via the REENTRY
   vector to the TRAP handler where the original code is replaced.
   The entire Z80 machine state is saved at TRAP time and restored to
   the machine at GOTO time.
   The REENTRY vector also allows you to terminate user programs even
   if you don't set a breakpoint.Ending your programs with CALL REENTRY
   will transfer you back to ZEN and save the machine state.ZEN won't
   replace the CALL as it is able to distinguish betwwen this and a
   genuine breakpoint (A CALL is needed rather than a JP in order to
   leave the program counter on the stack).

C ... COPY, moves a block of memory from START ADDRESS to STOP ADDRESS
   inclusive to a DESTINATION ADDRESS.

F ... FILL, will fill a block of memory from START ADDRESS to STOP ADDRESS
   inclusive with a DATA constant.

X ... XAMINE, will display the Z80 user machine state.All registers are
   displayed and labelled.The main registers are on the top line and
   the alternate registers on the second line.

M ... MODIFY, lets you examine and modify memory contents.If you supply an
   address parameter then display commences at that address.MODIFY takes
   a default of the CURRENT OBJECT POINTER which is an object pointer
   similar to the source current line pointer.
   The address and the byte at that address will be displayed in hex
   followed by a prompt for a parameter.The default response will cause
   a step onto the next address.Entering an actual parameter causes
   the byte to be replaced with your parameter.Your parameter is erased
   on the video display,the new value id displayed and ZEN steps to
   the next address.
   To exit simply key a full stop '.' as your parameter.
   The current object pointer is left pointing at the last byte displayed

Q ... QUERY, displays a block of memory in hex and as literal ASCII.This
   command also takes the current object pointer as the default parameter
   Memory is displayed as eight lines of eight bytes,each line comprises
   four fields.
   1 ... Address
   2 ... Eight bytes in hex
   3 ... Eight bytes literal with Bit 7 stripped
   4 ... Eight bytes literal with Bit 7 untouched

   Bit 7 is stripped in field three to show up text that has this bit
   set high.Field four displays these characters as graphics.
   Control characters (less than 20H and greater than OBFH) are converted
   to full stops to prevent the display reacting.
   The current object pointer is left pointing at one greater than the
   last byte displayed.

THE ASSEMBLER

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:

MESSAGE:LD HL,GREETING;Say hello

Label
........Operator
...........Operand(s)
....................Comment

We say conceptually divided because the components of a statement don't have to be 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,BIT,etc.).In addition there are the PSEUDO-OPS which are detailed later.

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

NOP ..... No operands
CP ...... One operand
BIT ..... Two operands
JR ...... One or two .... JR SYMBOL
                                  JR NZ,SYMBOL

Operands may be:

Register names (A,B,DE,IX,etc.)
Condition codes (Z,NZ,C,M,etc.)
Numbers

The number group is the most complex.All the following are accepted as numbers:

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

Numbers
Decimal,hex and octal bases are accepted with decimal the default.Hex numbers are 'H' postfixed and octal numbers are 'O' postfixed.Numbers must begin with a digit,a leading zero is sufficient.

Symbols
These are explained in detail later on.

Program Counter
This is the internal variable which simulates the run-time PC.It's accessed by using $ (dollar).

THE ASSEMBLER

OPERANDS .... In addition all of the preceding data types may be
elements of an expression formed using the infix math
operators:

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

An expression can be used anywhere that a simple number
can be used.The following are all valid:

LD DE,START*2-782
LD A,'P'.80H
JR NC,$-5

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

LABELS ...... A label is a way of marking a statement.Each time you
use an operator like 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.
BASIC is an example of a language without this facility,the
line numbers act as labels.
Symbols will be explained in greater detail.

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 placing it at the start of a statement.The assembler assigns the
value of the program counter to it.

2 ... By using the EQU pseudo-op.This allows you to assign your own value
to a symbol.
Example .... BACKSPACE:EQU 8

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
numbers after that.Letters may be upper or lower case.ZEN allows symbols
of any length although symbols longer than seven characters will affect
the listing.The symbol field width is easily changed.

There are certain reserved keywords which cannot be used as symbols.
These are:

Operator names,register names and condition code names.

Note that all keywords are uppercase,using the same name in lowercase
would be perfectly acceptable.

PSEUDO-OPS

These are additional operators which have no equivalent in the Z80
instruction set but are understood by the assembler. They are used
in the same way as the normal operators.

```
END    End assembly...(No operands)
DS     Define Storage.(One operand)
DW     Define Word....(One operand)
DB     Define Byte(s).(Multiple 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 bytes leaving a gap in the code.Commonly
      used to reserve space for a text buffer,stack,etc., where the
      object code doesn'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.

      Example .... BUFFER:DW VALUE

      Would make locations BUFFER equal to VALUE.

DB   Generates the value of the operand(s) in the object file.Takes as
      many operands as desired,seperated by commas.

      Example .... DB 6,93H,'T'.80H,NEWLINE

      Each operand may be an expression but obviously no expression can
      have a value greater than two hundred and fifty-five decimal.
      The program counter will be incremented after every operand as if
      each were on a seperate line.
      In addition to the usual data types any operand may be of the type
      ASCII literal string.

      Example .... MESSAGE:DB'STICKY FINGERS',NEWLINE

      Strings may be of any length but, unlike single character ASCII
      literals, may not form part of an expression.A string is formed
      in the same way as a single character literal, by enclosing in
      matching quotes.Note that single and double quotes are implied
      seperators like the infix math operators.
      You may use a quote,of either type, as a literal by using the OPPOSITE
      type of quote as the delimiters.

PSEUDO-OPS(cont.)

EQU   Assigns a value to a symbol.

        Example ...... NEWLINE:EQU 13

        The operand may,as usual,be an expression but there is a restriction
on the symbols you may use in the expression.This is because the
operand must be capable of immediate resolution.
The value of any symbols used in the expression must already be
known to the assembler,forward referenced symbols will result in
the UNDEF error flag.

        Example .... NEWLINE:EQU BACKSPACE+5
                        BACKSPACE:EQU NEWLINE-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 making
such a mistake inadvertently.
In practise you will probably never encounter such a situation as
most EQUATES have simple operands.

ORG   Defines the origin of the object file.This operator may be used as
often as desired throughout an assembly to produce sections of
code at different locations.
The operand must conform to the 'no forward reference' rule for
obvious reasons.

LOAD   Lets you load the object code into memory, at any address,as it is
produced.
The loading process is entirely independant of the output option
specified upon entry to the assembler.If the operator is not used
then no memory location outside ZEN will be altered.
Note that use of a subsequent ORG operator turns the loading
process off, each time you set a new origin you must specifically
re-establish the load process.

ASSEMBLER ERROR HANDLING

If the assembler finds an error in the source code the following will happen:

(1) Assembly terminates
(2) An ERROR message is displayed
(3) The incorrect line becomes the editor current line
(4) The line is displayed
(5) The command loop is re-entered

You can now correct the error and assemble again.

ERROR MESSAGES

DOUBLE SYMBOL ... You have declared the same symbol more than once.
UNDEF .... You have used an undefined symbol.
RSVD .... You have used a reserved keyword for a symbol.
SYMBOL .... An obligatory symbol is missing (eg with EQU).
FULL .... The symbol table is full (see later).
EOF .... You have forgotten END and have hit EOF.
ORG .... No origin specified.
HUH? .... The line doesn't make sense.
OPND .... Something wrong with an operand.......

               Examples .... LD A,256
                                  BIT 9,B
                                  LD(DE),C

               Also any attempts to index out of range or to jump relative out of range.

The assembler will catch all incorrect statements. The only incorrect statement which will damage ZEN is a LOAD over the code.

ASSEMBLY OUTPUT

LIST ..... ZEN supports two List devices,the video display and an EXTERNAL
device such as a printer.You must insert your own external
driver,currently the driver just returns.
ZEN generates an ASCII CR,LF at newline time and an ASCII Form
feed at new page time.No other control characters are used.

Listings are generated a page at a time with a pause at the en
of each page.The cursor will blink during this pause.
Pressing any key during the pause will hold listing until you
press another key to restart.
Pages are fifteen lines long for VIDEO options and sixty lines
long for EXTERNAL options.

OBJECT ... ZEN supports one object device,the cassette.Object output is
in LEVEL II format but you will notice that records are only
thirty-two bytes long and that there will be a short inter-
record gap.

EXTENDING THE SYMBOL TABLE

The ST is the area where each symbol is stored together with it's sixteen
bit value.
The ST is positioned between ZEN and the source file.The SOF is used as
the end of table boundary.
As supplied the ST is six hundred bytes long.This value can be decreased
or increased by moving the source file pointers down or up. Procedure:

KILL the file
Locate the SOF and EOF pointers
Change BOTH pointers to the new file position
Save the new version on cassette

SAVING ZEN

You can alter ZEN and then save the new version by using the Write Objec
(WO) command.You should be aware of two things:

(1) The source file must be KILLED.
(2) You must save up to the VERY last byte shown in the listing.