

WYE VALLEY SOFTWARE Compiler

This Compiler will convert your BASIC program into a machine code-program. Machine code is literally speaking "the native tongue" of the ZX Spectrum, thus enabling your program to run much faster. The increase in speed can be up to several hundred times faster than the normal BASIC program.

This Compiler comes on a cassette in two versions, one for the 16K Spectrum and one for the 48K Spectrum. From now on in this manual the addresses for the 48K version will be given in brackets { } next to the corresponding address for the 16K.

2

USE

Load the Compiler into the computer using LOAD " " Type in your BASIC program (or load off tape) and run it to check for errors, when all is correct type RAND USR 27000 {60000} and press enter to compile your program Now you will see some messages appearing on the screen

These messages have the following meanings.

1. "START ADDRESS"

This shows the start address of the machine code program being produced (this is always one byte above RAMTOP).

2. "END ADDRESS"

This is updated during compilation and shows the last address of the machine code.

3 "ARRAY END ADDRESS"

This message only occurs when using the ARRAY in your program. The array is stored directly after the machine code, this message shows after completion the last address of the array.

4. "FIRST PASS" "SECOND PASS"

To make a complete compilation the Compiler must code your program twice. These messages show which pass of your program it is compiling.

5 "ERROR" "NO ERRORS"

If at end of compilation you get the message "NO ERRORS," then you have a correct code program. If at any stage you have "ERROR" appearing on the screen, the compiler has found a fault in your program.

The Compiler will print out the offending line and put a question mark (?) after the error, it also changes the "current line" to this error line, so just press shift + 1 to edit.

RUNNING COMPILED PROGRAMS

Make a note while compiling of the START ADDRESS then type RAND USR (START ADDRESS)

WHERE DOES THE MACHINE CODE GO

You can set when you want the code to be compiled by using CLEAR N (where N is one less than start address required.)
On loading the Compiler RAMTOP is set to 26,000 {40,000}

By this method you can compile several programs, by locating each one at different areas of memory.

MEMORY

MEMORY can be exhausted in several ways.

1. During compilation a table of addresses of LINE numbers is built up. "OUT OF MEMORY" will occur if too many line numbers are used.

2. If either the machine code or the ARRAY threatens to over-write the compiler then "OUT OF MEMORY" will occur.

VARIABLES AND NUMBERS

There are 52 Variables allowed.

26 Upper Case letters.

26 Lower Case letters.

You are also allowed to dimension a single array A (). This can be up to 255 elements long.

Numbers are stored as integers only, in the range -32767 to 32767

Certain functions will take the number to be stored as 0 to 65536. e.g. PEEK, POKE, USR.

All calculations can be written as normal but must be enclosed in brackets ().

N.B
 Before using USER defined graphics you
 must reset the graphics pointer.
 We suggest the following.
 POKE 23675, 88
 AND ALSO
 FOR the 48K POKE 23676, 255
 FOR the 16K POKE 23676, 127

4

COMMANDS WHICH COMPUTER CAN HANDLE

The compiler can handle a large range of normal BASIC commands, there are also some very advanced commands that the compiler can handle but the normal Spectrum BASIC can't.

Here is a list of the commands allowed

KEY

a	represents a single letter variable
c	represents a sequence of colour items, separated by ";" or ","
\$	represents a sequence of statements, separated by colons " "
x,y,z	represents numerical expressions.
e	represents an expression.
n	represents a number.
BEEP x,y	This statement has been changed from normal BASIC to make it more versatile, X, Y, can be any integral numbers. X gives the length (it is really the number of square wave cycles output and has to be calculated for each frequency) Y is the delay between each square wave cycle output, so decreasing Y increases the frequency This may seem complicated but allows extremely complex sounds to be generated.
BORDER e	As normal
BRIGHT e	As normal
CIRCLE c;x,y,z	As normal
CLEAR	Sets all variables to zero, but leaves array intact.
CLS	As normal
COPY	As normal
DATA e1,e2,e3	As normal
DIM A(n)	Dimension an array with name A()
DRAW c;x,y	As normal
FLASH e	As normal

For a = x to v	As normal
For a = x to y STEP z	As normal
GOSUB n	As normal
Go to n	As normal
If x then s	As normal (),-,<are allowed)
INK e	As normal
INPUT	As normal (this is exactly the same as normal Spectrum BASIC since any PRINT items can be included and also a number of variables may be input. Normally the characters typed appear in the bottom section of the screen, but if desired by changing the contents of a memory location before compiling the input printing can be placed in the top section of the screen. The address of memory location is 27930, (60930). Its contents are 0 for normal input and 2 for special input.
INVERSE e	As normal
LET a = e	As normal
LOAD f CODE	As normal
L PRINT	As normal
NEXT a	As normal
OUT x,y	As normal
OVER e	As normal
PAPER e	As normal
PAUSE e	As normal
PLOT c;m,n.	As normal
POKE m,n	As normal
PRINT	As normal
RANDOMIZE	As normal
READ a	As normal

6

REM	This statement can be followed by a single letter and some parameters, each letter is a command which cannot be normally obtained in Spectrum BASIC. These are listed at the end. if the first character is not one of those key letters, then it is treated as a normal REM statement.
RESTORE n	As normal
RETURN	As normal
SAVE f CODE m,n	As normal
STOP	Causes a return to BASIC with STOP message.

EXTRA COMMAMDS (AFTER REM STATEMENTS)

A,e	Set colour of attribute to e
B	Check to see if BREAK pressed.
Ere	Cause an error number e + 1
F	Change whole screen to current attribute with clearing the screen
L,e	Scroll line e left 1 pixel
R,e	Scroll line e right 1 pixel
N	Output random noise to speaker (explosion)
S, e,x,y	(Mini Sprite graphics). This plots character number e at plot co-ordinates x,y. By using two or three of these Statements very impressive moving graphics may be produced.

FUNCTIONS

ATTR

CODE "Character" CODE INKEY\$ CODE SCREEN \$

IN Number

PEEK

POINT

RND

USR e + USR"CHARACTOR"

SAVING THE CODE

For the compiler to produce efficient machine code, there must be a set of RUN-TIME routines, which are called from the code. The machine code program cannot function without these routines and so must be saved along with the machine code.

These routines are copyright and saved copies must not be lent or sold without the written permission of "Wye Valley Software". We are always ready to receive new games or programmes and if suitable will buy outright or sell on a royalty basis.

The program can be saved as follows:

For the 48K Spectrum use SAVE "name" CODE start address,
65536 - start address.

For the 16K Spectrum use SAVE "name" CODE start address, 32767 -
start address.

8

EXAMPLES

```
10 FOR a - 40 TO 200 STEP 5
20 FOR b - 50 TO 600 STEP a
30 BEEP 50, b
40 NEXT b
50 NEXT a
```

```
10 FOR A = 1 TO 100
20 REM N
30 REM B
40 NEXT A
```

```
10 FOR a = 1 TO 128
15 LET b - (255-a)
20 REM S, 87, a, a
21 REM S, 88, b, a
30 REM B
31 BEEP 5,B
40 NEXT a
45 BORDER 0: PAPER 0: INK 7: REM F
50 REM N
55 BORDER 7: PAPER 7: INK 0: CLS
60 REM N
```