

FOR W = U TO V -W Increments in steps of 1 from U to V. NEXT W ends the loop. Note that (V-U) must be less than 32767. The loop is always traversed at least once.

GOSUB N Calls line N as a subroutine. If line N does not exist then it goes to the next line after N. Note that N must be a positive integer constant.

GOTO N Jumps unconditionally to line N. Otherwise as for GOSUB.

IF V op U THEN - where op is any of AND, OR, <>, =, <, >, <or>, <or >. Note that V and U must not differ by more than 32767. For string comparisons AND and OR are not applicable.

IN
INK
INKEY\$

INPUT A, A(m) or A\$ As BASIC. For numbers a leading sign is allowed. For strings the default length is 32 characters. Characters with a code value less than 32 and greater than 164 are not allowed.

INT Included to facilitate testing under BASIC.

INVERSE
LEN A\$ As BASIC except that A\$ cannot be sliced.

LET
LPRINT
NEW
NEXT W See for - NEXT
OR Boolean only allowed in IF statement.

OUT
OVER
PAPER
PAUSE
PEEK
PLOT
POKE
POINT

PRINT
RAND
READ Reads a list of integers or strings from a data statement. Only minimal checking is done to ensure correct data types.

REM
RESTORE N Restores the DATA pointer to line N. N must exist.

RETURN Returns from a subroutine started by a GOSUB. Make sure that your GOSUBS and RETURNS match as no check is made.

RND Returns a random integer between 0 and 32767. (NOT the same as BASIC). To obtain the same effect under BASIC use USR 59997.

SGN As BASIC
SQR Integer square root.
STOP If MCODER finds a STOP statement compilation ceases there. If you wish there to be a STOP in the middle of your program then the command LET L=USR 7406 will give an error 9 and stop.

STRINGS By default strings have a maximum length of 32 characters. If you exceed the maximum string length then you will write into whatever follows (either another string or an array). To change the default length you should POKE location 60200 with a value between 1 and 255. This should be done from within your program before the first use of the variable which is to have a different length. There are no string arrays. A\$(TO m) and A\$(nTO) not permitted. The general form A\$(m TO n) or A\$(m) must be used.

USR m Calls a user defined machine code routine.

USR "string"

EXAMPLE PROGRAM

Entering this simple program will give you an indication of the speed improvement possible with MCODER.

```
10 LET P = 0 : INK P
12 FOR F = 1 to 20
13 BEEP 1/ (2*F), F
14 NEXT F
20 LET A=0 : LET B = 255 : LET C = 175: LET
  D = 0: CLS
30 FOR I = 0 TO 255: PLOT OVER 1; I,A:
  NEXT I
40 FOR I = 0 TO 175: PLOT OVER 1; B, I:
  NEXT I
50 FOR I = 0 TO 255: PLOT OVER 1; (255 -I),
  C: NEXT I
60 FOR I = 0 TO 175: PLOT OVER 1;D,
  (175 -I): NEXT I
70 LET A = A+1: LET B = B-1: LET C = C-1:
  LET D = D+1
80 LET P = P+1: IF P = 7 THEN LET P=0
81 PRINT AT 10,12; "SPECTRUM"
82 PRINT AT 11,13; "MCODER"
85 INK P
90 IF A>176 THEN GO TO 10
100 GOTO 30
9000 STOP
```

How do I save the compiled Code?

This is done by using the normal save routine of the SPECTRUM ie. SAVE "name" CODE n where n is the start address of the compiled code. Notice, however, that MCODER must also be saved as it contains several runtime routines which the compiled code needs to run properly. The compiler stretches from 59990 to 65368 thus the full save command should be SAVE "name" CODE 40000, 25368.

MCODER II

SPECTRUM

© P.S.S. 1983

452, Stoney Stanton Road,
Coventry
CV6 5DG.

Telephone: (0203) 667556

MCODER (SPECTRUM)

MCODER is the first true compiler for use with the 48K SPECTRUM. It runs integer BASIC using numbers in the range -32768 to 32767 to allow considerable increase in the running speed of compiled programs.

Although MCODER is a sophisticated and versatile tool; do not expect miracles. You will have to become familiar with its method of operation if you are to use it to its best advantage.

How do I load MCODER?

MCODER must always be present in the computer before you enter any code. It is loaded using LOAD " ".

How do I use MCODER?

Using MCODER is really a very simple operation. After loading MCODER, type in your BASIC program. It is essential to run and check the program before compiling it, however we strongly suggest that you first read through the list of commands which MCODER will compile ensuring that your BASIC program consists only of these commands.

Once you are satisfied with the BASIC program you can compile it using the command RAND USR 60000 and pressing ENTER. The BASIC will then scroll up the screen. If MCODER comes across a statement it cannot compile then it will return to BASIC with an inverse S at or near the offending command.

Possible errors include:

- 1) Using illegal variable names.
- 2) Goto 'variable' or Gosub 'variable' which are not allowed.
- 3) Illegal statement type such as SAVE or DIM AS (10)

If an error is reported then you should correct it and recompile.

At the end of a successful compilation you will be shown 3 pieces of information:

- 1) A report saying that your progress has compiled OK.
- 2) The length of the compiled code.
- 3) The run address.

Can I compile basic programs I already have on tape?

YES — first load MCODER and then load the program you wish to compile. You should be aware that you will probably have to make considerable alterations to your program before it will successfully compile as it is likely that the BASIC was not written to suit MCODER.

Can I compile more than one program?

YES — it is possible to 'stack' several programs on top of each other. This is done by first moving RAMTOP using CLEAR n where n is a number between 24000 and 59000. If RAMTOP is initially set high then a routine may be entered, compiled and stored above RAMTOP. If RAMTOP is then lowered using CLEAR n then another routine can be inserted between the new RAMTOP and the bottom of the first routine. Care must be taken to lower RAMTOP sufficiently otherwise the first routine will be partly overwritten. These routines can then be called individually as required.

HOW DO I RUN COMPILED PROGRAMS

The compiled code is always stored immediately above RAMTOP with the exact call address being display after compilation.

TRACE FACILITIES

Three levels of trace and protection are available.

- o) BREAK is disabled except at 'scroll?' and INPUT. This results in the fastest and smallest code.
- 1) BREAK enabled. This results in longer code than

does o) and at work execution may only be half as fast.

- 2) BREAK enabled puts the line presently being executed displayed in the top right hand corner. This is clearly a powerful feature in debugging a program, its disadvantage is that if the code runs at about the same speed as BASIC.

By default option 1 is chosen but you may change this as many times as you like during a program by a REM statement which starts REM#0, REM#1 or REM#2 corresponding to the three modes above.

Access to MCODER variables

The address of the variables used by MCODER can be listed with a routine starting at location 65240.

To use this routine enter:—

PRINT USR 65240; "variable name".

For arrays and strings the name can be shortened to the first two characters ie. A\$ or B(

Two values are returned for arrays and strings, the first is the address of the pointer to the array or string, the second is the address of the first item in the string or array.

NOTE Arrays in MCODER contain an element zero and the array pointer will point to the address of this element. This address printed has been adjusted and points to element one of the array.

Arrays and strings are allocated at run time and if this routine is called before the program has run then only the address of the pointer will be correct.

WHICH COMMANDS WILL MCODER COMPILE

Variable names may consist of upper and lower case letters and numbers. Arbitrary expressions may be used anywhere except in LET statement where they must be enclosed in bracket.

AND Boolean AND. Allowed only in an IF statement.

ABS
ATTR
BEEP x,y

Do not use brackets if a BEEP of less than 1 sec is required. e.g. use BEEP 1/3.5. NOT BEEP (y/s),5

BIN
BORDER
BRIGHT
CIRCLE x,y,z
CHR\$
CLS
CLEAR
CODE
COPY
DATA list

Clear compiled code variables only.

List must be a list of integers or strings in quotes and may not contain expressions.

DIM A(V)

Only one dimensional arrays are available in MCODER and there must be at least 2*V bytes spare space at run-time. No runtime array bound checking is done so make sure it works under normal BASIC. If you redefine an array a new version of it is made but the old one is not deleted. This means that repeated allocation can eventually fill the machine and give an error (out of memory) either when allocating an array or a string. Array and string space stretches from 100 bytes above STKEND to 256 bytes below RAMTOP. All arrays, strings and variables are erased when you re-enter an MCODER program and all the space is available again. There are no string arrays.

DRAW x,y
DRAW x,y,z
FLASH