

## INSTRUCTIONS

## USING THE COMPILER

SUPA CODER is an integer compiler and can only use numbers in the range -32768 to 32767. Unless otherwise stated the Basic statements are standard MTX Basic. The compiler is invoked by the Basic commands USER or RAND USR(42000). If any Basic is found which cannot be compiled then a cursor will show where the problems lies and a return to Basic will be made. It is possible to run out of space, "NO MORE MEMORY", at several points in the compiler. The normal area reserved for variable names is about 200 bytes long and too many variables will cause this to be exceeded. If this is so the error will occur on the right hand side of a LET, in a READ or INPUT statement. The position of the bottom of the table is stored in locations 41948 and 41949 and may be poked to increase the space available (make the two byte number smaller). If the start address of the compiled code is too large then the compiler will give a "NO MORE MEMORY" message, normally when compiling the the right hand side of an expression. Finally it is possible to run out of space after the code has compiled but before returning to Basic. It is during this time that space is allocated for the values of the variables.

The MTX does not allow you to save machine code unless it is contained within the Basic environment. There are however two ways that you can save compiled code. The first is from within your compiled program using the SAVE command. This will save the compiled code and the compiler. The other way is to use the save/load program. With this program you can select the start address of the code you wish to save or load, please note however that compiled code must be reloaded to the address that it was compiled at. When you are ready to save your code, load the save/load program. This program MUST be used to reload compiled code.

## AVAILABLE COMMANDS

ABS

ADJSPR

AND Boolean AND allowed only in an IF statement.

ANGLE <angle> Set to zero on entry to SUPA CODER. See NOTE 1.

ARC <length>,<angle> See NOTE 1.

ASC

ATTR

CHR\$

CIRCLE

CLEAR

Clears compiled code variables only.

CLOCK

CLS

COLOUR

CRVS

CSR

CTLSPR

DATA

DIM A\$(n)

Only one dimensional arrays are available in SUPA CODER. The default string length is 64 but this can be changed by the use of the DIM statement and this must be the first reference to that variable.

A(n)

There must be at least  $2*n$  bytes spare space at run-time. No run-time 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 "NO MORE MEMORY" either when allocating an array or a string. Array and string space stretches from 100 bytes above Basic to 256 bytes below the start of the compiled code. All arrays, strings and variables are erased when you reenter a SUPA CODER program and all the space is available again.

DRAW

DSI

ELSE

FOR L = M TO N L increments in steps of 1 from M to N. Note that (N-M) must be less than 32767. The loop is always traversed at least once. STEP has not been included as this would slow down execution time for the loop. If an increment other than one is required then code the loop as follows.

```
FOR M = 1 TO N
-----
-----
-----
-----
LET M = M + (stepsize-1)
NEXT M
```

GENPAT

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

GOTO <lineno> Jumps unconditionally to <lineno>. Otherwise as for GOSUB.

GR\$

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

INK

INP (n) INPUT to port n. Not in the manual but it works from Basic.

INPUT For strings the maximum input length is 64 characters.

INT Included to facilitate testing under Basic.

LEFT\$ See NOTE 3.

LEN

LET

LPRINT

MID\$ See NOTE 3.

MOD

MVSPR

NEW

NEXT I The loop variable must be used.

ON GOTO

ON GOSUB

OR	Boolean OR allowed only in an IF statement.
OUT	
PAPER	
PAUSE	Maximum value 32767.
PEEK	
PHI <angle>	See NOTE 1.
PLOD	When PLOD is run from compiled code some of its links with the Operating System are lost. As a result some combination of PLOD commands may not work when compiled. NODDY pages are not compiled so the contents may be changed without the need to recompile the program.
PLOT	
POKE	
PRINT	
RAND	
READ	
REM	See NOTE 2. REM#0 Disables the BREAK key. REM#1 Enables the BREAK key. (Default) REM#2 Basic line number trace.
RESTORE <lineno>	Restores the DATA pointer to <lineno>. The linenummer must exist.
RETURN	Returns from a subroutine started by a GOSUB. Make sure that your GOSUBS and RETURNS match as no check is made.
RIGHT\$	See NOTE 3.
RND	Returns a value between 0 and 32767. (NOT the same as Basic). To obtain the same effect under MTX Basic use USR (41997).
SAVE	This will save the compiled code and the compiler, to reload the compiled code the "SAVE/LOAD" program MUST be used. NODDY pages and System variables are not saved. NODDY pages may be saved into a separate file by the SAVE command from the keyboard.
SBUF	
SGN	
SOUND	
SPK\$	Works only on a TEXT screen.
SPRITE	
SQR	Integer square root.
STOP	
THEN	

TIME\$  
TO  
USR  
VIEW  
VS n

#### NOTE 1

Where <angle> is given as an argument then floating point evaluation is possible. The value for <angle> may be in the form N/M, and N/M is evaluated to give a floating point result. N and M may be integers, integer variables or integer bracketed expressions. e.g. ANGLE 3/4 sets angle to 0.75

PHI (1+2)/(2\*4) sets phi to 0.375

A division sign is the only operator allowed outside of brackets.

#### NOTE 2

There are three levels of trace and protection available and are called with a REM#n command, where n has a value between 0 and 2. Note that there must not be a space between the REM and the # character otherwise it will be treated as a comment.

- 0) The BREAK key is disabled. This results in the fastest and smallest code.
- 1) The BREAK key is enabled. This is the default value on entry to SUPA CODER.
- 2) The BREAK key is enabled and the Basic line number presently being executed is displayed in the top left hand corner of the current virtual screen. The program may be slowed down by pressing the 'S' key.

The options may be changed as many times as you like during the program.

#### NOTE 3

By default strings have a maximum length of 64 characters, but this can be changed by the use of the DIM statement. If you exceed the maximum string length then you will write into whatever follows (either another string or an array). There is an alternative method of string slicing using the format A\$(n) or A\$(n,m) where n is the start position and m is the length of the substring. Using this form strings may be sliced on the left or right hand side of an expression.

MEMORY MAP

2768 37000 (MTX500)

3284 NBTOP+100 30000 (MTX512)

42000

!-----!-----!-----!-----!  
BASIC \* STRINGS+ARRAYS\* \* COMPILED CODE \* VARIABLES\* \* COMPILER

This file was downloaded from

***www.primrosebank.net***

## **Abridged Terms & Conditions (Downloads)**

### **Disclaimer**

[www.primrosebank.net](http://www.primrosebank.net), (*the website*) is provided by Dave Stevenson as a service to the public, is provided "as is" and carries no warranties, expressed or implied, of any kind.

*Dave Stevenson is not responsible for, and expressly disclaims all liability for, damages of any kind arising out of use, reference to, or reliance on any information contained within the website or made available for download.* Whilst the information contained within the website site is periodically updated, no guarantee is given that the information provided on the website is correct, complete, and up-to-date.

A number of articles on the website contain technical data and practical guidance which may be of use in testing and maintaining various items of vintage computer and electronics hardware. Such articles are not intended to cover all aspects of the tasks involved and may omit essential information, including necessary safety precautions. Performance of the tasks described may risk damage to equipment and/or people. The reader is responsible for ensuring that he/she is capable of performing the tasks described and well as assessing the inherent risks involved and taking appropriate measures to mitigate such risks.

*Dave Stevenson expressly disclaims all liability for, damages to equipment or injury of any kind arising out of use of such technical data and guidance.*

Unless otherwise noted, all data on the website is deemed to be *Copyright (c) Dave Stevenson, 2009-2013*

You are hereby granted permission to download data and software from the website *for your own personal use. Redistribution of any content from the website without written authorisation from Dave Stevenson is expressly forbidden. You are also expressly forbidden from offering for sale any material obtained from the website.*

As far as possible, information included on the website from other sources has been credited to the respective author and/or publisher. The majority of content on the website is derived from material first published in the 1980s. *This material is likely still under copyright of the original author and/or publishers.* The authors and/or publishers may not have given express permission to copy, transmit or make this information available for download, but I believe that they would have no objection to this archive information being placed into the public domain.

However, should the author and/or publisher of the original material find any content on the website for which they wish to assert their rights, they should notify Dave Stevenson (by e-mail to: [webmaster@primrosebank.net](mailto:webmaster@primrosebank.net)) who would be pleased to enter into a dialogue to agree a satisfactory resolution of their concerns.

If you obtained this file as part of a paid-for package, you have been scammed! I suggest that you request a refund from the seller, please also advise Dave Stevenson at the e-mail address above.