

# USER BASIC Toolkit

---

TAKE THE CHAINS OFF MTX BASIC

OVER 20 NEW COMMANDS.

UNLEASH THE POWER OF YOUR MEMOTECH



INSTRUCTIONS

## SYNTAXSOFT

# Syntaxsoft

---

USER BASIC

WRITTEN BY ERIC ROY



PRODUCED BY SYNTAXsoft

UNIT 109 GLENFIELD PARK, GLENFIELD ROAD, NELSON BB9 8AB

Copyright Syntaxsoft & Eric Roy 1986

# Syntaxsoft

## USER BASIC INTRO

As the name says USER BASIC uses the USER command to provide an extra 20 plus commands to the MTX BASIC. Have a pen and paper handy when you read the following pages, and note the syntax required for each new command.

There are three versions of the USER BASIC utility program. UB\_TAPE which loads into high memory and is saved to tape with the current program, and uses no user ram. The other two are the SDX disc versions for the MTX 512 & MTX 500 because the SDX requires a large amount of ram in high memory the code has been placed in CODE line 0 of the basic program area and uses just over 3000 bytes of user ram.

One problem I found with the USER command was that it could not be used in an IF - THEN - ELSE line without giving an error.

To overcome this all errors are now tested by USER BASIC to see if it was the USER command that caused the error.

If the error was caused by USER then the BELL will sound and the line will be inserted into the program, without any further checking by the computer. So be very careful when entering the USER command in IF - THEN - ELSE lines.

Pressing function key <F8> will insert the command USER. In IF - THEN - ELSE you will have to type it out in full.

### Syntax... USER BASIC

This command must be executed at the start of any program that makes use of the other commands in USER BASIC.

It sets up the stack pointers & scans the program looking for user @LINENAME (see USER CALL\_ for details). It also creates two variables TRUE and FALSE.

TRUE = -1 & FALSE = 0

All arrays must be dimensioned before the USER BASIC command, e.g.

```
10 DIM A (40),N(22),DS(2,40)
20 USER BASIC
30 Rest of program
```

### Syntax... USER CALL\_LINENAME

This command allows you to call subroutines by name rather than line number.

The USER BASIC command searches the program looking for USER @NAME and converts NAME into a variable with a value equal to the following line number.

USER CALL\_ gets the value of NAME and goes to that line number.

Example....

```
50 USER CALL_SUBROUTINE
60 Rest of program
```

```
360 USER @SUBROUTINE
```

# Syntaxsoft

---

390 Subroutine starts here  
440 RETURN : Returns to line 60

USER CALL\_ cont.

USER @NAME must be the only command on a line, as it is just used as a pointer to the start of a section of basic.

USER\_CALL uses the gosub stack to store its return addresses, so use the CALL\_ command in the same way you would use GOSUB. Error messages are also the same as GOSUB errors.

Note.. Avoid using variables with the same name as subroutine names.

The USER BASIC command will generate a No Line error if there is no line after USER @NAME or, on the MTX 512 if USER @NAME is the last line on memory page 0 and the next line is on page 1.

To cure insert/delete a REM line until both USER @NAME and the following line are both on the same page.

Syntax... USER JUMP\_ LINENAME & USER RESTORE LINENAME

These commands operate in the same way as USER CALL\_.

USER JUMP\_ allows you to GOTO a line following USER @NAME.

USER RESTORE will restore the first line of DATA following USER @NAME.

NOTE... there is no underline character between RESTORE and LINENAME.

Syntax... USER ON Var,

This command is similar to the MTX ON command, except that it is used with USER CALL\_, JUMP\_ and RESTORE.

Example.....VAR = 0    1    2  
                  :    :    :  
USER ON VAR,CALL\_ SUB0,SUB1,SUB2, etc..

USER ON VAR,JUMP\_ LINO,LIN1,LIN2, etc..

USER ON VAR, RESTORE 0, DAT0, DAT1, DAT2

NOTE...comma after VAR and 0, (zero comma) after RESTORE. No comma between CALL\_ or JUMP\_ and first linename.

Syntax...USER REPEAT and USER UNTIL\_expression=TRUE

These commands will repeat the section of basic between USER REPEAT and USER UNTIL\_ until the expression after USER UNTIL\_ is found to be true.

Example.....

```
10 LET COUNT=0
20 USER REPEAT
30 PRINT COUNT
40 LET COUNT=COUNT+1
```

# Syntaxsoft

---

```
50 USER UNTIL_COUNT=45
60 Rest of program
```

In the above example lines 30 and 40 will be repeated until COUNT=45. Only when COUNT is equal to 45 will control pass to line 60 and the rest of the program.

```
USER REPEAT, UNTIL_cont'
```

String expressions and command words can also be used after UNTIL\_.

Example....

```
USER UNTIL_REPLY$="YES"
USER UNTIL_INKEY$="Y"
```

The FALSE variable created by USER BASIC can be used to ensure that REPEAT UNTIL\_ loops repeat forever.

```
50 USER REPEAT
   Lines to be repeated forever.
90 USER UNTIL_FALSE
```

REPEAT UNTIL\_ loops may be nested to a depth of 20. Overflow error if repeat stack is full.

Syntax... USER WHILE\_expression = TRUE  
USER ENDW

These commands operate in a similar way to the REPEAT / UNTIL commands except that the expression is tested at the start. If the expression is TRUE then basic lines between USER WHILE\_ & USER ENDW will be executed. If the expression is FALSE then the lines between WHILE\_ & ENDW will be skipped and the program will start from the line following ENDW.

Both these commands must be the only command on a basic line.

WHILE\_ ENDW loops may be nested to a depth of 10. Overflow if stack is full. No Line error if it finds ENDW without a WHILE\_ or visa versa.

Syntax... USER %VARIABLE =

This command allows HEX, BINARY or logical AND, OR, XOR values to be assigned to variable following %. A 2 byte peek function is also provided.

Examples....

```
USER %BYT=PEEK #FA92      : DOUBLE PEEK

USER %HEX=#FF             : # = HEX
USER %BIN=&01110111       : & = BINARY

USER %LOA=[A]#FAA2,#0E    : [A] = AND
USER %LOO=[0]#15A,&10101  : [0] = OR
USER %LOX=[X]&11101,124   : [X] = XOR
```

No arithmetic expressions allowed,

# Syntaxsoft

---

USER %ADD=#FA + #56 is ILLEGAL

Syntax...USER PHEX\_ or USER PBIN\_

These commands will print the HEX or BINARY value following the command.

Example.....MAXIMUM VALUE 65535

```
USER PHEX_255 would print #FF
USER PHEX_&1111 " " #OF
USER PBIN_#FF " " &11111111
```

These commands do not print a CRLF after printing the number, so the next print position will follow the number.

Hex numbers will be printed as two or four digit numbers prefixed with #.

Binary numbers will be printed as eight or sixteen digits prefixed with &.

Syntax... USER INSTR\_SUB\$,MAIN\$

This command will find the first occurrence of one string in another, the position of the substring in the mainstring will be returned in the variable POS.

```
123456
Example...USER INSTR_"D" ,"ABCDEF" would return with POS = 4.
```

```
LET S$="MADAM IN ADAM"
LET F$="ADAM"
USER INSTR_F$,S$ : PRINT POS
```

Would print 2 as the ADAM in MADAM is the first occurrence of the string.

POS will equal 0 if the substring does not occur in the mainstring, or if the substring is longer than the mainstring, or if the strings are empty.

Syntax...USER SHAPE\_X,Y,N,"string"

This is the only graphics command in USER BASIC. It allows complex shape data to be held in a string and plotted to the screen starting from position X,Y. The N parameter determines how many pixels are to be plotted.

The string data should only contain the direction numbers 0 to 7 and the two square brackets [ ]. Any direction number inside the brackets will not plot points but the X and Y parameters will be increased by N pixels. So the [ bracket = PENUP ( dont plot) and the ] bracket = PENDOWN.

Full wrap round on both X and Y axis are provided.

Examples of this follow later.

USER SHAPE\_ cont'

```
DIRECTION      : To plot a square with a side length of 10.
NUMBERS        :
```

# Syntaxsoft

---

```

      6
5    : 7
 \   : /
  \  : /
4---+---0
  /   \
 /    \
3    : 1
      2

: LET N=10
: USER SHAPE_X,Y,N,"0642"
:
: To plot a square and then move
: 30 pixels to the right and
: plot a diamond. N = 10
:
: string = "0642[000]7531"
:
: To double the size of the shape multiply by N parameter
: by 2 and use the same data. If N = 0 then no points will be plotted.
[ = PENUP
] = PENDOWN
-----

```

String variables can be used. Error if the string contains any illegal data.

Syntax... USER INKEY\$

This command will stop a basic program and wait until a key has been pressed.

The ASCII value of the key pressed is returned in the variable KEY.

Example.... Wait until space is pressed.

```

50 USER REPEAT
60 USER INKEY$
70 USER UNTIL_KEY=32 or
70 USER UNTIL_KEY=ASC(" ")

```

SYNTAX...USER POKE Address, Value

This is an intelligent POKE in that if the value is less than 256 then only Address is loaded with Value.

If Value is greater than 256 and less than 65535 then Address is loaded with the low byte of Value and Address + 1 with the high byte of Value.

Any combinatin of Hex, Binary, Decimal or Variable can be used.

Example....

```

USER POKE #FA99,#4017
USEER POKE 64152,&11000011

```

Syntax.... USER LNUM\_200,500,10

This command will renumber basic line numbers only.

190 REM Before	190 REM After
200 REM LNUM	500 REM LNUM
210 REM command	520 REM command

# Syntaxsoft

USER LNUM\_ <RET> (no parameters) will renumber the lines from 100.

If your program contains GOTO, GOSUB or RESTORE then the line numbers after them will have to be changed using the EDIT command.

Why you should use GOSUB and GOTO when you have CALL\_ and JUMP\_ beats me.

Syntax...USER DELETE\_130,280  
or USER REM

USER DELETE\_ is used to delete blocks of basic from a program. In the above example lines 130 to 280 inclusive would be deleted.

Error if second parameter is smaller than the first, or on the SDX versions if you try to delete line 0.

USER REM will delete all basic lines that start with REM.

Note...Enter these commands as direct commands only.

Syntax...USER VAR or USER VARS

This is a debugging aid that will list all variable names and values (decimal only). Numeric and string array names are listed followed by () or \$() but the values are not listed.

USER VAR, VARS redefine VS 0 as a full text screen. The difference between the commands is that VARS will save any text on the screen and restore it on exit, the screen is stored from #F400 to #F7C0. (SDX versions #F000 - #F3C0). Use USER VAR if you do not want to corrupt this area.

If string variables contain control codes or non ASCII characters, then the decimal value of these codes are printed in brackets.

Syntax... USER FIND\_string to find

This command will search the basic program and list the line numbers that contain the string to be found.

Example.... To find the variable SIZE enter USER FIND\_SIZE <RET>.

This will also list any line that has the word SIZE in it. String variable and command words can also be found,

USER FIND\_ABC\$                      USER FIND\_PRINT

To find ASCII words then these should be enclosed in quotes,

USER FIND\_"This word"

Syntax...USER TSAVE\_Start, Length or USER TLOAD\_ " , "  
These commands will Save and Load blocks of code to tape.

Again Start and Length can be any combination of Hex, Binary etc.

No header information is saved so remember to note the address and length when saving.



# Syntaxsoft

---

Example.....USER TSAVE\_#8000,#F0

USER TLOAD\_#8000,240

Extra PANEL Commands KEY P and W

Both these commands are connected with dumping the PANEL display to a printer.

Pressing key 'W' will dump the whole screen to the printer.

Pressing key 'P' will dump the list screen only plus lines for comments.

```
0000      DI
0001      XOR A
0002      LD HL,#4000
0005      JP #0194
```

Example of key P output to printer.

The bottom line of the list screen is not sent to the printer, this is to prevent duplication of lines if more than one screen is printed.

## USER BASIC EXIT

Important note for MTX 512 owners with SOX disc fitted. Basic programs that use the USER BASIC utilities must not extend into memory page 1. A warning is displayed if it does.

Sorry about that, but it was the only way I could get the utilities to work with the SOX fitted.

If you get an error when entering the USER command, try the following,

```
TAPE version...LET ZZ=USR(59392) <RET>
SOX      "      RUN
```

To ensure that the utilities are saved to tape with the basic program enter...

TAPE only.....USER POKE #FA92,#E7FF

USER BASIC  
WRITTEN BY ERIC ROY  
PRODUCED BY SYNTAXsoft

UNIT 109 GLENFIELD PARK, GLENFIELD ROAD, NELSON 889 8AB

Copyright Syntaxsoft & Eric Roy 1986

