ACORNUSER

UserROM

Utility command ROM for the BBC micro, Master and Electron

UserROM

19 favourite
utilities
from Acorn User
in command form

for the BBC micro, Master
and Electron

compiled by Bruce Smith

WARNING

This product is supplied on an EPROM chip which can be damaged by static electrical discharge. Take care to 'earth' yourself by touching a large metal object before handling the chip, and do not work on surfaces which build up static charges, such as nylon carpets, or wear nylon clothing. The ideal surface is a dry metal kitchen sink. Always hold the chip by its ends, rather than by the metal legs.

This is one of a series of products tested and supported by Acorn User, the monthly magazine for users of Acorn computers, including the Master series, BBC micro and Electron. Look out for details of other products in Acorn User, or write to the address given below.

Issue 1 February 1986

While UserROM has been carefully tested, the publisher does not accept any liabilities with respect to the program.

UserROM and the UserROM manual were compiled and written by Bruce Smith. Edited by Tony Quinn.

Any correspondance about UserROM should be addressed to UserROM, Acorn User, Redwood Publishing, 141-143 Drury Lane, London WC2E 9JH.

INTRODUCTION

UserROM supplies you with 19 favourite programs from Acorn User on a chip. The advantages of this are that these programs or utilities are always instantly available for use either directly from the keyboard or from within your programs simply by typing in the appropriate command or commands. Also they do not take up any precious programming memory. The 19 commands are detailed in this manual.

COMPATIBILITY

UserROM will work with any version of Basic. However it does require that operating system version 1.0 (OS1.0) or later be fitted (earlier versions do not support ROM software). All commands will work with the BBC B and B+ micros. In the case of the latter the shadow screen may need to be disabled. Some commands by their nature will not work correctly across the Tube with a 6502 second processor fitted. Full details are given in the compatibility chart below (figure 1). Master users need not use the *EXPLODE or *CIRCLE commands. Electron users will find that all the commands will work apart from *BORDER and *SCROLL which require the teletext mode 7 screen.

Figure 1. UserROM Compatibility Chart

Command	BBC B	BBC B+	Master 128	Electron
*BORDER+	Yes	Yes*	Yes*	No
*CHECK	Yes	Yes	Yes	Yes
*CIRCLE	Yes	Yes	Yes	Yes
*COLFILL	Yes	Yes	Yes	Yes
*COMPRESS+	Yes	Yes*	Yes*	Yes
*DOTFILL	Yes	Yes	Yes	Yes
*EXPAND+	Yes	Yes*	Yes*	Yes
*EXPLODE	Yes	Yes	N/R	Yes
*FKEYS	Yes	Yes	No	Yes
*GOFF	Yes	Yes	Yes	Yes
*GON	Yes	Yes	Yes	Yes
*ITALICS	Yes	Yes	Yes	Yes
*MODERN	Yes	Yes	Yes	Yes
*NORMAL	Yes	Yes	Yes	Yes
*PROGRAM	Yes	Yes	Yes	Yes
*ROMS	Yes	Yes	Yes	Yes
*SCREEN	Yes	Yes	Yes	Yes
*SCROLL	Yes	Yes*	Yes*	No
*VARS	Yes	Yes	Yes	Yes

Key: * will not function in shadow screen mode
N/R not relevant as font already exploded

FITTING THE CHIP

BBC B,B+: Before fitting the chip it is important to switch off and unplug the micro. Then proceed as follows:

1) Remove the fixing screws which secure the computer lid and remove the lid.
2) Remove the screws and nuts which secure the keyboard to the computer chassis.
3) Unplug the keyboard ribbon cable, and unplug the loudspeaker connection. Lay both aside carefully.
4) Locate one of the vacant ROM sockets. These are at the bottom right of the BBC B board, and to the left near the power supply on the B+ (if in doubt, check the position of the sockets in your User Guide). UserROM will also function perfectly well in any of the ROM extension boards available. Now insert the chip into the vacant ROM socket you have chosen. Note that the semi-circular notch in the chip must be at the back -- that is, pointing away from the keyboard. Take care that all the pins are in the socket holes and then press home the chip carefully. But DO NOT FORCE the chip and bend the pins.
5) Replace the loudspeaker plug.
6) Replace the keyboard and its fixing screws.
7) Replace the keyboard ribbon cable.
8) Replace the lid and its fixing screws.

Master: Fit the chip into an EPROM cartridge.
Electron: Use ROM board (or cartridge in Plus 1).

TESTING

Once you have switched on your computer, UserROM should immediately tell you of its presence (though this may not be the case if you have other ROMs present). For example, when power is switched on the screen may show:

```
BBC Computer 32k
UserROM 1.00
Acorn DFS
BASIC
>
```

However, do not be too concerned at this stage if you do not see the UserROM prompt or if the version number, 1.00 in the above example, differs slightly. To test that your UserROM is installed and working correctly type in the following:

```
*HELP USERROM <Return>
```

The screen should then list the commands available from

UserROM, along with any parameters expected:

```
      UserROM 1.00
       BORDER
       CHECK
       CIRCLE X%,Y%,R%
       COLFILL X%,Y%,T%
       COMPRESS fsp
       DOTFILL X%,Y%,T%
       EXPAND fsp
       EXPLODE
       FKEYS
       GOFF
       GON addr
       ITALICS no
       MODERN no
       NORMAL
       PROGRAM
       ROMS
       SCREEN fsp
       SCROLL
       VARS
     OS 1.20
```

Again, don't worry if the version number given with your UserROM differs.

This *HELP menu may be called at any time. It provides you with a list of all the new commands that have been added to the vocabulary of your computer. Using these commands has been made as simple as possible and the rest of this manual describes their use with some practical demonstrations.

MEMORY USAGE

The information given here is not needed for normal use of UserROM, but is provided for completeness. In normal use UserROM requires three bytes of memory to operate. These are at the top of the user area in zero page and are bytes &8D, &8E and &8F. UserROM does make use of the rest of the user area from &70 to &8C as workspace, however it preserves the contents at the bottom of the hardware stack, restoring them before exit.

The recursive *COLFILL and *DOTFILL routines use page 9, ie memory from &900 to &9FF, as a scratchpad and workspace. It is highly unlikely that these areas will ever be needed while either of these commands is in operation. *SCROLL uses page 10, ie &A00 to &AFF to store the ASCII character string for scrolling. &900 to &9FF are also used by *SCROLL as workspace. Again, this area of memory is unlikely to be needed for other purposes during the operation of *SCROLL.

USING THE COMMANDS

Once installed all 19 commands are available for use. All of the UserROM's commands are accessed directly at the keyboard or form part of programs by typing an asterisk ('*') followed by the command name. As with all 'star' commands, a UserROM command must always be either the only command on a line or the last command in a multi-statement line. This is most important to remember. If you include commands on the same line as a UserROM command, but after the command they will be totally ignored by Basic. Remember the rule is 'after not before'.

The only other rule is that all UserROM commands must be typed in capital letters, and they may not be abbreviated. For example, *HELP USERROM will work, but *HELP userrom will not.

Some commands require extra information for them to carry out their tasks. This information may be passed to them in two ways, either direct or indirect. Direct information is typed after the command itself, leaving a space between the end of the command and the information required. For example, the command *SCREEN will save the current graphics screen to tape or disc. To do this however it needs a filename to save it under, and this should follow the command, ie:

```
*SCREEN SPIDER
```

where SPIDER is the filename.

Indirect information is used by commands such as *CIRCLE. This command draws a circle on the graphics screen. To do so it requires three items of information, namely the X and Y co-ordinates for the circle centre and the radius of the circle. Thus *CIRCLE expects to find this information in the integer variables X%, Y% and R% and so these should be set before using the command. For example:

```
10 MODE 2
20 X%=500:Y%=500:R%=100
30 *CIRCLE
```

You will find full details and examples of information needed by any UserROM commands in the command descriptions that follow.

CLASHING COMMANDS

In general, to use a command or facility of the UserROM you will normally only ever need to type in the command, remembering to prefix it first with *, ie:

```
*ROMS
*FKEYS
```

If you already have several ROMs fitted, there may be the odd occasion when they use a similar command. At such times the ROM in the highest priority socket will be given preference and will process the command (see *ROMS command below for details of ROM priority). This may mean that the command in the ROM with the lower priority can never be used. A means exists in UserROM that should allow you access to all commands at all times. All that you need to do is to prefix the command directly with the letter U (for UserROM). Note that as with all letters in a UserROM command the U must be a capital letter.

As an example suppose that the command *CHECK exists within a higher priority ROM than UserROM. To use the UserROM command *CHECK, simply place a U after the * but before the C. The command thus becomes *UCHECK. In the same way, *ROMS becomes *UROMS, *DOTFILL becomes *UDOTFILL and so forth.

FACILITIES

UserROM gives the following extra commands, either typed from the keyboard, or included as part of a program:

*BORDER - sets up a mode 7 screen with a standard screen border prior to use of *SCROLL.
*CHECK - produces a checksum listing of a program in memory.
*CIRCLE - will draw a circle in any graphics mode.
*COLFILL - a sophisticated, recursive colour filling routine. Provides 27 colourful fills.
*COMPRESS - takes a graphics screen and compresses it in size thereby taking less storage space on disc and tape.
*DOTFILL - a recursive, dot filling graphics routine. Provides 23 new black and white shades.
*EXPAND - returns a previously *COMPRESSed screen to its original state.
*EXPLODE - explodes the alphanumeric character set into RAM.
*FKEYS - lists the definitions in function keys.
*GOFF - turns the graphics compile mode off.
*GON - turns the graphics compile mode on.
*ITALICS - selects an italic character font.
*MODERN - selects a 'modern' character font.
*NORMAL - selects the normal character font.
*PROGRAM - recovers a bad program.
*ROMS - lists the sideways ROMs present in your machine.
*SCREEN - saves the current graphics screen to disc or tape.
*SCROLL - moves a scrolling message across the mode 7 screen.
*VARS - lists all names used for non-integer variables.

*HELP USERROM - lists commands available in UserROM.

The following abbreviations are used in the *HELP USERROM listing:

X%,Y% - denotes the command accesses X% and Y% variables.
R%,T% - denotes the command accesses R% and T% variables.
fsp - file specification expected, ie filename to follow.
addr - hexadecimal address expected (& not needed).
no - a number or numbers may be specified.

The prefix 'U' can be added to any command to avoid clashing with any other ROM software.

ERROR MESSAGES

The UserROM commands will generate error messages when information is incomplete or they encounter an error during operation. They are quite destinctive in that they will always begin with UserROM. So a typical UserROM error message is:

```
UserROM: Escape
```

In this instance it denotes that Escape was pressed during the action of a UserROM command. A complete list of UserROM error messages is listed below.

Message	Code	Error
Bad address	141	Illegal hexadecimal address was encountered.
Page error	142	Address specified for *GON is below PAGE memory marker.
Top error	143	Address for *GON is below TOP memory marker
Escape	144	Escape has been pressed.
Bad program error	145	No program can be detected, try *PROGRAM. This error generated usually by program corruption.
No program present	146	No program can be found at all.
Mode x	147	Not the correct graphics mode for *EXPAND. Select mode x.
No filename	148	The command expected a filename after the command.
Not graphics mode	149	A graphics mode was expected by *SCREEN.
Bad mode	150	Not a graphics mode for *COLFILL and *DOTFILL.
Bad tone	151	Not a legal tone number for *COLFILL or *DOTFILL.

All the above errors may be trapped using the ON ERROR command - see the User Guide for details. An error number may be printed with the command:

```
PRINT ERR
```

Similarly, REPORT will repeat the last error message. Further details on how to deal with errors can be found in the User Guide.

THE USERROM COMMANDS

*BORDER

This command is intended for use prior to *SCROLL. It first selects the mode 7 teletext screen and then sets up a border top and bottom through which a bulletin board message can scroll. See *SCROLL for further details. A simple design is provided, but you can construct your own borders. The procedure in listing 1 allows you to define your own border.

```
10REM Personal border set-up procedure
20MODE 7
30PROCborder("")
40PROCborder("   Acorn User Bulletin Board")
50PROCborder("")
60PRINT''''''''''
70PROCborder("")
80PROCborder("   Acorn User UserROM 1986")
90PROCborder("")
100END
110:
120DEF PROCborder (print$)
130FOR L%=1 TO 2
140PRINT CHR$(141);CHR$(131);CHR$(157);CHR$(132);print$
150NEXT
160ENDPROC
```

Listing 1. Personalised titles for *BORDER

Associated commands: *SCROLL.
Errors: *BORDER can not create an error.

*CHECK

This command should not normally be used within a program. It provides a list of checksums for each line of a program to be used with checksum listings presented in Acorn User. This therefore provides you with a quick and simple way to find exactly what line any mistakes are in by comparing the given checksum value. The checksum listing looks like this:

```
Line  Checksum
 10      291
 20      499
 30      517*
```

To allow for ease of checking each checksum is generated a line at a time. The next is printed by pressing the space bar. If a checksum figure is terminated by an asterisk then there is a space (or spaces) at the end of the line. These

of course cannot be seen on screen but they do effect the checksum values. Once spotted, they can be edited out.

Errors: UserROM: No program present.
Error number 146.
This error is generated if CHECK cannot see a program at PAGE. Type OLD to restore the program. If the error message 'Bad program' is generated see *PROGRAM.

***CIRCLE**

The standard BBC and Electron micros lack a circle drawing command. UserROM provides this command. Three indirect parameters are expected by *CIRCLE, namely the X and Y screen co-ordinates of the circle centre, and its radius. These values should be placed into the X%, Y% and R% variables prior to use. For example, to draw a circle roughly central in a mode 2 screen of diameter 100 use:

```
10 MODE 2
20 X%=500 : Y%=500
30 R%=50  : REM dia=2*radius
40 *CIRCLE
```

Coloured circles can be generated by using the GCOL statement (see User Guide). Listing 2 shows how the value of R% can be used as a loop variable to provide an incrementing radius. Figure 2 is its output.

Errors: *CIRCLE cannot create an error.

```
10REM *CIRCLE demonstration
20MODE 2
30X%=500:Y%=500
40FOR R%=0 TO 100 STEP 5
50GCOL 0,RND(4)
60*CIRCLE
70NEXT R%
```

Listing 2. Example of *CIRCLE

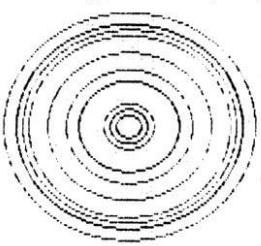


Figure 2. Printer dump output of listing 2

*COLFILL

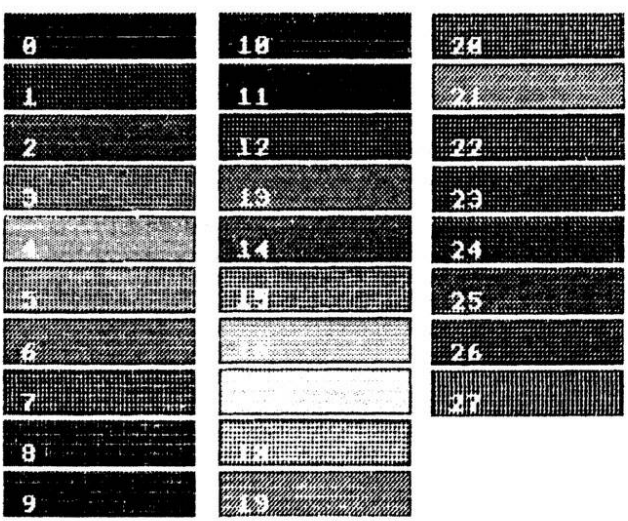
This command provides an extra 27 colourful shades. It is recursive and will completely fill an enclosed area. Three indirect parameters are required by *COLFILL: the starting point of the recursive colour fill should be placed in the X% and Y% variables; the tone in T%. A tone takes a value between 0 and 27 inclusive. This program will plot and colour fill a circle:

```
10 MODE 2
20 X%=500:Y%=500
30 R%=100
40 *CIRCLE
50 T%=2 : REM Tone number 2
60 *COLFILL
```

Listing 3 displays each *COLFILL shade. Keep a copy handy to help you select the correct colour. Figure 3 gives a limited indication of the output. *COLFILL is available in any graphics mode, but works best in mode 2.

Errors : UserROM: Bad tone - Error number 151.
Denotes T% not in range 0 to 27.
UserROM: Bad mode - Error number 150.
Denotes not a graphics mode.
UserROM: Escape - Error number 144.
Denotes Escape key terminated operation.

Figure 3. 28 colour shades from listing 3



```
10REM *COLFILL demonstration
20MODE1
30FOR T%=0 TO 27
40X%=50+400*(T% DIV12)
50Y%=940-84*(T% MOD12)
60IF T%>23 Y%=Y%-378
70PROCblock(X%,Y%,T%)
80NEXT
90X%=0:Y%=0:T%=4
100*COLFILL
110END
120DEFPROCblock(X%,Y%,T%)
130GCOL0,2
140MOVE X%,Y%
150PLOT1,0,72:PLOT1,350,0
160PLOT1,0,-72:PLOT1,-350,0
170X%=X%+32:Y%=Y%+32
180*COLFILL
190VDU5
200GCOL0,0
210MOVEX%,Y%
220PRINT;T%
230VDU4
240ENDPROC
```

Listing 3. Generates colour tones

***COMPRESS**

This command must be followed by a filename of up to seven letters. It will save the current graphics screen to disc or tape in a compressed form with the given name. This has the advantage that less storage space is required on disc or tape. It will also reduce tape loading and saving time.

The amount of compression will depend on the complexity of the graphics screen. For example the screen generated by listing 3 with *COLFILL will normally require a full 20k, or 20480 bytes, of storage on tape or disc. Saving this same screen with *COMPRESS compacts the storage space down to 7749 bytes - less than half and getting on for a third of the normal space required. A simple blank 20k screen will compress to just 185 bytes. The following program saves a simple 20k graphics screen in just 718 bytes.

```
10 MODE 2
20 MOVE 0,0
30 MOVE 1000,0
40 PLOT 85,500,1000
50 *COMPRESS TRIANG
```

The filename does not have to be enclosed within quotes. A

filename may also specify a disc drive number, eg:

```
50 *COMPRESS :2.test
```

would compress the current screen into a file called 'test' on drive 2. Tape users will find it advantageous to turn saving messages off first with *OPT 1,0.

The compression routine is performed in two passes to cope with hatching techniques. The current graphics mode is saved to disc or tape. Note that the colour palette is not saved however and you should note this if required using OSWORD 11 (see User Guide).

*COMPRESS will not work with *SHADOW mode in operation on the BBC B+ or Master. Nor will it work with the 6502 second processor operative. In such cases the screen should first be saved, using *SCREEN, and reloaded into the normal screen memory without Shadow or second processor memory enabled and compressed in the normal manner.

Associated commands: *EXPAND
Errors: UserROM: Escape - Error number 144.
Denotes program terminated by Escape.
UserROM: No filename - Error number 148.
No filename was specified after the command.

*DOTFILL

This command provides an extra 23 black and white shades. The command is recursive and as such it will completely fill in an enclosed area. Three indirect parameters are required by *DOTFILL. The starting point of the recursive colour fill should be placed in the X% and Y% variables. The tone of the dot fill should be placed in the variable T%. A tone takes a value between 0 and 23 inclusive. The following short program will shade fill a circle in the centre of the screen.

```
10 MODE 2
20 X%=500:Y%=500
30 R%=100
40 *CIRCLE
50 T%=6 : REM Tone No.6
60 *DOTFILL
```

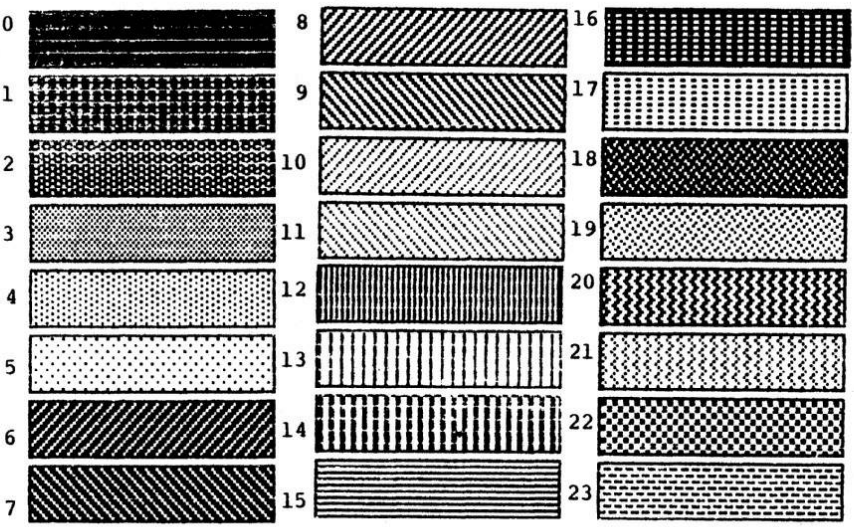
Listing 4 shows *DOTFILL to full effect, using a mode 2 screen to display each shade. It is recommended that you keep a copy of this program to hand to help you select the correct shade for your needs. Figure 4 gives some indication of the output. *DOTFILL can be used in any graphics mode.

Errors: UserROM: Bad tone - Error number 151.
Denotes T% not in range 0 to 23.
UserROM: Bad Mode - Error number 150.
Denotes not a graphics mode.
UserROM: Escape - Error number 144.
Denotes Escape key terminated operation.

Listing 4. DOTFILL demonstration

```
10REM *DOTFILL demonstration
20MODE1
30FOR T%=0 TO 23
40X%=50+400*(T% DIV12)
50Y%=940-84*(T% MOD12)
60IF T%>=12 Y%=Y%-378
70PROCblock(X%,Y%,T%)
80NEXT
90END
100DEFPROCblock(X%,Y%,T%)
110MOVEX%,Y%
120PLOT1,0,72:PLOT1,350,0
130PLOT1,0,-72:PLOT1,-350,0
140X%=X%+32:Y%=Y%+32
150*DOTFILL
160ENDPROC
```

Figure 4. Choice of 24 DOTFILL tones



*EXPAND

This command performs the reverse of *COMPRESS and should be followed by a filename. It will load the file specified, which should have previously been saved in a compressed form, into the current graphics mode expanding it to its correct size as it does so. The graphics mode must be the one from which it was originally saved otherwise an error will occur. If the specified file does not exist then a 'channel' error will occur. This program would expand a file called TRIANG back into a mode 2 screen.

```
10 MODE 2
20 *EXPAND TRIANG
```

The filename does not have to be enclosed within quotes, and it may specify a disc drive number, eg:

```
50 *EXPAND :2.test
```

would expand a file called test from drive 2. Tape users should turn saving messages off first with *OPT 1,0.

The expansion routine is performed in two passes to cope with hatching techniques. *EXPAND will not work with *SHADOW mode. Nor will it work with the 6502 second processor operative.

Associated command: *COMPRESS
Errors: UserROM: No filename - Error number 148.
No filename was specified after the command.
UserROM: Mode x - Error number 147.
Incorrect mode for expanding. Select mode x.

*EXPLODE (B,B+ and Electron only)

This command should be issued prior to using *ITALICS or *MODERN. In effect it performs a *FX20,6 and explodes the character set from ROM into RAM. This command should be used with caution as it has the effect of using up &600 bytes. As such it should always be used in immediate mode (ie not in programs) and PAGE should be increased by &600 before doing so. First find the hex value of PAGE in your micro. To do this type:

```
PRINT ~PAGE
```

In a standard DFS machine this will normally be &1900. If an ADFS is fitted it may be &1D00. Now calculate the new value needed for *EXPLODE. On a DFS micro this would be:

```
PRINT ~(&1900+&600)
```

Giving a value of &1F00. Set PAGE to this final result:

```
PAGE=&1F00
```

(or whatever your total is). Next type NEW. Now enter *EXPLODE and proceed as normal. Note however that PAGE on the Master is set at &E00 and there is no need to do any of this as the character font is always fully exploded.

Associated commands: *ITALICS, *MODERN, *NORMAL.
Errors: No errors can be generated by *EXPLODE.

*FKEYS

This command provides a list of the current function key definitions, for viewing or editing. Any keys not defined are listed as blank. Figure 5 shows the action of the command assuming several function keys have been defined.

Figure 5. Example result of *FKEYS

```
*KEY  0 OLD|M *CHECK|M
*KEY  1 *PROGRAM|M
*KEY  2 $&A00="Gone to lunch...."|M
*KEY  3 *BORDER|M *SCROLL|M
*KEY  4 PAGE=&2100|M *EXPLODE|M
*KEY  5 *ITALICS|M
*KEY  6 *MODERN|M
*KEY  7 *NORMAL|M
*KEY  8 *UROMS|M
*KEY  9 *SDUMP +|M
*KEY 10
*KEY 11
*KEY 12
*KEY 13
*KEY 14
*KEY 15
```

***GOFF**

This command switches a graphics compiler off. See *GON for details.

***GON**

This command switches the graphics compilation mode on. Normally *GON and *GOFF will be the first and last commands in a program. Sandwiched between them will be a program of graphics routines - these commands will be broken down into their VDU base components and stored within a memory buffer begining at the hexadecimal address specified after *GON. On completion a 'beep' will sound and an address will be printed. This is the end address of the graphics buffer. The memory contents between the buffer start address and the end address returned by the program can then be *SAVEd. This can subsequently be *EXECed at which time the graphics commands will be executed. In this way, a library of graphics routines can be stored on disc or tape and called in when needed. Enter listing 5 below.

```
 10*GON 6000
 20MODE 2
 30FOR N%=1 TO 50
 40GCOL 0,RND(3)
 50MOVE RND(1280),RND(1023)
 60MOVE RND(1280),RND(1023)
 70PLOT 85,RND(1280),RND(1023)
 80NEXT
 90PRINT
100*GOFF
```

Listing 5. Graphics compiler in action

Line 10 shows that the buffer start address is 6000. Any free area of memory can be used. Note that we are in fact using standard mode 2 screen memory, however mode 2 is not invoked at compile time and the program should remain in mode 7 or mode 6.

RUN the program. The message 'Compiling graphics - please wait...' should appear. After a few moments a beep will sound and the message 'End address:<&641E>' will appear. Now *SAVE this area of memory:

```
*SAVE test 6000 641E
```

Once saved simply *EXEC it back in to see the graphics in action:

```
*EXEC test
```

*ITALICS

This command selects an italic character font so that the letters appear to be slanted on the screen in all modes except mode 7. Before using *ITALICS, the value of the memory marker PAGE should be raised by &600 and the character font should be exploded with *EXPLODE (except on the Master). Text may be printed as normal. Figure 6 shows the italic character set.

*ITALICS may be postfixed with one of the numbers 1,2,3, or 4. In such cases only part of the character font will be italicised.

*ITALICS 1 - acts on 0-9 inclusive
*ITALICS 2 - acts on A-Z inclusive
*ITALICS 3 - acts on a-z inclusive
*ITALICS 4 - italicises !"£$%&'()~

It is quite legal to specify more than one parameter. The only proviso is that they are in numberic order. Thus *ITALICS 23 will italicise A-Z and a-z inclusive. Italic characters may be freely mixed with 'modern' and normal characters. See *MODERN below for further details.

Associated commands: *EXPLODE, *MODERN, *NORMAL
Errors: none.

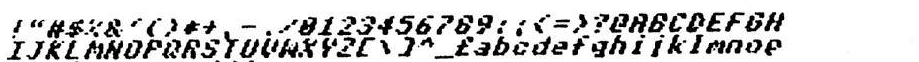


Figure 6. Italic character set

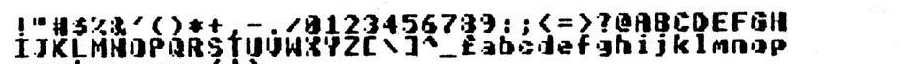


Figure 7. 'Modern' character set

Italics can be used for stress
modern can be mixed with *italics*

Figure 8. Italics, 'modern' and normal text mixed

***MODERN**

This commands selects what we'll call a 'modern' character font so that the character set appears chunkier on screen in all modes except mode 7. Before using *MODERN PAGE should be raised by &600 and the character font should be exploded with *EXPLODE. Text may be printed as normal. Figure 7 above shows the modern character set.

*MODERN may be postfixed with a number or series of numbers in the range 1 to 4. In such cases only part of the character font will be modernised.

```
*MODERN  1 - modernises 0-9 inclusive
*MODERN  2 - modernises A-Z inclusive
*MODERN  3 - modernises a-z inclusive
*MODERN  4 - modernises !"£$%&'()~
```

It is quite legal to specify more than one parameter. The only proviso is that they are in numberic order. Thus *MODERN 23 will modernise A-Z and a-z inclusive. Modern characters may be freely mixed with italics and normal characters. Figure 8 above was produced using listing 6.

Associated commands: *EXPLODE, *ITALICS, *NORMAL

```
 10MODE 6
 20*EXPLODE3
 30*ITALICS
 40PRINT"Italics ";
 50*NORMAL
 60PRINT"can be used for stress"
 70*MODERN
 80PRINT"modern";
 90*NORMAL
100PRINT" can be mixed with ";
110*ITALICS
120PRINT"italics"
130*NORMAL
```

Listing 6. Mixing text styles as in figure 8

***NORMAL**

This command reselects the standard character font by imploding it. It does so by performing a *FX20,0.

***MODERN**

This commands selects what we'll call a 'modern' character font so that the character set appears chunkier on screen in all modes except mode 7. Before using *MODERN PAGE should be raised by &600 and the character font should be exploded with *EXPLODE. Text may be printed as normal. Figure 7 above shows the modern character set.

*MODERN may be postfixed with a number or series of numbers in the range 1 to 4. In such cases only part of the character font will be modernised.

```
*MODERN  1 - modernises 0-9 inclusive
*MODERN  2 - modernises A-Z inclusive
*MODERN  3 - modernises a-z inclusive
*MODERN  4 - modernises !"£$%&'()~
```

It is quite legal to specify more than one parameter. The only proviso is that they are in numberic order. Thus *MODERN 23 will modernise A-Z and a-z inclusive. Modern characters may be freely mixed with italics and normal characters. Figure 8 above was produced using listing 6.

Associated commands: *EXPLODE, *ITALICS, *NORMAL

```
 10MODE 6
 20*EXPLODE3
 30*ITALICS
 40PRINT"Italics ";
 50*NORMAL
 60PRINT"can be used for stress"
 70*MODERN
 80PRINT"modern";
 90*NORMAL
100PRINT" can be mixed with ";
110*ITALICS
120PRINT"italics"
130*NORMAL
```

Listing 6. Mixing text styles as in figure 8

***NORMAL**

This command reselects the standard character font by imploding it. It does so by performing a *FX20,0.

*SCREEN

This command will intelligently save the current graphics screen under a specified filename. Note that it will not save text-only modes 3 and 6. The command works out which mode it is in and saves the corresponding memory block to tape or disc. Tape users should use *OPT 1,0 first to disable saving messages. To to save a mode 7 screen use:

```
10 MODE 7
20 PRINT "*SCREEN Demo"
30 *SCREEN test
```

Disc users may use a drive specification in the normal way:

```
*SCREEN :2.test
```

A saved *SCREEN may be reloaded by selecting the required mode and *LOADing:

```
10 MODE 7
20 *LOAD test
```

Errors: UserROM: No filename - Error number 148.
No filename was specified after the command.

*SCROLL

This command allows you to set up a scrolling bulletin board display across a mode 7 screen. It does this by extracting a character string from &A00 and exploding it onto the screen. This makes it ideal for leaving messages to people or for producing scrolling displays at home, school or work. The following program shows how it works. Note it is advisable to perform a CTRL-BREAK before using *SCROLL.

```
10 $&A00="Gone to lunch - back soon..."
20 *BORDER
30 REPEAT
40 *SCROLL
50 UNTIL 0
```

The message is placed as a string at &A00. Character strings may be up to 255 characters in length. Note that some full stops have been tagged onto the end of the message - this is deliberate to prevent the start and end butting up directly against one another. *BORDER is used to set up a suitable surround, alternatively use listing 1 to set your own. As *SCROLL will only scroll the message across the screen once it should be embedded within a suitable loop to ensure a continuous scroll.

***VARS**

This command will list all the non-integer variables used within a program. This is useful when writing long programs as it provides a ready made list thus ensuring that you do not use the same variable name twice. For ease of reference the variable name list is arranged alphabetically. When there is a large number of variable names, paged mode is entered to inhibit scrolling. Pressing the SHIFT key will cause the page to scroll. Enter and RUN the following program then type *VARS to see the effect:

```
10 alpha=10
20 beta=20
30 theta=30
40 gamma=40
50 sigma=50
60 delta=60

*VARS
  alpha
  beta
  delta
  gamma
  sigma
  theta
```

Redwood Publishing Ltd, 141-143 Drury Lane, London WC2B 5TF