

# ELBUG

## SUPPLEMENT TO BEEBUG

November 1985 — Volume 3, Issue 1

### Your Guide To BEEBUG

None of the programs appropriate to the Electron in this month's BEEBUG need any alteration to run. However, a few general comments on Beeb program conversion may prove useful to all Electron users.

Beeb programs sometimes fail to run on the Electron because of the Beeb's mode 7 and extra sound facilities. However, conversion rarely proves very difficult.

The Beeb's mode 7 display is of the same format as mode 6 but with a full range of colours and crude graphics. Mode 7 is not available on the Electron, however the Electron defaults to mode 6 when it encounters a MODE 7 command. So BBC micro programs that include a mode 7 display will therefore run on the Electron but often with a corrupted (and monochrome) display.

The Beeb has three sound channels and control over the volume of notes. Electron Basic includes all the SOUND and ENVELOPE parameters that are used in the BBC micro even though several have no effect. This means that Beeb programs making full use of sound will still run on the Electron unaltered, though not, of course, producing the full effect.

The other major difference between the Electron and the BBC micro is speed. The Electron is much slower and particularly slow in modes 0, 1, and 2. BBC micro

programs often benefit from a change of mode (admittedly, with a loss of some colours) when running them on the Electron. The changes needed will be given in this ELBUG supplement.

#### BEEBUG FILER

This useful database management program requires the facility of random access to files. This is something a cassette based system cannot provide. However, ELBUG members lucky enough to own a Plus-3 (or any other) disc system will find that this program, and the forthcoming extensions, work equally well on the Electron.

#### COMPRESSING SCREEN DISPLAYS

These two handy utilities work perfectly on the Electron allowing you to compress very long cassette picture files to around a quarter of their length.

#### WORDWISE PLUS PROGRAM COMPACTOR

Unfortunately the Wordwise Plus word processor is not available for the Electron

#### WRITING YOUR OWN COMPILER

The complete compiler program is very

G CASSETTE ELBUG CASSETTE ELBUG CASSETTE ELBUG CASSETTE ELBUG CASSETTE ELBUG CASSETTE E

The November ELBUG magazine cassette is packed with programs for Electron owners. Many of the programs from this month's BEEBUG are there - Hidden line removal, the first part of the compiler program, the BEEBUG Filer program (part 1), the two picture compression and decompression routines, the First Course print formatting function, the Workshop procedures, and the Quasimodo game - plus the two Mixed Modes programs from this ELBUG supplement. All this for just £3.00 plus 50p post and packing.

long so Electron users (without the benefits of the Beeb's mode 7) will find that memory is at a premium. However, the program does work perfectly on the Electron with no changes needed to the part listed this month or to those parts coming in the next two issues.

#### ROM UTILITIES FOR ALL

If you have a Plus-3 and a ROM board for your Electron (or a Plus-1 and one of ACP's ROM adaptors, reviewed last month) then you can use one of the ROMs reviewed here. ACP's Advanced Disc Toolkit is written for both the BBC micro and the Electron and offers the Electron disc user a wealth of utilities.

#### PROGRAMMING WITH WORDWISE PLUS (Part 2)

Unfortunately the Wordwise Plus word processor is not available for the Electron so this excellent tutorial series will not be of interest to ELBUG members.

#### HIDDEN LINE REMOVAL

This fascinating program will produce wireframe 3D projections of solid objects and then remove the lines that, in real life, would be hidden from view to give a realistic solid image.

The program makes good use of the Electron's excellent graphics abilities and works perfectly without alteration.

#### COMPUTER CONCEPTS SPEAKS OUT

The Computer Concepts speech ROM is not available for the Electron as it requires the Texas Instruments speech chip for the Beeb that cannot be fitted to an Elk.

#### WORKSHOP - VIRTUAL ARRAYS

This clever utility allows you to keep lengthy, memory-gobbling arrays on disc instead of using up the limited memory resources of your micro. ELBUG members with a Plus-3 unit, or another disc drive interface that supports the usual Basic disc filing system commands will find that this program works on an Electron too.

#### VIEWSTORE

The excellent Viewstore database manager requires random access just like our own BEEBUG Filer and so cannot work on the unexpanded Elk. Because of the comparatively small numbers of Plus-3 disc units sold, Acornsoft has no plans to release an Electron version.

#### PROGRAMMING SIDEWAYS RAM AND ROM

Although the concept of sideways software, in either ROM or RAM, is equally applicable to the Electron as to the BBC micro, this utility program to create sideways software will not run on the Electron because it requires the memory economy of the Beeb's mode 7 display.

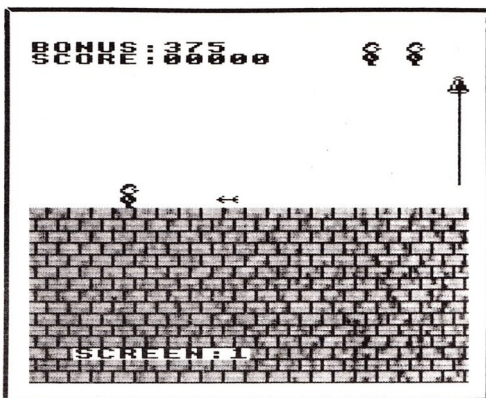
#### FIRST COURSE - PRINT FORMATTING (Part 2)

This useful, and very short, utility will help you to print any kind of table, or other tidy presentation of numbers, as easily and accurately as is possible.

The routine works perfectly on the Electron without any changes made to the listing in BEEBUG.

#### GENERATING DIAGRAMS AND DRAWINGS

The two packages reviewed here are, sadly, not available on the Electron.



#### QUASIMODO

This excellent rendition of the popular arcade game makes good use of the Electron's graphics and sound to produce a full colour action game.

The listing of Quasimodo in BEEBUG is fully compatible with the Electron and requires no changes.

#### HINTS

The following hints from this month's BEEBUG are applicable to ELBUG members:

FILLING UP  
EASIER BYTES  
\*FX VALUES  
QUICK LINES  
DEFAULT VECTOR VALUES

# ELBUG HIGH SCORES

It is a while now since the noble efforts of Electron owners in the never-ending fight against the evils on the screen has been recognised on these pages, so here is the latest high scores table for ELBUG members.

Needless to say, further entries are very welcome. If you have beaten the high score on one of these games, or produced a good rating on one not listed here, then don't be shy; drop us a line to give us the good news. Please send the record of your achievements to:

High Scores, PO Box 50, Holywell Hill, St. Albans, AL1 3YS.

| Supplier   | Game            | Score   | Player            | Supplier   | Game            | Score  | Player      |
|------------|-----------------|---------|-------------------|------------|-----------------|--------|-------------|
| AARDVARK   | Zalaga          | 171580  | D.Fell            | ELBUG MAG  | Scrumpy         | 124450 | A.Gould     |
| A&F        | Chuckie Egg     | 4078180 | D.Anderson        | ELBUG MAG  | Spider Man      | 800    | C.Bishop    |
| A&F        | Cylon Attack    | 77300   | J.Cantrill        | ELBUG MAG  | Wee Shuggy      | 7848   | J.Devlin    |
| A&F        | Kamikaze        | 65310   | J.Gault           | ELBUG MAG  | Trackman        | 7105   | D.Devlin    |
| ACORNSOFT  | Arcadians       | 10960   | M.Palmer          | ELBUG CASS | 3D Maze         | 712    | A.Green     |
| ACORNSOFT  | Boxer           | 97750   | A.Gould           | ELBUG CASS | Space City      | 144352 | H.Buchanan  |
| ACORNSOFT  | Freefall        | 692     | D.Fell            | GEMINI     | Missile Control | 56980  | J.Glazier   |
| ACORNSOFT  | Hopper          | 19276   | J.Glazier         | ICON       | Space Station   | 1620   | W.Devlin    |
| ACORNSOFT  | Meteors         | 42300   | P.Haslam          | ICON       | Zorakk          | 63850  | J.Devlin    |
| ACORNSOFT  | Monsters        | 114080  | T.Shoaib          | IJK        | Caterpillar     | 255001 | D.Devlin    |
| ACORNSOFT  | Planetoid       | 59525   | A.Gould           | IMAGINE    | Pedro           | 4025   | G.Hallett   |
| ACORNSOFT  | Snapper         | 142570  | C.Field           | MICRO POW  | Bandits at 3.00 | 618    | J.Cantrill  |
| ACORNSOFT  | Starship Com.   | 1075    | M.Hall            | MICRO POW  | Croaker         | 17860  | P.Warren    |
| ALLIGATA   | Blagger         | 19580   | J.Cantrill        | MICRO POW  | Cybertron       | 149500 | P.Ellison   |
| ALLIGATA   | Bug Blaster     | 265366  | D.Morland         | MICRO POW  | Danger UXB      | 683200 | G.Peak      |
| ALLIGATA   | Guardian        | 345300  | J.Glazier         | MICRO POW  | Felix/Factory   | 12600  | D.Morland   |
| ALLIGATA   | Hell Hole       | 34580   | M.Trow-<br>bridge | MICRO POW  | Felix/Monsters  | 34590  | J.Gault     |
| ALLIGATA   | Lunar Rescue    | 25840   | M.Trow-<br>bridge | MICRO POW  | Felix/Weevils   | 11700  | M.Tonbridge |
| BEEBUGSOFT | Snake           | 4268    | V.Hunt            | MICRO POW  | Frenzy          | 399248 | A.Webster   |
| ELBUG MAG  | Block Blitz     | 19090   | D.Seward          | MICRO POW  | Gauntlet        | 14490  | D.Fell      |
| ELBUG MAG  | Breakout        | 27297   | A.Webster         | MICRO POW  | Ghouls          | 3027   | C.Seward    |
| ELBUG MAG  | Brickie Nickie  | 150433  | D.Devlin          | MICRO POW  | Invaders        | 7900   | I.McMillan  |
| ELBUG MAG  | Caterpillar     | 16900   | K.Bishop          | MICRO POW  | Killer Gorilla  | 216500 | J.Cantrill  |
| ELBUG MAG  | Digger          | 1300    | D.Devlin          | MICRO POW  | Missile Control | 2460   | M.Gallagher |
| ELBUG MAG  | Divebomber      | 32800   | P.Haslam          | MICRO POW  | Moonraider      | 14430  | P.Haslam    |
| ELBUG MAG  | Evasion         | 39950   | C.Burnside        | MICRO POW  | Positron        | 154340 | P.Haslam    |
| ELBUG MAG  | Flowers of Hell | 2870    | K.Bishop          | MICRO POW  | Rubble Trouble  | 23428  | D.Devlin    |
| ELBUG MAG  | Galactic Inv.   | 83400   | D.Devlin          | MICRO POW  | Swoop           | 17950  | P.Warren    |
| ELBUG MAG  | George/Dragon   | 5300    | S.Bishop          | POSTERN    | Pengwyn         | 17450  | A.Morland   |
| ELBUG MAG  | Hedgehog        | 17205   | C.Bishop          | QUICKSILVA | Mined Out       | 2615   | K.Bishop    |
| ELBUG MAG  | Hi-Lo           | 23000   | S.Hodg-<br>kinson | ROMIK      | Alien Break in  | 74910  | D.Devlin    |
| ELBUG MAG  | Hunt the Nos.   | 22000   | J.Cantrill        | SOFT INV.  | Vortex          | 33603  | A.Green     |
| ELBUG MAG  | Invasion        | 10450   | P.Haslam          | SOFT INV.  | 3D Bomb Alley   | 30670  | P.Haslam    |
| ELBUG MAG  | Kayak Kapers    | 1820    | A.Morland         | STARCADE   | Savage Pond     | 11365  | P.Seward    |
| ELBUG MAG  | Mars Lander     | 7210    | P.Haslam          | SUPERIOR   | Invaders        | 13650  | R.Tacagni   |
| ELBUG MAG  | Munch Man       | 28460   | D.Morland         | SUPERIOR   | Percy Penguin   | 4950   | G.Smith     |
| ELBUG MAG  | Robot Attack    | 6330    | C.Williams        | SUPERIOR   | Mr.Wiz          | 65410  | G.Smith     |
|            |                 |         |                   | VISIONS    | Overdrive       | 344990 | J.Cantrill  |
|            |                 |         |                   |            | D D Dennis      | 26450  | P.Warren    |

# MIXING MODES

P.S.Bhandari explains how to mix the Electron's different modes on the screen at the same time, and produce displays not seen on this computer before.

In BEEBUG Vol.3 No.10 there was a program to produce a display, on the BBC micro, consisting of up to three modes mixed on the screen at once. This program did not work on the Electron and could not easily be made to do so. However, ELBUG now presents the Electron version of Mixed Modes. This program will allow your Electron to display two modes on the screen at once so that you could, for example, display some 80 column text along with 16 colour graphics. Before looking at the use of the program, we will examine some of the principles involved.

## HOW THE PROGRAM WORKS

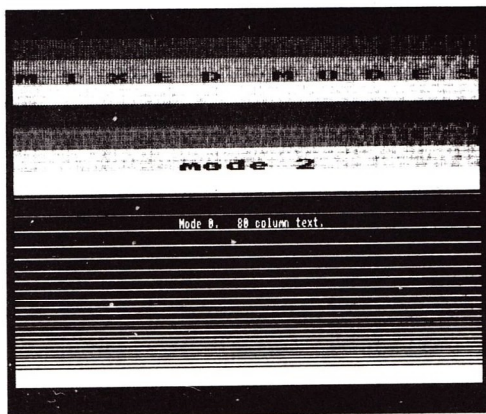
The picture displayed on a TV or monitor is made up from a dot which scans the screen horizontally from top to bottom. The total time for this dot to travel down the screen is  $1/50$ th of a second. The idea behind the Mixed Mode program is to change modes midway during the downward travel of this dot.

So, how do we trigger the changes of mode during this scan? This is done using 'Events' (see page 242 in the User Guide). Since the dot takes  $1/50$ th second to travel down, we need to 'interrupt' the processor at a rate greater than 50 times a second. Only one event can perform this task - the 'interval timer crossing zero' event. The interval timer can be set to  $1/100$ th of a second, and so we are able to have two modes on the screen at once (since  $2 \times 1/100 = 1/50$ ), the alternate mode chosen on every alternate event.

The display mode can be changed with the machine code equivalent of the MODE command. However, changing modes in this way also clears the screen. With our continuous mode selection, this would mean that nothing would ever appear on the screen. To change mode without clearing the screen we must access the Electron's ULA chip directly. One register in the ULA controls which mode is being displayed but this register also

controls other parts of the computer - the caps lock LED, the cassette motor, and whether the cassette system or sound is being used

This register resides in memory at location &FE07. Out of the eight bits, we are concerned only with three - bits 3, 4 and 5. These three bits hold the value of the mode in binary form. For example, for mode 5, these bits are set to 5 in binary (101). For mode 4 these bits would hold 100 - 4 in binary.



So, to change the mode half way through the TV's scan, we poke the required mode data into bits 3, 4 and 5 of &FE07, every  $1/100$ th of a second. However, this presents us with a problem: how do we poke values into bits 3, 4 and 5 of &FE07 without affecting the other bits, and so affecting the other functions controlled with this register? To do this we use the binary operators AND and OR, the operation of which is described on page 154 of the User Guide.

The trick is to read the value of the ULA register, clear bits 3, 4 and 5 (reset them to zero), put the mode value into these three bits and then poke the resulting value back into &FE07.

locations &FE08 and &FE09. If you want to change the actual colours (as you would normally do using VDU19), this can be done by altering these statements. However, the pattern followed by these is very complex and so trial and error is usually the best method.

For example, to change to mode 5, this procedure is followed: the three bits are ANDed with 0 and the rest are ANDed with 1. In other words the contents of &282 is ANDed with the byte:

$$= 128 + 64 + 0 + 0 + 0 + 4 + 2 + 1 = 199 = \text{6C7}$$

Next, the bits 3, 4, and 5 are set to represent the mode required, in this case 5. All the other bits are ORED with 0 and the three bits, 3, 4, and 5, are ORED with the value of the mode, in this case 101. The resulting byte to be ORED with the register to give mode 5 is therefore:

$$= 0 + 0 + 32 + 0 + 8 + 0 + 0 + 0 = 40 = 528$$

We now face another problem. How do we get the same mode always to be the top mode on the screen? The first time the  $1/100$ th of a second interrupt occurs the dot could be either on top or in the middle of the screen. Fortunately, there is another event called the 'vertical synchronization event' which occurs every time the dot is at the top of the screen. Thus, the first time a mode is chosen, this event should be used. When this happens the interval timer can also be set to change the display mode again half way down the screen scan.

### USING THE MIXED MODES

The first program generates the machine code necessary to actually mix the displayed modes. Type it into your Electron and enter the mode numbers for the modes you want displayed into lines 1000 and 1010. Remember you cannot mix the 'slow' modes (0, 1, 2, and 3) with the 'fast' modes (4, 5, and 6). The numbers in lines 1950, 1960, 1980, and 1990 of the first program control the colours displayed in any mode, by accessing the ULA registers residing at

Save the program before you run it. Now run the program and save the machine code (MIXCODE) generated onto another cassette, as prompted.

## LOCATING THE CODE

The machine code generated by this first program must be stored somewhere. The program reserves 256 bytes of memory by actually moving the Basic program up, so that PAGE is one page higher. This means that when this first program is loaded, no matter what the

|        |        |
|--------|--------|
| MODE 5 | MODE 5 |
| MODE 5 | MODE 5 |
| MODE 5 | MODE 5 |
| MODE 5 | MODE 5 |

```
MODE 4 MODE 4 MODE 4 MODE 4 MODE 4 MODE 4  
E 4 MODE 4 MODE 4 MODE 4 MODE 4 MODE 4 M  
ODE 4 MODE 4 MODE 4 MODE 4 MODE 4 MODE 4  
DE MODE 4 MODE 4 MODE 4 MODE 4 MODE 4  
MODE 4 MODE 4 MODE 4 MODE 4 MODE 4 MODE  
E 4 MODE 4 MODE 4 MODE 4 MODE 4 MODE 4 M  
ODE 4 MODE 4 MODE 4 DE 4 MODE 4 MODE 4  
4 MODE 4 MODE 4 MODE 4 MODE 4 MODE 4 MO  
DE 4 MODE 4 MODE 4 MODE 4 MODE 4 MODE 4  
MODE 4 MODE 4 MODE 4 MODE 4 MODE 4 MODE  
4 A MODE 4 A MODE 4 A MODE 4 A MODE 4
```

value of PAGE is originally, (dependent on whether you have a Plus-3, or any ROMs, etc. fitted), 256 bytes will be reserved on top of that.

If you do not want to raise PAGE in this way, the code can be stored elsewhere. Two suitable places are the function key buffer and the user defined character buffer. To use one of these change line 1070 of the first program to read P%=&B00 or P%=&C00 respectively.

```

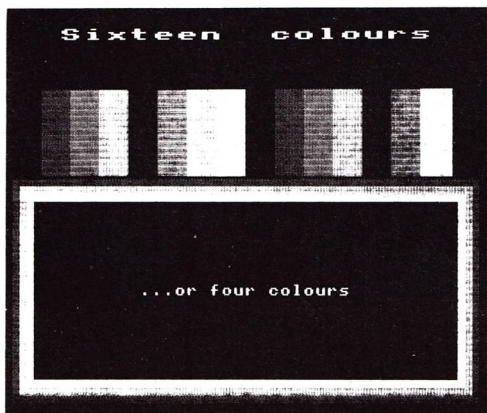
10 REM PROGRAM MIXED MODES
20 REM VERSION E0.2
30 REM AUTHOR P. BHANDARI
40 REM ELBUG NOV. 1985
50 REM PROGRAM SUBJECT TO COPYRIGHT
60 :

```

```

100 *KEY0F.I%=(TOP-PA.)TO0S.-1:?(PA.+I
%+256)=?(PA.+I%):NEXT|MPA.=PA.+256|MOLD|
MGOTO1000|M
110 *FX138,0,128
120 END
130 :
1000 topmode=2
1010 botmode=0
1020 *FX13,5
1030 *FX16
1040 origh=?&221
1050 origl=?&220
1060 FOR pass%=0TO3 STEP3
1070 P%=PAGE-254
1080 [OPT pass%
1090 .enter
1100 PHA \save registers
1110 PHP
1120 TXA:PHA
1130 TYA:PHA
1140 JSRinit
1150 JMP modeb
1160 .clock
1170 CMP#5
1180 BEQ onwards
1190 CMP#4

```



```

1200 BEQenter
1210 JMP (origh)
1220 .onwards
1230 PHA \save registers
1240 PHP
1250 TXA:PHA
1260 TYA:PHA
1270 .routine
1280 LDA&70 \check toggle
1290 BEQ modea
1300 .modeb
1310 LDA#0
1320 STA&70
1330 LDY #botmode

```

```

1340 LDA first,Y
1350 STA&FE08
1360 LDA second,Y
1370 STA&FE09
1380 LDA&282 \load ULA reg. mir
ror
1390 AND#&C7 \mask out mode bit
s
1400 ORA#(botmode*8) \set mode bits
1410 STA&FE07 \store in ULA reg.
1420 JMP finish
1430 .modea
1440 LDA#1
1450 STA&70
1460 LDY #topmode
1470 LDA first,Y
1480 STA&FE08
1490 LDA second,Y
1500 STA&FE09
1510 LDA&282 \load ULA reg. mir
ror
1520 AND#&C7 \mask out mode bit
s
1530 ORA#(topmode*8) \set mode bits
1540 STA&FE07 \store in ULA reg.
1550 .finish
1560 LDA#&FF \set
1570 STA&72 \ interval
1580 STA&73 \ timer
1590 STA&74 \ for
1600 STA&75 \ 1/100th
1610 STA&76 \ second
1620 LDA#4 \ (see
1630 LDY#0 \ User Guide
1640 LDX#&72 \ page
1650 JSR&FFF1 \ 239)
1660 PLA:TAY \restore registers
1670 PLA:TAX
1680 PLP
1690 PLA
1700 RTS
1710 :
1720 .init
1730 LDA#13 \disable
1740 LDX#4 \ sync.
1750 JSR&FFF4 \ event
1760 LDA#13 \disable
1770 LDX#5 \ timer
1780 JSR&FFF4 \ event
1790 LDA#clock MOD256\point
1800 STA&220 \ EVENTV
1810 LDA#clock DIV256\ to 'clock'
1820 STA&221 \ routine
1830 LDA#14 \enable
1840 LDX#5 \ timer
1850 JSR&FFF4 \ event
1860 RTS
1870 :
1880 .start

```

```

1890 LDA#enter MOD256\point
1900 STA &220 \ EVENTV
1910 LDA#enter DIV256\ to 'enter'
1920 STA &221 \ routine
1930 RTS
1940 .first
1950 EQU &30B07030 \data
1960 EQU &00307030 \ for
1970 .second
1980 EQU &311B7131 \ actual
1990 EQU &00317131 \ colours
2000 ]
2010 NEXT pass%
2020 MODE 6
2030 PRINT ""Insert cassette or disc t
o save MIXCODE"
2040 PRINT ""Press space bar when read
y"
2050 REPEAT UNTIL GET=32
2060 OSCLI "SAVE MIXCODE "+STR$~enter+"
"+STR$~P%+" "+STR$~start
2070 CHAIN "MIXDEMO"
2080 END

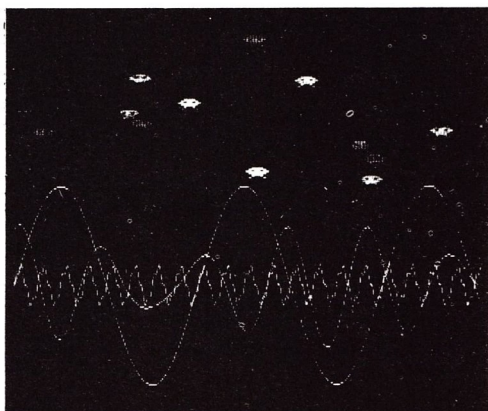
```

The second program is a demonstration of actually using the MIXCODE code generated by the first. This enables the screen synchronization event (line 110) which in turn starts the timer event. PROCdemo places some suitable graphics on each half of the screen to demonstrate the mode mixing. Of course, you can append your own program in its place using whichever modes you selected to be mixed by the first program. The demonstration program can be stopped at any time using Escape. To stop the mode mixing effect, type in \*FX13,4 followed by \*FX13,5 and then change mode.

Using the mixed modes display is not as simple as using a single mode display. Your program should first issue a MODE command (as in line 120). If two fast modes are being used this should be MODE 4. For two slow modes use MODE 2. To write or draw in one of the two modes displayed, without the result looking 'garbled', some memory locations in page 3 of memory need to be changed. The values they take depend upon the mode displayed in that part of the screen and must be set accordingly.

To make this easier the second program includes procedures that have all the required values for each mode. Thus the procedure PROCmode5 must be called first whenever doing anything in that half of the screen displaying mode 5. You need

only include the procedures for the modes you are using.



When writing a program to use the MIXCODE generated by the first program, these are the rules to follow:

1. Raise the value of PAGE (type PAGE=PAGE+256) before loading your program if you have not just run the first program.
2. The first line of your 'second' program must be \*RUN MIXCODE.
3. Set the base mode for your program. Use mode 4 for fast mixed mode displays and mode 2 for the slow ones.
4. Include the data procedures, PROCmode0, PROCmode1, etc., from the demo program as needed, and call the relevant one before using either section of the mixed mode display.

If you keep to these rules and instructions you will soon easily be able to make your Electron display text and graphics in just about any kind of format imaginable.

```

10 REM PROGRAM MIXDEMO
20 REM VERSION E0.2
30 REM AUTHOR P. BHANDARI
40 REM ELBUG NOV. 1985
50 REM PROGRAM SUBJECT TO COPYRIGHT
60 :

```

```

100 *RUN MIXCODE
110 *FX14,4
120 MODE 2
130 VDU23,1,0;0;0;0;
140 PROCdemo
150 END
160 :
1000 DEFPROCmode0
1010 ?&2F4=0: ?&304=127: ?&305=2: ?&320=12
7
1020 ?&321=2: ?&34F=8: ?&355=0: ?&360=1
1030 ?&361=7: ?&362=128: ?&363=1: ?&370=7
1040 ENDPROC
1050 :
1060 DEFPROCmode1
1070 ?&2F4=8: ?&304=63: ?&305=1: ?&320=63
1080 ?&321=1: ?&34F=16: ?&355=1: ?&360=3
1090 ?&361=3: ?&362=136: ?&363=17: ?&370=1
1100 ENDPROC
1110 :
1120 DEFPROCmode2
1130 ?&2F4=16: ?&304=159: ?&305=0
1140 ?&320=159: ?&34F=32: ?&355=2
1150 ?&360=15: ?&321=0: ?&361=1: ?&362=170
1160 ?&363=85: ?&370=1
1170 ENDPROC
1180 :
1190 DEFPROCmode3
1200 ?&2F4=24: ?&304=79: ?&305=0
1210 ?&320=0: ?&321=0: ?&34F=8: ?&355=3
1220 ?&360=1: ?&361=0: ?&362=128
1230 ?&363=1: ?&370=7
1240 ENDPROC
1250 :
1260 DEFPROCmode4
1270 ?&2F4=32: ?&304=63: ?&305=1: ?&320=63
1280 ?&321=1: ?&34F=8: ?&355=4: ?&360=1
1290 ?&361=7: ?&362=128: ?&363=1: ?&370=7
1300 ENDPROC
1310 :
1320 DEFPROCmode5
1330 ?&2F4=40: ?&304=159: ?&305=0
1340 ?&320=159: ?&321=0: ?&34F=16: ?&355=5
1350 ?&360=3: ?&362=136: ?&363=17: ?&370=1
1360 ?&361=3
1370 ENDPROC
1380 :
1390 DEFPROCmode6
1400 ?&2F4=48: ?&304=79: ?&305=0
1410 ?&320=79: ?&321=0: ?&34F=8: ?&355=6
1420 ?&360=1: ?&361=0: ?&362=128
1430 ?&363=1: ?&370=7
1440 ENDPROC
1450 :
1460 DEFPROCdemo
1470 PROCmode2
1480 FOR C%=1 TO 7
1490 GCOLOR,C%
1500 MOVE 0,1088-C%*64: MOVE 1279,1088-C
%*64
1510 PLOT85,0,1024-C%*64: PLOT85,1279,10
24-C%*64
1520 NEXT C%
1530 COLOUR 0
1540 COLOUR 130
1550 PRINT TAB(0,3)"M I X E D M O D E
S"
1560 COLOUR 134
1570 PRINT TAB(7,11)"mode 2"
1580 PROCmode0
1590 VDU26
1600 VDU5
1610 GCOLOR,1
1620 MOVE 450,490: PRINT "Mode 0. 80 c
olumn text."
1630 VDU4
1640 Y=2
1650 REPEAT
1660 MOVE 0,Y: DRAW 1279,Y
1670 Y=Y*1.1
1680 UNTIL Y>600
1690 ENDPROC

```

ACK ISSUES BACK ISSUES BACK ISSUES BACK ISSUES BACK ISSUES BACK ISSUES BACK ISSUE

We are pleased to announce that back issues of the ELBUG magazine, magazine binders and the ELBUG magazine cassette are now available at a discounted price especially for ELBUG members. All orders to the High Wycombe address please.

|                         |                 |                    |
|-------------------------|-----------------|--------------------|
| ELBUG magazine          | Vol.1 and Vol.2 | 50p per issue      |
| ELBUG magazine binder   | any volume      | £2.50 each         |
| ELBUG magazine cassette | Vol.1 and Vol.2 | £2.00 per cassette |

Postage is at the normal rates - For binders and cassettes: 50p for the first item and 30p for each subsequent item. For magazines: 30p for the first item and 10p for each subsequent item. Complete sets of volume 1 are available at £5.00 including postage. Purchasers of magazine back issues from Vol.2 No.4 onwards will receive both the ELBUG supplement for that issue and the corresponding issue of BEEBUG.