GRAPHIX SOURCE LISTING — NOTES

The most useful routines within GRAPHIX can be accessed via a jump table located at the top of the Rom; it is intended to retain this table in any future versions of GRAPHIX, and to ensure that your own machine code routines are upward compatible you should use the table where possible.

Certain routines terminate on an error by resetting the stack pointer to a value saved on entry. The line drawing routine always uses this technique. More details are given in the following detailed instructions.

GRAPHIX uses several zero-page locations. Refer to the source listing and ensure that your own routines do not make conflicting use of locations.

EXAMINE POINT

This routine tests a point on the high resolution screen, setting the status word (ST) to 0 if the point is reset, or 1 if it is set.

Set FLAG1 ($BB) to $04
Set XPOINT ($93EE/F) to X co-ordinate value (lo/hi)
Set YPOINT ($93F2) to Y co-ordinate value

Enter routine with JSR $EFF7
Location $0096 contains the result

CLEAR/INVERT SCREEN

The same routine will either clear or invert the screen, depending upon the condition of FLAG.

Set FLAG ($FF) to $00 (clear screen) or $FF (invert screen)

Call routine with JSR $EFF4

MAP FILL/ERASE

See GRAPHIX documentation for exact details of this command.

Set FLAG1 ($BB) to $04
Set MAPTYPE ($FB) to $00 (erase) or $FF (fill)
Set XPOINT ($93EE/F) to X co-ordinate value (lo/hi)
Set YPOINT ($93F2) to Y co-ordinate value

Enter with JSR $EFF1

LINE DRAWING COMMANDS

The same routine can be used to set, reset, flip, or dot a line depending upon the value in FLAG1.

Set XPOINT ($93EE/F) to the start X co-ordinate (lo/hi)
Set YPOINT ($93F2) to the start Y co-ordinate
Set XEND ($93EC/D) to the end X co-ordinate (lo/hi)
Set YEND ($93F1) to the end Y co-ordinate
Set FLAG1 ($BB) to 0 (set), 1 (reset), 2 (flip) or 3 (dot)
Set FLAG to 0 (or 1 ONLY for a dotted line to start with a set dot)

Call routine with the following code:

3400

```
TSX              - transfer stack pointer to X register
DEX              - now arrange so that it will return
DEX              - to the instruction after the JSR
STX STACKCOPY    - i.e. $00B4
JSR $EFEB        - not a JMP!
```

An alternative way to set up the return address is via a 'dummy' JSR in your own program, i.e.

```
JSR CALLINE      - JSR to the routine below
.............
TSX              - get stack pointer
STX STACKCOPY    - return address already there
JMP $EFEB        - could be a JSR, but would NOT return here
```

PLOT POINT

One routine allows a point to be set, reset, or flipped.

Set FLAG and FLAG1 as for line drawing
Set XPOINT and YPOINT as for line drawing

Call routine with JSR $EFEE

TEXT

To display text on the high resolution screen a table of characters must exist in memory. $A7F8/9 must point to the start (or deemed start) of the table.

Set TEXTYPE ($BE) to $00 (flip), $40 (overlay), or $ff
Set FLAG1 ($BB) to $04
Set ONEBYTE ($B1) to $00 (normal), $FF (reverse field)
Set PLOTDIR ($BD) to $FF (horizontal), $00 (vertical)
Set XPOINT ($93EE/F) and YPOINT ($93F2) to the co-ordinates of the top left-hand corner of the first character

TO DISPLAY A STRING:
Set TWOBYTE ($B2) to the string length (1 to 255)
Set STRINGPTR ($32/3) to point to the string (lo/hi)
Set STATUSWORD ($96) to 0

Enter routine with JSR $EFFA

TO DISPLAY A SINGLE CHARACTER
Set CHARNUMBER ($BC) to the character position in the table

Enter routine with JSR $EFFD or JSR $EFE5 which automatically updates YPOINT and XPOINT for continuing text either horizontally or vertically

LATCH CONTROL

The latch which controls the high resolution board cannot be read. This routine maintains a copy at $0279.

Load accumulator with the required value (0 to 15)

Enter routine with JSR $EFE2

‹AM ON TEST

This routine tests to see whether the high resolution screen is enabled. If not the status word (ST) is set to -2.

```
TSX                 - transfer stack pointer to X register
STX STACKCOPY       - $00B4
JSR $EFDF           - calls routine
```

Test $96 to discover the result ($FE or $00). Note that FLAG1 is set to $04 by this routine

SCREEN BYTE CALCULATION

This routine converts an X,Y co-ordinate pair into a screen byte, and returns with the Y-register equal to the number of right shifts required to set the bit of %10000000 in the correct position (end when Y=0).

Set FLAG1 ($BB) to $04
Set XPOINT ($93EE/F) to the X co-ordinate value (lo/hi)
Set YPOINT ($93F2) to the Y co-ordinate value

Enter routine with JSR $EFEE - the screen byte is returned in BYTE ($C9/CA)

MISCELLANEOUS ROUTINES

The following entry points may prove useful:

$EFE8 loads the accumulator with 199, and deducts from it the current value of YPOINT

$EFCD decrements the variables YPOINT and BYTE. You must first call $EFE8, then check that the value is below 199 before calling $EFCD

$EFD0 increments YPOINT and BYTE. First call $EFE8, then test for 0 before calling $EFD0

$EFD3 increments XPOINT

$EFD6 decrements XPOINT

* BYTE may need modifying after calling either of the last two routines: two suitable routines are available within GRAPHIX, $EFCA to increment BYTE or $EFC7 to decrement BYTE

$EFDC will store the contents of the accumulator and registers in XEND and YEND: load the X-register with the Y co-ordinate; load A and Y with the low and high bytes respectively of the X co-ordinate

$EFD9 is similar to $EFDC, but stores the registers in XPOINT and YPOINT