GRAPHIX

SUPERSOFT

INTRODUCTION

GRAPHIX software is designed to be used in
conjunction with the SUPERSOFT range of High
Resolution Graphics Boards for the PET. The HR40,
HR40B and HR80 boards offer 320 by 200 resolution, a
total of 64000 dots, each of which can be turned on
or off.

GRAPHIX assumes that the origin (0,0) is the bottom
left-hand corner of the screen. The top right-hand
corner is (319,199).

For convenience the GRAPHIX utilities are contained
in a 4k Eprom which is mounted on the graphics board.
The lower 2k contains part of Commodore Basic:
GRAPHIX itself is approximately 1800 bytes in length.

STARTING OUT

Many machine code utilities work by adding extra
commands to Basic. Although this makes them
'user-friendly' it does mean that any given
combination of products is unlikely to work together.

To avoid problems of incompatibility GRAPHIX works
not through new commands, but via a single SYS
command (SYS 59650). If you start out by setting a
variable to 59650 (e.g. HR=59650) it will make your
programs that much more readable (and quicker to
enter).

Because GRAPHIX commands are accessed in this way
there's no need to turn GRAPHIX on or off. It's
always there when you want it!

GRAPHIX SYNTAX

A typical GRAPHIX command looks like this:

SYS HR,SL,X1,Y1,X2,Y2

SL stands for 'Set Line', which means 'draw a white line'. The start of the line is (X1,Y1) and the end (X2,Y2). Here X1 etc are normal Basic variables, but you could use constants, array variables, or even expressions - as in the next example: ·

SYS HR,DL,10,Y,(X+SIN(X)*3.14),129

DL stands for 'dot line'. Guess what that means. Although the parameters used need not be whole numbers they must be in the range 0 to 319 in the X-axis and 0 to 199 in the Y-axis. Parameters which do not evaluate to integer values are rounded down to the next whole number.

If a parameter falls outside the legal range, but is not less than -32767 or greater than +32767, no great harm results - the command is not executed and the status byte (ST) is set to -1. Outside this wider range an 'ILLEGAL QUANTITY ERROR' will occur.

The status byte is used by most GRAPHIX functions to indicate whether the required operation was successfully completed. The Basic variable 'ST' returns the status value (e.g. PRINT ST or IF ST<>0 THEN...).

GETTING STARTED

The SUPERSOFT board has 8k of its own Ram which
occupies addresses from 36864 to 45055 (hex locations
$9000 to $AFFF). This is the Rom Expansion area
normally represented by sockets UD11 and UD12 (UD3
and UD4 if the machine has a small screen).

When you power-up your computer the Rom sockets are
enabled - and the high resolution memory disabled.
There are a number of ways that you can configure the
system:

	ROM SOCKETS	HI-RES RAM	HI-RES SCREEN	PET SCREEN
SYS HR,0	ON	OFF	OFF	ON
SYS HR,1	OFF	ON	OFF	ON
SYS HR,2	ON	OFF	OFF	OFF
SYS HR,3	OFF	ON	OFF	OFF
SYS HR,5	OFF	ON	ON	ON
SYS HR,7	OFF	ON	ON	OFF

A variable may be used (e.g. SYS HR,XY) provided that
the variable chosen does not clash with a GRAPHIX
command.

These options are controlled by a 'latch' mounted on
the high resolution board. It is possible to write
to the latch directly with a POKE command (the latch
is situated at memory location 35000 on boards marked
(C) 1981 or at 61439 on boards marked (C) 1982).

However, to maintain compatibility between the two
existing models and possible future models you should
use the commands given above. Another more important
reason is that because the latch cannot be read,
GRAPHIX maintains a copy at memory location 633 (hex
$0279). Certain GRAPHIX functions read the
'latchcopy', and may fail if it is incorrect.

When the high resolution screen is ON then the
contents are visible: the same applies to the PET's
screen. As you will see from the table the high
resolution memory must be switched on if it is to be
displayed. The high resolution screen can be read or
written to whether or not it is being displayed -
provided that the Ram is enabled (options 1, 3, 5 or
7): the normal screen can be read or written to at
any time. If option 2 or 3 is selected the computer
screen will appear blank.

If both screens are being displayed (option 5), then
the contents are overlayed - that is to say that a
dot will be displayed at each location where one or
other (or both) of the screens has a dot.

Although for most purposes the two screens are
completely independent there is one area of
interference. Areas of reverse field on the normal
screen will appear as reverse field on the high
resolution screen even if the PET screen is not being
displayed. In fact even if neither screen is being
displayed areas of reverse field will be visible as
solid blocks of white.

USING PLUG-IN CHIPS

If you install chips such as TOOLKIT, COMMAND-O,
ARROW, SUPERCHIP etc in the Rom sockets on the high
resolution board then you will need to disable them
before you use the graphics memory, i.e. before
selecting options 1, 3, 5 or 7. Some chips
incorporate 'kill' commands, but for convenience
GRAPHIX has a special function which resets the
'CHARGET' routine in zero page. SYS HR,O - it's a
letter 'O' not a zero - will unlink added commands
from Basic. Note that functions which operate
through the interrupt (such as auto-repeat) are not
disabled and should be switched off separately using
an appropriate command (e.g. OUT if you are using
COMMAND-O).

Later versions of GRAPHIX (version 17 or later) allow
either 'O' for OFF or 'K' for KILL to be used to
unlink chips. These versions also reset the
interrupt vector, so no other action is necessary by
the user. If you write programs that use high
resolution graphics you would be wise to start your
program with a KILL command.

Remember that chips which use sockets UD3/12 and
UD4/11 must be installed on the graphics board, not
plugged into the Rom sockets on the main board,
otherwise the high resolution memory cannot be used.

The Basic 2 TOOLKIT fits in socket UD5, which does
not clash with the high resolution Ram. You can
however still use SYS HR,O to turn it off - which is
particularly useful because there is no 'kill'
command built-in.

GRAPHIX PLOTTING COMMANDS

FUNCTIONS WHICH AFFECT THE ENTIRE SCREEN

CLEAR SCREEN - SYS HR,C

The entire high resolution screen is blanked.

INVERT SCREEN - SYS HR,I

Each dot on the high resolution screen is 'flipped'.
Black dots become white dots and vice versa.

FUNCTIONS WHICH RELATE TO INDIVIDUAL POINTS

These commands have two parameters. There are no
defaults.

SET POINT - SYS HR,SP,X,Y

A white dot is plotted at (X,Y).

RESET POINT - SYS HR,RP,X,Y

A black dot is plotted at (X,Y).

FLIP POINT - SYS HR,FP,X,Y

The dot at (X,Y) is flipped.

DOT POINT - SYS HR,DP,X,Y

If both X and Y are odd, or both are even, then a
white dot is plotted at (X,Y). Otherwise a black dot
is plotted.

EXAMINE POINT - SYS HR,E,X,Y

The contents of location (X,Y) are examined. If a
black dot is found then ST is set to 0, but if a
white dot is found ST is set to 1.

LINE DRAWING COMMANDS

There are two formats: SYS HR,SL,X1,Y1,X2,Y2 and SYS
HR,SL,X2,Y2. If two sets of parameters are given
then the line is drawn from the point defined by the
first pair to the point defined by the second pair,
e.g. from (X1,Y1) to (X2,Y2) in the first example
above. Where only one point is specified this is
assumed to be the end point, and the line is drawn
from EITHER the last dot specified OR the end of the
last line drawn, whichever is most recent.

SET LINE - SYS HR,SL,X1,Y1,X2,Y2

Draws a continuous white line from (X1,Y1) to (X2,Y2).

RESET LINE - SYS HR,RL,X1,Y1,X2,Y2

Draws a continuous black line from (X1,Y1) to (X2,Y2)
- i.e. all of the points on the line are RESET.

FLIP LINE - SYS HR,FL,X1,Y1,X2,Y2

Each of the points on a continuous line between
(X1,Y1) and (X2,Y2) is flipped.

DOT LINE - SYS HR,DL,X1,Y1,X2,Y2

Points on a continuous line joining (X1,Y1) and
(X2,Y2) are alternately SET and RESET. If X1 and Y1
are both even or both odd then the first point on the
line is set, otherwise it is reset.

FUNCTIONS WHICH RELATE TO AREAS

MAP FILL - SYS HR,MF,X,Y

Subject to certain conditions (see below) this
command would completely fill the area which contains
the point (X,Y). These conditions are as follows:

- the area must be surrounded by a continuous white
  line
- the area must be completely empty (no white dots)
- if the top boundary is not horizontal, then (X,Y)
  must be directly below the highest point
- if the bottom boundary is not horizontal, then
  (X,Y) must be directly above the lowest point
- the shape must not be re-entrant

An explanation of how the MAP FILL routine works may
make matters clearer. The routine looks vertically
above and below the point (X,Y) until a white dot is
found in either direction - these points are assumed
to be part of the top and bottom boundaries. If you
imagine parallel horizontal lines drawn through the
points, then the lines represent boundaries beyond
which filling cannot take place - any part of the
area not between the lines will be left unfilled.
The routine starts at the lower boundary and works
from side to side, one row of dots at a time until
the upper boundary is reached. As it scans from side
to side it stops when it reaches the first white dot
in each direction, so a stray dot or a strangely
shaped boundary would again result in part of the
area being left unfilled.

Another way of looking at it is to visualize a
vertical line drawn through (X,Y) to the upper and
lower boundaries. Now imagine you walk along the
line looking to the left and right. Only those parts

of the area that you can see will be filled in. MAP
FILL and MAP ERASE are most useful when used
interactively, perhaps as part of a drawing program.

MAP ERASE - SYS HR,ME,X,Y

A white area containing the point (X,Y) will be
erased subject to certain conditions - these are
similar to those for MAP FILL (see above), except
that the area should be solid (i.e. free of black
dots). If these conditions are only partly met then
it is likely that only part of the area will be
erased.

Note that the MAP ERASE function erases the boundary
of the area - so that MAP FILL followed by MAP ERASE
can be used to delete certain outlines.

TEXT COMMANDS

INTRODUCTION

Because the low and high resolution screens can be
overlaid (option 5) it will often not be necessary to
display text on the high resolution screen. However,
if non-standard characters are required, or if
standard characters are required to be displayed in
positions that do not conform with the row and column
organisation of the normal screen, then the GRAPHIX
text commands may be used.

Although there is an Eprom which contains a standard
character set mounted on the graphics board, this
character generator is (like the PET's standard
character generator) able to be read only by the
video circuitry. That is, it cannot be read by the

central processor - and hence by the GRAPHIX software.

If you wish to display anything meaningful using the
TEXT (T), TEXT OVERLAY (TO), or INFORMATION FLIP (IF)
commands then there are two necessary preliminaries:

- there must be a table of characters somewhere in
memory that GRAPHIX can refer to. Each character
must be defined by 8 bytes in the table, and there
can be up to 256 characters.

- the location of the first byte of the table must be
specified using the POINTER (P) command.

POINTER - SYS HR,P,36864

The POINTER command must be used before any TEXT
command is issued otherwise random blocks of 8 by 8
dots will be displayed. The character generator that
was removed when the board was fitted can be used if
there is a free Rom socket on the graphics board (or,
in the case of Basic 2 users, if UD5 is available):

Socket used	Memory addresses	Initialise with
UD3 or UD12	36864 to 38911	SYS HR,P,36864
UD4 or UD11	40960 to 43007	SYS HR,P,40960
UD5	45056 to 47103	SYS HR,P,45056

Text may be displayed in NORMAL or REVERSE field, and
text strings can be printed HORIZONTALLY or
VERTICALLY. Any combination of these options may be
used with the T, TO or IF commands. Note that when
text is printed vertically the characters are not

turned on their side, but appear as they do normally.
You could of course design your own sideways
characters.

TEXT - SYS HR,TNH,"This is a message",X,Y
       SYS HR,TRV,112,X*8,20

The syntax of the TEXT command requires the 'T' to be
followed by either an 'N' for Normal or an 'R' for
Reverse field - followed in turn by an 'H' for
Horizontal or a 'V' for Vertical. Any combination is
possible. The text to be printed can be a string as
in the first example above (A$ or MS$(3) or even
D$+"ABCDE" would have been equally acceptable), or
else a single character (again this could be a
variable).

The position of the top left-hand corner of the first
character is set by (X,Y) - these parameters can be
omitted, in which case the last plotted point is
assumed.

When a string is specified (as in the first example
above) GRAPHIX mimics a normal PRINT command. If you
are using a standard character generator the text
that appears on the high resolution screen will look
the same as if you had executed a PRINT statement -
with one or two provisos. First of all you should
remember that there are two character sets in the
PET's character generator: the first of these is the
UPPER CASE/GRAPHICS set which consists of 128
characters in the first 1k of the Rom; the second is
the UPPER/LOWER CASE set in the upper 1k. To access
the second set you must point GRAPHIX in the right
direction by issuing an appropriate 'P' command (i.e.
use an address 1024 bytes greater than for the first
character set).

When you use a string to generate text on the high resolution screen you are limited (as you would be in Basic) to 128 different characters - and their reverse field equivalents. Reverse on or off characters included in a string switch the display mode from normal to reverse field or vice versa; both have the same effect. Other cursor control characters are ignored.

If the specified string is a null string nothing happens, but the status byte is set to -3. The status byte is also set to -3 if the string or character to be printed would go off the right-hand edge of the screen, but any characters that can be displayed in full will be printed.

Another way to generate text on the graphics screen is one character at a time, using a numeric value that specifies a position within the character table. Values from 0 to 255 can be used, which means that there are 256 possible characters available.

TEXT OVERLAY - SYS HR,TORH,"Overlaid text",120,Y2

The TEXT OVERLAY command is identical to the TEXT command in every respect except one - the existing screen contents are not overwritten, but are overlaid by the text.

INFORMATION FLIP - SYS HR,IFNH,"Title",X*10,15

An image of text characters is created by 'flipping' points on the screen. This command is particularly useful because existing information is not destroyed and the contents of the graphics screen are restored by repeating the command.

TECHNICAL NOTES

The high resolution memory occupies locations 36864 to 45055 (hex $9000 to $AFFF). The 8k block is effectively divided into 1k sections which interleave to produce the screen display.

Each byte represents eight dot positions on the screen: the eight dots are organised in a horizontal line with the first (left-hand dot) referenced by the most significant bit of the byte and the far right-hand dot by the least significant bit.

The top row of dots on the screen relate to the contents of memory locations 36864 to 36903 (the first 40 bytes in the first 1k of graphics memory); the second row of dots relate to bytes 37888-37927 (the first 40 bytes in the second 1k block) and so on until the eighth row. The ninth row is represented by bytes 41 to 80 in the first 1k (i.e. 36904 to 36943), the tenth row by bytes 41 to 80 in the second 1k etc etc.

Negative logic is used - a white dot on the screen is generated by a 0 bit and a black dot by a 1 - but the TEXT routines allow for this, so that a standard character table can be used.

The last 24 bytes in each 1k block of Ram are available for use except for those in the first 1k (used by GRAPHIX) and locations 43000/1 ($A7F8/9) in the sixth 1k block which are used to store the character table vector which is created by the POINTER command.

A SOURCE CODE LISTING OF GRAPHIX IS AVAILABLE FROM SUPERSOFT PRICE £10 (NO VAT). INCLUDED WITH THE LISTING ARE INSTRUCTIONS WRITTEN BY THE AUTHOR OF GRAPHIX WHICH DESCRIBE HOW THE VARIOUS ROUTINES CAN BE ACCESSED IN MACHINE CODE.

CORRECTIONS & UPDATES

SUPERSOFT publish regular catalogues of high
resolution and other software. To obtain a
copy of the most recent catalogue or details
of any SUPERSOFT products telephone 01-861 1166