

5 The Graphics VDU.

The Graphics VDU allows individual pixels (dots) on the screen to be set or tested and lines to be drawn. The plotting takes place on an ideal screen that is always 640 points wide and 400 points high. This means that more than one point on the ideal screen will map onto a particular pixel on the real screen. The width of the ideal screen (640 points) is chosen to be the horizontal number of pixels on the screen in the highest resolution mode (mode 2). The height of the ideal screen (400 points) is chosen to be twice the vertical number of pixels on the screen in all modes. This ensures that the aspect ratio of the screen is approximately unity, i.e. a circle looks circular and not elliptical.

5.1 Graphics VDU Coordinate Systems.

The Graphics VDU uses 4 coordinate systems. The user specifies positions in user coordinates or relative coordinates or occasionally in standard coordinates. Internally the Graphics VDU uses base coordinates (or occasionally standard coordinates).

User coordinates, relative coordinates and standard coordinates are all very similar. They all use signed 16 bit numbers and work in points with X-coordinates running left to right and Y-coordinates running bottom to top, The screen is always 400 points high and 640 points wide whatever the screen mode is. This means that a pixel (dot on the screen) is mapped onto by 8 points in mode 0, 4 points in mode 1 and 2 points in mode 2. The origin (coordinate (0,0)) of these systems vary:

In standard coordinates the origin is the point at the bottom left corner of the screen.

The origin of user coordinates can be set by the user. The default origin is at the bottom left corner of the screen. This makes the default user coordinates the same as standard coordinates.

The origin of relative coordinates is the current position (see 5.2 below). This allows plotting to be carried out independently of the position on the screen and is useful if a particular shape is to be repeated on the screen a number of times or if it is inconvenient to keep track of the current location.

Base coordinates are a physical coordinate system which deals with pixels. X-coordinates run left to right and Y-coordinates run bottom to top. Pixel (0,0) is the pixel at the bottom left corner of the screen. Because this coordinate system works in pixels the coordinates of positions on the screen depend upon the screen mode. Base coordinates are unsigned 16 bit numbers and only coordinates that refer to a pixel on the screen are valid.

Graphics routines convert from relative coordinates to user coordinates, if necessary, and then from user coordinates to base coordinates before accessing the physical screen. During the latter conversion there is a loss of accuracy because of the mapping of multiple points onto a single pixel. This could make shapes drawn on the screen appear asymmetrical (particularly circles) but the Graphics VDU avoids this by rounding the coordinates towards the user origin. Thus symmetrical shapes should be drawn symmetrically about the user origin to take advantage of the rounding. If the shape is not centred on the user origin then the asymmetry may reappear.

In V1.1 firmware it is possible to call `GRA FROM USER` to convert from user to base coordinates - this will make using routines in the lower level screen pack easier (e.g. `SCR DOT POSITION`, `SCR HORIZONTAL`, `SCR VERTICAL`).

5.2 The Current Graphics Position.

The Graphics VDU stores a current position. This is the user coordinate of the last point specified to the Graphics VDU (or the origin after clearing the graphics window). The origin of relative coordinates is specified to be at this point, thus relative coordinates are an offset from the current position.

When drawing a line one end is at the position specified while the other end is at the current graphics position. When drawing a character on the screen using the graphics character write routine the character is placed with the current graphics position being the top left corner of the character.

After plotting or testing a point or drawing a line the current graphics position is moved to the position specified. After writing a character the current graphics position is moved right by the width of a character ready for the next character to be written.

5.3 Graphics Pen and Paper Inks.

The Graphics VDU has a pen (foreground) ink and a paper (background) ink. The graphics pen ink is used to plot pixels and to set foreground pixels when writing characters (see 5.6 below) and drawing lines (see 5.7 below). The graphics paper ink is used to clear the graphics window and to set background pixels when writing characters or drawing lines.

The pen and paper can be set to any ink valid in the current screen mode (see section 6.2). The default has the paper set to ink 0 and the pen set to ink 1. Changing the pen or paper ink does not change the screen it merely alters how pixels will be written in the future.

5.4 Graphics Write Mode.

Pixels plotted by the Graphics VDU are plotted using the current graphics write mode. This specifies how the ink to be plotted interacts with the ink a pixel is currently set to.

There are four write modes:

0: FORCE:	NEW	=	INK
1: EXCLUSIVE-OR:	NEW	=	INK xor OLD
2: AND:	NEW	=	INK and OLD
3: OR:	NEW	=	INK or OLD

NEW is the ink that the pixel will be set to.
OLD is the ink that the pixel is currently set to.
INK is the ink that is to be plotted.

The default Graphics write mode is FORCE mode. The Text VDU always sets pixels as if it is operating in this mode. Also the graphics window is cleared by writing in FORCE mode irrespective of the actual write mode.

Provided that suitable ink settings are chosen, AND mode and OR mode allow particular bits in a pixel to be cleared or set. This allows the Graphics VDU to write in 'bit planes' and by choosing the colours of the inks carefully overlapping shapes can be drawn and automatically hidden behind one another.

If the inks are chosen suitably, EXCLUSIVE-OR mode can be used to plot over the current contents of the screen. It is also useful because a shape can be removed from the screen by redrawing it since exclusive-oring with the same ink twice restores the original setting of a pixel.

The graphics write mode may be set by calling SCR ACCESS or by using a control code (see Appendix VII).

5.5 Graphics Window.

The Graphics VDU allows a single window to be specified. This allows the user to mix text and graphics on the screen without them interfering with each other. If the text windows are allowed to overlap the graphics window then the contents of the graphics window will be moved when the text windows are rolled. The graphics window cannot be rolled.

When plotting points, drawing lines or writing characters no pixel outside the graphics window is ever written. Unlike the text windows no action is taken to force a point inside the window - actions outside the window will be lost. Conversely, when testing points, points outside the window are all deemed to be set to the current graphics paper ink. Points inside the window are written and read as expected.

The graphics window is related to a specific area of the screen and not to the user coordinate system. Thus, changing the origin of the user coordinate system will not move the location of the window on the screen although it does change the user coordinates of points in the window.

The default graphics window, set at EMS and after changing screen mode, covers the whole of the screen.

5.6 Writing Characters.

The Graphics VDU write character routine draws a character with the current graphics position at the top left corner of the character. The current position is moved right by the width of a character in the current screen mode. The distance moved varies; in mode 0 it is 32 points; in mode 1, 16 points; and in mode 2, 8 points. Control codes, characters 0..31, are printed and are not obeyed.

In V1.0 firmware the character is always written opaquely irrespective of what mode the Text VDU is using to write characters, i.e. The character background is set to the graphics paper ink and the foreground is set to the graphics pen ink. However, the current graphics write mode is used to plot the pixels in the character (see 5.4 above).

In V1.1 firmware the foreground to the character is always written in the graphics pen ink using the current graphics write mode. How the background to the character is written depends on the background write mode set by calling GRA SET BACK. If the background mode is opaque then the background to the character is written in the graphics paper ink using the current graphics write mode. If the background mode is transparent then the background pixels are not plotted at all, the current settings of these pixels are left unchanged.

5.7 Drawing Lines (Only applicable to V 1. 1 firmware)

The Graphics VDU has a number of options that affect the way that lines are drawn on the screen. These include the background write mode, the first point mode, the line mask and the graphics write mode.

The line mask is an 8 bit, bit significant value that specifies whether pixels on the line are foreground or background. If the bit of the mask corresponding to a pixel is one then the pixel is foreground. If the bit is zero then the pixel is background. The mask is used repeatedly along the length of the line. i.e. Bit 7 of the mask corresponds to pixels 1, 9, 17,... of the line, bit 6 to pixels 2, 10, 18,... etc.

Foreground pixels of a line are plotted in the graphics pen ink using the graphics write mode. How background pixels are plotted depends upon the background write mode. If the background mode is opaque then the background pixels are plotted in the graphics paper ink using the graphics write mode. If the background mode is transparent then the pixels are not plotted at all, the current settings of the pixels are left unchanged.

The first point mode specifies whether the pixel at the current graphics position is to be included in the line or not. Not plotting the first pixel of a line is particularly useful for drawing lines when the graphics write mode is XOR. For instance, the corner pixels of a box will be plotted twice if the first pixels of lines are plotted and this will result in these pixels not being set in XOR mode.

5.8 Filling Areas (Only applicable to V1. 1 firmware)

The Graphics VDU provides a generalised area fill routine. The user draws the outline of the area to be filled using the standard line drawing and pixel plotting routines and then moves the current position to any pixel inside the area and calls the fill routine, GRA FILL. This will set all pixels inside the delimited area to the fill ink irrespective of what the current graphics write mode is set to. (i.e. GRA FILL always works as if FORCE mode was selected).

The fill routine recognises pixels making up the edge of the area by the ink that they are set to. All pixels set to either the current graphics pen ink or the ink that is being used to fill the area with are treated as edge pixels. (These two inks may be the same). Edge pixels need only lie diagonally adjacent to each other, it is not necessary for them to be orthogonally adjacent. Of course, the edges of the graphics window are also treated as edges of the area to fill and so no pixels outside the graphics window will be affected.

GRA FILL uses a buffer supplied by the user for storing information. The more complicated the area the longer the buffer will need be. If the supplied buffer is too short then parts of the area will be ignored and will not be filled. By supplying a long enough buffer any arbitrarily complicated shape may be filled.

