

**NAME**

vt – 11/20 (vt01) interface

**DESCRIPTION**

The file *vt0* provides the interface to a PDP 11/20 which runs a VT01A-controlled Tektronix 611 storage display. The inter-computer interface is a pair of DR-11C word interfaces.

Although the display has essentially only two commands, namely “erase screen” and “display point”, the 11/20 program will draw points, lines, and arcs, and print text on the screen. The 11/20 can also type information on the attached 33 TTY.

This special file operates in two basic modes. If the first byte written of the file cannot be interpreted as one of the codes discussed below, the rest of the transmitted information is assumed to ASCII and written on the screen. The screen has 33 lines (1/2 a standard page). The file simulates a 37 TTY: the control characters NL, CR, BS, and TAB are interpreted correctly. It also interprets the usual escape sequences for forward and reverse half-line motion and for full-line reverse. Greek is not available yet. Normally, when the screen is full (i.e. the 34th line is started) the screen is erased before starting a new page. To allow perusal of the displayed text, it is usual to assert bit 0 of the console switches. This causes the program to pause before erasing until this bit is lowered.

If the first byte written is recognizable, the display runs in graphic mode. In this case bytes written on the file are interpreted as display commands. Each command consists of a single byte usually followed by parameter bytes. Often the parameter bytes represent points in the plotting area. Each point coordinate consists of 2 bytes interpreted as a 2's complement 16-bit number. The plotting area itself measures  $(\pm 03777) \times (\pm 03777)$  (numbers in octal); that is, 12 bits of precision. Attempts to plot points outside the screen limits are ignored.

The graphic commands follow.

order (1); 1 parameter byte

The parameter indicates a subcommand, possibly followed by subparameter bytes, as follows:

erase (1)

The screen is erased. The program will wait until bit 0 of the console switches is down.

label (3); several subparameter bytes

The following bytes up to a null byte are printed as ASCII text on the screen. The origin of the text is the last previous point plotted; or the upper left hand of the screen if there were none.

point (2); 4 parameter bytes

The 4 parameter bytes are taken as a pair of coordinates representing a point to be plotted.

line (3); 8 parameter bytes

The parameter bytes are taken as 2 pairs of coordinates representing the ends of a line segment which is plotted. Only the portion lying within the screen is displayed.

frame (4); 1 parameter byte

The parameter byte is taken as a number of sixtieths of a second; an externally-available lead is asserted for that time. Typically the lead is connected to an automatic camera which advances its film and opens the shutter for the specified time.

circle (5); 6 parameter bytes

The parameter bytes are taken as a coordinate pair representing the origin, and a word representing the radius of a circle. That portion of the circle which lies within the screen is plotted.

arc (6); 12 parameter bytes

The first 4 parameter bytes are taken to be a coordinate-pair representing the center of a circle. The next 4 represent a coordinate-pair specifying a point on this circle.

The last 4 should represent another point on the circle. An arc is drawn counter-clockwise from the first circle point to the second. If the two points are the same, the whole circle is drawn. For the second point, only the smaller in magnitude of its two coordinates is significant; the other is used only to find the quadrant of the end of the arc. In any event only points within the screen limits are plotted.

dot-line (7); at least 6 parameter bytes

The first 4 parameter bytes are taken as a coordinate-pair representing the origin of a dot-line. The next byte is taken as a signed x-increment. The next byte is an unsigned word-count, with '0' meaning '256'. The indicated number of words is picked up. For each bit in each word a point is plotted which is visible if the bit is '1', invisible if not. High-order bits are plotted first. Each successive point (or non-point) is offset rightward by the given x-increment.

Asserting bit 3 of the console switches causes the display processor to throw away everything written on it. This sometimes helps if the display seems to be hung up.

**FILES**

/dev/vt0

**BUGS**