Laser-Scan Ltd.

LITES2

Tektronix 4100/4200 series and Westward 2300/3300 series Workstation Guide

Issue 1.4 - 16-August-1989

Science Park, Milton Road, Cambridge, England CB4 4FY tel: (0223) 420414

Document "LITES2 - Tektronix 4100/4200 Westward 2300/3300" Category "USER"

Document Issue 1.0	Clarke Brunt	12-May-1986
Document Issue 1.1	Ron Russell	19-Jun-1987
Document Issue 1.2	Ron Russell	30-Sep-1987
Document Issue 1.3	Clarke Brunt	27-Oct-1988
Document Issue 1.4	Ron Russell	16-Aug-1989

1 *Introduction*

This document describes the workstation dependent facilities available in the version of LITES2 for Tektronix 4100/4200 and Westward 2300/3300 series displays (image LITES2TEK4100.EXE). It is to be read as a supplement to the LITES2 Reference Manual and the LITES2 User's Guide.

2 *Display*

A single display is supported (both GRAPHICS and PRIMARY must be ENABLEd). This may be any of Tektronix 4107, 4109, 4111, 4115, or 4125, or the corresponding 4200 displays, or Westward 2215B, 2315, 2216, 2316, 2219, 2319, 2220, 2320, 3219, 3319, 3220, or 3320. The Westward displays must be fitted with the Westward Graphics Manager which is an extra option on some of the models. The display is accessed on logical name LSL$TK, which must be set up to the terminal line. If this is not the user's login terminal, then the line's protection must be set to allow access to it.

It is possible to log in on the graphics terminal, but normally a second terminal is used, in order to avoid having alphanumerics and graphics on the same screen. If a Westward bitpad is used, then it is not possible to see what you are typing on the Westward keyboard.

If the graphics terminal has sufficient local memory, then ENABLE SEGMENTS may be used, to allow local storage of the picture, and rapid redraws. If SEGMENTS is not enabled, then all redrawing is performed from the host computer.

It is possible to ENABLE SEGMENTS after maps have been read in. In this case, at the next redraw, the contents of local memory will be deleted and the next picture will be drawn from the host computer into the local memory. It is thus possible to have a subset of the map in local memory. This subset may be the result of selections in force at the redraw, and/or of a windowing or zoom command that has displayed only part of the data.
If the command SUPPRESS PRIMARY is given in INITIAL state, then when the map is read in, it will not be displayed (thus saving time) and the first draw command will be done by the host computer drawing into local memory.

By default, 12 bit coordinates are used to draw on Tektronix devices. This means that zooming the picture more than approximately 4 times results in the picture quality deteriorating. Some devices, however (eg 4111, 4115 and 4125), will allow 32 bit coordinates to be used, and this facility can be accessed in LITES2 by giving the WORKSTATION TYPE 1 32 command while in INITIAL state. In this case LITES2 transmits coordinates with 16 bit precision, and zoom factors up to at least 64 may be used with no loss of picture quality.

If the graphics terminal is a Westward, then the command WORKSTATION TYPE 1 1000 must be given while in INITIAL state. 15 bit coordinates are always used, so zoom factors up to at least 32 may be used with no loss of picture quality.

The LITES2 cursor, and all highlighting of found features etc., is drawn as highlighted segments using colour 15 in exclusive or mode. This will result in features blinking between their original colour and some other colour (colour 0 will highlight in colour 15, 1 in 14, 2 in 13 etc). The colour table should be arranged in such a way that the highlighted colours are distinct from the original colour. The cursor may be small or large, and blinking or steady

(ENABLE/DISABLE BIG/BLINK).

The colours used for the picture are defined using a text file on logical name LSL$TEK_COLOUR (if this is set up). See Appendix for the format of this file. Colours may be changed subsequently using the WORKSTATION COLOUR command.

3 *Hardware dependant commands*

Display overlays (OVERLAY command), multiple displays (DISPLAY command), raster backdrop (IMAGE command) or perspective viewing (VIEW command) are not supported in this version of LITES2.

The only FOLLOW subcommand that is available is FOLLOW TABLE, when the table is connected by a seperate serial line (see below).

4 *Interactive devices*

In addition to the login keyboard, this version of LITES2 is capable of interpreting commands from the graphics terminal keyboard (if this is different), a digitising table on a separate terminal line, a Tektronix or Westward bitpad, and a Tektronix mouse. It is also possible to use a screen menu if a bitpad or mouse is being used.

The digitising table input is interpreted either using the Table Monitor system, or by reading the table directly. The former allows the table to be set in stream mode, giving smooth cursor tracking.

To use the Table Monitor, a table monitor process must be started, using program STARTMON. If the 'named monitor' option is used, then logical name LSL$MONITOR_TABLE must point to the serial line. In addition, if the table is anything other than a standard ALTEK, then logical table LSL$TABMON_ROUTINE (or LSL$TABMON_ROUTINE_<terminal> for named monitor) must point to a suitable decoding shareable image. This logical name must be available to the table monitor process, and so should be in the group or system logical name table. If stream mode is used, to allow smooth tracking using the lowest numbered button, then the lowest acceptable stream rate above 4 points per second should be used. If set too high, then the table monitor will use large quantities of system resources, if too low, then buttons other than the 'tracking button' will repeat if held down.

If logical name LSL$MONITOR_TABLE is set up, but LITES2 determines that no table monitor process exists, the table will be accessed directly. This does not allow stream mode or smooth tracking.

If a bitpad or mouse is available, then commands ENABLE BITPAD and ENABLE BALL respectively should be used. If either of these is enabled, then the GIN cursor (full screen cross) will appear and may be moved around by moving either of the devices. The LITES2 cursor is moved to the position of the GIN cursor by pressing the lowest numbered button on the appropriate puck. The different pucks may be programmed after defining them using PUCK commands. The bitpad is device 2, the table device 3, and the mouse device 4. On a Westward, the GIN cursor may be toggled between a full screen cross and a small "Maltese" cross by pressing the Maltese Cross key on the keyboard.

If a menu is used on the bitpad, then attention should be drawn to the PRIORITY PUCK command. The lowest numbered button is always used for cursor tracking, so for the 4 button puck, 2 other buttons may be given priority so that their puck function is obeyed even if they are pressed over the menu. At least one button must not be given priority, otherwise it will be impossible to access the menu.

Whilst it is possible to set up a tracking area on the bitpad, this is not in general useful, since the default action is to map the whole bitpad onto the current window, using the lowest numbered button. If a tracking area is used, then the position to which the LITES2 cursor moves will not be the same as the GIN cursor, which can be confusing.

4.1 *Screen Menu*

If the bitpad or mouse has been enabled, a screen menu can be displayed on the screen, and accessed by moving the GIN cursor with the bitpad puck or the mouse over one of the individual boxes. This box can then be selected by pressing button 1.

Before a screen menu can be displayed on the screen, it must be defined as a PUCK on device 1 (the screen). This also defines the total number of boxes in the menu. The boxes of the screen menu can be programmed using MACRO commands as usual.

The size, layout, screen position and title of the screen menu are defined by the DESCRIBE SCREENMENU command. The menu is drawn in the current annotation colour.

On a Tektronix, the GIN cursor is constrained to lie within the map viewport (which may be smaller than the screen, for a non-screen shaped map) it is possible to have a screen menu that cannot be accessed; this situation is most often met when a whole (square) map is displayed on the screen, and the menu is on the right hand edge of the screen. In this case, as the user zooms in, the menu becomes available. Note that it is possible, through the use of the WORKSTATION VIEWPORT and DESCRIBE SCREENMENU commands to position a screen menu such that it can never be accessed by the GIN cursor! On the Westward, the bitpad always maps onto the whole screen, so this is not a problem.

The titling of the boxes is achieved by the DESCRIBE MACRO command. The height of the letters used is chosen so that they fill the height of the menu boxes, and the text strings are truncated, so that they do not overflow the sides of the boxes.

The menu is displayed on the screen by the ENABLE SCREENMENU command. If this command is given in INITIAL state, then a screen menu window will be created just after the initial draw has been completed.

When a map has been read in, the existence of the screen menu can be controlled by the ENABLE, DISABLE and TOGGLE SCREENMENU commands.

Note that it is possible to construct or edit a feature so that it lies over the menu. These features will go behind the menu after the next redraw.

An example of a file to set up a screen menu is given in the appendix.

APPENDIX A

Colour Table

The following is an example of a file describing the colours to be used. It should be set up on logical name LSL$TEK_COLOUR if the colours are to be set. The character ';' introduces a comment. The colours are specified as proportions of red, green, and blue, in hexadecimal in the range 0-FF. An example file is in LSL$LITES2CMD:TEK.COL

```
; Colour file for TEK 4100
; COLOUR DEFINITIONS
;       Red     Green   Blue
16                                      ; number of colours
         0       0       0
         0       0      FF
        FF       0       0
        FF       0      FF
         0      68       0
         0      80      FF
        FF      97       0
        FF      86      FF
         0      AF       0
         0      BE      FF
        FF      C0       0
        FF      B3      FF
         0      FF       0
         0      FF      FF
        FF      FF       0
        FF      FF      FF
```

APPENDIX B

Screen menu

The following is an example of a file that sets up a screen menu. It should be called LSL$LITES2CMD:SCREEN.LCM

```
! SCREEN.LCM
!
! Definition of screen menu
! ========================
!
! first define puck
!
%PUCK 1 32 SCREEN
!
! and describe what SCREEN is to look like
!
%DESCRIBE SCREEN 1 32 6 25.0 200.0 1.0 0.0 screen LSL
!
! now define the contents of each button (box) of SCREEN
!
%MACRO SCREEN2  #%START         #%ENDM
%MACRO SCREEN3  #%START#%END    #%ENDM
%MACRO SCREEN4  #%CURVE         #%ENDM
%MACRO SCREEN5  #%FIND          #%ENDM
%MACRO SCREEN6  #%END           #%ENDM
%MACRO SCREEN7  #%CLOSE         #%ENDM
%MACRO SCREEN8  #%CLOSE SQUARE  #%ENDM
%MACRO SCREEN9  #%MOVE          #%ENDM
%MACRO SCREEN10 #%ROTATE        #%ENDM
%MACRO SCREEN11 #%GET 3         #%ENDM
%MACRO SCREEN12 #%GET 4         #%ENDM
%MACRO SCREEN13 #%INVISIBLE     #%ENDM
%MACRO SCREEN14 #%REPEAT        #%ENDM
%MACRO SCREEN15 #%WINDOW MAP#%PING      #%ENDM
%MACRO SCREEN16 #%ABANDON       #%ENDM
%MACRO SCREEN17 #%SELECT ALL    #%ENDM
%MACRO SCREEN18 #%ZOOM          #%ENDM
%MACRO SCREEN19 #%ZOOM .5       #%ENDM
%MACRO SCREEN20 #%DRAW SCREEN   #%ENDM
%MACRO SCREEN21 #%DRAW MAP      #%ENDM
%MACRO SCREEN22 #%DELETE        #%ENDM
%MACRO SCREEN23 #%EDIT          #%ENDM
%MACRO SCREEN24 #%REMOVE        #%ENDM
%MACRO SCREEN25 #%BRIDGE        #%ENDM
```

```
%MACRO SCREEN26 #%SPLIT         #%ENDM
%MACRO SCREEN27 #%USER 1        #%ENDM
%MACRO SCREEN28 #%USER 2        #%ENDM
!%MACRO SCREEN29        #               #%ENDM
!%MACRO SCREEN30        #               #%ENDM
!%MACRO SCREEN31        #               #%ENDM
%MACRO SCREEN32 #%ABANDON       #%ENDM
!
! now what is to be written in each box
!
%DESCRIBE MACRO SCREEN2 Start
%DESCRIBE MACRO SCREEN3 Symbol
%DESCRIBE MACRO SCREEN4 Curve
%DESCRIBE MACRO SCREEN5 Find
%DESCRIBE MACRO SCREEN6 End
%DESCRIBE MACRO SCREEN7 Close
%DESCRIBE MACRO SCREEN8 Close Squ
%DESCRIBE MACRO SCREEN9 Move
%DESCRIBE MACRO SCREEN10 Rotate
%DESCRIBE MACRO SCREEN11 Get 3
%DESCRIBE MACRO SCREEN12 Get 4
%DESCRIBE MACRO SCREEN13 Invisible
%DESCRIBE MACRO SCREEN14 Repeat
%DESCRIBE MACRO SCREEN15 Window
%DESCRIBE MACRO SCREEN16 Abandon
%DESCRIBE MACRO SCREEN17 Sel all
%DESCRIBE MACRO SCREEN18 Zoom
%DESCRIBE MACRO SCREEN19 Zoom .5
%DESCRIBE MACRO SCREEN20 Draw screen
%DESCRIBE MACRO SCREEN21 Draw map
%DESCRIBE MACRO SCREEN22 Delete
%DESCRIBE MACRO SCREEN23 Edit
%DESCRIBE MACRO SCREEN24 Remove
%DESCRIBE MACRO SCREEN25 Bridge
%DESCRIBE MACRO SCREEN26 Split
%DESCRIBE MACRO SCREEN27 User 1
%DESCRIBE MACRO SCREEN28 User 2
!%DESCRIBE MACRO SCREEN29 MacroA
!%DESCRIBE MACRO SCREEN30 MacroB
!%DESCRIBE MACRO SCREEN32 MacroC
%DESCRIBE MACRO SCREEN32 Abandon
```