Laser-Scan Ltd.

LITES2

Sigmex 6100/6200 series Workstation Guide

Issue 1.8 - 16-August-1989

1 *Introduction*

This document describes the workstation dependent facilities available in the
version   of   LITES2   for   Sigmex   6100 and 6200 series displays (image
LITES2SIG6000.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 the Sigmex 6100 or 6200 range. The connection to the computer may
be by serial or parallel link. For serial operation, 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. Parallel mode will be used automatically if logical name WKSPLL
is assigned to a valid parallel interface device (this will probably be
something like XAA0: on a MicroVAX or IDA0: on a VAX).

It is possible to log in on the Sigmex, but a second terminal is often used, in
order to avoid having alphanumerics and graphics on the same screen.

If the Sigmex 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. When segments
are enabled, then the sort options (ENABLE SORT, SORT IFF/FSN) will have no
effect (and should normally be disbled). Features will be re-drawn from the
segment store in the order graphical type 12 (areas) first, graphical type 1
(lines) last.

It is possible to use ENABLE SEGMENTS after a map has 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.

If ENABLE HWTEXT is used, then the text fonts contained in the TRI file will be
loaded into the display (after the "Initialising workstation" message, but
before the prompt to set up maps etc.). The loading can take quite a while, but
the advantage is that text is drawn very quickly subsequently. The setting of
HWTEXT should not be altered except while in INITIAL state.

All highlighting of found features etc. is performed using the workplane. This
will normally result in features blinking between their original colour and the
workplane colour. If the bitpad is not in use, then the Sigmex cursor is used
as the LITES2 cursor - this appears in white. If the bitpad is in use, then the
Sigmex cursor (white, small, steady) is used by the bitpad, while the LITES2
cursor is drawn in colour 15 (if steady), or the workplane colour (if blinking).
The LITES2 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$SIGMA_COLOUR (if this is set up). See Appendix for the format of this file.
Colours may be changed subsequently using the WORKSTATION COLOUR command.   The
command WORKSTATION COLOUR -1 r g b sets the colour of the workplane, used for
highlighting features, while colour -2 also sets the workplane colour, but
non-blinking.   Colour 0 is the background, while positive colour numbers
correspond to the colour index of features in the FRT file.

3 *Hardware dependant commands*

Display overlays (OVERLAY command) are not currently supported in this version
of LITES2, but the display has the basic facilties required, and overlays may
eventually be provided.

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, and then only 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 a digitising table on a separate terminal line, and a
Sigmex bitpad. If a Laser-Scan MUART controller is used, then a digitising
table may be attached to the same serial line as the Sigmex display. It is also
possible to use a screen menu, if a Sigmex bitpad is being used.

4.1 *Digitising table*

If MONITOR is enabled, in addition to TABLE, then 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.   If MONITOR is not enabled, then it is assumed that the
digitising table is connected via a Laser-Scan MUART controller. In this case
the Laser-Scan WOSP workstation control microprocessor program must previously
have been loaded.

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.

4.2 *Bitpad*

If a Sigmex bitpad is available, then the command ENABLE BITPAD may be used.
When the bitpad is is use, a small white cursor cross will appear, in addition
to the LITES2 cursor, and may be moved around by moving the bitpad puck.   The
LITES2 cursor is moved to the position of the bitpad cursor by pressing button 1
on the puck. The different pucks may be programmed after defining them using
PUCK commands. The bitpad is device 2, and the table device 3.

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
bitpad cursor, which can be confusing.

4.3 *Keyboard*

If the Sigmex is used as the login terminal, and either the bitpad or the MUART
table are used, then it is impossible to use the Sigmex keyboard in the normal
fashion (with echo character by character on the alpha plane).   Under these
circumstances, keyboard input is echoed in the workplane in the lower left
corner of the screen. When an input line is terminated by carriage return, the
line is echoed on the alpha plane before being obeyed. The normal VAX control
sequences (e.g. CTRL/S, CTRL/Q, CTRL/T, CTRL/C, CTRL/Y, CTRL/Z) may not be used
while an input line is being prompted for, but may be used between commands, for
instance to hold output, or to interrupt a macro. If the bitpad is in use, then
CTRL/C may not be used to abort setup while in this mode of operation. Setup
must be completed, or retained/aborted using puck buttons.

4.4 *Screen Menu*

If the Sigmex bitpad has been enabled, a screen menu can be displayed on the
screen, and accessed by moving the white cross with the bitpad puck 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.

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 graphics window appears.
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. The menu
can be separated from the map, by suitable combinations of the WORKSTATION
VIEWPORT and DESCRIBE SCREENMENU commands.

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

5 *General hints*

If there appears to be no response from the Sigmex, then first try pressing
carriage return, or CTRL/C.   If it is impossible to obtain a response, then
Shift/f15 on the Sigmex keyboard will reset the display. If this is done during
editing, it will not be possible to redraw the maps, but the data in the IFF
file will not be affected. You may still EXIT normally. If the Sigmex hangs
while using it as the login terminal, it will be necessary to terminate LITES2
by CTRL/Y after a reset, and to recover the editing by the usual means.

The f1 light on the Sigmex keyboard should not be illuminated.   It is turned
on/off by Shift/f1, and if on, indicates diagnostic mode which drastically slows
down operation.

6 *Sigmex setup*

The following parameters should be set on the Sigmex setup menu (and permanently
saved using Shift/S).

Setup B
     COMP SPEED 9600 for MUART version, otherwise as required
     DELAY $00
     PROMPT CHAR $FF
     O/P TERM 1 $3E Note, this is not the default
     O/P TERM 2 $0D
     O/P PREC $0B Note, this is not the default

Options
     SERIAL PROTOCOL   7-bit
     PARALLEL PROTOCOL NONE (for serial) or 16-bit (for parallel)
     TABLET $01 (for Summagraphics 3 button tablet)
                        $41 (for Summagraphics 4 button tablet)

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$SIGMA_COLOUR if the colours are to be set.
The character ';' introduces a comment.   The colours are specified   as
proportions of red, green, and blue, in hexadecimal in the range 0-FF. It is
possible to put the numbers in decimal by preceding each with the characters ^D.
After defining the specified number of colours, if another line is present, this
defines the workplane colour (for highlighting, default is red). A zero in a
fourth column on this line will turn off blinking of the workplane. An example
file is in LSL$PUBLIC_ROOT:[LITES2.EXAMPLES]SIGMA.COL

```
; Colour file for Sigmex 6100
; 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
;
         0       FF        0                     ; workplane colour
(or      0       FF        0        0            ; non-blinking workplane colour)
```

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
```