Laser-Scan Ltd.

LITES2

HRD Workstation Guide

Issue 1.3 - 16-Aug-1989

Copyright (C) 2019 Laser-Scan Ltd
Science Park, Milton Road, Cambridge, England CB4 4FY tel: (0223) 420414

Document "LITES2 - HRD Workstation"		Category "USER
Document Issue 1.0	R.W. Russell	29-Jun-1987
Document Issue 1.1	Paul Hardy	29-Dec-1987
Document Issue 1.2	Clarke Brunt	28-Oct-1988
Document Issue 1.3	Ron Russell	16-Aug-1989

1 *Introduction*

This document describes the workstation dependent facilities available in the version of LITES2 for HRD type displays (image LITES2HRD.EXE). It is to be read as a supplement to the LITES2 Reference Manual and the LITES2 User's Guide. The user should also refer to the "VAX LDLIB and HR Driver Reference" manual for more information about the HRD-1 and its interface to VAX computers.

2 *Display*

This version of LITES2 requires a Laser-Scan HRD-1, HRD-1S, FASTRAK, or LASERTRAK display (generically referred to as HRD displays). A single graphics display is supported (both GRAPHICS and PRIMARY must be ENABLEd). The HRD to be used is addressed via the logical name SYS$HRD (which is normally set up system wide).

ENABLE SEGMENTS is not supported, so all redrawing is performed from the source IFF file. Plotting to Diazo film is not supported, use LAMPS software package PLOTTING option HRD instead.

The LITES2 program may be run from any VDU terminal, but on a LASERTRAK, it will normally be run from the Tek 4205 closeup screen terminal.

The refresh mode of the HRD is used for display of the interactive cursor and highlighting of found items etc.

The HRD display does not support display overlays (DISPLAY and OVERLAY commands), raster backdrop (IMAGE comamnd), perspective viewing (VIEW command) or stream digitising (FOLLOW command).

3 *Interactive devices*

In addition to the keyboard, this version of LITES2 is capable of interpreting commands from the HRD button box and tracker ball, and optionally a digitising table on a separate terminal line.

The digitising table is activated by the commands ENABLE TABLE and ENABLE MONITOR (both are required). The table puck may be programmed using the PUCK command on device 3. 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.

The tracker ball is activated automatically, and the LITES2 cursor will follow the ball movement without need for any button presses.

Before the buttons can be used, they must be defined as a PUCK on device 4. This defines the number of buttons that are to be used. The individual buttons can be programmed using MACRO commands as usual. A typical command file to do this is attached as an appendix.

APPENDIX A

Button Puck

The following is an example of a file that sets up the HRD buttons as a puck. It should be called LSL$LITES2CMD:HRDBUT.LCM

```
!
! Definition of HRD function buttons
! ==================================
!
MESSAGE  ***  Reading HRD function button macros  ***
!
! first define puck
!
PUCK 4 16 FB
!
! now define the contents of each FB
!
MACRO FB1          FIND            ENDMACRO
MACRO FB2          START           ENDMACRO
MACRO FB3          END             ENDMACRO
MACRO FB4          ABANDON         ENDMACRO
MACRO FB5          WINDOW MAP      ENDMACRO
MACRO FB6          START  END      ENDMACRO
MACRO FB7          CURVE           ENDMACRO
MACRO FB8          REMOVE          ENDMACRO
MACRO FB9          PREVIOUS        ENDMACRO
MACRO FB10         NEXT            ENDMACRO
MACRO FB11         INVISIBLE       ENDMACRO
MACRO FB12         CONTINUE        ENDMACRO
MACRO FB13         GET 1           ENDMACRO
MACRO FB14         GET 2           ENDMACRO
MACRO FB15         GET 3           ENDMACRO
MACRO FB16         ABANDON         ENDMACRO
```