Operating Manual

SATELLITE LINK EMULATOR SLE900

RF Test Equipment for Wireless Communications

Confidentially
This manual contains proprietary information of dBmCorp, Inc. It is provided under confidential custody for the sole purpose of specification, installation, maintenance and operation of dBm test system instruments and equipment and may not be used or disclosed to any person for any other purpose whatsoever. No portion of this manual or the information it contains may be reproduced, used, or disclosed to individuals not having a need to know consistent with its intended purpose, without the written permission of dBmCorp, Inc.

Disclaimer
dBmCorp, Inc. makes no representation or warranties with respect to the contents hereof and specifically disclaims any implied warranties of merchantability for any particular purpose.

dBmCorp, Inc. reserves the right to revise this publication and to make changes from time to time in the content thereof without obligation of dBmCorp, Inc. to notify any person of such revision or change.

MAN2001322 Rev 1.3 07/2011 Printed in U.S.A

Contents

PREFACE **7**

Conventions Used in This Manual **8**

Contacting dBm **9**

INTRODUCTION **11**

General Information **12**
Front panel view 12
Rear Panel Connections 13
Instrument Modules 14

Start and Shutdown Procedures **15**
Starting the SLE900 15
Shutting Down the SLE900 15

LOCAL OPERATION OVERVIEW **17**

Operating States **18**
Power up and Reset 18
Static 18
Dynamic 18

Viewing the Instrument's Hardware Configuration **19**

Editing Parameters **19**
Editing in Static Mode 19
Use of the Clear Key 20
Set Delay 20
Set Frequency Offset 20
Set Attenuation 20
Set Phase Offset 20
Set L-Band Center Frequency 21
Set Step Size (Delay, Freq Offset, Attn, Phase Offset) 21
↑ and ↓ (up arrow and down arrow) 21
Change the sign of a value 21

Utility Functions **21**
To store the instrument settings 22
To recall a saved instrument setting 22

To configure the instrument's LAN port 23
To set the instrument to local or remote (LAN) control 23
To preset the instrument settings 23

Static Delay Operation 24

Dynamic Delay Operation 24
Dynamic Modes 25
Dynamic File Type names 26

RF Center Frequency Operation 26

Additive White Gaussian Noise Operation (Optional) 27

Multipath Fading Operation (Optional) 28

LED Functions 28

REMOTE OPERATION OVERVIEW 31

Setting the SLE900 IP Address 33
Remote Control via LAN 33

Editing in Dynamic Mode 33
Change Update Rate 33
File Menu 33
Scroll Down 34
Scroll Up 34
Mode 34
Update 34
Loop 34
Trigger 35
Time Reference 35
Start 35
Reset 35
Pause 36

LAN File Formats 36

Data File Description 36
Delay Files 36
Phase Files 37
Attenuation Files 37
Frequency Offset Files 37
Data File sizes 37

ASCII file format 37

Description of the SLE900 Remote Client Application Program *SLEControl*..... 38
Installing *SLEControl* on a PC 38
Connecting to the SLE900 39
Static Controls 39
Dynamic Controls 40
Downloading Files into the SLE900 Memory 41
File Selection 42
Deleting Files from the SLE900 Memory 44
Starting a Dynamic Run 44
Graphical File Display 44

INSTALLATION AND TROUBLESHOOTING 47

Installation 48
Unpacking the SLE900 48
Initial Inspection 48
Applying power 48

System verification 49
Required Equipment 49
Passband Flatness testing 49
Attenuation testing 50
Frequency Offset testing 50
Delay testing 50
Phase Offset Testing 51

Error Messages 52
File Missing 52
Limit 52

DESCRIPTION AND SPECIFICATIONS 53

Functional Description 54
Delay Line Functions 54
Delay Slew Rate and Resolution Limits 54

Specifications 55

Specifications (cont'd) 56

MAINTENANCE AND WARRANTY 58

Maintenance Information 59
Adjustments and Calibration 59

Repair 59
Equipment Returns 59

Warranty Information 60

SLE900 DLL DESCRIPTION 62

Description of the SLE900 DLL Application Program Interface Version 2.03 63
Overview: 63
Linking C Programs with DLL: 63
DLL Data Types: 63
DLL/SLE Error Codes: 64
DLL Keypad 65
Definitions: 65
DLL API Function Argument Definitions: 66
DLL Structure Definitions: 67
Detailed Function Descriptions: 72
C Programming Examples to Access DLL: 89

Preface

This manual contains operation instructions and reference information for the **dBm** SLE900. The SLE900 reproduces link effects found in communications between earth stations and satellites.

This manual is prepared as a reference source for engineers and technicians to use the SLE900 as part of their earth station/satellite transceiver design and testing.

The SLE900 operations manual is divided into the following sections:

- **Section 1: Introduction** shows the SLE900 equipment, control and connector locations, and describes connector functions.
- **Section 2: Local Operation** describes how to operate the SLE900 from the front panel.
- **Section 3: Remote Operation** shows how to operate the SLE900 through the LAN interface.
- **Appendix A: Installation and Troubleshooting** describes installation procedures and lists error messages.
- **Appendix B: Description and Specifications** gives an overview of the SLE900 technical design and provides technical specifications, and verification testing .
- **Appendix C: Maintenance and Warranty** describes the SLE900 warranty and directs how to return the SLE900 for repair or calibration.

Conventions Used in This Manual

Text Conventions

This manual uses the following text conventions:

- *Italic text* indicates new terms, directories and/or filenames.
- Underlined Text indicates SLE900 selections or key presses.
- Monospaced text indicates SLE900 commands entered through remote mode.
- **Bold monospaced text** indicates SLE900 responses through remote mode.

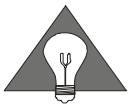
Symbols

The following symbols appear in the manual.

See also, 

This symbol and its "see also" text is placed next to subject matter in the manual to tell you where to find more information.

*This icon indicates a **warning**. Failure to follow the instructions given here may result in **personal injury** or **damage to the equipment**.*



*This icon indicates a **tip**. Text marked this way may be an optional procedure for accomplishing a task, or a time-saving procedure for advanced or familiar users.*

Contacting dBm

We encourage you to contact us if you want more information or have any questions or concerns about this or any other dBm product or manual. Use any of the following methods:

Mail	dBm 32A Spruce Street Oakland, NJ 07436
Telephone	(201) 677-0008
Fax	(201) 677-9444
E-mail—Technical Support	*Info@dbmcorp.com*
www	*http://www.dbmcorp.com*

1

Introduction Section

Introduction

This section introduces you to the Satellite Link Emulator instrument and describes the physical interface and turn-on procedure.

Topics include:

- Front, rear, and interior views.
- Power and cable connections.
- Startup and shutdown procedures.

General Information

Front panel view

The Satellite Link EmulatorThe SLE900 reproduces link effects found in communications between earth stations and satellites. A touch sensitive graphics panel is utilized to minimize the number of hard keys that would be required to control the multitude of functions contained in the instrument.

Figure 1-1. Instrument Front Panel View

Front view shows a single channel model. Multiple channel models have two additional Type N (f) connectors for each channel.

For custom SLE900 units, consult additional documentation provided with the instrument.

Rear Panel Connections

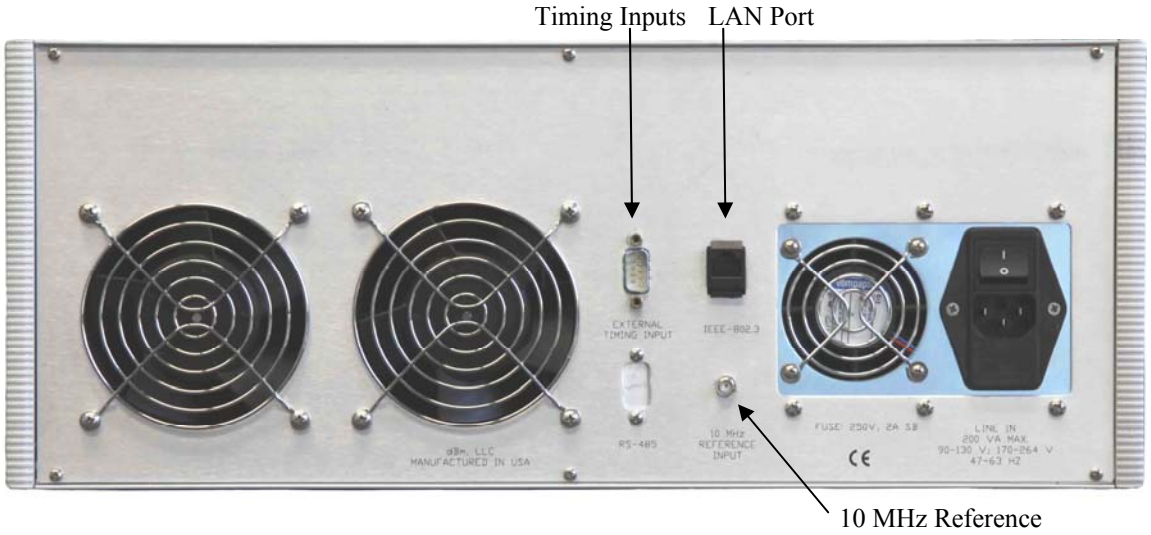


Figure 1-2. Instrument Rear Panel View

Pin #	Timing Control Connector
1	Output, +5V DC
2	Output, RS232 Tx
3	Output, RS232 Rx
4	Input, External trigger, TTL
5	GND (for RS232)
6	GND (for timing signals)
7	Input, External Update Clock, TTL
8	Not used
7	Not used

Figure 1-3 Rear Panel Timing Control Connector

10MHz Ref input: SMA (f) connector, 50 ohms, sine, 0 dBm +/- 3dB

(SLE900 will automatically switch to its internal reference if no external reference is present)

Due to multiple signals present in the Control connector, use of a standard RS-232 cable may damage the equipment.

Instrument Modules

The SLE900 is constructed using modular self-contained sub-assembly trays. Each sub-assembly tray is easily changed in the field.

Interior photo

Figure 1-4. Interior view of a 4 channel SLE900

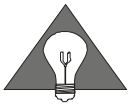
Start and Shutdown Procedures

Starting the SLE900

1. Press the Line on/off switch on the rear panel. The standby indicator should light. If not lighted make sure the power cable is connected properly. Press the power on switch on the front panel. The standby LED turns off and the instrument powers on.

Shutting Down the SLE900

1. Press the standby/on switch to standby. The standby LED illuminates.



If the SLE900 is going to be off for an extended period of time, you may wish to remove the main power by pressing the Line on/off switch on the rear of the instrument.

2

Local Operation Section

Local Operation Overview

The SLE900 is a laboratory instrument designed to apply impairments to an RF signal, simulating the effects that are encountered in a space based wireless channel. The impairments include a) delay, b) frequency offset, c) attenuation, d) phase offset, and optionally e) AWGN, f) multipath fading, and g) tunable L-Band frequency control. The instrument is controllable from the front panel or remotely via LAN.

Each of the installed parameters can be set to a fixed value. An additional feature is the capability to dynamically vary delay, frequency offset, attenuation and phase offset. The values of the parameters are controlled by data files that can be generated by the user or via SATGEN, an application that creates the files based on ephemeris satellite data. The parameters can be executed at time intervals ranging from 1 msec to 1000 msec. The number of points per file (i.e. per parameter) is up to 5 million. The RF output signal remains phase continuous during all dynamic updates. The time varying delay causes Doppler shift and chip rate variations.

During power-up, the PRESET state of the instrument is STATIC mode. The input signal passes through the instrument, and the link effects, as displayed on the front panel, are applied to the signal.

In dynamic mode, the SLE sequentially implements the data values in each parameter file at regular time intervals. The user selects either an internal parameter update rate or applies an external update clock to set the time intervals. Each rising edge of the update clock causes the next point in the data file to be implemented. An external start signal can also be used to accurately trigger the execution of a data file.

The delay and Doppler shift accuracy is directly related to the accuracy of the 10 MHz reference clock. The SLE900 has an internal reference and also accepts an external 10 MHz. To ensure excellent accuracy and synchronicity with other equipment, a common external 10 MHz reference it typically used for the test system.

Operating States

Power up and Reset

Upon power-up or reset, the instrument is set to the default static state. The display will momentarily (3-5 sec) indicate the version of the installed firmware and the model number.

Static

Static (non-varying) values of delay, frequency offset, attenuation, and phase offset are applied to the RF input signal. Additional optional functions include tunable L-band frequency control, additive noise, and multipath fading. The signal propagates through the instrument and appears at the output with the impairments applied. The user can modify any of the impairments by entering a new value, or by using the ↑ key or ↓ key to step. The step size is set using the Step Size menu.

Dynamic

Data files can be downloaded into the instrument through the LAN interface and stored on internal flash memory. These files control the values for delay, frequency offset, attenuation, and phase offset. Each parameter type in each channel can be loaded with the internally stored parameter data files. Upon receipt of a Start command, the instrument sequentially executes the values in the parameter data files. There are four file types per channel, and therefore up to sixteen files (for a four channel instrument) can be implemented simultaneously and synchronously with the same update clock. Each file can contain up to 5 million samples. At each rising edge of the clock, the next data point in each file is executed.

When the Start command is issued, the instrument begins executing the data files. A universal start command is simultaneously issued to all channel hardware to insure a synchronous start. If no parameter file is selected for a particular impairment, that parameter will remain at its static value. When the Reset command is received via the LAN or front panel, data implementation is immediately stopped and the instrument returns to the initial data point. When a Pause command is received via the LAN or front panel, data implementation is immediately stopped, parameters are frozen at that point, and the current parameter values are displayed on the front panel of the SLE. When in pause mode the increment ↑ key or decrement ↓ key can be used to single step through the parameter files. Pressing the Start key from the PAUSED or READY state beginning execution from the current point in the file. The Mode key on the front panel toggles the SLE between Static and Dynamic mode.

Viewing the Instrument's Hardware Configuration

The hardware configuration and parameter ranges installed in the instrument can be viewed by pressing the About softkey from any of the main windows.

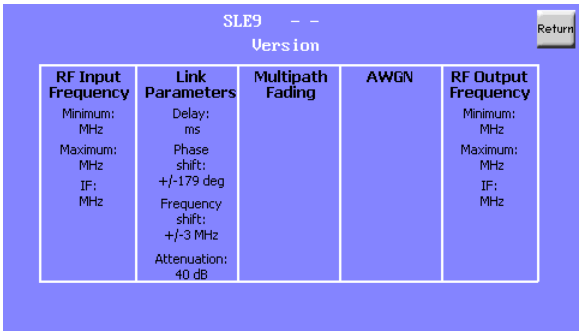


Figure 2-1. The About menu

Editing Parameters

Editing in Static Mode

Parameters are selected for editing by pressing the parameter field on the touch sensitive display. When touching a parameter, the instrument responds with an audible beep, and the background of the selected field is highlighted. The user then enters the new numeric value via the keypad, ending the entry with a units key (ms/MHz, us/kHz, or Enter for ns/Hz/dB). After a units key has been pressed, the new value is stored and editing is complete. Anytime a parameter field is highlighted, that field can be edited. If another parameter key is pressed prior to pressing a units key, the highlight immediately moves to the appropriate field and the original parameter is not overwritten.

When the data field to be edited is selected and a numeric value typed, the data must be entered by pressing one of 3 possible keys. The key assigns the order of magnitude to the data entered as follows:

Key	*units*
Enter	ns, Hz degrees, dB, or dBm
us kHz	us, kHz (not valid for phase or attenuation)
ms MHz	ms, MHz (not valid for phase or attenuation)

Use of the Clear Key

The Clear key is used to delete a new numeric value prior to pressing an enter key.

Example

- Press delay value to select that delay field. Type a value for delay.
- Press ms/us/Enter. The new value is entered.
- Press Delay and enter a value for delay. The characters appear as the new value is typed.
- Press Clear. The previous delay value appears.

Set Delay

Press a delay field for the desired channel (valid during Static mode only)
Press numeric value of delay
Press ms or us or Enter (units of ns) to enter current value.

Set Frequency Offset

Press the frequency offset field for the desired channel (valid during Static mode only)
Press numeric value (valid range: 0 to +/- 3000 kHz, 0.01 Hz steps)
Press MHz, kHz, or Enter (units of Hz) to enter.

Set Attenuation

Press the attenuation field for the desired channel (valid during Static mode only)
Press numeric value (valid range: 0.0 to 40.0, 0.25 dB steps)
Press Enter (units of dB) to enter.

Set Phase Offset

Press the phase offset field for the desired channel (valid during Static mode only)
Press numeric value (valid range: -90 to +90, 1 degree step)
Press Enter (units of degrees) to enter.

Set L-Band Center Frequency

Press the RF soft key to invoke the RF Frequency window.
Press the frequency field for the desired channel
Enter a numeric frequency value (in MHz).
Press MHz to enter.
Move to another main menu using the display softkeys.

Set Step Size (Delay, Freq Offset, Attn, Phase Offset)

Press the Step Size hard key once to invoke the step menu.
Press the desired parameter field to highlight the parameter. Use the keypad to enter a numeric value (valid ranges and resolutions are same as corresponding parameter).
Press the appropriate units key to enter.
Press the Return soft key to return to the main display.
A unique step size is saved for each parameter type, and each parameter step size can be unique for each channel.
Note that if the entered step size exceeds the realizable limit, the out of range condition will be indicated by displaying the parameter in red.

The following step sizes may be defined

Delay	0.1ns to 20,000ns
Freq Offset	0.01Hz to 3,000MHz
Attenuation	0.25dB to 40dB
Phase Offset	1 degree to +/- 90 degrees

↑ and ↓ (up arrow and down arrow)

In Static mode, press the desired link parameter field. Press ↑ or ↓ and the value changes by the selected Step size amount.

Note: During operation, if ↑ or ↓ forces a parameter out of range, the value will revert to the last valid value and the value will be displayed in red to indicate an out of range condition has occurred.

↑ and ↓ are also used in Dynamic when the mode is READY or PAUSED. These controls then single step through the dynamic files.

In the Dynamic File Menu, ↑ and ↓are used to scroll through the list of parameter files that are stored in the instrument.

Change the sign of a value

Press "-" while editing a parameter to change the sign of the entered value. Only Frequency offset and Phase offset can have negative values.

Utility Functions

Pressing Store or Recall, activates the memory utility display. A storage register contains all Instrument State settings, static parameter

values, Update Rate, RF Center frequency, step sizes, and loaded Dynamic file names. It does not retain the current channel selection, Mode, elapsed time, or Start/Reset/Pause status.

To store the instrument settings

1. Press Store. The store utility display appears. Press the number of the desired register to be updated. Store can be invoked from Static or Dynamic modes.
2. Press a number from 0-8, designating a register to store the current instrument settings. Register 0 (labeled Preset State) defines the power up and PRESET state. The register is overwritten and the display reverts to the Static Delay view.
3. Press Return to exit the store utility window without overwriting any registers. Note: Register 9 is not available to store settings as this always contains the factory defaults.

Figure 2-2. Store Menu

To recall a saved instrument setting

1. Press Recall. The utility display appears. Press the number of the desired register to be loaded. Recall can be invoked from Static or Dynamic modes.
2. Press a number from 0-9, designating a register to recall instrument settings. Register 9 cannot be modified and contains factory defaults.
3. Press Return to exit the recall utility window without changing the instrument settings. If the recalled register invokes Dynamic file names that are no longer on the SLE internal memory, then a "file missing" error is displayed and those parameters are set to the default value.

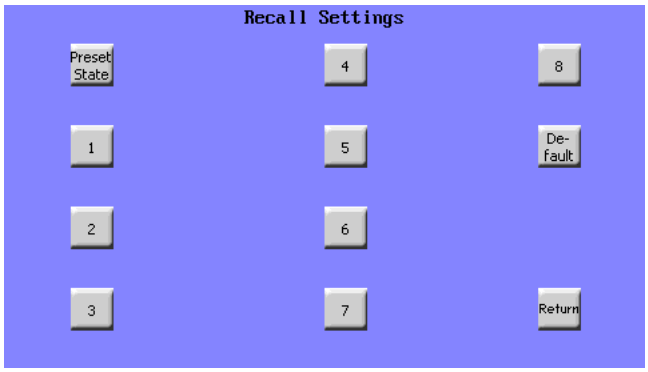


Figure 2-3. Recall Menu

To configure the instrument's LAN port

Press IP ADDR

4. Press IP ADDR hard key. The network utility display appears.
5. Press the IP address data field. Enter the desired IP address and then press Enter, or press Clear to revert to the previous value.
6. Press the Submask address data field. Enter the desired submask value, then press Enter, or press Clear to revert to the previous value.
7. The MAC address is unique for every instrument, and cannot be modified.
8. After editing, press Return to store the new values and return to the main window.

To set the instrument to local or remote (LAN) control

During remote LAN operation, all keys except Local are disabled. Pressing Local brings the instrument back to the local mode and activates the front panel keys. The Local key toggles the instrument between local and remote.

To enter remote mode from the front panel press Local. The instrument will then switch to remote mode. If a LAN connection is present when the instrument is powered on or the Preset key is pressed, the SLE will automatically go to remote mode.

The instrument must be in remote operation mode before it will accept commands from the LAN connection.

To preset the instrument settings

- Pressing Preset causes the instrument to return to the default state which is defined by the contents of register zero (0).

Static Delay Operation

When the Mode key is set to "Static", each of the link parameters is displayed and implemented immediately upon entering Static mode. The display will indicate that the mode is "Static". The parameter values in each channel are independent of other channels.

A unique step size can be set for each of the 4 parameters, as well as independently for each channel. The ↑ and ↓ keys can change the parameter by its step size value.

When delay is changed, the delay line slews to the new value at a rate of 20us/msec. All other parameters change immediately to the new value.

To edit a parameter, touch the parameter field area in the display. The instrument responds with an audible beep, and the background of the selected field is highlighted. The user then enters the new numeric value via the keypad, ending the entry with a units key (ms/MHz, us/kHz, or Enter for ns/Hz/dB). After a units key has been pressed, the new value is implemented and editing is complete. Anytime a parameter field is highlighted, that field can be edited. If another parameter key is pressed prior to pressing a units key, the highlight immediately moves to the appropriate field and the original parameter is not overwritten.

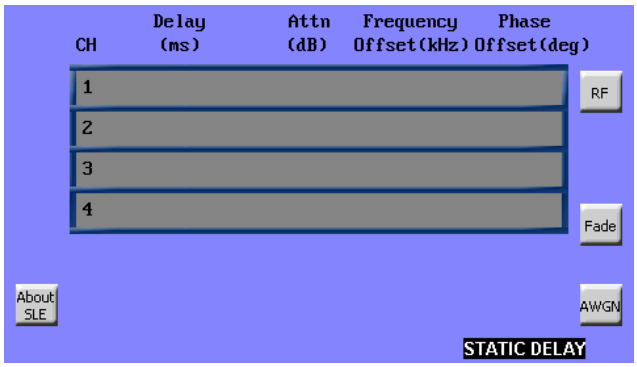


Figure 2-4 Static Delay View

Figure 2-4 shows a typical front panel display in static mode. Each of the 16 parameter fields can be edited by touching near the center of the value.

Dynamic Delay Operation

When Dynamic Delay is selected via the Mode hard key, the SLE automatically invokes the file menu display. The displayed file will be the one implemented in Dynamic mode. To change a file, press the parameter field to highlight the file name, and use the ↓ and ↑ arrow

keys to scroll through possible choices. A description of the currently highlighted file is shown at the bottom of the display. Once completed, press the Done softkey to return to the Dynamic delay view.

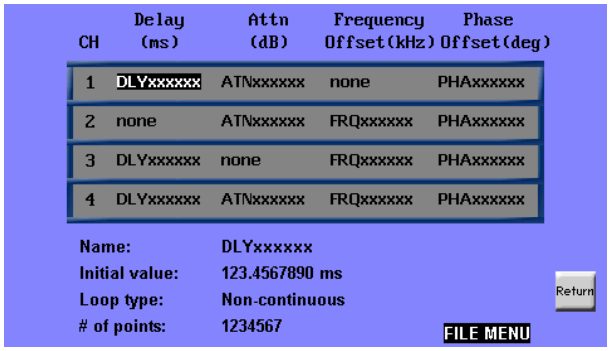


Figure 2-5. Typical display when loading dynamic files

Once the file loading sequence is complete, the display returns to the normal view, and the 1st point of each file is pre-loaded by the hardware. Implementation of subsequent data points begins upon the receipt of a start command.

While running, the displayed elapsed time (ET) counter increments. Parameter data is updated at a rate varying from 1 sample per second, up to 1 sample per millisecond, as defined by the update rate, which is also displayed.

The update clock source is controlled by the Update key. When set to internal, the SLE generates an update clock at the desired interval. When set to external, an external clock must be applied to the rear panel of the instrument at the appropriate rate.

The trigger source is controlled by the Trigger key. When set to internal, the SLE triggers dynamic execution in response to the front panel Start key or a start command received from the LAN interface. When set to external, dynamic execution begins when an external signal is applied to the rear panel on the instrument.

Implementation of phase offset and attenuation occurs simultaneously with the start of a delay change when the Time Ref key is set to input. Implementation of these two parameters is delayed such that they are changed when the delayed RF signal appears at the output of the delay line when the Time Ref key is set to output.

The Loop key determines how files are implemented. Files are run once from beginning to end (Single), repeatedly from beginning to end (Continuous) or repeatedly from beginning to end and then from end to beginning (Forward/Reverse).

Dynamic Modes

When Dynamic is selected, the instrument will be in one of 5 modes, which are displayed on the bottom right corner of the front panel display.

Ready

The instrument is initialized, and dynamic execution can begin.

Run

The instrument is currently executing dynamic files. The elapsed time counter will be incrementing.

Armed

The trigger has been set to external and the instrument will begin file execution upon receipt of a hardware trigger signal.

Paused

The instrument is paused during a dynamic run. The currently implemented parameter values are displayed. Pressing the ↑ key or the ↓ key will single step through the files. Pressing Start from this point will continue execution. Pressing Reset from this point will cause the instrument to initialize the current dynamic files.

Done

The instrument has finished execution of a single dynamic run. The last value of each parameter file is implemented in hardware.

Dynamic File Type names

The SLE distinguishes parameter file types by the first three letters in each file name. File names can be up to 10 alphanumeric characters, as follows:

- DLYxxxxxxx - designates a delay file
- FRQxxxxxxx- designates a frequency offset file
- ATNxxxxxxx - designates an attenuation file
- PHAxxxxxxx - designates a phase offset file

RF Center Frequency Operation

Pressing the RF softkey invokes the RF View, which allows modification of the input and output RF center frequencies. For IF instruments, these settings will be fixed at either 70 MHz or 140 MHz.

If the instrument is configured to operate at L-band, these settings will control the input and output center frequency of the L-band frequency converters.

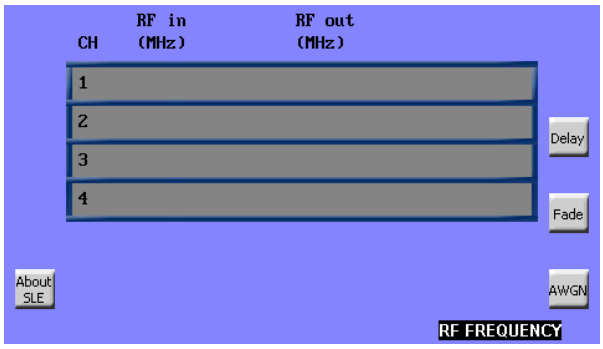


Figure 2-6. RF Frequency View

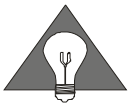
Additive White Gaussian Noise Operation (Optional)

The optional AWGN feature applies Gaussian noise to the user's signal. There are two modes of operation 1) Ratio mode that sets an Eb/No ratio, or 2) Noise Density mode, that sets the noise power spectral density to a fixed level, irrespective of the signal level. The noise has a constant power spectral density over the operating bandwidth of the instrument. For example, if the instrument has a 72 MHz bandwidth, the noise will occupy the entire 72 MHz.

The ratio mode uses a true rms detector to measure the input signal power. Ratio and Bit Rate settings in combination with this measured power determine the applied noise density, such that the desired Eb/No ratio is achieved. The resultant noise density is displayed.

The Mode softkey is used to toggle between the two modes.

Figure 2-7. Noise Ratio View



The power of the combined signal plus noise must be limited, othewise distortion will occur. When noise is applied, the maximum input signal amplitude (S_{max}) must be reduced:
*S_{max} = 10*Log{1-10^(Pn/10)} , where P_n = 10*Log(No)*

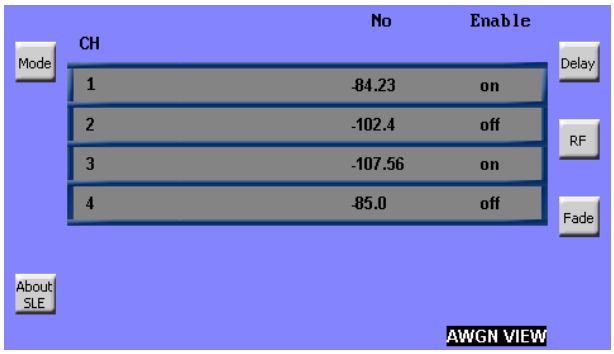


Figure 2-8. Noise Density View

Multipath Fading Operation (Optional)

The optional Multipath Fading feature applies up to 6 paths of multipath fading. Rayleigh and Rician distributions are available, as well as line of sight only (CW). Each path has control for the amount of spreading (via Speed), path loss, and path delay.

Rician K-factor can be set by pressing the K-factor softkey.

Correlation between paths can be set from 0 to 100% by pressing the Correlation softkey.

Multipath fading can be installed on any or all channls. To view an alternate channel, press the Channel soft key.

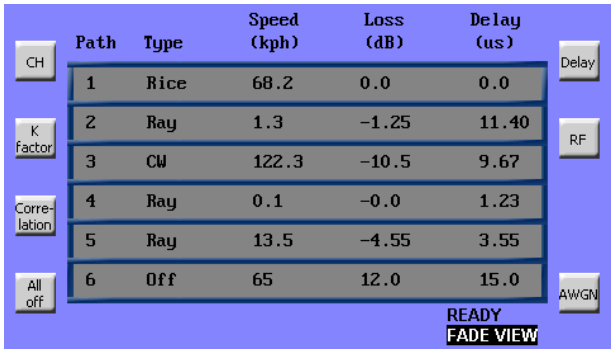


Figure 2-9. Multipath Fading View

LED Functions

This section describes how the front panel LED's function.

Standby	Illuminated while main power switch is in the off position, to indicate that power is applied to the instrument. This LED is controlled by hardware.
Remote	Illuminated when SLE has been put into remote by the Local key. All front panel controls except Local are disabled when in remote.
Update	Two LED's to indicate whether the current selection is Internal or External.
Loop	Three LED's to indicate Single, Continuous, or Forward/Reverse sequence of executing Dynamic files
Trigger	Two LED's to indicate whether the current selection is internal or external.
Mode	Two LED's to indicate whether the SLE is in Static or Dynamic mode
Time Ref	Two LED's to indicate whether RF parameters are implemented relative to the signal being at the input of the delay line, or when the signal appears at the output of the delay line.

3

Remote Operation Section

Remote Operation Overview

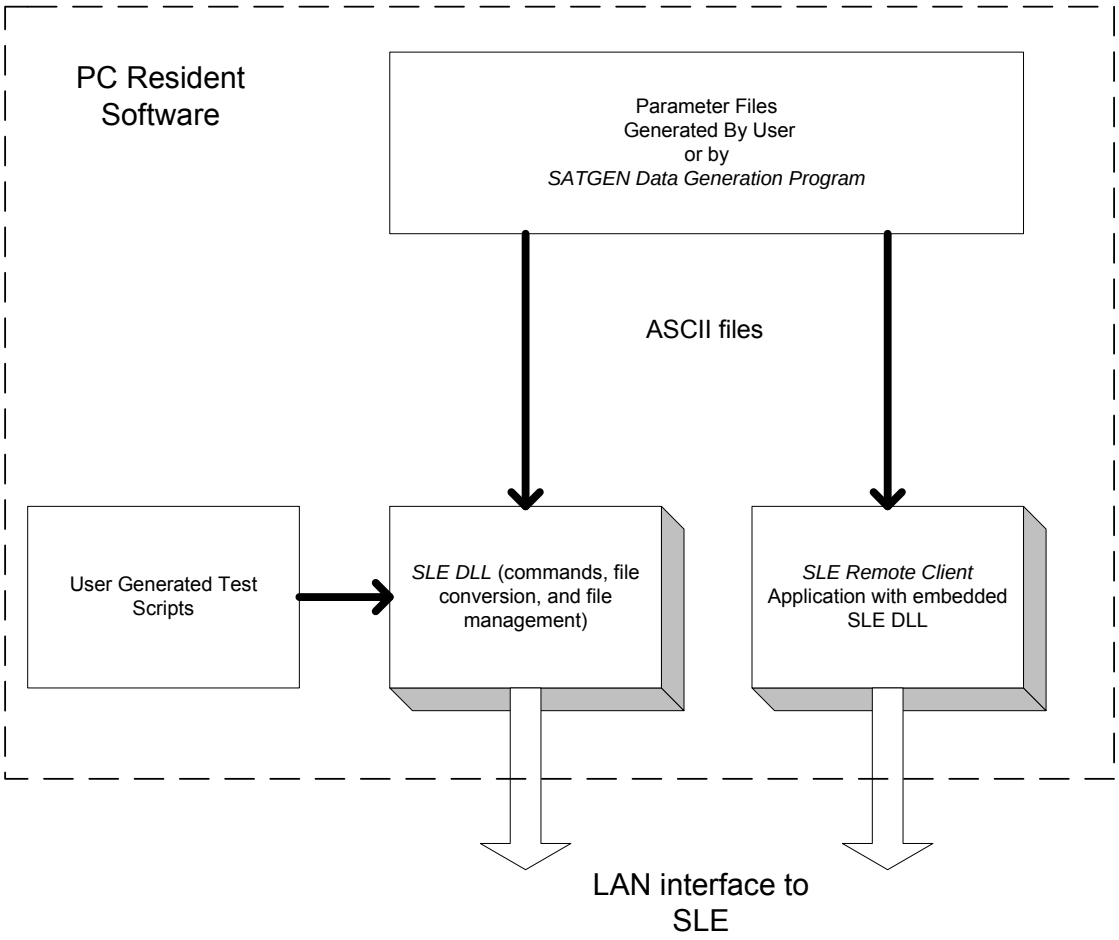
The SLE900 can be controlled remotely using its LAN interface. The instrument can be connected to any IEEE-802 network. It uses TCP/IP, and achieves transfer rates up to approximately 5 MBPS. All front panel controls are also implemented in the SLE900 LAN client application called *SLEControl*. In addition, parameter files may be downloaded into the SLE900 or deleted from the SLE900 memory through the LAN interface.

Programming control of the SLE900 can be implemented by two means: 1) Using *SLEControl* provided by dBm or 2) by creating an application, such as a test script, which makes calls to the DLL provided with the instrument. A complete description of SLEControl and the SLE900 DLL are given in this manual.

SLEControl provides a graphical user interface to manipulate parameter files, and to control and monitor the SLE, both in Static and Dynamic mode. The DLL is embedded in *SLEControl*, so that DLL functions can be exercised within the application through the GUI.

The SLE900 DLL contains numerous utilities including:

- Exercise of all front panel key functions for Static and Dynamic modes
- SLE900 mode and setup controls
- File conversion (ASCII format to compressed format)
- File download into the SLE900 memory
- File deletion from the SLE900 memory
- LAN setup and controls
- SLE900 status reporting
- SLE900 error reporting



The SLE900 can be controlled remotely using dBm's DLL or SLEControl

Figure 3-1

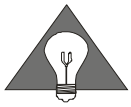
Setting the SLE900 IP Address

To set or view the SLE900 IP address, press IP ADDR key.
Enter the desired IP address. Press Enter.

Remote Control via LAN

The SLE is configured as a network server, and can communicate with a network client. The client can download and delete dynamic parameter files on the mass storage device in the SLE.
All front panel Static and Dynamic commands can be initiated with the LAN interface. While in remote all front panel controls except the Local button are disabled.

Editing in Dynamic Mode



Blue hard keys are specific to Dynamic mode

From the Static Delay window, press the gray Mode key to invoke the Dynamic File menu window.

Change Update Rate

Press the Update Rate key to select the update interval field. The pointer appears at the update rate field.
Type a value for update period.
Press Enter. The new value is entered and the pointer disappears.
Valid values are 1, 2, 5, 10, 20, 50, 100, 200, 500, and 1000 ms. Invalid entries are rounded to the nearest valid number.

File Menu

The File Menu appears automatically when entering Dynamic Delay mode. Press File Menu to invoke the file display while already in Dynamic mode. Each of the 4 dynamic parameter values is replaced with file names for that parameter. Selecting and de-selecting dynamic files is done from this menu.
Characteristics of each file are displayed when the parameter is selected by touching the parameter field. Scroll through the stored file names using the up/down arrow keys. Pressing the Done soft key loads the displayed files and returns to the main dynamic display.

Scroll Down

Once the file menu is displayed, the ↓ key will cause the displayed file name to scroll downward through the list of file names that are valid for that parameter and are currently stored on the internal storage medium.

Scroll Up

Once the file menu is displayed, the ↑ key will cause the displayed file name to scroll upward through the list of file names that are valid for that parameter and are currently stored on the internal storage medium.

Mode

Press Mode to toggle between Static and Dynamic modes. When exiting the Dynamic mode, parameter values retain the last Dynamic value. When entering Dynamic mode, parameter values become the 1st point in each respective data file.
The Mode key is enabled only in Static, and in Dynamic when the status on the front panel display shows "Ready".

Update

Press Update to toggle between an internally and externally applied update signal. LED's on the SLE front panel indicate the selection. When set to internal, the update clock is generated internally. The internal clock is derived from an internal timer. The accuracy of the internal timer is based on the accuracy of the 10 MHz reference clock. When set to external, the instrument executes one parameter datum on each external rising clock edge, after a START signal is received. The external clock is a TTL logic level signal that is applied at the external timing input connector at the rear of the instrument. The Update key is enabled only in Dynamic when the status on the front panel display shows "Ready".

Loop

Press Loop to toggle between "Single", "Continuous", and "Forward/Reverse". In "Single", parameter files are run from beginning to end, and then execution stops, and the mode changes to DONE. In "Continuous", files are repeatedly run from beginning to end. Mode remains at RUN. Files can only be run in Continuous looping if the first data point and the last data point in the file are identical. If Continuous is selected, and all selected parameter files do not have matching end points, Loop will automatically revert to "Single". All selected files must indicate "Loop:Continuous" in order for "continuous" to be active. In "For/Rev", files are repeatedly run from beginning to end, and then from end to beginning. The Loop key is enabled only in Dynamic when the status on the front panel display shows "Ready". If files of unequal lengths are run, the last point of the shorter file(s) is held until the last point of the longest file is executed.

Trigger

Press Trigger to toggle between "Internal" and "External". LED's on the SLE front panel indicate the current selection. When set to "Internal", file execution begins when a Start command is received from the front panel or via an Ethernet command. Latency of the software generated command can cause start time uncertainty when using this type of trigger.
When set to "External", the Start command arms the instrument, and file execution begins on the 2nd update clock after the receipt of the rising edge of a TTL trigger signal applied to the trigger input on the External Timing Control connector on the rear panel of the instrument. Once triggered, parameter changes are executed on the rising edge of the update clock applied to the External update clock input (pin 7) on the External/timing input connector on the rear panel of the instrument.
Latency from trigger to actual implementation of the 1st data point is equal to two update clock periods.
The Trigger key is enabled only in Dynamic, and when the status on the front panel display shows "Ready".

Time Reference

Press Time Ref to toggle between "Input" and "Output". When set to "Input", all link parameters are executed simultaneously. When set to "Output", the execution of Attenuation and Phase Offset is held off in time by an amount equal to the current delay file data point, such that these 2 parameters are implemented when the RF signal sample reaches the output of the delay line.
The Time Ref key is enabled only in Dynamic when the status on the front panel display shows "Ready".

Start

Press Start to begin Dynamic file execution at the current interval rate. The displayed status on the front panel display changes to "Run". If Trigger = External, then the status on the front panel display changes to "Armed", and data is implemented only after an external trigger signal has been received. The status on the front panel display will change to "Run" after receipt of the external trigger. Start is disabled when the status on the front panel display shows "Run" or "Done". Start is also used to re-start after Pause. Pressing Start after Pause causes execution to continue from its current point.

Reset

Press Reset to stop Dynamic file execution and reset each parameter to the 1st point in the current link parameter files. ET (elapsed time) display is reset to zero. Reset must also be used to re-initialize the data files after a single run is complete and the status on the front panel display shows "Done". Reset is disabled when the status on the front panel display shows "Ready".

Pause

Press Pause to stop Dynamic file execution and hold elapsed time at its current value. The displayed parameter values match the actual data implemented in hardware. Parameters are not reset. ↑ and ↓ keys may be used from the Paused state to single step through the dynamic files. Pressing Start causes execution to continue from its current point. Pause is disabled when the status on the front panel display shows "Ready", "Done" or "Paused". When the update interval is set to a longer value, there will be a noticeable delay after Pause is pressed until the parameters stop changing. During this interim period, the status will indicate "Halting" before changing to "Paused".

LAN File Formats

The parameter data files should be generated in ASCII format as a sequential data list with a carriage return <CR> separating the parameter fields. File names must begin with "DLY", "FRQ", "ATN", or "PHA" and end with ".DAT" When the files are converted (optimized) the files have the same prefix, but the suffix becomes ".SLE". The converted files are then transferred over the LAN interface, and stored in flash memory. The converted files are in binary format (to optimize storage) with an ASCII header (so that information can be displayed on the front panel).

Data File Description

Delay Files

- Each field consists of 10 to 14 characters, depending on resolution, including a decimal point
- Range is 0.1ms to 1400ms in 0.5 ps increments
- Valid change (delta) between any two adjacent points must not be greater then 20us
- Units are ms

Phase Files

- Each field consists of four characters including a"-" sign
- Range is ± 90° in 1° increments
- Step size between any two adjacent points can be up to ± 90°
- Units are degrees

Attenuation Files

- Each field consists of five characters including a decimal point
- Range is 0 to 40 dB in 0.25dB steps
- Step size between any two adjacent points can be up to 40 dB
- Units are dB

Frequency Offset Files

- Each field consists of eight or 10 characters, depending on resolution, including a decimal point
- Range is 0 to ± 3000KHz in 0.01Hz resolution
- Step size between any two adjacent points can be up to ± 3000KHz
- Units are kHz

In Dynamic mode, Frequency offset resolution is 0.01 Hz if there are 5 decimal places in the first data point in a frequency offset file, eg. x.12345 will invoke 0.01 Hz resolution. Otherwise, resolution will be 1 Hz.

Data File sizes

Parameter data files can contain up to 5 million data points.

ASCII file format

Parameter data files must be generated in ASCII format. The first line in the file is a value that represents the number of sample points in the file. Each subsequent line will contain one data value. Lines are separated by a carriage return. An example of a delay file with 3 points is:

3
12.456789
12.456788
12.456787

The ASCII parameter files must have a file extension of ".dat" in order to be recognized by the conversion program. The SATGEN satellite data generation program automatically generates the correct file extension. User generated files should be created with the .dat extension also. The first three letters of the file name should be one of DLY, FRQ, ATN, or PHA to represent delay, frequency offset, attenuation, or phase offset files. Up to 7 alphanumeric characters can follow the first three letters in the file name.

Description of the SLE900 Remote Client Application Program *SLEControl*

SLEControl provides a graphical interface to control the SLE900 from a PC via a LAN connection. The client application provides Static Controls, Dynamic Controls, allows download, deletion and selection of parameter files in the SLE900, and displays the parameter file data in graphical form.

*SLEContro*l provides several functions:

1. Imports ASCII based data files such as those generated by SATGEN.
2. Provides real time control of all Static and Dynamic SLE900 functions, including Start, Pause, and Reset in the Dynamic mode.
3. Converts the ASCII parameter files to a format compatible with the SLE900, and provides simple controls for downloading files into the SLE900 memory.
4. Allows selection and deletion of files in the SLE900 memory.
5. Provides a graphical representation of the parameter files, and a real time cursor to indicate execution progress during a Dynamic run.

Installing *SLEControl* on a PC

SLEControl and the SLE DLL can be copied from the provided CD to a directory on a PC. Create a directory, for example c:\SLE900 Client, and copy the following files into the directory:

SLEControl.exe
SLEDLL.dll
SLEDLL.lib
SLEDLL.h

Connecting to the SLE900

To establish a connection from the PC to the SLE900, connect the two devices to a local area network, or connect them directly using an ethernet crossover cable.

Press the Local button on the SLE900. The remote LED should illuminate. Pressing Preset with a LAN connection present also places the instrument in remote mode.

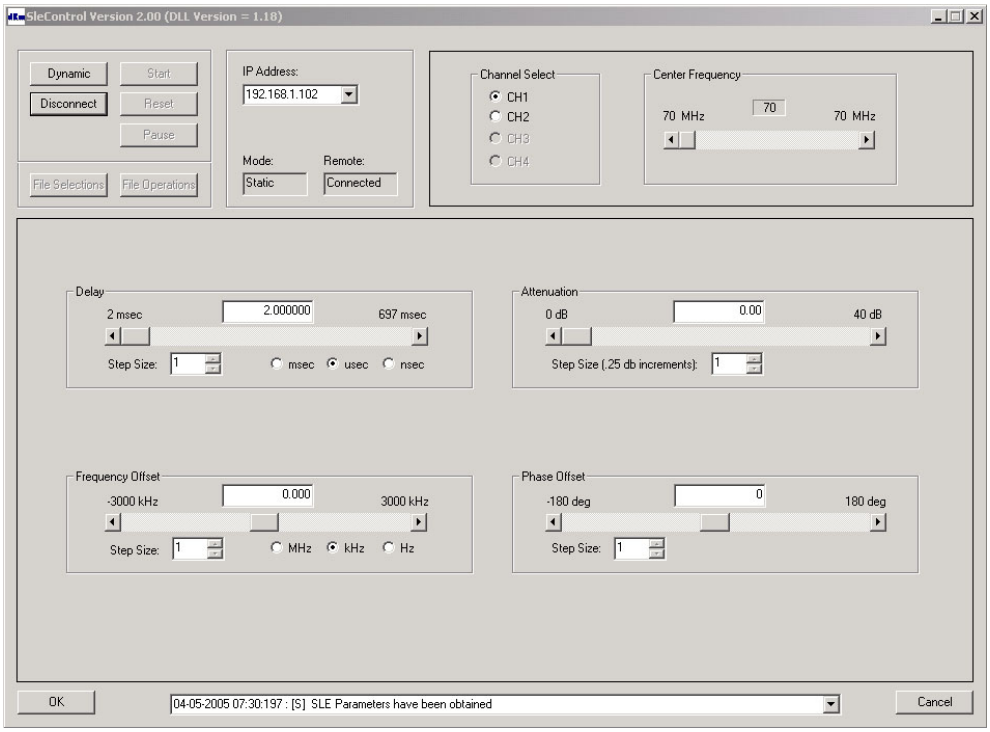


Figure 3-2. Static Control Window

In the Client main window, set the IP address to match the SLE900 IP address.

Press "Connect". When the link is established, the same button will indicate "Disconnect" and the status window labeled "Remote" will also indicate "Connected".

Static Controls

If the instrument is in Dynamic mode, press the "Static" button on the GUI to change to Static mode.

Select the desired Channel by clicking on a channel number.

Center Frequency is determined by the SLE900 hardware configuration, and cannot be changed.

Each of the four Static Parameter controls can be exercised, and the SLE900 will respond immediately.

Each control value can be changed in one of three ways:

- Drag the slide bar with the mouse.
- Click on the end of the scroll bar to change the parameter by the amount set in the step size. For Delay and Frequency Offset, the units of the Setp Size must also be set (msec, usec, nsec and MHz, kHz, Hz respectively)
- Click inside the scroll to change the value by a larger amount.
- Type a new value in the text box and press "Enter".

Dynamic Controls

If the instrument is in Static mode, press the "Dynamic" button on the GUI to change to Dynamic mode.

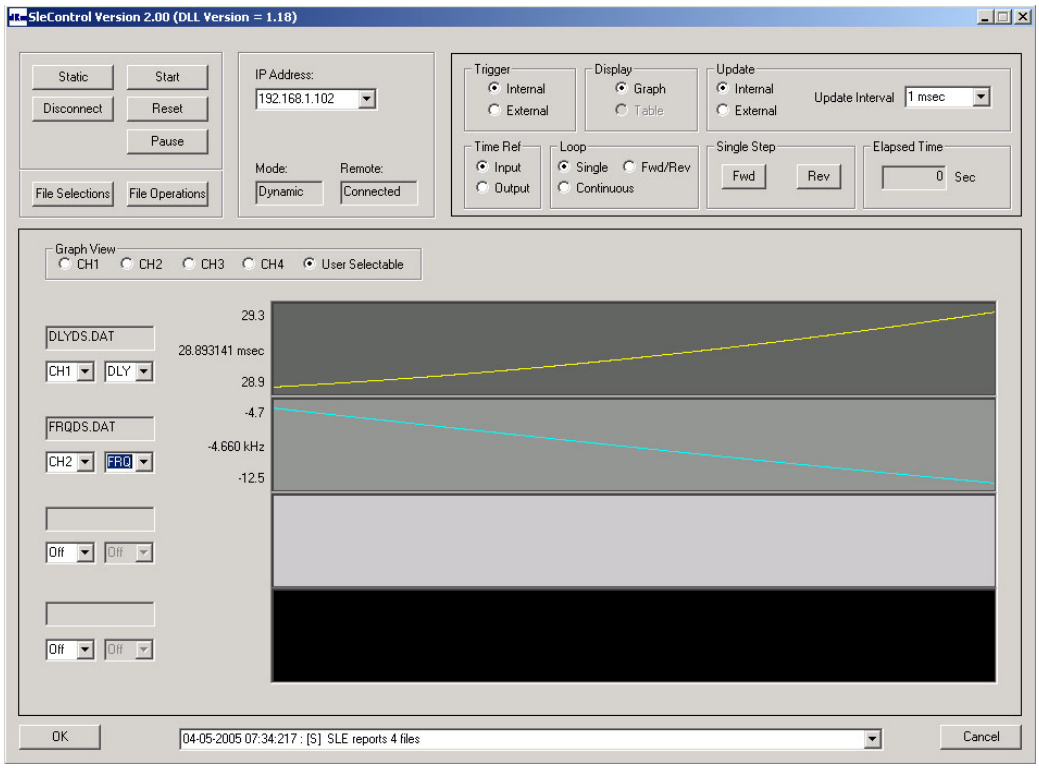


Figure 3-3. Dynamic Control Window

The Dynamic controls at the upper right of the display (Trigger, Time Ref, Loop, Update, and Update Interval) can be exercised and the SLE900 will respond immediately.

Note that Single Step and the Graphing functions are not valid until parameter files have been selected in the SLE900.

Downloading Files into the SLE900 Memory

Parameter files that are resident in the PC can be downloaded into the SLE900. The file must be in proper ASCII format, have a ".dat" extension, and have the appropriate file name.

Press the File Operations button.

Browse to the PC directory where the desired ASCII files are located.

Highlight the desired file(s) for download. Multiple files can be selected by clicking on multiple file names. The selected files will appear with an alternate color background to indicate they are ready for download.

Press Download, or Download All. All of the selected files will be sequentially downloaded. Progress of the download is indicated on the SLE900 display. Once a file has been downloaded, the background color for that file will return to normal.

When downloading is complete, the files will now appear in the lower box labeled "SLE Resident Files.

The List All button to be used to to read the SLE900 resident file directory.

Press Close in the File Operations window.

If a file is downloaded, and has the same name as a file already resident in the SLE900 memory, the previous file will automatically be overwritten.

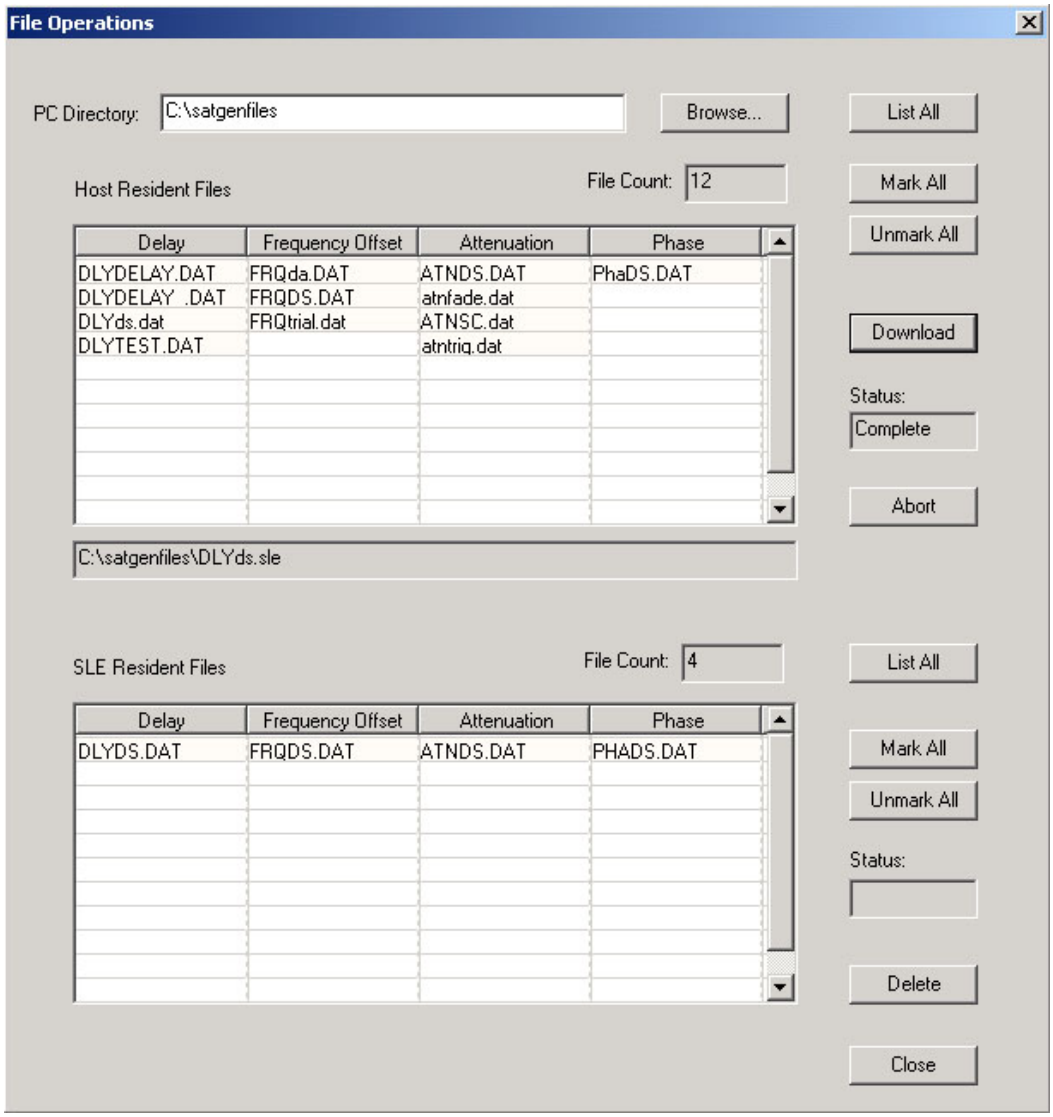


Figure 3-4. File Operations Window

File Selection

Once files have been loaded into the SLE900 memory, they must be selected to indicate which files are to be included in a Dynamic run.

Press the File Selections button.

For each channel/Parameter type grid, use the drop down box to view and select a file. These files will be used when a Dynamic run is initiated. If "None" is selected, the parameter value will not be changed from its existing value.

Once the file names have been designated, the SLE900 must be commanded to select those files. In addition, there is the option to generate graphs of the files. Under most conditions, it is desirable to generate the graphs to be viewed on the PC monitor. This is done by pressing the "Select Files and Build Graphs" button.

For parameter files with millions of samples, it can take several minutes to build a graph. Under this condition, to avoid the delay in execution, press the "Select Files without Building Graphs". The SLE900 will function normally but the graphic display of the parameter file data will not be available on the PC.

Once either of the Select Files buttons has been pressed, the window will close and the SLE900 will be initialized with the first point of each selected file.

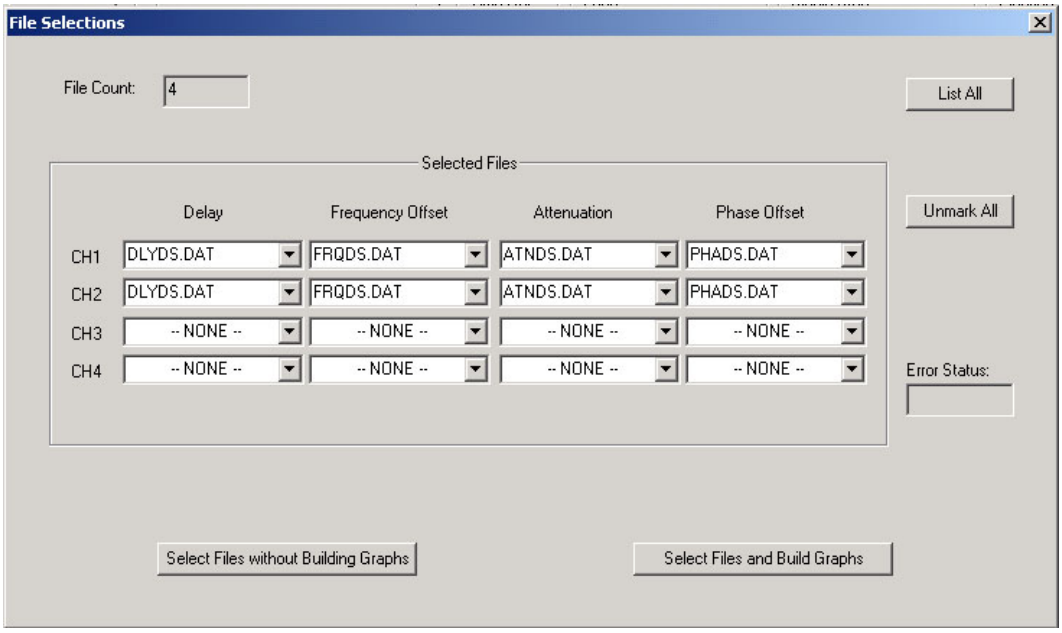
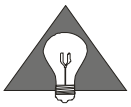


Figure 3.5. File Selection Window



If a file in the SLE900 memory is selected, and the corresponding .dat file is not resident in the PC, the file will automatically be uploaded into the PC and subsequently will be graphed.

Deleting Files from the SLE900 Memory

If it is desired to remove parameter files from the SLE900 memory, it can only be done using *SLEControl* or the SLE DLL.

Press the "File Operations" button.

Press the "List All" button associated with the SLE Resident Files to list all existing files in the SLE900 memory.

Select the files to be deleted from the listing of SLE900 resident files by clicking on the file name. Alternatively, press "Mark All" to mark all files for deletion.

Press the "Delete" button to clear the SLE900 memory.

Press "Close" to close the File Operations window.

Starting a Dynamic Run

From the main display, press Start. The Mode will change to Run, and the Elapsed Time counter will increment. If graphs have been selected, the cursor will increment, and the parameter values at the left hand edge of the graph will change in real time.

Press Pause. The instrument will stop at the current parameter value. Those values will be displayed on the graph, and the cursor will indicate the location.

Single Step can be used to manually increment or decrement through the data files.

Pressing Start while being Paused will cause the SLE900 to continue execution from the current point.

Press Reset to return to the initial data point. The instrument status returns to "Ready". Single Step can also be used in the "Ready" state.

Graphical File Display

Up to four parameter files can be graphically displayed simultaneously. The Graph View buttons determine if the graphs will be for a single channel only, or by pressing "User Selectable" any combination of channel and parameter file type can be displayed.

For each graph, choose the channel number, and then the parameter file type. The file name corresponding to those chosen in the File Selection window will appear.

In the case where files of unequal length have been selected, the time scale is adjusted to accommodate the longest file. The final data point of the shorter files is held to the last point of the longest file.

The vertical scale of the graphs is automatically adjusted to accommodate the maximum and minimum data point of the displayed file.

When execution begins by pressing Start, the cursor increments to indicate the file execution, and the current parameter data values are displayed at the left end of each graph.

When execution is halted by pressing Pause, the cursor and displayed data value indicate the actual parameter value.

A

Installation and Troubleshooting

Installation and troubleshooting

If your SLE900 unit does not appear to be performing as expected, vertification tests can determine whether the unit is functioning properly.

This section describes installation instructions, SLE900 vertification tests and provides a checklist for results.

Topic includes:

- System Installation
- Error messages

Installation

Unpacking the SLE900

Remove the SLE900 materials from the shipping containers. Save the containers for future use.

The standard SLE900 shipment includes:

Quantity	Description
1	Satellite Link Emulator
1	CD with application programs
1	AC power cord
1	SLE900 User Manual

Initial Inspection

Inspect the shipping container (s) for damage. If container is damaged, retain it until contents of the shipment have been vertified against the packing list and instruments have been inspected for mechanical and electrical operation.

If the SLE900 appears to have been damaged during shipping, do not install the unit. Contact dBm immediately.

Applying power

1. Place the SLE900 on the intended workbench and connect the AC power cord to the receptable on the rear of the unit.

2. Press the Line on/off switch on the rear panel. The standby indicator should illuminate

3. Press the power on switch on the front panel. The standby LED should turn off and the instrument should power on.

System verification

The following section provides the procedure to verify correct operation of the SLE900.

Required Equipment

You should have the following equipment (or equivalent) available for verification testing.

- HP8566 Spectrum Analyzer
- HP8341B Synthesized RF Generator
- HP5370B Frequency/Time Counter
- HP3325A Pulse Generator
- TEK2465B Oscilloscope
- TRILITHIC CD-50 RF Detector (2 PCs)
- EPM 441A Power Meter
- HP5340A RF Frequency Counter

Attach a main AC power cord and set the rear panel main power switch to the up position. The front panel standby LED will illuminate.

Turn on the front panel power switch. The model number and software version will appear in the display, followed by the Instrument State display.

Passband Flatness testing

1. Connect the equipment as shown in Figure A-1. The SLE900 should be in static mode, with delay set to 2 ms and all other parameters set to 0.

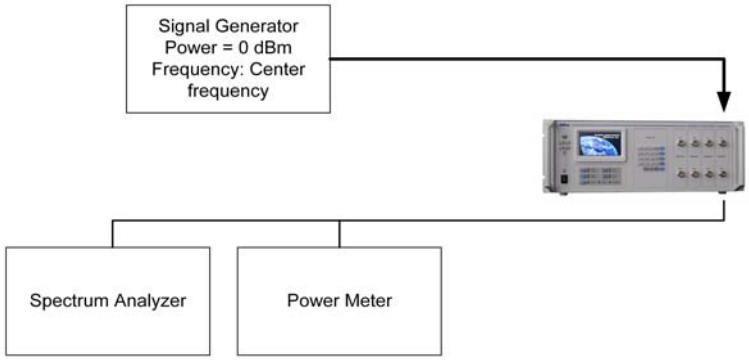


Figure A-1 Frequency Range Test Configuration

2. Measure the power; required output is -10dBm ± 3 dB. Using the spectrum analyzer, sweep frequency over the passband. The

spurious response to the signal should be less than or equal to -45 dBc.

3. Set the spectrum analyzer to 1 dB/div and sweep frequency over the passband. The flatness should be less than or equal to ±1.0 dB.

Attenuation testing

1. Connect the equipment as shown in Figure A-1. The SLE900 should be in static mode, with delay set to 2 msec and all other parameters set to 0.

2. Set signal and delta markers of the spectrum analyzer.

3. While monitoring the output of the spectrum analyzer, increase the attenuation parameter in 0.25 dB steps until you reach 40 dB.

4. For each step, the output signal should decrease by the corresponding amount, within ± 0.25 dB.

Frequency Offset testing

1. Connect the equipment as shown in Figure A-1, with the frequency counter replacing the spectrum analyzer. Set the SLE900 to static mode, with delay set to 2 msec and all other parameters set to 0.

2. Set the Doppler parameter to the settings in Table A-1. After each setting, use the frequency counter to verify that the signal has been offset by the indicated frequency.

Frequency shift (KHz)
0.001
3000
-0.001
-3000

Table A-1

Delay testing

1. Connect the SLE900 and testing equipment as shown in Figure A-2. Set the SLE900 to static mode, with delay set to 0.1 ms and all other parameters set to 0.

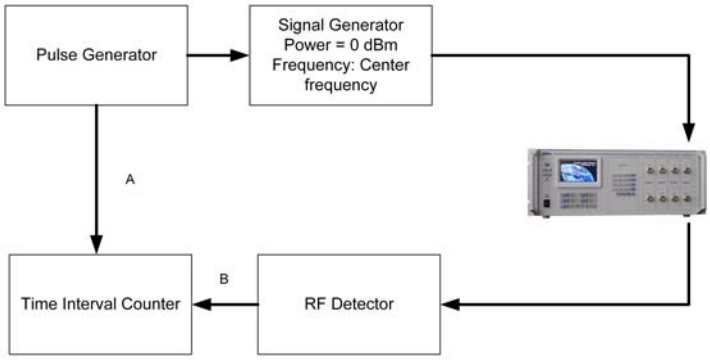


Figure A-2 Delay Test Configuration

2. Set the counter to time interval (A-B) mode.

3. Measure the delay to the output pulse.

4. Increase the SLE delay setting by 1 nanosecond. Measure the delay change.

5. Set the delay on the SLE to the maximum value and measure the delay.

6. Note: The SLE Static mode delay slew rate is 20 us/ms

Phase Offset Testing

1. Connect the equipment as shown in Figure A-3. The SLE900 should be in static mode, with delay set to 2 msec and all other parameters set to 0.

2. Using the network analyzer, increase the sweep time to 5 seconds.

3. Adjust the network analyzer electrical delay until the phase trace is horizontal.

4. Set the Phase Shift parameters to the settings in Table A-2. After each setting, use the network analyzer to verify that the signal has been offset by the indicated phase shift.

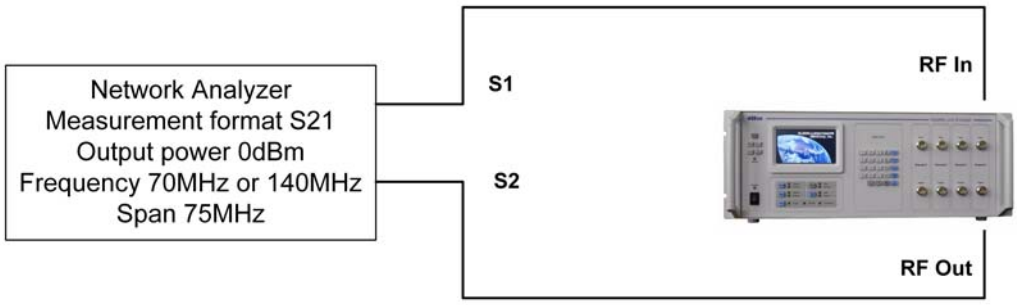


Figure A-3 Phase Offset testing

Phase shift
0
+1°
+190°
-1°
-190°

Table A-2 Phase Shift testing Values

Error Messages

Error messages are suppressed during remote operation.

File Missing

Appears when an instrument state register is recalled, and the stored file names no longer exist on the SLE900 internal storage.

Limit

Appears when a parameter value has exceeded the valid operation range.

B

Description and Specifications Section

Description and Specifications

This section describles the SLE900 technical details and specifications. Topics include:

- Functional decription of the instrument
- Instrument setting limitations
- Specifications

Functional Description

Delay Line Functions

The main function of the Digital Delay Line hardware is to delay an incoming analog signal by a desired time amount using a digital technique. To provide a relatively large amount of signal delay, the Digital Delay Line samples and converts an incoming analog signal into a 12-bit digital data stream, and stores the data in sequence into a large dynamic random access memory (DRAM) subsystem. After a desired amount of time is passed, the Digital Delay Line reads the stored data from the memory in sequence, and converts the digital data stream back to an analog signal.

For Dynamic operation, the magnitude of the delay is varied in real time by offsetting the clock that writes out of the delay line, relative to the fixed clock that reads into the FIFOs. The magnitude of the clock offset determines the rate of change of delay.

Delay Slew Rate and Resolution Limits

The delay slew rate has been limited, and will provide a rate of change in delay of no more than 20,000ns per update interval. In static mode, the default slew rate is 20us/ms. If the user selects a delay change exceeding this value in Static mode, then multiple update intervals are used to achieve the total desired delay change. The minimum delay slew rate in Static mode is 0.1ns/sec.

Delay resolution in the Dynamic mode is determined by the number of characters to the right of the decimal place in the ***first*** data value in a file. There are 5 possible resolution cases for delay:

Case #1:	ms.123456 invokes 1 nsec resolution
Case #2:	ms. 1234567 invokes 0.1 nsec resolution
Case #3:	ms. 12345678 invokes 0.01 nsec resolution
Case #4:	ms. 123456789 invokes 1 psec resolution
Case #5:	ms. 1234567890 invokes 0.5 psec resolution

Each resolution has a limited range of delay change that can occur during each update period. That range is from the resolution value up to 32,767 times the resolution value. For example, for Case #2 above having a resolution of 0.1 nsec, the minimum slew rate is 0.1 nsec per update period, and the maximum slew rate is 3276.7 nsec per update period. These ranges are valid for any update period from 1 msec to 1000 msec.

Specifications

Specifications are at IF unless otherwise noted.

Center Frequency	70 or 140 MHz
option	800 MHz -2600 MHz, tunable in 1 MHz steps
1 dB RF bandwidth	20 MHz/72 MHz/125 MHz/250 MHz
Number of independent channels	1,2, 3, or 4
RF input power	0 dBm max.
RF output power	0 dBm max. @ 0 dB attenuation
In-band spurious suppression	-55 dBc typ, -45 dBc max.
Noise floor	< -133 dBm/Hz
Group delay variation	< 3 nsec p-p
Passband Amplitude ripple	< 0.5 dB p-p
VSWR	1.2:1 max into 50 ohms
Delay	
Range	.1 ms to 2100 msec @ 20 MHz BW 1400 msec @ 72 MHz BW 940 msec @ 125 MHz BW 420 msec @ 250 MHz BW
Resolution	0.1 ns Static mode, 0.5 psec Dynamic mode
Slew rate	$3x10^{-15}$ sec/sec up to 20 us/ms
Relative accuracy	± 1 ns plus 10 MHz reference
Doppler	
Range	± 3.0 MHz
Resolution	1 Hz
Absolute accuracy	based on 10 MHz reference
Relative accuracy	± 0.1 Hz
Attenuation	
Range	0 dB to 40 dB
Resolution	0.25 dB
Slew rate	>40 dB/ms
Accuracy	± 0.25 dB + 3% of setting worst case
Control and Interface	
Local	Front panel
Remote	RJ45, IEEE-802.3
Real time update	1 ms to 1 sec
Data file size	to 5 million samples/file
Internal Frequency Reference error	< 2.5 PPM
Primary power	
Voltage	90 – 264 VAC autoranging
Frequency	48 – 66 Hz
Consumption	300 VA max.
Fuse	4A slow-blow
Operating ambient temperature	+10°C to +40°C
Dimensions	17”W x 7.0”H x 21”D

Specifications (cont'd)

Additive White Gaussian Noise

Sample Rate	90 MHz
Sample Depth	16 bits
Crest Factor	> 16 dB
Repetition Interval	>24 hrs
PDF accuracy	< tbd % from theoretical Gaussian over 6.66σ
Noise Bandwidth	1/2 sample rate
Spectral Density Flatness	< 0.1 dB p-p max
Noise Density Amplitude Range	Full scale to -42dB and off
Noise Density Amplitude Resolution	<=0.01 dB
Noise Density Amplitude Accuracy	<+/- 0.01 dB

Multipath Fading

Sample Rate	90 MHz
Sample Depth	16 bits
# of Paths	6
Path Characteristics	
Dynamic Profile update rate	10 msec to 1000 msec, affecting path on/off, attn, and delay
Probability Distribution Types	CW, Rayleigh, Ricean, and off
Spectral Distribution Shape (Ray, Rice)	{SQRT{1-(f/fd)^2}^-1 with 0 dB, 3 dB and 6 dB peak @ fd monontonic rolloff to less than -30 dB @ 2fd
PDF	within 0.5 dB of theoretical from 10 dB above to 30 dB below mean
Level Crossing Rate	<+/- 2.5% from theoretical
Correlation	0 to 100%, 1% steps
Ricean K factor	0 to 15 dB
Attenuation Range	0 to 30 dB
Attenuation Resolution	<=0.1 dB
Doppler Spread	0 to 12 kHz
Doppler Resolution	<=0.1 Hz
Delay Range	0 to 15 usec
Delay Resolution	1 ns
Output Summation	Sum of 6 paths plus AWGN with 12 bit signal, maintaining a minimum 12 bits output SNR.

Operating ambient temperature	+10°C to +40°C
Dimensions	17"W x 5.25"H x 21"D
Weight	
1 channel	*33lbs*
2 channel	*36lbs*
4 channel	*43lbs*

Figure B-1. SLE900 Specifications

C

Maintenance and Warranty Section

Maintenance and Warranty

This section describes the SLE900 maintenance procedures and warranty information.

- Maintenance Information.
- Warranty Information.

Maintenance Information

Adjustments and Calibration

To maintain optimum measurement performance, the SLE900 should be calibrated every year. It is recommended that the SLE900 be returned to dBm or to an authorized calibration facility. For more information please contact our Customer Service Department at (201) 677-0008.

Repair

The SLE900 should only be serviced by dBm service personnel or trained customer maintenance personnel using the dBm Service Manual for the SLE900.

For instruments requiring service, either in or out of warranty, contact dBm Customer Service Department at (201) 677-0008 for pricing and instructions before returning your instrument. When you call, be sure to have the following information available:

- Model number.
- Serial number.
- Full description of the failure condition.

Note: Model and serial number can be found on the rear of the SLE900 unit.

Equipment Returns

All instruments returned to dBm for repair must be shipped prepaid. Instruments that are eligible for in-warranty repair will be returned prepaid to the customer. For all other situations the customer is responsible for all shipping charges. An evaluation fee may be charged for processing units that are found to have no functional or performance defects.

For out of warranty instruments, dBm will provide an estimate for the cost of repair. Customer approval of the charges will be required before repairs can be made. For units deemed to be beyond repair, or in situations which the customer declines to authorize repair, an evaluation charge may be assessed by dBm.

Warranty Information

All dBm products are warranted against defects in material and workmanship for a period of one year from the date of shipment.

dBm will, at its option, repair or replace products that prove to be defective during the warranty period, provided they are returned to dBm and provided the preventative maintenance procedures are followed. Repairs necessitated by misuse of the product are not covered by this warranty. No other warranties are expressed or implied, including but not limited to implied warranties of merchantability and fitness for a particular purpose.

dBm is not liable for consequential damages. Please refer to the previous section for contact information and procedures to return the instrument to dBm.

D

SLE900 DLL Version 2.03

SLE900 DLL Description

This section describes the SLE900 Dynamic Link Library, used for remote Ethernet operation.

Description of the SLE900 DLL Application Program Interface Version 2.03

Overview:

The Dynamic Link Library (DLL) has been designed to provide the interface to an application program. This manual documents the commands available to the application programmer. The DLL creates the network connection between the application and the SLE. From that point the SLE is controlled solely by the DLL functions called by a client application. The DLL is a multitasking program designed to run on the Windows operating system. It will maintain the interface with the client application while controlling the SLE. Some of the functionality provided by the DLL is as follows:

1. Connect and Disconnect to the network
2. Obtain both SLE status or local DLL status
3. Convert ASCII data files to SLE compressed file format
4. Download/Delete files to/from SLE flash memory
5. Select/Deselect files to run in Dynamic mode
6. Issue keypad or button commands
7. Update parameter settings in static mode
8. Monitor data point position while running in Dynamic mode

The DLL has been designed to minimize the efforts to a client application, removing many of the details involved with controlling the SLE hardware. This manual assumes the reader is familiar with the capabilities, operation and control of the SLE product.

Linking C Programs with DLL:

There are three files of interest:

SleDll.h	Must be included in source code.
SleDll.dll	Must be installed in target directory.
SleDll.lib	Must be inserted in project builds to obtain DLL exported variables and functions.

DLL Data Types:

Data types used with all structures and SLE functions.

typedef void	**VOID;**	
typedef char	**CHAR;**	8 bits signed
typedef short	**SHORT;**	16 bits signed
typedef long	**LONG;**	32 bits signed
typedef unsigned char	**UCHAR;**	8 bits unsigned

typedef unsigned short	**USHORT;**	16 bits unsigned
typedef unsigned long	**ULONG;**	32 bits unsigned
typedef double	**DOUBLE;**	64 bit float

DLL/SLE Error Codes:

#define **SLE_NO_ERR**	0	No error. Function or command completed normally
#define **SLE_OPEN_ERR**	1	SLE already open
#define **SLE_SOCKET_ERR**	2	Unable to find usable WinSock DLL, network communication error
#define **SLE_THREAD_ERR**	3	Not able to create application timer thread, DLL inoperative
#define **SLE_SYSTEM_ERR**	4	General programming or operating system error
#define **SLE_CLOSE_ERR**	5	SLE already closed
#define **SLE_CONNECT_ERR**	6	Unable to connect to SLE
#define **SLE_ALREADY_CON_ERR**	7	Already connected, must first disconnect
#define **SLE_ALREADY_DISC_ERR**	8	Already disconnected, must first connect
#define **SLE_SWRITE_ERR**	9	Write on network socket operation to SLE failed
#define **SLE_SREAD_ERR**	10	Read of network socket operation from SLE failed
#define **SLE_SPARE1_ERR**	11	spare error, not defined
#define **SLE_TIMEOUT_ERR**	12	DLL timed out waiting for acknowledgement from SLE
#define **SLE_SPARE2_ERR**	13	spare error, not defined
#define **SLE_ARG_ERR**	14	Bad passed in argument to DLL function
#define **SLE_FILE_ERR**	15	DLL not able to open specified file
#define **SLE_OP_ERR**	16	DLL sent command to SLE properly however, SLE failed to perform operation. This could be for a variety of reasons, no room for file, file too large, SLE failed loading flash, SLE not functioning properly, etc.
#define **SLE_FREAD_ERR**	17	File read operation failed
#define **SLE_FWRITE_ERR**	18	File write operation failed
#define **SLE_SPARE3_ERR**	19	spare error, not defined
#define **SLE_FILE_FMT_ERR**	20	File format not as expected
#define **SLE_FRANGE_ERR**	21	File data point is not within expect range

DLL Keypad

Definitions:

Many DLL functions execute by mimicking the front panel keypad commands. These defines may be used as arguments to the functions where a key value is required.

#define **KEY_NONE**	0	
#define **KEY_PRESET**	1	
#define **KEY_INC**	2	
#define **KEY_DEC**	3	
#define **KEY_SCROLLUP**	4	
#define **KEY_SCROLLDOWN**	5	
#define **KEY_START**	6	
#define **KEY_RESET**	7	
#define **KEY_PAUSE**	8	
#define **KEY_UPDATE**	9	
#define **KEY_TIME_REF**	10	
#define **KEY_LOOP**	11	
#define **KEY_TRIGGER**	12	
#define **KEY_CHANNEL**	13	
#define **KEY_MODE**	14	
#define **KEY_LOCAL**	15	
#define **KEY_STEP**	16	
#define **KEY_FACTORY**	17	Factory use only
#define **KEY_CONFIG**	18	
#define **KEY_STORE**	19	
#define **KEY_RECALL**	20	
#define **KEY_FILE_MENU**	21	
#define **KEY_DISPLAY_FILE**	22	
#define **KEY_SELECT_FILE**	23	
#define **KEY_FREQCENTER**	24	
#define **KEY_DELAY**	25	
#define **KEY_FREQ_OFFSET**	26	
#define **KEY_ATTN**	27	
#define **KEY_PHASE_OFFSET**	28	
#define **KEY_UPDATE_INTERVAL**	29	
#define **KEY_FREQLO**	30	Factory use only
#define **KEY_FREQHI**	31	Factory use only
#define **KEY_NEG**	32	
#define **KEY_DECIMAL**	33	
#define **KEY_ENTER**	34	
#define **KEY_CLEAR**	35	
#define **KEY_KHZ**	36	
#define **KEY_MHZ**	37	
#define **KEY_ZERO**	38	
#define **KEY_ONE**	39	
#define **KEY_TWO**	40	
#define **KEY_THREE**	41	

#define **KEY_FOUR**	42	
#define **KEY_FIVE**	43	
#define **KEY_SIX**	44	
#define **KEY_SEVEN**	45	
#define **KEY_EIGHT**	46	
#define **KEY_NINE**	47	
#define **KEY_REMOTE**	48	

DLL API Function Argument Definitions:

SLEtestpattern(), SLEdownloadfile() function arguments
"Action" argument:
#define **DNLD_START** **0 start download**
#define **DNLD_STOP** **1 stop download**

SLEcnvtSATGENfile() function arguments
"FileType" arguments:
#define **TYPE_DLY** **0 Delay file**
#define **TYPE_FREQ** **1 Frequency File**
#define **TYPE_ATTN** **2 Attenuation File**
#define **TYPE_PHASE** **3 Phase File**
"InFileName" - xxx.dat file, complete path
"OutFileName" - xxx.sle file, complete path

DLL API Toggle Key Function Argument Definitions:

Update Clock toggle key Command
#define **CLK_INT** **0 internal clock**
#define **CLK_EXT** **1 external clock**

Mode toggle key Command
#define **MODE_STATIC** **0 operate in static mode**
#define **MODE_DYNAMIC** **1 operate in dynamic mode**

Loop toggle key Command
#define **LOOP_SINGLE** **0 single time**
#define **LOOP_CONTINUOUS** **1 continuous**
#define **LOOP_FWD_REVERSE** **2 forward/reverse**

Trigger toggle key Command
#define **TRIGGER_INT** **0 internal**
#define **TRIGGER_EXT** **1 external**

Channel toggle key Command
#define **CHAN1** **0**
#define **CHAN2** **1**
#define **CHAN3** **2**

#define **CHAN4** **3**

TimeRef toggle key Command
#define **TIMEREF_OFF** **0 off**

#define **TIMEREF_ON** **1 on**

Mode Status Definitions (0= Static, >=1 are all dynamic modes)
enum **MODE_STATUS**
{
M_STATIC, M_DYNAMIC, M_READY, M_RUN,
M_ARMED, M_PAUSE, M_DONE, M_INIT
};

SLE ParamCode Definitions:

SLEparam() function arguments
"ParamCode" argument:

#define **PARAM_DELAY**	**1**	
#define **PARAM_FREQOFFSET**	**2**	
#define **PARAM_ATTEN**	**3**	
#define **PARAM_PHASEOFFSET**	**4**	
#define **PARAM_FREQUENCY**	**5**	
#define **PARAM_UPDATEINTERVAL**	**6**	
#define **PARAM_STEP_DL**	**15**	step delay
#define **PARAM_STEP_FO**	**16**	step frequency offset
#define **PARAM_STEP_AT**	**17**	step attenuation
#define **PARAM_STEP_PO**	**18**	step phase offset

DLL Structure Definitions:

SLE Setup Data
default setup values
#define **SERVER_IP_ADDR** **"192.168.1.106"**
#define **SERVER_TCP_PORT** **5555** (limit range 1025-32767)
#define **SERVER_RESP_TIMEOUT** **5**

This setup information is required by the open command to establish connection to the SLE device.
typedef struct
{
CHAR **SleIpAddr[32]**; SLE Server IP Address
USHORT **SleTcpPort**; SLE Server Socket Port
USHORT **SleTimeout**; SLE Response Timeout in Secs
} **SLE_SETUP;**

SLE Channel Status Data

This status information may be returned by SLE after connection.
All parameters are specified in their base units.

```
typedef struct
{
        LONG  DelayTime;        nsecs
        LONG  FreqOffset;       hz
        LONG  Attenuation;      db (0-160 represents 0-40 in1/4  increments)
        LONG  PhaseOffset;      degrees (+-180)
        LONG  CenterFreq;       Mhz
} SLE_CHAN_STATUS;
```

SLE Status Info

This status information may be returned by SLE after connection.

```
typedef struct
{
        CHAR    SleVersion[8];  Current SLE App software version
        UCHAR   Channel;        Current active channel
        UCHAR   Clock;          UpdateSelect button setting
        UCHAR   Trigger;        TriggerSelect button setting
        UCHAR   Loop;           LoopSelect button setting
        UCHAR   TimeRef;        Time Reference button setting

        UCHAR   Mode;           Current operating mode (see enum
                                MODE_STATUS)
        USHORT  DnldFileCnt;    Number of files downloaded in flash
        UCHAR   ActiveChans;    Chans Detected (bit0=ch1, bit1=ch2,
                                 bit2=ch3, bit3=ch4)
        UCHAR   Spare;          unused
        SLE_CHAN_STATUS ChParams[4];    Channel parameter values
        LONG    UpdateInterval;         1-1000ms
        LONG    MinFrequencyRange;      configuration min base frequency
                                        (Mhz)
        LONG    MaxFrequencyRange;      configuration max base frequency
                                        (Mhz)
} SLE_STATUS;
```

SLE DLL Status Info

This status information is returned by the DLL, connection to SLE is not required.
Note: Once DLL status is read, all three error fields are reset to zero to avoid reading errors multiple times. Also, when StaticModeState reaches DONE, DLL resets it to IDLE.

```
typedef struct
```

```
{
        CHAR    DllVersion[8];          Current DLL software version
        UCHAR   LastError;              Last command error

        UCHAR   TimerThreadError;       Error results from DLL timer task
        UCHAR   StaticModeState;        Static Delay command operating
                                        state (0=IDLE, 1=BUSY,
                                        2=DONE)
        UCHAR   StaticModeError;        Error results from DLL Static
                                        Delay task
} SLE_DLL_STATUS;
```

SLE Download Status Data
This status information is returned by the SLE during a download operation.

```
typedef struct
{
        UCHAR   DownloadState;          0=IDLE, 1=BUSY, 2=DONE
        UCHAR   PercentComplete;        Download Percent Complete
        UCHAR   LastError;              Download Error (0=no error)
        UCHAR   Spare;                  keep word aligned
} SLE_DOWNLOAD_STATUS;
```

SLE Position Status Data
This status information is returned by the SLE during a SLEposition poll.

```
typedef struct
{
        ULONG   PointCnt;                       HW position of point in
file
        USHORT  Mode;                   mode to detect PAUSE or DONE
        USHORT  ElapsedTime;            elapsed time
} SLE_POSITION;
```

SLE Position32 Status Data
This status information is returned by the SLE during a SLEposition32 poll. (elapsed time has been lengthened from **SLE_POSITION)**

```
typedef struct
{
        ULONG   PointCnt;                       HW position of point in
file
        USHORT  Mode;                   mode to detect PAUSE or DONE
        ULONG   ElapsedTime;            elapsed time
} SLE_POSITION_32;
```

SLE File Preamble
File header information returned by the SLE due to a SLEpreamble command

```
typedef struct
{
        CHAR    FileName[16];           file name DLYxxxxxxx.dat (not
case sensitive)
```

```
        LONG    InitValue;          init value in ns
        ULONG   NumSamples;  number of sample points(up to 1.8 million)
        UCHAR   Continuous;  Continuous File
        UCHAR   Type;        0=DLY 1=FREQ 2=ATTN 3=PHASE
        USHORT Cksum;               Checksum of data part of file
        CHAR    Spare[100];  pad out structure to 128 bytes total
} SLE_PREAMBLE;
```

SLE Upload Status Data
This status information is returned by the SLE during a SLEupload.

```
typedef struct
{
        UCHAR  UploadState;         0=IDLE, 1=BUSY, 2=DONE
        UCHAR  PercentComplete;             Download Percent
Complete
        UCHAR  LastError;                   Download Error (0=no
error)
        UCHAR  Spare;
} SLE_UPLOAD_STATUS;
```

SLE Upload
This structure contains upload file information returned by the SLE during an SLE upload.

```
typedef struct
{
   LONG      NumberOfPoints; Number of points: continue with last data if
needed
   DOUBLE    MinX;
   DOUBLE    MaxX;
   DOUBLE    MinY;
   DOUBLE    MaxY;
   DOUBLE*   PointBuffer;          Pointer to allocated point buffer
   SHORT       Continuous;         Continuous File
} SLE_UPLOAD;
```

DLL Build and Export Definitions:

Below are definitions required for building the DLL and to be able to link DLL with a client application.

```
#ifdef __cplusplus
#define SLEDLL_FUNC_TYPE "C"        built in C++, force to reference
                                    as C
#else
#define SLEDLL_FUNC_TYPE            built in C, leave reference as C
#endif

#ifdef SLEDLL_EXPORTS               Set by developer only when
                                    Creating DLL library
#define SLEDLL_API extern SLEDLL_FUNC_TYPE
__declspec(dllexport)
```

```
#else
#define SLEDLL_API extern SLEDLL_FUNC_TYPE
__declspec(dllimport)
#endif
```

DLL Variables:

The following variables are exported by the DLL to be accessible by a client application.
DLL version may also be obtained from DLL status structure.
SLEDLL_API CHAR SleDLLversion[8];
Table of text strings associated with error codes.
SLEDLL_API CHAR *SleErrMsg[];

DLL Functions:

The following functions are exported by the DLL to be accessible by a client application.

SLEDLL_API SHORT **SLEopen(** SLE_SETUP *Setup **);**
SLEDLL_API SHORT **SLEclose(** VOID **);**
SLEDLL_API SHORT **SLEconnect(** VOID **);**
SLEDLL_API SHORT **SLEdisconnect(** VOID **);**
SLEDLL_API SHORT **SLEstatus(** SLE_STATUS *Status **);**
SLEDLL_API SHORT **SLEdllstatus(** SLE_DLL_STATUS *DllStatus **);**
SLEDLL_API SHORT **SLEtestpattern(** USHORT Action **);**
SLEDLL_API SHORT **SLEcnvtSATGENfile(** USHORT FileType, CHAR *InFileName, CHAR *OutFileName **);**
SLEDLL_API SHORT **SLEdownload(**CHAR *FileName,USHORT Action**);**
SLEDLL_API SHORT **SLEdnldstatus(** SLE_DOWNLOAD_STATUS *DnldStatus **);**
SLEDLL_API SHORT **SLEdeletefile(** CHAR AllFiles, CHAR *FileName **);**
SLEDLL_API SHORT **SLElistfile(** USHORT FileIndex, CHAR *FileBuf **);**
SLEDLL_API SHORT **SLEgetchans(** USHORT FileIndex, USHORT *FileChans **);**
SLEDLL_API SHORT **SLEsetchans(** USHORT FileIndex, USHORT FileChans **);**
SLEDLL_API SHORT **SLEkeypad(** USHORT NumKeys, UCHAR *KeyCodes **);**
SLEDLL_API SHORT **SLEparam(** UCHAR ParamCode**,** UCHAR Chan**,** LONG Value **);**
SLEDLL_API SHORT **SLEtoggle (** UCHAR Key, UCHAR Value **);**
SLEDLL_API SHORT **SLEposition(**UCHAR Timer, SLE_POSITION *SlePosition **);**
SLEDLL_API SHORT **SLEupload(** USHORT FileIndex, SLE_UPLOAD *SleUpload **);**
SLEDLL_API SHORT **SLEupldpreamble(** USHORT FileIndex, SLE_PREAMBLE *SlePreamble **);**
SLEDLL_API SHORT **SLEupldstatus(** SLE_UPLOAD_STATUS *UpldStatus **);**

All DLL functions return an error code. As a general note: Errors identified below are for the most part considered fatal errors. These errors are a result of a lost network connection and require the host program to reconnect. It is suggested to call SLEdisconnect() ignoring errors and then call SLEconnect() to re-establish connection. There should not be a need to close and reopen the interface but it would do no harm.

SLE_CONNECT_ERR
SLE_SWRITE_ERR
SLE_SREAD_ERR
SLE_SYSTEM_ERR
SLE_OP_ERR
SLE_TIMEOUT_ERR

Detailed Function Descriptions:

DLL function SLEopen:

SHORT **SLEopen(** SLE_SETUP ***Setup) –** This function opens the interface to the SLE application program and enables the DLL to configure for SLE operations. This must be the first function called. SLE application program sets up to communicate with a host device via Ethernet connection. A timer thread is created by the DLL to monitor the SLE and maintain connected status.

Input Arguments: **Setup** - Address to structure containing setup parameters (See structure definition **SLE_SETUP** for detailed setup parameters required) It is

recommended to use the default values for timeout and socket port. IP address must be set to the IP address of the SLE.

Output Argument: Returns **Error** code.

- **SLE_NO_ERR** – No error. Function completed normally.
- **SLE_OPEN_ERR** – SLE already open
  (Remedy- Issue a SLEclose() before attempting to open)
- **SLE_SOCKET_ERR** – Unable to find usable WinSock DLL.
  (Remedy- WS2_32.DLL may not exist on the client PC)
- **SLE_THREAD_ERR** – Not able to create application timer thread.
  (Remedy- PC lacks available resources, contact dBm service.)
- **SLE_SYSTEM_ERR** – General programming error, in this case, failed to create mutex operating system objects.
  (Remedy- PC lacks available resources, contact dBm service.)

DLL function SLEclose:

SHORT **SLEclose(** VOID **) –** This function closes the interface to the SLE application program. This must always be the last function called. SLE application program will disconnect and disassociate itself with the host application. The timer thread is terminated.

Input Arguments: none

Output Argument: Returns **Error** code.

- **SLE_NO_ERR** – No error. Function completed normally.
- **SLE_CLOSE_ERR** – SLE already closed
  (Remedy- Ignore error.)

DLL function SLEconnect:

SHORT **SLEconnect(** VOID **) –** This function is called to establish a network connection between the host and SLE. The host must have a valid open interface before connecting to the SLE. If the connection has been broken, new connections may be established without closing and reopening the interface. A successful connection will result in activity on the network. When system is idle, data packets are sent between the SLE and host every two seconds. Network activity may be determined by viewing the LED's on the network card.

Input Arguments: None

Output Argument: Returns **Error** code.

- **SLE_NO_ERR** – No error. Function completed normally.
- **SLE_SOCKET_ERR** – Unable to create communication socket
  (Remedy- PC lacks available resources, contact dBm service.)
- **SLE_CONNECT_ERR** – Unable to connect to SLE
  (Remedy- Assure SLE and client specify the same IP address)
- **SLE_ALREADY_CON_ERR** – SLE already connected
  (Remedy- Issue a SLEdisconnect() before attempting to connect)
- **SLE_SWRITE_ERR** – Write on network socket operation to SLE failed
  (Remedy- Lost connection, try to reconnect, if problem persists, contact dBm service.)
- **SLE_SREAD_ERR** – Read of network socket operation from SLE failed
  (Remedy- Lost connection, try to reconnect, if problem persists, contact dBm service.)
- **SLE_TIMEOUT_ERR** – DLL timed out waiting for acknowledgement from SLE
  (Remedy- Assure SLE is in Remote mode, not Local. Assure network connection. Lost connection, try to reconnect, if problem persists, contact dBm service.)
- **SLE_SYSTEM_ERR** – General programming error, in this case, system network select function failed.
  (Remedy- Error should not occur. Contact dBm service.)

DLL function SLEdisconnect:

SHORT **SLEdisconnect(** VOID **) –** This function terminates the network connection between the host and SLE.

Input Arguments: None

Output Argument: Returns **Error** code.

- **SLE_NO_ERR** – No error. Function completed normally.
- **SLE_ALREADY_DISC_ERR** – SLE is already disconnected.
  (Remedy- Ignore error.)

DLL function SLEstatus:

SHORT **SLEstatus(** SLE_STATUS ***Status) –** This function request status information from the SLE. See structure definition **SLE_STATUS** for supported status information. SLE must be connected to the client. On return, **Status** structure is updated with the latest status.

Input Arguments: **Status** - Address of **SLE_STATUS** structure

Output Argument: Returns **Error** code.

- **SLE_NO_ERR** – No error. Function completed normally.
- **SLE_CONNECT_ERR** – Lost connection between client and SLE
  (Remedy- Try reconnecting and re-issue command)
- **SLE_SWRITE_ERR** – Write on network socket operation to SLE failed
  (Remedy- Lost connection, try to reconnect, if problem persists, contact dBm service.)
- **SLE_SREAD_ERR** – Read of network socket operation from SLE failed
  (Remedy- Lost connection, try to reconnect, if problem persists, contact dBm service.)
- **SLE_SYSTEM_ERR** – General programming error, in this case, system network select function failed.
  (Remedy- Error should not occur. Contact dBm service.)
- **SLE_OP_ERR** – DLL sent command properly however, SLE failed to perform operation. SLE could not return status.
  (Remedy- SLE returned a NAK. Contact dBm service.)
- **SLE_TIMEOUT_ERR** – DLL timed out waiting for acknowledgement from SLE
  (Remedy- Assure SLE is in Remote mode, not Local. Assure network connection. Lost connection, try to reconnect, if problem persists, contact dBm service.)

DLL function SLEdllstatus:

SHORT **SLEdllstatus(SLE_DLL_STATUS *DllStatus) –** This function request status information from the DLL. See structure definition **SLE_DLL_STATUS** for supported status information. Unknown to the client application, the DLL is managing its own group of tasks. The DLL occasionally spawns tasks to perform SLE operations so that it may maintain the interface with the client. The DLL status contains the status of the spawned tasks when they exist. For example, if a Delay parameter setting is changed in static mode the DLL starts a static delay task. Since this particular command causes the SLE

hardware to run exclusively for as much as 35 seconds, the task is required to maintain the network connection and to monitor for the error response from the hardware when it completes. The client may request the DLL status and monitor the **StaticModeState** and **StaticModeError** fields for completion. Another example is the timer task. When the SLE is idle (not receiving commands), the timer task is sending an "I'm Alive" message to the SLE every 2 seconds. The client program may monitor the **TimerThreadError** field to know the state of the network connection. On return, **DllStatus** structure is updated with the latest DLL status.

Input Arguments: **DllStatus** - Address of **SLE_DLL_STATUS** structure

Output Argument: Returns **Error** code.

- **SLE_NO_ERR** – No error. Function completed normally.

DLL function SLEtestpattern:

SHORT **SLEtestpattern**(USHORT **Action**) – This function downloads an ASCII test pattern to the SLE. This function was created strictly for testing the Ethernet interface during program development. SLE does not do anything with the data except output it through its debug serial port if the serial port is enabled. This function is not recommended to exist in a client application.

DLL function SLEcnvtSATGENfile:

SHORT **SLEcnvtSATGENfile**(USHORT **FileType**, CHAR ***InFileName**, CHAR ***OutFileName**)

This function converts a SATGEN ".DAT" data file to a ".SLE" formatted file. The conversion process creates a preamble structure in the beginning of the file followed by an end of file marker, the point data and another end of file marker. The first end of file marker is used when the file is played back in the reverse direction. The point data is converted from absolute coordinates to relative offsets to the next point. This conversion reduces the format size, thus allowing for more files to fit in the SLE flash memory. Relative points are verified to not exceed file limits. The preamble structure contains 128 reserved bytes of the following format:

```
struct FilePreamble
{
  CHAR      FileName[16];  file name DLYxxxxxxx.dat (not case
                           sensitive)
  LONG      InitValue;     init value in ns

  ULONG     NumSamples;  number of sample points(up to 1.8 million)
  UCHAR     Continuous;  Continuous File
  UCHAR     Type;        0=DLY 1=FREQ 2=ATTN 3=PHASE
  USHORT    Cksum;       Checksum of file data part
  CHAR      Spare[100];  pad out structure to 128 bytes total
};
```

The Preamble is created and inserted by the DLL. All unused bytes are set to zero. Application programs work strictly with raw SATGEN data files, not ".SLE" files.

Units for the point data varies depending on the file type:

DLY- Delay files contain 16 bit data points with the range +- 32767 ns
End of File value is 8000hex;
FRQ- Frequency files contain 16 bit data points with the range +- 32767 hz
End of File value is 8000hex;
ATN- Attenuation files contain 8 bit data points ranging from 0-160 (.25 db)
End of File value is FFhex;
PHA- Phase files contain 8 bit data points with the range +- 127 degrees

End of File value is 80hex;

SLEcnvtSATGENfile Function Arguments:

Input Argument: **FileType**– TYPE_DLY, TYPE_FREQ, TYPE_ATTN or TYPE_PHASE

Input Argument: **InFileName**– Pointer to ASCII string containing full path of input .dat file.

Input Argument: **OutFileName**– Pointer to ASCII string containing full path of output .sle file.

Output Argument: Returns **Error** code.

- **SLE_NO_ERR** – No error. Function completed normally.
- **SLE_FILE_ERR** – DLL not able to open input or output file
  (Remedy- Verify file name and path. Check file permissions.)
- **SLE_FREAD_ERR** – Input file read operation failed
  (Remedy- Check file permissions and file format.)
- **SLE_FWRITE_ERR** – Output file write operation failed
  (Remedy- Check file permissions and file format.)
- **SLE_SYSTEM_ERR** – General programming error, in this case, system file function failed.
  (Remedy- Error should not occur. Contact dBm service.)
- **SLE_FRANGE_ERR** – Point data exceeds the expected value. Original data has not been created properly.
  (Remedy- Fix the data in the original .DAT file.)

DLL function SLEdownload:

SHORT **SLEdownload(** CHAR ***FileName**, USHORT **Action) –** This function downloads a SATGEN (.DAT) file that has been converted to SLE format

(.SLE) into the flash memory of the SLE. ".SLE" files are SATGEN files that have been converted from fixed point to relative offsets to the next point. ".SLE" files also contain a definition preamble structure in a format known by the SLE. The DLL first sends the file preamble packet to SLE. SLE determines if room exists to accept the download and responds with an ACK or NAK. If the DLL receives an ACK response, the download continues by sending fixed length packets 8KBytes in length to the SLE until complete. The SLE restrains the DLL from sending data while writing packets to flash on 64 Kbyte boundaries. A NAK response from the SLE at any time causes the DLL to terminate the download. Also, the download can be aborted by sending the download command with the argument **DNLD_STOP**.

Input Arguments: **Filename** – Pointer to ASCII string containing download file name. Valid file names must start with "DLY", "FRQ", "ATN, or "PHA" and end with ".SLE" extension. Valid file names must not exceed 14 characters in length. File names are not case sensitive. Filename argument contains the complete path followed by the 14 character file name. The DLL does not error check file names before downloading. The application must assure correct file format.

Input Arguments: **Action - DNLD_START** to start download. **DNLD_STOP** to abort download.

Output Argument: Returns **Error** code.

- **SLE_NO_ERR** – No error. Function completed normally.
- **SLE_CONNECT_ERR** – Lost connection between client and SLE
  (Remedy- Try reconnecting and re-issue command)
- **SLE_THREAD_ERR** – Not able to create DLL download thread
  (Remedy- PC lacks available resources, contact dBm service.)
- **SLE_FILE_ERR** – DLL not able to open specified file
  (Remedy- Verify file name and path. Check file permissions.)

- **SLE_FREAD_ERR** – File read operation failed
  (Remedy- Check file permissions and file format.)
- **SLE_SWRITE_ERR** – Write on network socket operation to SLE failed
  (Remedy- Lost connection, try to reconnect, if problem persists, contact dBm service.)
- **SLE_SREAD_ERR** – Read of network socket operation from SLE failed
  (Remedy- Lost connection, try to reconnect, if problem persists, contact dBm service.)
- **SLE_SYSTEM_ERR** – General programming error, in this case, system network select function failed.

  (Remedy- Error should not occur. Contact dBm service.)
- **SLE_OP_ERR** – DLL sent file properly however, SLE failed to complete operation. SLE could not accept download.
  (Remedy- SLE returned a NAK. Try download again, if error does not clear, contact dBm service. Assure SLE flash is not full.)
- **SLE_TIMEOUT_ERR** – DLL timed out waiting for acknowledgement from SLE
  (Remedy- System select function timed out waiting for ACK/NAK response.

Try reconnecting and execute download again, if error does not clear, contact dBm service.)

DLL function SLEdnldstatus:

SHORT **SLEdnldstatus(** SLE_DOWNLOAD_STATUS ***DnldStatus) –** This function request status information for a download in progress. See structure definition **SLE_DOWNLOAD_STATUS** for supported status information. SLE must be connected to the client. On return, **DnldStatus** structure is updated with the latest status.

Input Arguments: **DnldStatus** - Address of **SLE_DOWNLOAD_STATUS** structure

Ouput Argument: Returns **Error** code.

- **SLE_NO_ERR** – No error. Function completed normally.
- **SLE_CONNECT_ERR** – Lost connection between client and SLE
  (Remedy- Try reconnecting and re-issue command)
- **SLE_SWRITE_ERR** – Write on network socket operation to SLE failed
  (Remedy- Lost connection, try to reconnect, if problem persists, contact dBm service.)
- **SLE_SREAD_ERR** – Read of network socket operation from SLE failed
  (Remedy- Lost connection, try to reconnect, if problem persists, contact dBm service.)
- **SLE_SYSTEM_ERR** – General programming error, in this case, system network select function failed.
  (Remedy- Error should not occur. Contact dBm service.)
- **SLE_OP_ERR** – DLL sent command properly however, SLE failed to perform operation. SLE could not return status.
  (Remedy- SLE returned a NAK. Contact dBm service.)
- **SLE_TIMEOUT_ERR** – DLL timed out waiting for acknowledgement from SLE

(Remedy- Assure SLE is in Remote mode, not Local. Assure network connection. Lost connection, try to reconnect, if problem persists, contact dBm service.)

DLL function SLEdeletefile:

SHORT **SLEdeletefile(** CHAR **AllFiles**, CHAR ***FileName) –** This function causes the SLE to remove files from its flash memory. This is the mechanism to make room for more file downloads.

Input Arguments: **AllFiles** – 1 = delete all files in flash, 0=delete file specified by second argument.

Input Arguments: **FileName** – Pointer to ASCII string containing file name to delete. Valid file names must start with “DLY”, “FRQ’, “ATN, or “PHA” and end with “.DAT” extension. Valid file names must not exceed 14

characters in length. File names are not case sensitive. This argument contains just the file name, no path.

Output Argument: Returns **Error** code.

- **SLE_NO_ERR** – No error. Function completed normally.
- **SLE_CONNECT_ERR** – Lost connection between client and SLE
  (Remedy- Try reconnecting and re-issue command)
- **SLE_SWRITE_ERR** – Write on network socket operation to SLE failed
  (Remedy- Lost connection, try to reconnect, if problem persists, contact dBm service.)
- **SLE_SREAD_ERR** – Read of network socket operation from SLE failed
  (Remedy- Lost connection, try to reconnect, if problem persists, contact dBm service.)
- **SLE_SYSTEM_ERR** – General programming error, in this case, system network select function failed.
  (Remedy- Error should not occur. Contact dBm service.)
- **SLE_OP_ERR** – DLL sent command properly however, SLE failed to perform operation. SLE could not perform file delete.
  (Remedy- SLE returned a NAK. Verify file exists in SLE flash memory.)

- **SLE_TIMEOUT_ERR** – DLL timed out waiting for acknowledgement from SLE
  (Remedy- System select function timed out waiting for ACK/NAK response. Try reconnecting and execute file delete command again, if error does not clear, contact dBm service.)

DLL function SLElistfile:

SHORT **SLElistfile(** USHORT **FileIndex**, CHAR ***FileBuf) –** This function causes the SLE to return the name of the file loaded in flash memory associated with a file index.
The SLE maintains a count and order of files downloaded. Client may read the "**DnldFileCnt**" variable of **SLE_STATUS** to know how many filenames to request.

Any time a file is added to SLE flash or deleted from flash the file indexes will change. For any change of files in flash this function must be called to obtain the indexes for all files. The client uses the returned file indexes as arguments to the **SLEsetchans** or **SLEgetchans** functions.

Input Argument: **FileIndex**– File index position, this is a index maintained by the SLE. The SLE will use this value as an index into the SLE directory table.

Input Argument: **FileBuf -** Pointer to buffer for DLL to store found filename. Buffer must be a minimum of 16 characters in length. If no file exists for the specified FileBuf , DLL will return a filename string of size 0. The filename returned by this function has no path or extension associated with it. A downloaded file of "c:\filedir\DLYfile1.sle" will be return by this function as "DLYfile1".

Output Argument: Returns **Error** code.

- **SLE_NO_ERR** – No error. Function completed normally.
- **SLE_CONNECT_ERR** – Lost connection between client and SLE
  (Remedy- Try reconnecting and re-issue command)
- **SLE_SWRITE_ERR** – Write on network socket operation to SLE failed
  (Remedy- Lost connection, try to reconnect, if problem persists, contact dBm service.)
- **SLE_SREAD_ERR** – Read of network socket operation from SLE failed
  (Remedy- Lost connection, try to reconnect, if problem persists, contact dBm service.)

- **SLE_SYSTEM_ERR** – General programming error, in this case, system network select function failed.
  (Remedy- Error should not occur. Contact dBm service.)
- **SLE_OP_ERR** – DLL sent command properly however, SLE failed to perform operation. SLE could not obtain filename.
  (Remedy- SLE returned a NAK. **FileIndex** is not a valid value, should never exceed "**DnldFileCnt**" -1.)
- **SLE_TIMEOUT_ERR** – DLL timed out waiting for acknowledgement from SLE
  (Remedy- System select function timed out waiting for ACK/NAK response. Try reconnecting and execute listfile command again, if error does not clear, contact dBm service.)

DLL function SLEgetchans:

SHORT **SLEgetchans(** USHORT **FileIndex**, USHORT ***FileChans) –** This function returns a bitmask (lower 4 bits) of enabled/disabled channels associated with a downloaded file. An enabled channel indicates that the file has been selected to execute when running in dynamic mode. A disabled channel is an un-selected file.

Input Argument: **FileIndex**– File index position, this is a sequential index count maintained by the SLE. The SLE will use this value to reference the correct file channel status table.

Input Argument: **FileChans -** Pointer to variable to store found file channel status. Bit0=ch1, Bit1=ch2, Bit2=ch3, Bit3=ch4, all other bits are not used.

Output Argument: Returns **Error** code.

- **SLE_NO_ERR** – No error. Function completed normally.
- **SLE_CONNECT_ERR** – Lost connection between client and SLE
  (Remedy- Try reconnecting and re-issue command)
- **SLE_SWRITE_ERR** – Write on network socket operation to SLE failed
  (Remedy- Lost connection, try to reconnect, if problem persists, contact dBm service.)

- **SLE_SREAD_ERR** – Read of network socket operation from SLE failed
  (Remedy- Lost connection, try to reconnect, if problem persists, contact dBm service.)
- **SLE_SYSTEM_ERR** – General programming error, in this case, system network select function failed.
  (Remedy- Error should not occur. Contact dBm service.)
- **SLE_OP_ERR** – DLL sent command properly however, SLE failed to perform operation. SLE could not obtain **FileChans**.
  (Remedy- SLE returned a NAK. **FileIndex** is not a valid value, should never exceed "**DnldFileCnt**" -1. Also an error may occur if the SLE found no file associated with the **FileIndex.** Verify command arguments.)
- **SLE_TIMEOUT_ERR** – DLL timed out waiting for acknowledgement from SLE
  (Remedy- System select function timed out waiting for ACK/NAK response. Try reconnecting and execute getchans command again, if error does not clear, contact dBm service.)

DLL function SLEsetchans:

SHORT **SLEsetchans(** USHORT **FileIndex**, USHORT **FileChans) –** This function associated files selected to execute in dynamic mode with all 4 channels. The SLE is set to the values specified by the client. The **FileChans** argument must contain a bitmask to reflect selection values in the following bit order: Bit0=ch1, Bit1=ch2, Bit2=ch3, Bit3=ch4, all other bits are not used. A bit value of 1 indicates file is enabled and 0 indicates file is disabled to run.

Input Argument: **FileIndex**– File index position, this is a sequential index count maintained by the SLE. The SLE will use this value to index the channel status table.

Input Argument: **FileChans –** File channel status to update to.

Output Argument: Returns **Error** code.

- **SLE_NO_ERR** – No error. Function completed normally.
- **SLE_CONNECT_ERR** – Lost connection between client and SLE
  (Remedy- Try reconnecting and re-issue command)
- **SLE_SWRITE_ERR** – Write on network socket operation to SLE failed
  (Remedy- Lost connection, try to reconnect, if problem persists, contact dBm service.)
- **SLE_SREAD_ERR** – Read of network socket operation from SLE failed
  (Remedy- Lost connection, try to reconnect, if problem persists, contact dBm service.)
- **SLE_SYSTEM_ERR** – General programming error, in this case, system network select function failed.
  (Remedy- Error should not occur. Contact dBm service.)
- **SLE_OP_ERR** – DLL sent command properly however, SLE failed to perform operation. SLE could not set SLE to **FileChans**.
  (Remedy- SLE returned a NAK. **FileIndex** is not a valid value, should never exceed "**DnldFileCnt**" -1. Also an error may occur if the SLE found no file associated with the **FileIndex.** Verify command arguments.)

- **SLE_TIMEOUT_ERR** – DLL timed out waiting for acknowledgement from SLE
  (Remedy- System select function timed out waiting for ACK/NAK response. Try reconnecting and execute setchans command again, if error does not clear, contact dBm service.)

DLL function SLEkeypad:

SHORT **SLEkeypad(** USHORT **NumKeys**, UCHAR ***KeyCodes) –** This function causes the SLE to mimic a front panel key operations. Care must be taken to use the same key sequence that would be entered by the operator at the front panel. For example, some keys require a terminating enter key and some do not:

Example1:
Perform a Store to Settings #0
UCHAR keys[] = {KEY_STORE, KEY_ZERO, KEY_ENTER};
SLEkeypad (3, keys);

Example2:
Start Dynamic Mode
UCHAR keys[] = {KEY_START};
SLEkeypad (1, keys);

It is important to note that there are only certain times that the SLE polls for keys and keys are accepted and processed one at a time. Therefore the keypad command will return not knowing if the key sequence was accepted and processed properly by the SLE. Due to the large number of possible keypad sequences, the SLE does not error check the validity of each command. The return status in this case just implies the SLE received a proper formatted keypad command. The contents of the command will execute after the SLE responds to the network. The programmer must be cautious to debug the use of keypad commands thoroughly one at a time before stringing multiple commands. If commands are complex, it is suggested to use separate keypad commands. If the SLE is busy executing a keypad command and a second one is sent to early, the SLE will reject the second command. The application may require a delay between multiple keypad commands depending on the complexity of the commands.

Input Argument: **NumKeys**– This argument indicates the number of **KeyCodes** in the character buffer to process. Maximum size is 100. SLE buffer size is 100.

Input Argument: **KeyCodes** – This argument is a pointer to a buffer of key codes to process. The SLE will process the key codes in the same manner as if the user entered them at the front panel.

Output Argument: Returns **Error** code.

- **SLE_NO_ERR** – No error. Function completed normally.
- **SLE_CONNECT_ERR** – Lost connection between client and SLE
  (Remedy- Try reconnecting and re-issue command)
- **SLE_SWRITE_ERR** – Write on network socket operation to SLE failed

(Remedy- Lost connection, try to reconnect, if problem persists, contact dBm service.)

- **SLE_SREAD_ERR** – Read of network socket operation from SLE failed
  (Remedy- Lost connection, try to reconnect, if problem persists, contact dBm service.)

- **SLE_SYSTEM_ERR** – General programming error, in this case, system network select function failed.
  (Remedy- Error should not occur. Contact dBm service.)
- **SLE_OP_ERR** – DLL sent command properly however, SLE failed to perform operation. SLE could not perform keypad operation.
  (Remedy- SLE returned a NAK. Verify function arguments.)
- **SLE_TIMEOUT_ERR** – DLL timed out waiting for acknowledgement from SLE
  (Remedy- System select function timed out waiting for ACK/NAK response. Try reconnecting and execute keypad command again, if error does not clear, contact dBm service.)

DLL function SLEparam:

This data is for setting parameter values in static mode.

SHORT **SLEparam(** UCHAR **ParamCode,** UCHAR **Chan,** LONG **Value) –** This function sets channel parameters. Allowable parameters are:

Delay: (PARAM_DELAY)	May take as long as 35 secs
Frequency Offset:	(PARAM_FREQOFFSET)
Attenuation:	(PARAM_ATTEN)
Phase Offset:	**(PARAM_PHASEOFFSET)**
Center Frequency:	(PARAM_FREQUENCY) Only valid for systems with external frequency converters
Update Interval:	(PARAM_UPDATEINTERVAL)
Step Delay:	(PARAM_STEP_DL)
Step_Frequency Offset:	(PARAM_STEP_FO)
Step_Attenuation:	(PARAM_STEP_AT)
Step_Phase Offset:	**(PARAM_STEP_PO)**

All parameters will update immediately and return an error code except for Delay. The Delay parameter is treated as an exception by the DLL due to the time it takes to execute. Changing the Delay parameter causes the SLE hardware slew to the new value. During this time, the SLE cannot accept network commands. The DLL is running a separate process solely to monitor for the hardware to complete. The DLL maintains the interface and status to the client to avoid forcing the client into a wait loop. The client may periodically check the DLL status error results **StaticModeState or StaticModeError** to know when the hardware has completed. The DLL sets a 60 second timeout error code if processing exceeds this time. The SLE verifies parameters to be within a specified range and will not exceed minimum and maximum limits that have been hard coded.
The parameter PARAM_UPDATEINTERVAL is restricted to the following values specified in msecs:

1, 2, 5, 10, 20, 50, 100, 200, 500, 1000

If any other value other then these ten are used, the SLE will select the closest allowable value.

Example to set Frequency Offset to 1000 Hz for Channel 1:
err = SLEparam(PARAM_FREQOFFSET, CHAN1, 1000);
Replaces:

```
UCHAR keys[] = {KEY_FREQ_OFFSET, KEY_ONE, KEY_ZERO,
KEY_ZERO, KEY_ZERO , KEY_ENTER};

err = SLEtoggle( KEY_CHANNEL, CHAN1 );
err = SLEkeypad ( 6, keys );
```

Input Argument: **ParamCode**– This argument is the parameter code associated with parameter.

Input Argument: **Chan**– This argument specifies associated channel

Input Argument: **Value**– This argument specifies the parameter value in base units.

Output Argument: Returns Error code.

- **SLE_NO_ERR** – No error. Function completed normally.
- **SLE_CONNECT_ERR** – Lost connection between client and SLE
  (Remedy- Try reconnecting and re-issue command)
- **SLE_SWRITE_ERR** – Write on network socket operation to SLE failed
  (Remedy- Lost connection, try to reconnect, if problem persists, contact dBm service.)
- **SLE_SREAD_ERR** – Read of network socket operation from SLE failed
  (Remedy- Lost connection, try to reconnect, if problem persists, contact dBm service.)
- **SLE_SYSTEM_ERR** – General programming error, in this case, system network select function failed.
  (Remedy- Error should not occur. Contact dBm service.)
- **SLE_OP_ERR** – DLL sent command properly however, SLE failed to perform operation. SLE could not perform param operation.

(Remedy- SLE returned a NAK. Verify function arguments.)

- **SLE_TIMEOUT_ERR** – DLL timed out waiting for acknowledgement from SLE
  (Remedy- System select function timed out waiting for ACK/NAK response. Try reconnecting and execute param command again, if error does not clear, contact dBm service.)

DLL function SLEtoggle:

SHORT **SLEtoggle(** UCHAR **Key**, UCHAR **Value) –** Thisis a macro helper function for accessing the front panelkeys immediately below the VFD display.

These keys are known as toggle keys. They toggle between a fixed set of values. These keys may also be used with the keypad command however their usage would be slightly more difficult.

Input Argument: **Key**– This argument indicates key code to modify. Allowable toggle keys:

KEY_UPDATE
KEY_TIME_REF
KEY_LOOP
KEY_TRIGGER
KEY_CHANNEL
KEY_MODE.

Input Argument: **Value** – This argument specifies the value to set the key to. (Reference above section labeled "**Toggle Key Defines"** to obtain allowable values.)

Output Argument: Returns **Error** code.

- **SLE_NO_ERR** – No error. Function completed normally.
- **SLE_CONNECT_ERR** – Lost connection between client and SLE
  (Remedy- Try reconnecting and re-issue command)
- **SLE_SWRITE_ERR** – Write on network socket operation to SLE failed
  (Remedy- Lost connection, try to reconnect, if problem persists, contact dBm service.)
- **SLE_SREAD_ERR** – Read of network socket operation from SLE failed
  (Remedy- Lost connection, try to reconnect, if problem persists, contact dBm service.)
- **SLE_SYSTEM_ERR** – General programming error, in this case, system network select function failed.
  (Remedy- Error should not occur. Contact dBm service.)
- **SLE_OP_ERR** – DLL sent command properly however, SLE failed to perform operation. SLE could not perform toggle operation. Either an invalid toggle key was specified or the value is not within the allowable range.
  (Remedy- SLE returned a NAK. Verify command arguments.)
- **SLE_TIMEOUT_ERR** – DLL timed out waiting for acknowledgement from SLE
  (Remedy- System select function timed out waiting for ACK/NAK response. Try reconnecting and execute toggle command again, if error does not clear, contact dBm service.)

DLL function SLEposition:

SHORT **SLEposition(** UCHAR **Timer,** SLE_POSITION ***SlePosition) -** This function reads the position of the point being replayed when running in dynamic mode. A client application may periodically read this value to synchronize a point cursor on a graphical display.
The position is a running count from the start of replay. Client must take into account the position of the KEY_LOOP toggle key when calculating position. Must consider direction changes due to fwd/rev, continuous mode and number of points in each direction. On return, **SlePosition** structure is updated with the latest point count, mode and elapsed time information from the SLE.

Input Argument: **Timer** – This is a timer disable argument. 1=disable timer on the SLE, 0=re-enable timer on the SLE. The default at SLE power-up is the timer is on. The SLE uses the timer to detect when it has lost a network connection. Normally when the SLE is not running a dynamic file, the DLL is sending an "I'm Alive" message every two seconds. In the case when using the **SLEposition** command since it is typically sent at a one second rate, a returned failure is enough to indicate a network communication problem. The SLEposition command is sent in place of the "I'm Alive" command. Setting the timer to off during the polling of **SLEposition** simply cuts down overhead at the SLE. Once the dynamic file stops running, as the last step, the timer should then be re-enabled.

Input Argument: **SlePosition** – Address where DLL stores positional status data.

Output Argument: Returns **Error** code.

- **SLE_NO_ERR** – No error. Function completed normally.
- **SLE_CONNECT_ERR** – Lost connection between client and SLE
  (Remedy- Try reconnecting and re-issue command)
- **SLE_SWRITE_ERR** – Write on network socket operation to SLE failed
  (Remedy- Lost connection, try to reconnect, if problem persists, contact dBm service.)
- **SLE_SREAD_ERR** – Read of network socket operation from SLE failed
  (Remedy- Lost connection, try to reconnect, if problem persists, contact dBm service.)
- **SLE_SYSTEM_ERR** – General programming error, in this case, system network select function failed.
  (Remedy- Error should not occur. Contact dBm service.)
- **SLE_OP_ERR** – DLL sent command properly however, SLE failed to perform operation. SLE could not obtain position value.
  (Remedy- SLE returned a NAK. Verify command arguments.)
- **SLE_TIMEOUT_ERR** – DLL timed out waiting for acknowledgement from SLE
  (Remedy- System select function timed out waiting for ACK/NAK response. Try reconnecting and execute position command again, if error does not clear, contact dBm service.)

DLL function SLEupload:

SHORT **SLEupload(** USHORT **FileIndex**, SLE_UPLOAD ***SleUpload) –**
This function uploads the point data of file resident in flash on the SLE into a client allocated buffer. During the upload process the SLE relative file is converted to an absolute file format that is accessible by the client application. (Effectively converting back to the original .DAT file format). File uploads packet sizes are fixed at 1456 bytes. The upload process performs all conversions of the data from the SLE resident interleaved format.
A NAK response from the SLE at any time causes the DLL to terminate the upload.

Input Argument: **FileIndex**– File index position, this is a sequential index count maintained by the SLE. The SLE will use this value to index the file table.

Input Arguments: **SleUpload** – Pointer to the structure SLE_UPLOAD. Address where DLL stores uploaded file point data.

Ouput Argument: Returns **Error** code.

- **SLE_NO_ERR** – No error. Function completed normally.
- **SLE_CONNECT_ERR** – Lost connection between client and SLE
  (Remedy- Try reconnecting and re-issue command)
- **SLE_THREAD_ERR** – Not able to create DLL upload thread
  (Remedy- PC lacks available resources, contact dBm service.)
- **SLE_FILE_ERR** – DLL not able to open specified file
  (Remedy- Verify file name and path. Check file permissions.)
- **SLE_FREAD_ERR** – File read operation failed
  (Remedy- Check file permissions and file format.)
- **SLE_SWRITE_ERR** – Write on network socket operation to SLE failed
  (Remedy- Lost connection, try to reconnect, if problem persists, contact dBm service.)
- **SLE_SREAD_ERR** – Read of network socket operation from SLE failed
  (Remedy- Lost connection, try to reconnect, if problem persists, contact dBm service.)
- **SLE_SYSTEM_ERR** – General programming error, in this case, system network select function failed.
  (Remedy- Error should not occur. Contact dBm service.)
- **SLE_OP_ERR** – DLL sent command properly however, SLE failed to complete upload operation.
  (Remedy- SLE returned a NAK. Try upload again, if error does not clear, contact dBm service.)
- **SLE_TIMEOUT_ERR** – DLL timed out waiting for acknowledgement from SLE
  (Remedy- System select function timed out waiting for ACK/NAK response. Try reconnecting and execute upload again, if error does not clear, contact dBm service.)

DLL function SLEupldpreamble:

SHORT **SLEupldpreamble(** USHORT **FileIndex**, SLE_PREAMBLE ***SlePreamble) -** This function request the preamble information for a specified file on resident on the

SLE. See structure definition **SLE_PREAMBLE** for detailed member definitions. SLE must be connected to the client. On return, **SlePreamble** structure is updated with the latest preamble information.
The client program may issue this command before an upload operation to obtain the number of samples in order to calculate the size of the sample buffer to allocate.

Input Argument: **FileIndex**– File index position, this is a sequential index count maintained by the SLE. The SLE will use this value to index the file table.

Input Arguments: **SlePreamble** – Pointer to the structure SLE_SLE_PREAMBLE, address where DLL stores file preamble information.

Ouput Argument: Returns **Error** code.

- **SLE_NO_ERR** – No error. Function completed normally.
- **SLE_CONNECT_ERR** – Lost connection between client and SLE
  (Remedy- Try reconnecting and re-issue command)
- **SLE_SWRITE_ERR** – Write on network socket operation to SLE failed
  (Remedy- Lost connection, try to reconnect, if problem persists, contact dBm service.)
- **SLE_SREAD_ERR** – Read of network socket operation from SLE failed
  (Remedy- Lost connection, try to reconnect, if problem persists, contact dBm service.)
- **SLE_SYSTEM_ERR** – General programming error, in this case, system network select function failed.
  (Remedy- Error should not occur. Contact dBm service.)
- **SLE_OP_ERR** – DLL sent command properly however, SLE failed to return preamble information.
  (Remedy- SLE returned a NAK. Try command again, if error does not clear, contact dBm service.)
- **SLE_TIMEOUT_ERR** – DLL timed out waiting for acknowledgement from SLE
  (Remedy- System select function timed out waiting for ACK/NAK response. Try reconnecting and execute command again, if error does not clear, contact dBm service.)

DLL function SLEupldstatus:

SHORT **SLEupldstatus(** SLE_UPLOAD_STATUS ***UpldStatus) –** This function request status information for an upload operation that is in progress. See structure definition **SLE_UPLOAD_STATUS** for supported status information. SLE must be connected to the client. On return, **UpldStatus** structure is updated with the latest status.
Client program may issue this command to monitor for either completion of the upload or an error condition.

Input Arguments: **UpldStatus** - Address of **SLE_UPLOAD_STATUS** structure

Ouput Argument: Returns **Error** code.

- **SLE_NO_ERR** – No error. Function completed normally.
- **SLE_CONNECT_ERR** – Lost connection between client and SLE
  (Remedy- Try reconnecting and re-issue command)
  - **SLE_SWRITE_ERR** – Write on network socket operation to SLE failed
    (Remedy- Lost connection, try to reconnect, if problem persists, contact dBm service.)
  - **SLE_SREAD_ERR** – Read of network socket operation from SLE failed
    (Remedy- Lost connection, try to reconnect, if problem persists, contact dBm service.)
  - **SLE_SYSTEM_ERR** – General programming error, in this case, system network select function failed.
    (Remedy- Error should not occur. Contact dBm service.)
  - **SLE_OP_ERR** – DLL sent command properly however, SLE failed to perform operation. SLE could not return status.
    (Remedy- SLE returned a NAK. Contact dBm service.)
  - **SLE_TIMEOUT_ERR** – DLL timed out waiting for acknowledgement from SLE
    (Remedy- Assure SLE is in Remote mode, not Local. Assure network connection. Lost connection, try to reconnect, if problem persists, contact dBm service.)

C Programming Examples to Access DLL:

Below are typical operating scenarios.
(Note: All DLL commands return an error code that should be checked, however, for the purposes of this sample code most error checking has been omitted to keep the code illustration clear.)

Open Interface, Establish Connection and Read Current Status:

```
#include “SleDLL.h”

SLE_SETUP   SleSetup;
SLE_STATUS SleStatus;
SLE_DOWNLOAD_STATUS DnldStatus;
SLE_POSITION SlePositionStatus;
SLE_UPLOAD   SleUpload;
SLE_PREAMBLE   SlePreamble;
SLE_UPLOAD_STATUS UpldStatus;
SHORT err;

// Open interface to DLL
strcpy( SleSetup.SleIpAddr, “198.12.1.120” );          // IP address that SLE is set to
SleSetup.SleTcpPort = SERVER_TCP_PORT;
SleSetup.SleTimeout = SERVER_RESP_TIMEOUT;// secs
err = SLEopen( &SleSetup );
```

```
		if ( err != SLE_NO_ERR)
		{
			/* handle error */


		}
		else
		{
			err = SLEconnect();
			if ( err != SLE_NO_ERR)
			{
				/* handle error */
		}
		else
		{
		err = SLEstatus( &SleStatus );
		if ( err != SLE_NO_ERR)
		{
			/* handle error */
		}
		}
}
```

Setup Static Parameters for Channel 1 if Channel 1 exists and Run Static Mode:

```
// put SLE in default Static Mode
SLEtoggle( (UCHAR)KEY_MODE, (UCHAR)MODE_STATIC );

// Verify SLE has a Channel 1
if ( SleStatus.ActiveChans & 1)
{
		err = 0;
		err |= SLEparam( PARAM_FREQOFFSET, CHAN1, 1000 );  // 1000
hz
		err |= SLEparam( PARAM_ATTEN, CHAN1, 5 );
		// 1 db
		err |= SLEparam( PARAM_PHASEOFFSET, CHAN1, -90 );  // -90
degrees
		err |= SLEparam( PARAM_DELAY, CHAN1, 600000000 );   // 600
msecs

		// Static Mode Starts Executing. Wait for Delay command to have
completed.
// (Completion status may be monitored by a timer task however this sample
code simply waits in a loop)

		err |= SLEdllstatus( &SleDllStatus );
		while ( ! err )
		{
			if ( SleDllStatus.StaticModeState == 2 ||
SleDllStatus.StaticModeError )
		break;
```

```
                    Sleep( 1000 );              // delay 1 sec
                    err |= SLEdllstatus( &SleDllStatus );
            }
}
/* Done with commands, handle errors if any occurred */
```

Download a File and Run In Dynamic Mode:

```
// put SLE in Dynamic Mode
SLEtoggle( (UCHAR)KEY_MODE, (UCHAR)MODE_DYNAMIC );

// Setup Update Interval to 50 msecs
SLEparam( PARAM_UPDATEINTERVAL, CHAN1,  50 );  // chan field is
don't care

// Convert a SATGEN file to SLE format
err = SLEcnvtSATGENfile(TYPE_DLY,
"c:\filedir\dlyfile1.dat","c:\filedir\dlyfile1.sle");
if ( !err )
{
        // Download file and monitor when complete
        err = SLEdownload( "c:\filedir\dlyfile1.sle", DNLD_START );


        while( ! err )
        {
        err = SLEdnldstatus( &DnldStatus );
        if ( ! err )
        {
                err = DnldStatus.LastError;
        if ( err || DnldStatus.DownloadState == 2 )
                        break;
                else
        printf("Progress = %d\n", DnldStatus.PercentComplete );
                }
        }

        // If file downloaded without error, select file to execute
        // Note: The SLE maintains a FileIndex to be able to reference a file.
        // Every time a file is added or deleted to/from flash memory the SLE
        must be
        // queried to obtain indexes of all files. Indexes are not sequential, they
        // are determined according to how flash is most efficiently used.
        if ( ! err )
        {
                index = SubGetFileIndex( "dlyfile1" );

                // Select (enable) file to execute on channel 1 only
        err = SLEsetchans( index, 0x0001 );          // chan bitfield set to 1
        }
}

// If no error, start Dynamic Mode
if ( ! err )
```

```
{
        UCHAR   slekeybuf[2] = { KEY_RESET,  KEY_START };



// put SLE in fwd/rev mode
err |= SLEtoggle( (UCHAR)KEY_LOOP, (UCHAR)LOOP_FWD_REVERSE);

// Issue Reset and Start key commands to SLE
SLEposition(1, &SlePositionStatus); // disable timer and Keep Alive
        err |= SLEkeypad( 2, slekeybuf  );
while( 1 )
        {
        SLEposition(1, &SlePositionStatus);
        // Determine if stopped running
        if ( SlePositionStatus.Mode == M_PAUSE     // received a PAUSE cmd
         || SlePositionStatus.Mode == M_DONE       // job completed normally
         || SlePositionStatus.Mode == M_READY ) // received a RESET cmd
{
        break;    // done running
}
else
{
        // display elapsed time from SLE
        printf( 'time=%d\n",  SlePositionStatus.ElapsedTime);

        // use point position to display location or graph cursor
 }
}
        // obtain last position
        SLEposition(0, &SlePositionStatus); // re-enable timer and Keep Alive
}
```

Subroutines:

```
USHORT SubGetFileIndex( CHAR *file )
{
        SHORT offset;
        CHAR buf[16];

        for ( offset = 0; offset < SleStatus.DnldFileCnt; offset++ )
        {
                err = SLElistfile( offset, buf );
                if ( strcmp( file, buf ) == 0 )
                        return( offset );
        }
        return 0xFFFF;
}
```

Example of File Upload from the SLE:

```
FileIndex = 0;      // first file

// delete old client buffer if allocated
if ( SleUpload.PointBuffer )
        delete  SleUpload.PointBuffer

// init for new file
memset( &SleUpload, 0, sizeof(SLE_UPLOAD) );

// Get Number of Points from Preamble to know size of buffer to allocate
err = SLEupldpreamble( FileIndex, &SlePreamble );
if ( err != SLE_NO_ERR || SlePreamble.NumSamples <= 2 )
{
        Handle Error
}

// Allocate client buffer for sample points
SleUpload.PointBuffer = new double[SlePreamble.NumSamples];

// Read points from uploaded SLE file
err = SLEupload( FileIndex, &SleUpload );
if ( err != SLE_NO_ERR )
{
        Handle Error
}

// Wait for Upload to Complete
while( 1 )
{
        Sleep( 250 );
        err = SLEupldstatus( &SleUpldStatus );
        if ( err == SLE_NO_ERR )
        {
                if ( SleUpldStatus.UploadState == 2 )
                {
                        break;    // done
                }
                else if ( SleUpldStatus.LastError )
                {
                        Handle Error
        break;
        }
        }
        else
        {
                Handle Error
        break;
        }
}
```