SPIRENT Communications | Inspired Innovation

User Guide

DLS-5800 xDSL/DSM Custom Noise Generator/Sequencer

DLS-5410DC Differential/Common Mode Noise Injection Unit

DLS-5409 Mini Passive Noise Injector

July 2011
P/N 71-006581

Spirent Communications, Inc.

26750 Agoura Road

Calabasas, CA

91302 USA

Limited Warranty

Spirent Communications, Inc. (“Spirent”) warrants that its Products will conform to the description on the face of order, that it will convey good title thereto, and that the Product will be delivered free from any lawful security interest or other lien or encumbrance.

Spirent further warrants to Customer that hardware which it supplies and the tangible media on which it supplies software will be free from significant defects in materials and workmanship for a period of twelve (12) months, except as otherwise noted, from the date of delivery (the “Hardware Warranty Period”), under normal use and conditions.

To the extent the Product is or contains software (“Software”), Spirent also warrants that, if properly used by Customer in accordance with the Software License Agreement, the Software which it supplies will operate in material conformity with the specifications supplied by Spirent for such Software for a period of ninety (90) days from the date of delivery (the “Software Warranty Period”). The “Product Warranty Period” shall mean the Hardware Warranty Period or the Software Warranty Period, as applicable. Spirent does not warrant that the functions contained in the Software will meet a specific requirement or that the operation will be uninterrupted or error free. Spirent shall have no warranty obligations whatsoever with respect to any Software which has been modified in any manner by Customer or any third party.

Defective Products and Software under warranty shall be, at Spirent's discretion, repaired or replaced or a credit issued to Customer's account for an amount equal to the price paid for such Product provided that: (a) such Product is returned to Spirent after first obtaining a return authorization number and shipping instructions, freight prepaid, to Spirent's location in the United States; (b) Customer provides a written explanation of the defect or Software failure claimed by Customer; and (c) the claimed defect actually exists and was not caused by neglect, accident, misuse, improper installation, improper repair, fire, flood, lightning, power surges, earthquake, or alteration. Spirent will ship repaired Products to Customer, freight prepaid, based on reasonable best efforts after the receipt of defective Products. Except as otherwise stated, any claim on account of defective materials or for any other cause whatsoever will conclusively be deemed waived by Customer unless written notice thereof is given to Spirent within the Warranty Period. Spirent reserves the right to change the warranty and service policy set forth above at any time, after reasonable notice and without liability to Customer.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, ALL IMPLIED WARRANTIES, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE, ARE HEREBY EXCLUDED, AND THE LIABILITY OF SPIRENT, IF ANY, FOR DAMAGE RELATING TO ANY ALLEGEDLY DEFECTIVE PRODUCT SHALL BE LIMITED TO THE ACTUAL PRICE PAID BY THE CUSTOMER FOR SUCH PRODUCT. THE PROVISIONS SET FORTH ABOVE STATE SPIRENT'S ENTIRE RESPONSIBILITY AND CUSTOMER'S SOLE AND EXCLUSIVE REMEDY WITH RESPECT TO ANY BREACH OF ANY WARRANTY.

Contents

Chapter 1 Introduction

1.1 Introduction

This Operating Manual describes the setup, operation, and control features of DLS-5800, DLS-5410DC, and DLS-5409 products. These products are designed for use in physical testing of ANSI, ETSI, ITU-T, Broadband Forum, HomePNA and EFM xDSL network access equipment. General information is also provided on system administration functions, testing procedures, and diagnostics.

NOTE: The DLS 5410DC noise injection unit is designed to work only with the DLS-5800 noise generator.

1.2 Spirents Involvement in Noise Generation

Spirent Access Emulation Division (AE) has been in the wireline simulation business for over 20 years. Since the days of the S2, Spirent Communications has designed many new simulators both to customers' specifications and to conform to an ever-growing range of industry standards. By introducing the DLS 100 in 1985, we believe that we sold the world's first truly wideband wireline simulator with the capability to successfully simulate attenuation, characteristic impedance, and delay.

In association with wireline simulation, Spirent Access Emulation has also developed products that can effectively simulate the impairments found on real cable. Crosstalk, white noise, RF Ingress (RFI), and impulse noise are but a few of many impairments generated to meet requirements of ANSI, ETSI, ITU-T, Broadband Forum and Japanese standards.

The need for simulated impairments in xDSL network access equipment testing has grown in terms of both bandwidth requirements, noise shapes definitions and variation in time. It became necessary to quickly generate not only standards-based noise shapes for conformance testing, but arbitrary defined shapes for performance testing of new customer xDSL products. The DLS-5800 xDSL Custom Noise Generator, together with DLS-5410DC or DLS-5409 Noise Injectors, assists customers in designing and testing products that exceed these standards.

1.3 Hardware Handling/Cleaning Practices

The DLS-5800 and DLS-5410DC contain electronic components that are sensitive to Electrostatic Discharge (ESD) damage. To prevent premature component failure or latent product damage, it is crucial that you handle this equipment following industry standard ESD handling practices. Refer to *Appendix D, "ESD Requirements,"* for further information.

To clean the DLS-5800 or 5410DC, unplug the respective AC power cord from the facility power and from the inlet at the back of the unit or power supply. Use a lint-free damp cloth for cleaning the exterior of the units only.

1.4 Protecting Your Investment

Spirent Communications is committed to providing the highest quality products and customer support possible. An annual calibration is required to ensure that your units are operating properly.

Spirent Communications offers two cost-effective optional service programs, an extended

warranty and a three-year calibration agreement. Each of these programs is designed to improve the ease and efficiency of servicing Spirent Communications test equipment.

1.4.1 Extended Warranty

Spirent Communications' Extended Warranty gives two years in addition to the original one-year manufacturer's warranty. Under the warranty agreement, Spirent Communications repairs any covered product that needs service during the warranty period. At the time of repair, any required firm ware and/or software upgrades are installed free of charge and if required as part of the repair, the unit receives a complete calibration. Spirent Communications also provides return shipment of any unit covered under warranty at Spirent Communications' cost.

The Extended Warranty provides:

- Extension of the original one-year limited warranty by two years (thus, a total warranty coverage of three years)
- Required firmware and software upgrades installed free at time of repair
- Free calibration due to repair during the coverage period
- Prepaid, return shipment of repaired products worldwide.

Spirent Communications' Extended Warranty can be purchased at any time up until the expiration of the original one-year manufacturer's warranty.

1.4.2 Three-Year Calibration Agreement

Spirent Communications' three-year calibration agreement gives the opportunity to invest in a yearly calibration for three years at a significant cost saving, ensuring optimum product performance.

Timely calibrations are critical to ensuring the best accuracy from your Spirent equipment. A report containing all calibration data is shipped with the product.

The Spirent Communications' three-year calibration agreement provides:

- Notification from Spirent Communications when calibration is due
- Calibration data report
- Prepaid return shipment of calibrated unit worldwide.

Please contact Spirent Communications Customer Service for more information on these programs.

1.4.3 Where to Go from Here

For more information about:

- Controlling the DLS-5800/5410DC/5409 system, see Chapter 3, "DLS-5800 Software."
- Creating custom noise files, see Chapter 5, "Crosstalk Noise Profiles," and Chapter 6, "Ingress Noise Profiles (Optional)."

You should read *Getting Started* thoroughly before powering up the system. The

remainder of this manual contains information about control, specifications, performance and warranty.

We recommend you use our software to configure and control the noise generator. However, we describe common and device specific message sets that can be sent to the noise generator and injector in *Remote Control*.

Some features covered in this manual may not be available with the specific hardware and software configuration on your noise generator. Contact a Spirent Communications sales representative to discuss upgrades.

If you have any questions after reading this manual, please contact a member of the Customer Service team. See *Support Services Contact Information*.

1.5 Support Services Contact Information

We encourage our customers to take advantage of the self service tools available on our Spirent Customer Service Centre website, http://support.spirent.com:

- Submit or check the status of a service request
- Check the status of your RMA
- Download recently published user manuals
- Download application notes
- Download software updates
- Download firmware updates
- Request demo software
- License fulfillment
- Many other useful tools

Alternatively, see contact information below.

Americas: 1-800-SPIRENT (774-7368) (for US and Canada only)
From int'l: +1 818-676-2616
Fax: 1-818-880-9154
Email: support@spirent.com

China: +86 10 8233 0033, +86 800 810 9529
Email: supportchina@spirent.com

EMEA: +33 1 6137 2270
Email: support@spirent.com

Spirent Communications, Inc.
26750 Agoura Road
Calabasas, CA 91302

1.6 System Overview

The DLS-5800 system allows you to generate standards-based (DLS 5Bxx Noise Libraries) or customer-generated noises with a bandwidth of up to 30 MHz. The DLS-5410DC Noise Injector lets you apply these noises to a line in either differential or common mode and provides additional high-precision timed-impulse functionality. The DLS-5409 Noise Injector applies the noise generated by DLS-5800 in differential mode.

Figure 1-3: DLS-5800

Figure 1-2: DLS-5410DC

Figure 1-1: DLS-5409

The DLS-5800 has four independent output ports per card, all on the back panel. A maximum of 6 noise cards may be installed on the DLS-5800. Several upgrade options are available. Please contact Spirent Communications on pricing for these upgrades.

1.6.1 Hardware Configuration

The DLS-5800 noise generator consists of a rack-mountable 4U-high PC chassis

containing 2, 4 or 6 custom Arbitrary Waveform Generator (AWG) module(s), keyboard, and mouse, control software, and sample noise library (s/w).

The DLS-5800 is equipped with a 90 to 264 VAC, 50 to 60 Hz power supply (auto-ranging) or 100-120/220-240 VAC (±10%), 60/50 Hz power supply (switchable).

The DLS-5410DC is a 1U rack mountable chassis, which is powered by an external power supply (included).

The DLS-5409 is a matchbox-sized unit which needs no power supply.

The DLS-5800 configuration is as follows:

- DVD drive
- Hard drive (minimum 250 GB)
- 2 Gig RAM (minimum)
- 3 RS232 ports and 1 parallel printer port
- 6 USB ports
- LAN port
- Standard VGA monitor port
- 8 or 16 or 24 channels Arbitrary Waveform Generator (that is 2 or 4 or 6 AWG modules, each with 4 ports), Software controlled. BNC connectors for all Noise Outputs, designed to drive a 50 Ω load
- Included software:

WinXP™ Professional Operating System
DLS 1100 Software
DLS-5800 Control Software
DLS 5B07 Sample VDSL Noise Files

NOTE: DLS-5800 can be configured to handle 2 or 4 or 6 AWGN noise cards within a single chassis.

The DLS-5800 does not perform noise injection directly to the wireline simulator. It must be injected into the test circuit by means of a connection to any one of the following Spirent products:

- DLS 5410DC Differential/Common mode noise injector (combiner functionality)
- DLS 5409 Differential mode noise injector (mainly used to inject noise into multi-line simulators)

NOTE: DLS-5410DC Noise injectors have to be connected to DLS-5800 as follows:

1x DLS-5410DC Noise injector for DLS-5800 with 2 noise cards (8 channels)

2x DLS-5410DC Noise injectors for DLS-5800 with 4 noise cards (16 channels)

3x DLS-5410DC Noise injectors for DLS-5800 with 6 noise cards (24 channels)

NOTE: Each DLS-5409 noise injector will be connected to a noise output of DLS-5800.

NOTE: Combinations of DLS-5410DC and DLS-5409 are not allowed. For a DLS-5800

system, all injectors must be of the same type.

1.7 DLS-5800 / DLS-5410 / DLS-5409 Configurations

DLS-5800 is available in 3 configurations having 2, 4, and 6 noise cards. These configurations require 1, 2, and 3 DLS-5410 Noise Injectors or 8, 16, and 24 DLS-5409 Noise Injectors respectively.

1.7.1 Test Setup

The building blocks of this test set-up are:

- A test loop, being either a real cable or a cable simulator
- A noise generator DLS-5800 that generates a mixture of random noises
- An "adding" element (noise injector) to inject the impairment noise into the test loop (DLS-5410DC or DLS-5409)
- The DLS-5800 creates noise profiles that must meet many noise characteristics to enable a realistic imitation of (spectrally polluted) operational access networks.

Noise is:

- Frequency dependent
- Dependent on the length of the test loop, since FEXT coupling functions between wire pairs are length dependent
- Usually different for downstream and upstream performance tests (depending on the application)

A noise profile is a Power Spectral Density (PSD) description of the crosstalk noise as it is observed at the receiver of the xDSL modem under test (near the point of injection), so you use a different noise profile for each measurement.

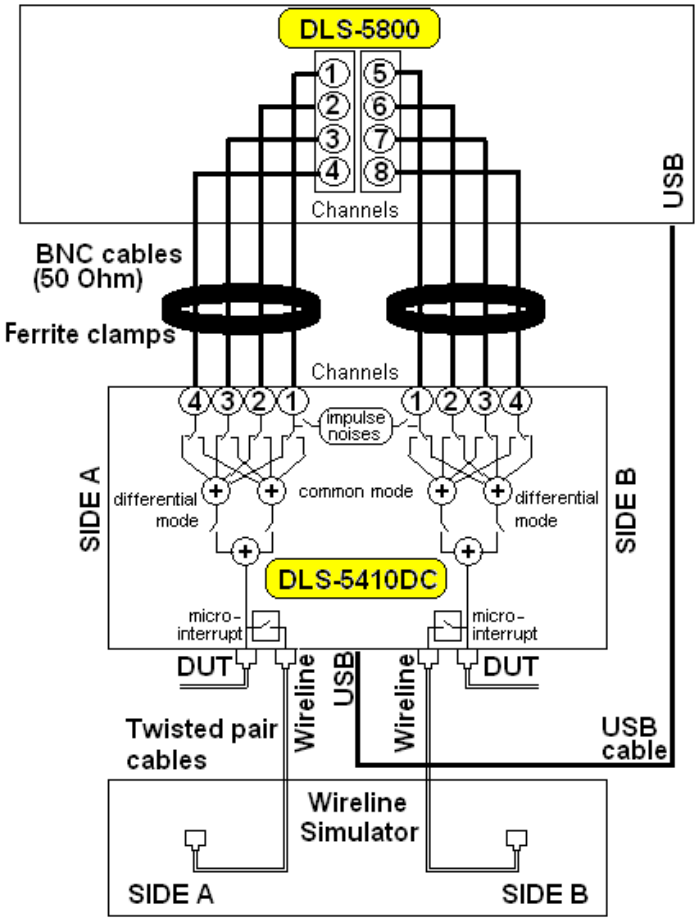


Figure 1-4: Example of Test System Setup

Figure 1-4 shows an example of a typical test setup using the DLS-5800 (2 AWG card configuration) and a DLS-5410DC. In this example, the noises generated by channels 1 to 4 of DLS-5800 are injected by DLS-5410DC at SIDE A. Similarly, the noises generated by channels 5 to 8 of DLS-5800 are injected by DLS-5410DC at SIDE B.

The DLS-5410DC allows each channel to be injected either in differential mode or in common mode. All noises that will be injected in differential mode are summed together by enabling the respective channels; all noises that will be injected in common mode are summed together by enabling the respective channels. For each SIDE A or SIDE B, differential and common mode injection can be applied simultaneously. Note: each channel can be enabled to inject either in differential mode or in common mode. A channel cannot simultaneously be injected in both differential and common mode.

NOTE that the DLS-5800 when fully loaded with 6 AWG noise cards must use 3 DLS-5410 Noise Injection (or 24 DLS-5409) units. Using 3 DLS-5410 units, the configuration allows a user to inject noise on up to 3 wireline simulator/noise generator test beds from 1 single DLS-5800 noise generator. For more information on the compatible simulators, see *"User Documentation" on page 8*.

1.8 DLS-5800 Features

The DLS-5800 combines PC architecture, custom Arbitrary Waveform Generator (AWG) module(s) and Spirent control software to generate noise. This combination of hardware and software results in an impairment generator that generates precision noise for testing of xDSL network access equipment. With the DLS-5800, you can generate noise that is a realistic replica of the spectral pollution induced in a wire pair of a real network access cable.

The DLS-5800 software implements the following noise sources:

- A crosstalk noise source to generate a replica of the noise that originates from xDSL transmission equipment (disturbers) that make use of other wire pairs in the same cable. The spectrum of crosstalk noise is predominantly continuous in nature.
- An optional RFI ingress noise source to generate a replica of the noise that originates from (broadcast) radio stations outside the cable. The spectrum of ingress noise is predominantly discrete in nature.

In addition, the DLS-5800 system supports:

- Several applications/tools are available to create custom impulse noise (DLS-5C80)
- Application to create custom xtalk impairment files (DLS-5C60/DLS-5D10)

The DLS-5410DC Noise injector adds:

- Selectable per channel differential or common mode injection
- DSM (Dynamic Spectrum Management) simulation capability to sequence or dynamically change impairment files instantaneously.
- Customizable REIN/PEIN/SHINE
- Micro-interrupts

The DLS-5800 and 5410DC introduce a feature to help validate modems that support DSM functionality. The DLS-5800 control software allows users to specify specific noise conditions which are stored as “frames”. Users can then create sequences of frames, and specify the playback duration for each frame.

Noise frames can be created using any DLS-5800 noise type, including crosstalk, time domain, RFI, and impulses.

Chapter 2 Getting Started

2.1 Receiving and Unpacking the Unit

The noise generator and noise injection units have been shipped in reinforced shipping containers. Please keep these containers in case you need to ship these units to another location or for repair.

The DLS-5800 is a 4U-high rack-mountable unit. The DLS 5410DC is a 1U-high rack-mountable instrument. The DLS-5409 is a matchbox-sized unit.

Items supplied with DLS-5800

- Front panel door key
- USB mouse and keyboard
- Pre-installed, licensed DLS-5800 Software
- Operating System information
- Optional licensed software (Noise Files). Contact Spirent for available noise files and optional upgrades
- One (1) AC power cord

Items supplied with DLS 5410DC

- DLS 5P02 External Power supply with AC Power Cord and spare fuses
- 8 cables BNC-BNC male
- One pair of ferrite clamps
- 1 USB A to B cable, 6 ft
- RJ45 CAT5 cables (2 cables of 1 ft, 2 cables of 2 ft)

Note: Confirm that you have received all the items on the list and report any discrepancies to Spirent Communications. Also see *Returning the DLS-5800/DLS-5410DC* in the next section.

2.2 Returning the DLS-5800/DLS-5410DC/DLS-5409

Note: An RMA number is mandatory and must be obtained from a Spirent Communications Customer Service center prior to shipping the unit (see "How to Contact Us" for details on how to contact the nearest Customer Service center).

To return the DLS-5800/5410DC/5409:

- Prepare the unit for shipment: turn the power off, disconnect all cables (including the power cable) and pack the unit in its original cartons. Do not place any cables or accessories directly against the front panel as this may scratch the surface of the unit.
- Mark all shipments with labels indicating that the contents are fragile.
- Ensure that the Return Material Authorization (RMA) number is shown on the outside of the package(s) if you are sending a unit back to the factory.

2.3 Front Panel DLS-5800 and Connections

The DLS-5800 has a locking front panel door, behind which are the Power and Reset switches, USB ports, LED indicators, the DVD/RW drive, etc. *Figure 2-1* shows the main components under the front panel.

The DLS-5800 front panel main components are as follows:

- Rack mount carrying handles
- Two USB ports. These can be used for a keyboard/mouse or for memory devices.
- Power switch (momentary switch)
- Reset button (momentary switch)
- LED indicator for Power
- LED indicator for HDD
- DVD/RW drive
- Air Filter for the cooling fan

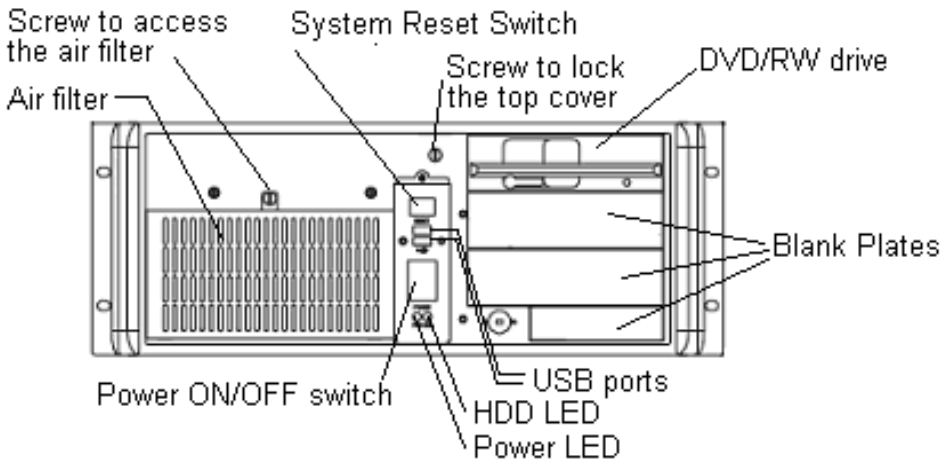


Figure 2-1: DLS-5800 Front Panel

2.4 Accessing the Power Switch, Reset Button, and Drives

The power switch, reset button, and drives are located behind the locking front panel door. Ensure you have completed all back panel connections before opening the locking front panel door and turning on the power. The key is supplied with the unit.

2.5 Back Panel DLS-5800 Components and Connections

The back panel (see *Figure 2-2*) is used to connect a mouse, keyboard, and monitor as well as to connect to the noise injectors and test circuit for both communication (optional remote control of test system components) and noise generation purposes.

The DLS-5800 back panel components (*Figure 2-2*) are as follows:

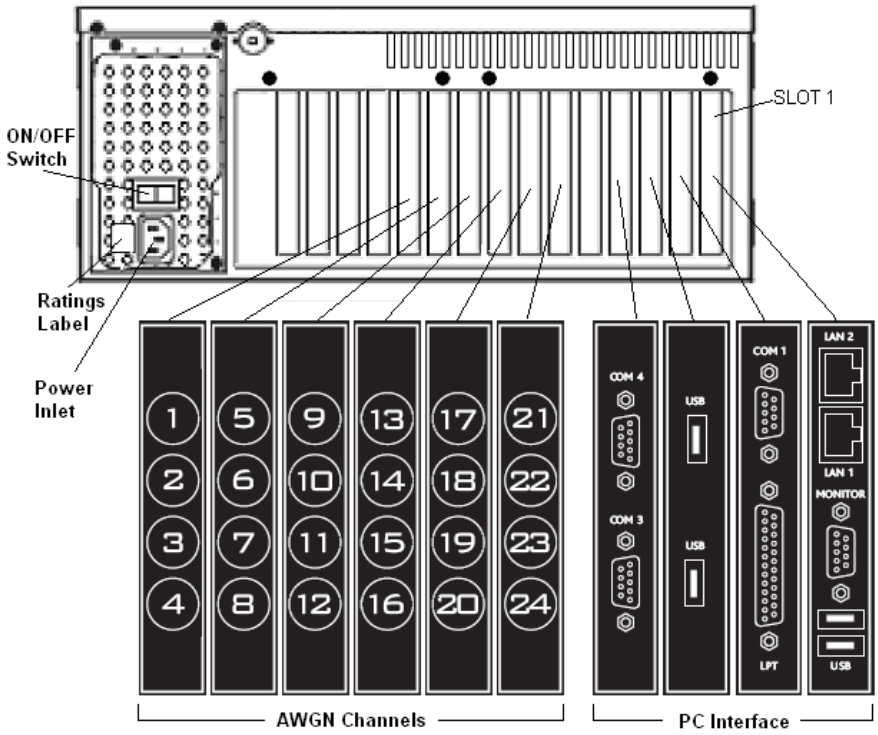


Figure 2-2: Back Panel Connections

- AC power input, ON/OFF switch
- AWG noise cards x2 (Base configuration) maximum 6 cards possible
- LPT (Printer port)
- Four USB ports to connect to DLS-5410DC units, license dongles, etc.
- LAN connectors. Ethernet ports, which allow remote control of the DLS-5800 by another computer. LAN2 port is not activated
- COM 1, COM 2, COM3 connectors. To remotely control wireline simulators
- Monitor port

2.6 Front Panel DLS 5410DC and Connections

Figure 2-3: DLS-5410DC Front Panel

The DLS 5410DC front panel components are as follows:

- Rack mount brackets
- USB port to be used for control by DLS-5800.
- Power LED (indicates that the unit is powered up)
- Remote LED (indicates that the unit is remotely controlled)
- DUT ports (RJ45) at Side A and Side B to connect to the devices under test at each end of the test loop
- WIRELINE ports (RJ45) at Side A and Side B to connect to the two sides of the wireline simulator
- Ground (GND) connectors

2.7 Rear Panel DLS-5410DC and Connections

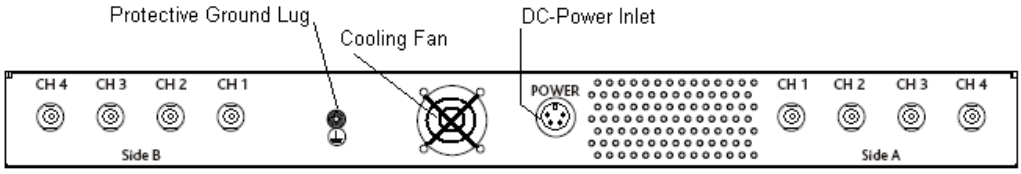


Figure 2-4: DLS-5410DC Rear Panel

The DLS-5410DC rear panel components are as follows:

- DC-Power Inlet, to connect to the power-supply DLS-5P02, which is provided as an accessory to DLS-5410DC.
- Ground Lug, to be connected to the protective Earth
- BNC female (CH1 to CH4) for the 4 input channels of the injector for Side A
- BNC female (CH1 to CH4) for the 4 input channels of the injector for Side B

2.8 DLS-5409 Connections

DLS-5409 has one BNC female jack for its input and one RJ-45 female jack for its output. The RJ45 output is designed to be fed into an external splitter to which the DUT and wireline are connected.

2.9 Connecting the Mouse, Keyboard and Monitor

To connect the mouse, keyboard, and monitor:

Attach the mouse and keyboard to the USB ports on the front or rear panel of the DLS-5800. Connect computer monitor to rear panel video input

Note: If using the front panel USB ports users will not be able to lock the chassis.

Four USB ports are provided on the rear panel of DLS-5800 and may be used to connect DLS-5410 units or other devices.

Ensure the mouse and keyboard are plugged in before turning on the power.

2.10 System Hardware Setups

There are specific connections depending on the number of Noise Cards in your DLS-5800 unit. The number of Noise Cards will also determine the number of DLS-5410 or DLS-5409 Noise Injectors that you need. Connect your DLS-5800 with your DLS-5410DC or DLS-5409 injectors and Wireline Simulators as shown in these diagrams:

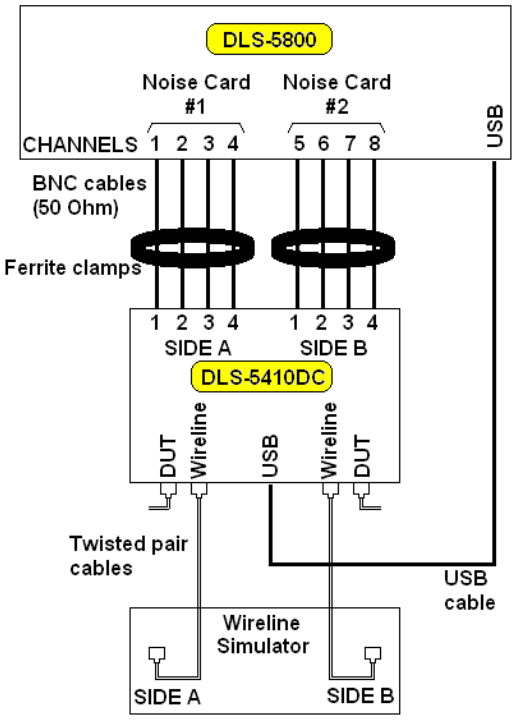


Figure 2-5: 2-card DLS-5800 with 1 DLS-5410DC

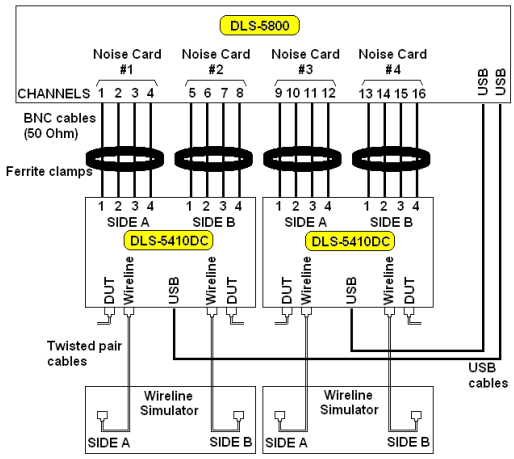


Figure 2-6: 4-card DLS-5800 with 2 DLS-5410DCs

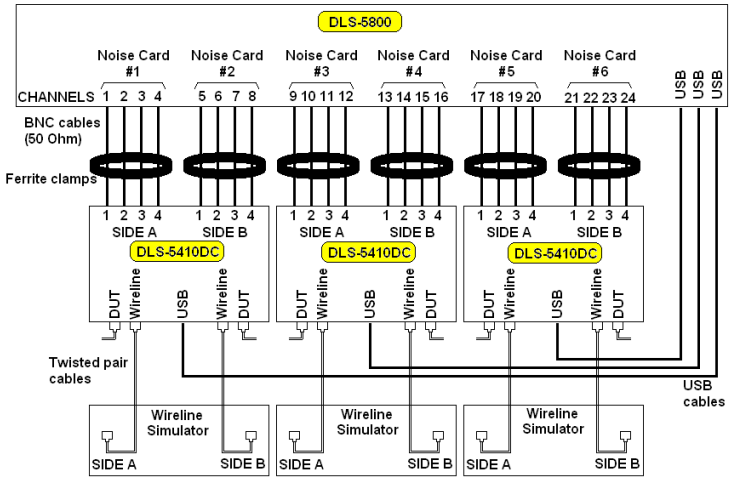


Figure 2-7: 6-card DLS-5800 with 3 DLS-5410DCs

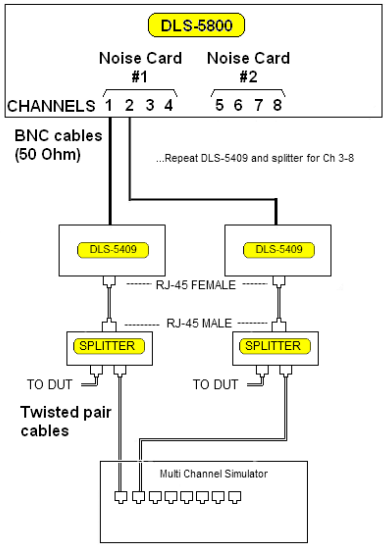


Figure 2-8: Production Testing Setup with 2-card DLS-5800 and 8 DLS-5409s (only 2 shown)

2.11 C onnecting the Ferrite Clamps

To ensure EMC compliance, attach the enclosed ferrite clamp to the Ethernet port cable, close to the unit.

Also, for each DLS-5410DC injector, two ferrite clamps 28A3851-0A2 will be mounted on the BNC cables coming from DLS-5800:

- One ferrite clamp is installed on the group of BNC cables corresponding to channels 1 to 4 for Injector Side A
- One ferrite clamp is installed on the group of BNC cables corresponding to channels 1 to 4 for Injector Side B

The two ferrite clamps are provided in the accessory package of the DLS-5410DC units.

Ferrite clamps are not required for the BNC connection to DLS-5409.

2.12 Connecting to Power

To connect to power:

- Make sure that the ON/OFF switch on the back of DLS-5800 is set to OFF.
- Connect the power input (on the DLS-5800 back panel) to an AC line, with 90-264 VAC, 50 to 60 Hz (auto-ranging). Use the power cord provided in the accessories package.
- If using DLS-5410DC:
    1. Verify the power supply selector located on the front panel of the DLS-5P02 Power Supply, which is provided in the accessories package of DLS-5410DC. Set it to the appropriate voltage setting.
    2. Make sure that the ON/OFF switch on the front panel of the DLS-5P02 Power Supply is set to OFF.
    3. Insert the round DIN connector of the DLS-5P02 Power Supply into the power inlet on the rear panel of DLS-5410DC.
    4. Connect the power input (on the DLS-5P02 front panel) to the AC line. Use the power cord provided in the accessories package.

The units will be powered up only:

- After all interconnections between units are done.
- After all additional devices (such as keyboard, mouse, monitor, etc) are connected
- After making sure that all safety requirements are met. Please see Appendix C, “Safety Information and Instructions” for more details

To power up DLS-5800: turn on the ON/OFF switch on the back panel of DLS-5800 then momentarily press the power switch on the front panel of DLS-5800. The Power LED located on the front panel of DLS-5800 will light when the unit is powered on.

To power up DLS-5410DC: turn on the ON/OFF switch on the front panel of DLS-5P02. The Power LED located on the front panel of DLS-5410DC will light when the unit is powered on.

Note: To power DLS-5410, use only the DLS-5P02 Power Supply, which is provided in the accessories package of DLS-5410DC. Do not use any other power supply.

Note: Always ground the external insertion circuit to the same ground circuit/power bar as the DLS-5800.

Note: Ensure that the equipment has enough clearance at the front and at the rear (minimum 4 inches or 100 mm) to allow proper air ventilation.

Please see *Chapter 10* for more details.

2.13 Controlling DLS-5800 / 5410DC / 5409

The DLS-5800 / 5410DC / 5409 is controlled using the factory-installed DLS-5800 Control Software. The DLS-5800 Control Software is pre-loaded on the DLS-5800, and allows you to select, customize, load, and generate noises onto real wire or simulated loops. This software also provides control of DLS-5410 and support for the DLS-5409 and provides remote access to the system hardware.

Thanks to this remote access functionality of the DLS-5800 Control Software, you can also write your own control applications for use with the DLS-5800, DLS-5410DC, and DLS-5409. If you are developing custom control software, *Remote Control* discusses the accepted commands to configure the DLS-5800/5410DC/5409.

Note: Spirent Communications warrants the DLS-5800 and associated software. Spirent Communications however, does not warrant non-Spirent application software that may be run on DLS-5800 units. Applications installed on the DLS-5800 units without prior approval from Spirent Communications void all warranties associated with this product.

2.13.1 Local Area Network Connections

To make LAN connections:

- Connect the Ethernet cable to the LAN1 port when connecting the computer to a local area network. Please note that port LAN2 of DLS-5800 is disabled.
- Set the IP address according to the instructions in the sections beginning on page 28.

This enables the DLS-5800 unit to be controlled remotely via the LAN.

2.13.2 IP Address Setup

The DLS-5800 unit can be used as a stand-alone PC or as part of a network configuration.

2.14 DLS-5800 in a Stand-Alone System

When you operate the DLS-5800 unit as a non-networked stand-alone system (default configuration), use the settings provided in this section to ensure maximum performance:

For maximum performance the PC's TCP/IP properties should be set to obtain a static IP address as shown.

To obtain a static IP address:

- Select Control Panel > Network > Configuration.
- Select TCP/IP from the list.
- Click on Properties.
- Select the IP Address tab.
- Click the Specify an IP address button.

Note: In the following steps, the specified IP Address and Subnet Mask field entries should only be used when the computer is not connected to a network.

- In the IP Address field, enter 192.168.0.1
- In the Subnet Mask field, enter 255.255.255.0

2.15 DLS-5800 as part of a Network

When you connect the DLS-5800 computer to a network, the settings above must be changed, either to obtain a dynamic IP address or to an assigned static IP address. Your Network Administrator can tell you which settings to use.

To specify IP addresses statically:

- Select Control Panel > Network > Configuration.
- Select TCP/IP from the list.
- Click on Properties.
- Select the IP Address tab.
- Click the Specify an IP address button.
- In the IP Address field, enter the value as specified by your Network Administrator.
- In the Subnet Mask field, enter the value as specified by your Network Administrator.

To obtain IP addresses dynamically

- Select Control Panel > Network > Configuration.
- Select TCP/IP from the list
- Click on Properties.
- Select the IP Address tab.
- Click the Obtain an IP address automatically button.

2.16 Controlling other Spirent Devices

Your DLS-5800 may include other installed Spirent applications for controlling various xDSL wireline simulators. Such applications may include DLS-1100, DLS-410B etc… and these applications will use one of several possible communications interfaces depending on the hardware capabilities. Some supported interfaces include RS-232C (serial) and IEEE 488.

Chapter 3 DLS-5800 Control Software

3.1 Introduction

This chapter explains the DLS-5800 Control Software, which is factory-installed on the DLS-5800. For information on controlling the DLS-5800, DLS-5410, and DLS-5409 using your own software, see Chapter 8, "Remote Control Programming.".

The DLS-5800 Control Software is a Windows™ Graphical User Interface (GUI) which provides you with an easy, intuitive method of configuring the various properties of the generated noise and controlling the DLS-5410DC Noise Injector. The software also supports the use of DLS-5409, which is a passive injector which does not require any control.

The DLS-5800 can simulate the combined spectral pollution from hundreds of arbitrary xDSL transmission systems that use the same multi-wire telephony cable. Several add-on applications come pre-packaged with the DLS-5800 to further assist users at developing various noise types – in particular, the DLS 5C60/5D10 custom xtalk noise creation tool and DLS 5C80 custom impulse noise creation tool.

For more information about the noise shape configurations see:

- Chapter 5, "Crosstalk Noise Profiles"
- Chapter 6, "Ingress Noise Profiles"
- Chapter 7, "Time Domain Noise Profiles"

3.2 Overview of the DLS-5800 Software Features

The DLS-5800 Control Software calculates a noise sample from the noise profiles that you select. The noise sample is downloaded to the DLS-5800's built-in AWG card which converts the sample into an analog signal. The DLS-5800 outputs the signal to the inputs of a suitable Noise Injector, which injects the signal into a real or simulated wire line. The DLS-5800 is designed for use with the Spirent DLS-5409 Passive Injector or DLS-5410DC Noise Injector.

The DLS-5410DC Noise Injector provides on-off switching and common- or differential-mode injection, as well as high-precision timed switching in order to create impulses such as REIN, SHINE, PEIN etc…

The DLS-5409 Passive Noise Injector injects the signal directly in differential mode.

- A *Noise Profile* is a description of the characteristics of the noise, such as the power spectral density (PSD) of crosstalk noise or the radio frequency interference (RFI) tones of ingress noise.
- A *Noise Sample* is a sequence of numbers representing the noise profile in the time domain. Noise samples exist in the internal memory of the software, and are not made available as a separate entity.

The software implements the following features:

- Importing a *Crosstalk Noise Profile* from a file that specifies the PSD of crosstalk noise in a human readable ASCII table format.
- Importing an *Ingress Noise Profile* from a file that specifies the RFI tones of ingress noise in a human readable ASCII table format.

- Importing a *Time Domain Profile* from a file that specifies amplitude levels vs. time.
- Choosing the settings of the noise generator. These correspond to fundamental parameters of the Noise Sample, such as the length of the Noise Sample.
- Various amplification settings to control and fine-tune the level of the total output noise.
- Calculating a noise sample from an active noise profile.
- Downloading the synthesized noise sample to the DLS-5800 and activating its output.
- Displaying a graphic representation of the requested spectrum, and the one that will be generated (in case the output signal is amplified to compensate for the attenuation in the noise injection network).
- Display of the key properties (power, crest factor, and so on) of the synthesized noise sample.
- A graphic representation of the characteristics of the generated noise, both in the time-domain and in the frequency-domain. This includes the spectral density and the distribution function of the noise.
- A remote-only feature which allows starting noise on multiple channels with minimal delay between. See section 7.3.6, "Synchronized Start" for details.
- A remote-only feature which allows generation of precision-timing noise bursts. See section 7.3.4, Noise Burst Commands, for details.

NOTE: A 2-card DLS-5800 uses 1 DLS-5410DC Noise Injector or 8 DLS-5409 Passive Noise Injectors. A 4-card DLS-5800 uses 2 DLS-5410DC Noise Injectors or 16 DLS-5409 Passive Noise Injectors. A 6-card DLS-5800 uses 3 DLS-5410DC Noise Injectors or 24 DLS-5409 Passive Noise Injectors.

NOTE that depending on your configuration, some GUI features may look different from what is shown here in the manual.

3.3 Installing the Software

The DLS-5800 Control Software is factory-installed. If it becomes necessary to re-install the software:

5. Insert a software installation CD/DVD in the computer CD/DVD drive.
6. The Installation Wizard starts. If it does not start, launch Setup.exe from the CD or DVD.
7. Follow the on-screen prompts.

You may also need to re-install the DLS-5410DC driver. When you connect the DLS-5410DC Noise Injector (via USB) and switch it on for the 1st time, if the Windows "Found New Hardware" wizard appears, follow the prompts and direct it to look in C:\Program Files\Spirent Communications\DLS 5800\DLS-5410 Driver. You may need to repeat this for all connected DLS-5410DC injectors.

NOTE: For systems that use more than 1 DLS-5410DC injector, the software identifies the DLS-5410DC units by serial number. The DLS-5410DC unit with the lowest serial number is always considered unit 1, the next lowest is unit 2, and the highest serial number is unit 3. This is important because the DLS-5410DC injector unit ordinal (1, 2, or 3) determines which generator channels are associated with the injector, and the software assumes that the physical connections are done appropriately. In other words, injector unit 1 has the lowest serial number, and it must be connected to channels 1 – 8 of the DLS-5800 noise generator. Injector 2 has the 2nd lowest serial number, and must connect to channels 9 – 16. Injector 3 has the highest serial number and must connect to channels 17 – 24.

3.4 Launching and Exiting the DLS-5800 Control Software

Before starting the DLS-5800 Control Software, if you are using DLS-5410DC, ensure that all DLS-5410DC Noise Injector units are powered up and connected to the DLS-5800 via the USB ports.

To launch the DLS-5800 Control Software, select Start | All Programs | Spirent Communications | DLS 5800 | DLS 5800.

The *s*oftware will attempt to connect to the DLS-5410DC noise injector(s) required for your hardware configuration.

If the software cannot establish communications with the correct number of DLS-5410DC units, or if the associated "Display this dialog on startup" box is checked, the software will display the Injector Configuration dialog.

If the software was successful in establishing communications with the correct number of DLS-5410DC units, and the Injector Configuration dialog appears because the "Display this dialog on startup" box is checked, you will see that the Apply button is disabled and the appropriate Locate buttons are enabled. Just click OK to close the dialog.

If you are not using DLS-5410DC, you must use DLS-5409. Select this injector type at the Injector Configuration dialog, click Apply, then click OK to close the dialog.

NOTE that DLS-5800 must use DLS-5410DCs for all channels OR DLS-5409s for all channels. A mix of injector types is not supported.

*To exit the DLS-5800 Control Software, select **File** | **Exit**.*

3.5 Injector Configuration Dialog

There are two Spirent noise injectors that may be used with the DLS-5800 noise generator:

- DLS-5410DC multi-input noise injection unit
- DLS-5409 passive single input/single output injection unit

The DLS 5409 does not require a control connection. It has a fixed insertion loss which is automatically applied by the software.

Each noise injector model has its own set of electrical characteristics, for example injector loss, frequency response, and bandwidth. The DLS-5800 Control Software automatically configures itself accordingly based on the Injector Type that you select.

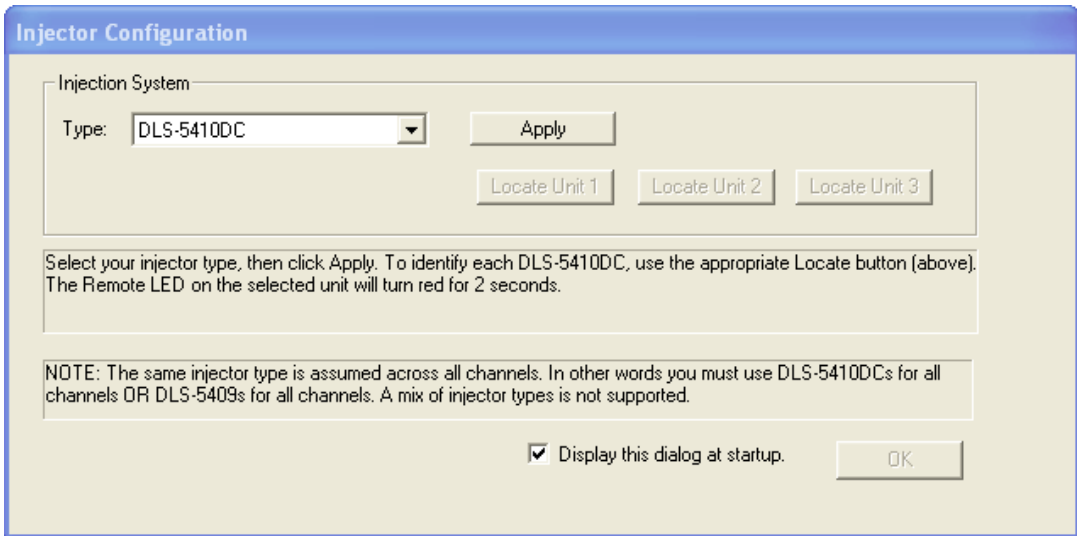


Figure 3-1: Injector Configuration Dialog

The Injector Configuration dialog allows you to select the type of Noise Injector(s) to be used. You can select DLS-5410DC or DLS-5409 from the Type combo box. Then click Apply. If you chose DLS-5410DC, the software will then attempt to connect to the injector units via USB. If successful, the Locate Unit buttons will become enabled, and you can use these buttons to turn the Remote LED of the associated unit RED for a couple of seconds, then turning it GREEN again. This helps identify individual DLS-5410DC units in order to make the correct connections.

DLS-5409 is a passive injector, with no communications or control. Thus, after you click Apply, just click OK to close the Injector Configuration dialog.

3.6 Quick Start - Generating a Noise Sample

This procedure provides you with the basics to quickly select one or more noise profiles and generate a noise sample from them. To configure and generate a noise sample for the DLS-5800:

If it is not already running, launch the DLS-5800 application from Start | All Programs | Spirent Communications | DLS 5800 | DLS 5800.

1. From the DLS 5800 Control Software main window, use the browsing tree at the left to locate the desired noise file(s).
2. Click on file names to load the files into the workspace.
3. Make any required settings (for example, Calibration Impedance).
4. Choose the **Target Channel** from the Noise Calculation area and select the **Sample Number** required.
5. If using DLS-5410, from the DLS-5410 pull down menu, choose the appropriate injector to correspond to the target channel. For example, if you chose Channel 1-8, select Injector 1, and for Channels 9-16, choose Injector 2. You will see the

DLS-5410 Control dialog for the specified injector unit.

6. If using DLS-5410, for the target channel, select the coupling mode (common or differential). You may also wish to turn on the output for that channel.
7. If using DLS-5410, click OK to close the DLS-5410 Control dialog, or just click anywhere on the main GUI window.
8. Click the **Calculate** button.
9. Click **Generate** to load the calculated sample into the Noise Generator and start signal output on the previously selected target channel.
10. If using DLS-5410, if you didn't previously turn on the channel output , then return to the appropriate DLS-5410 Control dialog, and do so now.

At this point the noise should be present at the output of the DLS-5409 or DLS-5410 Noise Injector.

3.7 Control Software Window Layout

Spirent DLS-5500 users will notice that, in order to minimize the learning curve, the DLS-5800 Control Software window remains similar to the DLS 5500 software. The main GUI window is divided into five regions, as shown in *Figure 3-2*:

1 Work Space

2 Combined Noise

3 Noise Calculation

4 Noise Output

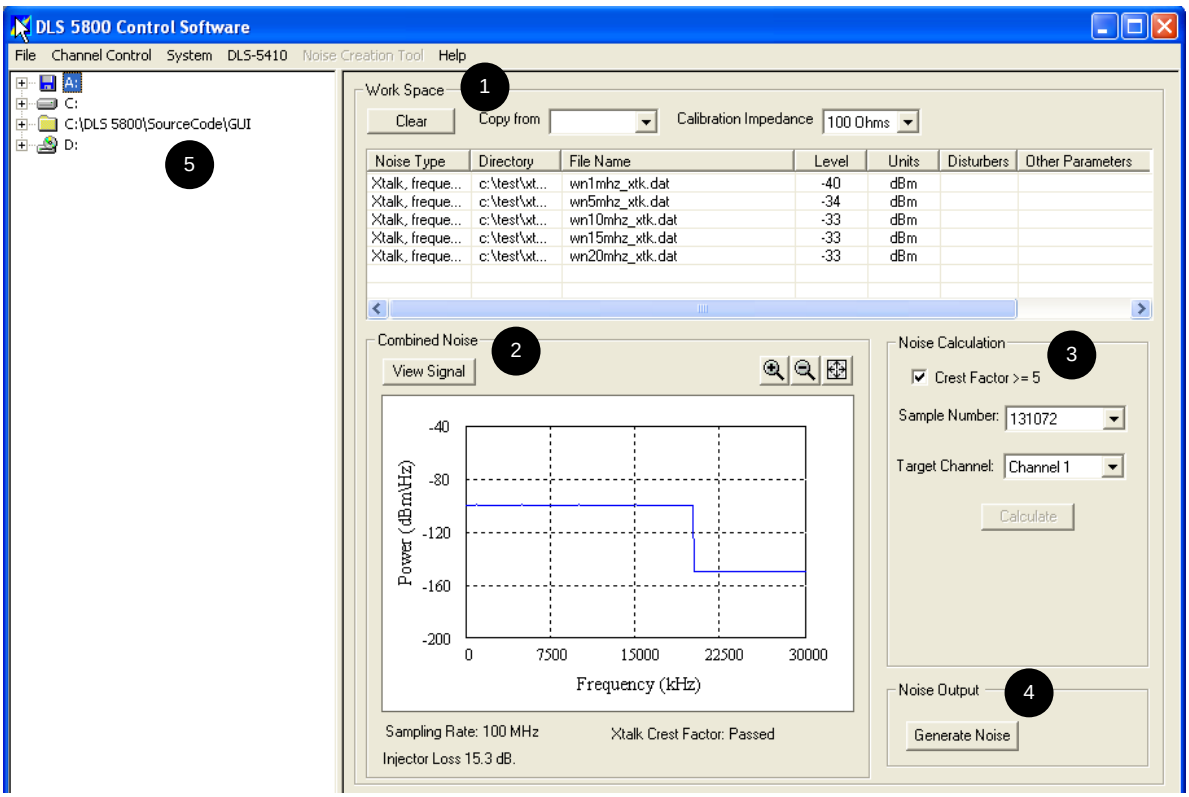


Figure 3-2: Main GUI Window

5 File browsing tree

3.7.1 Work Space Region

The ***Work Space*** region allows you to combine various noise profiles available. The table shown within the Work Space is referred to as the ***combiner window***. This term is used extensively throughout this chapter.

Up to 7 Crosstalk (XTK) and Radio Frequency Interference (RFI) files can be combined.

Only 1 Time Domain (TD) file, Impulse (IMP), or Custom Saved File (CST) can be loaded at any time.

NOTE: user may combine TD and Impulse noise types together if the sample rate of each file is identical.

The combined profiles can be generated onto a single channel output. The combiner window displays the noise file properties described in *Table 3-1*.

Table 3-1: Combiner Window Table Description

Column	Description
Noise Type	Identifies the file type as a frequency OR time domain file type
Directory	File location
File Name	Shows the noise file name and extension
Level	Shows the reference power level of the selected noise. This level is also known as the 0 dB reference level calculated by Spirent across a specific calibration impedance.
Units	Shows the units of measurement for the level field
Disturbers	Shows the number of disturbers for specific files that use this method of specifying level. ANSI specifies some of its crosstalk files related to the quantity of disturbers within a binder group. Note: If the file does not use this method, the field is grayed out.
Other Parameters	Displays specific file properties for information purposes only. ETSI complex load impedance file types will be displayed in this field as CLIC (Complex Load Impedance Compensation)

Notes:

- The DLS-5800 cannot combine time domain noise (_td) or impulse (_imp) files with any other type of noise files, such as (_xtk), (_td), (_rfi) or (_imp) on the same channel.
- X-Talk files can be combined with other xtalk files (_xtk) and with RFI files (_rfi).

3.7.1.1 Clear

Click the ***Clear*** button to delete all listed noise profiles from the combiner window.

3.7.1.2 Copy from

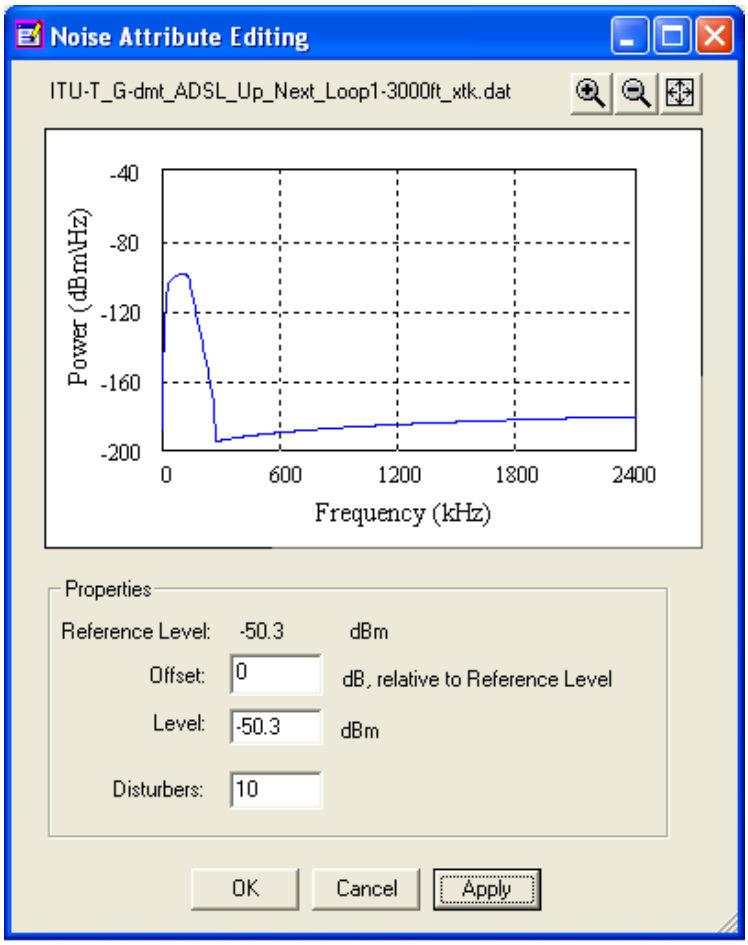
To apply the noise characteristics of a previously loaded channel to another channel, select the channel in the ***Copy from*** drop-down list. The noise profile from the selected channel will be loaded into the combiner window.

3.7.1.3 Calibration Impedance

Noise files are calibrated and measured using different load impedance values, which are outlined in the various xDSL standards. To identify which calibration impedance should be selected, refer to the relevant xDSL standard.

The DLS-5800 allows you to select one of three calibration impedance values using the ***Calibration Impedance*** drop-down list: 100 Ohms, 135 Ohms, ETSI Load

Note: The DLS-5800 allows just one calibration impedance value to be specified per channel. ETSI complex impedance noise files can only be combined with other ETSI complex impedance noise types.



3.7.1.4 Edit

Click ***Edit*** to modify various parameters of the currently selected noise file. You can edit noise levels or disturber quantity either by double-clicking on the appropriate row of the combiner window or by selecting a table entry and clicking the ***Edit*** button. When you click the ***Edit*** button, the *Noise Attribute Editing* window appears:

- Reference Level: the value displayed identifies the calculated power level for a specified noise file and is considered to be the 0 dB reference level using a specified load.

- Offset dB Relative to Reference Level: enter the level (in dB) for the specified noise in the text box and click ***Apply***. Placing a value in this field changes the total power based on the specified reference level of the noise file.
- Level: This field identifies the reference power level of the noise selected. This value is also known as the 0 dB reference level across the specified calibration impedance. Enter a new level for the specified noise in the text box and click ***Apply***.
- Disturbers: NA (grayed out)

3.7.1.5 Remove

Click ***Remove*** to remove the currently selected noise profile from the combiner window.

3.7.2 Combined Noise Region

The *Combined Noise* region (Figure 3-4) provides a visual indicator of the noise characteristics for the noise files listed in the combiner window. The combined noise parameters are only visible after a *Target Channel* has been selected followed by clicking **Calculate** in the *Noise Calculation* region.

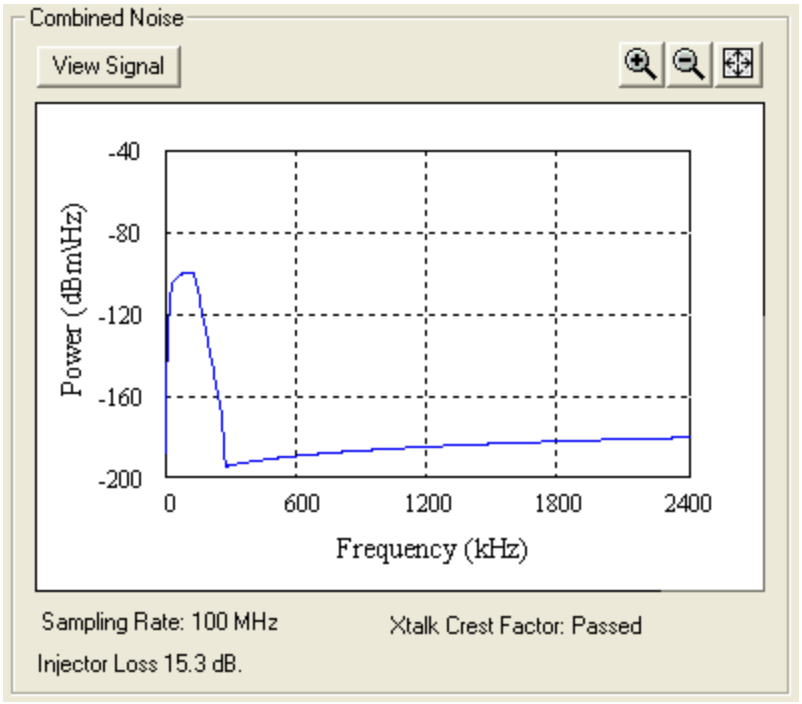


Figure 3-4: Combined Noise region

The areas of the *Combined Noise* region are explained in the following sections.

3.7.2.1 View Signal

Click ***View Signal*** to produce a plot that shows various characteristics of the Noise

Sample, as it is downloaded onto the AWG noise card.

3.7.2.2 Sampling Rate

This value is fixed and is determined by the file type being used. This value is displayed as 12.5 MHz, 32 MHz, or 100 MHz.

3.7.2.3 Xtalk Crest Factor

If crest factor checking is enabled, if the crest factor is equal to or greater than 5, this field will display Passed".

3.7.3 Noise Calculation Region

The controls in this region are explained in the following sections.

3.7.3.1 Crest Factor Greater Than or Equal to 5

Check the **Crest Factor >=5** checkbox to ensure that generated noise samples have amplitude distribution compliant with the 2nd-generation ETSI xDSL standards (such as SDSL).

ETSI has specified this amplitude distribution by means of a tight mask, to give meaning to concepts like "near Gaussian" and "crest factor above 5". This compliance is achieved using an iterative algorithm for calculating the noise sample. This algorithm repeats up to 5 times to try to ensure that the crest factor of the calculated sample is >=5. If it succeeds, "Passed" is displayed (in the Combined Noise region under the graph). If after 5 tries the crest factor is still not satisfactory, the display indicates "Failed".

Note: When ETSI-compliant crosstalk noise (crest factor >5) is combined with ingress noise (for example, a single RFI tone at very high level), the crest factor of this combined noise may be lower than the crest factor of crosstalk noise. Therefore, always deactivate the ingress noise when checking ETSI compliance of the crosstalk noise.

These figures show how the cumulative distribution function may be different between an ETSI and non-ETSI compliant noise sources. To be ETSI compliant, the cumulative distribution function must fit completely between the two limits of the mask.

Note that the greater the number of samples, more likely the calculation will be able to successfully resolve the crest factor requirement.

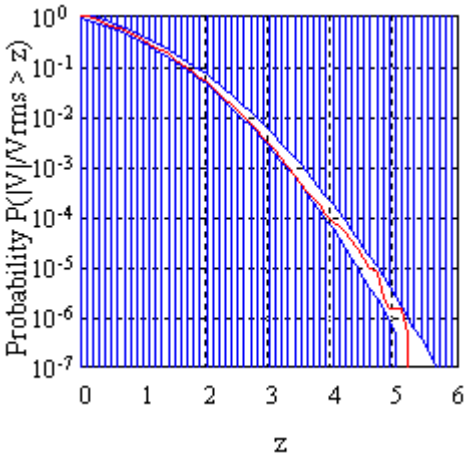


Figure 3-6: CAD of non-ETSI-compliant sample

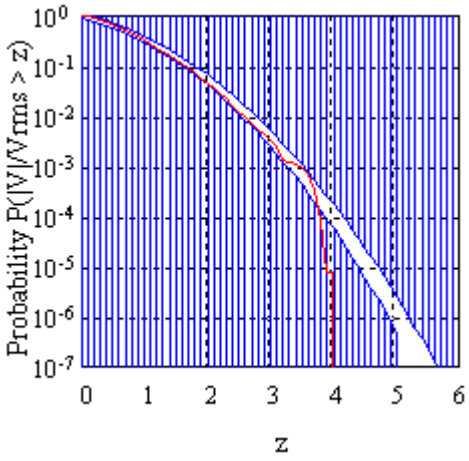


Figure 3-5: CAD of an ETSI-compliant sample

3.7.3.2 Sample Number

Choose the number of points in the calculated noise sample. The more samples, the more accurate the noise sample, however, the time to calculate the sample will increase.

3.7.3.3 Target Channel

Select the generator channel for noise output. Your DLS-5800 can have 2, 4, or 6 AWG modules, providing 8, 16, or 24 channels.

3.7.3.4 Calculate

Click the **Calculate** button to recalculate the noise sample from the noise profiles and settings in the combiner window.

NOTE for DLS-5410DC users: the calculation is affected by the coupling mode selection made on the DLS-5410 Control dialog. Always select the proper coupling mode (Common or Differential) BEFORE calculating.

3.7.4 Noise Output

Click the **Generate Noise** button to send the current Noise Sample to the AWG card, and tell the hardware to start the generation of the output signal to the *Target Channel* you selected in the *Noise Calculation* region. Use this button in conjunction with the Channel Control pull down located in the top left of the DLS-5800 Control Software to start or stop the actual generation of the noise signal.

NOTE that the Generate button only starts output from the DLS-5800 Noise Generator. If you are using DLS-5410DC, in order to connect the output to the wireline, you must switch on the appropriate DLS-5410 channel using the DLS-5410 Control dialog. If you are using DLS-5409 you don't need to do any further action, as the DLS-5409 is 'always on'.

3.7.5 Channel Specific Settings

When the Channel Control menu (*Figure 3-7*)is pulled down, you will see that any channels which are actively generating noise are highlighted with a check mark. Selecting a channel from the menu will display the Channel Output dialog for that channel. This dialog provides status information for that channel and also displays the current noise type/file settings (*Figure 3-8*).

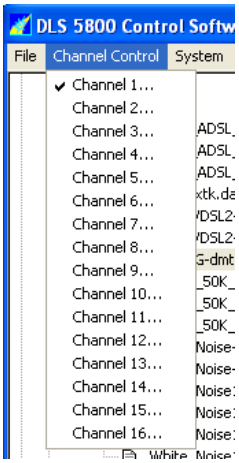


Figure 3-7: Channel Control pull down showing Channel 1 active

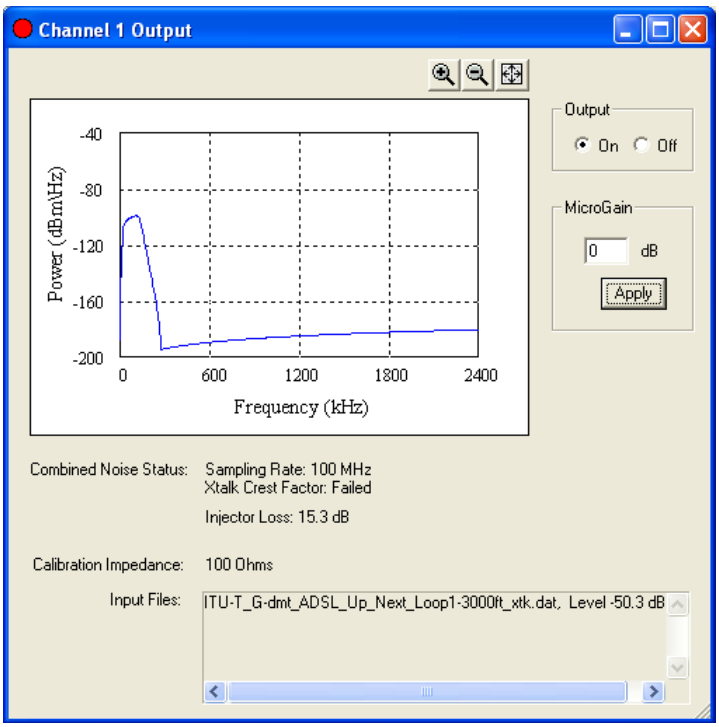


Figure 3-8: Channel output dialog

3.7.5.1 Output On/Off

Click the **Off** radio button to stop the generation of the output signal by the DLS-5800 AWG noise card. Click the **On** radio button to resume the output. Again, note that this button deals only with the noise generator, not with the DLS-5410DC injector.

3.7.5.2 MicroGain

The *MicroGain* field allows you to enter small power level changes without requiring a recalculation of the Noise Sample. This allows you to make power level changes with virtually no delay.

The *MicroGain* field increases or decreases the aggregate output power level, and is used for Time Domain or Frequency Domain Crosstalk files. Time Domain based noise types have a range of -3 dB to 7 dB. Frequency based noise PSDs have a range of -3 dB to 9 dB.

If you need to change the power level by an amount that exceeds the MicroGain range, edit the *Level* or *Offset* in the *Noise Attribute Editing* window, then recalculate and regenerate the output.

Impulse noise types are not supported with MicroGain functionality.

3.7.5.3 Spectral Monitor (Graph)

The display within the channel output window (*Figure 3-8*) allows you to monitor the following spectra of the noise shape you are fine tuning.

- The blue line indicates the desired spectrum, as specified in the Noise Profile.
- The black line denotes the spectrum of the actual Noise Sample as it has been downloaded into the DLS-5800 hardware. This is a calculated spectrum, but it is based on the actual Noise Sample. The black curve provides a realistic prediction of how the output signal will look when measured with a spectrum analyzer.

When selecting Time Domain profiles, the graph displays amplitude vs. time.

3.8 Custom Noise Files

The DLS-5800 Control Software allows users to save the calculated noise sample for later use. The advantage of this is that loading the pre-calculated sample is much faster than loading noise profiles and then recalculating a new sample.

Any valid calculated sample can be saved as a custom file.

To save a custom noise file:

1. Define your profiles then calculate the noise sample.
2. Select File | Save Custom….
3. Enter the desired descriptive text when the following prompt appears: *Input your Noise Information*.
4. Click **OK**.
5. When the Save As dialog appears, choose the file name and location.

The file is saved with an extension _cst.enc, which identifies it as a custom noise file.

Note: Because *-cst files are time domain data, the DLS-5800 Control Software does not graphically display custom generated files.

3.9 Combining Several Noise Files with .NCD Files

A .NCD file is a text file which allows you to combine noise files for later recall. An .ncd file can be used to combine several noise files, each with a specified power level. The .ncd files can be used to recall a specific combination of up to six _xtk files and one _RFI file, and are loaded just like regular noise files.

The same rules that apply to combining noises in the work space also apply to .NCD files:

Up to 7 Crosstalk (XTK) and Radio Frequency Interference (RFI) files can be combined.

Only 1 Time Domain (TD) file, Impulse (IMP), or Custom Saved File (CST) can be loaded at any time.

NOTE: user may combine TD and Impulse noise types together if the sample rate of each file is identical.

To create an .ncd file, use a text editor such as Notepad and type out a combining specification, then save it with an .ncd extension.

Four file format options are illustrated below.

For frequency domain noise files like *_xtk* or *_rfi*, these file types must contain file headings such as:

$name

$offset

$disturber

If a file type contains both the "*$offset"* and "*$disturber"* heading within the same .ncd file, the last heading will overwrite the previous value since both offset and disturber quantity values are directly related to each other.

Example: Xtalk and RFI noise files

```
$name<C:\Program Files\Spirent Communications\DLS 5204\
 NoiseFiles\DSL  Forum\ADSL_TR-048  (Apr  2002)\8-1-
 1_White_Noise_Impairment\Downstream_At_ATU-R\
 White_Noise_xtk.enc>

$offset<20  dB>

$name<DLS5B17\1.0\NoiseFiles\Broadband forum\ADSL_TR-048 (Apr
 2002)\8-1-2_24_HDSL_Impairment\Downstream_At_ATU-R\
 HDSL_Next_24DPlusWN_xtk.enc>
$disturber<24>
```

Example: Time Domain noise files

```
$name<C:\Program Files\Spirent Communications\DLS 5204\
 NoiseFiles\DSL  Forum\ADSL_TR-048  (Apr  2002)\8-8-
 1_Impulse_Tests\  Downstream_At_ATU-R\
 HDSL_Next_20DPlusWN_td.enc>
$offset<-2  dB>
```

Example: Time Domain noise files with Impulse noise

```
$name<C:\Program Files\Spirent Communications\DLS 5204\
 NoiseFiles\DSL  Forum\ADSL_TR-048  (Apr  2002)\8-8-
 1_Impulse_Tests\Downstream_At_ATU-R\
HDSL_Next_20DPlusWN_td.enc>
$offset<-2 dB>

$name<C:\Program Files\Spirent Communications\DLS 5204\
 NoiseFiles\Base  Noise  Files\ADSL-C1_imp.dat>
$level<100  mV>
$  impulse  repeat  number  <15>
```

Example: Impulse noise only

```
$name<C:\Program Files\Spirent Communications\DLS 5204\
 NoiseFiles\Base  Noise  Files\ADSL-C1_imp.dat>
$level<50  mV>
$impulse  rate<50  pps>
$impulse  repeat  number<15>
```

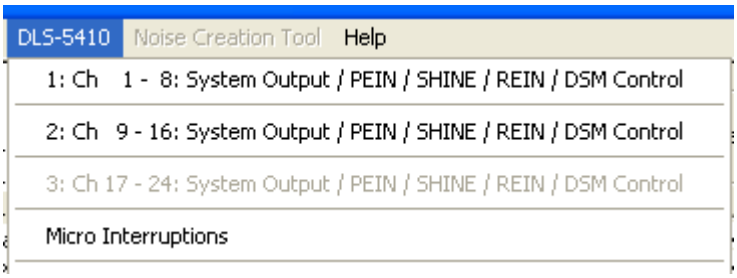


Figure 3-9: DLS-5410 pull down for a 4-card DLS 5800 system

Note: When selecting Xtalk and RFI noise types, Impulse and Time Domain noise must not be combined. An error message will appear if you attempt to combine these different noise types.

3.10 Controlling DLS-5410DC Features

The DLS-5410DC adds differential- and /or common- mode noise injection, channel switching, PEIN, SHINE, REIN, and DSM capabilities to the DLS-5800.

NOTE: The DLS-5410DC provides accurately-timed switching of signals generated by DLS-5800. It does NOT generate noise on its own.

To control these features, select DLS-5410 and choose the injector which affects the channel you want to use.

For example, to control channels 1-8 choose "1: Ch 1 – 8: System Output / PEIN / SHINE / REIN / DSM Control". The DLS-5410 Control dialog appears:

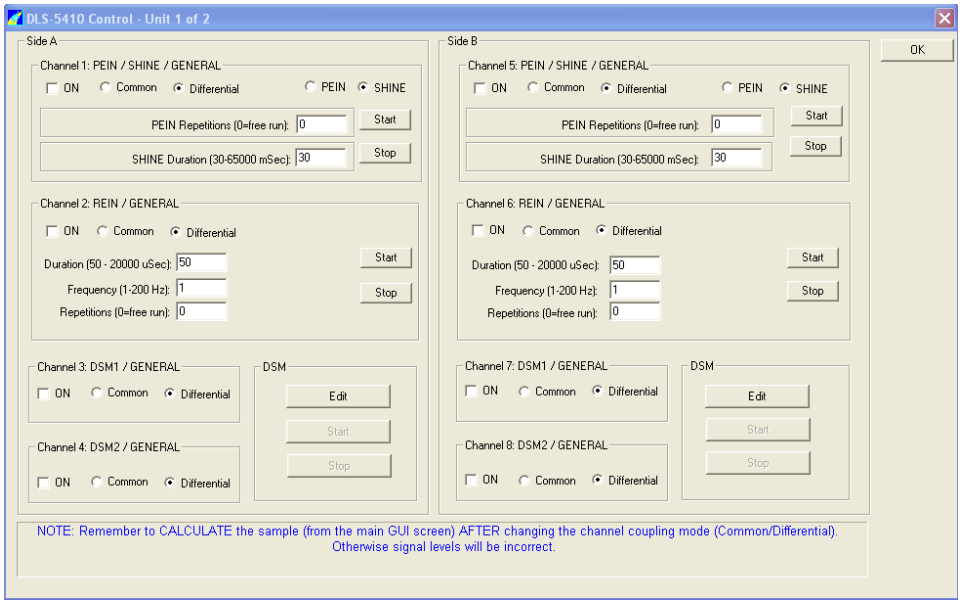


Figure 3-10: DLS-5410 Control Dialog

3.10.1 DLS-5410DC Channels

The DLS-5410DC allocates channels for specific purposes using the following table:

Channel (Injector 1 / 2 / 3)	Purpose
1 / 9 / 17 (Side A Channel 1)	PEIN / SHINE / General purpose
2 / 10 / 18 (Side A Channel 2)	REIN / General purpose
3 / 11 / 19 (Side A Channel 3)	DSM1 / General purpose
4 / 12 / 20 (Side A Channel 4)	DSM2 / General purpose
5 / 13 / 21 (Side B Channel 1)	PEIN / SHINE / General purpose
6 / 14 / 22 (Side B Channel 2)	REIN / General purpose
7 / 15 / 23 (Side B Channel 3)	DSM1 / General purpose
8 / 16 / 24 (Side B Channel 4)	DSM2 / General purpose

Table 3-2: DLS-5410 function / channel allocations

Note that the following instructions refer to the entire set of available channels for a 6-card (24 channel) DLS-5800/DLS-5410DC. The exact available channels will change depending on your hardware configuration.

3.10.1.1 General purpose:

General purpose usage means that the DLS-5410DC injector input, when switched on, is connected directly to the appropriate side output without any further switching or processing. Any of the 8 inputs on the DLS-5410DC can be used in this way. General

purpose usage means that the signal applied by the generator is not processed in any way by the DLS-5410DC.

Additionally, as shown previously, the DLS-5410DC offers additional functions on several of its channels:

3.10.1.2 PEIN (Prolonged Electrical Impulse Noise)

The DLS-5410 generates PEIN signals according to TR-114 Issue 1 (D.3.3).

1. On the DLS-5410 Control dialog, ensure that the channel you wish to use is turned OFF (check box is clear, not checked). Valid channels for PEIN are 1, 5, 9, 13, 17, and 21.
    1. Using the main GUI of the DLS-5800 Control Software, load and generate the file "White_Noise_Pein-90dBm-Hz_xtk.dat" on one of the channels that corresponds to an injector channel that supports PEIN (channels 1, 5, 9, 13, 17, or 21). Note that this file is designed for the WT-114 PEIN noise implementation, and if a different file is used, the output may not meet WT-114 specifications.
    2. On the DLS-5410 Control dialog, click the PEIN radio button, set the number of repetitions, then click Start.

       NOTE that a single repetition of the PEIN pattern will take around 35 minutes to complete.

3.10.1.3 SHINE (Single High Impulse Noise Event):

1. On the DLS-5410 Control dialog, ensure that the channel you wish to use is turned OFF (check box is clear, not checked). Valid channels for SHINE are 1, 5, 9, 13, 17, and 21.
    1. Using the main GUI of the DLS-5800 Control Software, define, load, and generate an appropriate noise sample on one of the channels that corresponds to an injector channel that supports SHINE (channels 1, 5, 9, 13, 17, or 21).
    2. On the DLS-5410 Control dialog, click the SHINE radio button, set the duration of the signal, then click Start.

3.10.1.4 REIN (Repetitive Electrical Impulse Noise)

1. On the DLS-5410 Control dialog, ensure that the channel you wish to use is turned OFF (check box is clear, not checked). Valid channels for REIN are 2, 6, 10, 14, 18, and 22.
2. Using the main GUI of the DLS-5800 Control Software, define, load, and generate an appropriate noise sample on one of the channels that corresponds to an injector channel that supports REIN (channels 2, 6, 10, 14, 18, or 22).
3. On the DLS-5410 Control dialog, set the desired duration, frequency, and number of repetitions of the signal. Click Start.

3.10.1.5 DSM (Dynamic Spectrum Management)

Modems which implement Dynamic Spectrum Management (DSM) change characteristics

of their transmitted signals depending on the line conditions that they encounter. This will cause corresponding changes in the resulting crosstalk seen on other cables within the binder.

When used with the DLS-5410DC, the DLS-5800 is capable of simulating this changing noise environment by allowing you to create individual noise profiles, called “frames”, which will be stitched together in a DSM “sequence”**.** Each frame can play for a specific amount of time from 30 to 300 seconds. When the DSM sequence is played back, the DLS-5800 will automatically load each frame and play it for the specific amount of time. In order to ensure that the transitions between frames are seamless, the DLS-5410DC switches between 2 generator channels.

1. To use DSM, you must first create all the frames that you wish to use in your sequence. This is done from the main GUI of the DLS-5800 Control Software.

   For each frame:

   a) Load the noise files you wish to use

   b) Choose a target channel and click Calculate.

   c) Select File | Save DSM Frame…

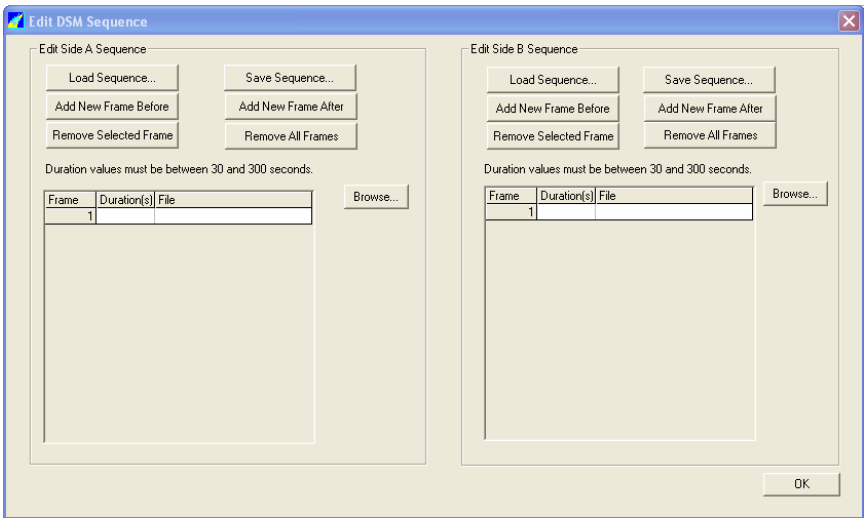
   d) Optionally enter some descriptive text at the “Input your Noise Information” prompt

   e) Click OK.

   f) Enter a name for the DSM frame. Frame files must be stored in “C:\Program Files\Spirent Communications\DLS 5800\DSM Frames.

2. Once you have created all the frames, you now need to define a DSM sequence.

   a) Select DLS-5410 then select the injector which affects the channels you wish to use. Click the appropriate Side A or Side B Edit button under DSM.



   b) The “Edit DSM Sequence” dialog appears:

Figure 3-11: Edit DSM dialog

c) Each side shows a grid that displays the current DSM sequence and the file and duration associated with each frame in the sequence. Each row in the grid represents a frame.

d) Double click in the Duration column then type a value, in seconds, representing for how long the frame should play back.

e) Click in the File column, and click Browse. Choose one of the frames that you previously saved.

f) Use the Add New Frame Before / After buttons to add frames before or after the currently active frame. Use Remove Selected frame to remove the currently active frame, or Remove All Frames to clear the entire sequence. If no frame is active, the active frame is assumed to be Frame 1.

g) Repeat steps d), e), and f) until you have defined all the frames in the sequence.

h) If you wish to be able to recall the sequence, use the Save Sequence button to save the DSM Sequence into a file. DSM Sequence files must be stored in "C:\Program Files\Spirent Communications\DLS 5800\DSM Sequences. You can then use the Load Sequence button to recall the sequence.

i) Click OK to close the dialog.

3. On the DLS-5410 Control dialog, ensure that the channel pair you wish to use is turned OFF (check boxes are clear, not checked). Valid channel pairs for DSM are 3-4, 7-8, 11-12, 15-16, 19-20, and 23-24.

4. On the DLS-5410 Control dialog, click the Start button (under DSM) corresponding to the channels you wish to use.

3.10.1.6 Micro Interrupts

Micro interrupts cause a very short interruption in the end-to-end continuity of a wireline. Some testing standards require the application of repeated micro interruptions within a specified amount of time. A micro interrupt can be specified to last 1 - 60,000 ms.

The noise injector is updated with new micro interrupt parameters whenever you press **Enter** or move from one field to another.

The *Side A* and *Side B* radio buttons select the noise injector (output) side at which the micro interrupt will take place.

To start the micro interrupt sequence, click the **Start** button. The first micro interrupt is generated immediately, and then periodically at the frequency specified in the *Interrupt every* field. The total number of micro interrupts is specified in the *Total number of interrupts* field.

Note: The *Interrupt Time* field must be shorter than the *Interrupt every* field.

When the Start button is clicked, the DLS-5800 Control Software verifies that the micro interrupt values entered are valid, and that the *Interrupt Time* field is shorter than the *Interrupt every* field.

3.11 Remote Control

Although the DLS-5800 Control Software is very easy to use, in some cases you may want to control the DLS-5800 / DLS-5410DC from a remote computer. This is especially useful when integrating the DLS-5800 / 5410 into a larger automated test environment.

The DLS-5800 Control Software has 2 modes – local and remote. Remote mode allows the DLS-5800 to be controlled from a remote computer. For a detailed explanation of remote control, see *Remote Control.*

Note: The DLS-5800 must be in remote mode before sending remote commands to the unit.

Remote mode can operate in one of two ways:

Single-session operation means that only one session (or instance) of the software runs and all the generator and injector channels are available to that session.

Multiple-session operation launches more than 1 session of the software. The number of sessions depends on your DLS-5800 / DLS-5410DC hardware configuration. A new session of software is created for each DLS-5410DC Noise Injector that is connected.

Thus, if your DLS-5800 contains 16 channels, you need 2 DLS-5410 injectors, and multiple-session mode creates 2 sessions. If your DLS-5800 contains 24 channels, you need 3 DLS-5410 injectors, and multiple-session mode creates 3 sessions. Session 1 will control channels 1 to 8, session 2 will control channels 9-16, and session 3 will control channels 17-24.

Note that if your DLS-5800 configuration contains 8 channels, multiple-session mode is not supported, as there is only 1 DLS-5410 Noise Injector.

To use the remote-control mode, first configure the DLS-5800 via System | Properties.

Then select **System | System Mode** to choose Remote mode and later return to Local mode. Note that if you are running Multiple-session remote mode, you can return to Local only from the Main-session GUI.

3.11.1 System Properties

The System Properties dialog configures several settings mainly dealing with Remote Control mode.

System Identifier: This is a string of up to 30 characters that you can use to identify the DLS-5800. There are remote commands that can be used to retrieve this string.

Remote to Local Password: This is a password to prevent inadvertently exiting Remote mode.

Retype Password: This field must match the Remote to Local Password. This helps prevent typing errors when setting the password.

NOTE: If you forget your password, edit the “DLS5204Value.dat” file, and remove the “RemotePassword: xxxxxxx” line.

Start DLS-5800 when unit boots up: When checked, whenever the system is switched on, the DLS-5800 Control Software will start automatically.

The default startup mode is: If the system is configured to start DLS-5800 on startup, these buttons let you choose whether the software will start in local or remote mode.

Remote Mode: The DLS-5800 can run in either single- or multiple-session remote control mode.

Remote Port: This is the port value to which you must connect in order to control the main session of the software.

Remote Port (2nd instance): This is the port value to which you must connect in order to control the slave 1 session of the software. This field is not used in Single-session mode.

Remote Port (3rd instance): This is the port value to which you must connect in order to control the slave 2 session of the software. This field is not used in Single-session mode or if only 2 sessions are available.

Once you have finished setting the System Properties, click OK to close the dialog.

NOTE The *DISABLE_REMOTE* command will not work when using the multiple-session remote mode.

3.11.2 Display Remote Commands

The DLS-5800 allows you to display the remote control commands sent to the noise generator. Choose Help | Remote Command…

3.12 Licenses

Spirent Communications uses a license checking scheme to protect its intellectual property from unauthorized copying. Spirent supplies you with licenses for purchased software and noise file packages. These licenses allow use of the covered product on your specific target platform.

To view installed licenses or add licenses**:**

1. Select System | License…
2. The *License Management* dialog box appears:

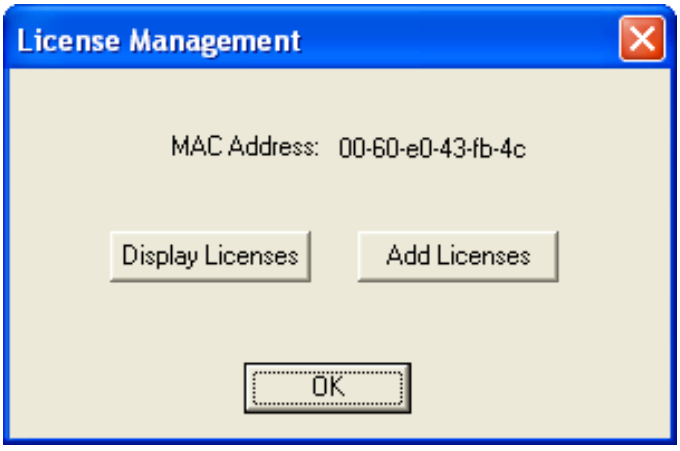


Figure 3-12: License Management Dialog

3. Click the **Display Licenses** button to see a list of currently loaded licensed packages for this DLS-5800. The *License Information* window appears. For each license installed, the *License Information* window displays the product name, platform type, identification, expiration date, and key.

4. Click the **Add Licenses** button to add additional noise licenses.

Note: You can also access this window without starting the DLS-5800 Control Software. Select **Start | Programs | Spirent Communications | DLS 5800 | License Info**.

Warning: Licensed noise files supplied by Spirent Communications must not be moved in the directory structure. A licensing failure will occur if any of the *.enc (encrypted) file types are moved.

Warning: Do not install a second NIC card into the DLS-5800 chassis. This will cause license checking to fail, you will no longer be able to start the DLS-5800 Control Software, and your noise file licenses will no longer be valid.

Chapter 4 Crosstalk Noise Profiles

4.1 Introduction

A noise profile is a Power Spectral Density (PSD) description of the crosstalk noise as it is observed at the receiver of the xDSL modem under test (near the point of injection). It is generated and applied as follows:

The DLS-5800 reads a file carrying a noise profile that describes what kind of noise has to be generated (this is in ASCII/ frequency vs. power)

The DLS-5800 synthesizes and generates the noise PSD and applies it, via a noise injection mechanism, onto the wireline or simulated wireline near the receiver of xDSL modem under test. The DLS-5800 automatically takes the loss of the noise injector into account.

In older xDSL standards (for example, ISDN & HDSL), PSDs were few and relatively generic. This was done to simplify testing and had the consequence of not being very representative of real access network conditions.

Today, with the advent of automated test set-ups and higher precision test equipment, xDSL Standard's organizations now define noise PSDs that include changing test loop length and type as well as changing bitrate or transmission direction for the xDSL modem under test. The result is a more realistic set of noise shapes, as a set that sometimes numbers in the thousands of shapes.

Therefore, to simplify xDSL testing, noise profiles are stored in separated files so the DLS-5800 can load a dedicated or specific noise profile for each required test. Spirent Communications has developed noise libraries for most xDSL test requirements.

As an example of noise PSD development, the following section describes the way ETSI defines noise profiles, provides some sample noise profiles for SDSL, and provides the associated DLS-5800 file format of these noise profiles.

4.2 Defining Noise Profiles Using Building Blocks

The profile of the crosstalk noise defined in current ETSI xDSL standards varies with the length of the test loop. The rationale behind this is that the FEXT coupling function between the wire pairs in a real cable is length dependent, and this can have a significant impact on the crosstalk noise when upstream and downstream signals of xDSL modems do not use the same frequency band.

Figure 4-1 shows how various ETSI standards compose the crosstalk noise from isolated building blocks, and is used for impairment testing in downstream as well as in upstream direction.

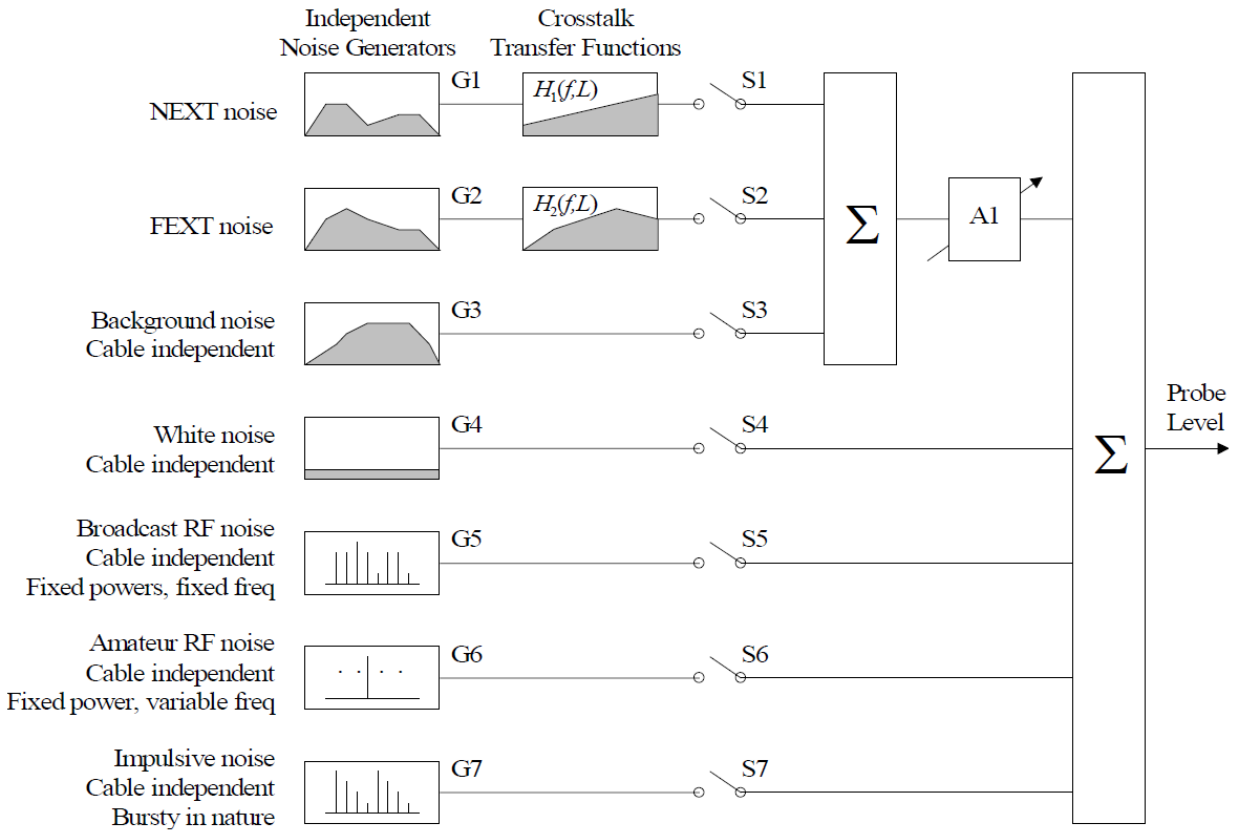


Figure 4-1: ETSI Functional Description of Crosstalk Noise Calculations

- Generator G1 is an equivalent noise source that represents the combined equivalent disturbance of a mixture of impairing xDSL systems, located at the *receiver* side of the xDSL modem under test. Its noise spectrum is defined in the appropriated ETSI standards, and may vary with the bitrate of the xDSL modem under test.
- Generator G2 is similar to G1, but represents the combined equivalent disturbance at the *transmitter* side of the xDSL modem under test.
- Generator G3 represents background noise, (if any). For now, ETSI currently sets background noise to zero.
- Generator G4 represents white noise. To generate white noise, test equipment must be designed with an extremely low-noise floor (generally at -140 dBm/Hz or lower).
- The transfer function $H_1(f,L)$ models the length and frequency dependency of the NEXT crosstalk coupling. The transfer function is dependent of the loop-set number and changes *slightly* with the length of the test loop, and is well defined in ETSI xDSL standards.
- The transfer function $H_2(f,L)$ models the length and frequency dependency of the FEXT crosstalk coupling. The transfer function is dependent of the loop-set number and changes *significantly* with the length of the test loop, and is well

defined in ETSI xDSL standards.

The transfer functions $H_1(f,L)$ and $H_2(f,L)$ are defined in Table 4-1for calculating the NEXT and FEXT coupling functions.

Table 4-1: Definition of Crosstalk Transfer Functions

$H_1(f, L) = K_{xn} \times (f/f_0)^{0.75} \times 1 - \lvert sT(f, L)\rvert 4)$ $H_2(f, L) = K_{xf} \times (f/f_0) \times (L/L0)) \times \lvert s_T(f, L)\rvert$
$K_{xn} = 10^{(-50/20)} \approx 0.0032$, $f_0 = 1$ MHz $K_{xf} = 10^{(-45/20)} \approx 0.0056$, $L_0 = 1$ km $s_{T0}(f, L)$ = test loop transfer function

Because the signals or transfer functions of the individual building blocks change when the test conditions are altered, the output noise of the impairment generator changes as well. This occurs when changing:

- from upstream testing to downstream testing
- the test loop
- the test loop length
- the bit-rate of the xDSL modem under test
- the environmental scenario from noise model "A" to "B" or "C", etc.

4.3 Definitions of Alien, Self, and Full Noise

Noise generators G1 and G2 represent the combined equivalent disturbance of a mixture of impairing xDSL systems. Usually, the xDSL modem under test is one of them. When changing the bit-rate of the xDSL modem under test, its transmission spectrum may change as well. In the ETSI noise scenarios, it is assumed that such a modem is deployed in a cable, connected to similar systems. So the crosstalk noise may change as well when the bit-rate of the xDSL modem under test changes. To cope with this in a convenient way, ETSI xDSL standards identified the following crosstalk components:

- **Alien noise:** is the combined equivalent disturbance caused by modems that are different from the xDSL modem under test. For instance, using ADSL and HDSL disturbers when testing SDSL. Recent ETSI xDSL standards define the Alien noise by means of a simple table, describing their PSD.

- **Self noise:** is the combined equivalent disturbance caused by modems that are equal to the xDSL modem under test. For instance, using SDSL disturbers when testing SDSL at the same bit-rate of the SDSL modem under test. If Self noise behavior conforms the nominal PSD values specified in the standard, it can be calculated in advance; if not, it has to be measured from the xDSL modem under test.

- **Full noise:** is the combined equivalent disturbance from both Alien and Self-disturbers. This is a noise type that is composed of the outputs of generators "G1" and "G2" in *Figure 4-1*.

Combining Alien and Self noise into Full noise is *not* a linear power sum of all individual disturbers, since only one disturber can occupy the worst-case wire pair (from a crosstalk coupling point of view). Therefore a *weighted* power sum is specified in recent ETSI

xDSL standards.

When Self noise has to be evaluated by measuring the PSD of the modem under test, then a noise profile library cannot provide the Full noise in advance. It must be restricted to Alien noise only, and the you must convert it into Full noise before performance testing can start.

4.4 Description of the Noise Profile File Format

The ASCII format to describe a noise profile is very simple. It consists of a number of lines, each containing two numbers separated by spaces or tabs. In this way, two columns are formed. The first column describes a number of frequency values (in Hz); the second column contains the corresponding value for the power spectral density. This second column (the PSD) may be expressed in either dBm/Hz or in V/sqrt(Hz). The DLS-5800 distinguishes between the two by the *sign* of the numbers in the second column. If these numbers are negative, they are interpreted as amounts of dBm/Hz; if they are positive, they are interpreted as amounts of V/sqrt(Hz).

3. Optionally, the file may contain a line of which the frequency value is negative. This tells the DLS-5800 to interpret the second number on this line as the reference impedance. If this negative symbol is not in the *Frequency* column, the noise file will not generate.
4. An example of a frequency vs. power is shown in *Table 4-2*.

Table 4-2: Example of Noise Profile in ASCII Formats

Frequency (Hz)	Power (dBm/Hz)
999	-140
1e6	-140
1.00001e6	-70
4e6	-70
5.0000e6	-140
-1	50

The DLS-5800 interpolates between the values provided in the noise profile ASCII file. By default, the interpolation scheme is based on 'linear frequency, linear dB.' See *Figure 4-2* for a picture of the noise profile corresponding to the data in *Table 4-2: Example of Noise Profile in ASCII Formats*.

You can create custom noise files using several formats, which include .txt editors (Notepad) or Microsoft® Excel® and so on. The file format outlined in *Table 4-2* must be saved as an **_xtk.dat** extension, similarly for RFI ingress these file types would be saved as **_rfi.dat** extensions. These

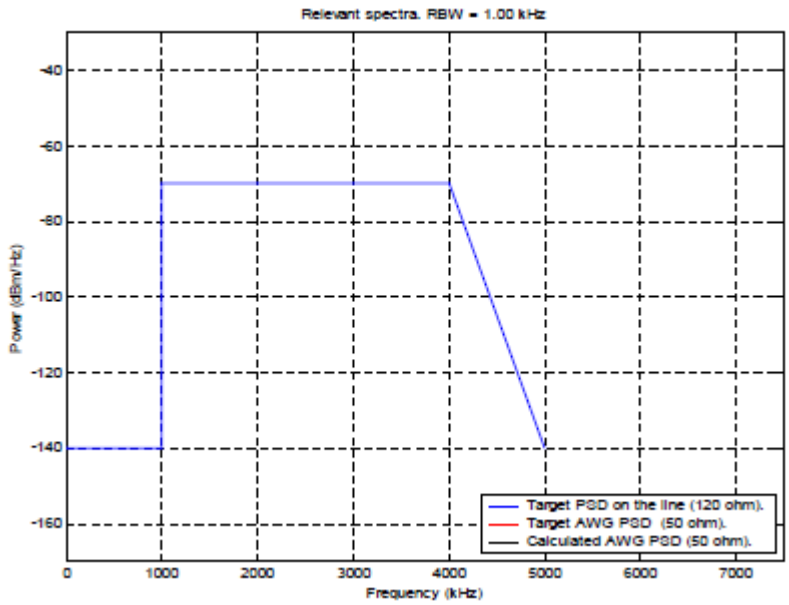


Figure 4-2: Sample PSD Calculated Profile

files can be saved in any directory specified, and can be recalled using the DLS-5800 software.

Chapter 5 Ingress Noise Profiles (Optional)

5.1 Introduction

An ingress noise profile is an RFI-tone description of the ingress noise, as observed at the receiver of the xDSL modem under test (near the point of injection). The Spirent RFI technology, available with DLS-5B36, synthesizes this ingress noise according to the associated profile. The profile is used as follows:

- In order to use Ingress (RFI) noise, you must have the DLS 5B36 license (the older DLS 5B14 is no longer supported). The section *"DLS 5B36 RFI Modulation Method" below* describes the modulation method.
- DLS-5800 reads a file containing a noise profile that describes what kind of noise has to be generated (this is in ASCII / frequency vs. power)
- DLS-5800 synthesizes and generates the noise PSD and applies it via a noise injection mechanism onto the wireline or simulated wireline near the receiver of xDSL modem under test.

Different xDSL standards use different RFI-tone descriptions. To simplify xDSL testing, ingress noise profiles can be stored in separate files, so that for each test the DLS-5800 can select and load a dedicated ingress noise profile from a library with ingress profiles. You define noise profiles as explained in the following sections.

5.2 Definition of the RFI Tones in Ingress Noise

The ingress noise generated by the DLS-5800 software is a superimposition of a number of random modulated carriers (AM). The total voltage $U(t)$ of this noise is defined as:

$$U(t) = \sum_k U_k \times cos(2\pi \cdot f_k \times t + \varphi_k) \times (1 + m \times \alpha_k(t))$$

The individual components of the ingress noise voltage $U(t)$ are defined as follows:

U_k

The voltage U_k of each individual carrier is specified as a *parameter* in the ingress noise profile by means of a power level P_k (dBm) into a resistive load of R. Note that spectrum analyzers will detect levels that are slightly higher then the values of the individual carriers when their resolution bandwidth is set to 10 kHz or more, since they will detect the modulation power as well.

f_k

The frequency f_k of each individual carrier is specified as a *parameter* in the ingress noise profile.

φ_k

The phase offset φ_k of each individual carrier is a random value that is uncorrelated with the phase offset of each other carrier in the ingress noise signal.

m

The modulation depth m of each individually modulated carrier is specified as a *parameter* in the ingress noise profile. This value is a RMS-modulation depth; the modulation index equals the peak levels of the modulation signal $m \times \alpha_k(t)$.

$\alpha_k(t)$

The normalized modulation noise $\alpha_k(t)$ of each individually modulated carrier is random and

Gaussian distributed in nature, has an RMS value of α_{rms}=1, has a crest factor of 2 to 5 or more, and is uncorrelated with the modulation noise of each other modulated carrier in the ingress noise signal.

Δb

The modulation width Δ*b* of each modulated carrier shall be close to 2 x 5 KHz. This is equivalent to creating α_k (*t*) from white noise, filtered by a low-pass filter with a cut-off frequency at Δ*b*/2 and having a rectangular filter shape. This modulation width represents the *double sided* modulation band used by AM broadcast stations.

5.3 Description of the RFI Noise Profile File DLS 5B36 Format

The ASCII format to describe an ingress noise profile is very simple. After the header line "$data<begin> , there are 2 columns- the first column is the frequency in Hz and the second column is the power in dBm (see *Table 5-1*).

The rfi file must be named with extension "**_rfi.dat**".

Note: the modulation depth will be the maximum 0.31 or 0 (non).

Table 5-1: Ingress Noise Profile ASCII Format & example

$name<Format_Example_default_rfi>

$standard<use defined>

$ver<1.0>

$clic<>

$depth<>

$power<-10.0 dBm>

$data<begin>

300000 -10

Explanations

After the line "$data<begin>", the left column is the carrier frequency in Hz, while the right column is for carrier power in dBm.

DLS-5800 is limited to 150 discrete tones.

Since the clock is not defined, the default clock 32 MHz will be applied.

Because the reference impedance is not defined, the default impedance 100 Ω will be used. Currently, Spirent supports 100 Ω, 135 Ω and 150 Ω reference impedance.

In $power<-10.0>, -10.0 dBm is only for reference - the display power in the GUI will depend on the actual power output. $ver<> and $standard<> are also for reference only – i.e. they do not impact output at all.

$clic<> is a command for ETSI load compensation that is described in TS 101 388 documentation.

If user needs ETSI load compensation, clic command must be either $clic<ETSI100Ohm.clc> or $clic<ETSI135Ohm.clc>, where "$clic<ETSI100Ohm.clc>" means 100 Ω reference impedance while "$clic<ETSI135Ohm.clc>" 135 Ω impedance.

The above example will have the same features as below example.

Table 5-2: Additional commands explicitly setting clock and impedance to defaults

$name<Format_Example_default_rfi>

$standard<use defined>

$ver<1.0>

$clk<32 MHz>

$clic<>

$depth<>

$power<-10.0 dBm>

$data<begin>

300000 -10

-1 100

Table 5-3: Modulated carrier example

$name<Format_Example_Modulation_rfi>

$standard<DLS 5B36>

$ver<1.0>

$clk<100 MHz>

$clic<>

$depth<0.31>

$power<-5.4 dBm>

$data<begin>

1000000 -10

2000000 -12

3000000 -13

4000000 -14

20000000 -20

-1 100

If ETSI load compensation (100 Ω) is required, the fifth line of the above file can be modified from"$clic<>" to "$clic<ETSI100Ohm.clc>".

Note:

- The last line of the above file, "-1 100", will be regarded the same as "-1 50".
- When the modulation depth is applied, the depth value will be the maximum value for each carrier, that is, 0.31. If the depth is set to be 0, such as "$depth<0>", then all the carriers will be single tones at the level indicated beside the frequency. User is able to see the displayed power level in the GUI (the output total power).
- For the purpose of obtaining good quality modulation width, close to 5 KHz each side, Spirent suggests using the largest possible number of samples – the software may warn if insufficient samples are used.

5.4 Accuracy Limits of the AWG

The Spirent Arbitrary Waveform Generator (AWG) has been implemented with a wide dynamic range by using an output amplifier with a discrete number of gain factors. These gain factors are switchable and under control of the DLS-5800 software.

The hardware accuracy of these gain factors set a limit to the overall accuracy of the output level of the impairment generator. The DLS-5800 software does not compensate for these *systemic* hardware errors of the output levels, since it would require a (manual) calibration of each gain factor by a level detector with a much higher accuracy.

5.5 Reference Ingress Noise Profile

The reference tests assume that the crosstalk noise generation is deactivated, that the output levels are measured with an accurate true power meter (or true RMS-meter), and that their spectra are monitored with a spectrum analyzer.

Table 5-4: Reference Ingress Noise Profiles for Functional Checks.
Profile A1 is the same as Example 1 in *Table 5-1*

Profile	Frequency (Hz)	Power (dBm)	$depth<m>
Ingress ref Profile A1	300e3	-10	0
Ingress ref Profile A2	1000e3	-30	0.31

The test should verify that for each reference profile, the correct power levels are observed on the true power meter, and that the spectrum analyzer displays the correct spectral width.

The accuracy should be within the combined accuracy of the AWG card and the true power meter (for example, within 0.5 dB). *Figure 5-1* illustrates a spectral shape that can be observed when the signal is fed to a spectrum analyzer.

Ingress reference Profile A1 (*Table 5-1:* Ingress Noise Profile ASCII Format & example) generates a single unmodulated carrier of -10 dBm at 300 kHz. The true RMS-meter should display the same -10 dBm. If this not the case, you should first compensate for this *systematic* error of the setup by

adjusting the amplifier gain of the source settings.

See *Figure 5-1* for the spectrum observed at the output of the AWG - Profile A1, unmodulated carrier of -10 dBm.

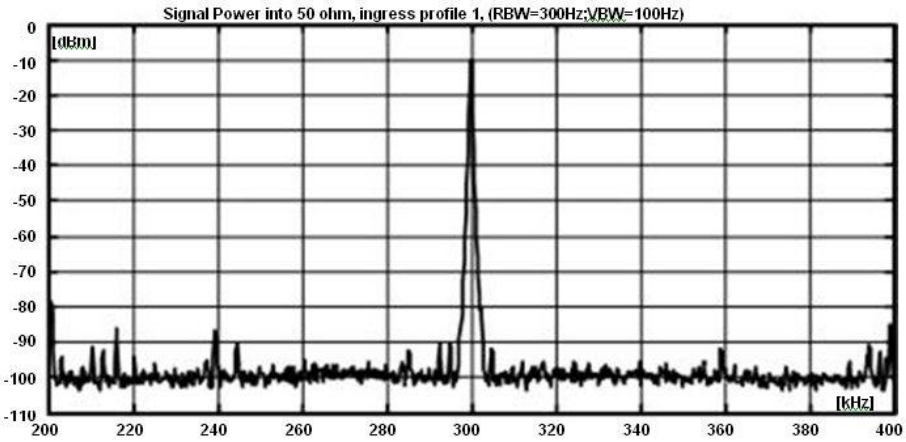


Figure 5-1: Measured Spectrum of the Reference Ingress Noise (A1) (Aggregate powerinto 50Ω P = -10 dBm = 0.1 mW)

5.6 DLS-5B36 RFI Modulation Method

This RFI modulation is described according to ETSI guidelines in *TD 23R1 from ETSI TM6 September 20 to October 3 2003*, (file name 033t23r1.pdf) titled "Text Proposal for RFI Testing.".

Note: The DLS-5800 noise generator supports RFI only via the DLS-5B36 license. The older DLS-5B14 RFI license (previously supported by DLS-5204 and DLS-5500 products) is no longer supported. Please consult Spirent Communications if necessary (see *Support Services Contact Information*).

NOTE for users of DLS-5204 / DLS-5500: Previous RFI noise generation had a non-zero modulation index specified. This will now be a fixed modulation index as required by the new ETSI proposal. Files which previously had a modulation of zero (that are intended to be a single tone) will be a single tone.

Figure 5-3 displays the frequency representation of the observed signal when using this new method.

5.6.1 Practical Modulation (Left And Right Sidebands, With 10 Lines Total)

In the actual noise spectra, there are 10 additional lines, symmetrical around the main AM carrier, all with amplitudes of .1 or 20dB below the main carrier. The total power added to the main carrier by the side lobes would be 10%, or a 0.41 power increase.

The amplitude envelope of the composite AM signal for this 5-tone based signal shown in *Figure 5-2* is displayed in *Figure 5-3*. The tones are around .5 kHz, 1.5 kHz, 2.5 kHz, 3.5 kHz and 4.5 kHz. The frequencies are multiples of the repetition rate of the noise generator. However, the tones received some offset in frequency to make their frequencies odd multiples and mutually prime.

Note that *Figure 5-2* shows the envelope of the amplitude of the carrier, with the modulation modeled with six aligned cosines, each with amplitude of 20% of the carrier, or with 10 lines in the AM sidebands.

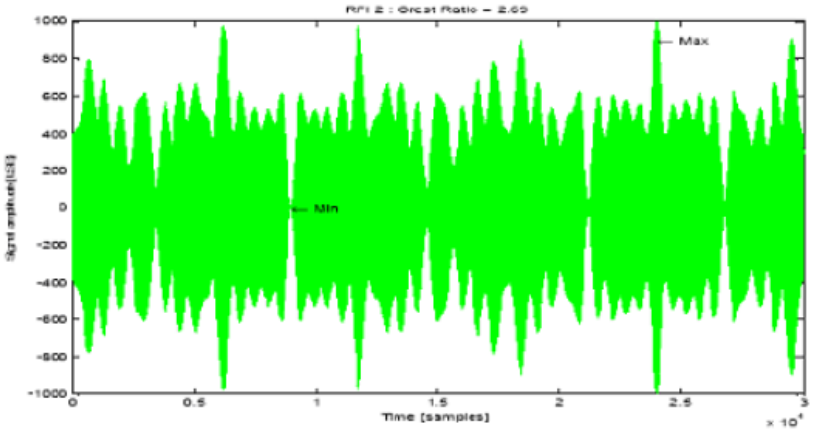


Figure 5-2: Carrier Envelope amplitude recorded on an Oscilloscope

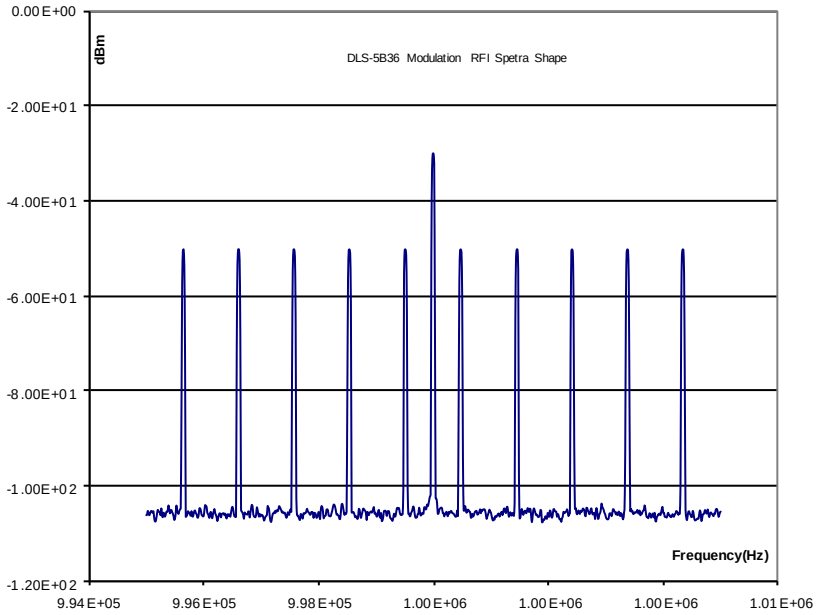


Figure 5-3: Example of Five-band RFI Tone Modulation (Profile A2)

Chapter 6 Time Domain Noise Profiles

6.1 Introduction

The DLS-5800 is capable of reading and playing user-generated time-domain noise files. This chapter provides information on the file format and an example to assist in describing this feature.

6.2 Principles of Operation

The system works by repeatedly playing back a number of data samples at a specific rate. These samples are read out of RAM memory kept inside the DLS-5800. You must first put the data samples into memory and then generate the specified noise type. The data samples are converted into voltages by a Digital to Analog Converter (DAC) inside the DLS-5800 at the rate specified by the sample rate (specified within the header information of the file). This is normally 12.5, 32, or 100 Megasamples per second (Msps). The output changes every time a new sample is converted by the DAC. The DAC is a 14-bit device, which accepts numbers from 0 to 16383. Half scale (8191.5) corresponds to 0 V out of the DAC, and out of the BNC connector at the back of the DLS-5800.

In practice, for noise files with Gaussian amplitude distributions, we recommend a minimum of 2 Megasamples before the output repeats. To create large file types, you will need an ASCII text editor capable of handling large files.

Applications such as Microsoft® Excel® have a limitation of 65536 rows, which does not allow the file to achieve a Gaussian crest factor higher than 5.

If you use pre-recorded files created by Spirent, you will not need to make samples into files. Select the appropriate file from the DLS-5800 Control Software and generate the specified noise type.

- You must create time domain noise files. Save these files with a .txt or .prn extension.
- When you load such a file, the DLS-5800 Control Software creates a new file having the same name but with an extension, _td.dat, which represents the time domain file format which can be read by the DLS-5800. If your original file is large, this step may take a long time.

These are some of the functions you can control:

- Number of samples in the file
- Sample rate
- External synchronization
- Power output
- Relative amplitude for each sample
- Channel selection

You have limited control over:

- Output scaling

You cannot directly control:

- Type of triggering of the sample file. Once started, the triggering is continuous and the file continues until stopped.
- Alternating sample rates

- Alternating between files during playback
- File structure

6.3 Description of the Time Domain Profile File Format

The file structure consists of a header and a main body. The main body of the file must be a number of lines that corresponds to successive data samples. One line corresponds to one data sample. The lines are numeric. Prepare this portion of the file in ASCII format. The *header information* and *relative amplitude* file format requirements is provided below. *Figure 6-3*details the corresponding graph to the ASCII data shown in *Figure 6-1* and *Figure 6-2.*

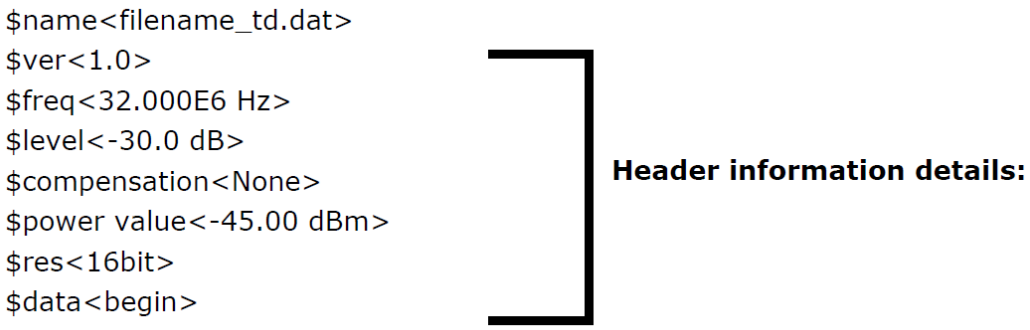


Figure 6-1: Header

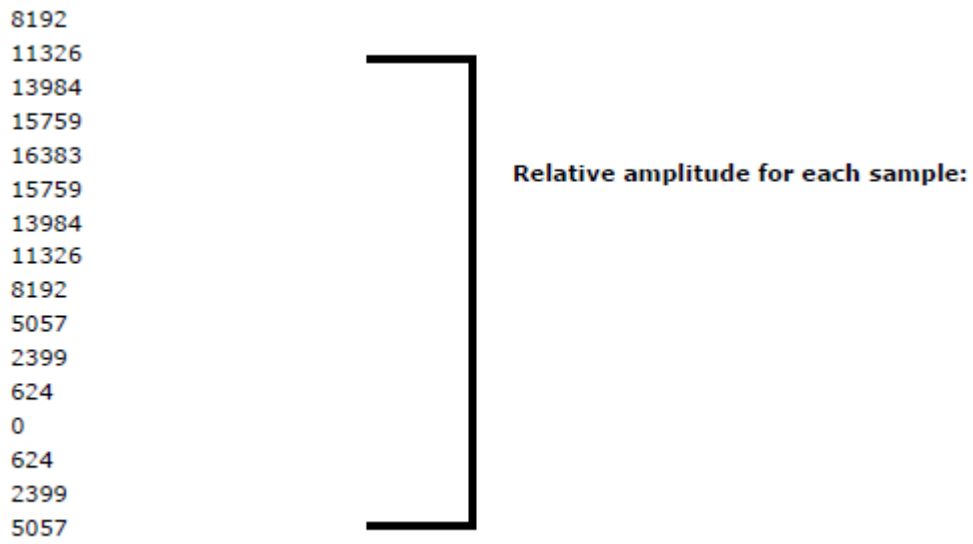


Figure 6-2: Data samples

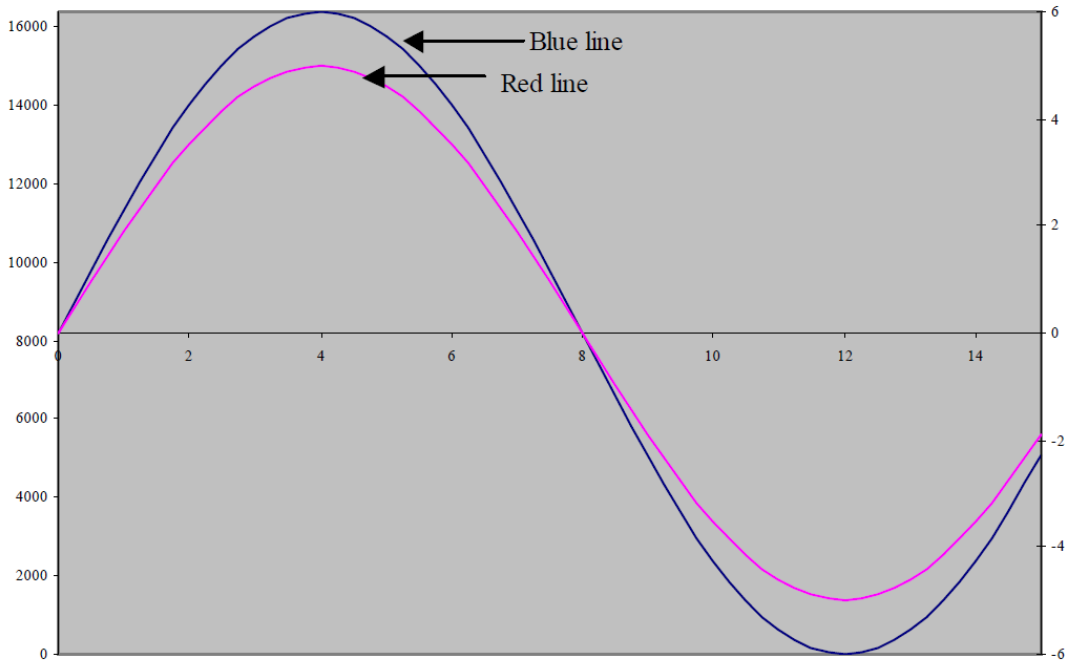


Figure 6-3: Output waveform

6.4 Header Information Details

$name<filename_td.dat> Enter a descriptive name in the *filename* field. The _td.dat extension is required.

$ver<1.0> is the version number of the file created and can be changed.

$freq<32.000E6 Hz> is the sampling rate used by the 5800 to generate the time domain profile. The sampling frequency options are 32.000E6 Hz, 12.500E6 Hz, or 100.000E6 Hz.

$level<-30.0 dB> The RMS value of the samples is required to calculate the attenuator levels set by the DLS-5800 (for large data files this is the same as the standard deviation of the data). This in turn controls output power. Output power is based on a 50 Ω load.

The level needed in the file header is calculated (in dBm) from the formula:

51.28 + Power - 20*log(RMS)

where *Power* is the required power output in dBm and *RMS* is the RMS value of all the numbers in the body of the file.

$compensation<None> presently defined as "None" for future development by Spirent.

$power value<-45.00 dBm> This value does not affect the power of the signal. The value in this field will be displayed by the DLS-5800 Control Software when the file is being generated. You can leave this field blank, or put in a value of meaning for reference purposes.

$res<16bit> presently defined as "16bit" for future development by Spirent.

$data<begin> presently defined as "begin" The main body of the file **must** start on the line after this header information.

6.4.1 Relative Amplitude for Each Sample

Each line of the body of the file must be a decimal number between 0 and 16383. A value of 0 means that the DAC will output -5 V, and a value of 16383 means that the DAC outputs a voltage of +5 V. For telecommunications purposes, the average (mean) value of the data samples should be 8191.5 ±20.

For instance, you could create a simple file with values of a sine wave. The example shown in *Figure 6-1* and *Figure 6-3* is a theoretical example of one that has 16 samples to create the sine wave. A value is required on each row of the time domain file main body.

In *Figure 6-3,* the blue line, plotted on the scale at the left hand side of the graph represents the data needed by the body of the file, and the red scale, plotted on the scale at the right hand side of the graph, is the corresponding output in Volts. Once started, this plays the 16 amplitude levels and repeats to create a sine wave. If no attenuation is set, this gives a sine wave from the BNC channel output of the DLS-5800 with peak values of ±5 Volts as shown on the right-hand scale.

6.4.2 Number of Samples In the File

In practice, for noise files with Gaussian amplitude distributions, we recommend a minimum number of samples of 2 Meg before the output repeats.

As an example, suppose we use a 32-MHz sampling rate and 2^21 (=2097152) data samples. This means that after the file reaches the end of its 2097152 samples, it will start again and repeat the same set of data values. The time taken to cycle once through all the samples is (2^21)/ (32.0E+6) = 0.065536 seconds. This is a rate of 15.26 times per second. The subsequent spectrum cannot contain frequencies lower than 15.26 Hz. It can have frequencies at harmonics of 15.26 Hz all the way up to half the clock rate.

6.4.3 Maximum Memory Size

Each channel can record and play back a maximum of 16 Megasamples

6.4.4 Output Scaling And Injector Loss

Normally, the unit is used with an external injector circuit that performs two main functions:

- to condition the signal so that it is balanced
- to provide a high-impedance or current output from the injector, so that when the injector is attached to the line it does not significantly affect the modem signal already on the line.

Normally, the injector circuit has a signal loss through it .For instructions on how to measure the loss of an injector, see *Appendix A - Measuring Injection Loss*. The injector losses for both DLS-5410DC and DLS-5409 are built in to the DLS-5800 Control Software.The DLS-5800 software automatically sets a gain equal to the injector loss, and the signal is correspondingly higher at the BNC back panel connectors of the DLS-5800.

The graph does not display the entire _td file from start to finish of the file format.

6.4.5 Maximum Level and Overload Conditions

There are several ways in which you can accidentally overload the output. Normally the DLS-5800 warns you if this happens.

- If the .td file contains values outside the range of 0 to 16383, the software rejects the whole file and generates an error message.

You might indirectly overload the channel by specifying inappropriate IL and Level settings. For example, the following data set:

8192
8505
8771
8948
9011
8948
8771
8505
8192
7878
7612
7435
7372
7435
7612
7878

has an RMS value of 579.23. If the Level in the file header is set to 15 dB, and the IL of the injector is 0, the above data set will not overload the DLS-5800. If, however, the IL is set to 19.3 dB, the above data set will overload the DLS-5800. In this case the DLS-5800 will not process the file.

6.4.6 Calculation of RMS

As noted before, the RMS value is the same as the standard deviation of the file. It is calculated according to the following formula:

$$rms := \sqrt{\frac{1}{N}\sum_{n=0}^{N-1}\left(A_n - mean(A)\right)^2}$$

where *A* is the amplitude of the samples.

6.4.7 Converting to dBm from Volts

The output from the DLS-5800 is specified as power output in dBm into a 50 Ω load.

0 dBm corresponds to 1 mW. This means that 0 dBm provides 0.2236 Volts of output across the 50 Ω load. You can convert voltages to dBm for the output of the DLS-5800 by using the following formula:

Output in dBm = 20*log(V/0.2236)

where *V* is the RMS voltage across the 50 Ω load.

6.4.8 File Generation

The most convenient way of generating a large digest of samples is with some type of mathematical software. Output the data to an ASCII text file with a .prn or .txt file extension.

Some software applications generate values in the range of +1 to -1, with 0 as the mean value. This

needs to be converted to a 0 to 16383 range with 8191.5 as the mean value.

6.4.9 Text File Editor

You can use any ASCII text editor that does not add extra characters, and can handle the number of lines in the file. For this reason, we recommend you avoid:

- Microsoft Excel (not long enough)
- Notepad (not long enough)
- Wordpad.exe (slow, output may be incorrect)
- Word processors in general, which add formatting characters.

6.4.10 Total File Length

The body of the data file can be almost any length from a few samples to maximum length. The maximum length of the body of the file is 16 Mbytes (2^24) bytes. The minimum recommended length depends on the nature of the amplitude variations (or probability distribution function, pdf) of the file that you have calculated, but for Gaussian noise should not be less than 2 Mbytes, and preferably 4.

6.4.11 Re-using the _td.dat file

As stated previously, when you load a user-defined time-domain file (in .txt or .prn format), the DLS-5800 Control Software creates a new file having the same name but with the extension _td.dat. If your original file is large, this step may take a long time. You can avoid having to repeat this lengthy process by loading the software-generated _td.dat file on subsequent loads rather than re-loading the original .txt file. Also note that this _td.dat file will be usable on other DLS-5800 platforms.

Chapter 7 Remote Control

7.1 Developing a Remote Controller

Remote control of the DLS-5800 means controlling it from a separate application running on the same, or a remote, machine by using TCP/IP to transport messages that follow the DLS-5800 inter-process communications protocol. Such an application can be thought of as a remote controller.

Remote control of the DLS-5800 and DLS-5410DC is achieved via the DLS-5800 Control Software, which is placed in Remote mode, and in this mode, will accept commands that allow remote control of the DLS-5800 and DLS-5410DC functions. DLS-5409, being a passive device, requires no control.

You may use any programming language to develop a remote controller, as long as the language allows you to "talk" TCP/IP. Indeed, you may develop the remote controller for any operating system and/or host machine. The essential requirement is that you adhere to the messaging protocol defined for the DLS-5800.

Your remote control software must create a client socket, and initiate a TCP/IP connection request to the targeted DLS-5800 unit before they can start exchanging messages.

Once the connection is established, the DLS-5800 will respond to command messages sent via TCP/IP.

NOTE: When connected, the listener socket in the DLS-5800 Control Software ignores any further connection requests until the current connection is ended. If remote control access is disabled, either at the DLS-5800 unit, or by a command from the remote controller, while the connection is established, the listener ignores connection requests following the termination of the connection, but it does not close the connection. Re-enabling of remote control access must always be done at the DLS-5800 unit itself.

NOTE: The DLS-5800 Control Software runs on Windows® XP platforms, which typically employ Intel® x86 processors. These processors use little-endian formatting for data. The remote control software inside the DLS-5800 does not perform host-to-network byte-order translation of the data it sends, so generally the remote controller must accommodate for this, in both sending and receiving, if its host processor uses big-endian data. However, since the message contents, as seen, consist of byte-sized ASCII character sequences (non-zero terminated strings), this is not a concern in the present case; their sequence should remain the same regardless.

With the above in mind, this chapter defines the messaging required to effect remote control operation of the DLS-5800.

7.2 Inter-Process Communications

7.2.1 DLS-5800 Messaging Format

DLS-5800 messages are formatted as follows:

!STX: <message body> ;ETX!

where the message body is described below. !STX: is the message start token, and ;ETX! is the message end token. Messages are not case-sensitive.

Every byte in the DLS-5800 message set is a printable, 8-bit ASCII character.

Messages sent by a remote controller that do not follow the protocol format are dropped by the DLS-5800, and no feedback is provided to the sender. It is up to remote control application developers to ensure that the protocol of messages being sent is correct.

If the protocol of the message is correct, then the DLS-5800 will check the syntax of the message body. If the message body is syntactically correct, the DLS-5800 will process the command, and will return a message indicating whether the operation succeeded or not.

7.2.2 Message Body Specifications

There are 4 types of remote control messages used by the DLS-5800:

SET(parameter ID)

SET(parameter ID):VAL(value)

GET(parameter ID)

TRAP(parameter ID):VAL(value)

The VAL parameter is used to pass a parameter value to and from the DLS-5800.

For example:

!STX:SET(M_ENABLE_OUTPUT):VAL(OUTPUT_1:ON);ETX!
!STX:GET(M_SELECTED_OUTPUT);ETX!

!STX:TRAP(ERROR):VAL(BAD_PARAMETER_ID);ETX!

Refer to Table 8-1 on page 95 for a comprehensive list of messages and their associated parameter identifiers and values.

7.2.3 Functional Details

To operate the DLS-5800 under remote control:

Enable remote control access using the DLS-5800 Control Software. This causes the software to begin listening for a remote connection request on ports ranging from 1024 to 65535. For instructions on enabling remote access on the DLS-5800, see *Remote Control*.

Launch your remote controller application.

Your application should

Connect to the unit using TCP/IP, port range 1024-65535.

Send the desired messages and wait for replies.

When ready to relinquish remote control, close the client socket. Do NOT issue a SET(DISABLE_REMOTE) message – it will prevent subsequent access to the DLS-5800 remote control capabilities for all remote controllers. There is no way to remotely re-enable remote control.

Some commands are channel-specific commands, and are applied to the most recently selected channel. It is therefore a good idea to select the channel as the first step. The messages that follow apply to the selected channel.

Once the channel has been selected, a typical sequence of operation to produce an output is as follows:

Load a noise file.

Set all the necessary parameters.

Generate the sample.

Load the output.

Enable the output, or outputs, as desired.

Switch on the corresponding injector channel

7.2.4 Using Telnet Client

The remote commands can be sent by means of a Telnet client. Both character-mode input and line-mode input are accepted. It is recommended that the Telnet client be configured in line mode.

7.3 Remote Commands

Note that in this section, channel ranges shown are for a DLS-5800-6, which supports 24 channels. The range of channels available to you depends on your hardware configuration. A DLS-5800-2 supports channels 1 to 8 and DLS-5800-4 supports channels 1 to 16.

7.3.1 System and Network Commands

Command: M_SOFTWARE_VERSION

Function: Get software version.

Telnet example:

!STX:GET(M_SOFTWARE_VERSION);ETX!

Command: M_INSTALLED_LICENSE

Function: Get installed license

Returns: A string containing all licensed product names such as DLS5C20_Key, DLS5B13, DLS5B17, and so on.

Telnet example:

!STX:GET(M_INSTALLED_LICENSE);ETX!

Command: M_INSTALLED_PACKAGES

Function: Returns a list of all installed noise packages, with the package version.

Parameter: None

Telnet example:
Sending: !STX:GET(M_INSTALLED_PACKAGES);ETX!

Receiving: !STX:SET(REPORT):VAL(DLS 5B11 v1.0; DLS 5B13 v1.1;);ETX!

If no noise packages are installed, the control software returns the following message:

!STX:TRAP(ERROR):VAL(NO_INSTALLED_PACKAGES);ETX!

Command: M_MAC_ADDRESS

Description: Get MAC address.

Telnet example:

!STX:GET(M_MAC_ADDRESS);ETX!

Command: M_NETWORK_NAME

Description: Get network name. Telnet example:

!STX:GET(M_NETWORK_NAME);ETX!

Command: M_SYSTEM_ID

Description: Retrieves the identification tag that was set in the DLS-5800 Control Software.

Telnet example:

!STX:GET(M_SYSTEM_ID);ETX!

7.3.2 Channel/Output Commands

Command: M_SELECTED_OUTPUT

Description: Queries which output is currently selected, and to which subsequent commands will apply.

Telnet example:

!STX:GET(M_SELECTED_OUTPUT);ETX!

Command: M_FILE_NAMES

Description: Get channel file name(s).

Parameter:A string that may contain more than one name if number of noises combined.

Format: 1. name1; 2. name2;
Telnet example:

!STX:GET(M_FILE_NAMES):VAL(OUTPUT_1);ETX!

Command: M_OUTPUT_LEVEL

Description: Get channel output level

Parameter:A string that contains noise file name and level at 100 Ω calibration impedance. For example:

1. ETSI-A_xtk.dat level -49.9 dBm.

2. DSL_Next_24D_xtk.dat level -43 dBm, at 100 Ω calibration impedance.

Telnet example:

!STX:GET(M_OUTPUT_LEVEL):VAL(OUTPUT_1);ETX!

Command: M_SELECT_OUTPUT

Description: Selects the current channel (output) to operate on. Selects the output to which subsequent commands will apply.

Valid selector strings:

"OUTPUT_1", through "OUTPUT_24" Telnet example:

!STX:SET(M_SELECT_OUTPUT):VAL(OUTPUT_3);ETX!

Command: M_LOAD_FILE

Description: Loads a noise file

Parameter: The absolute path and file name

Telnet examples:

!STX:SET(M_LOAD_FILE):VAL(C:\Program Files\Spirent Communications\DLS 5800\NoiseFiles\Broadband forum\ADSL_TR-048 (Apr 2002)\8-1-2_24_HDSL_Impairment\Downstream_At_ATU-R\HDSL_Next_24DPlusWN_xtk.enc);ETX!

Command: M_GENERATE_SAMPLE

Description: Constructs and combines the noise shape before output. Starts generating the noise sample.

Telnet example:

!STX:SET(M_GENERATE_SAMPLE);ETX!

Command: M_LOAD_OUTPUT

Description: Loads the output with the combined noise file to the current selected channel.

Telnet example:

!STX:SET(M_LOAD_OUTPUT);ETX!

Command: M_NOISE_GAIN

Description: Sets the gain of the last loaded noise. In the case of impulse noise, the gain is based from a starting level of 0 mV.

Parameter:

For Xtalk and RFI: A number in the range of -72.25 to +72.25

For Time domain: A number in the range of -7.0 to +7.0

For Impulse: A number in the range of 0.0 to 100.0
Telnet example:

!STX:SET(M_NOISE_GAIN):VAL(22);ETX!

Command: M_NOISE_GAINEX

Description: Extended gain command. This allows setting the gain of a particular loaded noise by index or by filename. In the case of impulse noise, the gain is based from a starting level of 0 mV.

Parameter:index:level_or_filename:level. Indices start from 0.

For Xtalk and RFI: A number in the range of -72.25 to +72.25

For Time domain: A number in the range of -7.0 to +7.0

For Impulse:A number in the range of 0.0 to 100.0

Telnet example:

!STX:SET(M_NOISE_GAINEX):VAL(0:10);ETX!

!STX:SET(M_NOISE_GAINEX):VAL(White_Noise_xtk.enc:10);ETX!

Command: M_MICRO_GAIN

Description: Adjusts the power level of the currently generated noise.

Parameter:dB value range:

For frequency domain noises: -3.0 to +9.0.

For time domain noise: -3.0 to +7.0.

Telnet example:

!STX:SET(M_MICRO_GAIN):VAL(4.2);ETX!

Command: M_CREST_FACTOR

Description: Enables or disables crest factor checking for xtalk noise. Value string:“ON”, “OFF”

Telnet example:

!STX:SET(M_CREST_FACTOR):VAL(ON);ETX!

Command: M_NUMBER_SAMPLES

Description: Sets the number of samples to use for frequency domain noises.

Valid Values: Powers of 2, ranging from 32768 to 4194304
Telnet example:

!STX:SET(M_NUMBER_SAMPLES):VAL(32768);ETX!

Command: M_TD_WHITENOISE

Description: Enables or disables time domain white noise. Value string:“ON”, “OFF”.

Telnet example:

!STX:SET(M_TD_WHITENOISE):VAL(ON);ETX!

Command: M_IMPULSE_RATE

Description: Sets the number of pulses per second for impulse noise when not used with time domain noise.

Range: 1 - 100 pps (pulses per second)

Telnet example:

!STX:SET(M_IMPULSE_RATE):VAL(30);ETX!

Command: M_CLEARWORKSPACE

Description: Clears any loaded files from the workspace.

Telnet example:

!STX:SET(M_CLEARWORKSPACE);ETX!

Command: M_RESET_CHANNEL

Description: Clears the loaded files in the workspace and stops output on the current selected channel. This command should be used immediately after selecting a channel.

Telnet example:

!STX:SET(M_RESET_CHANNEL);ETX!

Command: M_ENABLE_OUTPUT

Description: Enables or disables the output(s) identified by the value selectors. More than one selector may be specified, and they must be separated by white spaces.

Valid selectors: "OUTPUT_1:ON" … "OUTPUT_24:ON" to "OUTPUT_1:OFF" … "OUTPUT_24:OFF"

Valid status: ON, OFF

Telnet example:

!STX:SET(M_ENABLE_OUTPUT):VAL(OUTPUT_1:OFF OUTPUT_2:ON);ETX!

Command: DISABLE_REMOTE

Description: Disables remote control and terminates the connection.

Telnet example:

!STX:SET(DISABLE_REMOTE);ETX!

Command:M_SAVE_CUSTOM_FILE

Description: Saves a calculated noise sample to disk.

Parameter: The path and file name of the noise. The file name must end with "_cst.enc".

Telnet example:

!STX:SET(M_SAVE_CUSTOM_FILE):VAL(C:\MyCustomFile_cst.enc);ETX!

Command: M_INFO_CUSTOMNOISE

Description: Sets the user defined description for saved files. Use a string with a maximum of 256 characters.

Telnet example:

!STX:SET(M_INFO_CUSTOMNOISE):VAL(This is my custom file);ETX!

Command: M_NOISE_DISTURBER

Description: Set number of noise disturbers for a particular file by index (starting from 0) or by file name.

Telnet example:

!STX:SET(M_NOISE_DISTURBER):VAL(0:10);ETX!

!STX:SET(M_NOISE_DISTURBER):VAL(ETSI-A_xtk.enc:10);ETX!

Command: M_SAVE_DSM_FRAME_FILE

Description: Saves a calculated sample as a DSM frame file.

Parameter: The file name of the noise. The file name must end with "_dsm.enc".

Telnet example:

!STX:SET(M_SAVE_DSM_FRAME_FILE):VAL(frame1_dsm.enc);ETX!

Note that DSM frames must always be stored in the C:\Program Files\Spirent Communications\DLS 5800\DSM Frames subfolder, but the software sets this path automatically, so you don't need to specify it when saving DSM frames.

7.3.3 DLS-5410DC Injector Remote Commands

The commands listed in this section are relayed by the DLS-5800 to the DLS-5410DC noise injector.

7.3.3.1 Channel Control

Command: M_INJ_CHAN<n>_STATE

Function: Set Side A Channel <n> ON / OFF

Parameter: <n>: 1 to 8 for 2 card system

1 to 16 for 4 card system

1 to 24 for 6 card system

Parameter: 0 = OFF, 1 = ON

Telnet example: !STX:SET(M_INJ_CHAN1_STATE):VAL(1);ETX!

If the value is not in range, the return will be: 'VALUE_IS_OUT_RANGE'.

Command: M_INJ_CHAN<n>_MODE

Function: Set Side A Channel 1 coupling mode to Differential or Common

Parameter: <n>: 1 to 8 for 2 card system

1 to 16 for 4 card system

1 to 24 for 6 card system

Parameter: 0 = Differential, 1 = Common

Telnet example: !STX:SET(M_INJ_CHAN1_MODE):VAL(1);ETX!

If the value is not in range, the return will be: 'VALUE_IS_OUT_RANGE'.

7.3.3.2 REIN

Command: M_INJ_CHAN<n>_REIN_DURATION

Function: Set the duration of the REIN noise

Parameter:<n>: valid REIN channel number – 2, 6, 10, 14, 18, 22

Parameter: REIN Noise Duration from 50 – 20000 microseconds in 10 uS steps

Telnet example: !STX:SET(M_INJ_CHAN2_REIN_DURATION) :VAL(50); ETX!

NOTE: the software accepts values in 1 uS increments, but the system will use the nearest multiple of 10 uS.

Command: M_INJ_CHAN<n>_REIN_FREQUENCY

Function: Sets the frequency of the REIN noise

Parameter:<n>: valid REIN channel number – 2, 6, 10, 14, 18, 22

Parameter: REIN Frequency from 1 to 200 Hz

Telnet example: !STX:SET(M_INJ_CHAN2_REIN_FREQUENCY):VAL(1); ETX!

Command: M_INJ_CHAN<n>_REIN_REPETITIONS

Function: Sets the number of repetitions of the REIN noise

Parameter:<n>: valid REIN channel number – 2, 6, 10, 14, 18, 22

Parameter: REIN repetitions from 0 to 65000 (0=free run)

Telnet example: !STX:SET(M_INJ_CHAN2_REIN_REPETITIONS):VAL(1); ETX!

Command: M_INJ_CHAN<n>_REIN_START

Function: Start REIN test

Parameter:<n>: valid REIN channel number – 2, 6, 10, 14, 18, 22

Telnet example: !STX:SET(M_INJ_CHAN2_REIN_START); ETX!

NOTE: REIN can run simultaneously on both sides A and B.

Command: M_INJ_CHAN<n>_REIN_STOP

Function: Stop REIN test

Parameter: <n>: valid REIN channel number – 2, 6, 10, 14, 18, 22

Telnet example: !STX:SET(M_INJ_CHAN2_REIN_STOP); ETX!

If the REIN test is not started, this command will produce the response: "REIN_TEST_NOT_RUNNING"

Command: M_INJ_CHAN<n>_REIN_STATE

Function: Return status of REIN test – ACTIVE or INACTIVE

Parameter: <n>: valid REIN channel number – 2, 6, 10, 14, 18, 22

Telnet example: !STX:GET(M_INJ_CHAN2_REIN_STATE);ETX!"

7.3.3.3 DSM

Command: M_INJ_CHAN<n>_DSM_LOADSEQUENCE

Function: Loads a DSM Sequence file

Parameter:<n>: valid DSM channel pair number – 3_4, 7_8, 11_12, 15_16, 19_20, 23_24

Parameter: The file name to load. DSM Sequences are always stored in the C:\Program Files\Spirent Communications\DLS 5800\DSM Sequences subfolder, and the software sets this path automatically.

Telnet example:
!STX:SET(M_INJ_CHAN7_8_DSM_LOADSEQUENCE):VAL(DSMSequence1.dsq);ETX!

Command: M_INJ_CHAN<n>_DSM_START

Description: Starts the DSM test

Parameter:<n>: valid DSM channel pair number – 3_4, 7_8, 11_12, 15_16, 19_20, 23_24

Telnet example: !STX:SET(M_INJ_CHAN7_8_DSM_START);ETX!

Command: M_INJ_CHAN<n>_DSM_STOP

Description: Stops the DSM test

Parameter:<n>: valid DSM channel pair number – 3_4, 7_8, 11_12, 15_16, 19_20, 23_24
Telnet example: !STX:SET(M_INJ_CHAN7_8_DSM_STOP);ETX!

Command: M_INJ_CHAN<n>_DSM_STATE

Function: Return status of DSM test – ACTIVE or INACTIVE

Parameter:<n>: valid DSM channel pair number – 3_4, 7_8, 11_12, 15_16, 19_20, 23_24

Telnet example: !STX:GET(M_INJ_CHAN3_4_DSM_STATE);ETX!"

7.3.3.4 SHINE

Command: M_INJ_CHAN<n>_SHINE_DURATION

Description: Set the length of the SHINE burst

Parameter:<n>: valid SHINE channel number – 1, 5, 9, 13, 17, 21

Parameter: SHINE burst length from 30 to 65000 mSecs, step 1 mSec

Telnet example: !STX:SET(M_INJ_CHAN1_SHINE_DURATION):VAL(50);ETX!

Command: M_INJ_CHAN<n>_SHINE_START

Description: Starts the SHINE test

Parameter:<n>: valid SHINE channel number – 1, 5, 9, 13, 17, 21

Telnet example: !STX:SET(M_INJ_CHAN1_SHINE_START);ETX!

Command: M_INJ_CHAN<n>_SHINE_STOP

Description: Stops the SHINE test

Parameter:<n>: valid SHINE channel number – 1, 5, 9, 13, 17, 21

Telnet example: !STX:SET(M_INJ_CHAN1_SHINE_STOP);ETX!

Command: M_INJ_CHAN<n>_SHINE_STATE

Function: Return status of SHINE test – ACTIVE or INACTIVE

Parameter:<n>: valid SHINE channel number – 1, 5, 9, 13, 17, 21

Telnet example: !STX:GET(M_INJ_CHAN1_SHINE_STATE);ETX!"

NOTE: SHINE can run simultaneously on both sides A and B.

7.3.3.5 PEIN

Command: M_INJ_CHAN<n>_PEIN_REPETITIONS

Description: Set the number of times to repeat the PEIN pattern

Parameter:<n>: valid PEIN channel number – 1, 5, 9, 13, 17, 21

Parameter: Number of repeats from 1 to 255 or 0 for free run

Telnet example: !STX:SET(M_INJ_CHAN5_PEIN_REPETITIONS):VAL(50);ETX!

NOTE that a single repetition of the PEIN pattern will take around 35 minutes to complete.

Command: M_INJ_CHAN<n>_PEIN_START

Description: Starts the PEIN test

Parameter:<n>: valid PEIN channel number – 1, 5, 9, 13, 17, 21
Telnet example: !STX:SET(M_INJ_CHAN5_PEIN_START);ETX!

Command: M_INJ_CHAN<n>_PEIN_STOP

Description: Stops the PEIN test

Parameter:<n>: valid PEIN channel number – 1, 5, 9, 13, 17, 21
Telnet example: !STX:SET(M_INJ_CHAN5_PEIN_STOP;ETX!

Command: M_INJ_CHAN<n>_ PEIN _STATE

Function: Return status of PEIN test – ACTIVE or INACTIVE

Parameter:<n>: valid PEIN channel number – 1, 5, 9, 13, 17, 21

Telnet example: !STX:GET(M_INJ_CHAN1_PEIN_STATE);ETX!"

NOTE: PEIN can run simultaneously on both sides A and B.

NOTE: PEIN and SHINE cannot run simultaneously on the same channel.

7.3.3.6 Micro Interrupts

Command: M_INJ<n>_INTERRUPT_LENGTH_A

Function: Set Injector <n> Side A Micro Interrupt length

Parameter:<n>: Injector ordinal index (based on serial number) from 1 to 3 depending on number of injectors present.

Parameter: Side A micro interrupt length in mS.

Range: 1 - 60000

Telnet example: !STX:SET(M_INJ1_INTERRUPT_LENGTH_A):VAL(1);ETX!

Command: M_INJ<n>_INTERRUPT_LENGTH_B

Function: Set Side B Micro Interrupt length

Parameter:<n>: Injector ordinal index (based on serial number) from 1 to 3 depending on number

of injectors present.

Parameter: Side B micro interrupt length in mS.

Range: 1 - 60000

Telnet example: !STX: SET(M_INJ1_INTERRUPT_LENGTH_B):VAL(1);ETX!

Command: M_INJ<n>_INTERRUPT_EVERY

Function: Set the Repetition Rate for Micro Interrupts

Parameter:<n>: Injector ordinal index (based on serial number) from 1 to 3 depending on number of injectors present.

Parameter: Repeat rate in seconds

Range: 1 - 60000

Telnet example: !STX: SET(M_INJ1_INTERRUPT_EVERY):VAL(1); ETX!

Command: M_INJ<n>_INTERRUPT_SIDE

Function: Set the side at which Micro Interrupts are generated.

Parameter:<n>: Injector ordinal index (based on serial number) from 1 to 3 depending on number of injectors present.

Parameter: Side index Range: 1 = Side A, 2 = Side B

Telnet example: !STX: SET(M_INJ1_INTERRUPT_SIDE):VAL(1); ETX!

Command: M_INJ<n>_INTERRUPT_TOTAL

Function: Set the total number of Micro Interrupts

Parameter:<n>: Injector ordinal index (based on serial number) from 1 to 3 depending on number of injectors present.

Parameter: Total number of micro interrupts

Telnet example: !STX: SET M_INJ1_INTERRUPT_TOTAL):VAL(10); ETX!

Command: M_INJ<n>_INTERRUPT_TRIGGER_SINGLE

Function: Trigger Single Micro Interrupt

Parameter:<n>: Injector ordinal index (based on serial number) from 1 to 3 depending on number of injectors present.

Parameter: none
Telnet example: !STX: SET(M_INJ1_INTERRUPT_TRIGGER_SINGLE); ETX!

Command: M_INJ<n>_INTERRUPT_TRIGGER_START

Function: Start automatically generated micro interrupt triggering
Parameter:<n>: Injector ordinal index (based on serial number) from 1 to 3 depending on number of injectors present.
Parameter: none
Telnet example: !STX: SET(M_INJ1_INTERRUPT_TRIGGER_START); ETX!

Command: M_INJ<n>_INTERRUPT_TRIGGER_STOP

Function: Stop automatically generated micro interrupt triggering

Parameter:<n>: Injector ordinal index (based on serial number) from 1 to 3 depending on number of injectors present.

Parameter: none

Telnet example: !STX: SET(M_INJ1_INTERRUPT_TRIGGER_STOP); ETX!

Command: M_INJ<n>_MI_STATE

Function: Return status of micro interrupt test – ACTIVE or INACTIVE

Parameter:<n>: Injector ordinal index (based on serial number) from 1 to 3 depending on number of injectors present.

Telnet example: !STX:GET(M_INJ1_MI_STATE);ETX!

7.3.4 Noise Burst Commands

The Noise Burst is a remote-only feature of the DLS-5800 software. This feature allows precision timing control of the noise generator output.

This feature was developed before the introduction of DLS-5410DC, but still is useful for generating signals that fall outside the DLS-5410DC capabilities – for example – extremely long REIN bursts. It is also useful for generating timed bursts using DLS-5409 Passive Injectors. However for most precision applications we recommend using the hardware-based timing capabilities of the DLS-5410DC.

The Noise Burst feature generates single bursts of programmable duration using a trigger command. The Noise Burst Test lets you repeat bursts a specified number of times at a specified interval after an optional programmable delay. Noise bursts are generated by switching on and off the output of the affected channel.

7.3.5 How to Use it

The command sequence for using the Noise Burst feature is shown in this section. The new commands are shown in bold. See “Commands” on page 106 for details on these commands, and see “Example Sequence for Noise Burst Test” on page 107 and “Example Sequence for Single Noise Burst” on page 108.

To use the noise burst feature:

1. Select the output channel (M_SELECT_OUTPUT)
2. Reset the channel (M_RESET_CHANNEL)
3. If using DLS-5410DC, turn on the appropriate injector output. (M_INJ_CHAN<n>_STATE)
4. Load the file (M_LOAD_FILE)
5. Set the gain (M_NOISE_GAIN)
6. Generate the sample (M_GENERATE_SAMPLE)
7. Load the output buffer but do not turn on the output (M_LOAD_OUTPUT_BUFFER) of the noise generator.
8. Specify the burst duration in milliseconds (M_NOISEBURST_DURATION)
   To generate single bursts:
   Trigger the burst (M_NOISEBURST_TRIGGER)

OR

9. To generate repeated bursts:

Specify an optional delay (in seconds) before the first burst (M_NOISEBURST_DELAY)
Specify the number of bursts to generate (M_NOISEBURST_REPEATS)
Specify the interval (in seconds) between burst triggers (M_NOISEBURST_INTERVAL)
Start bursts (M_NOISEBURST_STARTTEST)

Note: If you need to stop the burst test before it is finished, use M_NOISEBURST_STOPTEST.

Command: M_LOAD_OUTPUT_BUFFER

Description: Loads the output buffer of the noise generator without turning on the output. This is used after you generate the sample, and replaces the M_LOAD_OUTPUT command that is used in regular (non-burst-noise) applications.

Telnet example:

!STX:SET(M_LOAD_OUTPUT_BUFFER);ETX!

Command: M_NOISEBURST_DURATION

Description: Sets the duration of the noise burst in milliseconds.

Range: 10 to 10000 ms
Telnet example:

!STX:SET(M_NOISEBURST_DURATION):VAL(50);ETX!

IMPORTANT: Although the command will accept steps of 1 ms, the measured results actually showed increments of 10 ms. The value is rounded to the next largest multiple. In other words, entering 10 ms will give 10 ms duration; entering 11 ms will give 20 ms duration.

Command: M_NOISEBURST_TRIGGER

Description: Generates a single noise burst with the programmed duration.

Telnet example: !STX:SET(M_NOISEBURST_TRIGGER);ETX!

Command: M_NOISEBURST_REPEATS

Description: Sets the number of times to repeat noise bursts when the test is started. Range: 1 to 60000 bursts

Telnet example: !STX:SET(M_NOISEBURST_REPEATS):VAL(5000);ETX!

Command: M_NOISEBURST_INTERVAL

Description: Sets the interval between bursts in seconds.

Range: 1 to 300 seconds in 1 second steps

Telnet example: !STX:SET(M_NOISEBURST_INTERVAL):VAL(1);ETX!

Important: The interval between bursts refers to how often the burst is triggered, and does not account for the duration of the burst. You need to ensure that the interval is greater than the burst duration.

Command: M_NOISEBURST_DELAY

Description: Sets the delay between receiving the M_NOISEBURST_STARTTEST command and the first burst. Subsequent bursts are generated at the interval specified by M_NOISEBURST_INTERVAL.

Range: 1 to 10 seconds in 1 second steps Telnet example:

!STX:SET(M_NOISEBURST_DELAY):VAL(2);ETX!

Command: M_NOISEBURST_STARTTEST

Description: Starts the Noise Burst test. This will cause the programmed number of noise bursts to be generated at the programmed interval, after the programmed delay.

Telnet example: !STX:SET(M_NOISEBURST_STARTTEST);ETX!

Command: M_NOISEBURST_STOPTEST

Description: Stops the Noise Burst test. This is useful for stopping the test if you have accidentally programmed an extremely long time span.

Telnet example: !STX:SET(M_NOISEBURST_STOPTEST);ETX!

Notes:

- The timer used for bursts is not synchronized with the generation of impulses from the noise generator.
- Bursts can be used only on one channel at a time. If attempting to generate bursts on multiple channels, unpredictable results will occur.
- Although the duration command will accept steps of 1 ms, the measured results actually showed increments of 10 ms. The value is rounded to the next largest multiple. In other words, entering 10 ms will give 10 ms duration; entering 11 ms will give 20 ms duration.
- The interval between bursts refers to how often the burst is triggered, and does not account for the duration of the burst. You need to ensure that the interval is greater than the burst duration.

7.3.6 Synchronized Start

The Synchronized Start feature allows you to start multiple channels on a single noise card with minimal delay between channels.

The feature is available from the remote interface only – it is not available from the GUI. The feature uses a new command, M_SYNCSTART. This command receives a single parameter which is a bit mapped value which specifies which channels should be switched on.

Measured results using a 3.8 GHz P4-based system were on the order of 80 – 160 nS per channel, with most results around 160 nS per channel. But there were a couple of aberrations which showed results around 80 – 84 nS.

In other words 2 adjacent channels (i.e. channels 1 and 2) started at the same time show a 160 nS delay between the 2 channels. If the channels are not adjacent, the results show 160 nS for each channel between. So Channels 1 and 3 (1 channel between) would be 320 nS, and Channels 1 and 4 (2 channels between) show 480 nS.

7.3.6.1 Using Synchronized Start

To use Synchronized Start you need to first load the output buffers for all channels you wish to use. Use the M_LOAD_OUTPUT_BUFFER for this.

Command: M_ SYNCSTART

Description: Start 1 to 4 channels in rapid sequence

Valid Values: 24-Bit bit mapped value representing channels from 24 to 1 from left to right

i.e.

```
bits  23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
chan  24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
```

Telnet examples:

!STX:SET(M_ SYNCSTART):VAL(15);ETX!

This example will turn on all 4 channels on card 1.

!STX:SET(M_ SYNCSTART):VAL(3);ETX!

This example will turn on channels 1 and 2 on card 1.

!STX:SET(M_ SYNCSTART):VAL(240);ETX!

This example will turn on all channels on card 2.

Note that all affected channels must be on the same noise card. If you attempt to start channels on different noise cards, a VALUE_IS_OUT_OF_RANGE error is returned.

Synchronized Start Sequence

M_SELECT_OUTPUT

M_RESET_CHANNEL

M_LOAD_FILE

M_NOISE_GAIN

M_GENERATE_SAMPLE

M_LOAD_OUTPUT_BUFFER

M_SYNCSTART

7.4 Sample Remote Control Sequences

Here are some samples showing how to remotely control the DLS-5800 and DLS-5410DC. If you are using DLS-5409, you will need to comment out or remove the appropriate lines as shown in the listings.

The sequences are written in TCL. Of course you can use any programming language which supports TCP/IP communications. Even if you are not using TCL you can at least see the correct sequence of writes and reads for various operations. Simply replace the write_socket and read_socket calls with calls to whatever your functions are for writing to and reading the socket.

Note that read_socket is called after every call to write_socket. Although we have not shown it here, a well-written script should check the data received by read_socket and ensure that it is not a TRAP message indicating an error. Rather, if the message received by the DLS-5800 was processed successfully, most commands will cause the DLS-5800 to respond with the string:

!STX:SET(REPORT):VAL(CMD_SUCCEEDED);ETX!

At the end of this section, you will find various TCL fragments which may be useful. These fragments include the functions which are called from the scripts.

7.4.1 Simple Load and Output sequence

if not using DLS-5410DC, comment the next 2 lines which turn on Channel 1 of the DLS-5410DC

write_socket "!STX:SET(M_INJ_CHAN1_STATE):VAL(1);ETX!”

read_socket

if using DLS-5410DC, change (0) to (1) if you want common mode

if not using DLS-5410DC, comment the next 2 lines which set Channel 1 into differential mode

write_socket "!STX:SET(M_INJ_CHAN1_MODE):VAL(0);ETX!”

read_socket

```
# select channel 1
write_socket "!STX:SET(M_SELECT_OUTPUT):VAL(OUTPUT_1);ETX!"
read_socket

# reset the channel
write_socket "!STX:SET(M_RESET_CHANNEL);ETX!"
read_socket

# load a file. Remember to replace <filename1> with the desired full path, name, and extension
# do not add quotes around the file name.
write_socket "!STX:SET(M_LOAD_FILE):VAL(<filename1>);ETX!"
read_socket

# load a file. Remember to replace <filename2> with the desired full path, name, and extension
# do not add quotes around the file name.
write_socket "!STX:SET(M_LOAD_FILE):VAL(<filename2>);ETX!"
read_socket

# set the number of samples
write_socket "!STX:SET(M_NUMBER_SAMPLES):VAL(131072);ETX!"
read_socket

# set the noise gain
write_socket "!STX:SET(M_NOISE_GAIN):VAL(0.0);ETX!"
read_socket

# set the reference level to  +10 dBm for both files
write_socket “!STX:SET(M_NOISE_GAINEX):VAL(0:10);ETX!“
read_socket
write_socket ”!STX:SET(M_NOISE_GAINEX):VAL(1:10);ETX!”
read_socket
```

```
# enable crest factor  >  5 checking
!STX:SET(M_CREST_FACTOR):VAL(ON);ETX!

# generate the sample from the loaded files
write_socket "!STX:SET(M_GENERATE_SAMPLE);ETX!"
read_socket

# load the output and start signal generation
write_socket "!STX:SET(M_LOAD_OUTPUT);ETX!"
read_socket
```

7.4.2 Noise Burst single-burst script

Note: Replace < filename> and <noise level> with appropriate values. Also, the script assumes channel 1.

```
# if not using DLS-5410DC, comment the next 2 lines which turn on Channel 1 of the
DLS-5410DC
write_socket "!STX:SET(M_INJ_CHAN1_STATE):VAL(1);ETX!”
read_socket

# if using DLS-5410DC, change (0) to (1) if you want common mode
# if not using DLS-5410DC, comment the next 2 lines which set Channel 1 into differential mode
write_socket "!STX:SET(M_INJ_CHAN1_MODE):VAL(0);ETX!”
read_socket

# select channel 1
write_socket “!STX:SET(M_SELECT_OUTPUT):VAL(OUTPUT_1);ETX! “
read_socket

# reset the current channel
write_socket “!STX:SET(M_RESET_CHANNEL);ETX!”
read_socket
```

```
# load a noise file
write_socket “!STX:SET(M_LOAD_FILE):VAL(<filename>);ETX! “
read_socket

# set a gain value
write_socket “!STX:SET(M_NOISE_GAIN):VAL(<noise level>);ETX! “
read_socket
# generate the sample
write_socket “!STX:SET(M_GENERATE_SAMPLE);ETX!“
read_socket

# load to the AWG output buffer, but do not start generating signal
write_socket “!STX:SET(M_LOAD_OUTPUT_BUFFER);ETX!”
read_socket

# set the noise burst duration in milliseconds
write_socket “!STX:SET(M_NOISEBURST_DURATION):VAL(50);ETX!”
read_socket

# trigger a single noise burst
write_socket “!STX:SET(M_NOISEBURST_TRIGGER);ETX!”
read_socket
```

7.4.3 Noise Burst multiple-burst script

Note: Replace < filename> and <noise level> with appropriate values.

```
# if not using DLS-5410DC, comment the next 2 lines which turn on Channel 1 of the
DLS-5410DC
write_socket "!STX:SET(M_INJ_CHAN1_STATE):VAL(1);ETX!”
read_socket
```

```
# if using DLS-5410DC, change (0) to (1) if you want common mode
# if not using DLS-5410DC, comment the next 2 lines which set Channel 1 into differential mode
write_socket "!STX:SET(M_INJ_CHAN1_MODE):VAL(0);ETX!”
read_socket

# select channel 1
write_socket "!STX:SET(M_SELECT_OUTPUT):VAL(OUTPUT_1);ETX!”
read_socket

# reset the current channel
write_socket "!STX:SET(M_RESET_CHANNEL);ETX!
read_socket

# load a noise file
write_socket "!STX:SET(M_LOAD_FILE):VAL(<filename>);ETX!
read_socket

# set a gain value
write_socket "!STX:SET(M_NOISE_GAIN):VAL(<noise level>);ETX!
read_socket

# generate the sample
write_socket "!STX:SET(M_GENERATE_SAMPLE);ETX!
read_socket

# load to the AWG output buffer, but do not start generating signal
write_socket "!STX:SET(M_LOAD_OUTPUT_BUFFER);ETX!
read_socket

# set the noise burst duration in milliseconds
write_socket "!STX:SET(M_NOISEBURST_DURATION):VAL(50);ETX!
read_socket
```

```
# set the number of times to repeat the noise burst
write_socket "!STX:SET(M_NOISEBURST_REPEATS):VAL(50);ETX!
read_socket

# set how often to repeat the burst in seconds
write_socket "!STX:SET(M_NOISEBURST_INTERVAL):VAL(2);ETX!
read_socket

# set the delay between receiving the start command, and the actual start of burst generation
write_socket "!STX:SET(M_NOISEBURST_DELAY):VAL(2);ETX!”
read_socket

# start generation of noise bursts
write_socket "!STX:SET(M_NOISEBURST_STARTTEST);ETX!
read_socket
```

7.4.4 Synchronized Start Sample Script Fragment

Modify this to use your IP address, appropriate noise files and channels.

```
#########################################################
# MAIN Scipt
#########################################################
#Modify this to your IP address
set ip 10.20.2.44

#local
#set ip 127.0.0.1
#########################################################
set port 2001

set noiseDir "C:/Program Files/Spirent Communications/DLS5800/noiseFiles"
set noiseFile "$noiseDir/ADSL-C1_imp.dat"
```

```
set chan "OUTPUT_"

# fire up the box
Open $ip $port

# set the current card to use (numbered from 1 to n where n is 2 to 6 depending on your hardware)
set cardNum 4

# calculate which channels are on that card
set loIdx [ expr $cardNum + ($cardNum-1)*3   ]
set hiIdx [ expr $loIdx+3 ]

# calculate the channel mask so that 4 channels on the card are started
# card 1: 15
# card 2: 240
# card 3: 3840
# card 4: 61440
# card 5: 983040
# card 6: 15728640

set chanMask [ expr (15 << ($cardNum-1)*4 ) ]

# load buffers 4 channels on the selected noise card
for {set idx $loIdx} {$idx <= $hiIdx} {incr idx} {

  write_socket "!STX:SET(M_SELECT_OUTPUT):VAL($chan$idx);ETX!"
  read_socket
  write_socket "!STX:SET(M_RESET_CHANNEL);ETX!"
  read_socket
  write_socket "!STX:SET(M_LOAD_FILE):VAL($noiseFile);ETX!"
  read_socket
  write_socket "!STX:SET(M_NOISE_GAIN):VAL(100);ETX!"
```

```
 read_socket
 write_socket "!STX:SET(M_GENERATE_SAMPLE);ETX!"
 read_socket
 write_socket "!STX:SET(M_LOAD_OUTPUT_BUFFER);ETX!"
 read_socket
}

# start the channels
write_socket "!STX:SET(M_SYNCSTART):VAL($chanMask);ETX!"
read_socket

# pause execution
tk_messageBox -message "Click OK to continue" -icon info -type ok

# stop the output on all 4 channels
for {set idx $loIdx} {$idx < $hiIdx} {incr idx}  {
 #Section for repeating the impulses
 write_socket "!STX:SET(M_SELECT_OUTPUT):VAL($chan$idx);ETX!"
 read_socket
 write_socket "!STX:SET(M_RESET_CHANNEL);ETX!"
 read_socket
}
Close
```

7.4.5 DSM Sample

This sample illustrates starting DSM on all available channels on a 4-card DLS-5800 and waiting until all tests are done.

```
#########################################################
# MAIN Scipt
#########################################################

# DLS-5800
 set ip 10.20.2.44
 set port 2001
```

```
Open $ip $port

#this channel list is for a 4-card system
set channellist [list 3_4 7_8 11_12 15_16]

set m_inj M_INJ_

set _loadsequence _DSM_LOADSEQUENCE

set _start _DSM_START
set _stop _DSM_STOP

set _getparamstate _DSM_STATE


foreach chn $channellist {

   set chan "CHAN"; append chan $chn
   puts "Testing DSM synchronization GET command on $chan"

#change short.dsq to whatever dsm sequence you wish to use

   write_socket "!STX:SET($m_inj$chan$_loadsequence):VAL(short.dsq);ETX!"
   read_socket
   write_socket "!STX:SET($m_inj$chan$_start);ETX!"
   read_socket
}

# wait until all DSM sequences are done

set bActive 1
while {$bActive==1} {
   set ret [IsFunctionActive $channellist $_getparamstate]
   if {$ret == ""} {
 set bActive 0
   }
}

Close
```

7.4.1 REIN Sample

This code fragment illustrates configuring and starting REIN on all available channels on a 6-card DLS-5800, then waiting until the system indicates that all the REIN tests are finished.

```
#########################################################
```

```
# MAIN Scipt
#########################################################

# DLS-5800
set ip 10.20.2.44

#local
#set ip 127.0.0.1
#########################################################

set port 2001

# fire up the box
Open $ip $port

set cmd ":SET(M_INJ_CHAN"
set state "_STATE"
set mode "_MODE"
set cmdclose ")"

#channel list for a 6-card system
set channellist [list 2 6 10 14 18 22]

set m_inj M_INJ_

set _duration _REIN_DURATION
set _frequency _REIN_FREQUENCY
set _repetitions _REIN_REPETITIONS

set _start _REIN_START
set _stop _REIN_STOP

set _getparamstate _REIN_STATE


foreach chn $channellist {

   set chan "CHAN"; append chan $chn
   puts "Testing REIN synchronization GET command on $chan"

   write_socket "!STX:SET($m_inj$chan$_duration):VAL(20000);ETX!"
   read_socket
   write_socket "!STX:SET($m_inj$chan$_frequency):VAL(1);ETX!"
   read_socket
   write_socket "!STX:SET($m_inj$chan$_repetitions):VAL(100);ETX!"
   read_socket
```

```
   write_socket "!STX:SET($m_inj$chan$_start);ETX!"
   read_socket


}

# loop until all inactive
set bActive 1
while {$bActive==1} {
   set ret [IsFunctionActive $channellist $_getparamstate]
   if {$ret == ""} {
 set bActive 0
   }
}

Close
```

7.4.2 PEIN Sample

```
#########################################################
# MAIN Scipt
#########################################################
# DLS-5800
set ip 10.20.2.44

#local
#set ip 127.0.0.1
#########################################################


set port 2001

# fire up the box
Open $ip $port

set cmd ":SET(M_INJ_CHAN"
set state "_STATE"
set mode "_MODE"
set cmdclose ")"


set channellist [list 1 5 9 13 17 21]

set m_inj M_INJ_
```

```
set _duration _REIN_DURATION
set _frequency _REIN_FREQUENCY
set _repetitions _PEIN_REPETITIONS

set _start _PEIN_START
set _stop _PEIN_STOP

set _getparamstate _PEIN_STATE

foreach chn $channellist {

   set chan "CHAN"; append chan $chn
   puts "Testing PEIN synchronization GET command on $chan"


   write_socket "!STX:SET($m_inj$chan$_repetitions):VAL(1);ETX!"
   read_socket

   write_socket "!STX:SET($m_inj$chan$_start);ETX!"
   read_socket

}

# loop until all inactive
set bActive 1
while {$bActive==1} {
   set ret [IsFunctionActive $channellist $_getparamstate]
   if {$ret == ""} {
 set bActive 0
   }
}

Close
```

7.4.3 SHINE Sample

```
#########################################################
# MAIN Scipt
#########################################################

set ip 10.20.2.44

#local
#set ip 127.0.0.1
#########################################################
```

```
set port 2001

# fire up the box
Open $ip $port

set cmd ":SET(M_INJ_CHAN"
set state "_STATE"
set mode "_MODE"
set cmdclose ")"

set channellist [list 1 5 9 13 17 21]


set m_inj M_INJ_

set _duration _SHINE_DURATION
set _frequency _REIN_FREQUENCY
set _repetitions _REIN_REPETITIONS

set _start _SHINE_START
set _stop _SHINE_STOP

set _getparamstate _SHINE_STATE


foreach chn $channellist {

   set chan "CHAN"; append chan $chn
   puts "Testing SHINE synchronization GET command on $chan"

   write_socket "!STX:SET($m_inj$chan$_duration):VAL(25000);ETX!"
   read_socket

   write_socket "!STX:SET($m_inj$chan$_start);ETX!"
   read_socket

}

# loop until all inactive
set bActive 1
while {$bActive==1} {
   set ret [IsFunctionActive $channellist $_getparamstate]
   if {$ret == ""} {
 set bActive 0
   }
```

```
}

Close
```

7.4.4 Micro Interrupt Sample

```
#########################################################
# MAIN Scipt
#########################################################
set ip 10.20.2.44

#local
#set ip 127.0.0.1
#########################################################

set port 2001

# fire up the box
Open $ip $port

set cmd ":SET(M_INJ_CHAN"
set state "_STATE"
set mode "_MODE"
set cmdclose ")"


set channellist [list 1 2 3]

set m_inj M_INJ

set _intlengthA _INTERRUPT_LENGTH_A
set _intlengthB _INTERRUPT_LENGTH_B
set _intside _INTERRUPT_SIDE
set _inttrig _TRIGGER_SINGLE
set _intevery _INTERRUPT_EVERY
set _inttotal _INTERRUPT_TOTAL

set _start _INTERRUPT_TRIGGER_START
set _stop _INTERRUPT_TRIGGER_STOP

set _getparamstate _MI_STATE


foreach chn $channellist {
```

```
    puts "Testing Microinterrupt commands on injector $chn"

    write_socket "!STX:SET($m_inj$chn$_intlengthA):VAL(100);ETX!"
    read_socket
    write_socket "!STX:SET($m_inj$chn$_intlengthB):VAL(200);ETX!"
    read_socket
    write_socket "!STX:SET($m_inj$chn$_intevery):VAL(1);ETX!"
    read_socket

    #use 1 for A and 2 for B in the next line
    write_socket "!STX:SET($m_inj$chn$_intside):VAL(1);ETX!"
    read_socket

    write_socket "!STX:SET($m_inj$chn$_inttotal):VAL(60);ETX!"
    read_socket

    write_socket "!STX:SET($m_inj$chn$_start);ETX!"
    read_socket

}

# loop until all inactive
set bActive 1
while {$bActive==1} {
    set ret [IsFunctionActive $channellist $_getparamstate]
    if {$ret == ""} {
 set bActive 0
    }
}

Close
```

7.5 TCL Code Fragments

These sections include functions for opening, writing, and reading sockets. Note that these are not intended as complete code elements, and as such, do not implement proper error checking. But they may be useful in order to get started.

7.5.1 Required Global Variables for TCL samples

TCL code:

```
## ]-- Start Global Variables
```

```
# Socket descriptor
set _sock_desc ""

# Server Response (Must be global for vwait)
set line ""

# time out
set _timeout 60000
# for debugging
set _verbose on

## ]-- End Global Variables
```

7.5.2 Communications Functions

7.5.2.1 write_socket

TCL code:

```
# Procedure : write_socket
# Purpose   : Sends 'command' to the 5204
# Exits if a connection is not established

proc write_socket { command } {
  global _sock_desc _verbose
  set snd "Sending: "

  if {[eof $_sock_desc]} {
    puts "Error: A connection has not been established!!"
  }
  if { $_verbose == "on" } {
   puts "$snd $command"
  }
```

```
  puts $_sock_desc $command
}
```

7.5.2.2 read_socket

TCL code:

```
# Procedure : read_socket
# Purpose   : Reads response from 5800
# Sets 'line' to reponse if successful, otherwise
# sets 'line' to "Timeout" if a timeout occured while
# waiting for a response.
# Returns a parsed 5800 response or 'timed out' on failure.

proc read_socket { } {

 #global sock descriptor and line
 global _sock_desc line _verbose _timeout

 # set the timer
 set time_id [after $_timeout {set line -1}]

 # when socket becomes readable
 set line ""
 fileevent $_sock_desc readable {
  # Disable fileevent until set again
  fileevent $_sock_desc readable ""
  # poll until read line or timed out

  while { $line == "" } {
   set line [read $_sock_desc]
   update
  }
```

```
  }

  # wait for variable line to change
  vwait ::line

  if { $line == -1 } {
     # Timeout Occured
     set line "Timeout"
     set result "Connection timed out."
  } else {
     # received message. disable timer
     after cancel $time_id
     if { $_verbose == "on" } {
       puts "Received: $line"
     }
}

return $line

}
```

7.5.2.3 Open

TCL code:

```
proc Open { ip port } {
  global _sock_desc
  # Check for valid args
  if { $ip != "" && $port != "" } {

    # check if already connected
```

```
  if { [IsConnected] } {
    perror "Dls_5800_Connect" "Already Connected"
  }

  set _sock_desc [socket $ip $port]
  if {[eof $_sock_desc]} {
    perror "Dls_5800_Connect:" "Cannot establish connection to $ip on port $port."
  } else {
    fconfigure $_sock_desc -blocking 0 -buffering line -translation auto
    puts "Dls_5800_Connect: Successful."
    return $_sock_desc
  }
 }
 perror "Dls_5800_Connect" "Invalid number of parameters."
}
```

7.5.2.4 IsConnected

TCL code:

```
proc IsConnected { } {
 global _sock_desc

 if { $_sock_desc == "" } {
  return 0
 }
 return 1

}
```

7.5.2.5 perror

TCL code:

```
proc perror { command errormsg } {

 # Display Error messages
 error "$command: $errormsg"
}
```

7.5.2.6 IsFunctionActive

This function returns a list of channels on which the passed function is active.

Passed parameters:

Function: _DSM_STATE, _REIN_STATE, _PEIN_STATE, _MI_STATE, _SHINE_STATE
ChanList: List of channels to test – i.e. set ChanList [list 3_4 7_8 11_12 15_16] for DSM

```
proc IsFunctionActive { ChanList Function } {

   # return a list of channels which are ACTIVE regarding the passed function.

   set sValIDToken "VAL("
   set sCloseParenthesesToken ")"

   set m_inj M_INJ

   set sReturnList ""

   foreach chn $ChanList {

        set chan "_CHAN"; append chan $chn

        set retval "ACTIVE"

        set _getstat $m_inj$chan$Function

        write_socket "!STX:GET($_getstat);ETX!"
        set retval [read_socket]
        set iValpos [string first $sValIDToken $retval]
        set iValpos [expr {$iValpos+[string length $sValIDToken]}]
```

```
            set iEndPos [string first $sCloseParenthesesToken $retval $iValpos]
            set retval [string range $retval $iValpos $iEndPos-1]

            if {$retval != "INACTIVE"} {
               # add this channel to the 'active' list
               append sReturnList $chn
            }
      }

      return $sReturnList
}
```

7.5.2.7 Close

TCL code:

```
proc Close { } {
  global _sock_desc
  close $_sock_desc
    set _sock_desc ""
}
```

7.5.3 Putting it all Together: Beginning of the script:

First, we need to open the socket and create a connection. In your main TCL script , do the following:

TCL Code:

```
# change the next line to the IP address of your DLS-5800
set ip aa.bb.cc.dd

# change the next line to the correct port you have set in the DLS-5800. defaults are 2001, 2002, 2003
# for the master and multi-session mode slave sessions.

set port 2001
Open $ip $port
```

7.5.4 Putting it all Together: Ending the script:

At the end of the TCL script, your program should close the socket.

TCL Code:

Close

7.6 Report Strings

Report: CMD_SUCCEEDED

Description: The message has been accepted and processing was successful.

Report: CHANNEL_x_SELECTED

Description: The selected output is channel x where x is 1 - 24.

Report: LEVEL_TOO_HIGH

Description: Power level of combined crosstalk exceeds maximum output power level.

Report: PACKAGE_MANAGER_ERROR

Description: Package manager error: “Load file” not executed.

Report: CREST_FACTOR_UNATTAINED

Description: The resulting crest factor is less than 5 even after attempting 5 recalculations.

Report: RFI_INVALID_LICENSE

Description: Invalid license for RFI.

7.7 Error Strings

Error: BAD_PARAMETER_ID

Description: Invalid task or parameter identifier in message.

Error: BUSY

Description: Busy operating on a previous command.

Error: CMD_OVERFLOW

Description: A command has been sent after DLS-5800 remote control access has been disabled.

The first command sent after disabling is accepted and queued. If a second command is then sent, an overflow will occur. The first command is not processed until remote control is re-enabled.

OR

The DLS-5800 received a command before it had finished processing a previous command.

Error: PROCESS_FAILURE

Description: Unspecified error.

Error: FILE_ACCESS_ERROR

Description: File does not exist.

Error: UNABLE_TO_APPLY_XTALK_NOT_SELECTED

Description: The command received applies to XTALK noise files, but the current profile is not XTALK.

Error: VALUE_IS_OUT_OF_RANGE

Description: Value is out of valid range.

Error: CHANNEL_NOT_SELECTED

Description: Output channel was not be selected.

Error: SELECT_CHANNEL_NOT_EXIST

Description: A selection attempt was made on a channel does not exist.

Error: LOAD_OUTPUT_ERROR

Description: Failure to load the hardware output channel.

Error: INVALID_PACKAGE_LICENSE

Description: Invalid License.

Error: UNABLE_TO_APPLY_IMPULSE_NOT_SELECTED

Description: The command received applies to IMPULSE noise, but the current profile is not IMPULSE.

Error: FAILURE_BOTH_XTALK_AND_RFI_NOT_SELECTED

Description: A command was received which applies to RFI or XTALK, but neither is active.

Error: FAILURE_OUTPUT_SIGNAL_TOO_HIGH

Description: The output signal in dB is higher than maximum allowed value.

Error: FAILURE_TD_NOISE_NOT_COMPATIBLE_WITH_IMPULSE

Description: Attempt to combine Time Domain with Impulse, but the sample frequency is different.

Error: HARDWARE_FAILURE

Description: Disable output channel failed.

Error: FAILURE_BASE_OF_POWER_IS_NOT_TWO

Description: An invalid value used for a power-of-two parameter. Valid values are powers of 2 ranging between 32768 and 4194304.

Error: MICROGAIN_IS_NOT_ADJUSTABLE

Description: A noise profile must be loaded to the hardware before the microgain can be adjusted.

Error: UNABLE_TO_APPLY_IMPULSE_RATE

Description: A time domain noise type has been selected for which rate pps values cannot be set.

Error: UNABLE_TO_APPLY_IMPULSE_NOT_SELECTED

Description: A noise type has been selected that is not compatible with impulse noise.

Error: VALUE_IS_OUT_RANGE

Description: Impulse parameter was not between 1 and 100 pps (rate).

Error: FAILURE_SAVE_CUSTOM_NOISE

Description: Custom noise file save sequence is incorrect.

Error: CUSTOMNOISE_NOT_SELECTED

Description: Custom noise not enabled or selected.

Error: FAILURE_LOAD_CUSTOM_NOISE

Description: File path or file name is not recognized. File may not end with "_CST.ENC".

Error: FAILURE_OTHER_NOISE_SELECTED

Description: Other noise types are currently in use or selected.

Error: IMPULSE_TYPE_WRONG

Description: Incorrect impulse noise type was selected.

Error: FAILURE_CANNOT_SET_WHITENOISE

Description: TD white noise cannot be set because sampling frequency is incompatible.

Error: FAILURE_FILE_LIMIT_EXCEEDED

Description: The maximum number of one or more file types has been exceeded.
The maximum number of files according to type is as follows:
Xtalk:6
RFI: 1
Impulse:1
Time Domain:1
Custom:1

Error: UNABLE_TO_COMBINE_WITH_OTHER_NOISE_TYPE

Description: Attempt to load a file with a different type of noise. For example, xtalk with impulse.

Error: FAILURE_FILE_ALREADY_LOADED

Description: This file has already been loaded.

Error: FAILURE_COMPLEX_LOAD_NOT_COMPATIBLE

Description: This file does not have the same complex load impedance compensation as the previously loaded file.

Error: FAILURE_FILE_NOT_LOADED

Description: Attempt to perform an operation requiring a loaded file (such as setting the noise

gain) when no file has been loaded.

Error: FAILURE_INVALID_INDEX

Description: The index position is invalid relative to the extended noise gain command.

Error: FAILURE_GAIN_NOT_ADJUSTABLE

Description: Attempt to adjust gain for a file with a fixed gain, for example a custom noise profile.

Error: UNABLE_TO_APPLY_DISTURBER_NOT_AVAILABLE

Description: Disturbers cannot be set for the selected noise file.

Error: CHANNEL_NOT_ACCESSIBLE

Description:

a) Attempt to invoke a command which would access a channel in another remote instance.

b) Attempt to invoke a command which would access a channel which is not available due to hardware configuration. For example, attempting to select channel 17 on a system containing 2 noise cards would produce this error, as the maximum allowed channel number would be 16.

Error: UNABLE_TO_COMPLY_IN_DOUBLE_SESSION_MODE

Description: The command affects both remote instances and therefore cannot be completed.

For example, a remote instance requests a return to local mode.

Error: FAILURE_LOAD_NOISE_COMMAND_FILE

Description: A command in the noise command file (NCD file) contains an error that prevents it from successfully loading.

Error: FAILURE_NOT_CUSTOMER_SAVED_FILE

Description: The file is not a customer saved file (ending with "_cst.enc").

Error: FAILURE_SAVE_DSM_FRAME

Description: An error occurred when attempting to save a DSM frame file.

Error: UNSUPPORTED_ON_SPECIFIED_CHANNEL

Description: The selected function (REIN / PEIN / SHINE / DSM) is not allowed on the specified DLS-5410 injector channel.

Error:INJECTOR_NOT_ACCESSIBLE_FROM_THIS_SESSION

Description: The injector that the remote script is trying to access is owned by another session.

7.8 For DLS-5200 / DLS-5204 / DLS-5500 users

10. For the most part, scripts that will work on older Spirent Communications noise generators will work on DLS-5800. However, as necessitated by hardware architecture changes, there are a few commands that will have to be removed from your older scripts. You will also need to add commands to control the DLS-5410DC injector, for example, to select the channel coupling mode and to turn on the output.

The following commands are not supported by DLS-5800:

Obsolete Command	Replacement action
M_SAMPLE_FREQ	None. Sample rate is defined in file header.
M_INJ_CONNECT	None. Connection is done automatically on software startup.
M_INJ_DISCONNECT	None. Disconnection is done automatically on software shutdown.
M_TD_NOISE_GAIN	Use M_NOISE_GAIN, M_NOISE_GAINEX, or M_MICRO_GAIN. See command set documentation to help determine which to use.
M_RFI_MUTE	None
M_XTALK_MUTE	None
RFI_MUTE	None
XTALK_MUTE	None
M_IMPULSE_MUTE	None
M_TD_NOISE_MUTE	None
M_LOAD_XTALK_FILE	M_LOAD_FILE
LOAD_RFI_FILE	M_LOAD_FILE
LOAD_XTALK_FILE	M_LOAD_FILE
M_LOAD_TD_FILE	M_LOAD_FILE
M_LOAD_CUSTOM_FILE	M_LOAD_FILE
RFI_GAIN	Use M_NOISE_GAIN, M_NOISE_GAINEX, or

	M_MICRO_GAIN. See command set documentation to help determine which to use.
XTALK_GAIN	Use M_NOISE_GAIN, M_NOISE_GAINEX, or M_MICRO_GAIN. See command set documentation to help determine which to use.
M_RFI_GAIN	Use M_NOISE_GAIN, M_NOISE_GAINEX, or M_MICRO_GAIN. See command set documentation to help determine which to use.
M_XTALK_GAIN	Use M_NOISE_GAIN, M_NOISE_GAINEX, or M_MICRO_GAIN. See command set documentation to help determine which to use.
M_OUTPUT_FREQ	None
M_IMPULSE_TYPE	M_LOAD_FILE followed by sequence to calculate, generate, and output.
M_IMPULSE_LEVEL	Use M_NOISE_GAIN, M_NOISE_GAINEX, or M_MICRO_GAIN. See command set documentation to help determine which to use.
M_CUSTOMNOISE_MUTE	None
M_CFRC_ENABLE	None – CFRC no longer supported
M_OUTPUT_CFRC	None – CFRC no longer supported
CREST_FACTOR	M_CREST_FACTOR
NUMBER_SAMPLES	M_NUMBER_SAMPLES
GENERATE_SAMPLE	M_GENERATE_SAMPLE

Also the following is available only as 'get' (cannot be set): M_INJECTOR_LOSS.

Chapter 8 Specifications

8.1 Standards

The DLS-5800 simulates the noise requirements as currently defined by ETSI, ANSI, ITU and Broadband Forum. Updates will be provided by Spirent Communications Access Emulation as the standards progress. Other xDSL noise file CD/DVDs are available for purchase for these products. Please contact your local sales representative for details.

Spirent Communications Access Emulation reserves the right to offer additional and different noise profiles in the future as separately packaged offerings, compatible with the DLS-5800.

8.2 System Specifications

- PC with minimum 2 GB RAM, hard disk drive, minimum 6 USB ports, and CD/DVD drive.
- Arbitrary Waveform Generator and Storage
- Pre-installed software:
    a. Windows XP
    b. DLS-5800 Control Software
    c. Sample ADSL and VDSL Noise Profiles – included

8.3 Mechanical and Environmental Specifications

Rear Connectors

Noise	DLS-5800 BNC 4 outputs per card (DLS 5410, 8 BNC Inputs/card)
Interface	RS232 standard, 6 USB ports
Control	DLS-5800: Ethernet; DLS-5410DC: USB; DLS-5409: None
Printer	Standard PC printer connector
Item	Description
Mouse	USB interface
Monitor	Standard VGA connector
Power	World wide standard (universal power supply)

Spirent Communications Supplied

PC compatible 101 QWERTY keyboard

PC compatible mouse

User Supplied or Spirent Communications Options

PC compatible color VGA compatible monitor (SVGA and

XVGA supported) PC compatible printer

Dimensions (with rack-mounting brackets) mounted	DLS-5800: 4U Rack mounted unit. DLS-5410DC: 1U Rack
	DLS-5800: 483 x 545 x 177 mm (19 x 21.5 x 7 inch)
	DLS-5410DC: 483 x 483 x 45 mm (19 x 19 x 1.8 inch)
Net Weight	DLS-5800: 19kg (42 lbs). DLS-5410DC: 4.8 kg (10.6 lbs)
Environmental	
Operating Temperature	10°C to 40°C (50°F to 104°F)
Storage Temperature	-4ºF to 158ºF (-20ºC to +70ºC).
Relative Humidity	10 to 95% RH non-condensing

8.4 Electrical Specifications

Item	Description
Rated Input Voltage	DLS-5410DC: Switchable 115/230 VAC 50/60Hz. DLS-5800: Autoranging: 100-240VAC (±10%), 50/60 Hz.
Rated Power Consumption	DLS-5800: 650VA max.
	DLS-5410DC: 1.5(0.75)A max 115(230)VAC
Line fuses	DLS-5800: no user-replaceable fuses DLS-5410DC: 115 VAC (T1.5A Glass Tube Type qty = 2) or 230 VAC (T0.75A Glass Tube Type qty = 2)
	Where “T” denotes time-lag type of fuses

Note: All DLS-5800 units come with an auto-switching power supply. DLS 5410DC may require user intervention when powering up the unit. Please verify voltage settings of the external power supply of DLS-5410DC prior to powering on the unit.

8.5 DLS-5800Technology

The DLS-5800 is a PC-based Arbitrary Waveform Generator with 8, 16, or 24 independent channels. Each generator (channel) uses a 14-bit DAC for excellent dynamic range. Designed for the testing of xDSL modems and DSLAMs, the DLS-5800 generates impairments such as Crosstalk Noise, White Noise, RFI tones, and Impulse noise (time domain based).

The DLS-5410DC Noise Injector can be used to apply these impairments to a real or simulated wireline in differential or common mode, and adds high-speed, high precision hardware-based signal switching in order to add impulse noise types such as REIN, PEIN and SHINE, as well as time varying noise sequences such as DSM (Dynamic Spectrum Management).

The DLS-5409 Passive Noise Injector applies the impairments generated by DLS-5800 to a real or simulated wireline in differential mode.

Additionally, various applications (DLS-5C80) are available to help create custom impulse noises.

Item	Description
Crest Factor	Greater than 5 in ETSI compliant mode using memory of 0.5 MegaWords or less. By using a longer memory, truly Gaussian distribution and crest factor greater than 5 can also be generated
Execution Memory DLS-5800	32 MB (16 Megasamples) for each channel (32 MB internal fast execution)
Maximum Output DLS-5800	±10V (20 V p-p) unloaded, ±5 V (10 V p-p) into 50 Ω
Maximum RMS Output DLS-5800	+ 13 dBm into 50 Ω with a crest factor of 5
Output Impedance DLS-5800	50 Ω single-ended channels
Output Impedance DLS 5410DC	Differential mode >4K Ω, Common mode 150 Ω +/- 5%
Channel Noise Floor output	DLS 5410DC Less than -150 dBm/Hz in differential and common mode injection
Frequency Range	4 kHz to 30 MHz for differential mode. 150 kHz to 30 MHz for common mode
Channel Dynamic Range DLS-5800	100 dB (one single sine wave)
Effective Dynamic Range	Greater than 60 dB from peak to lowest power in the impairment profile within the dynamic range of the instrument for any impairment with crest factor of 5
Accuracy of PSD Profile	Less than 0.5 dB mean absolute error (MAE) for all profiles
Accuracy of PSD Level	The DLS-5800 is calibrated to be within 0.5 dB

	mean error (ME) for all profiles, as measured using an HP 4395A Analyzer
DLS 5410DC (Injector Loss)	Differential mode 15.3dB +/-.25dB, Common mode 4.8dB +/- .25dB

DLS-5409 Specifications	
DC ratings between Tip-Ring, Tip-Ground, Ring-Ground:	400VDC
Attenuation (Input-Output):	22 dB -/+0.6 dB between 400 kHz and 30 MHz 22 dB –0/+1 dB between 150 kHz and 400 kHz 22 dB –0/+1.5 dB at 100 kHz 22 dB –0/+3.5 dB at 50 kHz
Loading of the line:	< 0.108 dB from 100 kHz to 15 MHz < 0.14 from 15 MHz to 30 MHz
Maximum input power:	+10 dBm with crest factor = 5
Size:	3 x 1.4 x 1.2 inch
Weight:	less than 100 grams

8.6 Spirent Communications Options

- IEEE 488: ACC-GPPCINT
- DLS 5Bxx Noise Impairment Libraries
- DLS 5C80 Custom Impulse Noise Creation Tool
- DLS 5C60 xDSL xtalk creation tool and performance prediction tool

Chapter 9 Appendix A - Measuring Injection Loss

To measure the actual insertion loss of a specific noise generator injection unit:

1. Generate several signals within the required xDSL frequency band using a frequency generator.
2. Capture the signals using a spectrum analyzer.
3. Connect the Noise Injection Circuit as shown in *Figure 9-1: Calculating Gain/Injector Attenuation* and apply a fixed load to the output of the circuit with a Balun to match the load impedance.
4. Take a new capture of the signals with the spectrum analyzer.

The idea is to have a substantial output power of the AWG (above 5 dBm), which should then be attenuated by an external attenuator. This has two advantages:

- The final desired signal can be created with maximum accuracy, without, for example, the influence of background noise in the hardware of the noise generator.
- By *decreasing* the external attenuation, the power of the noise signal on the line can be smoothly increased during a performance test.

Chapter 10

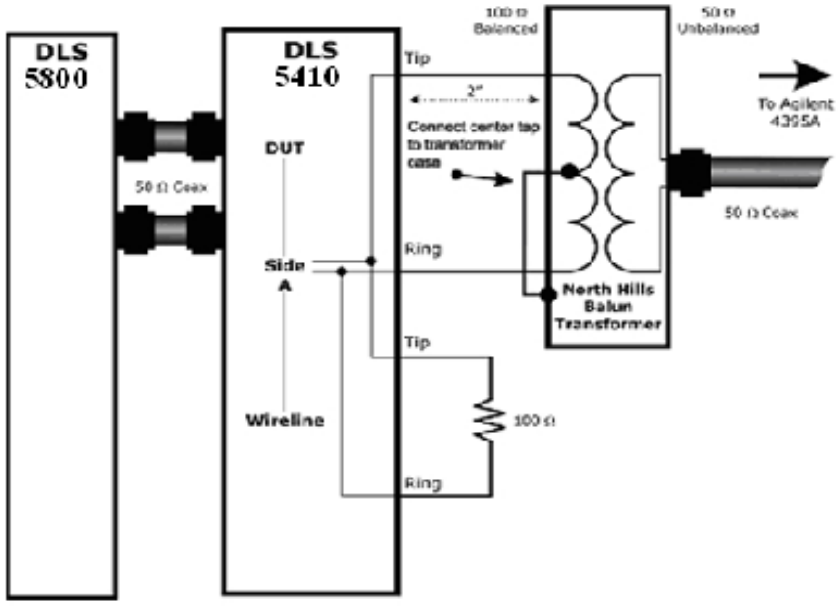


Figure 9-1: Calculating Gain/Injector Attenuation

Appendix B - Safety Information and Instructions

10.1 Safety Information

10.1.1 Protective Grounding (Earthing)

These units consists of an exposed metal chassis that must connect directly to a ground (earth) by means of a protective grounding conductor in the power cord.The symbol used to indicate a protective grounding conductor terminal in the equipment is shown in *Symbols*.

10.1.2 Before Operating the Unit

- Inspect the equipment for any signs of damage, and read this manual thoroughly.
- Become familiar with all safety symbols and instructions in this manual to ensure that the equipment is used and maintained safely.
- Warning: To avoid risk of injury or death, ALWAYS observe the following precautions before operating the unit:
- Use only a power supply cord with a protective grounding terminal.
- Connect the power supply cord only to a power outlet equipped with a protective earth contact. Never connect to an extension cord that is not equipped with this feature. Do not interrupt the protective earth connection.
- Caution: When lifting or moving the units, do not touch the cooling fan, which is located at the back of the unit. Lift the unit by using the handles on the front.

10.1.3 Power Supply Requirements

The units can operate from any single phase AC power source that supplies between 100 VAC and 240 VAC (±10%) at a frequency range of 50 Hz to 60 Hz.

Warning: To avoid electrical shock, do not operate the equipment if it shows any sign of damage to any portion of its exterior surface, such as the outer casting or panels.

10.1.4 Main Fuse Type

The fuse type used is specified in Appendix A, “Specifications.”

10.1.5 Connections to a Power Supply

In accordance with international safety standards, the units use a three-wire power supply cord. When connected to an appropriate AC power receptacle, this cord grounds the equipment chassis.

Note: When connecting the AC power cable input into the back of the DLS-5800 and DLS-5410DC, please ensure that the AC power cable is positioned so that it is not difficult to remove and so that there are no obstructions present that would prevent its removal.

10.1.6 Operating Environment

To prevent potential fire or shock hazard, do not expose the equipment to any source of excessive moisture.

10.1.7 Class of Equipment

The unit consists of an exposed metal chassis that is connected directly to earth via the power supply cord. In accordance with HARMONIZED EUROPEAN STANDARD EN 61010-1:1993, it is classified as Safety Class I equipment.

Warning: This is a class A product. This product may cause radio interference. In this case you may be required to take adequate measures to mitigate this problem in a domestic environment.

10.2 Safety Instructions

The following safety instructions must be observed whenever the unit is operated, serviced or repaired. Failing to comply with any of these instructions or with any precaution or warning contained in the Operating Manual is in direct violation of the standards of design, manufacture and intended use of the equipment.

Spirent Communications assumes no liability for the customer's failure to comply with any of these requirements.

10.2.1 Before Operating the Unit

- Inspect the equipment for any signs of damage, and read the User Guide (this document) thoroughly.
- Install the equipment as specified in the relevant section of this manual.
- Ensure that the equipment and any devices or cords connected to it are properly grounded.
- Ensure that the equipment has enough clearance at the front and at the rear (minimum 4 inches or 100 mm) to allow proper air ventilation.

10.2.2 Operating the Unit

- Do not operate the equipment when its covers or panels have been removed.
- Do not interrupt the protective grounding connection. Any such action can lead to a potential shock hazard that could result in serious personal injury.
- Do not operate equipment if an interruption to the protective grounding is suspected.
- Use only the type of fuse specified.
- Do not use repaired fuses and avoid any situation that could short circuit the fuse.
- Unless absolutely necessary, do not attempt to adjust or perform any maintenance or repair procedure when the equipment is opened and connected to a power source at the same time. Any such procedure should only be performed by qualified service professional.
- Do not attempt any adjustment, maintenance or repair procedure to the equipment if first aid is not accessible.
- Disconnect the power supply cord from the equipment before adding or removing any components.
- Operating the equipment in the presence of flammable gases or fumes is extremely hazardous.
- Ensure that the equipment has enough clearance at the front and at the rear (minimum 4 inches or 100 mm) to allow proper air ventilation.

- Do not perform any operating or maintenance procedure that is not described in this User Guide.
- Protection will be impaired if the equipment is used in a manner not specified by the manufacturer.
- The battery, which is installed on the Single-Board Computer of DLS-5800, is not replaceable by the user.
- Some of the equipment's capacitors may be charged even when the equipment is not connected to the power source.

10.3 Symbols

FUNCTIONAL GROUND TERMINAL

PROTECTIVE GROUNDING CONDUCTOR TERMINAL

CAUTION - REFER TO ACCOMPANYING DOCUMENTS

Appendix C - ESD Requirements

Spirent Communications manufactures and sells products that require industry standard precautions to protect against damage from electrostatic discharge (ESD). This document explains the proper process for handling and storing electrostatic discharge sensitive (ESDS) devices, assemblies, and equipment.

The requirements presented in this document comply with the EIA Standard, *ANSI/ESD S20.20-1999: Development of an Electrostatic Discharge Control Program,* and apply to anyone who handles equipment that is sensitive to electrostatic discharge. Such equipment includes, but it not limited to:

- All electronic assemblies manufactured by Spirent Communications
- Discrete and integrated circuit semiconductors
- Hybrid microcircuits
- Thin film passive devices

  Memory modules

Caution: Failure to comply with the requirements explained in this document poses risks to the performance of ESDS devices, as well as to your investment in the equipment.

11.1 General Equipment Handling

Whenever you handle a piece of ESDS equipment, you must be properly grounded to avoid harming the equipment. Also, when transporting the equipment, it must be packaged properly. Follow the requirements below to help ensure equipment protection.

- Wrist straps must be worn by any person handling the equipment to provide normal grounding.
- The use of foot straps is encouraged to supplement normal grounding. If foot straps are used exclusively, two straps (one on each foot) should be used. Note that foot straps are only applicable in environments that use ESD flooring and/or floor mats.
- Hold ESDS equipment by the edges only; do not touch the electronic components or gold connectors.
- When transporting equipment between ESD protected work areas, the equipment must be contained in ESD protective packaging. Equipment that is received in ESD protective packaging must be opened either by a person who is properly grounded or at an ESD protected workstation.
- Any racks or carts used for the temporary storage or transport of ESDS equipment must be grounded either by drag chains or through direct connection to earth ground. Loose parts that are not protected by ESD-safe packaging must not be transported on carts.

11.2 Workstation Preparation

The ideal setup for working with ESDS equipment is a workstation designed specifically for that purpose. Please follow the requirements listed below to prepare a proper ESD protected workstation.

The ESD Ground must be the equipment earth ground. Equipment earth ground is the electrical ground (green) wire at the receptacles.

An ESD protected workstation consists of a table or workbench with a static dissipative surface or mat that is connected to earth ground. A resistor in the grounding wire is optional, providing that surface resistance to ground is ≥ 105 to ≤ 109 Ω.

The workstation must provide for the connection of a wrist strap. The wrist strap must contain a current limiting resistor with a value from ≥ 250K Ω to ≤ 10M Ω.

ESD protective flooring or floor mats are required when floor-grounding devices (foot straps/footwear) are used or when it is necessary to move in between ESD protected workstations when handling ESDS equipment.

Note: The equipment needed for proper grounding is available in ESD service kits, such as the ESD Field Service Kit available from Spirent Communications (P/N 170-1800). Additional information on ESD can be found on the following website:

http://www.esda.org/aboutesd.html