Tektronix®

Tektronix LE160 & LE320
Linear Equalizers
Instruction Manual

Warning

The servicing instructions are for use by qualified personnel only. To avoid personal injury, do not perform any servicing unless you are qualified to do so. Refer to all safety summaries prior to performing service.

Revision A, October 10, 2016

077088303

Contacting Tektronix

Tektronix, Inc.
14150 SW Karl Braun Drive
P.O. Box 500
Beaverton, OR 97077
USA

For product information, sales, service, and technical support:

- In North America, call 1-800-833-9200.
- Worldwide, visit *www.tek.com* to find contacts in your area.

Table of Contents

Getting started

Operating basics

Important safety information

This manual contains information and warnings that must be followed by the user for safe operation and to keep the product in a safe condition.

To safely perform service on this product, see the *Service safety summary* that follows the *General safety summary*.

General safety summary

Use the product only as specified. Review the following safety precautions to avoid injury and prevent damage to this product or any products connected to it. Carefully read all instructions. Retain these instructions for future reference.

Comply with local and national safety codes.

For correct and safe operation of the product, it is essential that you follow generally accepted safety procedures in addition to the safety precautions specified in this manual.

The product is designed to be used by trained personnel only.

Only qualified personnel who are aware of the hazards involved should remove the cover for repair, maintenance, or adjustment.

This product is not intended for detection of hazardous voltages.

While using this product, you may need to access other parts of a larger system. Read the safety sections of the other component manuals for warnings and cautions related to operating the system.

When incorporating this equipment into a system, the safety of that system is the responsibility of the assembler of the system.

To avoid fire or personal injury

Use proper power cord. Use only the power cord specified for this product and certified for the country of use. Do not use the provided power cord for other products.

Ground the product. This product is grounded through the grounding conductor of the power cord. To avoid electric shock, the grounding conductor must be connected to earth ground. Before making connections to the input or output terminals of the product, ensure that the product is properly grounded. Do not disable the power cord grounding connection.

Power disconnect. The power cord disconnects the product from the power source. See instructions for the location. Do not position the equipment so that it is difficult to operate the power cord; it must remain accessible to the user at all times to allow for quick disconnection if needed.

Use proper AC adapter. Use only the AC adapter specified for this product.

Observe all terminal ratings. To avoid fire or shock hazard, observe all rating and markings on the product. Consult the product manual for further ratings information before making connections to the product.

Do not apply a potential to any terminal, including the common terminal, that exceeds the maximum rating of that terminal.

Do not float the common terminal above the rated voltage for that terminal.

The measurement terminals on this product are not rated for connection to mains or Category II, III, or IV circuits.

Do not operate with suspected failures. If you suspect that there is damage to this product, have it inspected by qualified service personnel.

Disable the product if it is damaged. Do not use the product if it is damaged or operates incorrectly. If in doubt about safety of the product, turn it off and disconnect the power cord. Clearly mark the product to prevent its further operation.

Examine the exterior of the product before you use it. Look for cracks or missing pieces.

Use only specified replacement parts.

Use proper fuse. Use only the fuse type and rating specified for this product.

Wear eye protection. Wear eye protection if exposure to high-intensity rays or laser radiation exists.

Do not operate in wet/damp conditions. Be aware that condensation may occur if a unit is moved from a cold to a warm environment.

Do not operate in an explosive atmosphere.

Keep product surfaces clean and dry. Remove the input signals before you clean the product.

Provide proper ventilation. Refer to the manual's installation instructions for details on installing the product so it has proper ventilation.

Slots and openings are provided for ventilation and should never be covered or otherwise obstructed. Do not push objects into any of the openings.

Provide a safe working environment. Always place the product in a location convenient for viewing the display and indicators.

Avoid improper or prolonged use of keyboards, pointers, and button pads. Improper or prolonged keyboard or pointer use may result in serious injury.

Be sure your work area meets applicable ergonomic standards. Consult with an ergonomics professional to avoid stress injuries.

Service safety summary

The *Service safety summary* section contains additional information required to safely perform service on the product. Only qualified personnel should perform service procedures. Read this *Service safety summary* and the *General safety summary* before performing any service procedures.

To avoid electric shock. Do not touch exposed connections.

Do not service alone. Do not perform internal service or adjustments of this product unless another person capable of rendering first aid and resuscitation is present.

Disconnect power. To avoid electric shock, switch off the product power and disconnect the power cord from the mains power before removing any covers or panels, or opening the case for servicing.

Use care when servicing with power on. Dangerous voltages or currents may exist in this product. Disconnect power, remove battery (if applicable), and disconnect test leads before removing protective panels, soldering, or replacing components.

Verify safety after repair. Always recheck ground continuity and mains dielectric strength after performing a repair.

Terms in the manual

These terms may appear in this manual:

WARNING. *Warning statements identify conditions or practices that could result in injury or loss of life.*

CAUTION. *Caution statements identify conditions or practices that could result in damage to this product or other property.*

Terms on the product

These terms may appear on the product:

- DANGER indicates an injury hazard immediately accessible as you read the marking.
- WARNING indicates an injury hazard not immediately accessible as you read the marking.
- CAUTION indicates a hazard to property including the product.

Symbols on the product

When this symbol is marked on the product, be sure to consult the manual to find out the nature of the potential hazards and any actions which have to be taken to avoid them. (This symbol may also be used to refer the user to ratings in the manual.)

The following symbols may appear on the product:

Compliance Information

This section lists the EMC (electromagnetic compliance), safety, and environmental standards with which the instrument complies.

EMC compliance

EC Declaration of Conformity – EMC

Meets intent of Directive 2004/108/EC for Electromagnetic Compatibility. Compliance was demonstrated to the following specifications as listed in the Official Journal of the European Communities:

EN 61326-1. EMC requirements for electrical equipment for measurement, control, and laboratory use. [1 2 3]

- CISPR 11. Radiated and conducted emissions, Group 1, Class A
- IEC 61000-4-2. Electrostatic discharge immunity
- IEC 61000-4-3. RF electromagnetic field immunity
- IEC 61000-4-4. Electrical fast transient / burst immunity
- IEC 61000-4-5. Power line surge immunity
- IEC 61000-4-6. Conducted RF immunity
- IEC 61000-4-11. Voltage dips and interruptions immunity

IEC 61000-4-8. Power frequency magnetic field immunity test. EN 61000-3-2

EN 61000-3-3. Voltage changes, fluctuations, and flicker

Mfr. Compliance Contact.
Tektronix, Inc. PO Box 500, MS 19-045
Beaverton, OR 97077, USA
www.tek.com

---

1 This product is intended for use in nonresidential areas only. Use in residential areas may cause electromagnetic interference.
2 Emissions which exceed the levels required by this standard may occur when this equipment is connected to a test object.
3 For compliance with the EMC standards listed here, high quality shielded interface cables should be used.

Australia / New Zealand Declaration of Conformity – EMC

Complies with the EMC provision of the Radiocommunications Act per the following standard, in accordance with ACMA:

- CISPR 11. Radiated and conducted emissions, Group 1, Class A, in accordance with EN 61326-1.

Safety compliance

This section list the safety compliance information.

Equipment type

Test and measuring equipment.

Safety class

Class 1 – grounded product.

Pollution degree description

A measure of the contaminants that could occur in the environment around and within a product. Typically the internal environment inside a product is considered to be the same as the external. Products should be used only in the environment for which they are rated.

- Pollution Degree 1. No pollution or only dry, nonconductive pollution occurs. Products in this category are generally encapsulated, hermetically sealed, or located in clean rooms.
- Pollution Degree 2. Normally only dry, nonconductive pollution occurs. Occasionally a temporary conductivity that is caused by condensation must be expected. This location is a typical office/home environment. Temporary condensation occurs only when the product is out of service.
- Pollution Degree 3. Conductive pollution, or dry, nonconductive pollution that becomes conductive due to condensation. These are sheltered locations where neither temperature nor humidity is controlled. The area is protected from direct sunshine, rain, or direct wind.
- Pollution Degree 4. Pollution that generates persistent conductivity through conductive dust, rain, or snow. Typical outdoor locations.

Pollution degree

Pollution Degree 2 (as defined in IEC 61010-1). Note: Rated for indoor, dry location use only.

Measurement and overvoltage category descriptions

Measurement terminals on this product may be rated for measuring mains voltages from one or more of the following categories (see specific ratings marked on the product and in the manual).

- Measurement Category II. For measurements performed on circuits directly connected to the low-voltage installation.
- Measurement Category III. For measurements performed in the building installation.
- Measurement Category IV. For measurements performed at the source of low-voltage installation.

NOTE. *Only mains power supply circuits have an overvoltage category rating. Only measurement circuits have a measurement category rating. Other circuits within the product do not have either rating.*

Mains overvoltage category rating

Overvoltage Category II (as defined in IEC 61010-1)

Environmental compliance

This section provides information about the environmental impact of the product.

Product end-of-life handling

Observe the following guidelines when recycling an instrument or component:

Equipment recycling. Production of this equipment required the extraction and use of natural resources. The equipment may contain substances that could be harmful to the environment or human health if improperly handled at the product's end of life. To avoid release of such substances into the environment and to reduce the use of natural resources, we encourage you to recycle this product in an appropriate system that will ensure that most of the materials are reused or recycled appropriately.

This symbol indicates that this product complies with the applicable European Union requirements according to Directives 2012/19/EU and 2006/66/EC on waste electrical and electronic equipment (WEEE) and batteries. For information about recycling options, check the Tektronix Web site (*www.tek.com/productrecycling*).

Restriction of hazardous substances

This product is classified as an industrial monitoring and control instrument, and is not required to comply with the substance restrictions of the recast RoHS Directive 2011/65/EU until July 22, 2017.

Preface

The LE160 and LE320 linear equalizers support signal equalization on data rates from 8 Gbps to 32 Gbps with 29 dB automatic or manual gain control. The linear equalizers can be used as a stand-alone instrument with a host computer or can be used as part of a Tektronix BERTScope configuration.

Documentation

The following documentation is available to support your linear equalizer:

- *LE160 & LE320 Quick Reference Installation Instructions* (Tektronix part number, 071-3222-xx). Printed instructions shipped with each instrument providing high-level installation information.
- *LE160 & LE320 Linear Equalizer Instruction Manual* (Tektronix part number, 077-0883-xx). This manual, PDF-only. This document provides high-level information for using and maintaining the linear equalizer.
- *LE160 & LE320 Linear Equalizer Declassification & Security Instructions* (Tektronix part number, 077-0882-xx) PDF only. This document provides instructions to declassify or sanitize your instrument.
- *LE160 & LE320 Linear Equalizer Product Specification and Performance Verification Technical Reference Manual* (Tektronix part number 077-0884-xx). This manual lists the product specifications and provides procedures to verify instrument performance.

Getting started

Product description

The LE160 and LE320 Linear Equalizers are intended for use with Tektronix BERTScope instruments and performance oscilloscopes up to 32 Gb/s. The linear equalizers are remotely controlled by an external host PC or a Tektronix BERTScope through a USB 2.0 connection.

The LE160 Linear Equalizer operates at rates up to 16 Gb/s and the LE320 Linear Equalizer operates at rates up to 32 Gb/s.

The small physical design allows you to place the instrument as close to the DUT (Device-Under-Test) as possible.

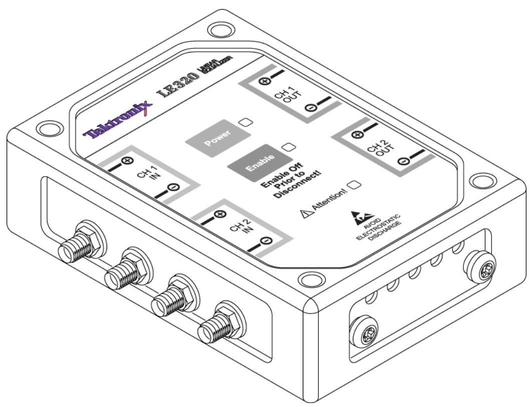


Figure 1: LE320 Linear Equalizer

Key features

- Clock-less design transfers incident signal impairments to outputs. No clock-to-data alignment concerns.
- Multi UI linear analog (9 tap, 18 ps tap-delay spaced for the LE320 only. The LE160 has 24 ps tap spacing.)
- Fixed tap delay design eliminates external reference clock requirements.
- USB programmable receiver gain and multi UI equalization allows 20 dB of equalization.
- USB programmable channel modeling allows electronic channel emulation.
- USB programmable pre-emphasis and differential outputs amplitude as high as 2.7 V differential.
- USB programmable output duty cycle symmetry control.
- Precision output level controls permit signaling from “0” (Return to Zero) to 2.7 V differentially.
- User defined presets and stored configurations tailored for key technologies.
- Two models:
  - LE160, 16 GB/s linear equalizer
  - LE320, 32 GB/s linear equalizer

Standard accessories

The instrument comes with the following accessories:

- AC power adapter, Tektronix part number 119-7836-xx
- USB cable, USB-A to Mini USB-B for PC connection, Tektronix part number 174-6106-xx
- Two (2) 6-in coaxial cables, K male to K male, for linear equalizer-to-DUT connection, Tektronix part number 174-6362-xx
- 1 m coaxial cable, phase-matched pair, Tektronix part number 174-5969-xx

Instrument options

The following instrument options are available for your product:

Option	Description
CDS	Adds channel design software
SPM	Adds S-parameter channel modeler

Installation

Unpack the instrument and check that you have received all items listed as Standard Accessories. Check the Tektronix Web site (www.tektronix.com) for the most current information.

Site considerations

The linear equalizer is intended to be operated in a controlled laboratory environment. The instrument can be secured above the DUT to keep the distance between the linear equalizer and the DUT to a minimum. Use the four hanging screws, provided, to secure the instrument from user-supplied hardware.

⚠ ***CAUTION.***

Using the wrong size hanging screws can damage the linear equalizer by over-tightening the screws. To avoid damaging the linear equalizer use only the screws supplied with the product.

Provide adequate clearance around the vent holes of the linear equalizer to provide proper instrument cooling.

⚠ ***CAUTION.***

To prevent damage to the linear equalizer, terminate any used input or output ports.

Operating requirements

You will need the following equipment and software to install the instrument:

- Host PC for stand-alone operation or a Tektronix BERTScope BSA175C or BSA286C when used in a BERTScope configuration.
- Linear equalizer
- Tektronix DSA8300 oscilloscope with the following:
  - 82A04B Phase reference module
  - 80E09 or 80E10 Sampling head (50 GHz minimum bandwidth)
  - 80X01 – DSA8300 Extender cable (2 required)
- USB 2.0 cable, A-to-mini-B
- 2.92 mm cables, 6 in, coaxial, skew-matched pair
- 2.92 mm cable, 1 m, coaxial
- (Optional) PC w/Windows XP, or Windows 7 32-Bit, or 64-Bit operating system

NOTE. *Cables that connect an instrument from the Tektronix BERTScope Data Out signals to the linear equalizer should be of nominal length and/or phase matched, depending on your application.*

Preventing electrostatic discharge (ESD)

⚠ ***CAUTION.***

A direct electrostatic discharge can damage the instrument input. To learn how to avoid this damage, read the following information.

Electrostatic discharge (ESD) is a concern when handling any electronic equipment. The instrument is designed with robust ESD protection; however it is still possible that large discharges of static electricity directly into the signal input may damage the instrument. To avoid damage to the instrument, use the following techniques to prevent electrostatic discharge to the instrument:

- Discharge the static voltage from your body by wearing a grounded antistatic wrist strap while connecting and disconnecting cables and adapters. The instrument provides a front panel connection for this purpose.
- Discharge any static voltage from all cables before connecting them to the instrument or device under test by momentarily grounding the center conductor of the cable, or by connecting a 50 Ω termination to one end, prior to attaching the cable to the instrument.

A cable that is left unconnected on a bench can develop a large static charge.

- Nothing capable of generating or holding a static charge should be allowed on the work station surface.

Configuration

The following figure shows a typical connection block diagram of the linear equalizer connected to Tektronix DSA8300 oscilloscope and to a Tektronix BERTScope or host PC. Your configuration may vary.

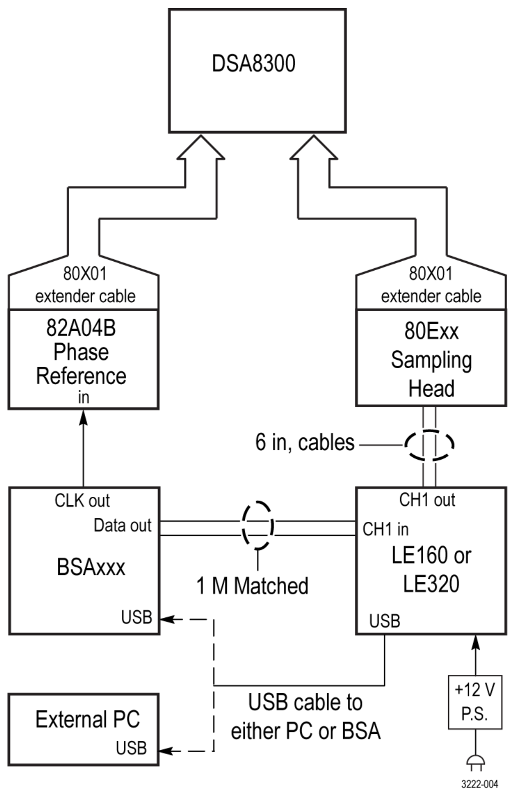


Figure 2: Block diagram of the linear equalizer connected to other equipment

Download the LE application software

Complete the following steps to download the latest LE application software on the host PC:

1. Go to the Tektronix Web site at (http://www.tek.com).
2. Enter **Signal Conditioning and Equalization** in the search field and click **Go**.
3. Locate and double-click the link for the LE Application Software.
4. Follow the on-screen instructions.

Incoming inspection and first-time operation

Power on the instrument

Complete the following steps to power on the linear equalizer and to perform a brief functional check.

1. Connect the USB cable from the linear equalizer to the host PC.
2. Connect the power adapter to a power source and to the linear equalizer.

   NOTE. *The power adapter provides the required power for the input and output signals to work properly. Although the linear equalizer might appear to operate by only connecting to the USB port, some USB ports do not provide enough power for the linear equalizer hardware. Avoid operating the linear equalizer without the power adapter.*

3. Push the Power button on top of the linear equalizer to turn the power on; the Power LED should turn on.
4. Open the LE application on the host PC.

   The Connect view displays showing a list of instruments.
5. Select the linear equalizer from the list and click **CONNECT**; verify that the Connect view displays the message confirming the connection.
6. Depending on your configuration, select one of the operating modes from the buttons at the top of the Connect view.

   NOTE. *Any grayed-out buttons at the top of the Connect view indicate that you might not have the current licenses for those operating modes.*

7. If the input and output cables are connected to the linear equalizer, Push the Enable button on top of the linear equalizer to enable the signals.

   NOTE. *Do not connect or disconnect any cables while the Enable LED is turned on.*

Power off the instrument

1. If you have not done so, push the Enable button on top of the linear equalizer to turn off any signals; verify that the Enable LED is off.
2. Push the Power button to turn the power off; the power LED should turn off.
3. Disconnect the power adapter and the USB cable from the linear equalizer.

Operating basics

Instrument connections

This section describes the controls and connectors of the linear equalizer. The controls and connectors are the same for each version of the linear equalizer, unless specifically mentioned.

Top view The following figure and table describe the controls and connectors on the linear equalizer from the top of the instrument.

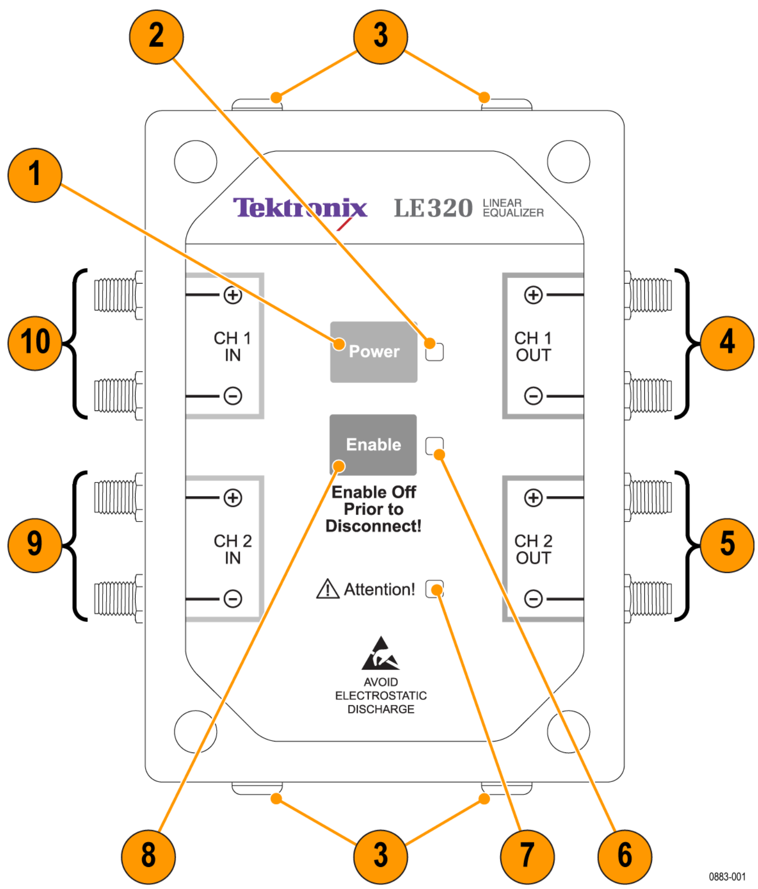


Figure 3: Linear equalizer top view

Table 1: Top view

Item		Description
1	Power button	Push this button to apply power to the linear equalizer from the connected power adapter.
2	Power LED	This LED turns on when the Power button is pushed to apply power to the linear equalizer. Power is applied to the controller section as soon as the power adapter or USB connector are connected.
3	Hanging screws	Use these screws (two on the front and two on the rear) as needed to secure the linear equalizer above the DUT. Do not use any screws besides the ones supplied with the linear equalizer to avoid damaging the instrument.
4	CH 1 OUT	Plus and minus differential Channel 1 output connectors.
5	CH 2 OUT	Plus and minus differential Channel 2 output connectors.
6	Enable LED	Push this button to enable signals from the attached cables to the connectors.
7	Attention LED	If this LED turns on, there may be a hardware fault. To clear this LED, try cycling the power on the linear equalizer. If the LED stays on, contact your local Tektronix representative for further action.
8	Enable button	Push this button to turn on the Enable LED.
9	CH 2 IN	Plus and minus differential Channel 2 input connectors.
10	CH 1 IN	Plus and minus differential Channel 1 input connectors.

Side view

The following figure and table describe the controls and connectors on the side of the linear equalizer.

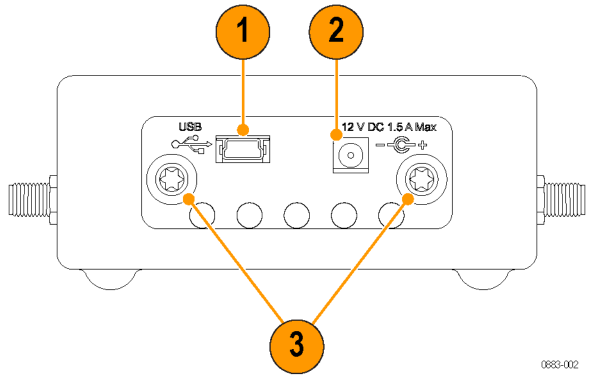


Figure 4: Linear equalizer side view

Table 2: Side view

Item		Description
1	USB IN	USB 2.0 type A connector used to interface the linear equalizer to a host PC or BERTScope instrument.
2	12 V DC input	Power adapter connector. Provides power to the linear equalizer from the power adapter. Always operate the linear equalizer with the power adapter connected. The USB ports on the host computer might not have enough power to drive the linear equalizer hardware.
3	Hanging screws	Use these screws (two on the front and two on the rear) as needed to secure the linear equalizer above the DUT. Do not use any screws besides the ones supplied with the linear equalizer to avoid damaging the instrument.

User interface

This section provides a high-level overview of the linear equalizer user interface.

Connect view

The Connect view displays when you start the LE software application; it shows a list of instruments available for connection.

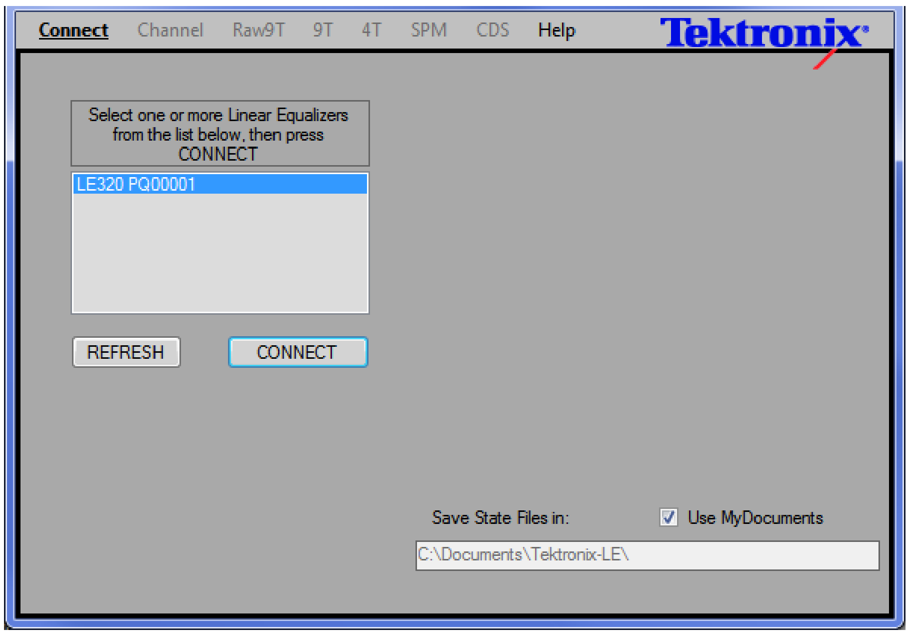


Figure 5: Connect view before connecting to the linear equalizer

If instrument list does not show your linear equalizer, verify that the USB cable is connected to the host PC and to the linear equalizer. Verify that the Power LED is on. Click **REFRESH** to refresh the list after verifying that the linear equalizer is properly connected and power is turned on.

***NOTE.** The application can connect to more than one linear equalizer at a time. Left-click the mouse to select the linear equalizer. To connect to more than one linear equalizer, use a shift-left-click or Ctrl-left-click mouse combination to select the linear equalizer.*

Click **CONNECT** to connect to the linear equalizer. After connecting to the linear equalizer, the button label changes to DISCONNECT. Click **DISCONNECT** to disconnect from the linear equalizer highlighted in the instrument list.

After a successful connection the application sets the linear equalizer to an initial configuration with both channel outputs disabled. Any hardware settings from a previous connection session will be overwritten during the connection process.

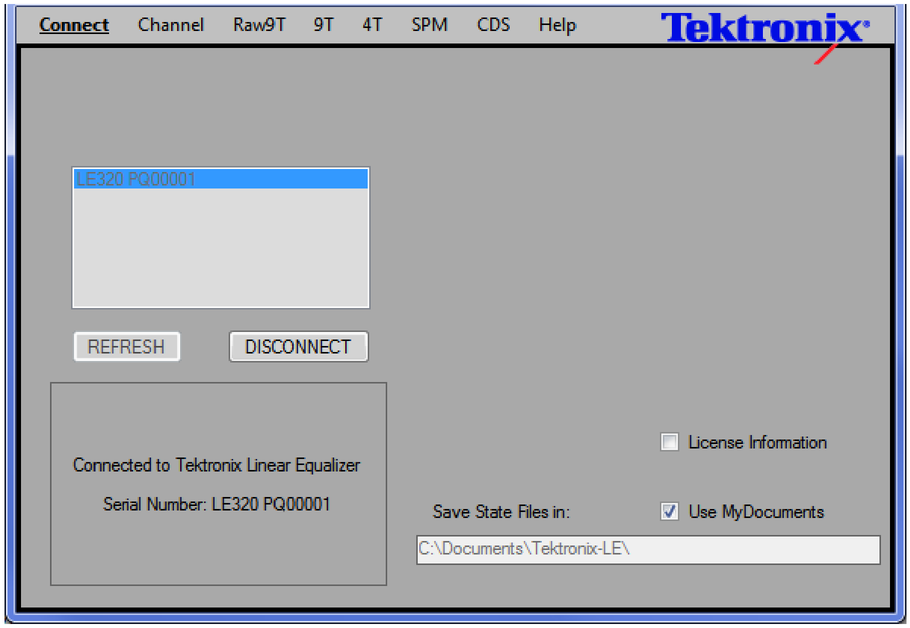


Figure 6: Connect view after successful connection

Select the **License information** check box to show the current license information at the bottom of the window. Refer to *Appendix A: LE160/LE320 software license upgrade information* for information on upgrading the license information for your instrument.

***NOTE.** The License Information check box is not available when you connect to multiple linear equalizers. To access the license information, connect only to one linear equalizer at a time.*

Use the **Use MyDocuments** check box to determine the location of the save/ recall locations. If a MyDocuments folder does not exist, the application will create a Tektronix-LE subfolder where it can save the state files. Clear the check box to save or recall the state information to a custom location on the PC listed in the **Save State Files in:** field.

After you successfully connect to a licensed linear equalizer, click the buttons at the top of the window to open one of the operating modes. Grayed-out selections indicate that a license for the operating mode is not available. The current operating mode for a selected channel is always shown in underlined, bold-face type at the top of the view.

***NOTE.** Use the Channel menu to control either Channel 1 or Channel 2 from a particular mode. When multiple linear equalizers are connected, the Channel menu also allows the choice of a specific linear equalizer to be adjusted.*

Common controls

All operating modes have the following controls in common between the views:

- Enable/Disable control
- Output Amplitude control
- Input Offset control
- Save and Load controls
- AGC control
- AGC Lock control

Enable/Disable controls. Use the Enable/Disable button to turn the outputs of the linear equalizer on or off. This is similar to pushing the Enable button on the linear equalizer unit. Each operating mode has an independent state; entering a mode restores the enable/disable for that mode.

For example, assume following sequence of events:

1. Place the linear equalizer into the CDS mode and enable the output.
2. Switch to the 4-Tap mode and then disable the output.
3. Switch back to the CDS mode.

When you switch back to the CDS mode, the output will be enabled. Each mode (CDS in the above example) remembers the previous output-enabled setting and restores it when that mode is selected.

Output Amplitude control. The Output Amplitude control sets the peak-to-peak output in millivolts. The value is calibrated approximately, and within limits. Use the slider control to select a value between 0 and 6,000 mV (6 volts). The amplitude value is shown at the bottom of the slider control. If the requested value is not available, the output amplitude value shown at the bottom of the control will turn red.

Due to many factors, it is not possible to provide an accurate output amplitude over the wide range of possible tap settings (input signal characteristics that are typical in actual operation). While the output levels will be approximately correct in simple situations, it is not uncommon to see discrepancies between the setting and actual output amplitude levels. It is highly recommended to check the actual output amplitudes and adjust the slider to achieve intended output levels.

Refer to *Raw 9-Tap operating mode* for details on the output level control in Raw 9-Tap mode.

Input Offset control. Inputs of the linear equalizer are AC coupled and the input voltage offsets are normally removed automatically by the hardware. In special cases, (such as in the presence of asymmetric differential inputs), the signal quality can be improved by manually setting an offset.

When you click the **AUTO** button (under Input Offset mV), the button label changes to **MAN**and you can use the slider control to adjust the offset. The actual values shown at the bottom of the control are not calibrated; adjust the offset while monitoring the output signal for the best quality.

Save and Load controls. The settings in each mode can be saved and recalled as an XML file on the computer that runs the LE application software. Each mode produces a different state file which only contains the settings for the current mode.

When the state information is saved, it includes the channel number for which state was saved. However, the channel number is ignored during the recall (load) operation. It is permissible to load a state saved for Channel 1 into Channel 2.

State information saved for the LE160 can be recalled and loaded onto an LE320, but the results might not make sense. The linear equalizer software produces a warning when this occurs, but you must determine if the results were as intended. The following examples show one situation that makes sense and a second that does not make sense:

- **Situation 1.** Consider a state file that was saved in the 4-Tap mode with an LE160. If the emulated bit rate in the file overlaps with the available bit rates for an LE320, then the file can be successfully loaded onto an LE320 and it should produce the desired results. This works because the 4-Tap mode settings are inputs to an emulation package in the software which computes the appropriate hardware gains for the current connected hardware.
- **Situation 2.** Consider loading a state file saved in the 9-Tap mode with an LE160 onto an LE320. This will not produce the same results because the tap spacing in the two units is different (24 ps versus 18 ps). This state file contains lower-level information (the nine tap gains) and is not processed by the emulation package in the software.

To summarize, cross-loading state files into emulated modes (4-Tap, SPM, CDS) might work, but the results depend on the specific state data. In general, cross-loading state files into non-emulated modes (9-Tap and Raw 9-Tap) will not produce expected results.

AGC control. All control modes have adjustments for the automatic gain control (AGC). The linear equalizer automatically adjusts the input gain to maintain the internal signals at a fixed level.

The internal signals impact the signal quality. To improve or reduce the impact on the signal quality, it might be necessary to experiment with the AGC level setting.

The AGC level control provides direct control over the signal levels internal to the linear equalizer. The value is an integer between zero and seven, where lower numbers indicate lower internal signal levels. Use the following guidelines when adjusting the AGC levels:

- Lower levels (from zero to three) might be desirable for tasks such as receiver equalization where the desired frequency response is flat or high-pass in nature.
- Higher levels (between four and seven) work well when emulating channel impairments, which are essentially low-pass in nature.
- The maximum achievable output level is a function of the AGC level; lower levels will not be capable of as much output amplitude as higher levels.
- When troubleshooting signal quality issues, experimenting with the AGC level sometimes helps improving the signal quality.

AGC Lock control. The automatic gain control measures the input signal amplitude averaged over a period of time. The averaging time is usually long enough for good amplitude measurements of normal bit sequences. However, some signals might be at a fixed level for very long periods of time (such as, hundreds of microseconds or longer). In this case, the AGC circuitry in the linear equalizer will respond as if the input signal level had changed; it might be useful to lock the AGC function in a fixed gain mode.

When the AGC Lock check box is selected, the AGC Level control is replaced with an AGC Gain Adjust control. This is an uncalibrated gain adjustment and can be set between 1.8 and 2.5. This directly determines the input gain of the linear equalizer. Large numbers correspond to larger gains, but the actual gains are uncalibrated and do not vary linearly with the Gain Adjust value.

When the AGC is locked, the linear equalizer provides a fixed amount of gain based on the tap values and the output amplitude control. The actual output signal amplitude will be a function of the input signal level and is not adjustable in a calibrated fashion. As a reminder of this, the output amplitude label changes color and includes the designation **UNCAL** indicating that the output value is uncalibrated. The proper combination of the AGC Gain Adjust and Output Amplitude settings must be determined by trial and error.

There is an exception in the Raw 9-Tap mode because the output amplitude control directly sets the gain of the pre-driver block. In this case, the output level is already uncalibrated and the UNCAL notice does not appear. Refer to *Raw 9-Tap operating mode* for more information.

When the AGC is locked, the internal signal levels are determined by the input signal level and the AGC Gain Adjust settings. If the internal levels are too high, signal compression will result (this might or might not be desirable). If the internal levels are too low, then noise might become a problem.

Operating the linear equalizer with the AGC Lock enabled requires careful management of the signal levels to achieve good results. Use the following steps to manage the signal levels:

1. Using the AGC in the unlocked mode, find a combination of the input signal level and the AGC Level that works well. Use a "normal" bit sequence which does not contain long periods of zeros or ones for this purpose.
2. Enable the AGC Lock mode and set the gain adjustment to achieve the same output amplitude as in the previous step.
3. If the Gain Adjust is set ot maximum and the output level is still too low, the input signal level needs to be increased. Go back to the first step and try to find an acceptable setup using a higher input signal level.

Individual tap controls

The 4-Tap, 9-Tap, and Raw 9-Tap modes provide slider controls for adjusting the tap gain settings. The following figure shows the Tap controls in the 4-Tap mode.

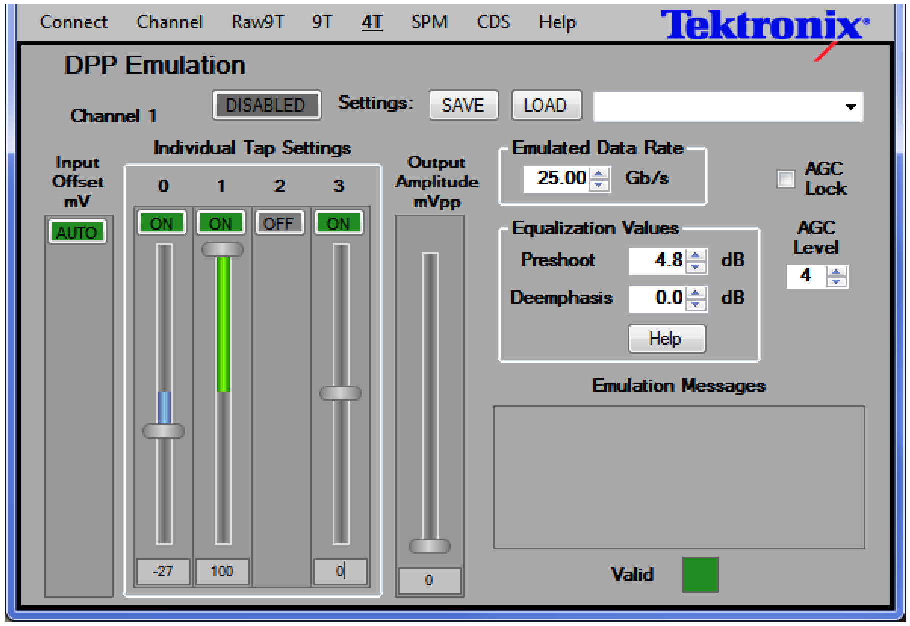


Figure 7: Tap controls in 4-Tap mode

The control for each tap has the following features:

- Tap gain readout and numeric entry
- Tap gain slider control
- Tap ON/OFF button

Tap gain readout and numeric entry. The actual tap values are shown at the bottom of the view, just below the slider controls. The frequency responses from all of the different components internal to the linear equalizer are combined and a set of hardware tap values are selected to best represent the desired aggregate time/ frequency domain response.

The gain settings run from -100 to +100 in the 4-Tap and 9-Tap modes; in the Raw 9-Tap mode the gain settings run from -63 to +63. The settings are integer values. Use the slider controls to set the gain values or enter the values directly into the readout.

Tap ON/OFF button. Each tap control has an ON/OFF button just above the slider control. If you set the tap control to OFF, the slider control and the readout disappear. The actual behavior in the OFF state depends on the operating mode. In 4-Tap mode, there is no difference between setting a tap value to zero and turning it off.

In the two 9-Tap modes, setting a tap to OFF disables the tap output in the linear equalizer hardware. Although not normally required, turning off unused taps can reduce noise levels in situations with low input signal amplitudes.

All emulated modes always enable all nine taps (ON). The only way to turn off the hardware taps in an emulated setup, is to switch to one of the 9-Tap modes and turn off the tap. However, switching back to the emulated mode will not preserve that change.

Frequency response graphs

In all operating modes, graphs of amplitude and phase versus frequency are shown for each channel. The data in the graph depends on the operating mode and is described under each individual operating mode. Here is a summary of the graphed data:

- In 4-Tap mode, the graph shows the response of an ideal 4-tap FIR with the current tap settings depicted.
- In the two 9-Tap modes, the actual estimated frequency response is shown.
- In the SPM and CDS modes, the target frequency response to be emulated is graphed.

In the 4-Tap mode the chart represents an approximation of the frequency domain response of the filter settings. For example, this can be useful when you manually set up a CTLE frequency response profile for the first time. The resultant taps can be saved for future use.

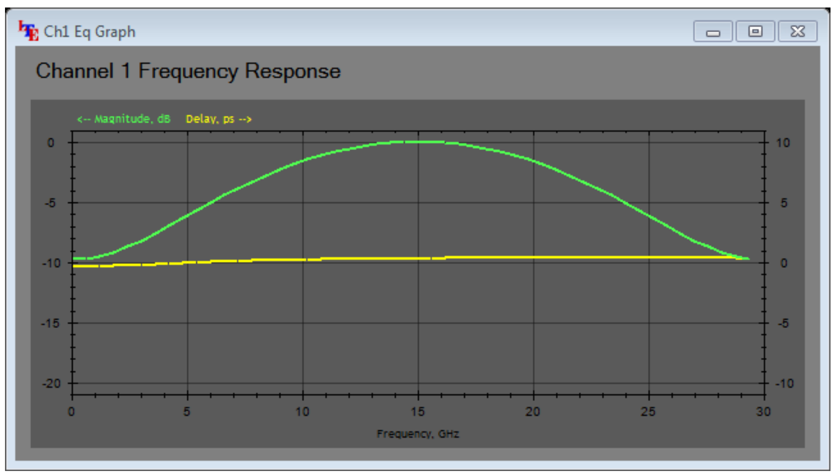


Figure 8: Amplitude and Phase vs. Frequency chart

4-Tap operating mode

The 4-Tap operating mode is standard with each linear equalizer; it does not require an optional license key.

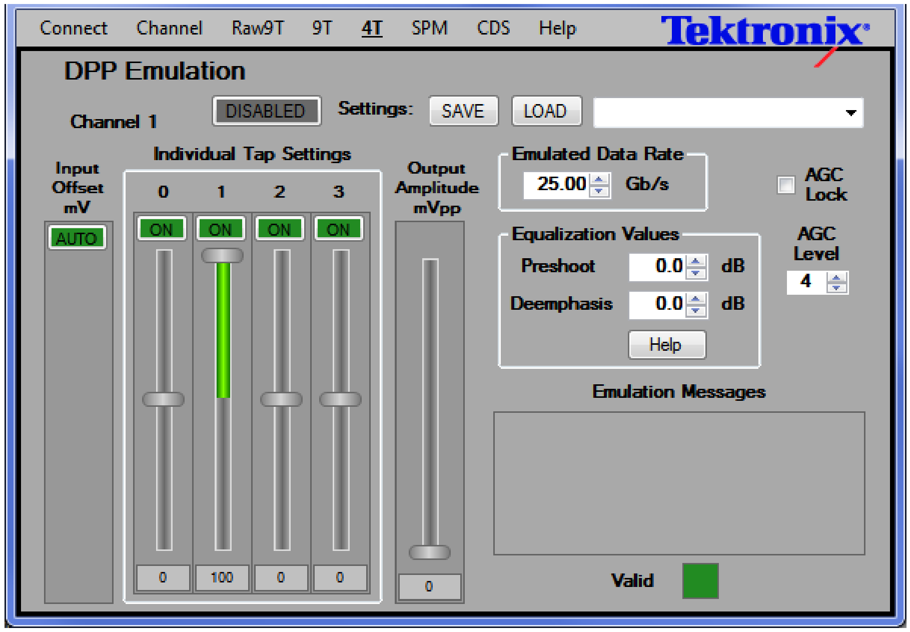


Figure 9: 4-Tap operating mode

In the 4-Tap mode, the linear equalizer emulates the behavior of a clocked BERTScope DDP unit. The frequency response graphs depict the response of an ideal 4-Tap FIR filter. Because the linear equalizer has limited bandwidth (to provide good signal quality), the actual resulting behavior has some high-frequency roll-off. This becomes apparent when switching from a 4-Tap setup to one of the 9-Tap modes where the actual expected frequency response is shown.

The following controls are unique to the 4-Tap operating mode:

- Emulated Data Rate
- Equalization Values (Preshoot and Deemphasis)
- Emulation Messages
- Valid indicator

Emulated Data Rate. To emulate a clocked DPP unit, a clock rate must be supplied. The ranges of available clock rates depend on the linear equalizer tap spacing, which is different for the LE160 and the LE320. Enter the data rate either directly or click the buttons to select the data rate.

Equalization Values. Instead of setting the four tap values one-by-one, use the Equalization Values controls to enter values for pre-shoot or de-emphasis. The software will compute the required tap settings. Pre-shoot values can only be positive and de-emphasis values must be negative.

Click the Help button to open a dialog window which explains the operation of pre-shoot and de-emphasis in more detail.

When you modify the tap values directly, the linear equalizer software attempts to compute the equivalent amounts of pre-shoot and de-emphasis that correspond to those tap values. There are many ways to set tap values that do not have valid equivalents and these are not error conditions. It merely indicates that pre-shoot or de-emphasis values are not defined for the tap setup.

Emulation Messages. The Emulation Messages window displays messages from the emulation package. Examples of information includes the following items:

- Tap settings which do not have valid equivalents in terms of pre-shoot or de-emphasis. This is information only, not an error condition.
- The emulation package was unable to create a hardware setup to match the current combination of emulated clock rate and tap values. Typically, this occurs when the spacing between first and last non-zero taps at the current emulation clock rate exceeds the total hardware tap delay range.

Valid indicator. The Valid indicator will be green when the current settings have been successfully emulated and loaded into the hardware. Tap setting which do not have valid pre-shoot or de-emphasis equivalents can still be successfully emulated and will not cause the Valid indicator to be red.

9-Tap operating mode

The 9-Tap operating mode provides direct control of all nine tap gains. Set the tap gains on an integer scale from -100 to +100. The linear equalizer software automatically computes the linearity corrections and makes the necessary hardware tap settings to achieve the specified amount of gain. The 9-T operating mode is standard with each linear equalizer; it does not require an optional license key.

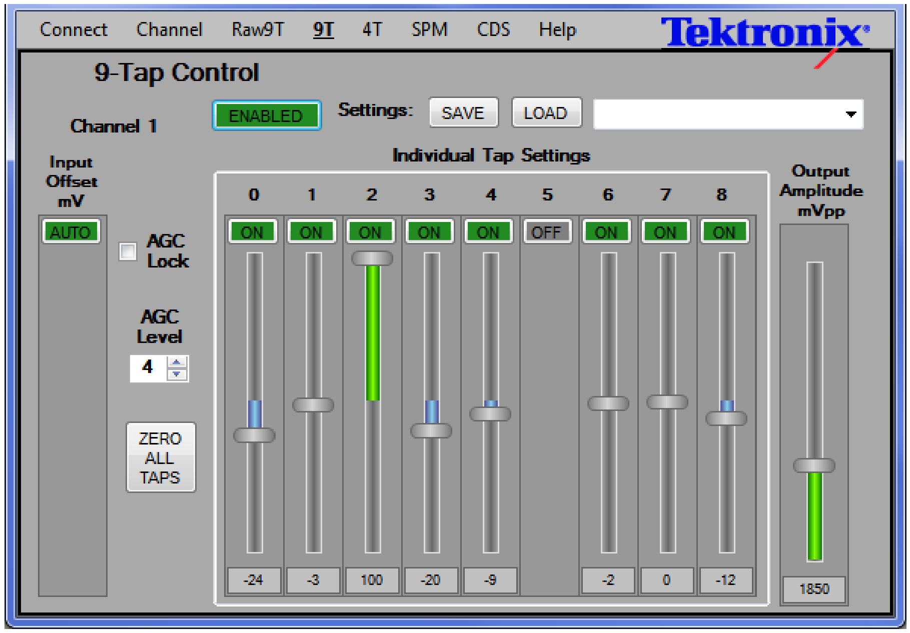


Figure 10: Controls available in 9-Tap mode

The signal path through the linear equalizer is not perfect from either a time or frequency domain perspective. Refer to the discussion under *Block diagram* for more information on this.

As with all electronic hardware of this nature, the linear equalizer is inherently low-pass and has a bandwidth sufficient for digital signals as fast as 32 Gb/s. In the 9-Tap modes, no attempt is made to compensate for the non-ideal nature of the hardware and this is reflected in the frequency response graphs corresponding to various tap gain settings. Although linearity corrections for tap gains are provided in this operating mode, you will still see deviations from the frequency and time domain responses of a theoretically ideal 9-tap linear FIR filter.

Raw 9-Tap operating mode

In the Raw 9-Tap operating mode, direct control of the hardware settings for the tap gain and the output amplitude input offsets is provided. No corrections of any type are provided for this mode. The Raw 9-T operating mode is standard with the linear equalizer; it does not require an optional license key.

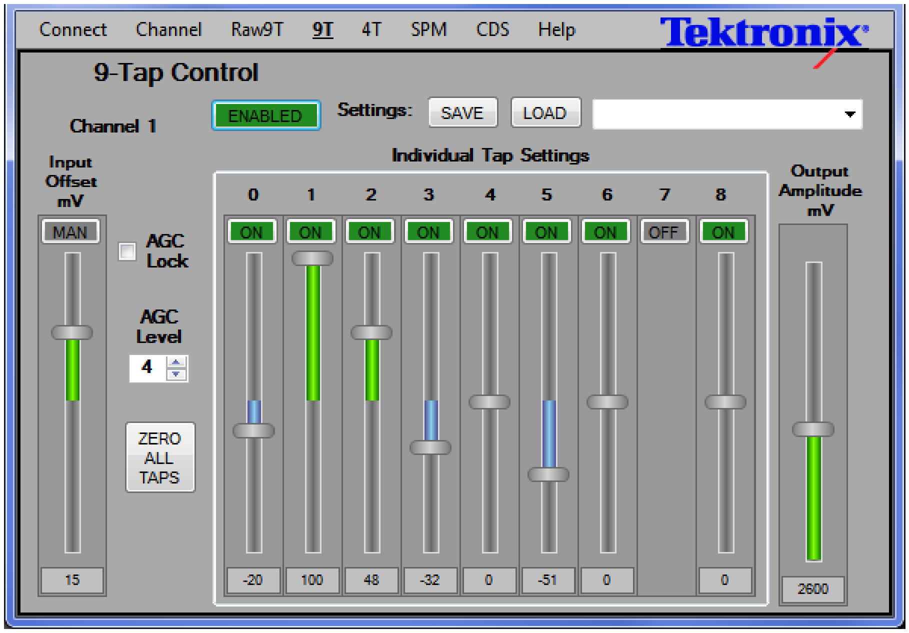


Figure 11: Raw 9-Tap operating mode

Instead of setting the output amplitude, the gain value for the pre-driver block is set directly. See the *Block diagram* for information on the pre-driver block. The pre-driver setting is an integer between zero and 63; higher values correspond to higher gains, but the numbers are otherwise uncalibrated.

The tap gain numbers are integers between -63 and +63 in this mode. Larger numbers indicate larger gains; but they are not linearly related to the actual gain and no corrections for this non-linear behavior are applied in this mode.

SPM - S-Parameter Modeler operating mode

The SPM (S-Parameter Modeler) operating mode provides a means to input a desired filter shape using .s2p, .sp3, or .s4p files. This view is available with Option SPM.

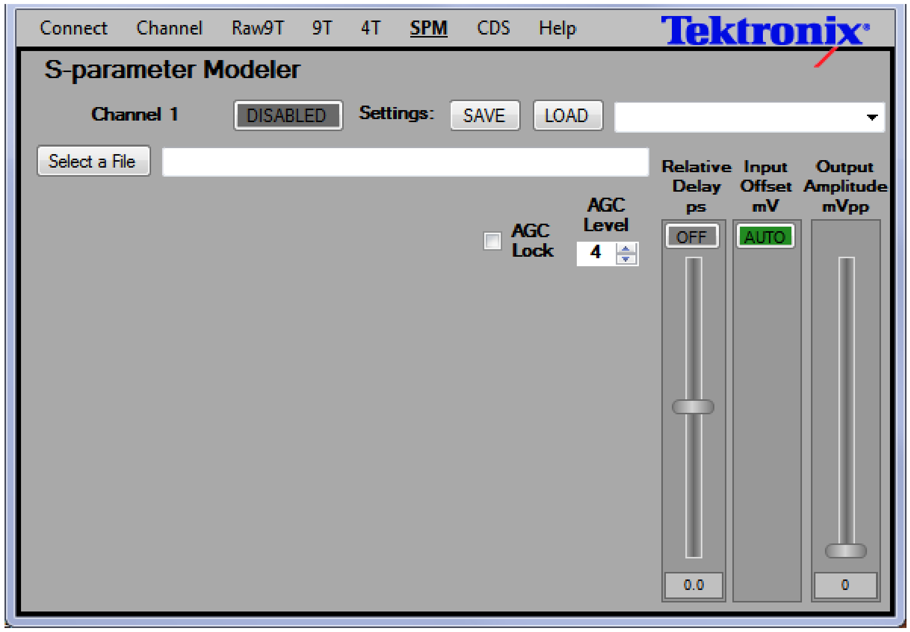


Figure 12: Default SPM operating mode

The .s2p, .sp3, and .s4p files can be loaded from the default location set in the Save State Files in field in the Connect view. Click **Select a File** to open a Windows Explorer window to navigate to the files. When you select a valid file, additional controls are added to the SPM view. If you select an invalid file, an error message displays below the button and might include helpful information explaining why the file was invalid.

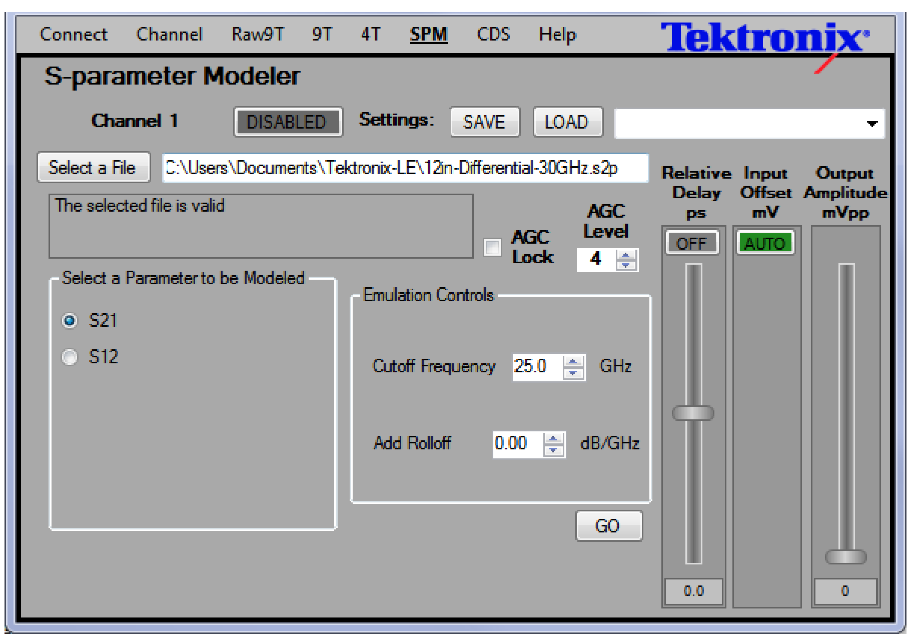


Figure 13: SPM view with a valid .s2p file

Selecting parameters. Select an S-parameters from the list on the left side of screen. In the preceding illustration, an SP2 file was loaded and the S21 (forward) and S12 (reverse) parameters are available. The S11 and S22 parameters in the file do not represent transfer functions and are not included in the list. If you load an S3P or S4P file, a longer list of parameters is available.

To get reasonable results, the S-parameter files should cover a range from a low frequency (150 MHz or less) up to at least 150% of the data transfer rate. For example, if you are working with digital signals at 20 Gb/s, the S-parameter data should be valid up to at least 30 GHz. This should be considered an absolute minimum and data to even higher frequencies (up to a maximum of 50 GHz) is recommended. The frequency sample points must be spaced no more than 150 MHz apart.

Cutoff frequency. To provide a good emulation result, it is necessary to specify a cutoff frequency. The cutoff frequency sets the high-end of the frequency range over which the linear equalizer will attempt to accurately model the specified transfer function. The adjustment limits for the cutoff frequency are automatically determined.

Consider an S-parameter file that shows a flat response from DC up to 50 GHz. It is not possible to emulate this accurately with the linear equalizer because it does not have the necessary bandwidth. In this case, a low cutoff frequency needs to be specified to get a target response that is within the emulation capabilities of the linear equalizer.

Alternately, if the S-parameter data contains a lot of high frequency roll-off, then the cutoff frequency might be set to a high value because the linear equalizer is capable of emulating the target response without further modification.

If the S-parameter data was abruptly truncated at the cutoff frequency, it would create unacceptable distortions in the time-domain behavior (such as, over-shoot and ringing in the step response). Instead, the equivalent of a high-order, linear phase low-pass filter is used to remove data above the cutoff frequency; this preserves, as much as possible, the time-domain behavior represented by the S-parameter data.

Add roll-off. The Add Rolloff control has two possible uses. For some situations, a better simulation result is obtained if the S-parameter data is linearly rolled-off in magnitude starting at zero frequency. Non-zero values achieve this effect. This is done to preserve the phase response in the S-parameter data. It is also possible to enter negative numbers, which creates a roll-up in magnitude at frequencies above the cutoff.

A second use of the roll-off setting is to emulate linear-phase impairments with a linear slope (linear in dB). This requires loading a flat S-parameter first and it might be more convenient to use the roll-off setting in the CDS mode for this purpose.

GO button. Click the GO button to have the linear equalizer calculate a set of values for the nine hardware taps based on the S-parameter data, as modified by the emulation control settings. This updates the frequency response graph to include any additional roll-off (the cutoff frequency is not depicted in the graph).

After the calculations are completed, status information displays at the bottom of the screen. The status information indicates success or displays any problems that might have occurred.

At this point the resulting hardware tap settings can be viewed by switching to one of the 9-Tap modes. Adjust the emulation controls in the SPM mode or make small changes to the tap gains in one of the 9-Tap modes to "dial-in" a desired emulation. Any adjustments made in the 9-Tap modes will be lost when you switch back to the SPM mode. Therefore, it is a good idea to save the state in the 9-Tap mode before going back to the SPM mode.

CDS-Channel Designer operating mode

The CDS-Channel Designer (CDS) operating mode provides a means to easily input a desired response in pole-zero format. This is similar to Option SPM; the linear equalizer calculates the best settings to emulate the desired response. This mode is available with Option CDS.

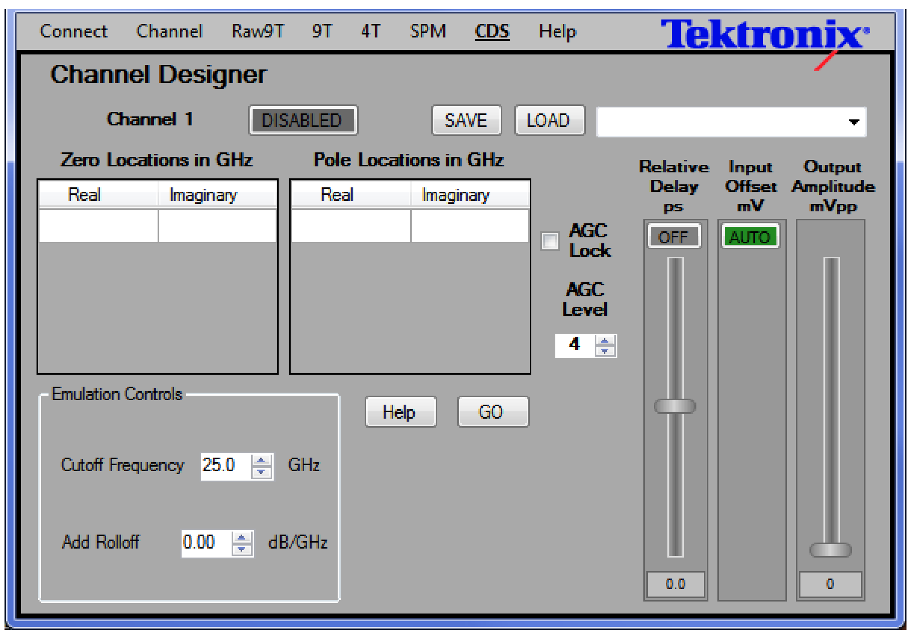


Figure 14: CDS operating mode

This view provides two tables to enter the zero and pole locations. Click the Help button in the view to see helpful tips on inputting data.

Zero Locations in GHz. This table lists the zeros to be modeled. If the zero is real (with no imaginary component), enter zero in the Imaginary column or leave it blank. Zeros with non-zero imaginary components are always applied in conjugate pairs. The software automatically manages the conjugate pairs and only one of the two members of the pair (the one with a positive imaginary component) should be entered into the table. Negative imaginary values are automatically changed to positive, which represents a complex conjugate pair of zeros.

If you desire a pair of real zeros at a single location, make two entries in the table; both with the same real value and zero for the imaginary components.

The real components of zeros can be positive or negative.

If you enter a numeric value in the last row, the software automatically adds a new blank row. Delete an existing row or rows by selecting one or more table entries in the row or rows and pressing the Delete key on your keyboard. Values must be entered in units of GHz. If you have a specification of zero locations in radians/sec, divide the value by ($2\pi \times 10^9$) before entering the information into the table.

Pole Locations in GHz. This table lists the system poles to be modeled and is similar to the Zero Locations table. The main difference is that all poles entered must have non-positive real components. Real components cannot be zero; if a pole is to be located on the imaginary axis, enter a small negative value for the real component.

Emulation controls. The emulation controls (cutoff frequency and Add Rolloff controls) function the same as they do in the SPM mode. Refer to *SPM - S-Parameter Modeler operating mode* for more information on these controls.

GO button. Click the Go button to have the linear equalizer calculate a set of values for the nine hardware taps based on the entered pole and zero locations, as modified by the emulation control settings. A new status area appears at the bottom of the screen. This also updates the frequency response graph which includes the added roll-off (the cutoff frequency is not depicted in the graph).

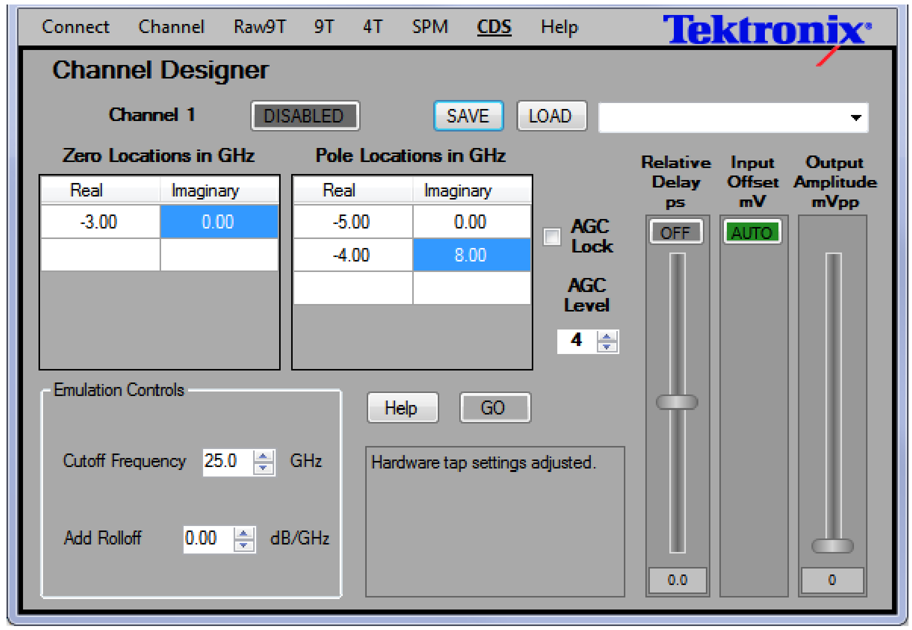


Figure 15: CDS controls after clicking the GO button

The previous image shows the poles and zeros entered for the following transfer function:

$$H(s) = \frac{(s + 3 \times 2\pi \times 10^9)}{(s + 5 \times 2\pi \times 10^9) \times (s + (4 + j8)) \times (2\pi \times 10^9) \times (s + (4 - j8) \times 2\pi \, x 10^9)}$$

As shown in the equation, each pole and zero in the data entry tables is automatically multiplied by (2π x 10^9) in the software to convert from units of GHz to radians-per-second. The software also created a conjugate pole pair from the second entry in the pole table because it has a non-zero imaginary part.

At this point, the resulting hardware tap settings can be viewed by switching to one of the 9-Tap modes. Small adjustments to the pole and zero locations can be made in the CDS mode or to the tap settings in the 9-Tap modes to directly "dial-in" a desired emulation. Be sure to save a state in the 9-Tap mode before switching back to the CDS mode because the changes will be lost when switching back to the CDS mode.

Help button. Click the Help button to open a dialog window containing useful tips for entering pole and zero locations.

Specifications

Refer to the LE160 & LE320 Linear Equalizer Product Specification and Performance Verification Technical Reference manual (Tektronix part number, 077-0884-xx) for the list of product specifications and procedures to verify those specifications.

Maintenance

User service information

This section provides high-level service information and procedures for your instrument.

Service offerings

Tektronix provides service to cover repair under warranty as well as other services that are designed to meet your specific service needs.

Whether providing warranty repair service or any of the other services listed below, Tektronix service technicians are well equipped to service your instrument. Services are provided at Tektronix Service Centers and on-site at your facility, depending on your location.

Warranty repair service

Tektronix warrants this product as described in the warranty statements at the front of this manual. Tektronix technicians provide warranty service at most Tektronix service locations worldwide. The Tektronix product catalog lists all service locations worldwide.

Calibration and repair service

In addition to warranty repair, Tektronix Service offers calibration and other services that provide cost-effective solutions to your service needs and quality standards compliance requirements. Our instruments are supported worldwide by the leading-edge design, manufacturing, and service resources of Tektronix to provide the best possible service.

Inspection and cleaning

General care

Protect the instrument from adverse weather conditions. The instrument is not waterproof.

CAUTION. *To avoid damage to the instrument, do not expose it to sprays, liquids, or solvents.*

Preventive maintenance

Preventive maintenance mainly consists of periodic cleaning. Periodic cleaning reduces instrument breakdown and increases reliability. Clean the instrument as needed, based on the operating environment. Dirty conditions may require more frequent cleaning than computer room conditions.

Exterior cleaning. Clean the exterior surfaces with a dry, lint-free cloth or a soft-bristle brush. If dirt remains, use a cloth or swab dampened with a 75% isopropyl alcohol solution. A swab is useful for cleaning in narrow spaces around the controls and connectors. Do not use abrasive compounds on any part of the instrument.

Cleaning guidelines. To avoid damaging the instrument follow these precautions:

- Avoid getting moisture inside the instrument during external cleaning and use only enough solution to dampen the cloth or swab.
- Use only deionized water when cleaning. Use a 75% isopropyl alcohol solution as a cleanser and rinse with deionized water.
- Do not use chemical cleaning agents; they may damage the instrument. Avoid chemicals that contain benzene, toluene, xylene, acetone, or similar solvents.

Repack the instrument for shipment

If the instrument is to be shipped to a Tektronix service center for repair, attach a tag showing the following information:

- Name of the product owner
- Address of the owner
- Instrument serial number
- A description of the problems encountered and/or service required.

When packing an instrument for shipment, use the original packaging. If it is unavailable or not fit for use, contact your Tektronix representative to obtain new packaging.

Troubleshooting information

This section provides some high-level procedures in the event of problems with your instrument.

Check for common problems

Use the following table to quickly isolate possible failures. The table lists problems related to the instrument and possible causes. The list is not exhaustive, but it may help you eliminate a problem that is quick to fix, such as a loose cable.

Table 3: Failure symptoms and possible causes

Symptom	Possible causes and recommended action
Instrument does not turn on	Verify that the power adapter is connected to the instrument and to the power source. Check that the instrument receives power when you press the Power button. Check that power is available at the power source. Instrument failure: contact your local Tektronix service center.
Signals do not pass through the instrument to the DUT	Verify that input and output cables are securely attached to the linear equalizer and to the DUT. Verify that the Power LED is on when you push the Power button on the linear equalizer. Verify that the Enable LED is on when you push the Enable button on the linear equalizer. Instrument failure: contact your local Tektronix service center.
Wrong signals from the instrument	Verify that the cables are correctly connected to the linear equalizer and to the DUT.
Linear equalizer application does not appear on the host PC.	Verify that the linear equalizer software is installed on the host PC. Verify that the linear equalizer software is running on the host PC. Verify that the linear equalizer is connected to the host PC via the USB cable.
LE software cannot connect to the linear equalizer.	Verify that the Power LED is turned on. Verify that the USB 2.0 cable is properly connected to the linear equalizer and to the host PC.

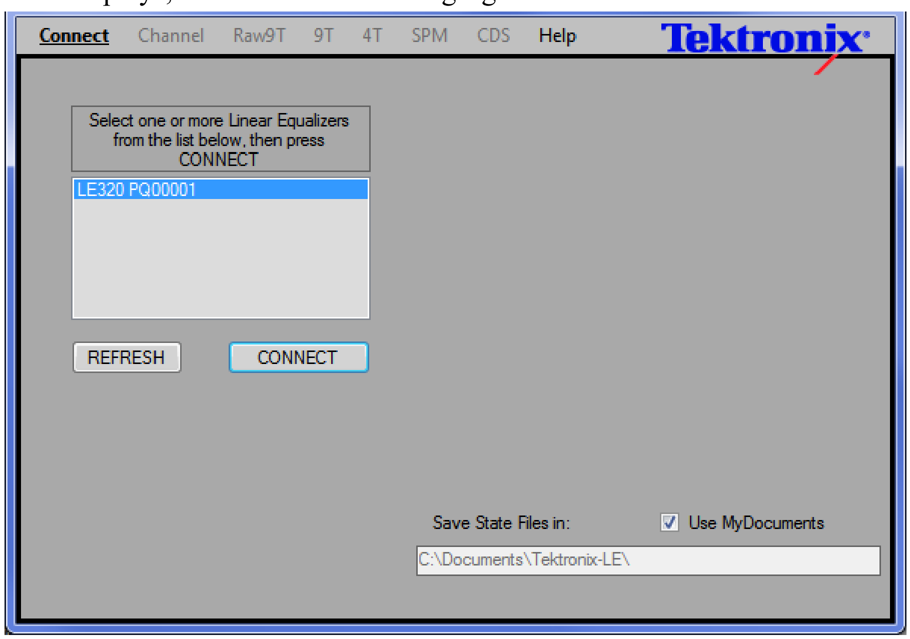
Appendix A - LE160/LE320 software license upgrade information

The LE160/LE320 software features, available at time of manufacture, are controlled by a software license key which is generated by Tektronix during the manufacturing process. To add software features to your product, you can purchase an upgrade from Tektronix. This will result in a new license certificate which contains a new license key. This appendix contains procedures to enter this new key to your product.

Upgrade the software license on the instrument

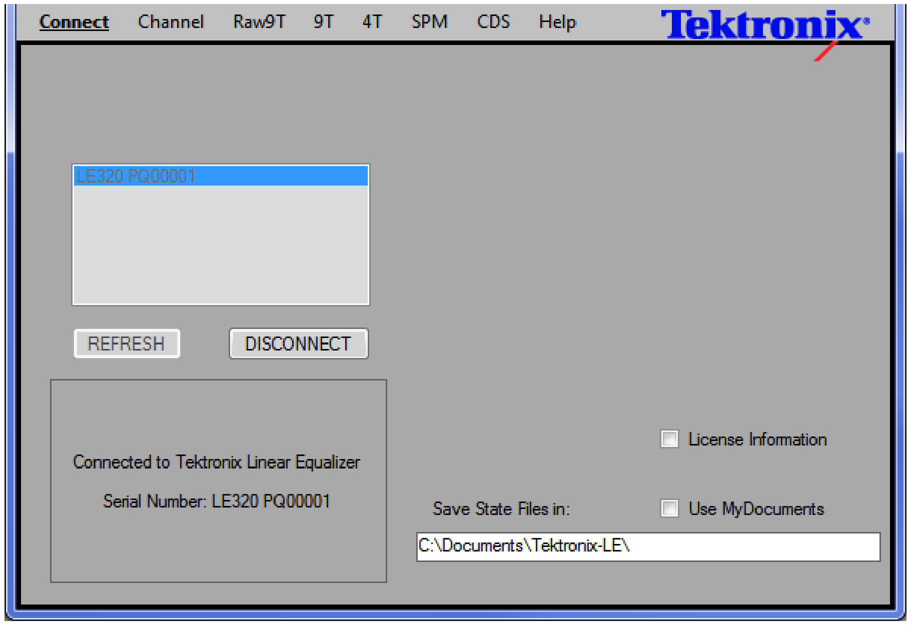
1. Connect the linear equalizer to the power supply and to the host PC.
2. Power on the linear equalizer and start the LE application.

   After the Splash screen appears and the hardware initializes, the Connect view displays, similar to the following figure.

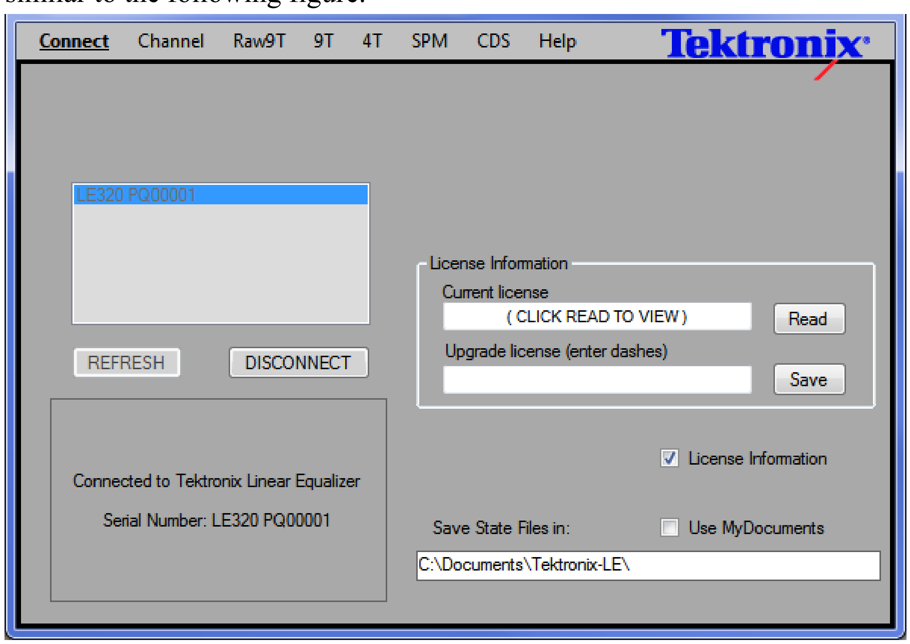


3. Click the **Connect** button to connect the software to your instrument.

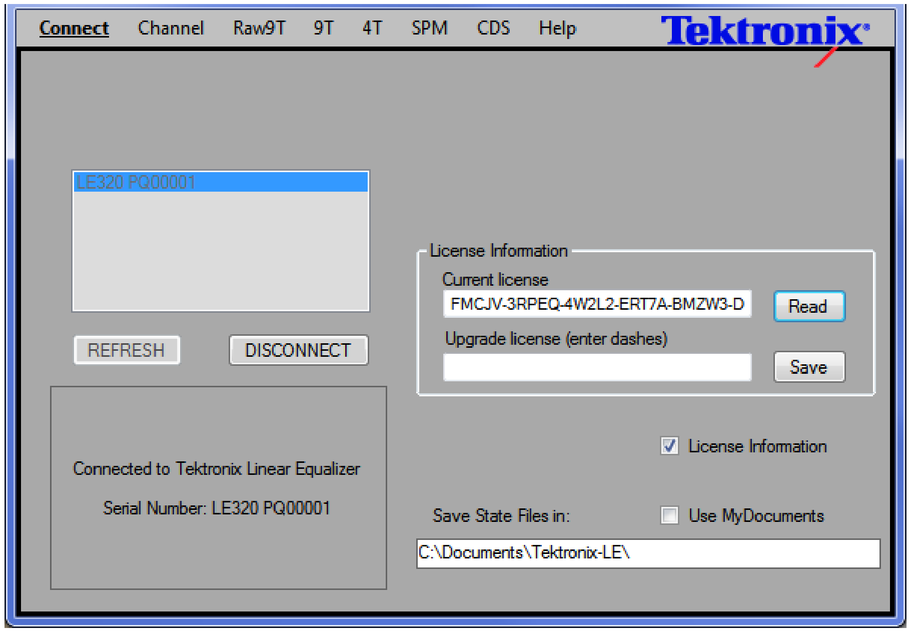
You should see the serial number of your instrument in the list of connected devices.



4. Verify that the product serial number is the same as your instrument.
5. Select the License Information check box to display the License information, similar to the following figure.



6. Click **Read** to obtain the current software license key.

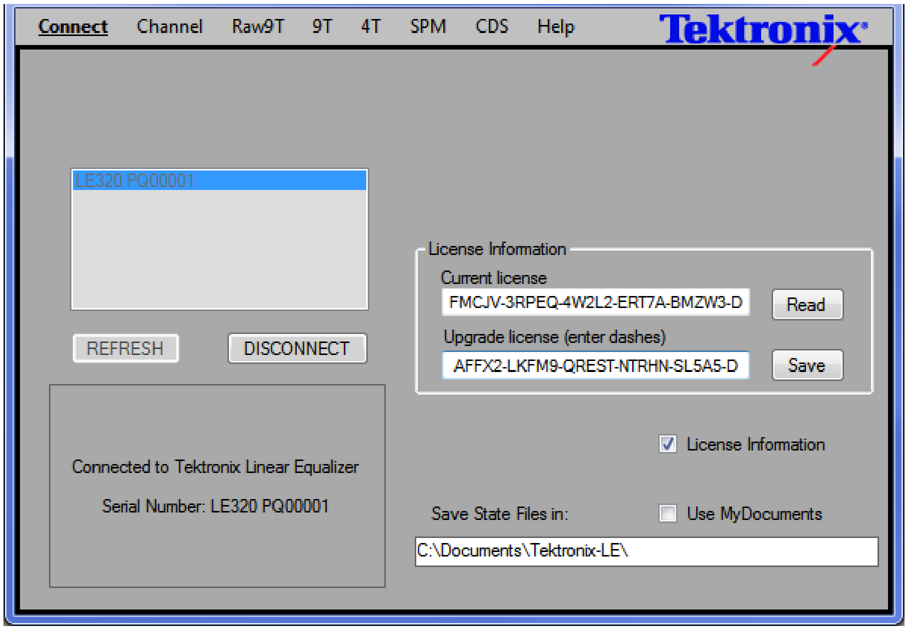


7. Refer to your upgrade license certificate; it should look similar to the following figure.

The upgrade license certificate is called an *Option Key*. It lists the features enabled or authorized by the Option Key.

8. Enter the Option Key number exactly as printed on the certificate (including dashes) into the Upgrade license field in the Connect view.

The filled out information should look similar to the following figure.



9. Click **Save** to save the upgrade information.

The application records the upgrade key and then clears the License information field.

CAUTION. *Errors can occur if you remove the power from the instrument during the upgrade process. Do not remove power from the instrument during the upgrade process.*

If any errors occur during the upgrade process, turn the power off and back on again. Then repeat the above steps. If the error persists, record the error message and contact your local Tektronix Service representative.

Verify the license information for the upgrade

Perform the following procedure to verify that the upgrade was successful and matches the information on the upgrade certificate.

1. Click **Refresh** to refresh the information in the Control view after you performed the upgrade procedure.
2. Select the License Information check box.
3. Click **Read** to read the current license information installed on your instrument.
4. Verify that the current license displayed in the Control view is identical to the License key on your upgrade certificate.
5. Click the selections at the top of the Control view for the newly authorized features (such as SPM or CDS).

   The new views should display.

   If you experience any problems, verify that you correctly entered the upgrade information. If you experience any problems, contact your local Tektronix representative for further action.

Appendix B - Theory of operation

The linear equalizer is a continuous time linear equalizer (CTLE) intended for processing digital signals at data rates as high as 32 Gb/s. It is a linear device; it does not re-clock the input data. Because of this, the time-domain responses will differ from a clocked system.

The linear equalizers are intended to compensate for, or simulate degradation in signal step response resulting from a cable fault or a long backplane. The degradation typically results in a step with the first few bits attenuated, which then rises asymptotically to the final value. To compensate for this effect, the linear equalizer increases the relative amplitude, or pre-emphasizes, the first few bits of the step. To simulate the signal degradation, the first few bits of the step are decreased in relative amplitude, or de-emphasized. The linear equalizer contains a Multi-Tap Linear Analog filter that allows modifying the bits of the signal in either way. The frequency response graph indicates how the linear equalizer affects the bit-stream, as a function of frequency.

Block diagram

The following diagram provides information on the operation of the linear equalizer.

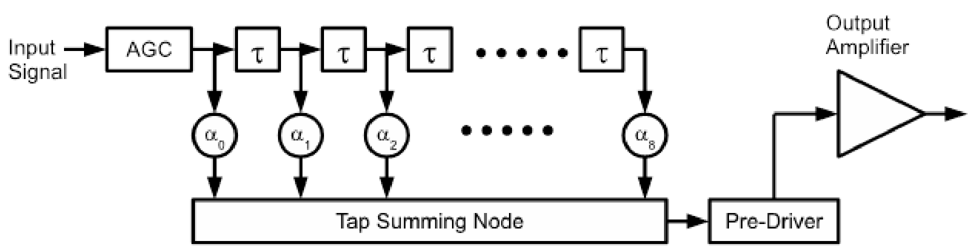


Figure 16: Simplified linear equalizer block diagram

The linear equalizer contains a chain of eight fixed delays (18 ps for the LE320 and 24 ps for the LE160) which provides nine copies of the input signal. Each copy is delayed by the same amount from the previous one.

The delay chain is driven by an automatic gain control (AGC) circuit which maintains internal signal levels at a constant level. The AGC block contains a variable gain amplifier that automatically adjusts the gain based on the input signal level to achieve the desired internal signal level.

Each of the nine input signal copies is applied to a variable gain block (labeled α_0, α_1, . . . α_8 in the block diagram). Collectively, these are called the tap gains. The outputs of all of the tap gain blocks are summed together to produce the output signal. A variable gain pre-driver block is included to allow adjustments of the output signal amplitude.

Output amplitude

The output signal level depends on the following things:

- The signal level set by the AGC circuit
- The value of each of the nine tap gain blocks
- The gain chosen for the pre-driver block
- Input signal characteristics such as frequency content, pre-emphasis, and impairments

The software applications used to control the linear equalizer provide a calibrated output level control, but the accuracy has limitations. The software does not take input characteristics into account. The output setting is only an approximation and you may need to take measurements and make adjustments to achieve the desired output level.

Operating modes

Depending on the licenses installed on the linear equalizer the application provides different operating modes to set the tap values:

- 4-Tap (4T) mode. This mode provides a simple interface to the tap settings. Although the linear equalizer is not a clocked data system, this mode emulates clocked BERTScope DPP behavior with four taps. Pre-shoot and de-emphasis inputs provide an optional means for the tap values in this mode.
- Raw 9-Tap (Raw9T) mode. This mode allows direct control of the hardware tap values. No corrections are provided for tap gain linearity and other non-ideal behaviors. The output amplitude is only set in a relative fashion and is uncalibrated. All other modes provide calibrated output levels.
- 9-Tap (9T) mode. This mode accepts linear gain settings for each tap (on a scale from zero to 100). Linearity corrections are internally applied to the hardware tap values to achieve the requested relative gain for each tap.

  NOTE. *9T mode provides finer control over tap settings than a standard 4T linear equalizer. You can set the tabs in the 4T mode as a starting point and then fine-tune the frequency responses directly in the 9T mode.*

- S-parameter (SPM) mode. This mode reads S-parameter data files (.s2p, .s3p, and .s4p) configures the nine tap settings to best emulate a transfer function in the file (such as, S21).
- Channel designer (CDS) mode. This mode accepts the specification of frequency response through pole and zero locations in the complex plane.

The operating modes are independent and each has a separate memory of its state. Settings such as, the output amplitude and output enabled are remembered on a mode-by-mode basis.

The current operating mode for a selected channel is always shown in underlined, bold-face type at the top of the view. The selected channel is shown at the top left side of the view; if multiple linear equalizers are connected, the channel information includes the serial number. The mode setting for each channel is independent. For example Channel 1 can be in 4-Tap mode while Channel 2 might be in CDS mode. Once taps are set for a particular channel, they will persist until changed, the power is cycled, or the software is disconnected and then reconnected. A software disconnect, by itself, does not change that hardware settings.

One or two secondary window show the Amplitude and Phase vs. Frequency Response per channel. The views represent an approximation of the frequency domain response of the filter settings in 4-Tap mode.

Emulated versus non-emulated modes

There is an important distinction between the two 9-Tap modes and all other modes.

Both 9-Tap modes allow direct control of the nine internal tap values; these are known as *non-emulated modes*. The normal 9-Tap mode corrects tap settings for non-linearity and provides calibrated output amplitudes; but no other corrections are performed. For example, each the frequency and time-domain step responses for each tap are not ideal and these modes make no attempt to compensate for this. The Raw 9-Tap mode does not provide linearity corrections or calibrated output amplitude settings.

The remaining modes (4-Tap, SPM, and CDS) are *emulated modes*. In these modes, the desired linear equalizer behavior is specified through either theoretically ideal tap settings (4-Tap mode) or by an ideal targeted frequency response (SPM, CDS modes). In these modes, the linear equalizer software runs an internal emulation package to determine a set of values for the nine hardware taps that will most closely match the specified behavior.

In 4-Tap mode, the emulation calculations are fast and the nine tap values are computed in real time as the four tap values are adjusted. The other emulated modes require more time to make calculations; they contain a GO button that must be activated (clicked) after specified behavior is specified or changed.

Mode transitions

The linear equalizer software preserves the hardware state during certain mode transitions. This behavior supports making small adjustments to fine-tune the setups created in emulated modes. The following illustration shows the transitions for which hardware state is maintained.

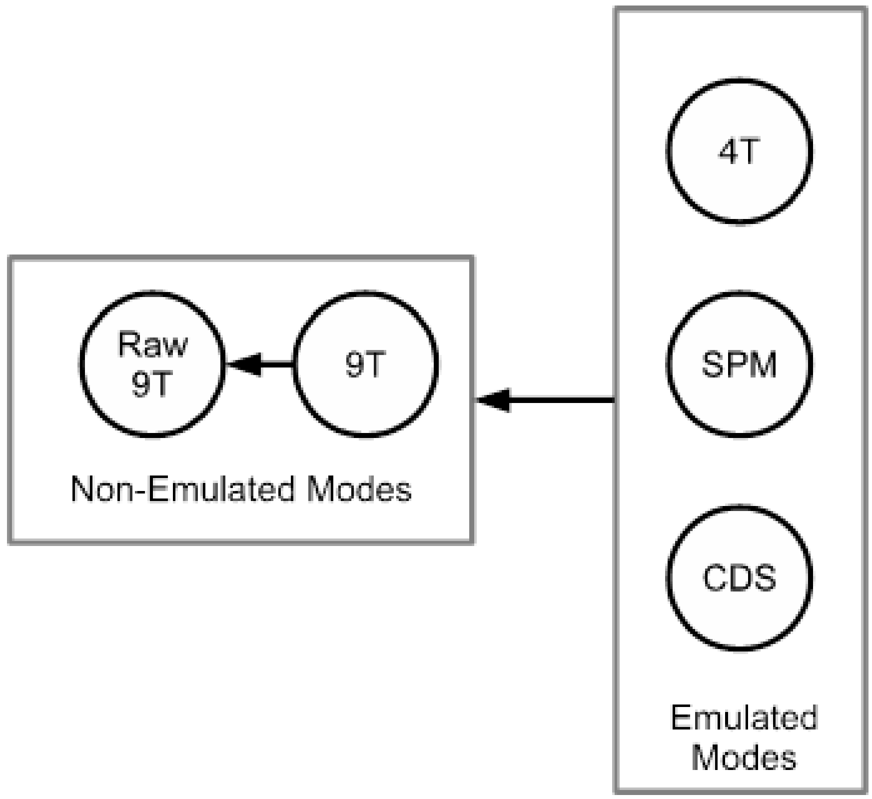


Figure 17: Mode transitions that preserve the hardware state

The hardware state is maintained for the following transitions:

- From any emulated mode into any non emulated mode
- From 9-Tap mode into Raw 9-Tap mode

For all other mode transitions, the hardware state is not preserved and can be altered when entering the new mode.

Fine-tuning the emulated results

After generating a configuration in one of the emulated modes, switching to the 9-Tap (or Raw 9-Tap) mode will not alter the hardware tap gains. This can be useful for making fine adjustments to the emulated settings.

These adjustments will often involve moving one or two of the end taps by just one or two integer steps. When a good setup is achieved by fine tuning, it is a good idea to save the state before leaving the 9-Tap mode in which the tuning was done.

Changing from the 9-Tap modes back into an emulated mode will re-load the nine tap values with calculated values; this will overwrite any fine tuning that may have been done.

Appendix - C Remote control guidelines and commands

The Tektronix Linear Equalizer (LE) programmable interface is a COM API which was registered by the installation process. For convenience, the COM API type library is also available in the installation directory.

COM has been selected because it supports a broad range of programming languages and environments, such as LabView, C++, Python, Perl, VB, VB.NET, PowerShell.

This appendix provides general guidance, characteristics and requirements for correct programmatic behavior, and then describes all software commands for controlling the LE160 and LE320 instruments.

This version of the COM interface does not support remote COM, DCOM or COM+.

General guidance

The linear equalizer hardware imposes some non-trivial limitations or requirements on the order of some commands. For example:

- The command to turn on power must be given before most other commands will be allowed. The connect group commands are an exception to this rule.
- An enable-command must be given before any signal will be applied to any output.

These program requirements are imposed because the linear equalizer contains powerful amplifiers capable of damaging sensitive attached devices. These two commands perform the same functions as the front-panel buttons, POWER and ENABLE. During manual operation, you must push both the POWER button, and the ENABLE button, before a signal will appear on any output. The POWER button controls the DC voltage to the output power amplifiers. The ENABLE button controls the enable pins on each channel of the LE chip.

A second major order dependency deals with the internal state of the linear equalizer. The state is the set of memory variables required to accurately create or re-create a given equalization profile. There is a separate state vector for each of the five operating modes (4-Tap, 9-Tap, Raw 9 Tap, SPM, and CDS). The limitation is that from a given mode, only commands for that mode are allowed; off-mode commands will generate an error.

A side effect of having mode-specific state vectors is that most settings are not common between modes. For example, if you set the output amplitude to 1.0 V in SPM mode, and then switch to 4-Tap mode, the output voltage will change to the value that you specified for the 4-Tap mode, or its default, which is zero. There are exceptions to this general rule under *Appendix B: Theory of operation, Mode transitions*. The COM API supports load and save commands to make it convenient to easily set up conditions for each mode.

A third order characteristic is that all linear equalizer commands are blocking (not overlapped). No new command will be accepted until the previous command has completed.

Most commands execute in a few milliseconds, but when first connected, the linear equalizer must read a large calibration file from the embedded controller. This read operation takes a few seconds. This start-up delay means that, for fastest operation, the programmer should bring up the COM interface, and leave it up for the duration of all control operations, so that only the first connect command suffers the cal-file-read delay. (Refer to the program structure in step 4 below.)

The above recommendations can be summarized by the following recommended structure of a control program:

1. Initialize the COM interface.
2. Query for the number of linear equalizers attached to the USB bus.
3. Turn on the power to one of the attached linear equalizers.
4. Connect to one of the attached linear equalizers (the cal-read delay occurs here).
5. Set the operating mode.
6. Make mode-specific changes.
7. Enable the output(s); repeat as needed for other outputs.
8. Set the unit to another operating mode.
9. Make any mode-specific changes; repeat as needed.
10. Disable the outputs.
11. Disconnect from the unit.
12. Release the COM interface.

A sample program has been provided, located in the installation directory, in a subfolder named "remote control examples." The program is written in C++ and follows the above guidelines. While it only implements about 15 commands, it follows the general principles and is richly commented for pedagogical purposes. It is provided as a compressed archive of a Visual Studio 2010 project. Please note that this program is copyright protected by Tektronix and is provided for informational purposes only.

The executable of the example C++ program can be found in the installation directory.

Run the program (with a linear equalizer attached to the installation computer) to verify correct type library registration and operation of the COM API. If successful, the program produces, in Windows command shell, the results of approximately 30 individual tests of the COM API. Refer to the README.TXT document provided with the example program, for explanations of each test and the expected results.

There is a strong relationship between the COM API commands and the linear equalizer graphical application (GUI) behavior. For this reason, the remote control commands are grouped by the operating modes (command categories). The operating modes correspond directly to equivalent screens or dialogs of the linear equalizer GUI application.

Each command category contains a table listing the commands, including input and output information. Click the command name to quickly go to the actual command description. Each COM command, for a given operating mode, is designed to allow you to achieve programmatically what can be done, in that mode, in the user interface.

Command reference overview

This overview provides useful information for using the remote control commands.

Commands and queries

Commands consist of set commands and query commands (usually called commands and queries). Commands modify instrument settings or tell the instrument to perform a specific action. Queries cause the instrument to return data and status information.

All commands return true on success and false on failure. All commands also return an error number, and error string. On failures, use the error number and string to find out what went wrong. Do not use any other returned values if the command failed.

NOTE. *Please note that because all commands always return a pass-fail or Boolean result (returned by value), as well as a 32-bit error number and an error string (returned by reference), these values or arguments are generally not listed, and their correct usage is not discussed in the reference section which follow.*

It is essential for correct operation, that you check the pass-fail result of every command. You must not trust the results of any command which returns false. If you experience a command failure, use the error number and error string to help diagnose why the command failed.

Most commands have both a set form and a query form. The query form of the command differs from the set form by a "Q" on the end. Not all commands have both a set and a query form. Some commands have only set and some have only query.

Syntax

Refer to the following table for the symbols that are used with the remote control commands.

Table 4: Syntax symbols and their meanings

Symbol	Meaning
< >	Defined element
::=	Is defined as
|	Exclusive OR
{ }	Group, at least one element is required
[ ]	Optional, can be omitted
...	Previous elements can be repeated
( )	Comment

Parameter types

Parameters are indicated by angle brackets, such as <file_name>. There are different types of parameters listed in the following table.

Table 5: Parameter types, descriptions, and examples

Parameter type	Description	Example
Boolean	Boolean numbers or values	ON or ≠ 0; OFF or 0
NR1 numeric	Integers	0, 1, 15, –1
NR2 numeric	Decimal numbers	1.2, 3.141,–6.5
NR3 numeric	Floating point numbers	3.1415E+9
NRf numeric	Flexible decimal numbers that can be type NR1, NR2, or NR3	See NR1, NR2, and NR3 examples in this table
Double	Double precision numbers	NRf numeric
String	Alphanumeric characters (must be within quotation marks)	"Testing 1, 2, 3"

Connection commands

The following table summarizes the Connection commands.

Command	Inputs	Outputs
bool *PI_connect*	int unit	int e_num, string e_str
bool *PI_conn_avail_Q*		string[] MSN_list, int e_num, string e_str
bool *PI_conn_power*	int unit, bool turnON	int e_num, string e_str
bool *PI_conn_power_Q*	int unit	bool isON, int e_num, string e_str
bool *PI_conn_refresh*		int numAvailable, int e_num, string e_str
bool *PI_disconnect_all*		int e_num, string e_str

PI_connect (Command only)

Connect to an available linear equalizers by the index (unit) number.

Conditions Only one linear equalizer can be connected at one time.

Group Connection

Syntax PI_connect <unit>

Arguments <unit> ::= <NR1>, a value between 1 and 4.

Examples PI_connect 1 connects to linear equalizer 1.

PI_conn_avail_Q (Query only)

Returns the list, by serial number, of available linear equalizers for connection.

Conditions

Up to four linear equalizers can be connected simultaneously. This command can be issued from any mode.

Group

Connection

Syntax

PI_conn_avail_Q

Returns

<string> ::=
"<index>,<model><sp><serial_number>,<index>,<model><sp><serial_number>, ..., <index>,<model><sp><serial_number>"

<index> identifies the position of the linear equalizer within the string array. Use this number as the <unit> variable in other commands.

<model> is the LE160 or LE320.

<sp> ::= space

<serial_number> is the manufacturing serial number of the linear equalizer.

Examples

PI_conn_avail_Q might return "1,LE320 B001235,2,LE320 B0012435" indicating that two LE320 instruments with serial numbers B001235, and B0012435 are available for connection.

PI_conn_power

Turns on or turns off the power for the available linear equalizer. The query returns 1 if on or 0 if off.

Conditions

Most commands are ignored when the linear equalizer is powered off. Only the connection commands are recognized when the linear equalizer is powered off. This command can be issued from any mode.

Group

Connection

Syntax

PI_conn_power <unit>, {1|0}

PI_conn_power_Q <unit>

Arguments

<unit> ::= <NR1>, a value between 1 and 4.

1 turns the power on.

0 turns the power off.

Returns

A single <Boolean> value.

Examples

PI_conn_power_Q 1 might return 1 indicating that linear equalizer #1 is turned on.

PI_conn_refresh

Scans the USB bus for linear equalizers and returns the number available for connection.

Conditions

Supports connections of up to four linear equalizers. This command can be issued from any mode.

Group

Connection

Syntax

PI_conn_refresh

Arguments

None

Returns

A single <NR1> value.

Examples

PI_conn_refresh might return 2 indicating that two linear equalizers are available for connection.

PI_disconnect_all (Command only)

Disconnect all linear equalizers from the application.

Conditions This command can be issued from any mode.

Group Connection

Syntax PI_disconnect_all

Arguments None

Examples PI_disconnect_all disconnects all linear equalizers managed by the application.

Mode commands

The following table summarizes the Mode commands.

Command	Inputs	Outputs
bool *PI_mode*	int unit, int chan, int mode	int e_num, string e_str
bool *PI_mode_Q*	int unit, int chan	int mode, int e_num, string e_str

PI_mode

Sets or returns the mode information. The following modes are available: 9-Tap, Raw 9-Tap, 4-Tap, CDS, SPM. The instrument stores a different state vector for each major feature or mode.

Conditions

The linear equalizer must be powered on and must be connected (through the PI_connect command).

Group

Mode

Syntax

PI_mode <unit>,<channel>,<mode>

PI_mode_Q <unit>,<channel>

Related commands

PI_connect, *PI_conn_power*

Arguments

<unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1> , channel 1 or 2.

<mode> ::= <NR1>

1 represents Raw 9-Tap mode.

2 represents 9-Tap mode.

3 represents 4-Tap mode.

4 represents SPM mode.

5 represents CDS mode.

Returns

A single <NR1> value.

Example

PI_mode 1,1,1 sets the linear equalizer #1, Channel 1, to the 9-Tap mode.

PI_mode_Q 2,1 which returns 3, indicates that the linear equalizer #2, Channel 1, is in the 4-Tap mode.

4-Tap commands

The following table summarizes the 4-Tap operating mode commands.

Command	Inputs	Outputs
bool *PI_tap4_agc_gain*	int unit, int chan, double gain	int e_num, string e_str
bool *PI_tap4_agc_gain_Q*	int unit, int chan	double gain, int e_num, string e_str
bool *PI_tap4_agc_lev*	int unit, int chan, int agc level	int e_num, string e_str
bool *PI_tap4_agc_lev_Q*	int unit, int chan	int agc level, int e_num, string e_str
bool*PI_tap4_agc_state*	int unit, int chan, bool lockIt	int e_num, string e_str
bool *PI_tap4_agc_state_Q*	int unit, int chan	bool isLocked, int e_num, string e_str
bool *PI_tap4_deemph*	int unit, int chan, double deemp	int e_num, string e_str
bool *PI_tap4_deemph_Q*	int unit, int chan	double deemp, int e_num, string e_str
bool *PI_tap4_emu_rate*	int unit, int chan, double rate	int e_num, string e_str
bool *PI_tap4_emu_rate_Q*	int unit, int chan	double rate, int e_num, string e_str
bool *PI_tap4_offs_state*	int unit, int chan, bool setAuto	int e_num, string e_str
bool *PI_tap4_offs_state_Q*	int unit, int chan	bool isAuto, int e_num, string e_str
bool *PI_tap4_offs_val*	int unit, int chan, double offs mV	int e_num, string e_str
bool *PI_tap4_offs_val_Q*	int unit, int chan	double offs mV, int e_num, string e_str
bool *PI_tap4_out_amp*	int unit, int chan, double ampMv	int e_num, string e_str
bool *PI_tap4_out_amp_Q*	int unit, int chan	double ampMv, int e_num, string e_str
bool *PI_tap4_out_cal_Q*	int unit, int chan	bool isCalib, int e_num, string e_str
bool *PI_tap4_tap_enable*	int unit, int chan, bool enable	int e_num, string e_str
bool *PI_tap4_tap_enable_Q*	int unit, int chan, int tap	bool isEnabled, int e_num, string e_str
bool *PI_tap4_preshoot*	int unit, int chan, double presh	int e_num, string e_str
bool *PI_tap4_preshoot_Q*	int unit, int chan	double presh, int e_num, string e_str
bool *PI_tap4_statefile_load*	int unit, int chan, string path	bool isCompat, int e_num, string e_str
bool *PI_tap4_statefile_save*	int unit, int chan, string path, bool replace	int e_num, string e_str
bool *PI_tap4_out_enable*	int unit, int chan, bool enable	int e_num, string e_str
bool *PI_tap4_out_enable_Q*	int unit, int chan	bool isEnabled, int e_num, string e_str
bool *PI_tap4_tap_enable_all*	int unit, int chan, bool enable All	int e_num, string e_str
bool *PI_tap4_tap_enable_all_Q*	int unit, int chan	bool all Enabled, int e_num, string e_str
bool *PI_tap4_tap_val*	int unit, int chan, int tap, int val	int e_num, string e_str
bool *PI_tap4_tap_val_Q*	int unit, int chan, int tap	int tap val, int e_num, string e_str

PI_tap4_agc_gain

Sets or returns the value of the AGC Adjust control in the 4-Tap mode.

Conditions

Only valid when the AGC state is set to Lock.

Must be in 4-Tap mode (3) and connected.

Group

4-Tap

Syntax

PI_tap4_agc_gain <unit>,<channel>, <gain>

PI_tap4_agc_gain_Q <unit>,<channel>

Related commands

PI_tap4_agc_state

Arguments

<unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<gain> ::= <double>, the uncalibrated gain adjustment value in the range from 1.80 to 2.50.

Returns

A single <NRf> number.

Examples

PI_tap4_agc_gain 1,1,2.10 sets the AGC Gain adjust value to 2.10 for Channel 1 of linear equalizer #1.

PI_tap4_agc_lev

Sets or returns the automatic gain control level in the 4-Tap mode.

When the PI_tap4_agc_state is OFF or 0, the label in the user interface is set to "AGC Level" and the AGC chip circuitry has eight ranges.

Conditions Must be in 4-Tap mode (3) and connected.

Group 4-Tap

Syntax PI_tap4_agc_lev <unit>,<channel>, <level>

PI_tap4_agc_lev_Q <unit>,<channel>

Related commands *PI_tap4_agc_state*

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<level> ::= <NR1> the AGC level; a number from 0 to 7.

Returns A single <NR1> value.

Examples PI_tap4_agc_lev_Q 1,1 might return 3 indicating that the AGC level for Channel 1 of linear equalizer #1 is set to 3.

PI_tap4_agc_state

Set or returns automatic gain control lock status in the 4-Tap mode.

Conditions Must be in 4-Tap mode (3) and connected.

Group 4-Tap

Syntax PI_tap4_agc_state <unit>,<channel>, {0|1}

PI_tap4_agc_state_Q <unit>,<channel>

Related commands *PI_tap4_agc_lev*, *PI_tap4_agc_gain*, *PI_tap4_out_amp*

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

0 indicates that the lock is false and the AGC is controlled by the chip ALC (automatic level control) loop.

1 indicates that the lock is true and turns on the UNCAL state and warning.

Returns A single <Boolean> value.

Examples PI_tap4_agc_state 1,1 enables the automatic gain control lock and turns on the UNCAL state or warning on Channel 1 of linear equalizer #1.

PI_tap4_deemph

Sets or returns the de-emphasis value in the 4-Tap mode.

Conditions Must be in 4-Tap mode (3) and connected.

Group 4-Tap

Syntax PI_tap4_deemph <unit>,<channel>, <dBvalue>

PI_tap4_deemph_Q <unit>,<channel>

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<dBvalue> ::= <double>, deemphasis value.

Returns A single <NRf> value.

Examples PI_tap4_deemph 1,1,-10 sets the deemphasis value for Channel 1 of linear equalizer #1 to -10 dB.

PI_tap4_emu_rate

Sets or returns the clock rate for the emulated data in the 4-Tap mode.

Conditions The clock rates depend on the tap spacing for the selected linear equalizer.

Must be in 4-Tap mode (3) and connected.

Group 4-Tap

Syntax PI_tap4_emu_rate <unit>,<channel>, <rate>

PI_tap4_emu_rate_Q <unit>,<channel>

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<rate> ::= <double>, data rate.

Returns A single <NRf> value.

Examples PI_tap4_emu_rate 1,1,2.5E10 sets the rate for Channel 1 of linear equalizer #1 to 25.00 Gb/s.

PI_tap4_offs_state

Sets or returns the state of the input offset control in the 4-Tap mode.

Conditions Must be in 4-Tap mode (3) and connected.

Group 4-Tap

Syntax PI_tap4_offs_state <unit>,<channel>,{1|0}

PI_tap4_offs_state_Q <unit>,<channel>

Related commands *PI_tap4_offs_val*

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<state> ::= <Boolean>. 1 sets the input offset control to AUTO; 0 sets the input offset control to manual control.

Returns A single <Boolean> value. 1 or ON indicates that the input offset control is set to AUTO; 0 or OFF indicates that the input offset control is set to manual control.

Examples PI_tap4_offs_state 1,1,0 sets the input offset control for Channel 1 of linear equalizer #1 to manual control.

PI_tap4_offs_val

Sets or returns the input offset value in the 4-Tap mode in millivolts.

Conditions

If the input offset control is set to AUTO, any warnings are ignored.

Must be in 4-Tap mode (3) and connected.

Group

4-Tap

Syntax

PI_tap4_offs_val <unit>,<channel>, <offset>

PI_tap4_offs_val_Q <unit>,<channel>

Related commands

PI_tap4_offs_state

Arguments

<unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<offset> ::= <double>, the input offset value, a number between -78 and 78.

Returns

A single <NR1> value. The returned value is in millivolts.

Examples

PI_tap4_offs_val 1,1,50 sets the input offset for Channel 1 of linear equalizer #1 to 50 mv.

PI_tap4_out_amp

Sets or returns the output amplitude value in the 4-Tap mode. The value is in mV_{p-p}.

Conditions

Must be in 4-Tap mode (3) and connected.

The value might be calibrated when the AGC Lock control is off. If the AGC Lock control is on, the value is uncalibrated.

Use the query (PI_tap4_out_cal_Q) to determine if an amplitude value was able to be calibrated. Refer to *Appendix B: Theory of operation, Output amplitude*, for conditions which affect amplitude calibration.

Group

4-Tap

Syntax

PI_tap4_out_amp <unit>,<channel>, <value>

PI_tap4_out_amp_Q <unit>,<channel>

Related commands

PI_tap4_agc_state, *PI_tap4_out_cal_Q*

Arguments

<unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<value> ::= <double>, the value of the output control from 0 to 6000 (equivalent to 6.0 volts, but must be specified in mV).

Returns

A single <double> value.

Examples

PI_tap4_out_amp_Q 1,1 might return 0 indicating that the output amplitude for Channel 1 of linear equalizer #1 is 0 mV_{p-p}.

PI_tap4_out_cal_Q (Query only)

Use this query to determine if the output amplitude setting resulted in a calibrated value. It returns the state of the Output Amplitude control in the 4-Tap mode. When the AGC Lock control is on, this query always returns false (uncalibrated).

Conditions Must be in 4-Tap mode (3) and connected.

Group 4-Tap

Syntax PI_tap4_out_cal_Q <unit>,<channel>

Related commands *PI_tap4_agc_state*, *PI_tap4_out_amp*

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

Returns A single <Boolean> value. A 1 indicates that the Output Amplitude control is calibrated. A 0 indicates that the Output Amplitude control is uncalibrated and the label is red.

Examples PI_tap4_out_cal_Q 1,1 might return 1 indicating that the Output Amplitude control for Channel 1 of linear equalizer #1 is calibrated (the AGC Lock control is off).

PI_tap4_out_enable

This command sets the output-enable pin of the LE chip, for the specified channel. It enables or disables the output state in the 4-Tap mode.

Conditions Must be in 4-Tap mode (3) and connected.

Group 4-Tap

Syntax PI_tap4_out_enable <unit>,<channel>,{1|0}

PI_tap4_out_enable_Q <unit>,<channel>

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<output-state> ::= <boolean>, 1 enables the output state and 0 disables the output state.

Returns A single <boolean> value. 1 indicates that the output state is enabled or on; 0 indicates that the output state is disabled or off.

Examples PI_tap4_out_enable 1,1,0 enables the Channel 1 output of linear equalizer #1.

PI_tap4_preshoot

Sets or returns the pre-shoot value in the 4-Tap mode.

Conditions Must be in 4-Tap mode (3) and connected.

Group 4-Tap

Syntax PI_tap4_preshoot <unit>,<channel>,<dBvalue>

PI_tap4_preshoot_Q <unit>,<channel>

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<dBvalue> ::= <double>, pre-shoot value.

Returns A single <NRf> value.

Examples PI_tap4_preshoot 1,1,15 sets the pre-shoot value for Channel 1 of linear equalizer #1 to 15 dB.

PI_tap4_statefile_load (Command only)

Loads a previously saved XML state file on the host computer.

Conditions The file is loaded from the folder specified in the Connect view.

Must be in 4-Tap mode (3) and connected.

Group 4-Tap

Syntax PI_tap4_statefile_load <unit>,<channel>, <path-filename>

Related commands *PI_tap4_statefile_save*

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<path-filename> ::= <string>

Examples PI_tap4_statefile_load 1,2,"C:\Users\Documents\Tektronix-LE\TekLE-Tap4-State-Qualtest1.xml" loads the TekLE-Tap4-State-Qualtest1.xml file.

PI_tap4_statefile_save (Command only)

Saves a state file on the host computer.

The file is saved in the folder locations specified in the Connect view.

Conditions Must be in 4-Tap mode (3) and connected.

Group 4-Tap

Syntax PI_tap4_statefile_save <unit>,<channel>, <path-filename>.

Related commands *PI_tap4_statefile_load*

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<path-filename> ::= <string>

Examples PI_tap4_statefile_save 1,2,"C:\Users\Documents\Tektronix-LE\TekLE-TAP4-State-test4-18.xml" saves the TekLE-TAP4-State-test4-18.xml file to the C:Users \Documents\Tektronix-LE folder on the host computer.

PI_tap4_tap_enable

Sets or returns the enable/disable state of the specified tap in the 4-Tap mode.

Conditions Must be in 4-Tap mode (3) and connected.

Group 4-Tap

Syntax PI_tap4_tap_enable <unit>,<channel>,<tap, tap-state>

PI_tap4_tap_enable_Q <unit>,<channel>, <tap>

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<tap> ::= <NR1>, a tap number between 1 and 4.

<tap, tap-state>, the tap number, where the tap-state is a boolean value. 1 is enabled; 0 is disabled.

Returns The tap number and a <Boolean> value for the tap state.

Examples PI_tap4_tap_enable_Q 2,2,2 might return 0 indicating that tap #2 is disabled on Channel 2 of linear equalizer #2.

PI_tap4_tap_enable_all

Enable or disables all taps in the 4-Tap mode.

Conditions Must be in 4-Tap mode (3) and connected.

Group 4-Tap

Syntax PI_tap4_tap_enable_all <unit>,<channel>, <enable>

PI_tap4_tap_enable_all_Q <unit>,<channel>

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<enable> ::= <Boolean> where 1 enables all taps or 0 disables all taps.

Returns A <Boolean> value.

Examples PI_tap4_tap_enable_all 2,1,1 enables all taps on Channel 1 of linear equalizer #2.

PI_tap4_tap_val

Sets or returns the value of the specified tap in the 4-Tap mode.

Conditions Must be in 4-Tap mode (3) and connected.

Group 4-Tap

Syntax PI_tap4_tap_val <unit>,<channel>,<tap, tap-value>

PI_tap4_tap_val_Q <unit>,<channel>, <tap>

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<tap> ::= <NR1>, a tap number between 1 and 4.

<tap, tap-value>, the tap number, where tap-value is an integer between -100 and 100.

Returns An <NR1> tap value.

Examples PI_tap4_tap_val 2,2,2,45 sets tap #2 on linear equalizer #2, Channel 2, to 45.

9-Tap commands

The following table summarizes the 9-Tap operating mode commands.

Command	Inputs	Outputs
bool *PI_tap9_agc_gain*	int unit, int chan, double gain	int e_num, string e_str
bool *PI_tap9_agc_gain_Q*	int unit, int chan	double gain, int e_num, string e_str
bool *PI_tap9_agc_lev*	int unit, int chan, bool level	int e_num, string e_str
bool *PI_tap9_agc_lev_Q*	int unit, int chan	int agc level, int e_num, string e_str
bool *PI_tap9_agc_state*	int unit, int chan, bool lockit	int e_num, string e_str
bool *PI_tap9_agc_state_Q*	int unit, int chan	bool isLocked, int e_num, string e_str
bool *PI_tap9_offs_state*	int unit, int chan, bool set Auto	int e_num, string e_str
bool *PI_tap9_offs_state_Q*	int unit, int chan	bool isAuto, int e_num, string e_str
bool *PI_tap9_offs_val*	int unit, int chan, bool double offset_mV	int e_num, string e_str
bool *PI_tap9_offs_val_Q*	int unit, int chan	double offs_mV, int e_num, string e_str
bool *PI_tap9_out_amp*	int unit, int chan, double ampMv	int e_num, string e_str
bool *PI_tap9_out_amp_Q*	int unit, int chan	double aMv, int e_num, string e_str
bool *PI_tap9_out_cal_Q*	int unit, int chan	bool isCalib, nt e_num, string e_str
bool *PI_tap9_out_enable*	int unit, int chan, bool enable	int e_num, string e_str
bool *PI_tap9_out_enable_Q*	int unit, int chan	bool isEnabled, int e_num, string e_str
bool *PI_tap9_statefile_load*	int unit, int chan, string path	bool isCompat, int e_num, string e_str
bool *PI_tap9_statefile_save*	int unit, int chan, string path, bool replace	int e_num, string e_str
bool *PI_tap9_tap_enable*	int unit, int chan, int tap, bool en	int e_num, string e_str
bool *PI_tap9_tap_enable_Q*	int unit, int chan, int tap	bool enabled, int e_num, string e_str
bool *PI_tap9_tap_enable_all*	int unit, int chan, bool enableAll	int e_num, string e_str
bool *PI_tap9_tap_enable_all_Q*	int unit, int chan	bool allEnabled, int e_num, string e_str
bool *PI_tap9_tap_val*	int unit, int chan, int tap, int val	int e_num, string e_str
bool *PI_tap9_tap_val_Q*	int unit, int chan, int tap	int tap_val, int e_num, string e_str

PI_tap9_agc_gain

Sets or returns the value of the AGC adjust control in the 9-Tap mode.

Conditions

Only valid when the AGC state is set to Lock.

Must be in 9-Tap mode (2) and connected.

Group

9-Tap

Syntax

PI_tap9_agc_gain <unit>,<channel>, <gain>

PI_tap9_agc_gain_Q <unit>,<channel>

Related commands

PI_tap9_agc_state

Arguments

<unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<gain> ::= <double>, the uncalibrated gain adjustment value in the range from 1.80 to 2.50.

Returns

A single <NRf> number.

Examples

PI_tap9_agc_gain 1,1,2.10 sets the AGC Gain adjust value to 2.10 for Channel 1 of linear equalizer #1.

PI_tap9_agc_lev

Sets or returns the automatic gain control level in the 9-Tap mode.

When the PI_tap9_agc_state is OFF or 0, the label in the user interface is set to "AGC Level" and the AGC chip circuitry has eight ranges.

Conditions

Must be in 9-Tap mode (2) and connected.

Group

9-Tap

Syntax

PI_tap9_agc_lev <unit>,<channel>, <level>

PI_tap9_agc_lev_Q <unit>,<channel>

Related commands

PI_tap9_agc_state

Arguments

<unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<level> ::= <NR1> the AGC level; a number from 0 to 7.

Returns

A single <NR1> value.

Examples

PI_tap9_agc_lev_Q 1,1 might return 3 indicating that the AGC level for Channel 1 of linear equalizer #1 is set to 3.

PI_tap9_agc_state

Set or returns automatic gain control lock status in the 9-Tap mode.

Conditions

Must be in 9-Tap mode (2) and connected.

Group

9-Tap

Syntax

PI_tap9_agc_state <unit>,<channel>, {0|1}

PI_tap9_agc_state_Q <unit>,<channel>

Related commands

PI_tap9_agc_lev, *PI_tap9_agc_gain*, *PI_tap9_out_amp*

Arguments

<unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

0 indicates that the lock is false and the AGC is controlled by the chip ALC (automatic level control) loop.

1 indicates that the lock is true and enables the UNCAL state or warning.

Returns

A single <Boolean> value.

Examples

PI_tap9_agc_state 1,1 enables the automatic gain control lock and enables the UNCAL state or warning on Channel 1 of linear equalizer #1.

PI_tap9_offs_state

Sets or returns the state of the input offset control in the 9-Tap mode.

Conditions

Must be in 9-Tap mode (2) and connected.

Group

9-Tap

Syntax

PI_tap9_offs_state <unit>,<channel>,{1|0}

PI_tap9_offs_state_Q <unit>,<channel>

Related commands

PI_tap9_offs_val

Arguments

<unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<state> ::= <Boolean>. 1 sets the input offset control to AUTO; 0 sets the input offset control to manual control.

Returns

A single <Boolean> value. 1 indicates that the input offset control is set to AUTO; 0 indicates that the input offset control is set to manual control.

Examples

PI_tap9_offs_state 1,1,0 sets the input offset control for Channel 1 of linear equalizer #1 to manual control.

PI_tap9_offs_val

Sets or returns the input offset value in the 9-Tap mode in millivolts.

Conditions

If the input offset control is set to AUTO, any warnings are ignored.

Must be in 9-Tap mode (2) and connected.

Group

9-Tap

Syntax

PI_tap9_offs_val <unit>,<channel>, <offset>

PI_tap9_offs_val_Q <unit>,<channel>

Related commands

PI_tap9_offs_state

Arguments

<unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<offset> ::= <double>, the input offset value, a number between -78 and 78.

Returns

A single <NRf> value. the returned value is in millivolts.

Examples

PI_tap9_offs_val 1,1,20 sets the input offset for Channel 1 of linear equalizer #1 to 20 mv.

PI_tap9_out_amp

Sets or returns the Output Amplitude value in the 9-Tap mode. The value is in $mV_{p\text{-}p}$.

Conditions

Must be in 9-Tap mode (2) and connected.

The value might be calibrated when the AGC Lock control is off. If the AGC Lock control is on, the value is uncalibrated. Use the query (PI_tap9_out_cal_Q) to determine if an amplitude value was able to be calibrated. Refer to *Appendix B: Theory of operation, Output amplitude*, for conditions which affect amplitude calibration.

Group

9-Tap

Syntax

PI_tap9_out_amp <unit>,<channel>, <value>

PI_tap9_out_amp_Q <unit>,<channel>

Related commands

PI_tap9_agc_state, *PI_tap9_out_cal_Q*

Arguments

<unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<value> ::= <double>, the value of the output control from 0 to 6000 (equivalent to 6.0 volts, but must be specified in mV).

Returns

A single <NRf> value.

Examples

PI_tap9_out_amp_Q 1,1 might return 0 indicating that the Output Amplitude for Channel 1 of linear equalizer #1 is 0 $V_{p\text{-}p}$.

PI_tap9_out_cal_Q (Query only)

Use this query to determine if the output amplitude setting resulted in a calibrated value. It returns the state of the output amplitude control in the 9-Tap mode. When the AGC Lock control is on, this query will always return false (uncalibrated).

Conditions Must be in 9-Tap mode (2) and connected.

Group 9-Tap

Syntax PI_tap9_out_cal_Q <unit>,<channel>

Related commands *PI_tap9_agc_state*, *PI_9tap_out_amp*

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

Returns A single <Boolean> value. A 1 indicates that the Output Amplitude control is calibrated. A 0 indicates that the Output Amplitude control is uncalibrated.

Examples PI_tap9_out_cal_Q 1,1 might return 1 indicating that the Output Amplitude control for Channel 1 of linear equalizer #1 is calibrated (the AGC Lock control is off).

PI_tap9_out_enable

This command sets the output enable pin of the LE chip, for the specified channel. It enables or disables the output state in the 9-Tap mode.

Conditions

Must be in 9-Tap mode (2) and connected.

Group

9-Tap

Syntax

PI_tap9_out_enable <unit>,<channel>,{1|0}

PI_tap9_out_enable_Q <unit>,<channel>

Arguments

<unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<output-state> ::= <boolean>, 1 enables the output state and 0 or disables the output state.

Returns

A single <boolean> value. 1 indicates that the output state is enabled or on; 0 indicates that the output state is disabled or off.

Examples

PI_tap9_tap_out_enable 1,1,0 enables the Channel 1 output of linear equalizer #1.

PI_tap9_statefile_load (Command only)

Loads a previously saved 9-Tap XML state file on the host computer.

Conditions

The file is loaded from the folder specified in the Connect view.

Must be in 9-Tap mode (2) and connected.

Group

9-Tap

Syntax

PI_tap9_statefile_load <unit>,<channel>, <path-filename>

Related commands

PI_tap9_statefile_save

Arguments

<unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<path-filename> ::= <string>

Examples

PI_tap9_statefile_load 1,2,"C:\Users\Documents\Tektronix-LE\TekLE-Tap9-State-Qualtest1.xml" loads the TekLE-Tap9-State-Qualtest1.xml file.

PI_tap9_statefile_save (Command only)

Saves a 9-Tap state file on the host computer.

The file is saved in the folder locations specified in the Connect view.

Conditions Must be in 9-Tap mode (2) and connected.

Group 9-Tap

Syntax PI_tap9_statefile_save <unit>,<channel>, <path-filename>.

Related commands *PI_tap9_statefile_load*

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<path-filename> ::= <string>

Examples PI_tap9_statefile_save 1,2,"C:\Users\Documents\Tektronix-LE\TekLE-TAP9-State-test4-18.xml" saves the TekLE-TAP9-State-test4-18.xml file to the C:Users \Documents\Tektronix-LE folder on the host computer.

PI_tap9_tap_enable

Sets or returns the enable/disable state of the specified tap in the 9-Tap mode.

Conditions Must be in 9-Tap mode (2) and connected.

Group 9-Tap

Syntax PI_tap9_tap_enable <unit>,<channel>,<tap, tap-state>

PI_tap9_tap_enable_Q <unit>,<channel>, <tap>

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<tap> ::= <NR1>, a tap number between 1 and 9.

<tap, tap-state>, the tap number, where the tap-state is a boolean value. 1 is enabled; 0 is disabled.

Returns The tap number and a <Boolean> value for the tap state.

Examples PI_tap9_tap_enable_Q 2,2,2 might return 0 indicating that tap #2 is disabled on Channel 2 linear of equalizer #2.

PI_tap9_tap_enable_all

Enable or disables all taps in the 9-Tap mode.

Conditions The command requires the enable/disable states of all taps to be specified.

Must be in 9-Tap mode (2) and connected.

Group 9-Tap

Syntax PI_tap9_tap_enable_all <unit>,<channel>,<enable>

PI_tap9_tap_enable_all_Q <unit>,<channel>

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<enable> ::= <Boolean>, where 1 enables all taps or 0 disables all taps.

Returns A <Boolean> value.

Examples PI_tap9_tap_enable_all 2,1,1 enables all taps on Channel 1 of linear equalizer #2 in the 9-Tap mode.

PI_tap9_tap_val

Sets or returns the value of the specified tap in the 9-Tap mode.

Conditions Must be in 9-Tap mode (2) and connected.

Group 9-Tap

Syntax PI_tap9_tap_val <unit>,<channel>,<tap, tap-value>

PI_tap9_tap_val_Q <unit>,<channel>, <tap>

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<tap> ::= <NR1>, a tap number between 1 and 9.

<tap, tap-value>, the tap number, where tap-value is an integer between -100 and 100.

Returns The tap number and an <NR1> tap value.

Examples PI_tap9_tap_val 2,2,2,45 sets tap #2 on linear equalizer #2, Channel 2, to 45.

Raw 9-Tap Commands

The following table summarizes the Raw 9-Tap commands.

Command	Inputs	Outputs
bool *PI_tap9R_agc_gain*	int unit, int chan, double gain	int e_num, string e_str
bool *PI_tap9R_agc_gain_Q*	int unit, int chan	double gain, int e_num, string e_str
bool *PI_tap9R_agc_lev*	int unit, int chan, bool level	int e_num, string e_str
bool *PI_tap9R_agc_lev_Q*	int unit, int chan	int agc level, int e_num, string e_str
bool *PI_tap9R_agc_state*	int unit, int chan, bool lockit	int e_num, string e_str
bool *PI_tap9R_agc_state_Q*	int unit, int chan	bool isLocked, int e_num, string e_str
bool *PI_tap9R_offs_state*	int unit, int chan, bool set Auto	int e_num, string e_str
bool *PI_tap9R_offs_state_Q*	int unit, int chan	bool isAuto, int e_num, string e_str
bool *PI_tap9R_offs_val*	int unit, int chan, bool double offset_mV	int e_num, string e_str
bool *PI_tap9R_offs_val_Q*	int unit, int chan	double offs_mV, int e_num, string e_str
bool *PI_tap9R_out_amp*	int unit, int chan, double ampMv	int e_num, string e_str
bool *PI_tap9R_out_amp_Q*	int unit, int chan	double aMv, int e_num, string e_str
bool *PI_tap9R_out_enable*	int unit, int chan, bool enable	int e_num, string e_str
bool *PI_tap9R_out_enable_Q*	int unit, int chan	bool isEnabled, int e_num, string e_str
bool *PI_tap9R_statefile_load*	int unit, int chan, string path	bool isCompat, int e_num, string e_str
bool *PI_tap9R_statefile_save*	int unit, int chan, string path, bool replace	int e_num, string e_str
bool *PI_tap9R_tap_enable*	int unit, int chan, int tap, bool en	int e_num, string e_str
bool *PI_tap9R_tap_enable_Q*	int unit, int chan, int tap	bool enabled, int e_num, string e_str
bool *PI_tap9R_tap_enable_all*	int unit, int chan, bool enableAll	int e_num, string e_str
bool *PI_tap9R_tap_enable_all_Q*	int unit, int chan	bool allEnabled, int e_num, string e_str
bool *PI_tap9R_tap_val*	int unit, int chan, int tap, int val	int e_num, string e_str
bool *PI_tap9R_tap_val_Q*	int unit, int chan, int tap	int tap_val, int e_num, string e_str

PI_tap9R_agc_gain

Sets or returns the value of the AGC adjust control in the Raw 9-Tap mode.

Conditions

Only valid when the AGC state is set to Lock.

Must be in Raw 9-Tap mode (1) and connected.

Group

Raw 9-Tap

Syntax

PI_tap9R_agc_gain <unit>,<channel>, <gain>

PI_tap9R_agc_gain_Q <unit>,<channel>

Related commands

PI_tap9R_agc_state

Arguments

<unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<gain> ::= <double>, the uncalibrated gain adjustment value in the range from 1.80 to 2.50.

Returns

A single <NRf> number.

Examples

PI_tap9R_agc_gain 1,1,2.10 sets the AGC Gain adjust value to 2.10 for Channel 1 of linear equalizer #1.

PI_tap9R_agc_lev

Sets or returns the 9-Tap automatic gain control level in the Raw 9-Tap mode.

When the PI_tap9R_agc_state is OFF or 0, the label in the user interface is set to "AGC Level" and the AGC chip circuitry has eight ranges.

Conditions None

Group Raw 9-Tap

Syntax PI_tap9R_agc_lev <unit>,<channel>, <level>

PI_tap9R_agc_lev_Q <unit>,<channel>

Related commands *PI_tap9R_agc_state*

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

level> ::= <NR1> the AGC level; a number from 0 to 7.

Returns A single <NR1> value.

Examples PI_tap9R_agc_lev_Q 1,1 might return 3 indicating that the AGC level for Channel 1 of linear equalizer #1 is set to 3.

PI_tap9R_agc_state

Set or returns automatic gain control lock status in the Raw 9-Tap mode.

Conditions Must be in Raw 9-Tap mode (1) and connected.

Group Raw 9-Tap

Syntax PI_tap9R_agc_state <unit>,<channel>, {0|1}

PI_tap9R_agc_state_Q <unit>,<channel>

Related commands *PI_tap9R_agc_lev*, *PI_tap9R_agc_gain*, *PI_tap9R_out_amp*

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

0 indicates that the lock is false and the AGC is controlled by the chip ALC (automatic level control) loop.

1 indicates that the lock is true and turns on the UNCAL state or warning.

Returns A single <Boolean> value.

Examples PI_tap9R_agc_state 1,1 enables the automatic gain control lock and turns on the UNCAL state or warning on Channel 1 of linear equalizer #1.

PI_tap9R_offs_state

Sets or returns the state of the input offset control in the Raw 9-Tap mode.

Conditions

None

Must be in Raw 9-Tap mode (1) and connected.

Group

Raw 9-Tap

Syntax

PI_tap9R_offs_state <unit>,<channel>,{1|0}

PI_tap9R_offs_state_Q <unit>,<channel>

Related commands

PI_tap9R_offs_val

Arguments

<unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<state> ::= <Boolean>, 1 sets the input offset control to AUTO; 0 sets the input offset control to manual control.

Returns

A single <Boolean> value. 1 or ON indicates that the input offset control is set to AUTO; 0 or OFF indicates that the input offset control is set to manual control.

Examples

PI_tap9R_offs_state 1,1,0 sets the input offset control for Channel 1 of linear equalizer #1 to manual control.

PI_tap9R_offs_val

Sets or returns the input offset value in the Raw 9-Tap mode.

Conditions If the input offset control is set to AUTO, any warnings are ignored.

Must be in Raw 9-Tap mode (1) and connected.

Group Raw 9-Tap

Syntax PI_tap9R_offs_val <unit>,<channel>, <offset>

PI_tap9R_offs_val_Q <unit>,<channel>

Related commands *PI_tap9R_offs_state*

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<offset> ::= <double>, the input offset value, a number between -31 and +31.

Returns A single <NRf> value.

Examples PI_tap9R_offs_val 1,1,20 sets the input offset for Channel 1 of linear equalizer #1 to 20.

PI_tap9R_out_amp

Sets or returns the output amplitude value in the Raw 9-Tap mode. In this mode, the output amplitude is a DAC value (not in mV as in the other modes).

Conditions Must be in Raw 9-Tap mode (1) and connected.

Group Raw 9-Tap

Syntax PI_tap9R_out_amp <unit>,<channel>, <value>

PI_tap9R_out_amp_Q <unit>,<channel>

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<value> ::= <double>, the value of the output control from 0 to 63.

Returns A single <NRf> value.

Examples PI_tap9R_out_amp_Q 1,1 might return 0 indicating that the Output Amplitude for Channel 1 of linear equalizer #1 is 0.

PI_tap9R_out_enable

This command sets the output-enable pin of the LE chip, for the specified channel. It enables or disables the output state in the Raw 9-Tap mode.

Conditions Must be in Raw 9-Tap mode (1) and connected.

Group Raw 9-Tap

Syntax PI_tap9R_out_enable <unit>,<channel>,{1|0}

PI_tap9R_out_enable_Q <unit>,<channel>

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<output-state> ::= <boolean>, 1 enables the output state and 0 disables the output state.

Returns A single <boolean> value. 1 indicates that the output state is enabled or on; 0 indicates that the output state is disabled or off.

Examples PI_tap9R_out_enable 1,1,0 enables the Channel 1 output of linear equalizer #1.

PI_tap9R_statefile_load (Command only)

Loads a previously saved Raw 9-Tap XML state file on the host computer.

Conditions

The file is loaded from the folder specified in the Connect view.

Must be in Raw 9-Tap mode (1) and connected.

Group

Raw 9-Tap

Syntax

PI_tap9R_statefile_load <unit>,<channel>, <path-filename>

Related commands

PI_tap9R_statefile_save

Arguments

<unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<path-filename> ::= <string>

Examples

PI_tap9R_statefile_load 1,2,"C:\Users\Documents\Tektronix-LE\TekLE-Tap9R-State-Qualtest1.xml" loads the TekLE-Tap9R-State-Qualtest1.xml file.

PI_tap9R_statefile_save (Command only)

Saves a Raw 9-Tap state file on the host computer.

The file is saved in the folder locations specified in the Connect view.

Conditions

Must be in Raw 9-Tap mode (1) and connected.

Group

Raw 9-Tap

Syntax

PI_tap9R_statefile_save <unit>,<channel>, <path-filename>.

Related commands

PI_tap9R_statefile_load

Arguments

<unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<path-filename> ::= <string>

Examples

PI_tap9R_statefile_save 1,2,"C:\Users\Documents\Tektronix-LE\TekLE-TAP9R-State-test4-18.xml" saves the TekLE-TAP9R-State-test4-18.xml file to the C:Users\Documents\Tektronix-LE folder on the host computer.

PI_tap9R_tap_enable

Sets or returns the enable/disable state of the specified tap in the Raw 9-Tap mode.

Conditions

Must be in Raw 9-Tap mode (1) and connected.

Group

Raw 9-Tap

Syntax

PI_tap9R_tap_enable <unit>,<channel>,<tap, tap-state>

PI_tap9R_tap_enable_Q <unit>,<channel>, <tap>

Arguments

<unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<tap> ::= <NR1>, a tap number between 1 and 9.

<tap, tap-state>, the tap number, where the tap-state is a boolean value. 1 is enabled; 0 is disabled.

Returns

The tap number and a <Boolean> value for the tap state.

Examples

PI_tap9R_tap_enable_Q 2,2,2 might return 0 indicating that tap #2 is disabled on Channel 2 linear of equalizer #2.

PI_tap9R_tap_enable_all

Enable or disables all taps in the Raw 9-Tap mode.

Conditions

The command requires the enable/disable states of all taps to be specified.

Must be in Raw 9-Tap mode (1) and connected.

Group

Raw 9-Tap

Syntax

PI_tap9R_tap_enable_all <unit>,<channel>,<enable>

PI_tap9R_tap_enable_all_Q <unit>,<channel>

Arguments

<unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<enable> ::= <Boolean>, where 1 enables all taps and 0 disables all taps.

Returns A <Boolean> value.

Examples PI_tap9R_tap_enable_all 2,1,1,1 enables all taps on Channel 1 of linear equalizer #2 in the Raw 9-Tap mode.

PI_tap9R_tap_val

Sets or returns the value of the specified tap in the Raw 9-Tap mode.

Conditions Must be in Raw 9-Tap mode (1) and connected.

Group Raw 9-Tap

Syntax PI_tap9R_tap_val <unit>,<channel>,<tap, tap-value>

PI_tap9R_tap_val_Q <unit>,<channel>, <tap>

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<tap> ::= <NR1>, a tap number between 1 and 9.

<tap, tap-value>, the tap number, where tap-value is an integer between -63 and 63.

Returns The tap number and an <NR1> tap value.

Examples PI_tap9R_tap_val 2,2,2,45 sets tap #2 on linear equalizer #2, Channel 2, to 45.

SPM Commands

The following table summarizes the SPM mode commands.

Command	Inputs	Outputs
PI_spm_agc_gain	int unit, int channel, double gain_V	int e_num, string e_str
PI_spm_agc_gain_Q	int unit, int channel	double gain_V, int e_num, string e_str
PI_spm_agc_lev	int unit, int channel, int agcLevel	int e_num, string e_str
PI_spm_agc_lev_Q	int unit, int channel	int agcLevel, int e_num, string e_str
PI_spm_agc_state	int unit, int channel, bool lockIt	int e_num, string e_str
PI_spm_agc_state_Q	int unit, int channel	bool isLocked, int e_num, string e_str
PI_spm_emu_cuttoff	int unit, int channel, double cutOff	int e_num, string e_str
PI_spm_emu_cuttoff_Q	int unit, int channel	double cutOff, int e_num, string e_str
PI_spm_emu_rolloff	int unit, int channel, double rollOff	int e_num, string e_str
PI_spm_emu_rolloff_Q	int unit, int channel	double rollOff, int e_num, string e_str
PI_spm_enable_reldelay	int unit, int channel, bool enabled	int e_num, string e_str
PI_spm_enable_reldelay_Q	int unit, int channel	bool enabled, int e_num, string e_str
PI_spm_process ***	int unit, int channel, int outPort, int inPort	bool emuOk, int e_num, string e_str
PI_spm_offs_state	int unit, int channel, bool setAuto	int e_num, string e_str
PI_spm_offs_state_Q	int unit, int channel	bool isAuto, int e_num, string e_str
PI_spm_offs_val	int unit, int channel, double offset_mV	int e_num, string e_str
PI_spm_out_amp	int unit, int chan, double ampMv	int e_num, string e_str
PI_spm_out_amp_Q	int unit, int chan	double aMv, int e_num, string e_str
PI_spm_out_cal_Q	int unit, int chan	bool isCalib, nt e_num, string e_str
PI_spm_offs_val_Q	int unit, int channel	double offset_mV, int e_num, string e_str
PI_spm_out_enable	int unit, int channel, bool isON	int e_num, string e_str
PI_spm_out_enable_Q	int unit, int channel	bool isON, int e_num, string e_str
PI_spm_reldelay	int unit, int channel, double relDelay	int e_num, string e_str
PI_spm_reldelay_Q	int unit, int channel	double relDelay, int e_num, string e_str
PI_spm_set_spmfile ***	int unit, int channel, string filePath	int e_num, string e_str
PI_spm_statefile_load ***	int unit, int channel, string filePath	bool compatible, int e_num, string e_str
PI_spm_statefile_save ***	int unit, int channel, string filePath, bool overwrite	int e_num, out string e_str

SPM command preconditions

The programming or reading of SPM values should only be performed after the following precondition commands have succeeded:

- A valid S-parameter file has been entered using PI_spm_set_spmfile.
- The transform from S-parameters to 9-tap values using PI_spm_process has succeeded as evidenced by emuOK set to true.

In the above table, only commands identified by the three asterisks (***) are independent of the SPM command preconditions.

Until valid S-parameters have been entered and successfully emulated with new 9-tap values, the tap settings will be at random, pre-existing, or default values. Setting (or reading back) amplitude, offset, AGC, or other controls will not represent or approximate the desired S-parameter response unless the emulation has succeeded as indicated by emuOK set true.

The programming interface does not prevent you from executing most commands before the above conditions are met. However, the LE320 tap values, and all values affected by tap settings, will not represent the S-parameter response. Taps, and all parameters affected by taps, will be at default, random, or pre-existing values.

Similarly, the commands that modify an emulation, PI_spm_emu_cutoff and PI_spm_emu_rolloff, have no meaning unless a valid emulation exists.

PI_spm_agc_gain

Sets or returns the value of the AGC Adjust control.

Conditions

Only valid when the AGC state is set to Lock.

Must be in SPM mode (4) and connected.

See *SPM command preconditions* on page 90.

Group

SPM

Syntax

PI_spm_agc_gain <unit>,<channel>, <gain>

PI_spm_agc_gain_Q <unit>,<channel>

Related commands

PI_spm_agc_state

Arguments

<unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<gain> ::= <NRf>, the uncalibrated gain adjustment value in the range from 1.80 to 2.50.

Returns A single <NRf> number.

Examples PI_spm_agc_gain 1,1,2.10 sets the AGC Gain adjust value to 2.10 for Channel 1 of linear equalizer #1.

PI_spm_agc_lev

Sets or returns the 9-Tap automatic gain control level.

When the PI_spm_agc_state is OFF or 0, the AGC chip circuitry has eight ranges.

Conditions Must be in SPM mode (4) and connected.

See *SPM command preconditions* on page 90.

Group SPM

Syntax PI_spm_agc_lev <unit>,<channel>, <level>

PI_spm_agc_lev_Q <unit>,<channel>

Related commands *PI_spm_agc_state*

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<level> ::= <NR1> the AGC level; a number from 0 to 7.

Returns A single <NR1> value.

Examples PI_spm_agc_lev_Q 1,1 might return 3 indicating that the AGC level for Channel 1 of linear equalizer #1 is set to 3.

PI_spm_agc_state

Set or returns automatic gain control lock status.

Conditions

Must be in SPM mode (4) and connected.

See *SPM command preconditions* on page 90.

Group

SPM

Syntax

PI_spm_agc_state <unit>,<channel>, {0|1}

PI_spm_agc_state_Q <unit>,<channel>

Related commands

PI_spm_agc_lev, *PI_spm_agc_gain*

Arguments

<unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

0 indicates that the lock is false and the AGC is controlled by the chip ALC (automatic level control) loop.

1 indicates that the lock is true and enables the UNCAL state or warning.

Returns

A single <Boolean> value.

Examples

PI_spm_agc_state 1,1 enables the automatic gain control lock and enables the UNCAL state or warning on Channel 1 of linear equalizer #1.

PI_spm_emu_cutoff

Sets or returns the maximum cutoff frequency in SPM mode.

Conditions

The cutoff frequency depends on the instrument type, an LE160 or LE320.

Must be in SPM mode (4) and connected.

See *SPM command preconditions* on page 90.

Group

SPM

Syntax

SPI_spm_emu_cutoff <unit>,<channel>,<fmax>

SPI_spm_emu_cutoff_Q <unit>,<channel>

Related commands

PI_spm_emu_rolloff

Arguments

<unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<fmax> ::= <NRf>, maximum cutoff frequency.

Returns

A single <NRf> value.

Examples

SPI_spm_emu_cutoff 1,1,2.5E10 sets the maximum cutoff frequency to 25 GHz.

PI_spm_emu_rolloff

Sets or returns the Add roll-off value in SPM mode.

Use non-zero values to linearly roll off the magnitude of the S-parameter data for a better simulation. This preserves the phase response in the S-parameter data. Enter negative numbers to roll-up the magnitude at frequencies above the cut-off frequency.

Conditions

Must be in SPM mode (4) and connected.

See *SPM command preconditions* on page 90.

Group

SPM

Syntax

PI_spm_emu_rolloff <unit>,<channel>,<roll-off>

PI_spm_emu_rolloff_Q <unit>,<channel>

Related commands

PI_spm_emu_cuttoff

Arguments

<unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<roll-off> ::= <NRf>, roll-off value in dB/GHz.

Returns

A single <NRf> number.

Examples

PI_spm_emu_rolloff_Q 1,1 might return 0.00 indicating that the roll-off value for Channel 1 of linear equalizer #1 was 0.00 dB/GHz.

PI_spm_enable_reldelay

Sets or returns the ON/OFF state of the Relative Delay control in SPM mode.

Conditions

Must be in SPM mode (4) and connected.

See *SPM command preconditions* on page 90.

Group

SPM

Syntax

PI_spm_enable_reldelay <unit>,<channel>,{1|0}

PI_spm_enable_reldelay_Q <unit>,<channel>

Related commands

PI_spm_reldelay

Arguments

<unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<state> ::= <Boolean>, a 1 enables the relative delay; a 0 disables the relative delay.

Returns

A single <Boolean> value. A 1 indicates that the relative delay is on; a 0 indicates that the relative delay is off.

Examples

PI_spm_enable_reldelay 1,1,1 enables the relative delay on Channel 1 of linear equalizer #1.

PI_spm_process

Computes a set of values for the nine hardware taps based on the S-parameter data as modified by the emulation control settings. Upon completion of the command, an emulation status will be returned indicating a successor failure.

Conditions Must be in SPM mode (4) and connected.

Group SPM

Syntax PI_spm_process <unit>,<channel>,<outPort>,<inPort>

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<outPort> ::= <NR1>, specifies the stimulus port.

<inPort> ::= <NR1>, specifies the receive port.

Returns <emulation_Valid>, a boolean value where 1 indicates emulation success and 0 indicates emulation failure.

Examples PI_spm_process 1,1,1,2 calculates a set of tap values for Channel 1 of liner equalizer #1 for output port 1 and input port 2.

PI_spm_offs_state

Sets or returns the state of the input offset control in the SPM mode.

Conditions Must be in SPM mode (4) and connected.

See *SPM command preconditions* on page 90.

Group SPM

Syntax PI_spm_offs_state <unit>,<channel>,{1|0}

PI_spm_offs_state_Q <unit>,<channel>

Related commands *PI_spm_offs_val*

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<state> ::= <Boolean>. 1 sets the input offset control to AUTO; 0 sets the input offset control to manual control.

Returns

A single <Boolean> value. 1 indicates that the input offset control is set to AUTO; 0 indicates that the input offset control is set to manual control.

Examples

PI_spm_offs_state 1,1,0 sets the input offset control for Channel 1 of linear equalizer #1 to manual control.

PI_spm_offs_val

Sets or returns the input offset value in the SPM mode in millivolts.

Conditions

If the input offset control is set to AUTO, any warnings are ignored.

Must be in SPM mode (4) and connected.

See *SPM command preconditions* on page 90.

Group

SPM

Syntax

PI_spm_offs_val <unit>,<channel>, <offset>

PI_spm_offs_val_Q <unit>,<channel>

Related commands

PI_spm_offs_state

Arguments

<unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<offset> ::= <double>, the input offset value, a number between -78 and 78.

Returns

A single <NRf> value. the returned value is in millivolts.

Examples

PI_spm_offs_val 1,1,20 sets the input offset for Channel 1 of linear equalizer #1 to 20 mv.

PI_spm_out_amp

Sets or returns the output amplitude value in the SPM mode. The value is in $mV_{p\text{-}p}$.

Conditions

Must be in SPM mode (4) and connected.

See *SPM command preconditions* on page 90.

The value might be calibrated when the AGC Lock control is off. If the AGC Lock control is on, the value is uncalibrated.

Use the query (PI_spm_out_cal_Q) to determine if an amplitude value was able to be calibrated. Refer to *Appendix B: Theory of Operation, Output amplitude*, for conditions which affect amplitude calibration.

Group

SPM

Syntax

PI_spm_out_amp <unit>,<channel>, <value>

PI_spm_out_amp_Q <unit>,<channel>

Related commands

PI_tap4_agc_state, *PI_spm_out_cal_Q*

Arguments

<unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<value> ::= <double>, the value of the output control from 0 to 6000 (equivalent to 6.0 volts, but must be specified in mV).

Returns

A single <double> value.

Examples

PI_spm_out_amp_Q 1,1 might return 0 indicating that the output amplitude for Channel 1 of linear equalizer #1 is 0 $mV_{p\text{-}p}$.

PI_spm_out_cal_Q

Use this query to determine if the output amplitude setting resulted in a calibrated value. It returns the state of the Output Amplitude control in the SPM mode. When the AGC Lock control is on, this query always returns false (uncalibrated).

Conditions

Must be in SPM mode (4) and connected.

See *SPM command preconditions* on page 90.

Group

SPM

Syntax

PI_spm_out_cal_Q <unit>,<channel>

Related commands

PI_spm_agc_state, *PI_spm_out_amp*

Arguments

<unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

Returns

A single <Boolean> value. A 1 indicates that the Output Amplitude control is calibrated. A 0 indicates that the Output Amplitude control is uncalibrated and the label is red.

Examples

PI_spm_out_cal_Q 1,1 might return 1 indicating that the Output Amplitude control for Channel 1 of linear equalizer #1 is calibrated (the AGC Lock control is off).

PI_spm_out_enable

This command sets the output-enable pin of the LE chip for the specified channel. It enables or disables the output state in the SPM mode.

Conditions

Must be in SPM mode (4) and connected.

See *SPM command preconditions* on page 90.

Group

SPM

Syntax

PI_spm_out_enable <unit>,<channel>,{1|0}

PI_spm_out_enable_Q <unit>,<channel>

Arguments

<unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<output-state> ::= <boolean>, 1 enables the output state and 0 disables the output state.

Returns

A single <boolean> value. 1 indicates that the output state is enabled or on; 0 indicates that the output state is disabled or off.

Examples

PI_spm_out_enable 1,1,0 enables the Channel 1 output of linear equalizer #1.

PI_spm_reldelay

Sets or returns the relative delay in ps in the SPM mode.

Conditions

Only valid when the relative delay is turned on.

Must be in SPM mode (4) and connected.

See *SPM command preconditions* on page 90.

Group

SPM

Syntax

SPI_spm_reldelay <unit>,<channel>,<delay>

PI_spm_reldelay_Q <unit>,<channel>

Related commands

PI_spm_enable_reldelay

Arguments

<unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<delay> ::= <NR1>, a value between -70 and 70.

Returns A single <NR1> value.

Examples PI_spm_reldelay 1,2,35 sets the relative delay for Channel 2 of linear equalizer #1 to 35 ps.

PI_spm_set_spmfile (Command only)

Sets the file path name of the S-parameter file.

Conditions Must be in SPM mode (4) and connected.

Group SPM

Syntax PI_spm_set_spmfile <unit>,<channel>,<path-filepath

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<path-filename> ::= <string>

Examples PI_spm_set_spmfile 1,1, "C\Users|Documents|Tektronix-LE\TekLE-SPM-file1" sets the path name to C\Users|Documents|Tektronix-LE.

PI_spm_statefile_load (Command only)

Loads a previously saved XML state file on the host computer.

Conditions

The file is loaded from the folder specified in the Connect view.

Must be in SPM mode (4) and connected.

Group

SPM

Syntax

PI_spm_statefile_load <unit>,<channel>, <path-filename>

Related commands

PI_spm_statefile_save

Arguments

<unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<path-filename> ::= <string>

Examples

PI_spm_statefile_load 1,2,"C:\Users\Documents\Tektronix-LE\TekLE-SPM-State-Qualtest1.xml" loads the TekLE-SPM-State-Qualtest1.xml file.

PI_spm_statefile_save (Command only)

Saves a state file on the host computer.

The file is saved in the folder locations specified in the Connect view.

Conditions

Must be in SPM mode (4) and connected.

Group

SPM

Syntax

PI_spm_statefile_save <unit>,<channel>, <path-filename>.

Related commands

PI_spm_statefile_load

Arguments

<unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<path-filename> ::= <string>

Examples

PI_spm_statefile_save 1,2,"C:\Users\Documents\Tektronix-LE\TekLE-SPM-State-test4-18.xml" saves the TekLE-SPM-State-test4-18.xml file to the C:Users \Documents\Tektronix-LE folder on the host computer.

CDS Commands

The following table summarizes the CDS mode commands.

Command	Inputs	Outputs
PI_cds_add_pole	int unit, int channel, double real, double imaginary	int e_num, string e_str
PI_cds_add_zero	int unit, int channel, double real, double imaginary	int e_num, string e_str
PI_cds_agc_gain	int unit, int channel, double gain_V	int e_num, string e_str
PI_cds_agc_gain_Q	int unit, int channel	out double gain_V, int e_num, string e_str
PI_cds_agc_lev	int unit, int channel, int agcLevel	int e_num, string e_str
PI_cds_agc_lev_Q	int unit, int channel	int agcLevel, int e_num, string e_str
PI_cds_agc_state	int unit, int channel, bool lockIt	int e_num, string e_str
PI_cds_agc_state_Q	int unit, int channel	bool isLocked, int e_num, string e_str
PI_cds_emu_cuttoff	int unit, int channel, double cutOff	nt e_num, string e_str
PI_cds_emu_cuttoff_Q	int unit, int channel	double cutOff, int e_num, string e_str
PI_cds_emu_rolloff	int unit, int channel, double rollOff	int e_num, string e_str
PI_cds_emu_rolloff_Q	int unit, int channel	double rollOff, int e_num, string e_str
PI_cds_enable_reldelay	int unit, int channel, bool enabled	int e_num, string e_str
PI_cds_enable_reldelay_Q	int unit, int channel	bool enabled, int e_num, string e_str
PI_cds_offs_state	int unit, int channel, bool setAuto	int e_num, string e_str
PI_cds_offs_state_Q	int unit, int channel	bool isAuto, int e_num, string e_str
PI_cds_offs_val	int unit, int channel, double offset_mV	int e_num, string e_str
PI_cds_offs_val_Q	int unit, int channel	double offset_mV, int e_num, string e_str
PI_cds_out_amp	int unit, int channel, double ampV	int e_num, string e_str
PI_cds_out_amp_Q	int unit, int channel	double ampV, int e_num, string e_str
PI_cds_out_cal_Q	int unit, int channel	bool calib, int e_num, string e_str
PI_cds_out_enable	int unit, int channel, bool isON	int e_num, string e_str
PI_cds_out_enable_Q	int unit, int channel	bool isON, int e_num, string e_str
PI_cds_pz_clear	int unit, int channel	int e_num, string e_str
PI_cds_pz_process	int unit, int channel	bool enuOk, int e_num, string e_str
PI_cds_reldelay	int unit, int channel, double relDelay	int e_num, string e_str
PI_cds_reldelay_Q	int unit, int channel	double relDelay, int e_num, string e_str
PI_cds_statefile_save	int unit, int channel, string filePath, bool overwrite	int e_num, string e_str
PI_cds_statefile_load	int unit, int channel, string filePath	bool compatible, int e_num, string e_str

PI_cds_add_pole (Command only)

Sets the real and imaginary pole locations in GHz.

Use multiple commands to build an array of poles and zeros.

Conditions None

Group CDS

Syntax PI_cds_add_pole <unit>,<channel>,<real>, <imaginary>

Related commands *PI_cds_add_zero*, *PI_cds_pz_clear*

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<real> ::= <NRf>, the real component of the pole.

<imaginary> ::= <NRf>, the imaginary component of the pole.

Returns A single <Boolean> value, 1 if successfully added or 0 if not.

Examples PI_cds_add_pole 1,1,-3.00,0 defines the pole location with a real component of -3.00 GHz and an imaginary component of 0 on Channel 1 of linear equalizer #1.

PI_cds_add_zero (Command only)

Sets the real and imaginary zero locations in GHz.

Use multiple commands to build an array of poles and zeros.

Conditions None

Group CDS

Syntax PI_cds_add_zero <unit>,<channel>,<real>, <imaginary>

Related commands *PI_cds_add_pole*, *PI_cds_pz_clear*

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<real> ::= <NRf>, the real component of the zero.

<imaginary> ::= <NRf>, the imaginary component of the zero.

Returns A single <Boolean> value, 1 if successfully added or 0 if not.

Examples PI_cds_add_zero 1,1,-5.00,0 defines the zero location with a real component of -5.00 GHz and an imaginary component of 0 on Channel 1 of linear equalizer #1.

PI_cds_agc_gain

Sets or returns the value of the AGC adjust control in the CDS mode.

Conditions Only valid when the AGC Lock is enabled.

Group CDS

Syntax PI_cds_agc_gain <unit>,<channel>

PI_cds_agc_gain_Q <unit>,<channel>

Related commands *PI_cds_agc_state*

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<gain> ::= <NRf>, the uncalibrated gain adjustment value in the range from 1.80 to 2.50.

Returns A single <NRf> number.

Examples PI_cds_agc_gain 1,1,2.10 sets the AGC Gain adjust value to 2.10 for Channel 1 of linear equalizer #1.

PI_cds_agc_lev

Sets or returns the automatic gain control level in the CDS mode.

When the PI_tap9_agc_state is OFF or 0, the label in the user interface is set to "AGC Level" and the AGC chip circuitry has seven ranges.

Conditions None

Group CDS

Syntax PI_cds_agc_lev <unit>,<channel>, <level>

PI_cds_agc_lev_Q <unit>,<channel>

Related commands *PI_cds_agc_state*

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<level> ::= <NR1> the AGC level; a number from 0 to 7.

Returns A single <NR1> value.

Examples PI_cds_agc_lev_Q 1,1 might return 3 indicating that the AGC level for Channel 1 of linear equalizer #1 is set to 3.

PI_cds_agc_state

Set or returns automatic gain control lock status in the CDS mode.

Conditions None

Group CDS

Syntax PI_cds_agc_state <unit>,<channel>, {0|1|OFF|ON}

PI_cds_agc_state_Q <unit>,<channel>

Related commands *PI_cds_agc_lev*, *PI_cds_agc_gain*, *PI_cds_out_cal_Q*

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

OFF or 0 indicates that the lock is false and the AGC is controlled by the chip loop.

ON or 1 indicates that the lock is true and enables the UNCAL state or warning.

Returns A single <Boolean> value.

Examples PI_cds_agc_state 1,1 enables the automatic gain control lock and enables the UNCAL state or warning on Channel 1 of linear equalizer #1.

PI_cds_emu_cutoff

Sets or eturns the maximum cutoff frequency in CDS mode.

Conditions

The cutoff frequency depends on the instrument type, an LE160 or LE320.

Must be in CDS mode (5) and connected.

Group

CDS

Syntax

PI_cds_emu_cutoff <unit>,<channel>,<fmax>

PI_cds_emu_cutoff_Q <unit>,<channel>

Arguments

<unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<fmax> ::= <NRf> value.

Returns

A single <NRf> value.

Examples

PI_cds_emu_cutoff_Q 1,1, might return 2.5E10 indicating that the maximum cutoff frequency is 25 GHz.

PI_cds_emu_rolloff

Sets or returns the Add roll-off value.

Use this command to emulate linear-phase impairments with a linear slope.

Conditions

To obtain linear phase response, execute the PI_cds_pz_clear command before adjusting the roll-off value. The later command removes all poles and zeros and sets the emulation engine for a flat frequency response and a flat phase response. This clears the arrays first and allows you to see the effects of just the roll-off specified by this command.

Group

CDS

Syntax

PI_cds_emu_rolloff <unit>,<channel>,<roll-off>

PI_cds_emu_rolloff_Q <unit>,<channel>

Arguments

<unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<roll-off> ::= <NRf>, roll-off value in dB/GHz.

Returns A single <NRf> number.

Examples PI_cds_emu_rolloff_Q 1,1 might return 0.00 indicating that the roll-off value for Channel 1 of linear equalizer #1 was 0.00 dB/GHz.

PI_cds_enable_reldelay

Sets or returns the ON/OFF state of the Relative Delay control in CDS mode.

Conditions None

Group CDS

Syntax PI_cds_enable_reldelay <unit>,<channel>,{1|0|ON|OFF}

PI_cds_enable_reldelay_Q <unit>,<channel>

Related commands *PI_cds_reldelay*

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<state> ::= <Boolean>, a 1 or ON enables the relative delay; a 0 or OFF disables the relative delay.

Returns A single <Boolean> value. A 1 or ON indicates that the relative delay is on; a 0 or OFF indicates that the relative delay is off.

Examples PI_cds_enable_reldelay 1,1,1 enables the relative delay on Channel 1 of linear equalizer #1.

PI_cds_reldelay

Sets or returns the relative delay in ps in the CDS mode.

Conditions Only valid when the relative delay is turned on.

Group CDS

Syntax PI_cds_reldelay <unit>,<channel>,<delay>

PI_cds_reldelay_Q <unit>,<channel>

Related commands *PI_cds_enable_reldelay*

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<delay> ::= <double>, a value between -70 and 70.

Returns A single <NRf> value.

Examples PI_cds_reldelay 1,2,35 sets the relative delay for Channel 2 of linear equalizer #1 to 35 ps.

PI_cds_offs_state

Sets or returns the state of the input offset control in the CDS mode.

Conditions None

Group CDS

Syntax PI_cds_offs_state <unit>,<channel>,{1|0|ON|OFF}

PI_cds_offs_state_Q <unit>,<channel>

Related commands *PI_cds_offs_val*

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<state> ::= <Boolean>. 1 or ON sets the input offset control to AUTO; 0 or OFF Sets the input offset control to manual control.

Returns A single <Boolean> value. 1 or ON indicates that the input offset control is set to AUTO; 0 or OFF indicates that the input offset control is set to manual control.

Examples PI_cds_offs_state 1,1,0 sets the input offset control for Channel 1 of linear equalizer #1 to manual control.

PI_cds_offs_val

Sets or returns the input offset value in the CDS mode in millivolts.

Conditions If the input offset control is set to AUTO, any warnings are ignored.

Group CDS

Syntax PI_cds_offs_val <unit>,<channel>, <offset>

PI_cds_offs_val_Q <unit>,<channel>

Related commands *PI_cds_offs_state*

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<offset> ::= <double>, the input offset value, a number between -78 and 78.

Returns A single <NRf> value. the returned value is in millivolts.

Examples PI_cds_offs_val 1,1,20 sets the input offset for Channel 1 of linear equalizer #1 to 20 mv.

PI_cds_out_amp

Sets or returns the Output Amplitude value in the CDS mode. The value is in mV_{p-p}.

Conditions The value is calibrated when the AGC Lock control is off. If the AGC Lock control is on, the value is uncalibrated.

Group 9-Tap

Syntax PI_cds_out_amp <unit>,<channel>, <value>

PI_cds_out_amp_Q <unit>,<channel>

Related commands *PI_cds_out_enable*

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<value> ::= <double>, the value of the output control from 0 to 6.00.

Returns A single <NRf> value.

Examples PI_cds_out_amp_Q 1,1 might return 0 indicating that the Output Amplitude for Channel 1 of linear equalizer #1 is 0 V_{p-p}.

PI_cds_out_cal_Q (Query only)

Returns the state of the output amplitude control in the CDS mode. When the AGC Lock control is on, the output amplitude is uncalibrated.

Conditions List conditions if any.

Group CDS

Syntax PI_cds_out_cal_Q <unit>,<channel>

Related commands *PI_cds_agc_state*

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

Returns A single <Boolean> value. A 1 indicates that the Output Amplitude control is calibrated. A 0 indicates that the Output Amplitude control is uncalibrated and the label is red.

Examples PI_cds_out_cal_Q 1,1 might return 1 indicating that the Output Amplitude control for Channel 1 of linear equalizer #1 is calibrated (the AGC Lock control is off).

PI_cds_out_enable

Enables or disables the output state in the CDS mode.

Conditions None

Group CDS

Syntax PI_cds_out_enable <unit>,<channel>,{1|0|ON|OFF}

PI_cds_out_enable_Q <unit>,<channel>

Related commands *PI_cds_out_amp*

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<output-state> ::= <boolean>, 1 or ON enables the output state and 0 or OFF disables the output state.

Returns A single <boolean> value. 1 indicates that the output state is enabled or on; 0 or OFF indicates that the output state is disabled or off.

Examples PI_cds_out_enable 1,1,0 enables the Channel 1 output of linear equalizer #1.

PI_cds_pz_clear (Command only)

The poles and zeros of the CDS mode are stored in the instrument state as a complex array. The array elements are accumulated one at a time, using the related command for entering a single pole or zero (see the Related commands below).

This command is useful to clear all poles and zeros from the instrument state.

It is important to realize that when the pole and zero arrays are cleared, the emulation engine sets the target frequency response for zero loss and zero delay, also known as a flat frequency response.

An example of using this function might be when you need to set the relative delay and to see the effects, independent of the current pole-zero values.

Conditions Must be in CDS mode.

Group CDS

Syntax PI_cds_pz_clear <unit>,<channel>

Related commands *PI_cds_add_pole*, *PI_cds_add_zero*

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

Examples PI_cds_pz_clear 1,1 clears the poles and zeros for Channel 1 of liner equalizer #1.

PI_cds_pz_process (Command only)

Computes a set of values for the hardware taps based on the CDS poles and zeros previously entered. Upon completion of the command, status of the emulation can be determined by reading the emuOK Boolean.

Conditions None

Group CDS

Syntax PI_cds_pz_process <unit>,<channel>,<emuOK>

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<emuOK> ::= <boolean>, 1 if successful, 0 if not successful.

Examples PI_cds_pz_process 1,1, emuOK calculates a set of tap values for Channel 1 of liner equalizer #1 and returns 1 if the operation is successful.

PI_cds_statefile_load (Command only)

Loads a previously saved XML state file on the host computer. The state file for this mode holds the poles and zeros entered or cleared by the related PI commands (add pole, add zero, clear all poles and zeros).

Conditions The file is loaded from the folder specified in the Connect view.

Group CDS

Syntax PI_cds_statefile_load <unit>,<channel>, <path-filename>

Related commands *PI_cds_statefile_save*, *PI_cds_add_pole*, *PI_cds_add_zero*, *PI_cds_pz_clear*

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<path-filename> ::= <string>

Examples PI_cds_statefile_load 1,2,"C:\Users\Documents\Tektronix-LE\TekLE-CHM-State-test2.xml" loads the TekLE-CHM-State-test2.xml file.

PI_cds_statefile_save (Command only)

Saves a state file on the host computer.

The state file for this mode holds the poles and zeros entered or cleared by the related PI commands (add pole, add zero, clear all poles and zeros).

The file is saved in the folder locations specified in the Connect view.

Conditions None

Group CDS

Syntax PI_cds_statefile_save <unit>,<channel>, <path-filename>

Related commands *PI_cds_statefile_load*, *PI_cds_add_pole*, *PI_cds_add_zero*, *PI_cds_pz_clear*

Arguments <unit> ::= <NR1>, a value between 1 and 4.

<channel> ::= <NR1>, channel 1 or 2.

<path-filename> ::= <string>.

Examples PI_cds_statefile_save 1,2,"C:\Users\Documents\Tektronix-LE\TekLE-CHM-State-test4-18.xml" saves the TekLE-CHM-State-test4-18.xml file to the C:Users \Documents\Tektronix-LE folder on the host computer.

Index