M8000 Series of BER Test Solutions

M8070ADVB Advanced Measurement Package

KEYSIGHT

USER GUIDE

Notices

Trademarks

PCI Express® and PCIe® are registered trademarks of PCI-SIG.

Manual Part Number

M8000-91B10

Edition

Edition 9.0, November 2023

Keysight Technologies Deutschland GmbH
Herrenberger Strasse 130,
71034 Böblingen, Germany

Technology Licenses

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

U.S. Government Rights

The Software is “commercial computer software,” as defined by Federal Acquisition Regulation (“FAR”) 2.101. Pursuant to FAR 12.212 and 27.405-3 and Department of Defense FAR Supplement

(“DFARS”) 227.7202, the U.S. government acquires commercial computer software under the same terms by which the software is customarily provided to the public. Accordingly, Keysight provides the Software to U.S. government customers under its standard commercial license, which is embodied in its End User License Agreement (EULA), a copy of which can be found at http://www.keysight.com/find/sweula. The license set forth in the EULA represents the exclusive authority by which the U.S. government may use, modify, distribute, or disclose the Software. The EULA and the license set forth therein, does not require or permit, among other things, that Keysight: (1) Furnish technical information related to commercial computer software or commercial computer software documentation that is not customarily provided to the public; or (2) Relinquish to, or otherwise provide, the government rights in excess of these rights customarily provided to the public to use, modify, reproduce, release, perform, display, or disclose commercial computer software or commercial computer software documentation. No additional government requirements beyond those set forth in the EULA shall apply, except to the extent that those terms, rights, or licenses are explicitly required from all providers of commercial computer software pursuant to the FAR and the DFARS and are set forth specifically in writing elsewhere in the EULA. Keysight shall be under no obligation to update, revise or otherwise modify the Software. With respect to any technical data as defined by FAR 2.101, pursuant to FAR 12.211 and 27.404.2 and DFARS 227.7102, the U.S. government acquires no greater than Limited Rights as defined in FAR 27.401 or DFAR 227.7103-5 (c), as applicable in any technical data.

Warranty

THE MATERIAL CONTAINED IN THIS DOCUMENT IS PROVIDED "AS IS," AND IS SUBJECT TO BEING CHANGED, WITHOUT NOTICE, IN FUTURE EDITIONS. FURTHER, TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, KEYSIGHT DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED WITH REGARD TO THIS MANUAL AND ANY INFORMATION CONTAINED HEREIN, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. KEYSIGHT SHALL NOT BE LIABLE FOR ERRORS OR FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, USE, OR PERFORMANCE OF THIS DOCUMENT OR ANY INFORMATION CONTAINED HEREIN. SHOULD KEYSIGHT AND THE USER HAVE A SEPARATE WRITTEN AGREEMENT WITH WARRANTY TERMS COVERING THE MATERIAL IN THIS DOCUMENT THAT CONFLICT WITH THESE TERMS, THE WARRANTY TERMS IN THE SEPARATE AGREEMENT WILL CONTROL.

Safety Notices

CAUTION

A CAUTION notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in damage to the product or loss of important data. Do not proceed beyond a CAUTION notice until the indicated conditions are fully understood and met.

WARNING

A WARNING notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in personal injury or death. Do not proceed beyond a WARNING notice until the indicated conditions are fully understood and met.

Safety Summary

The following general safety precautions must be observed during all phases of operation of this instrument. Failure to comply with these precautions or with specific warnings or operating instructions in the product manuals violates safety standards of design, manufacture and intended use of the instrument. Keysight Technologies assumes no liability for the customer's failure to comply with these requirements. Product manuals are provided with your instrument on CD-ROM and/or in printed form. Printed manuals are an option for many products. Manuals may also be available on the Web. Go to www.keysight.com and type in your product number in the Search field at the top of the page.

General

This product is a Safety Class 1 instrument (provided with a protective earth terminal). The protective features of this product may be impaired if it is used in a manner not specified in the operation instructions.

All Light Emitting Diodes (LEDs) used in this product are Class 1 LEDs as per IEC 60825-1.

Environment Conditions

This instrument is intended for indoor use in an overvoltage category II, pollution degree 2 environment. It is designed to operate at a maximum relative humidity of 95% and at altitudes of up to 2000 meters.

Refer to the specifications tables for the ac mains voltage requirements and ambient operating temperature range.

Before Applying Power

Verify that all safety precautions are taken. The power cable inlet of the instrument serves as a device to disconnect from the mains in case of hazard. The instrument must be positioned so that the operator can easily access the power cable inlet. When the instrument is rack mounted the rack must be provided with an easily accessible mains switch.

Ground the Instrument

To minimize shock hazard, the instrument chassis and cover must be connected to an electrical protective earth ground. The instrument must be connected to the ac power mains through a grounded power cable, with the ground wire firmly connected to an electrical ground (safety ground) at the power outlet. Any interruption of the protective (grounding) conductor or disconnection of the protective earth terminal will cause a potential shock hazard that could result in personal injury.

Do Not Operate in an Explosive Atmosphere

Do not operate the instrument in the presence of flammable gases or fumes.

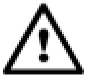
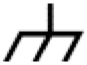
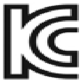
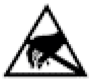
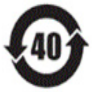
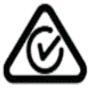
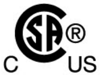
Do Not Remove the Instrument Cover

Operating personnel must not remove instrument covers. Component replacement and internal adjustments must be made only by qualified personnel.

Instruments that appear damaged or defective should be made inoperative and secured against unintended operation until they can be repaired by qualified service personnel.

Safety Symbols

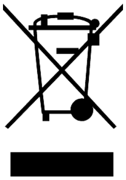
Table 1 Safety Symbols

Symbol	Description
	Indicates warning or caution. If you see this symbol on a product, you must refer to the manuals for specific Warning or Caution information to avoid personal injury or damage to the product.
	Frame or chassis ground terminal. Typically connects to the equipment’s metal frame.
	KC is the Korean certification mark to demonstrate that the equipment is Class A suitable for professional use and is for use in electromagnetic environments outside of the home.
	Contains parts or assemblies susceptible to damage by electrostatic discharge (ESD). Use electrostatic discharge protective handling procedures to avoid malfunctions or potential damage to the instruments.
	Indicates the time period during which no hazardous or toxic substance elements are expected to leak or deteriorate during normal use. Forty years is the expected useful life of the product.
	The RCM Mark is a compliance mark to the ACMA (Australian Spectrum Management Agency). This indicates compliance with all Australian EMC regulatory information.
	Indicates that the product was tested and has met the certification requirements for electrical, plumbing and/or mechanical products.

Symbol	Description
CAN ICES/NMB-001(A) ISM GRP 1-A	The CE mark is a registered trademark of the European Community. This CE mark shows that the product complies with all the relevant European Legal Directives. CAN ICES/NMB-001(A) - This ISM device complies with the Canadian ICES-001(A). Cet appareil ISM est conforme a la norme NMB-001(A) du Canada. ISM GRP 1-A - This is an Industrial Scientific and Medical (ISM) Group 1 Class A product.
	This symbol on all primary and secondary packaging indicates compliance to China standard GB 18455-2001.

Compliance and Environmental Information

Table 2 Compliance and Environmental Information

Safety Symbol	Description
	The crossed out wheeled bin symbol indicates that separate collection for waste electric and electronic equipment (WEEE) is required, as obligated by DIRECTIVE 2012/19/EU and other National legislation. See http://about.keysight.com/en/companyinfo/environment/takeback.shtml to understand your Trade in options with Keysight in addition to product takeback instructions.

Contents

3 Utilities

1 Introduction

This chapter provides an overview of M8070ADVB Advanced Measurement Package.

Introduction

About Advanced Measurement Package Plugin

The **Advanced Measurement Package** Plugin can be installed in the M8070B System Software. This plugin includes advanced measurements, utilities and configurations to support hardware. The **Advanced Measurement Package** requires license for its activation. For details on license, see “License Information” on page 18.

The Advanced Measurement Package Plugin includes the following:

- Advanced Measurements
  - Output timing measurement
  - Output level measurement
  - Jitter tolerance measurement
  - Eye diagram measurement
  - Parameter sweep measurement
- Utilities
  - SCPI Editor
  - DUT control interface
  - Script Editor
- Supported Configurations
  - Controlling N1076A/77A from M8070B
  - Controlling N1076B/78A from M8070B
  - Controlling a Real-Time Oscilloscope from M8070B
  - Controlling a UXR-Series Real-Time Oscilloscope from M8070B
  - Controlling 86108B/N1060A/N1094A/N1094B/N1092C/N1092E from M8070B

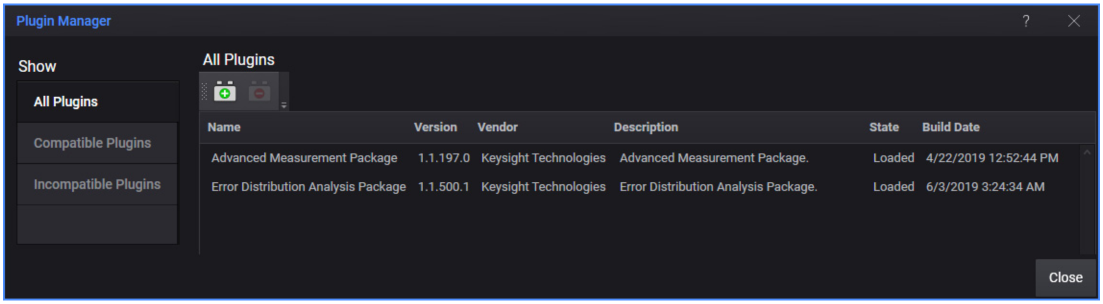
Prerequisites for Plugins Installation

- M8070B software version 6.0 or later installed.
- Advanced Measurement Package plugin file (*.M8KP) which can be downloaded from Keysight web page.
- A valid license for Advanced Measurement Package plugin. For details on the required licenses, please refer to the “License Information” on page 18.

Installing Advance Measurement Package Plugin

The M8070B software comes with a **Plugin Manager** to simplify all the tasks related to plugin management. The **Plugin Manager** displays a list of plugins that are installed in the software. For each plugin, it displays the information such as Name, Version, Vendor, Description, State and Build Date. In addition, the **Plugin Manager** also allows you to install, uninstall and upgrade the plugins.

To launch **Plugin Manager**, open the M8070B user interface and then go to the menu bar. Select **Utilities** and then **Manage Plug-ins...**. The **Plugin Manager** window appears as shown in the following figure:



Name	Version	Vendor	Description	State	Build Date
Advanced Measurement Package	1.1.197.0	Keysight Technologies	Advanced Measurement Package.	Loaded	4/22/2019 12:52:44 PM
Error Distribution Analysis Package	1.1.500.1	Keysight Technologies	Error Distribution Analysis Package.	Loaded	6/3/2019 3:24:34 AM

The **Plugin Manager** window enables you to install a plugin.

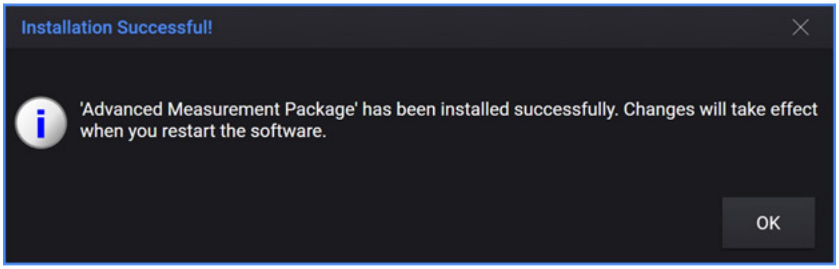
To do so, perform the following steps:

1 Download the required plugin file from the Keysight web page.

2 Click the **Install plugin from file** icon. A standard Window's **Open** dialog appears.

3 Locate the plugin file (*.M8KP) you want to install and click **Open**.

4 On the successful installation of plugin, a message similar to the following appears:



Installation Successful!

'Advanced Measurement Package' has been installed successfully. Changes will take effect when you restart the software.

5 Restart the M8070B software. Once you restart the software, the plugin state will change to **Loaded**.

NOTE

Ensure to restart the M8070B software for the changes to take effect.

For further details on how to update or uninstall plugins, please refer to the *M8000 Series Plugins Getting Started Guide*.

License Information

The usage of Advanced Measurement Package is govern by Keysight Licensing. Keysight Licensing provides tools and processes for floating, USB portable, node-locked, and transportable licenses.

The Keysight Licensing provides four types of licenses:

- **Node-locked** - A node-locked license permits the licensed software to run on only one machine. Each node-locked license is locked to an instrument or computer. Trial licenses are node-locked, time-based licenses.
- **USB portable** - A USB portable license is locked to a USB dongle (also called a USB key). Systems that run the licensed feature or product must have the license file resident on their hard disks, and have the dongle attached when they run the licensed feature or product.

Node-locked and USB portable licenses may be counted or uncounted.

- **Transportable** - A transportable license is a type of node-locked license that can be unlocked from one client host and then locked to another client host, via a network-enabled process performed in conjunction with the Keysight Software Manager website.
- **Floating** - Floating licenses (network licenses) reside on a license server (a separate computer) that can be used on another computer or instrument. Floating licenses can be borrowed for a specified number of days. Once you have borrowed a license, you can disconnect the licensed instrument or computer from the license server and continue to use the license offline for the duration of the borrow period. Some older floating licenses do not support borrowing.

Each license is either perpetual (permanent) or time-based (good for a limited amount of time).

These licenses can be installed using the **Keysight License Manager**. It helps you install licenses on your local machine (instrument or computer), or configure your local machine to use licenses from a remote license server.

Depending upon the license types, the following version of **Keysight License Manager** can be used to install the licenses:

- The node-locked and transportable licenses are installed by **Keysight License Manager 5**.
- The floating and USB probable licenses are installed by **Keysight License Manager 6**

NOTE

Please note that the Keysight License Manager 5 and Keysight License Manager 6 get installed on your system when you install M8070B system software.

For details on how to install these licenses, you can refer the following documents:

- M8000 Series User Guide (https://literature.cdn.keysight.com/litweb/pdf/M8000-91B08.pdf)
- Keysight Licensing Administrator's Guide (https://literature.cdn.keysight.com/litweb/pdf/5951-5739.pdf)

The following table shows the various licenses available for **Advanced Measurement Package**:

M8070ADVB-1FP	Advanced Measurement Package for M8000 Series BERT Test Solutions, node-locked perpetual license
M8070ADVB-1TP	Advanced Measurement Package for M8000 Series BERT Test Solutions, transportable perpetual license
M8070ADVB-1NP	Advanced Measurement Package for M8000 Series BERT Test Solutions, floating perpetual license
M8070ADVB-1UP	Advanced Measurement Package for M8000 Series BERT Test Solutions, USB portable perpetual license
M8070ADVB-1FL	Advanced Measurement Package for M8000 Series BERT Test Solutions, node-locked 12 month license
M8070ADVB-1TL	Advanced Measurement Package for M8000 Series BERT Test Solutions, transportable 12 month license
M8070ADVB-1NL	Advanced Measurement Package for M8000 Series BERT Test Solutions, floating 12 month license
M8070ADVB-1UL	Advanced Measurement Package for M8000 Series BERT Test Solutions, USB portable 12 month license
M8070ADVB-1FX	Advanced Measurement Package for M8000 Series BERT Test Solutions, node-locked 24 month license
M8070ADVB-1TX	Advanced Measurement Package for M8000 Series BERT Test Solutions, transportable 24 month license
M8070ADVB-1NX	Advanced Measurement Package for M8000 Series BERT Test Solutions, floating 24 month license
M8070ADVB-1UX	Advanced Measurement Package for M8000 Series BERT Test Solutions, USB portable 24month license
M8070ADVB-1FY	Advanced Measurement Package for M8000 Series BERT Test Solutions, node-locked 36 month license
M8070ADVB-1TY	Advanced Measurement Package for M8000 Series BERT Test Solutions, transportable 36 month license
M8070ADVB-1NY	Advanced Measurement Package for M8000 Series BERT Test Solutions, floating 36 month license
M8070ADVB-1UY	Advanced Measurement Package for M8000 Series BERT Test Solutions, USB portable 36 month license
M8070ADVB-1FF	Advanced Measurement Package for M8000 Series BERT Test Solutions, node-locked 6 month license
M8070ADVB-1TF	Advanced Measurement Package for M8000 Series BERT Test Solutions, transportable 6 month license
M8070ADVB-1NF	Advanced Measurement Package for M8000 Series BERT Test Solutions, floating 6 month license
M8070ADVB-1UF	Advanced Measurement Package for M8000 Series BERT Test Solutions, USB portable 6 month license
M8070ADVB-TRL	Advanced Measurement Package for M8000 Series BERT Test Solutions, 30 days free Trial

NOTE

The M8070ADVB license also enables the histogram-based eye diagram in the M8070B Sampling Point Setup window. For details, refer to the *M8000 Series User Guide* or *Online Help*.

Locate Documentation

To access documentation related to the M8070ADVB Advanced Measurement Package, use one of the following methods:

- Browse the product CD
- Browse the manuals
- Go to the product web site (www.keysight.com/find/M8000) and browse the manuals
- On installing the M8070ADVB Advanced Measurement Package, you will find documentation at the plugin directory: *C:\Program Files\Keysight\M8070B\Plugins\M8070ADVB\doc*.

Related Documents

Refer the following related documents:

Table 3 Related Documents

Document Part No.	Document Title
M8000-91B01	M8020A Start Here
M8000-91B03	M8040A Start Here
M8000-91B20	M8050A Start Here
M8000-91B04	M8000 Series Tips for Preventing Damage
M8000-91B05	M8000 Series Installation Guide
M8000-91B06	M8020A Getting Started Guide
M8000-91B07	M8040A Getting Started Guide
M8000-91B21	M8050A Getting Started Guide
M8000-91B08	M8000 Series BER Test Solutions User Guide
M8000-91B09	M8000 Series BER Test Solutions Programming Guide
M8000-91B14	M8000 Series Plugins Getting Started Guide
M8000-91B10	M8000 Series Advance Measurement Package User Guide
M8000-91B11	M8000 Series Error Distribution Analysis Package User Guide
M8000-91B15	M8070ISIB Adjustable ISI Package for M8000 Series User Guide

Additional Documents

Refer the following additional documents:

http://www.keysight.com/find/M9505A for 5-slot chassis related documentation.

http://www.keysight.com/find/M9502A for 2-slot chassis related documentation.

Contact Keysight Technologies

For more information on Keysight Technologies’ products, applications or services, please contact your local Keysight office. The complete list is available at: www.keysight.com/find/contactus

2 Measurements

This chapter describes the setup, execution, monitoring and results of the measurements provided by Advanced Measurement Package.

Overview

The Advanced Measurement Package includes the following measurements:

- Output timing measurement
- Output level measurement
- Jitter tolerance measurement
- Eye diagram measurement
- Parameter sweep measurement

NOTE **Please note that the measurements supported by the Advanced Measurement Package requires license for its activation. For details on license, see "License Information" on page 18.**

The M8020A modules (M8041A + M8051A) and (M8041A + M8062A) supports the following measurements:

- Output timing measurement
- Output level measurement
- Jitter tolerance measurement
- Eye diagram measurement
- Parameter sweep measurement

The M8040A modules (M8045A + M8046A) supports the following measurements:

- Output timing measurement
- Jitter tolerance measurement
- Parameter sweep measurement

NOTE **Please note that the Output Timing measurement is available in the user interface, but just for debugging/troubleshooting purposes. The accuracy of jitter separation results is unspecified in case of NRZ and invalid in case of PAM4 signals.**

The M8050A modules (M8042A + M8043A) supports the following measurements:

- Output timing measurement
- Output level measurement
- Jitter tolerance measurement
- Eye diagram measurement
- Parameter sweep measurement

NOTE

The BER measurement processes at least as many bits as defined by the acquisition parameters, it stops when the end criteria is reached or exceeded. Due to update period and hardware granularities, an individual BER measurement will typically process more bits than configured.

Exploring Measurement User Interface

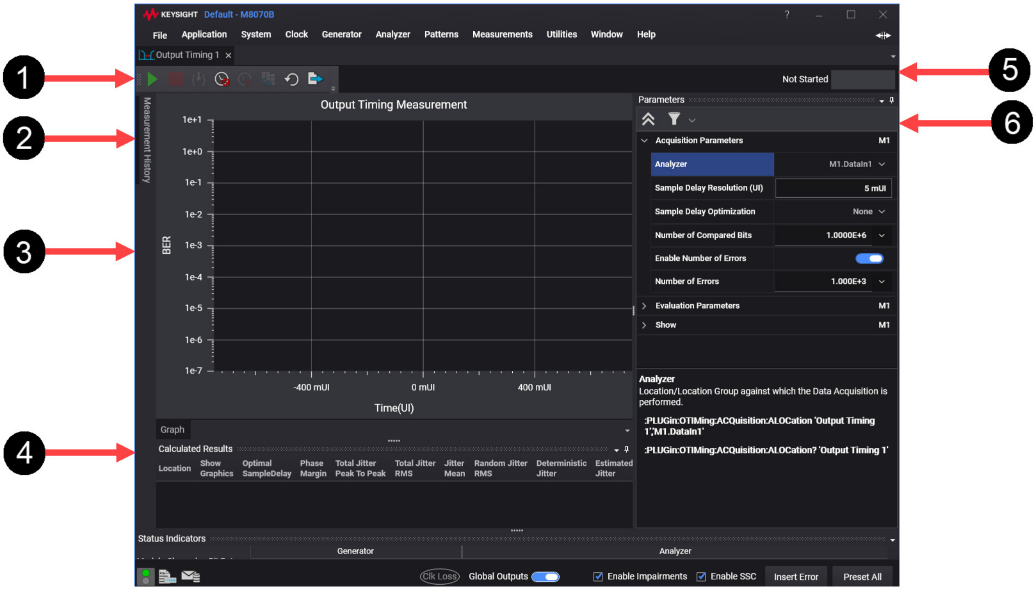
This section describes the functionality provided by the measurement user interface.

Launching the Measurement User Interface

To launch the measurement user interface:

- Go to the **Menu Bar** > **Measurements** and then select the respective measurements (**Error Ratio**, **Output Timing, Output Level** or **Jitter Tolerance**) to launch the measurement user interface.

The following figure shows an example of the measurement user interface:



Analyzer: M1.DataIn1
Sample Delay Resolution (UI): 5 mUI
Sample Delay Optimization: None
Number of Compared Bits: 1.0000E+6
Enable Number of Errors: ●
Number of Errors: 1.000E+3

Analyzer
Location/Location Group against which the Data Acquisition is performed.

```
:PLUGin:OTIMing:ACQuisition:ALOCation 'Output Timing 1','M1.DataIn1'
:PLUGin:OTIMing:ACQuisition:ALOCation? 'Output Timing 1'
```

☑ Enable Impairments ☑ Enable SSC

The measurement user interface has the following GUI elements which are common to all measurements:

- Toolbar
- Status Indicator
- Measurement History Pane
- Measurement Graph

- Parameter Window
- Calculated Results Pane

Lets discuss these GUI elements in the following sections.

Toolbar

The toolbar contains the following icons:

Table 4

Elements	Name	Description
▶	Start /Continue Measurement	Starts a measurement.
⏸	Break Measurement	Halts the measurement at that point. Once paused, you can continue the measurement again by pressing **Continue Measurement** icon. Note: This option is not available in **Error Ratio Measurement**.
🟥	Stop Measurement	Stops the measurement.
{↓}	Step Into Measurement	Steps further into the measurement. Note: This option is not available in **Error Ratio Measurement**.
✓	Enable/Disable Measurement Run History	Enables or disables the measurement run history. For details, refer to "Measurement History Window" on page 29
✗	Clear Measurement History	Clears the measurement run history.

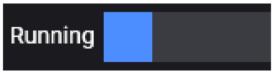
Elements	Name	Description
	Copy Measurement History Properties	Copies the measurement history properties to the currently running measurement.
	Reset Measurement	Resets the measurement to its default values.
	Export Data to CSV	Exports the measurement data as CSV file. For details see "Exporting Measurement Results" on page 33.

Status Indicator

The status indicator shows the current state of a measurement. There can be various states of a measurement, depending on the type of measurement. These may be as follows:

- **Not Started**: Indicates that the measurement has not yet started.
- **Running**: Indicates that the measurement is currently running.
- **Stop**: Indicates that the measurement is stopped.
- **Error**: Indicates an error while executing the measurement which is caused due to invalid parameter settings.
- **Suspended**: Indicates that the measurement is suspended.
- **Finished**: Indicates that the measurement is completed.

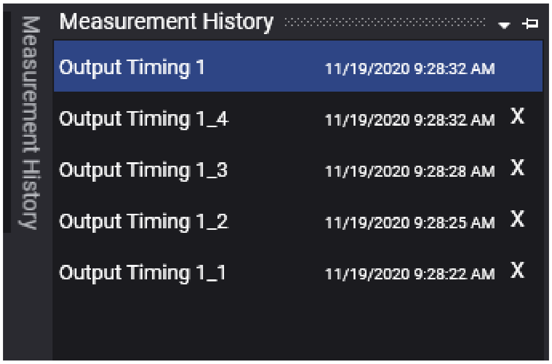
The following figure shows the status indicator while the measurement is running:



Measurement History Window

The **Measurement History** window maintains the history of executed measurement along with their time stamp. This allows you to refer to the previously run measurements and compare their results.

The **Measurement History** window is shown in the figure below:



Measurement History

Output Timing 1	11/19/2020 9:28:32 AM	
Output Timing 1_4	11/19/2020 9:28:32 AM	X
Output Timing 1_3	11/19/2020 9:28:28 AM	X
Output Timing 1_2	11/19/2020 9:28:25 AM	X
Output Timing 1_1	11/19/2020 9:28:22 AM	X

Click the icon to toggle between the enable/disable measurement run history in the Measurement History window.

Copy Measurement History Properties

This feature allows you to copy the properties of run measurement to currently running measurement.

To do so,

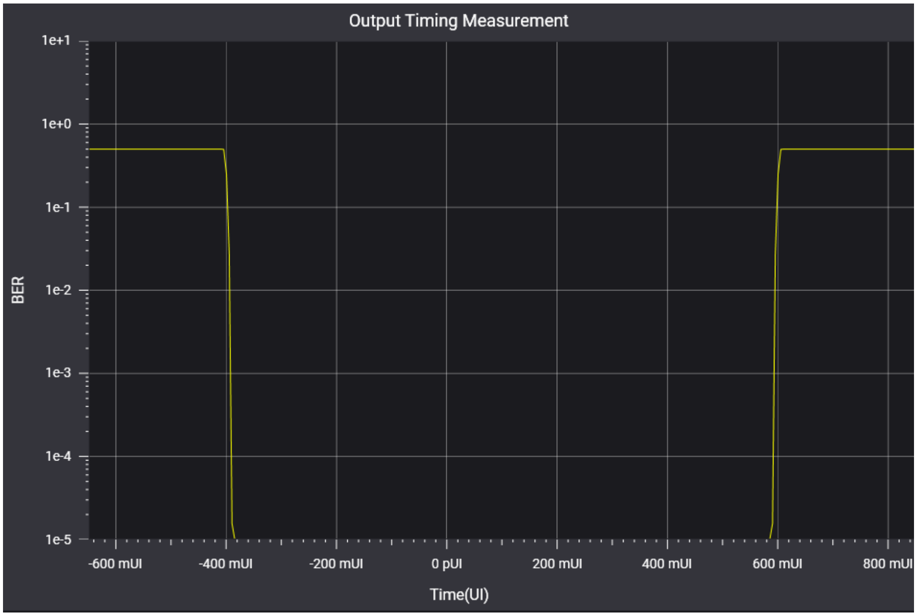
- Select a measurement history from the list shown in the **Measurement History** window.

- Click the **Copy Measurement History Properties** icon. The properties of the selected measurement will be copied to the current measurement.

Measurement Graph

The **Measurement Graph** displays the calculated graph and results.

The following figure displays the **Measurement Graph** of the **Output Timing Measurement**.



The **Measurement Graph** contains the following tabs:

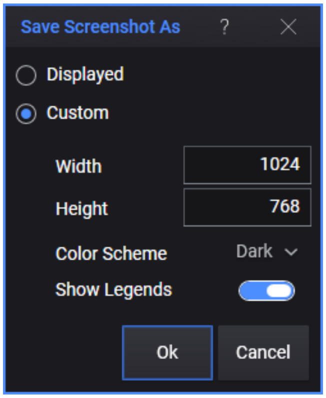
- **Graph**: Displays the graphical representation of the measurement. The graph varies from measurement to measurement. The details of each measurement graph are further described in their respective sections.
- **Location**: Displays the raw measurement data for that location. However, if you are running measurement for a group, multiple tabs will appear that display the raw measurement for each location.

When you right-click on the **Measurement Graph**, a context menu appears which provides the following options:

- **Turn ON/OFF Fit to view (Ctrl+Home)**- Turns ON/OFF **Fit to view option**.

- **Fit to view (Home)** - Makes the visible area fit to display entire contents.
- **Copy screenshot (F11)** - Copies the screenshot of charts to clipboard.
- **Save screenshot (Ctrl+S)** - Saves the screenshot as an image (PNG) under a name.

  Clicking this option displays the **Save Screenshot As** dialog box.

  Save Screenshot As

  ○ Displayed

  ● Custom

  Width 1024

  Height 768

  Color Scheme Dark

  Show Legends ●

  a On the **Save Screenshot As** dialog box, select one of the following:

  - **Displayed:** Saves the screenshot with default properties.
  - **Custom:** Enables you to set the custom properties of the image. The following properties are available:
    - **Width**: Sets the width value of the image.
    - **Height**: Sets the height value of the image.
    - **Color Scheme**: Enables you to select a color scheme for the image. Two options; Dark and Light are available.
    - **Show Legends**: Allows you to save the legends in the screenshot.

  b Click **OK** to save the screenshot.
- **Quick Help (Alt+F1)** - Opens a window that provides brief information about the dynamic display.

Parameters Window

The **Parameter** window allows you to set the parameters for a location or a group. For each measurement, it contains two types of parameters:

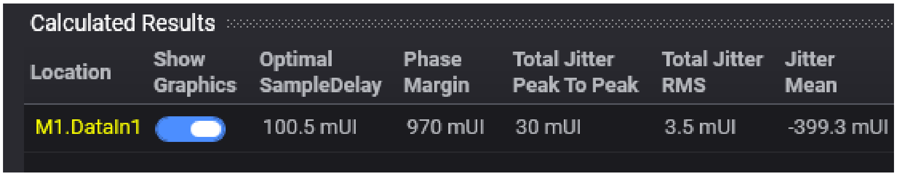
- **Acquisition Parameters** - Pre-Parameters influence how the data for a measurement is collected; changes require a re-run in order to be effective. It also allows you to select a location or location group against which the data acquisition is performed.
- **Evaluation Parameters** - Post-Parameters influence how the collected measurement data is evaluated. Changes do not require a re-run in order to be effective.

The acquisition and evaluation parameters differ from measurement to measurement. The detailed description of these parameters are explained in the sections that follow.

Calculated Results Pane

The **Calculated Results** pane displays the calculated results in the form of measurement parameters for each location. The calculated measurement parameters varies from measurement to measurement.

The **Calculated Results** pane is shown in the following figure:



Calculated Results

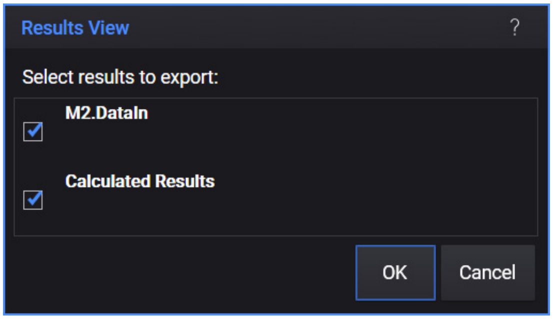
Location	Show Graphics	Optimal SampleDelay	Phase Margin	Total Jitter Peak To Peak	Total Jitter RMS	Jitter Mean
M1.DataIn1	●	100.5 mUI	970 mUI	30 mUI	3.5 mUI	-399.3 mUI

For each location, you can click on the slide button to show/hide the measurement graph.

Exporting Measurement Results

All the measurements supported by the **Advanced Measurement Package** allows you to export the measurement results into a CSV format. To do so:

1 Click the (**Export Data to CSV**) button. The **Results View** dialog will appear that allows you to select one or more results to export:



Results View

Select results to export:

☑ M2.DataIn

☑ Calculated Results

2 Select the required results, such as, "M*.DataIn" or "Calculated Results", and so on (depending on the type of measurement).

3 Click **OK**. A standard **Save As** dialog will appear.

4 Provide a filename and location.

5 Click **Save**. A confirmation dialog indicates that the measurement results has been successfully exported to a CSV format.

NOTE

The BER measurement processes at least as many bits as defined by the acquisition parameters, it stops when the end criteria is reached or exceeded. Due to update period and hardware granularities, an individual BER measurement will typically process more bits than configured.

Output Timing Measurement

Overview

The **Output Timing** measurement is used to measure the timing and jitter behavior for a device under test (DUT). It uses a bit error rate (BER) measurement to evaluate the shape of the eye for the output signal of the DUT. It also analyzes the jitter, separates the random jitter and deterministic jitter components, and estimates the total jitter.

A direct result is the determination of the optimum sampling point delay for receiving data from the DUT with maximum confidence.

DUT **Output Timing/Jitter** includes the **Fast Total Jitter** measurement that can be used to measure the total jitter for devices which generate a very low error density in a reasonable time span.

Output Timing Characteristics

The sampling point is swept automatically within a 1.5 clock period to generate a "bathtub" curve. The resulting graph is centered around the optimum sampling point of the port.

In addition, the results are available in a tabular view. If a clock signal is defined, the software measures the data to clock alignment and displays the absolute delay.

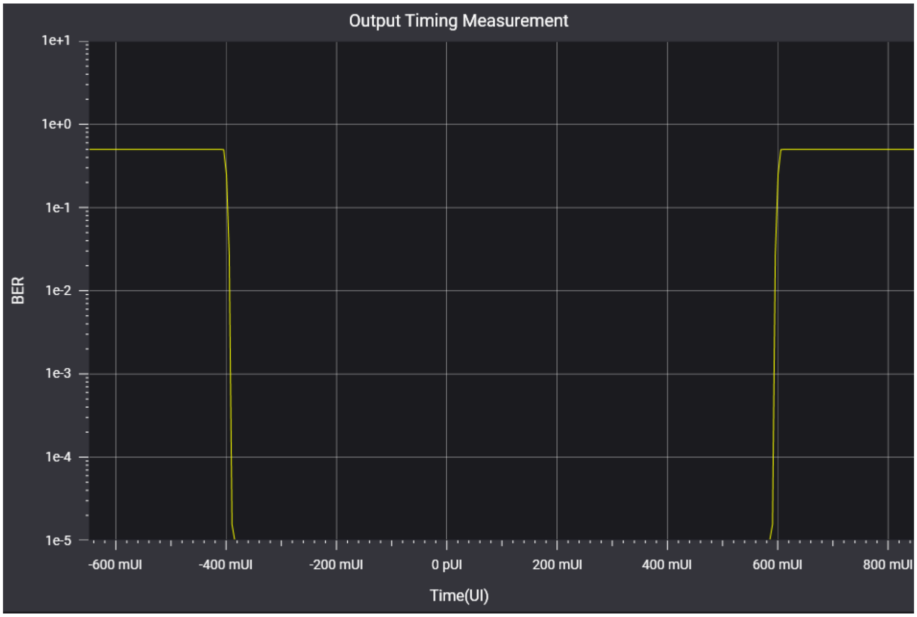
Jitter Characteristics

The **DUT Output Timing/Jitter** measurement calculates the different components making up the jitter:

- Random Jitter (RJ)
- Deterministic Jitter (DJ)

Example Results

The following illustration shows the resulting graph of a typical DUT **Output Timing** measurement:



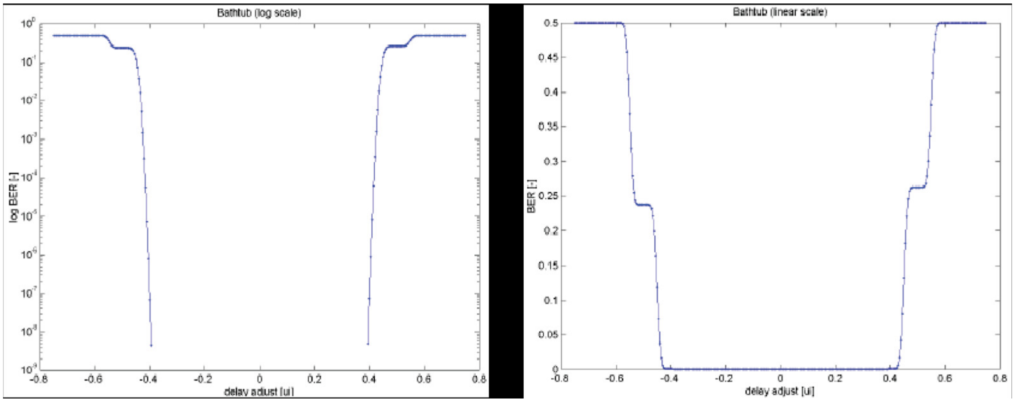
Understanding the Jitter Calculation

After the output timing behavior of the DUT is measured, the M8020A calculates the different jitter components:

Random and Deterministic Jitter

To understand the RJ and DJ results, it is helpful to first understand how the software generates the results:

1 The bathtub curve is measured.



2 All measurement points that have BER between the BER Threshold and Minimum BER for RJ/DJ Separation are transformed into Q-space.

The Q-factor describes the signal-to-noise ratio at the decision circuit.

3 Linear regression is performed for both the left and right edges.

4 The mean and sigma are calculated for both lines:

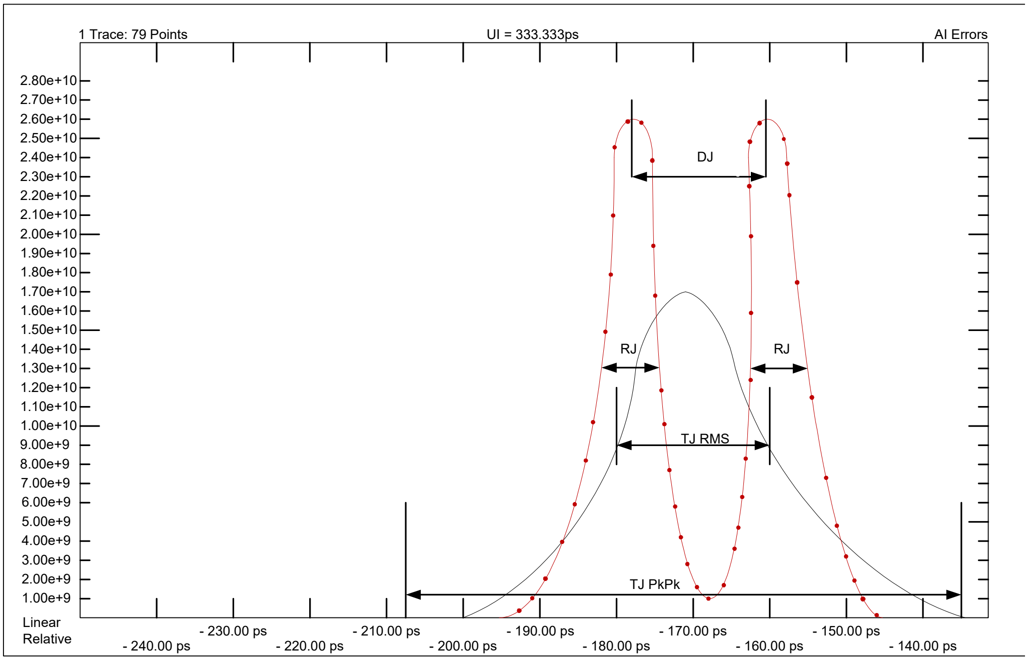
- RJ is calculated as the mean of the two sigmas.
- DJ is calculated as the period minus the difference of the two means.

5 The estimated TJ is calculated:

- Linear regression is used to extrapolate the bathtub curve to lower BER values.
- The intersections of the resulting lines with the Residual BER for RJ/DJ Separation are located.
- The eye opening is calculated.

The estimated TJ is the period minus the width of the eye opening.

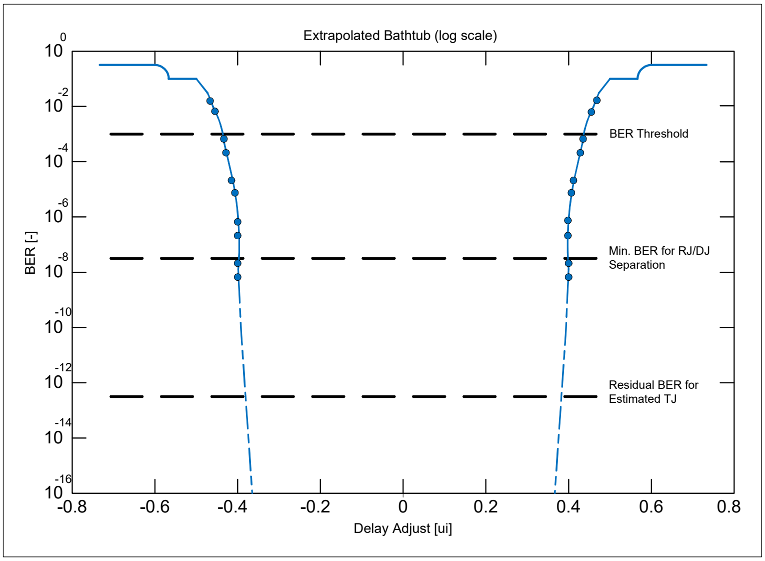
The following illustration shows a jitter curve where both RJ and DJ are present. It also shows how the TJ peak-to-peak and RMS are calculated.



Estimated Total Jitter

The **Estimated Total Jitter** (TJ) allows you to predict the jitter expected for very low bit error rates that would take a long time to measure. It is obtained by extrapolating the measured BER curves.

The TJ is estimated by extending the BER curves (based on the points detected between the BER Threshold and the Minimum BER for RJ/DJ Separation) to the Residual BER for RJ/DJ Separation level. The estimated TJ is the period minus the width of the measured eye.



Explanation of the Fast Total Jitter Measurement

The **Fast Total Jitter** measurement is an optimized method to determine the total jitter for devices that generate a very low error density (BER well below 10^{-10}).

To measure (not estimate) the total jitter for a device with a BER of 10^{-12} with conventional methods, one usually needs to compare more than 10^{12} bits for each sample point. To measure a full eye opening this way with appropriate timely resolution takes time (maybe days or weeks, depending on the data rate), and the probability of seeing one or no error in 10^{12} bits is not higher than 37%.

If one would compare 10^{13} bits for the same device, the probability of seeing ten errors is even lower (12%), but the probability of observing no error is almost zero.

The **Fast Total Jitter** measurement implements a method that reduces the measurement time considerably and provides a higher accuracy. It is based on statistical and probability calculations.

Measurement Duration

The duration of a **Fast Total Jitter** measurement depends on:

- the BER threshold
- the bit rate
- the sample delay step size
- the contribution of random jitter
- the contribution of deterministic jitter

Compared to a conservative bathtub measurement, the **Fast Total Jitter** measurement can reduce the measurement time by more than a factor of 40.

Typical test times are approximately 20 minutes at 10 Gbit/s and slightly more than one hour at 2.5 Gbit/s for a measurement at the 10^{-12} BER threshold with an overall confidence level better than 90%.

Acquisition Parameters

You can use the **Properties** window to specify the following acquisition parameters:

Set the criteria for moving to the next sample point:

- Number of Compared Bits

  After this number of compared bits, the measurement stops for the current sample point and moves to the next one.

- Number of Errors

  After this number of errors, the measurement stops for the current sample point and moves to the next one. This allows you to speed up the measurement. You can switch off this option if only the number of compared bits is important.

Set the criteria for the sample delay:

- Sample Delay Resolution

  Specifies the time distance between sampling points. A smaller value yields more sampling points in a unit interval. You can enter the resolution in UI.

  If the resolution used for the measurement is not high enough, the bathtub curve does not clearly show the edges. For example, you may wish to change the Resolution from 0.01 to 0.005 and run the measurement again.

- Sample Delay Optimization

  You can choose between Fast Total Jitter at BER or None.

- Fast Total Jitter at BER

  This enables the **Fast Total Jitter** measurement. Before enabling this measurement, you need to know the BER floor of the device and to specify a BER threshold that is above that floor.

Evaluation Parameters

You can use the **Properties** window to specify the following evaluation parameters:

- Analyze Errors

  You can analyze for:

  - All Errors

  To calculate the BER values from all bits/errors.

  - Errored Zeros

  To calculate the BER values if "0" is expected, but "1" received.

  - Errored Ones

  To calculate the BER values if "1" is expected, but "0" received.

- BER Threshold

  To calculate the parameters for the given BER threshold. This is the BER level for which output timing numerical values (phase margin, skew, etc.) are calculated. It is also the upper limit of the BER range for RJ/DJ separation.

  The BER threshold influences some of the parameters of the DUT Output Timing measurement. You can also drag and drop the horizontal BER threshold in the graphical display to change this value.

- Min BER for RJ/DJ Separation

  Lower limit of the BER range for RJ/DJ separation.

- Residual BER for Estimated Total Jitter

  BER level for which the estimated total jitter is calculated.

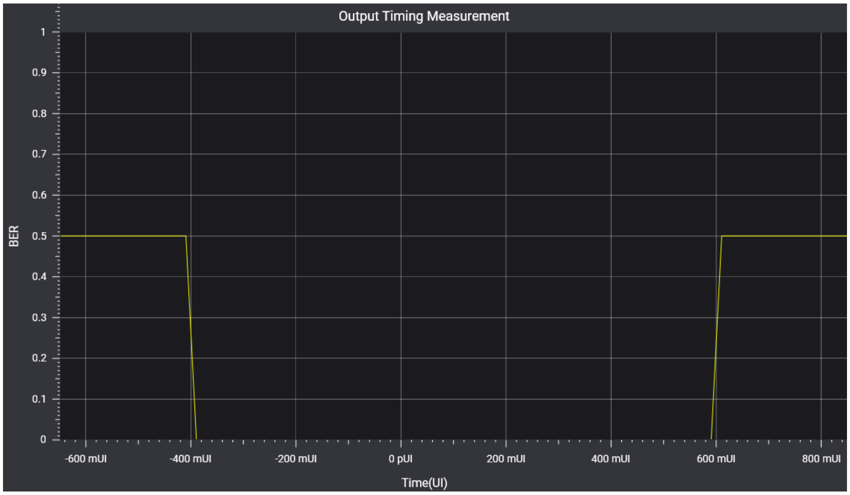
- Display Unit

  Choose between Unit Interval and Seconds to select the timebase for the display's x-axis.

- Vertical Axis Scale

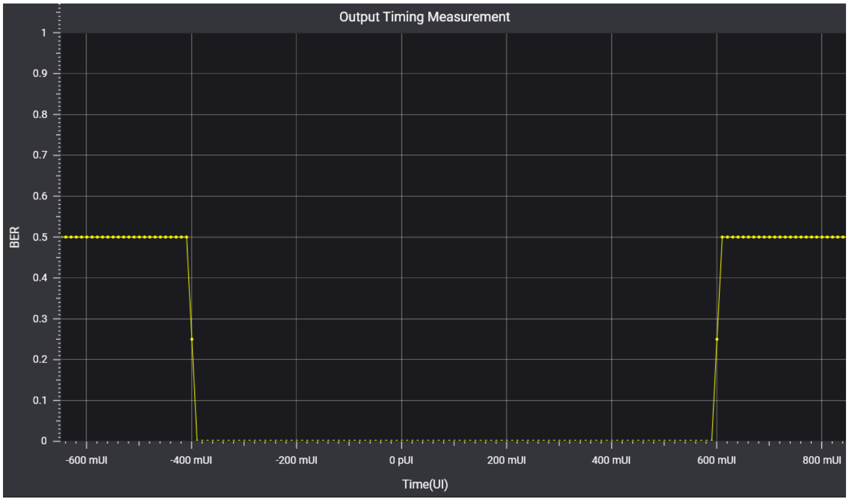
  Choose between Logarithmic and Linear to select the scale for the display’s y-axis.

  For example, a DUT Output Timing measurement displayed on a linear scale may look like this:



- Show Measured Points

  If you want to see the points that have actually been measured, click the **Show Measured Points** slide button.

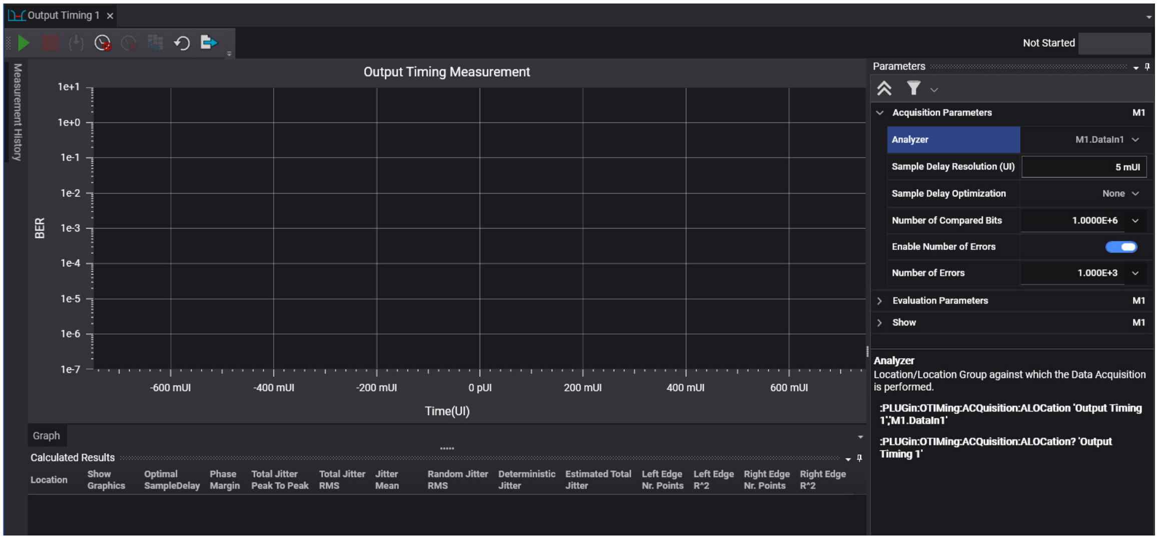


Launching the Output Timing Measurement

To launch the **Output Timing** user interface:

- Go the **Menu Bar** > **Measurements** and then select **Output Timing**.

The **Output Timing** user interface will appear as shown in the following figure:



Location	Show Graphics	Optimal SampleDelay	Phase Margin	Total Jitter Peak To Peak	Total Jitter RMS	Jitter Mean	Random Jitter RMS	Deterministic Jitter	Estimated Total Jitter	Left Edge Nr. Points	Left Edge R^2	Right Edge Nr. Points	Right Edge R^2

Acquisition Parameters	M1
Analyzer	M1.DataIn1
Sample Delay Resolution (UI)	5 mUI
Sample Delay Optimization	None
Number of Compared Bits	1.0000E+6
Enable Number of Errors	●
Number of Errors	1.000E+3
Evaluation Parameters	M1
Show	M1

Analyzer
Location/Location Group against which the Data Acquisition is performed.

```
:PLUGin:OTIMing:ACQuisition:ALOCation 'Output Timing 1','M1.DataIn1'
```

```
:PLUGin:OTIMing:ACQuisition:ALOCation? 'Output Timing 1'
```

The **Output Timing** user interface includes the following elements:

- **Toolbar**: For details, refer to "Toolbar" on page 27
- **History Pane**: For details, refer to "Measurement History Window" on page 29.
- **Measurement Graph**: For details, refer to "Measurement Graph" on page 45
- **Parameters Window**: For details, refer to "Parameters Pane" on page 43.
- **Calculated Results**: For details, refer to "Calculated Results" on page 46.

Parameters Pane

The **Parameters** window shows the acquisition parameters and evaluation parameters for output timing measurement.

Table 5

Parameters	Description	Values
Acquisition Parameters		
Acquisition Location	Location or Location Group against which the data acquisition is performed.	
Sample Delay	Specify the sample delay resolution.	Min - 5 mUI Max - 200 mUI
Sample Delay Optimization	Specify the sample delay optimization.	None | Fast Total Jitter
Fast Total Jitter at BER	Specifies the Fast Total Jitter BER. This is a conditional parameter and appear when **Sample Delay Optimization** is selected as “Fast Total Jitter”	Min - 1E-15 Max - 1E-9
No. of Compared Bits	Specify the criteria for moving to the next measurement and is only enabled when the Sample Delay Optimization is “None”.	Min - 1 E+0 Max - 1E+18
Enable Number of Errors	Use the slide switch to enable this property. Once this property is enabled, you can specify the additional number of errored bits to consider while moving to the next measurement	ENABLE | DISABLE
Number of Errors	Specify the criteria for moving to the next measurement and is only enabled when the Sample Delay Optimization is “None”. This parameter gets enabled when you enable the “Enable Number of Errors”.	Min - 1 E+0 Max - 1E+18
Evaluation Parameters		
BER Threshold	Specify the BER threshold	Min - 1E-15 Max - 1E-1
Min BER for RJ/DJ Separation	Specifies the minimum BER for random and deterministic jitter components separation.	Min - 1E-18 Max - 1E+0
Residual BER for Estimated Total Jitter	Specify the residual BER for estimated total jitter.	Min - 1 E-12 Max - 1E-6

Parameters	Description	Values
Analyze Errors	Specify options to analyze errors	All Errors | Errored Zeros | Errored Ones
Vertical Axis Scale	Specify the Y axis scale.	Linear | Logarithmic
Display Unit	Specify the unit to be displayed.	Seconds | UI
Show Measured Points	Show/hides the measured points on the graph.	Show | Hide

How to Run Output Timing Measurement

To run an **Output Timing** measurement, perform the following steps:

- Use the **Parameters** window to select the location or location group for which the data acquisition has to be performed.

- Set the acquisition and evaluation parameters for **Output Timing** in the **Parameters** Window. For details, refer to "Acquisition Parameters" on page 39 and "Evaluation Parameters" on page 40.

- Click the ▶ **Start Measurement** icon to run the measurement. The measurement status indicator will indicate **Running**.

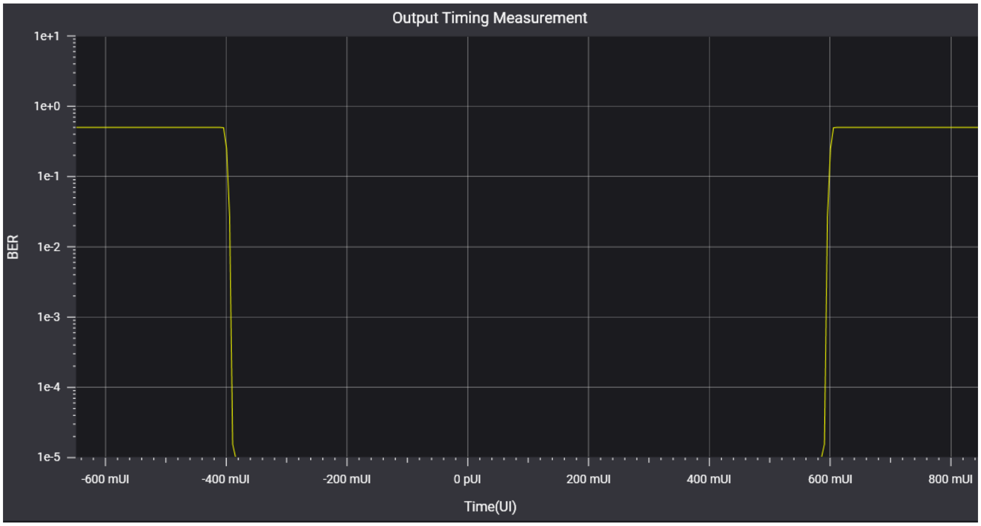
How to Stop Output Timing Measurement

To stop a measurement:

- Click the 🟥 **Stop Measurement** icon to stop the measurement.

Measurement Graph

Once you run an **Output Timing** measurement for a specified duration, the following graph is shown on the measurement graph:



The bathtub curve in the graph shows the overall jitter distribution over a unit interval and serves as the basis for bit error rate estimation.

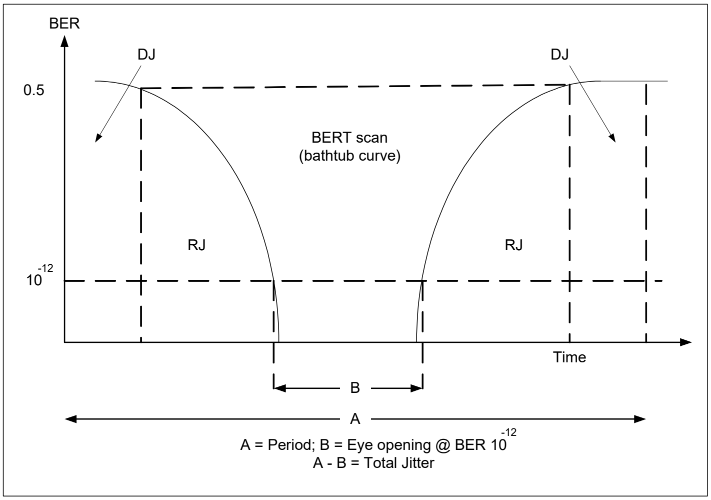
How an Output Timing Measurement Works

The sampling point is swept automatically within a 1.5 clock period to generate a “bathtub” curve. The resulting graph is centered around the optimum sampling point.

The BER Bathtub graph plots TJ (Total Jitter) data and BER values across an entire bit period, which is labeled as the unit interval (UI) on the horizontal axis. BER values are calculated using DJ and RJ and are used to extrapolate to low BER levels. In general, the BER values will differ from the measured TJ values at high probabilities, but the two should converge at low probabilities. Jitter can be measured by moving the sampling point across the eye pattern into the crossing region step by step.

The sampling point- the amplitude decision threshold and the time at which the bits are measured – is scanned across the eye pattern over the time period while monitoring the BER. The bathtub curve can also be used to separate random from the deterministic jitter.

The slope of the bathtub curve is the measure of the random jitter, whereas the slope offset positions on the time axis are set by deterministic jitter. Total jitter is quantified by noting points where BER reduces to at both the eye edges and subtracting this interval from the bit period.



Calculated Results

The calculated results for the **Output Timing** measurement are displayed in the tabular format. It includes the following elements:

- **Location**: Location or Location/Group against which the data acquisition is performed.
- **Show Graphics**: Displays the measurement graph in the measurement output window.
- **Optimal Sample Delay**: Sample delay coordinate of the center of a bounding box around the BER threshold contour line.
- **Phase Margin**: Period of time where the bit error rate is lower than the BER threshold.
- **Total Jitter Peak-to-Peak**: Peak-to-peak value for total jitter. Calculated as the pulse period (unit interval) minus the Phase Margin.

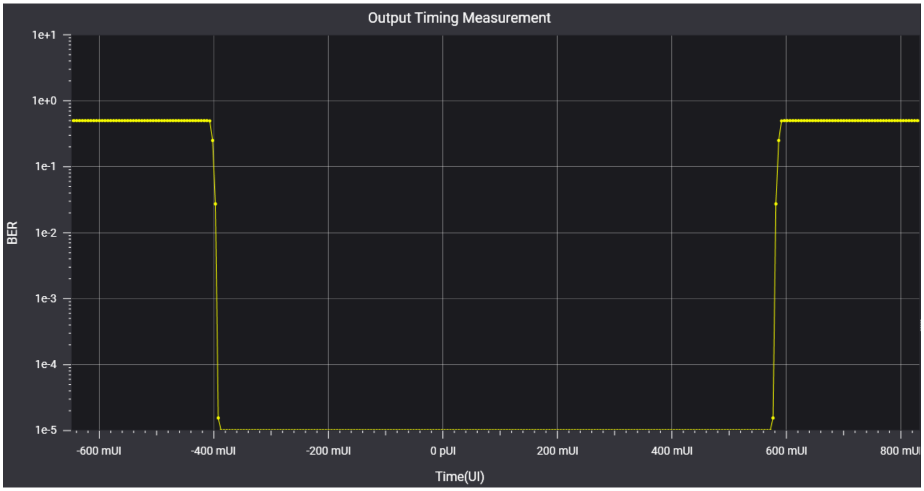
- **Total Jitter RMS**: The average of the left and right jitter histogram root mean squared values.
- **Jitter Mean**: Mean value for total jitter. Calculated as the weighted average of the left edge jitter histogram.
- **Random Jitter RMS**: The total jitter component with Gaussian distribution. After transforming a contiguous range of measured points into Q space and performing a linear regression, it is calculated as the mean of the sigmas of the two straight lines. The contiguous range is limited by the BER Threshold and the Min. BER for RJ/DJ Separation threshold.
- **Deterministic Jitter**: The total jitter component with non-Gaussian distribution. After transforming a contiguous range of measured points into Q space and performing a linear regression, it is calculated as the period minus the difference between the means of the two straight lines.
- **Estimated Total Jitter**: A forecast of the expected jitter for very low bit error rates. After extrapolating the measured BER curves, it is calculated as the period minus the expected width of the eye opening.
- **Left Edge No. Points**: The the number of points that has been measured between the BER Threshold and the Min. BER for RJ/DJ Separation threshold. It is displayed for both slopes. This number has to be greater than 2 for the RJ, DJ, and estimated TJ values to be applicable.
- **Left Edge R^2**: The R^2 values are calculated for both slopes of the bathtub curve. They are a measure of how well the transformed points between BER Threshold and Min. BER for RJ/DJ Separation fit to the linear regression. They have to be greater than 0.75 for the RJ, DJ, and estimated TJ values to be applicable.
- **Right Edge No. Points**: The number of points that has been measured between the BER Threshold and the Min. BER for RJ/DJ Separation threshold. It is displayed for both slopes. This number has to be greater than 2 for the RJ, DJ, and estimated TJ values to be applicable.
- **Right Edge R^2**: The R^2 values are calculated for both slopes of the bathtub curve. They are a measure of how well the transformed points between BER Threshold and Min. BER for RJ/DJ Separation fit to the linear regression. They have to be greater than 0.75 for the RJ, DJ, and estimated TJ values to be applicable.

Fast Total Jitter Measurement Results

The **Fast Total Jitter** measurement provides both graphical and numerical results:

The example below shows a copied result, and the display of measured points was enabled.

The results of the recent measurement are disabled, because they refer to a standard "bathtub" measurement that provides many more results. By actuating the Show indicator, you can inspect both alternatively.



The resulting graph shows you the points in time that have been investigated and whether the actual BER at these points was higher or lower than the BER threshold specified for the measurement.

NOTE

For coded patterns, the Fast Total Jitter measurement will only work when the bit recovery mode is enabled.

Calculated Results for Fast Total Measurement

The calculated results for the **Fast Total Jitter** measurement are displayed in the **Calculated Results** pane. It includes the following elements:

- **Location**: Location or Location/Group against which the data acquisition is performed.
- **Show Graphics**: Displays the measurement graph in the measurement output window.
- **Optimal Sample Delay**: The mean value of the left and right bathtub/Total Jitter BER Threshold intersections.
- **Phase Margin**: The period of time where the bit error rate is lower than the Total Jitter BER Threshold.
- **Total Jitter Peak-Peak**: Peak-to-peak value of the total jitter. Calculated as the pulse period (unit interval) minus the Phase Margin at the Total Jitter BER Threshold.
- **Total Jitter Uncertainty**: Displays the total jitter uncertainty.

NOTE

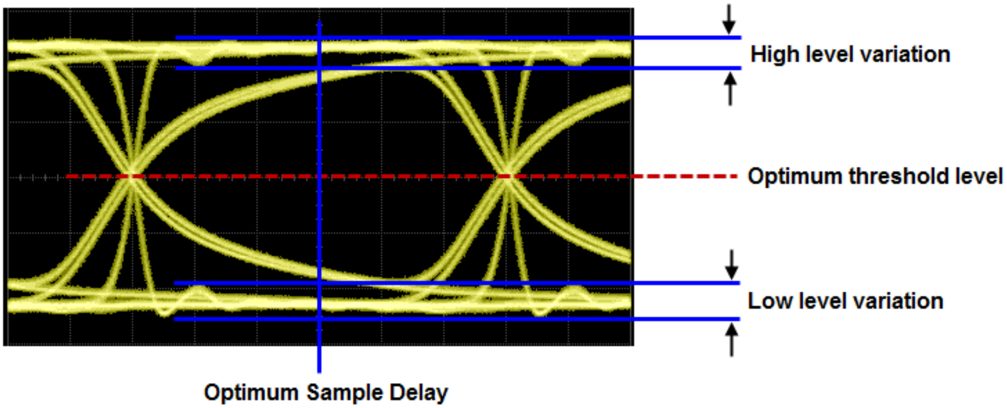
The BER measurement processes at least as many bits as defined by the acquisition parameters, it stops when the end criteria is reached or exceeded. Due to update period and hardware granularities, an individual BER measurement will typically process more bits than configured.

Output Level Measurement

Overview

The **Output Levels** measurement allows you to characterize the behavior of the output levels of a device under test (DUT). The sampling delay is fixed. The analyzer's decision threshold is automatically swept within a user-defined range.

A direct result is the determination of the optimum decision threshold level for receiving data from the DUT with maximum confidence.

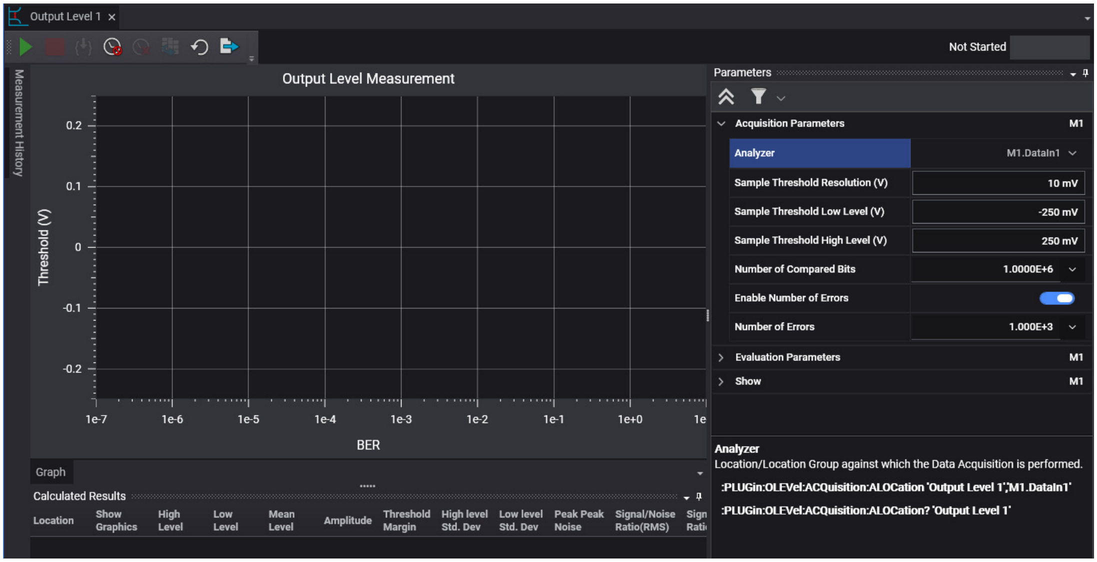


Launching the Output Level Measurement

To launch the **Output Level** user interface:

- Go the **Menu Bar** > **Measurements** and then select **Output Level**.

The **Output Level** user interface will appear as shown in the following figure:



Acquisition Parameters	M1
Analyzer	M1.DataIn1
Sample Threshold Resolution (V)	10 mV
Sample Threshold Low Level (V)	-250 mV
Sample Threshold High Level (V)	250 mV
Number of Compared Bits	1.0000E+6
Enable Number of Errors	●
Number of Errors	1.000E+3
Evaluation Parameters	M1
Show	M1

Analyzer
Location/Location Group against which the Data Acquisition is performed.

```
:PLUGin:OLEVel:ACQuisition:ALOCation 'Output Level 1','M1.DataIn1'
:PLUGin:OLEVel:ACQuisition:ALOCation? 'Output Level 1'
```

The **Output Level** user interface includes the following elements:

- **Toolbar**: For details, refer to "Toolbar" on page 27
- **History Pane**: For details, refer to "Measurement History Window" on page 29.
- **Measurement Graph**: For details, refer to "Measurement Graph" on page 45
- **Parameters Window**: For details, refer to "Parameters Window" on page 51.
- **Calculated Results**: For details, refer to "Calculated Results" on page 58.

Parameters Window

The **Parameters** window have the following acquisition, evaluation and show parameters for output level measurement.

Acquisition Parameters

- **Analyzer**: Specifies location/location group against which data acquisition is performed.

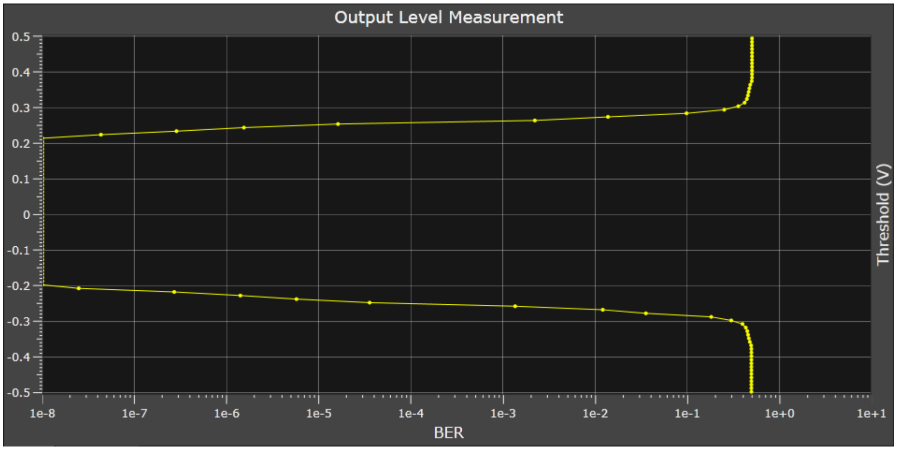
- **Resolution**: Specifies the distance between the sampling points. The lower this value is, the more sampling points you have in the selected voltage range. The minimum step width is hardware-dependent. At the time being, the minimum is 1 mV.
- **Low Level**: This is the lower end of the measured voltage range.
- **High Level**: This is the upper end of the measured voltage range.
- **Number of Compared Bits**: After this number of compared bits, the measurement stops for the current sample point and moves to the next one.
- **Number of Errors**: After this number of errors, the measurement stops for the current sample point and moves to the next one. This allows you to speed up the measurement. You can switch off this option if only the number of compared bits is important.

Evaluation Parameters

- **BER Threshold**: This is the bit error rate threshold at which the Threshold Margin is determined. It is also the upper threshold for the Q-factor calculations. The BER Threshold is displayed in the BER vs. Threshold graph.
- **Min BER for Q**: This is the lower threshold for the Q-factor calculations.

Show Graphics View

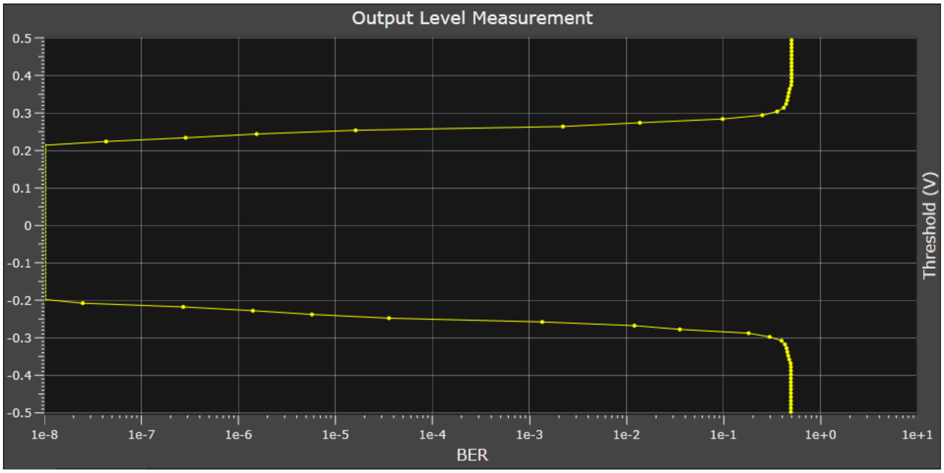
BER vs. Threshold Graph: This graph shows the relationship between the analyzer decision threshold and the measured BER.



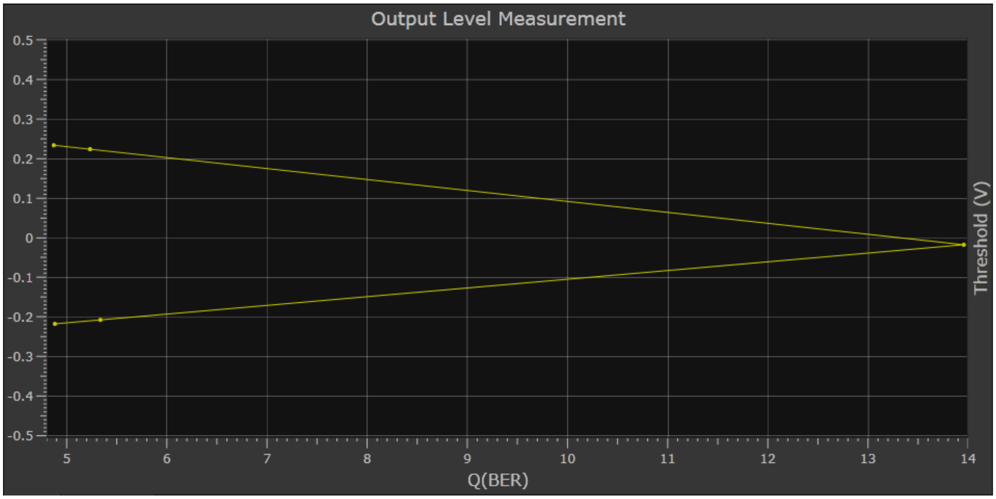
The BER considers all errors. It is calculated as:

$$\mathrm{BER}_{\mathrm{AllErrors}} = \frac{(\sum \mathrm{Error\,1s} + \sum \mathrm{Error0s})}{(\mathrm{total\ \#\ of\ Bits})}$$

- **QBER vs. Threshold Graph**: This graph shows the extrapolation of the optimum Q-factor and the optimum threshold level from a limited number of measured points.
  The measured data points to be used for the calculation have to be within a contiguous BER range. This range is defined by specifying the Min BER for Q (lower threshold) and the BER Threshold (upper threshold). Both thresholds can be set in the lower section of the View tab.
  The Q-factor can only be calculated if for both high and low level rails two or more points fall within the defined BER range. For reliable results use at least five measured points.
  To ensure proper settings, enable the **Show Measured Points** function and switch to the **BER vs. Threshold** graph. Move the upper BER threshold marker (vertical line) so that a sufficient number of measured points is included for the calculation:



This graph illustrates the calculation of the Q-factor as the best fit line through the calculated points.



Changing the Measurement Setting

- If you change the acquisition settings through **Parameter** window after the measurement has been run, please note that the changes on the parameters tab will take only effect if you run the measurement again.
- However, if you change the parameters that change the display of the measured data (graphics, show measurement points and axis scale), there is no need to repeat the measurement.

Available Views

The **Output Levels** measurement provides three different graphical views to visualize the calculated results:

- **BER versus Threshold**:
  This graph shows the relationship between the analyzer decision threshold and the resulting BER. It presents the raw data.
- **Q from BER versus Threshold**:
  This graph shows the extrapolation of the Q-factor and the optimum threshold level from a limited number of measured points.

How to Execute Output Level Measurement

To run an **Output Level** measurement, perform the following steps:

- Use the **Parameters** window to select the location or location group for which the data acquisition has to be performed.
- Set the acquisition and evaluation parameters for **Output Level** in the **Parameters** Window. For details, refer to “Parameters Window” on page 51.
- Click ▶ **Start Measurement** icon to execute the measurement. The measurement status indicator will indicate **Running**.

How to Stop Output Level Measurement

To stop a measurement:

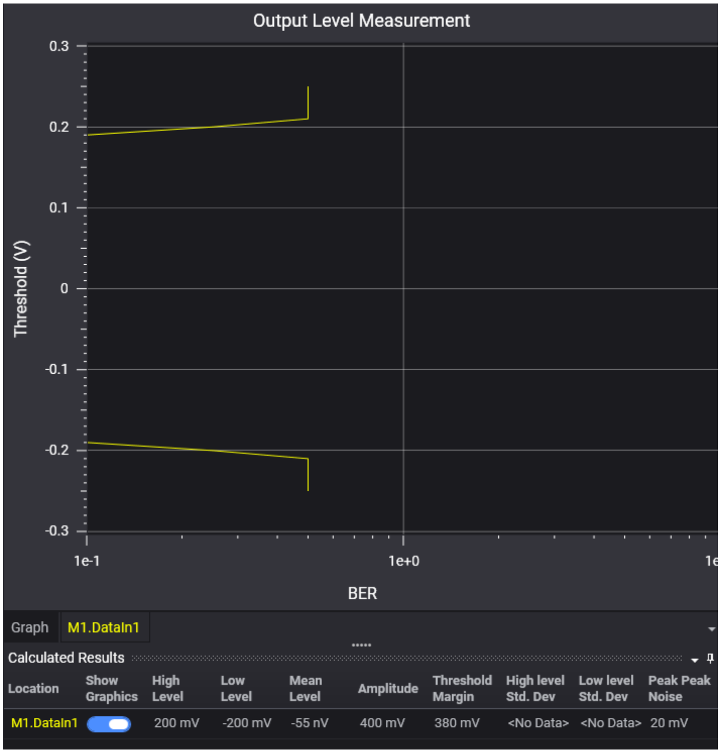
- Click 🟥 **Stop Measurement** icon to stop the measurement.

Measurement Graph

The **Measurement Graph** displays the absolute values of the derivative of the bit error rates over the thresholds (dBER/dTh). It visualizes the data that forms the basis for the calculations of the level and noise values.

The **Output Level** measurement provides Q from BER versus Threshold graph. This graph refers to the Q-factor calculations.

The following illustration shows the **BER versus Threshold** graph of a simple **Output Level** measurement:



Calculated Results

Location	Show Graphics	High Level	Low Level	Mean Level	Amplitude	Threshold Margin	High level Std. Dev	Low level Std. Dev	Peak Peak Noise
M1.DataIn1	●	200 mV	-200 mV	-55 nV	400 mV	380 mV	<No Data>	<No Data>	20 mV

The **Measurement Graph** contains the following tabs:

- **Graph**: Displays the graphical representation of the measurement. The graph varies from measurement to measurement. The details of each measurement graph are further described in their respective sections.
- **Location**: Displays the raw measurement data for that location. However, if you are running measurement for a group, multiple tabs will appear that display the raw measurement for each location.

When you right-click on the **Measurement Graph**, a context menu appears which provides the following options:

- **Turn ON/OFF Fit to view (Ctrl+Home)** - Turns ON/OFF Fit to view option.
- **Fit to view (Home)** - Makes the visible area fit to display entire contents.
- **Copy screenshot (F11)** - Copies the screenshot of charts to clipboard.

- **Save screenshot (Ctrl+S)** - Saves the screenshot as an image (PNG) under a name.
- **Quick Help (Alt+F1)** - Opens a window that provides brief information about the dynamic display.

How to Improve the Output Levels Display

You can consider the following points to change the display of an existing measurement:

- **Measurement Points**: If you wish to see more details to investigate the graph, select **Show Measured Points** on the **Graph** option of the **Properties** window. With this option selected, all measured points are indicated in the graph with small squares.
- **Zoom**: Several zoom factors are available. When you show the zoom graph, you can also allow the zoom graph to track the mouse
- **Scale**: Choose between **Logarithmic** and **Linear** to select the scale for the displays x-axis.
  The scale of the **QBER vs. Threshold** graph is always linear.
- **Graph Context Menu**: When you right-click on the **Measurement Graph**, a context menu appears which provides the options such as fit to view, copy and save screenshot and quick help of dynamic display.

How to Change the Output Levels Properties

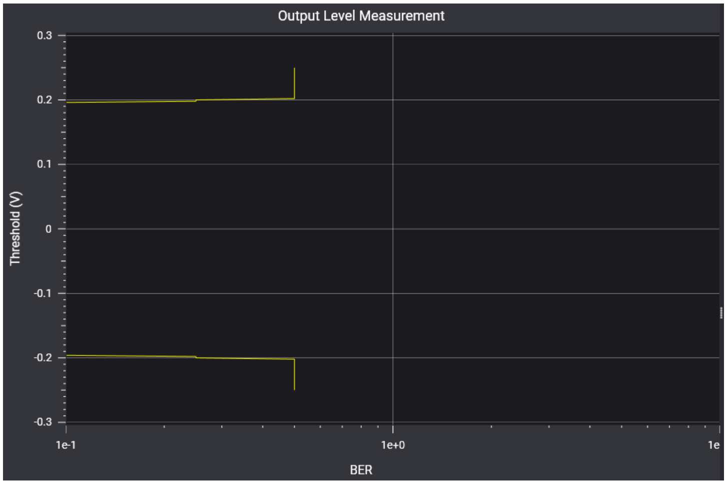
In the example measurement, we have set the focus on speed: 100 threshold levels and 1,000,000 compared bits per measuring point. You may wish to obtain more precise results.

1. Switch to the **Parameters** window.
2. Increase the **Number of Compared Bits** to 100,000,000.

   Remember: One failure per 1 million bits yields a BER resolution of 10-6. One failure per 100 million bits yields a BER resolution of 10-8.
3. Decrease the **Sample Threshold Resolution** to 2mV.

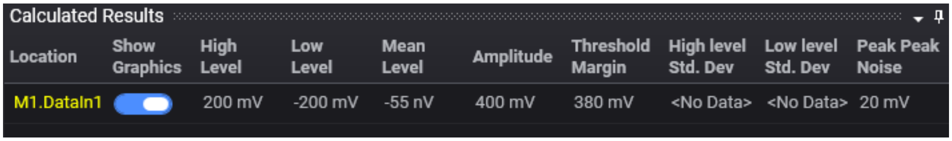
   This gives us 500 steps per Volt.
4. Press the **Start Measurement** icon to repeat the measurement with the new parameters.

The measurement now takes more time than the previous, but it is also much more precise.



Calculated Results

The following figure illustrates the calculated results:


Calculated Results

Location	Show Graphics	High Level	Low Level	Mean Level	Amplitude	Threshold Margin	High level Std. Dev	Low level Std. Dev	Peak Peak Noise
M1.DataIn1		200 mV	-200 mV	-55 nV	400 mV	380 mV	<No Data>	<No Data>	20 mV

The calculated results are divided into three groups:

- Level Results
- Noise Results
- Q-factor Results

Level Results

The level results are defined as follows:

- **High Level**: The High Level is the mean of the upper dBER/dTh distribution. It is calculated as:

$$Mean = \frac{\sum (dBer \cdot threshold)}{\sum dBer}$$

- **Low Level**: The Low Level is the mean of the lower dBER/dTh distribution. It is calculated as:

$$Mean = \frac{\sum (dBer \cdot threshold)}{\sum dBer}$$

- **Mean Level**: The Mean Level is the middle between the High and Low Levels, calculated as:

$$Mean\ Level = \frac{LowLevel + HighLevel}{2}$$

- **Amplitude**: The Amplitude is the difference between its High and Low Levels.
- **Threshold Margin**: The Threshold Margin is the distance between the upper and the lower BER curves at the position given by the BER Threshold setting.

Noise Results

The noise parameters are defined as follows:

- **High Level Std. Dev.**: The standard deviations are derived from the dBER/dTh histogram.
  The High Level Standard Deviation is calculated as:

$$StdDev = \sqrt{\frac{\sum ((threshold - Mean)^2 \cdot dBer)}{\sum dBer}}$$

where Mean is the High Level of the terminal.

- **Low Level Std. Dev.**: The Low Level Standard Deviation is calculated as:

$$StdDev = \sqrt{\frac{\sum ((threshold - Mean)^2 \cdot dBer)}{\sum dBer}}$$

where Mean is the Low Level of the terminal.

- **Peak Peak Noise**: The peak-to-peak Noise is calculated as:

$$Peak\ Peak\ Noise = \mathrm{High}Level - \mathrm{Low}Level - Threshold\ Margin$$

Note that the **Threshold Margin** depends on the position of the **BER Threshold**.

- **Signal/Noise Ratio (RMS)**: The RMS Signal-to-Noise Ratio is calculated as:

$$SNR\ (RMS) = \frac{HighLevel - LowLevel}{StdDev\ 1 + StdDev\ 0}$$

- **Signal/Noise Ratio (Peak Peak)**: The peak-to-peak Signal-to-Noise Ratio is calculated as

$$\mathrm{SNR\ (PeakPeak)} = \frac{HighLevel\ \text{-}\ LowLevel}{PeakPeakNoise}$$

Note that the **Peak-to-Peak Noise** depends on the position of the **BER Threshold**.

Q Factor Results

The numerical Q-factor parameters are defined as follows:

- **Q Factor**: The Q-factor is calculated as:

$$Q = \frac{\mu_1 - \mu_0}{\sigma_1 + \sigma_0}$$

where µ1,0 is the mean level of the 1 and 0 rails, respectively, and σ1,0 is the standard deviation of the noise distribution on the 1 and 0 rails.

- **Q Optimum Threshold**: The Q Optimum Decision Threshold is calculated as:

$$\text{Optimum Decision Threshold} = \frac{\sigma_0\mu_1 + \sigma_1\mu_0}{\sigma_1 + \sigma_0}$$

- **Q Residual BER**: The Q Residual BER is the expected BER at the Optimum Decision Threshold. It is calculated as:

$$BER = \frac{e^{-(Q^2/2)}}{Q_{opt}\sqrt{2\pi}}$$

Numbers below 1e-255 are expressed as zero.

- **Q High Level**: The Q High Level is the mean, calculated from the linear regression curve for the high level data:

$$\mu = \frac{-A}{B}$$

- **Q High Level Std.Dev**: The Q High Level Standard Deviation is the σ (Sigma), calculated from the linear regression curve for the high level data:

$$\sigma = \left| -\frac{1}{B} \right|$$

- **Q High Level Nr. Points**: This is the number of data points used for the calculation of the Q High Level value. It depends on the setting of the BER Threshold and also on the setting of the Min BER for Q parameter. The minimum for calculating Q-factor values is two points. It is recommended to use more than 5 points.
- **Q High Level R^2**: The R^2 parameter is an indicator that shows how well the converted data points fit to the straight line. It is calculated as:

$$R^2 = \frac{\left(\Sigma XY - \frac{(\Sigma X)(\Sigma Y)}{n}\right)^2}{\left(\Sigma X^2 - \frac{(\Sigma X)^2}{n}\right)\left(\Sigma Y^2 - \frac{(\Sigma Y)^2}{n}\right)}$$

The R^2 parameter should be examined before trusting the Q-values. Its maximum value is 1.0. It must be seen in conjunction with the number of data points.

For example: Two data points always fit perfectly well, but the resulting Q-factor calculations are not reliable. On the other hand, 50 data points may reveal a poor R^2 value. This tells you that the linearization is prone to errors.

If the R^2 value falls below 0.75, the Q-factor calculations are not applicable.

- **Q Low Level**: The Q Low Level is the mean, calculated from the linear regression curve for the low level data:

$$\mu = \frac{-A}{B}$$

- **Q Low Level Std.Dev**: The Q Low Level Standard Deviation is the σ (Sigma), calculated from the linear regression curve for the low level data:

$$\sigma = \left| -\frac{1}{B} \right|$$

- **Q Low Level Nr. Points**: This is the number of data points used for the calculation of the Q Low Level value. It depends on the setting of the BER Threshold and also on the setting of the Min BER for Q parameter.

  The minimum for calculating Q-factor values is two points. It is recommended to include more than 5 points.

- **Q Low Level R^2**: The R^2 value can also be seen as an indicator of how well the noise distribution fits to Gaussian shape. It will not fit, for example, if the received signal is dominated by cross-talk or modal noise.

NOTE

The BER measurement processes at least as many bits as defined by the acquisition parameters, it stops when the end criteria is reached or exceeded. Due to update period and hardware granularities, an individual BER measurement will typically process more bits than configured.

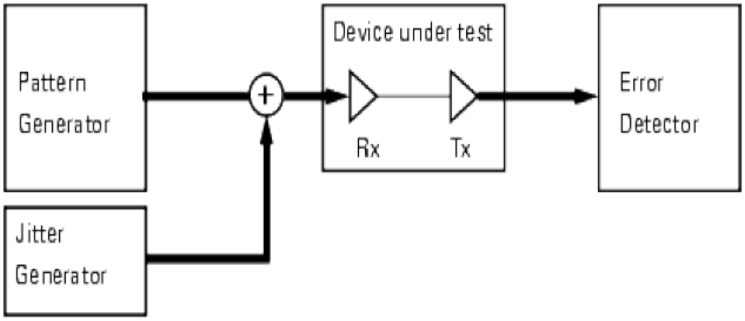
Jitter Tolerance Measurement

Overview

A **Jitter Tolerance** measurement is used to determine the ability of a device or system to maintain communication quality in the presence of jitter. It comes in two varieties:

- **Jitter Tolerance Characterization** determines the jitter levels where the device under test can no longer maintain a desired bit error ratio (BER).
- **Jitter Tolerance Compliance** verifies that the device under test is able to maintain a BER level at pre-defined jitter levels and jitter frequencies, as defined by a standard.

The basic setup of a jitter tolerance test is illustrated below:



Understanding Jitter Tolerance

The behavior and the passed/failed classification of data receiving devices or circuits are determined by their ability to withstand:

- Jitter
- Intersymbol interference
- Level noise

This refers particularly to the performance of phase-locked loops (PLLs) or clock data recovery circuits (CDR).

Jitter tolerance can be measured by applying a distorted data signal to the DUT and measuring the resulting bit error ratio. To make jitter tolerance tests reproducible, the signal distortion must also be reproducible. This requires some definitions.

Jitter

Receivers react differently on

- different types of jitter
- the jitter composition
- the jitter frequency spectrum
- the combination of jitter frequencies and data rate

Various test standards specify the jitter composition to be used for jitter tolerance tests.

The M8020A provides the following means for generating artificial (reproducible) jitter:

- Two types of voltage-controlled signal delay lines
- A phase shifter for modulating the generated clock

Intersymbol Interference

Conductors on PC boards have a limited bandwidth. This causes intersymbol interference which changes the shape of the received eye opening. The eye becomes asymmetrical.

Level Noise

Level noise affects the voltage amplitude of the eye opening.

PLL Performance Test Testing a PLL (or CDR) is not complete until the entire frequency range of the PLL has been checked under worst-case conditions.

For this purpose, the M8020A provides the Jitter Tolerance Characterization measurement.

PLL Standards and Compliance Tests

The Optical Internetworking Forum (OIF) as well as other institutions have proposed standards for testing the performance of data receivers and receiver circuits in the presence of jitter.

For this purpose, the M8020A provides the Jitter Tolerance Compliance measurement.

Types of Jitter

The standards for jitter tolerance tests prescribe combinations of certain jitter types. This section gives an overview of these types.

Random Jitter

Random jitter (RJ) is always present. Resistors, transistors, and other components generate noise that impacts transmitters and receivers.

This kind of jitter has a Gaussian distribution over time.

Bounded Uncorrelated Jitter

Similar to random jitter, bounded uncorrelated jitter (BUJ) has also a Gaussian distribution, but this distribution is cut (bounded) at both sides.

This kind of jitter can be caused, for example, by crosstalk on a parallel bus or by intersymbol interference of random or long pseudo random binary bit sequences.

To simulate bounded uncorrelated jitter for jitter tolerance tests, it can be generated from a filtered (frequency-limited) PRBS.

Periodic Jitter

Periodic jitter (PJ) is generally picked up from other periodic sources.

- **Rectangular Jitter**: Rectangular jitter may be caused, for example, by a switching power supply. The jitter amplitude is usually specified in UI. One UI (Unit Interval) is always the reciprocal of the present data rate.
- **Sinusoidal Jitter**: This kind of jitter can be picked up from any adjacent signal or clock. The jitter histogram has a U-shape.
- **Triangular Jitter**: Triangular jitter is always generated by a spread spectrum clock but can also be caused by other periodic sources. Triangular jitter has a uniform jitter distribution.

Intersymbol Interference

Intersymbol interference (ISI) is caused by bandwidth limitations of cables and backplanes. It cuts off higher frequencies and hence changes the shape of the eye opening.

Intersymbol interference can be simulated by inserting a defined transmission path between the Generator and the DUT.

Crosstalk

Crosstalk between adjacent signal paths modulates the vertical eye amplitude. This narrows the voltage range of the receiver needed for capturing data correctly.

For jitter tolerance tests, amplitude modulation is most often simulated by modulating the data output with a sinewave signal. This is called Sinusoidal Interference (SI).

Total Jitter

Total jitter is the sum of all kinds of jitter.

BER Setup

Table 6 BER setup description

Name	Description
Data Rate	Displays the Data Rate for the measurement.
Target BER	Sets the required target BER. The default is 1.0E-09.
Confidence Level	Confidence Level = 1 – exp(-data rate * gate time * measurement depth) The range is 0 % to 99.9 %. The default is 95 %.
Frequency Relax Time	Sets the amount of time to pause the measurement after a change in jitter modulation frequency.
Amplitude Relax Time	Sets the amount of time to pause the measurement after a change in jitter modulation amplitude.

Graph Setup

Table 7 Graph setup description

Name	Description
Template Limits	Show/hides the maximum and minimum jitter amplitude limits in the graph.
Template Points	Show/hides the template points in the graph.
Compliance Limits	Show/hides the compliance curve in the graph.

Instrument Setup

Table 8 Instrument setup description

Name	Description
Generator	The Generator drop down list box shows all available Data Out locations that are used as a data source. A generator must be specified to perform the measurement.
Analyzer	The Analyzer drop down list box shows all available Data In locations that are used for data acquisition. A detector must be specified to perform the measurement.

Measurement Setup

Table 9 Measurement setup description

Name	Description
Measurement Template	Click the 📂 **Open** icon to open the **Jitter Tolerance Measurement** template.
Start Frequency	Sets the start frequency of the **Jitter Tolerance Measurement**.
Stop Frequency	Sets the stop frequency of the **Jitter Tolerance Measurement**.
Num Points	Sets the number of measurement points.
Mode	Sets the Jitter Tolerance Measurement mode. The options are **Compliance** or **Characterization**. In **Characterization** mode you have to set the search algorithms. For details, refer to “Characterization” on page 69. To set Compliance margins, see “Compliance Margin” on page 74.
Step Size	Sets the step size for characterization algorithms. The option are Binary Step Size, Linear Step Size and Log Step Size. For details, see “Step Size” on page 73.
CDR LBW Auto	Automatically sets the external CDR Loop Bandwidth.

Characterization

Table 10 Characterization

Name	Description
Algorithm	Search algorithms apply to characterization measurements only.

Name	Description
Binary	The Binary algorithm uses a variable step size to find the highest passing SJ amplitude at each SJ frequency in the template. Starting at the maximum SJ amplitude determined by the upper limit in the template file or the instrument limit, whichever is lower, followed by the minimum SJ amplitude determined by the lower limit in the template file, the Binary algorithm then sets subsequent SJ amplitudes at the logarithmic midpoint between the highest passing and lowest failing SJ amplitudes. A test point is determined to be a passing point if the BER measured is below the BER threshold set by the user. The equation for the midpoint (c) between points a and b on a log scale is: c = 10^((log(a) + log(b)) / 2). The algorithm exits if the channel is tolerant of the maximum jitter amplitude or is intolerant of the minimum jitter amplitude. Otherwise, the algorithm continues until the last step size is less than the minimum step size set by the user for Binary algorithms. The jitter tolerance result at each jitter modulation frequency is the highest jitter amplitude at which the measurement was made below the threshold BER. However, if the device is intolerant of even the minimum jitter value, the result is not valid and is not plotted. In many cases, the Binary algorithm is the quickest. However, devices with hysteresis may not have consistent results when using the Binary algorithm since the approach direction (up vs. down) will cause the result to vary. This occurs because the PLL in the clock data recovery unit will typically hold lock longer as the jitter is increased, then it would establish lock as the jitter is reduced. For example, a receiver may have a higher passing SJ amplitude when the previous step was a lower (passing) SJ amplitude, compared to a lower passing SJ amplitude when the previous step was a higher (failing) SJ amplitude. If this is a problem, see the Binary + Down Linear and Binary + Up Linear algorithms.

Name	Description
	The following is an example of the Binary algorithm. At a single SJ frequency, if the true pass/fail SJ amplitude is 0.7 UI, the minimum step size is 0.1 UI, the template minimum is 0.1 UI, and the template maximum is 1.0 UI, then the binary algorithm would follow these test points: **1** Template maximum 1.0 UI - FAIL **2** Template minimum 0.1 UI - PASS **3** Log midpoint between 1.0 UI (FAIL) and 0.1 UI (PASS) = 0.32 UI - PASS **4** Log midpoint between 1.0 UI (FAIL) and 0.32 UI (PASS) = 0.57 UI - PASS **5** Log midpoint between 1.0 UI (FAIL) and 0.57 UI (PASS) = 0.75 UI - FAIL **6** Log midpoint between 0.75 UI (FAIL) and 0.57 UI (PASS) = 0.65 UI - PASS **7** Log midpoint between 0.75 UI (FAIL) and 0.65 UI (PASS) 0.70 UI - PASS **8** Last step size < 0.1 UI, highest passing SJ amplitude is 0.70 UI
Binary + down linear	The Binary + Down Linear algorithm is the same as the Binary algorithm followed by the Down Linear algorithm. Refer to the descriptions for Binary and Down Linear in this table.
Binary + up linear	The Binary + Up Linear algorithm is the same as the Binary algorithm followed by the Up Linear algorithm. Refer to the descriptions for Binary and Up Linear in this table.
Down linear	The Down Linear search algorithm starts at the maximum jitter value determined by the template. If a BER measurement has errors above the measurement threshold, the jitter amplitude is adjusted lower by the linear step size. This algorithm exits when the BER is measured below the measurement threshold BER, or if the last amplitude measurement point was at the minimum jitter value in the template. The jitter tolerance result at each jitter modulation frequency is the highest jitter amplitude at which the measurement was made below the threshold BER. However, if the device is intolerant of even the minimum jitter value, the result is not valid and is not plotted.

Name	Description
Down logarithmic	The Down Logarithmic search algorithm starts at the maximum jitter value determined by the template. If a BER measurement has errors above the measurement threshold, the jitter amplitude is adjusted lower by the coefficient calculated from the logarithmic step size. For example, if the coefficient is 10%, then the next amplitude equals the previous amplitude minus 10% of the previous amplitude. This algorithm exits when the BER is measured below the measurement threshold BER, or if the last amplitude measurement point was at the minimum jitter value in the template. The jitter tolerance result at each jitter modulation frequency is the highest jitter amplitude at which the measurement was made below the threshold BER. However, if the device is intolerant of even the minimum jitter value, the result is not valid and is not plotted.
Up linear	The Up Linear search algorithm starts at the minimum jitter value determined by the template. If a BER measurement has errors below the measurement threshold, the jitter amplitude is adjusted higher by the linear step size. This algorithm exits when BER is measured above the measurement threshold BER, or if the last amplitude measurement point was at the max jitter value in the template. The jitter tolerance result at each jitter modulation frequency is the highest jitter amplitude at which the measurement was made below the threshold BER. However, if the device is intolerant of even the minimum jitter value, the result is not valid and is not plotted.
Up logarithmic	The Up Logarithmic search algorithm starts at the minimum jitter value determined by the template. If a BER measurement has errors below the measurement threshold, the jitter amplitude is adjusted higher by the coefficient calculated from the logarithmic step size. For example, if the coefficient is 10%, then the next amplitude equals the previous amplitude plus 10% of the previous amplitude. This algorithm exits when the BER is measured above the measurement threshold BER, or if the last amplitude measurement point was at the maximum jitter value determined by the template. The jitter tolerance result at each jitter modulation frequency is the highest jitter amplitude at which the measurement was made below the threshold BER. However, if the device is intolerant of even the minimum jitter value, the result is not valid and is not plotted.

Name	Description
Up Log + Linear	The Up Log + Linear search algorithm performs the same algorithm as Up Logarithmic, but in addition, it returns to the last passing amplitude and steps linearly up until it reaches a fail point. The algorithm continues to increase the SJ amplitude until failure, even if it surpassed the original failed point.
Adaptive Binary	The adaptive binary algorithm is a combination of an up logarithmic search and a binary search. The initial search at the start frequency uses the logarithmic step size for the upwards search and switches to a binary search when the first failed measurement point is found. The binary search is reducing the step size until it becomes smaller as the defined binary step size. For all following jitter frequencies, the search starts at the last known pass amplitude. This algorithm is most efficient when the search is configured to start with the high jitter frequency, and therefore usually less points to measure until it finds the first fail. The advantage over the binary algorithm is, that it avoids measuring unnecessary pass points at low jitter amplitude, as well as avoiding the maximum jitter amplitude. Avoiding the excessive jitter condition helps saving measurement time because the number of times the device under test needs to recover will be less than with a simple binary search.
Custom	Custom algorithms can be implemented using the DUT Control Interface. This allows to implement search algorithms that select the jitter amplitude in a way that it optimizes measurement time, or excessive jitter conditions in a way that is not covered by the standard search algorithms. The search algorithm is defined by the implementation of three dedicated DUT Control Interface hooks. ▪ JTOL_CustomAlgorithm_InitializeFrequency ▪ JTOL_CustomAlgorithm_GetNextAmplitude ▪ JTOL_CustomAlgorithm_IsLastPointAtCurrentFrequency A tutorial script 'JTOL-Custom-Algorithm-with-Virtual-DUT.py' is part of the M8070ADVB installation and is demonstrates how to implement the algorithm correctly. This algorithm requires the DUT Control Interface license (M8070ADVB-1xx).

Step Size

Table 11 Step size descriptions for characterization algorithms

Name	Description
Binary Step Size	The Step Size for the Binary algorithm defines the exit criteria for the algorithm. The Binary search algorithm stops once its step size falls below this user defined Step Size. The Binary Step Size applies to the Binary algorithm and the binary portions of the Binary + Down Linear and Binary + Up Linear algorithms..
Linear Step Size	This defines the step size for each step of the Up Linear and Down Linear algorithms. When Down Linear is enabled as the search algorithm, the measurement starts from the maximum (which depends on the jitter frequency). A step size of 100 mUI, for example, may result in a sequence of 1000 UI, 999.9 UI, 999.8 UI, and so on. The test for one frequency stops when the BER limit is met or minimum specified amplitude is reached. When Up Linear is enabled as the search algorithm, the measurement starts from the minimum. A step size of 100 mUI, for example, will result in a sequence of 0.1 UI, 0.2 UI, 0.3 UI, and so on. The test for one frequency stops when the BER limit is exceeded or the maximum amplitude is reached.
Log Step Size	This defines the step size for each step of the Up Logarithmic and Down Logarithmic algorithms. When Down Logarithmic is enabled as the search algorithm, the measurement starts from the maximum (which depends on the jitter frequency). A percentage of 50%, for example, may result in a sequence of 1000 UI, 500 UI, 250 UI, and so on. The test for one frequency stops when the BER limit is met or the specified minimum amplitude is reached. When Up Logarithmic is enabled as the search algorithm, the measurement starts from the specified minimum amplitude. A percentage of 50%, for example, may result in a sequence of 0.1 UI, 0.15 UI, 0.23 UI, and so on. The test for one frequency stops when the BER limit is crossed or the maximum amplitude (which depends on the jitter frequency) is reached.

Compliance Margin

Table 12 Compliance Margin

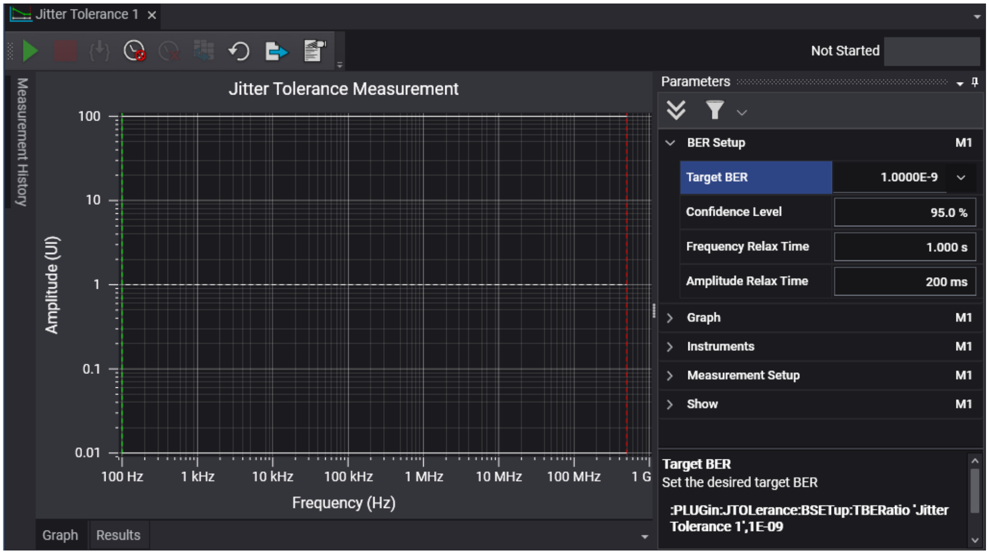
Name	Description
Compliance Margin	In Compliance mode, the test starts at points defined on the jitter tolerance curve and is increased or decreased by the specified margin percent. The compliance template is adjusted according to the equation SJ Amplitude (with margin) = original SJ Amplitude * (1.0 + margin(%)/100). This can be used to test if a receiver can withstand a certain percentage more jitter than dictated by the compliance template. Conversely, if a receiver is failing, the margin can be lowered to test if a receiver can pass a compliance test with a less stringent template.

Launching Jitter Tolerance Measurement

To launch the **Jitter Tolerance Measurement**:

- Go to **Menu Bar** > **Measurements** and then select **Jitter Tolerance**.

The **Jitter Tolerance** user interface will appear as shown in the following figure:



Not Started

BER Setup M1

Target BER 1.0000E-9

Confidence Level 95.0 %

Frequency Relax Time 1.000 s

Amplitude Relax Time 200 ms

Graph M1

Instruments M1

Measurement Setup M1

Show M1

Target BER
Set the desired target BER

:PLUGin:JTOLerance:BSETup:TBERatio 'Jitter Tolerance 1',1E-09

The **Jitter Tolerance** user interface includes the following GUI elements:

- **Toolbar**: For details, refer to “Toolbar” on page 27.
- **Status Indicator**: For details, refer to “Status Indicator” on page 28.
- **Measurement History Window**: For details, refer to “Measurement History Window” on page 29.
- **Measurement Graph**: For details, refer to “Measurement Graph” on page 76
- **Results Pane**: For details, refer to “Viewing the Jitter Tolerance Results” on page 77.

How to Run Jitter Tolerance Measurement

Once you have specified the Frequency, BER, and Search criteria, press the ▶ **Start Measurement** icon to run the measurement. The measurement status indicator will show the progress.

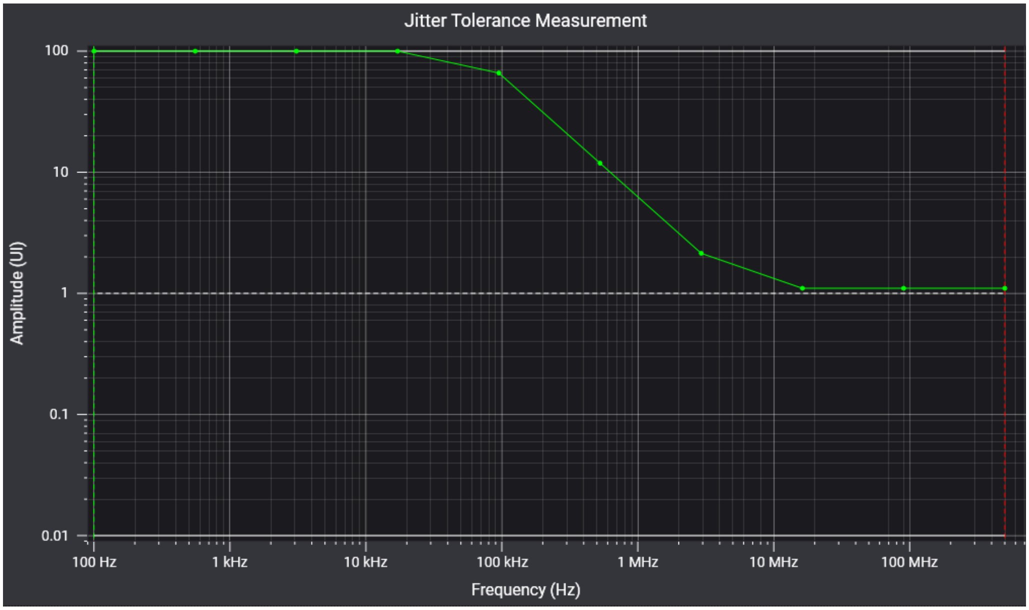
How to Stop/Abort Jitter Tolerance Measurement

To stop a measurement, click the 🟥 **Stop Measurement** icon.

Measurement Graph

After a jitter tolerance measurement has been run, results can be viewed by clicking on the **Graph** tab. The graph displays the sinusoidal jitter frequency on the x-axis, and the sinusoidal jitter amplitude on the y-axis.

The following figure shows the measurement results in the graph view.



Result Points

The graph displays the BER measurement results of each tested sinusoidal jitter point. Passing results are displayed as a green dot. If the **Show All Points** option is enabled, all measured pass points will be shown by green dots, else the only last passed result for a given frequency will be shown by a green dot. If this green dot is selected, its respective measurement result can be seen as highlighted in the **Results** tab and vice-versa. Also, a green line connects the last passed result (green dot) of each frequency. In the graph, the green “+” means that you’ve reached the instruments max jitter capability without a fail. The red cross "x" means that you have failed to reached the instruments max jitter capability.

Maximum and Minimum Template Limit Lines

The solid white lines on the graph show the search range during characterization. This feature reduces test time by eliminating points where the DUT is likely to pass when its performance range is known.

Template Points

The green dots on the graph are the measurement points defined in the Template File.

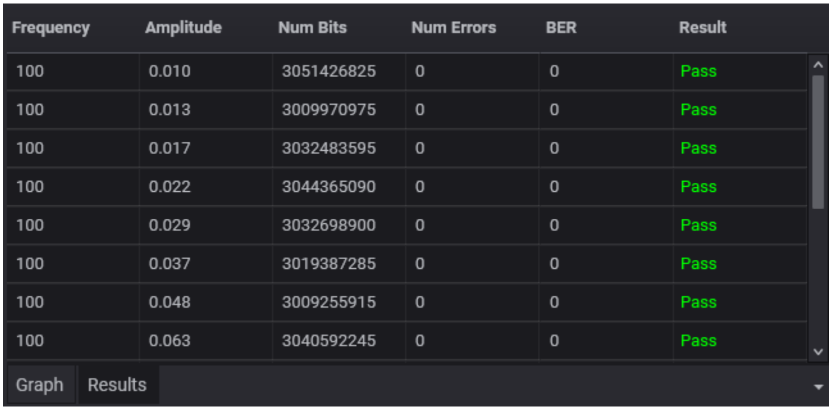
Compliance Limit Line

The dashed white line on the graph shows the compliance test defined in the Template File.

Viewing the Jitter Tolerance Results

Measurement results can also be viewed in tabular form by clicking on the **Results** tab.

The following figure shows the results at each measurement point in tabular form.



Frequency	Amplitude	Num Bits	Num Errors	BER	Result
100	0.010	3051426825	0	0	Pass
100	0.013	3009970975	0	0	Pass
100	0.017	3032483595	0	0	Pass
100	0.022	3044365090	0	0	Pass
100	0.029	3032698900	0	0	Pass
100	0.037	3019387285	0	0	Pass
100	0.048	3009255915	0	0	Pass
100	0.063	3040592245	0	0	Pass

If a passed result point (green dot) on the measurement graph is selected, its respective measurement results can be seen as highlighted in the **Results** tab and vice-versa.

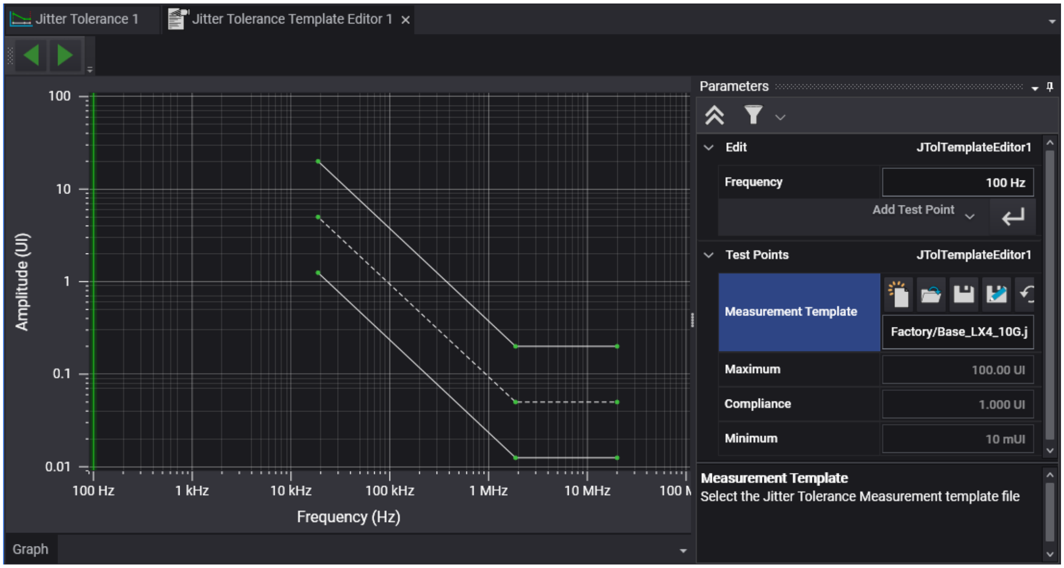
NOTE

The BER measurement processes at least as many bits as defined by the acquisition parameters, it stops when the end criteria is reached or exceeded. Due to update period and hardware granularities, an individual BER measurement will typically process more bits than configured.

Using the Jitter Tolerance Template Editor

The **Jitter Tolerance Template Editor** is used to modify the currently loaded template file. It can be accessed by clicking on the **Open Jitter Tolerance Template Editor** icon, present on the toolbar.

The following is an example of jitter tolerance template editor showing the measurement point values (frequency and amplitude) for the currently loaded template file.



Edit JTolTemplateEditor1

Frequency 100 Hz

Add Test Point

Test Points JTolTemplateEditor1

Measurement Template Factory/Base_LX4_10G.j

Maximum 100.00 UI

Compliance 1.000 UI

Minimum 10 mUI

Measurement Template
Select the Jitter Tolerance Measurement template file

Template Editor Toolbar

Table 13

Elements	Name	Description
◀ ▶	Previous /Next Measurement Point	The left and right arrow buttons cause the vertical marker to jump to the next/previous measurement point.
▼	Auto Scale Graph	Scales the graph to the optimum setting.

Template Editor Functions

Table 14

Function	Description
Edit	
Frequency	The frequency at the current vertical marker position is displayed in this window. Changing this frequency using the keypad, allows new measurement points to be defined.
Add Test Point/Remove Test Point	Use the drop-down menu to add or remove test points and then click on the **Execute** button.
Test Points	
Measurement Template	Selects the **Jitter Tolerance Measurement** template file. It provides the following functions: **New** - Opens a new template file. **Open...** - Opens an existing template file which includes limits, compliance, and jitter tolerance measurement state. **Save** - Saves the currently open template file to the current filename. **Save As...** - Saves the currently open template file to a different filename and/or location.

Function	Description
Maximum	Changes the maximum amplitude of the measurement point at the current vertical marker position. These are the points along the solid white line on the graph.
Compliance	Changes the compliance amplitude of the measurement point at the current vertical marker position. These are the points along the upper solid white line on the graph.
Minimum	Changes the minimum amplitude of the measurement point at the current vertical marker position. These are the points along the lower solid white line on the graph.

Eye Diagram Measurement

What is an Eye Diagram?

The M8070B System Software provides quick design analysis with the Eye Diagram capability.

The Eye Diagram allows a quick check for the DUT’s signal output, and determines the signal quality. The eye contour lines display the measured eye at a deeper BER level, for accurate results.

The Eye Diagram feature provides the following measurement capabilities:

- Eye Diagram Waveform
- BER Contour Lines
- Automatic Measurement for the most relevant eye parameters:-
  - Eye Height
  - Eye Width
  - Jitter P-P
  - Jitter RMS
  - Cross Voltage

The Eye Diagram generates a three dimensional graph of the bit error rate (BER).

This measurement helps in determining and analyzing the quality of the DUT’s signal output. The Eye Diagram results comprise of voltage(y), time(x), and BER(z).

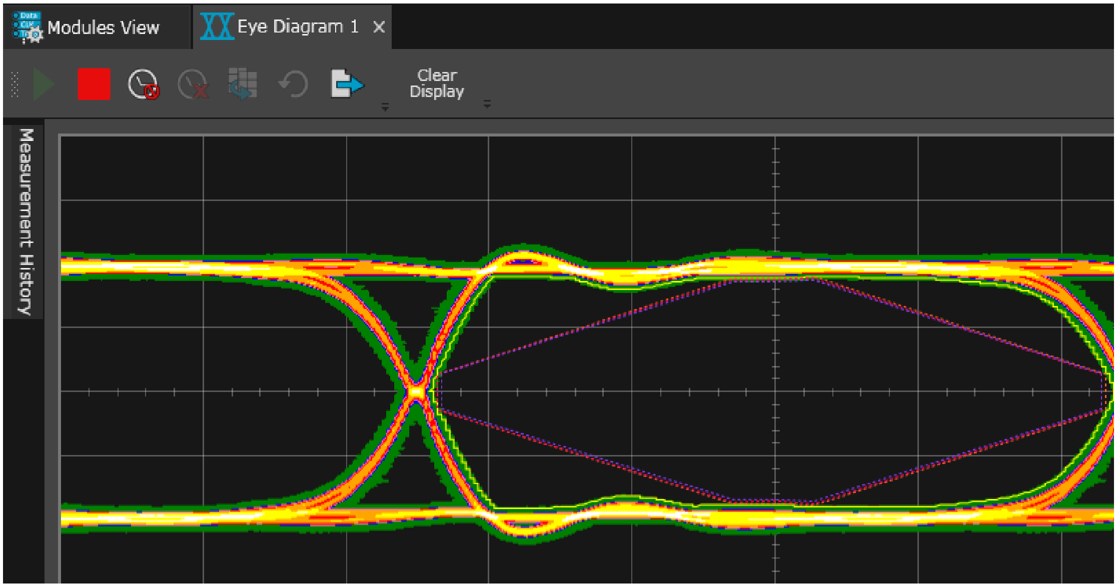
The Eye Diagram, and the table of Calculated Eye Measurement Results form simultaneously. They become more precise with the increasing number of measured bits.

Once the M8070ADVB plugin license is installed, it also enables the M8043A analyzer to generate an histogram-based eye diagram of the input signals using the **M8070B Sampling Point Setup** window. For details, refer to the *M8000 Series User Guide*.

The histogram-based approach allows:

- More frequent updates
- Display NRZ as well as PAM4 encoded signals

The following graph shows the Eye Diagram integrated in the M8070B system software:



Methods of Representation of Eye Diagram

There are two methods of representing the eye diagram:

- Waveform
- Contour

Waveform

Waveform is the shape, and form of a signal. The waveform graph shows the periodical variation of voltage against time.

The waveform in the M8020A is similar to the one in the oscilloscope. In this case, the waveform initially gives a coarse, but quick picture of the signal quality; while the ‘smooth waveform’ quickly generates a high resolution graph. The waveform is displayed in an incremental way, showing the coarse picture quickly, and then refining it further.

The BER Threshold is configurable. The BER Threshold is the level up to which the signal is represented as waveform, and BER values below this threshold are represented as contours.

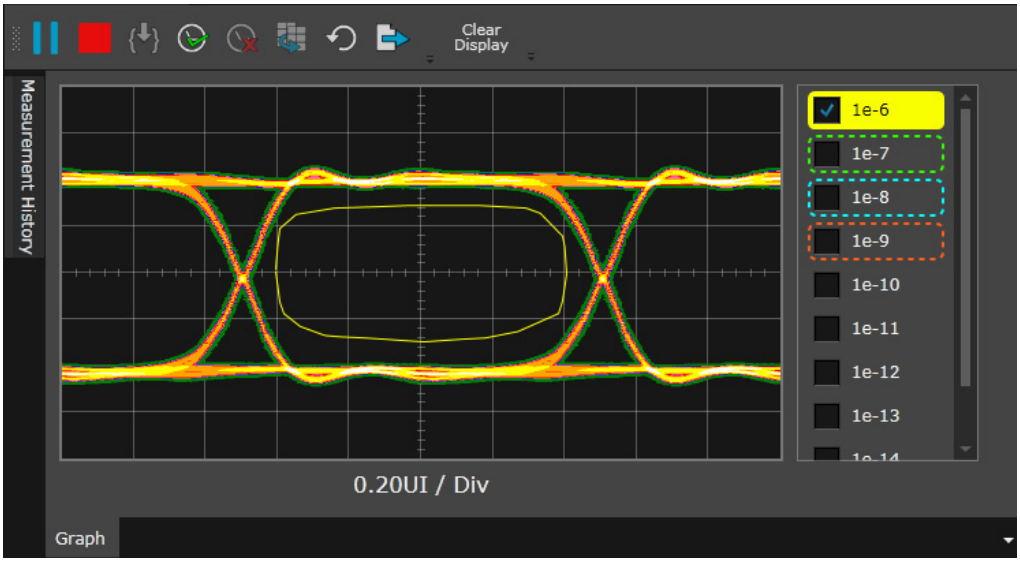
BER Contour

Contour is a curve connecting points where the BER has a same particular value.

The contour graph is plotted within the Eye Diagram, and it helps to determine the Eye Opening at deep bit error rates, such as 1e-10, 1e-12, and so forth. Depending upon the user's requirement the deep BERs can be calculated. The extrapolated eye contour lines display the eye opening for lower BER levels, such as 1e-15. The BER is displayed as a function of sampling delay, and sampling threshold.

The eye opening is one of the main characteristics of a high speed device.

The legend on the right side of the screen represents the contour BER values in different colors. When you select a BER value from the legend, it displays the contour for the selected BER in the same color. Initially all the BER values on the legend are struck across. When the contour is measured, the BER value changes into normal font; and when the contour is extrapolated the BER value changes into italics. A BER value for which a Contour does not exist is struck across. You can select multiple BER values. The screenshot below displays the contour and the legend.



The outer contour represents the measured BER contour, while the inner lines are extrapolated.

The Minimum Number of Errors indicates the minimum number of hits required to plot the contour.

Eye Diagram Measurement Parameters

Acquisition Parameters

Name	Description
Analyzer	Location or location group against which the data acquisition is performed.
Persistence	Set the criteria for Persistence.

Name	Description
Infinite	With 'Infinite', the eye appears with the Automated Eye Parameter Measurement, and the results keep on improving in accuracy, as more number of points get measured and represented on the diagram along with the deeper BER values. The BER threshold in the Graph Tab decides how long the measurements will be done at the boundaries of the eye. Once the measured points exceed the BER Threshold defined by the user, the measurement is done deep inside the eye. This is a default setup for persistence.
Fixed Time (Secs)	With 'Secs', the measurement runs for the specified time, and the measurement automatically restarts showing only the refined eye diagram. To stop the measurement, click the abort button.
Number of eyes	Set the criteria for the number of eyes. 1.5ui This option displays '1.5' eyes on the graph. The transitions of the complete eye are placed such, that, they show '0.25' eyes on both sides. So, the total number of displayed eyes is 1.5. 2ui This option displays '2.0' eyes on the graph. The transitions of the complete eye are placed such, that, they show '0.5' eyes on both sides. So, the total number of displayed eyes is 2.0.

Evaluation Parameters

Name	Description
Calculate Results For	Set the criteria for calculating the eye width, eye height, JPP, JRMS, cross voltage according to the BER threshold selected.
0 Errors	To see ‘0 errors’ select the option. This gives the results according to the last measured contour for the current measurement. This is more significant in the case of deep BERs.
BER Threshold	Enter the BER threshold at which the five results, eye width, eye height, JPP, JRMS, and cross voltage, will be calculated. The show checkbox displays the contour at which the results are calculated.
Transition Time	Set criteria for transition time 10/90 Measures horizontal scan from 10% to 90% of the signal amplitude. 20/80 Measures horizontal scan from 20% to 80% of the signal amplitude.

Graph Settings

Name	Description
Display Unit	Set the criteria for the Timing Units: Unit Interval Unit used to measure delay relative to the eye width. Seconds Unit used to measure eye width in absolute terms of time.
Show Waveform	This option shows the waveform graphic.
Smooth Waveform Graphics	The smooth waveform gives a finer waveform output. It logarithmically interpolates between the measured points to give a finer waveform.

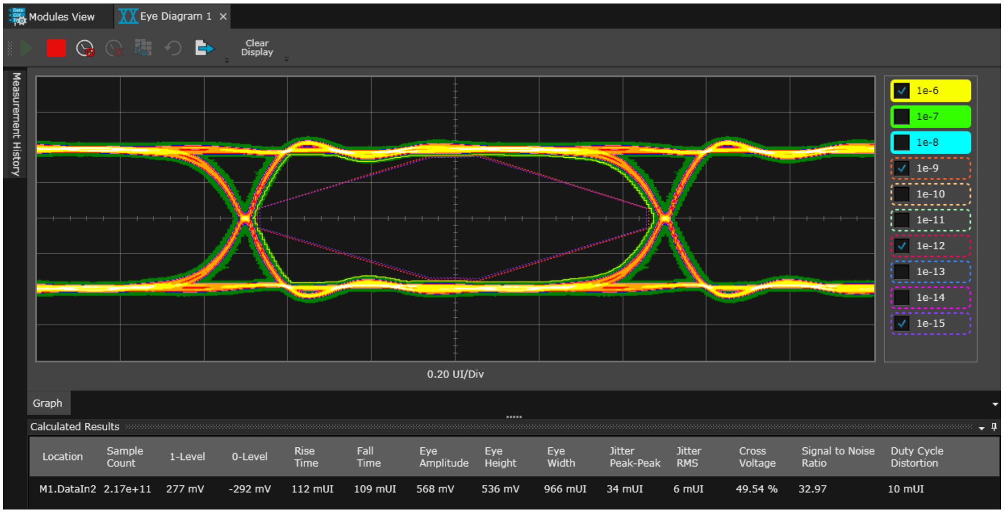
Name	Description
Waveform BER Threshold	The BER Threshold value allows the user to measure around the eye boundaries first, and then, deeper inside the Eye for BERs beyond the specified threshold.
Contours	This option plots the contour.
Contour Legend	This shows the list of BER values.

Launching the Eye Diagram Measurement

To launch the **Eye Diagram Measurement**:

- Go to **Menu Bar** > **Measurements** and then select **Eye Diagram**.

The **Eye Diagram** user interface will appear as shown in the following figure:



Location	Sample Count	1-Level	0-Level	Rise Time	Fall Time	Eye Amplitude	Eye Height	Eye Width	Jitter Peak-Peak	Jitter RMS	Cross Voltage	Signal to Noise Ratio	Duty Cycle Distortion
M1.DataIn2	2.17e+11	277 mV	-292 mV	112 mUI	109 mUI	568 mV	536 mV	966 mUI	34 mUI	6 mUI	49.54 %	32.97	10 mUI

The **Eye Diagram** user interface includes the following GUI elements:

- **Toolbar**: For details, refer to “Toolbar” on page 27.
- **Status Indicator**: For details, refer to “Status Indicator” on page 28.
- **Measurement History Window**: For details, refer to “Measurement History Window” on page 29.

- **Results Pane**: For details, refer to “Calculated Eye Diagram Results” on page 88.

How to Start/Abort an Eye Diagram Measurement

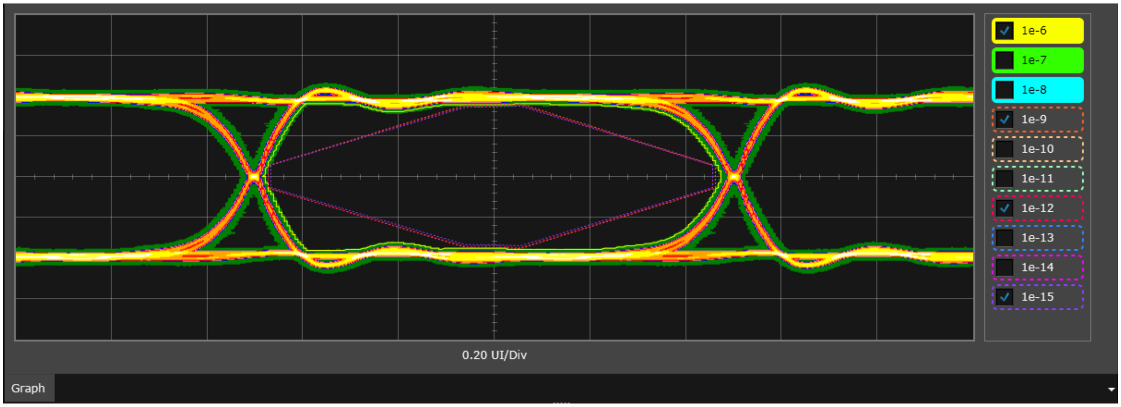
To initiate the eye diagram measurement:

1 Click the Eye Diagram menu item from the Analysis submenu.

2 Click the Start button to execute the measurement.

The measurement runs, and the eye diagram along with the Calculated Eye Diagram Results get updated constantly.

The following graph shows the eye diagram.



3 To stop the measurement click on Abort.

How to Change the Default Settings

To achieve desired results, you can change the measurement parameters through the Parameters window.:

1 Click the Properties button to open the Properties dialog box.

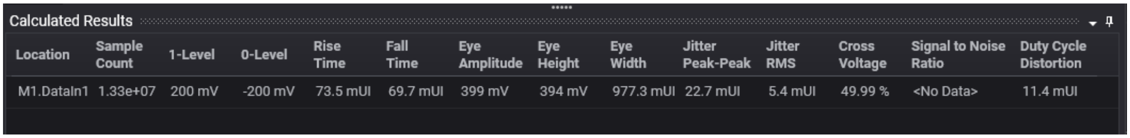
2 Use the different tabs in this dialog box to make the required settings:

For details, see Acquisition, Evaluation and Graph Settings.

Please note that you can only change the Evaluation and Graph Settings while the measurement is running.

Calculated Eye Diagram Results

The following results are displayed when an eye diagram measurement is completed:



Calculated Results

Location	Sample Count	1-Level	0-Level	Rise Time	Fall Time	Eye Amplitude	Eye Height	Eye Width	Jitter Peak-Peak	Jitter RMS	Cross Voltage	Signal to Noise Ratio	Duty Cycle Distortion
M1.DataIn1	1.33e+07	200 mV	-200 mV	73.5 mUI	69.7 mUI	399 mV	394 mV	977.3 mUI	22.7 mUI	5.4 mUI	49.99 %	<No Data>	11.4 mUI

- **Location**: Location or Location/Group against which the data acquisition is performed.
- **Sample Count**: The number of points measured for the eye diagram. If persistence, ‘infinite’, is enabled, the value will gradually increase.
- **Error Bits**: Displays the number of errored bits during the accumulation period.
- **1-Level:** One Level is a measure of the mean value of the logical 1 of an eye diagram.
- **0-Level:** Zero level is the measurement of the mean value of the logical 0 of an eye diagram.
- **Rise time:** Measures the mean transition-time taken by the data on the rising edge of the eye diagram. The data crosses three thresholds: lower, crossover point, upper and the eye transition.

  Note: Rise time is dependent on the Transition Time of the View Tab. The two options are 10/90 and 20/80.
- **Fall time:** Measures the mean transition-time taken by the data on the falling edge of the eye diagram. The data crosses three thresholds: lower, crossover point, upper and the eye transition.

  Note: Fall time is dependent on the Transition Time of the View Tab. The two options are 10/90 and 20/80.
- **Eye Amplitude:** The difference between the logic 1 level and the logic 0 level.
- **Eye Height:** Measures the vertical opening of an eye diagram with respect to BER threshold. This determines “eye closure” due to noise.

  Note: The Eye Height is calculated according to the BER Threshold set in the View Tab. Eye Height is calculated with 'Height at' in the View tab.

- **Eye Width:** The horizontal measurement of the eye opening at a specified BER Threshold.

  Note: The Eye Width is calculated according to the BER Threshold set in the View Tab. Eye Width is calculated as per the selection of either Width at Crossing Point or Custom defined Width.

- **Jitter Peak-Peak:** Full width of the histogram at the eye crossing point with respect to BER threshold.

  Note: The Jitter P-P is calculated according to the BER Threshold set in the Parameter window.

- **Jitter RMS:** A standard deviation of the crossing point histogram with respect to BER threshold.

  Note: The Jitter RMS is calculated according to the BER Threshold set in the Parameter window.

- **Cross Voltage:** Crossing percentage is a measure of the amplitude of the crossing points relative to the 1 level and 0 level.

  Note: The Cross Voltage is calculated according to the BER Threshold set in the Parameter window.

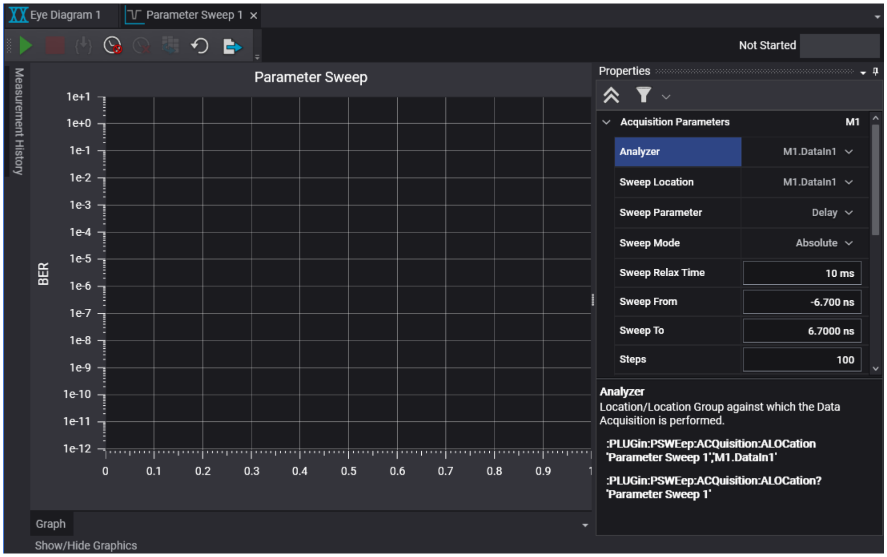
- **Signal to Noise Ratio:** In signal-to-noise, the "signal" is the information power of the signal indicated by the difference between the ‘1’, and ‘0’ level. The "noise" is the combined standard deviations of the ‘1’ level spread, and the ‘0’ level spread.
- **Duty Cycle Distortion:** This value is the difference between the period of a 1 bit and a 0 bit.

Parameter Sweep Plugin

The **Parameter Sweep** plugin allows a user to sweep the parameters of the selected location/location group to the desired analyzer in order to calculate BER. This is helpful in scenarios when a user wants to pick a property to sweep within a specified range with specified steps and measure BER with respect to that sweep. Using this plugin, a user can sweep the parameters using the acquisition parameters provided in the Parameter window and then run the measurement to calculate BER.

Launching Parameter Sweep Plugin

To launch the **Parameter Sweep** plugin, go to the **Menu Bar** > **Measurements** and then select **Parameter Sweep**. The user interface of **Parameter Sweep** plugin will appear as shown in the following figure:



Acquisition Parameters M1

Analyzer	M1.DataIn1
Sweep Location	M1.DataIn1
Sweep Parameter	Delay
Sweep Mode	Absolute
Sweep Relax Time	10 ms
Sweep From	-6.700 ns
Sweep To	6.7000 ns
Steps	100

Analyzer
Location/Location Group against which the Data Acquisition is performed.

```
:PLUGin:PSWEep:ACQuisition:ALOCation
'Parameter Sweep 1','M1.DataIn1'

:PLUGin:PSWEep:ACQuisition:ALOCation?
'Parameter Sweep 1'
```

The **Parameter Sweep** Plugin user interface includes the following GUI elements:

- **Toolbar**: For details, refer to “Toolbar” on page 27.
- **Status Indicator**: For details, refer to “Status Indicator” on page 28.
- **Measurement History Window**: For details, refer to “Measurement History Window” on page 29.

- **Parameters Window**: For details, refer to "Parameter Sweep Plugin Parameters" on page 91.
- **Measurement Graph**: For details, refer to "Measurement Graph for Parameter Sweep" on page 94.
- **Measurement Data**: For details, refer to "Measurement Data for Parameter Sweep" on page 95.

Parameter Sweep Plugin Parameters

Measurement Setup

The measurement setup includes the following acquisition parameters:

- **Analyzer** - This parameter lists all available Data In locations that are used for data acquisition.
- **Sweep Location** - This parameter sets the location/location group string against which the sweep property needs to be performed.
- **Sweep Parameter** – This parameter sets the sweep parameter to be swept amongst the list of other parameters present in current location or group.
- **Sweep Mode** - This parameter sets the sweep mode. The choices are 'ABSolute' and 'RELative'.
  - In the 'ABSolute' mode, the property will sweep from the value specified in the 'Sweep From' parameter to a value specified in the 'Sweep To' parameter.
  - In the 'RELative' mode, the property will be sweep from a value relative to the current value specified in the 'Sweep From' parameter to a value relative to the current value specified in the 'Sweep To' parameter.
- **Sweep Relax Time** - This parameter sets the relaxing time between the sweep before reading the BER. The relaxing time can be between 0 sec to 60 sec.
- **Sweep From** - This parameter sets the value from where the sweep needs to be performed for that particular parameter.
- **Sweep To** - This parameter sets the value till the sweep needs to be performed for that particular parameter.
- **Steps** - This parameter sets the value as for how many steps user wants to sweep the parameter between the current "Sweep From" and "Sweep To" range.
- **Steps Accumulation End** - This parameter sets the steps accumulation end criteria to sweep. The choices are "No. of Bits" or "Pass/Fail".

- In case the **Step Accumulation End** is selected as **Pass/Fail**, set **Target Error Ratio** and **Target Confidence Level**.
  - **Target Error Ratio** - This parameter sets the target error ratio of the accumulated results.
  - **Target Confidence Level** - This parameter sets the target confidence level of the accumulated results.
- In case the **Step Accumulation End** is selected as **No. of Bits**, specify **Number of Compared Bits** and **Number of Errors**.
  - **Number of Compared Bits** - This parameter allows the user to specify the number of compared bits; that would be used as criteria for moving to the next measurement point in case when the sample delay optimization is “NONE”.
  - **Number of Errors** - This parameter enables the user to set the additional no. of errored bits to consider while moving to the next measurement points. To enable this feature, you have to click on the toggle button.
- **Pre BER Alignment** - This parameter sets the pre BER alignment criteria before it sweeps to next measurement point. The choices are:
  - **None** - This parameter does not initiate any pre BER alignment.
  - **Data Alignment** - This parameter initiates pattern resynchronization.
  - **Sample Delay Optimization** - This parameter initiates search for the value of data/clock delay that puts the active clock edge in the center of the data eye, midway between the two relative delay points with a measured BER
  - **Threshold Optimization** - This parameter initiates search for the 0/1 threshold voltage midway between the two 0/1 threshold voltages with a measured BER
  - **Full Sample Point Optimization** - This parameter initiates auto alignment.

Graph Setup

The graph setup includes the following parameters:

- **Vertical Axis Scale** – This parameter specify the Y axis scale. The choices are Linear and Logarithmic. If Linear is specified then the BER values are in terms of a linear scale. If Logarithmic is specified then the BER values are in terms of a logarithmic scale.
- **Measured Points** – This parameter show/hides the measured points on the graph. When this option selected, all measured points are indicated on the graph.
- **Graph Legends** - This parameter show/hide graph legends. When this option selected, the legends are shown on the right side of the graph. The color scheme of the graph corresponds to color of the respective legend.

How to Run Parameter Sweep Measurement

To run an **Parameter Sweep** measurement, perform the following steps:

- Use the **Parameters** window to select the location or location group for which the data acquisition has to be performed.
- Set the acquisition and graph setting parameters for **Parameter Sweep** in the **Parameters** window. For details, refer to “Parameter Sweep Plugin Parameters” on page 91.
- Click the ▶ **Start Measurement** icon to run the measurement. The measurement status indicator will indicate **Running**.
- Wait till the status indicator shows **Finish** status.

NOTE

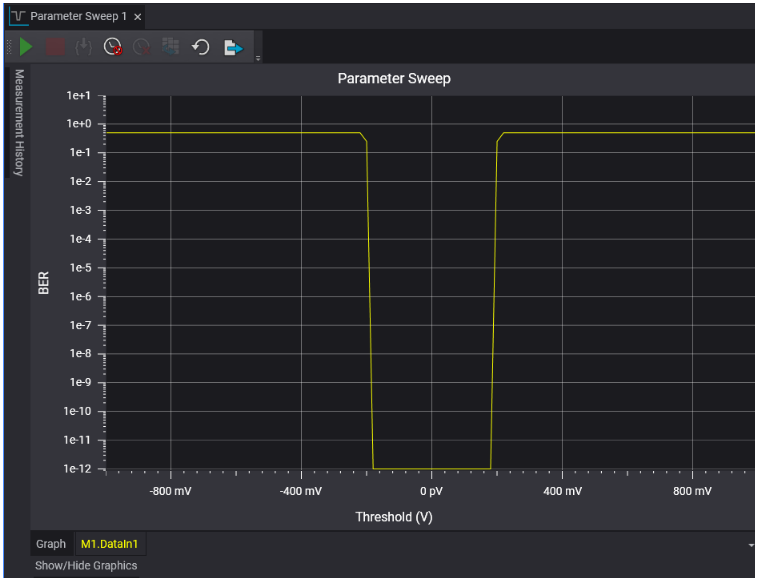
Please note that once you run the measurement you cannot modify the acquisition parameters. However, if you try to modify acquisition parameters by stopping the measurement and then run the measurement, a new instance of measurement will be executed.

How to Stop Parameter Sweep Measurement

To stop a measurement, click the 🟥 **Stop Measurement** icon to stop the measurement.

Measurement Graph for Parameter Sweep

Once you run a **Parameter Sweep** measurement for a specified duration, the calculated graph and the measurement data is shown on the measurement graph as follows:



The **Measurement Graph** contains the following tabs:

- **Graph**: Displays the graphical representation of the measurement. You can apply various graph settings to the selected graph to view it as per the requirement. For graph settings, refer to "Graph Setup" on page 93.
- **Location**: Displays the raw measurement data for the location/location group. If you are running measurement for a group, multiple tabs will appear that display the raw measurement for each location.

NOTE The threshold value is displayed in volts for electrical signals. In the case of an optical signal, when the probe is connected with real-time scope as ED, the graph will show the threshold value in watt.

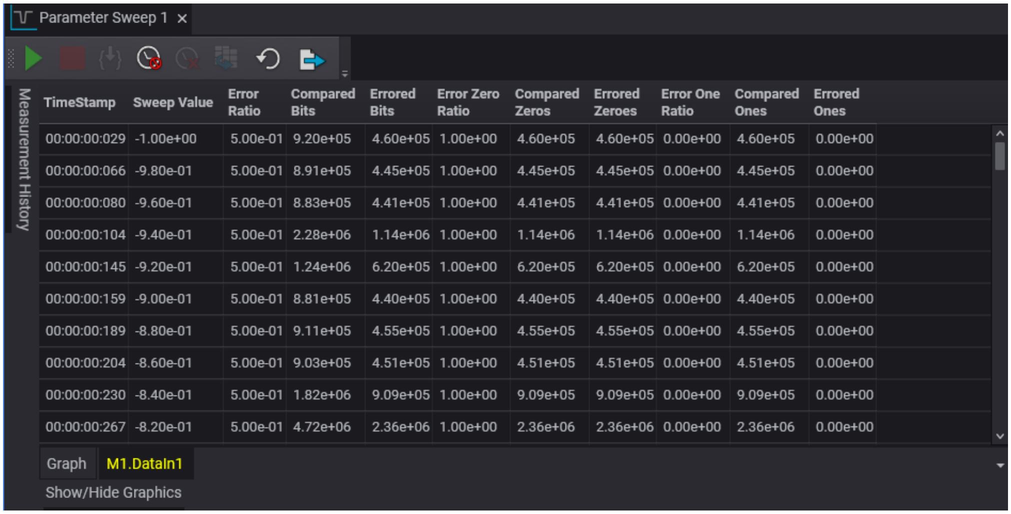
When you right-click on the **Measurement Graph**, a context menu appears which provides the following options:

- **Turn ON/OFF Fit to view (Ctrl+Home)**- Turns ON/OFF **Fit to view** option.
- **Fit to view (Home)** - Makes the visible area fit to display entire contents.
- **Copy screenshot (F11)** - Copies the screenshot of charts to clipboard.
- **Save screenshot (Ctrl+S)** - Saves the screenshot as an image (PNG) under a name.
- **Quick Help (Alt+F1)** - Opens a window that provides brief information about the dynamic display.

Measurement Data for Parameter Sweep

The measurement data can be viewed by clicking on the location/location group tab which appears at the bottom of the measurement graph. This tab is only available once the parameter sweep measurement is successfully executed.

The following figure shows the measurement data for the selected location.



TimeStamp	Sweep Value	Error Ratio	Compared Bits	Errored Bits	Error Zero Ratio	Compared Zeros	Errored Zeroes	Error One Ratio	Compared Ones	Errored Ones
00:00:00:029	-1.00e+00	5.00e-01	9.20e+05	4.60e+05	1.00e+00	4.60e+05	4.60e+05	0.00e+00	4.60e+05	0.00e+00
00:00:00:066	-9.80e-01	5.00e-01	8.91e+05	4.45e+05	1.00e+00	4.45e+05	4.45e+05	0.00e+00	4.45e+05	0.00e+00
00:00:00:080	-9.60e-01	5.00e-01	8.83e+05	4.41e+05	1.00e+00	4.41e+05	4.41e+05	0.00e+00	4.41e+05	0.00e+00
00:00:00:104	-9.40e-01	5.00e-01	2.28e+06	1.14e+06	1.00e+00	1.14e+06	1.14e+06	0.00e+00	1.14e+06	0.00e+00
00:00:00:145	-9.20e-01	5.00e-01	1.24e+06	6.20e+05	1.00e+00	6.20e+05	6.20e+05	0.00e+00	6.20e+05	0.00e+00
00:00:00:159	-9.00e-01	5.00e-01	8.81e+05	4.40e+05	1.00e+00	4.40e+05	4.40e+05	0.00e+00	4.40e+05	0.00e+00
00:00:00:189	-8.80e-01	5.00e-01	9.11e+05	4.55e+05	1.00e+00	4.55e+05	4.55e+05	0.00e+00	4.55e+05	0.00e+00
00:00:00:204	-8.60e-01	5.00e-01	9.03e+05	4.51e+05	1.00e+00	4.51e+05	4.51e+05	0.00e+00	4.51e+05	0.00e+00
00:00:00:230	-8.40e-01	5.00e-01	1.82e+06	9.09e+05	1.00e+00	9.09e+05	9.09e+05	0.00e+00	9.09e+05	0.00e+00
00:00:00:267	-8.20e-01	5.00e-01	4.72e+06	2.36e+06	1.00e+00	2.36e+06	2.36e+06	0.00e+00	2.36e+06	0.00e+00

The measurement data displays measurement points such as Sweep Value, Error Ratio, Compared Bits, Errored Bits, Errored Zero Ratio, Compared Zeros, Errored Zeros, Error One Ratio, Compared Ones and Errored Ones which are repeated in the same order for subsequent measured points.

NOTE

The BER measurement processes at least as many bits as defined by the acquisition parameters, it stops when the end criteria is reached or exceeded. Due to update period and hardware granularities, an individual BER measurement will typically process more bits than configured.

3 Utilities

This chapter describes the utilities provided by the by Advanced Measurement Package.

Overview

The **Advanced Measurement Package** provides the following utilities:

- Script Editor
- DUT Control Interface
- SCPI Editor

NOTE

Please note that the utilities supported by the Advanced Measurement Package requires license for its activation. For details on license, see "License Information" on page 18.

Script Editor

The **Script Editor** provides flexibility to the programmers to automate their plug-ins thus allowing them to do everything they want in the measurements.

The **Script Editor** uses the IronPython programming language which is a flexible and powerful language suitable to a range of tasks. The IronPython is an open-source implementation of the Python programming language which is tightly integrated with the .NET Framework. IronPython can use the .NET Framework and Python libraries, and other .NET languages can use Python code just as easily. Whether you're interested in writing full applications, tackling scripting tasks, or embedding a scripting language into another application, IronPython has something to offer you.

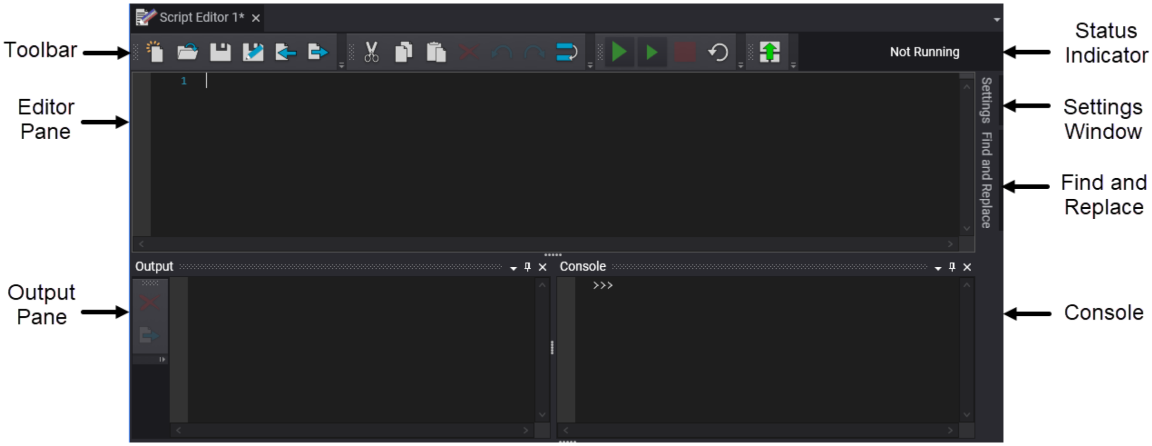
For more details on the IronPython, please refer to the following web-page: http://ironpython.net

Launching the Script Editor

To launch the **Script Editor**:

- Go to the **Menu Bar** > **Utilities** and then select **Script Editor**.

The **Script Editor** interface will appear as shown in the following figure:



The **Script Editor** interface consists of the following GUI elements:

- Toolbar
- Editor Pane
- Output Pane
- Console
- Settings Window
- Find and Replace Dialog

The listed GUI elements are described in the section that follows.

Toolbar

The **Script Editor** toolbar provides the following convenient script editing functions:

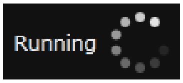
Table 15

Icon	Name	Keyboard Shortcut	Description
🗋	New	CTRL + N	Click this icon to create a new project/module. For details, refer to “Creating a Script” on page 108
💾	Save	CTRL + S	Click this icon to save the current project/module. For details, refer to “Saving a Script” on page 108.
⇦	Import	CTRL + O	Click this icon to import external project/module (.PY) in to the current project/module. For details, refer to “Importing a Script” on page 108.
⇨	Export	CTRL + E	Click this icon to export the project/module (.PY) to the local drive. For details, refer to “Exporting a Script” on page 109.
✂	Cut	CTRL + X	These functions follow Microsoft Windows conventions.
⧉	Copy	CTRL + C	
📋	Paste	CTRL + V	
✕	Delete	DEL	
↶	Undo	CTRL + Z	Click this icon to undo your last action.
↷	Redo	CTRL + SHIFT + Z	Click this icon to redo your last action.

Icon	Name	Keyboard Shortcut	Description
▶	Start Script	F5	Click this icon to execute the script. For details, refer to "Running a Script" on page 109.
▶	Start Selected	F8	Click this icon to execute the select portion of the script.
🟥	Stop Script	F6	Click this icon to stop the currently running script.
⟲	Reset local variables and loops		Click this icon to reset the local variable and loops in the script.
↩	Duplicate Row/Selection	CTRL + D	Click this icon to duplicate the code of the current cursor position or selected code to a new row. For details, refer to "Adding Duplicate Row/Selection in the Code" on page 111.
⬆	Restore default window layout		Click this icon to restore the default window layout.

Status Indicator

The **Status Indicator** indicates the current state of the script. It is located on the right side of the toolbar.



Running

It shows the following status:

- **Running** - When the script is being executed.
- **Not Running** - When the script is either not executed or the execution is completed.
- **Failed** - When the script execution fails.

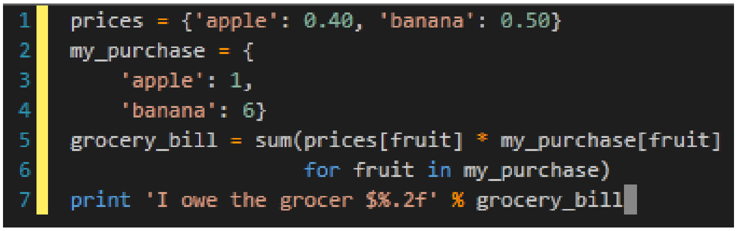
Editor Pane

The **Editor** pane allows you to:

- Write, edit, debug and run the scripts
- Open, save, import and export the scripts
- Debug the script code
- Run the entire script or selected lines from the script
- Apply settings to the editor using the **Settings** window

The **Editor** pane is an area where you enter code and execute it immediately. It also allows the execution of a single line of script code, with the resulting messages printed to the **Output** pane. For details, refer to "Output Pane" on page 104. In addition, the **Editor** pane has many features to help you enter, edit, check syntax, and debug script code. Beside this, it also allows you to use the Microsoft Window's standard functionally such as cut, copy, paste, undo and redo operations in the script code.

The following figure shows a simple program written in the **Editor** pane.

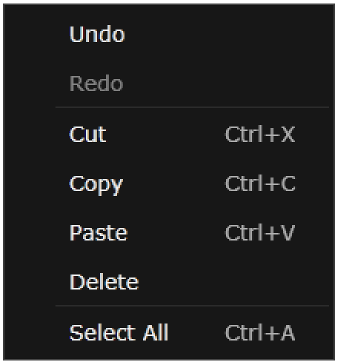


```
prices = {'apple': 0.40, 'banana': 0.50}
my_purchase = {
    'apple': 1,
    'banana': 6}
grocery_bill = sum(prices[fruit] * my_purchase[fruit]
                   for fruit in my_purchase)
print 'I owe the grocer $%.2f' % grocery_bill
```

Context Menu

The **Context** menu appears when you right-click on the editor pane. It provides the Window's standard functionally such as undo, redo, cut, copy, paste, delete and select all operations in the script code.

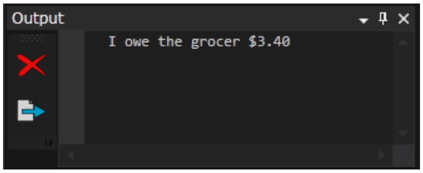
The following figure shows the options provided by the **Context** menu:



Output Pane

The **Output** pane displays the result of the executed script code. However, if the interpretor finds any error in the code, it is displayed in this pane.

The following figure shows an output of a program shown in an **Output** pane:



```
I owe the grocer $3.40
```

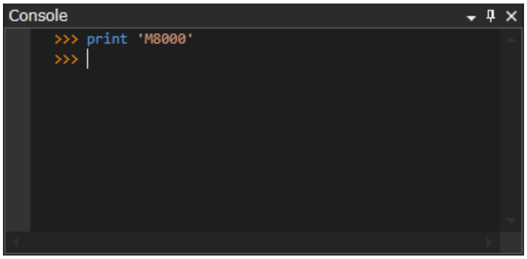
It provides the following options:

Clear Window - Click the **Clear Window** icon to clear the output pane.

Export - Click the **Export** icon to open a Window’s standard **Save As** dialog to save the content of the **Output** pane in a log file.

Console

The **Console** allows you to enter, interact with and visualize data, inside a command interpreter. All the commands entered in the **Console** are executed in a separate process, thus allowing you to interrupt any process at any time.



```
>>> print 'M8000'
>>> |
```

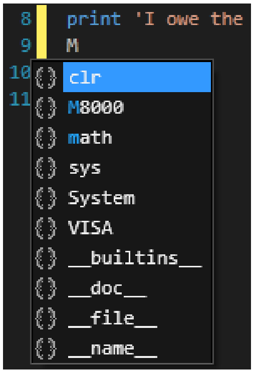
Settings Window

The **Settings** window provides settings for **Editor** pane, **Output** pane and **Console**.

The following table lists the settings for **Editor** pane:

Table 16

Settings	Description
Show Whitespace	Click this switch to show whitespace in the code.
Convert Tabs to Spaces	Click this switch to convert tabs to spaces in the code.
Tab Size	Use this setting to increase/decrease the tab size in the code.

Settings	Description
Word Wrapping	Click this switch to enable word wrapping in the code.
Zoom	Use this setting to zoom in or zoom out the code.
Code Completion	Click this switch to enable code completion feature. It saves the key strokes for caret movement when coding which can greatly reduce typing required for coding. The following figure shows how the code completion feature can reduce the typing while coding:  print 'I owe the M clr M8000 math sys System VISA __builtins__ __doc__ __file__ __name__

The following table lists the settings for **Output** pane:

Table 17

Settings	Description
Word Wrapping	Click this switch to enable word wrapping.
Zoom	Use this setting to zoom in or zoom out the output of the program executed.
Autoscroll Output	Click this switch to enable autoscroll feature.

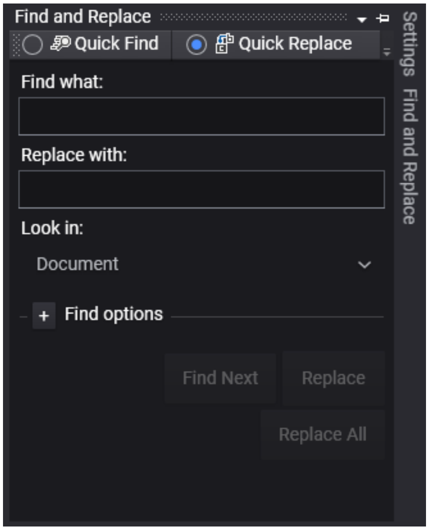
The following table lists the settings for **Console**:

Table 18

Settings	Description
Zoom	Use this setting to zoom in or zoom out.
Code Completion	Click this switch to enable code completion feature. It saves the key strokes for caret movement when coding which can greatly reduce typing required for coding.

Find and Replace

The **Find and Replace** dialog provides the search and replace capability in the script. The **Quick Find** option allows you to locate a specified text string while the **Quick Replace** option allows you to locate and replace a specified text string.



The **Find and Replace** dialog provides the following options:

- **Find What** - Displays the search string you want to locate.
- **Replace With** - Displays the string that will replace the string you searched for.

- **Look In** - Provides the search operation either in the entire document (script) or in the selection (selected part of the script).
- **Find Options** - Provides several other options (Match case, Match whole word, Search up and Search type) in the find dialog that you can use to customize your search.
- **Find Next** - Locates the next instance of the search string.
- **Replace** - Displays the string that will replace the string you searched for.
- **Replace All** - Replaces all instances where the search string is found.

Creating a Script

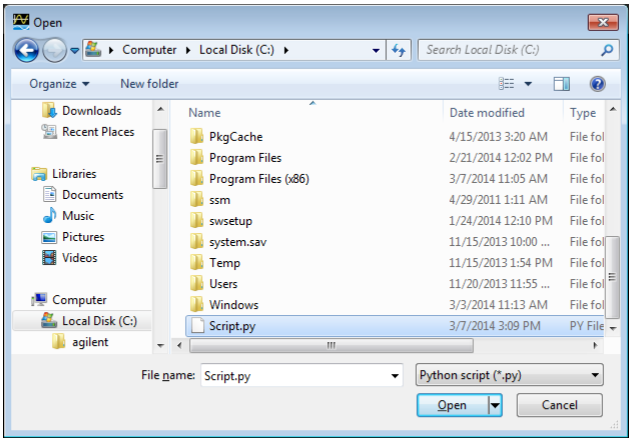
To create a script, click the **New** icon present on the toolbar. You can use **Editor** pane to write a script code.

Saving a Script

To save a current script, click the **Save** icon present on the toolbar.

Importing a Script

To import a script, click the **New** icon present on the toolbar. A Window's standard **Open** dialog will appear.



Name	Date modified	Type
PkgCache	4/15/2013 3:20 AM	File fol
Program Files	2/21/2014 12:02 PM	File fol
Program Files (x86)	3/7/2014 11:05 AM	File fol
ssm	4/29/2011 1:11 AM	File fol
swsetup	1/24/2014 12:10 PM	File fol
system.sav	11/15/2013 10:00 ...	File fol
Temp	11/15/2013 1:54 PM	File fol
Users	11/20/2013 11:55 ...	File fol
Windows	3/3/2014 11:13 AM	File fol
Script.py	3/7/2014 3:09 PM	PY File

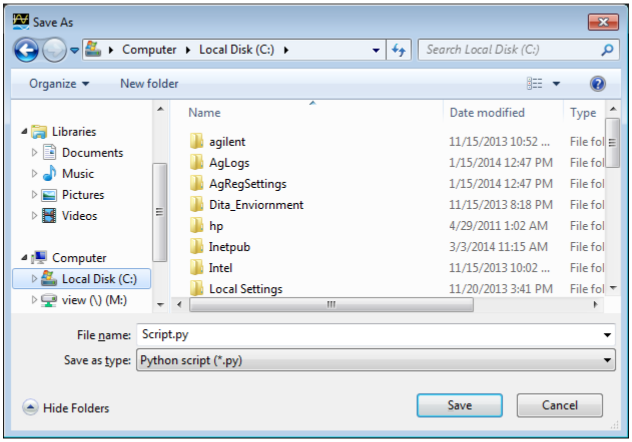
File name: Script.py

Python script (*.py)

- Locate the python script file (.PY) you want to import.
- Click **OK**.

Exporting a Script

To export a script, click the **New** icon present on the toolbar. The Window’s standard **Save As** dialog will appear.



Name	Date modified	Type
agilent	11/15/2013 10:52 ...	File fol
AgLogs	1/15/2014 12:47 PM	File fol
AgRegSettings	1/15/2014 12:47 PM	File fol
Dita_Enviornment	11/15/2013 8:18 PM	File fol
hp	4/29/2011 1:02 AM	File fol
Inetpub	3/3/2014 11:15 AM	File fol
Intel	11/15/2013 10:02 ...	File fol
Local Settings	11/20/2013 3:41 PM	File fol

File name: Script.py

Save as type: Python script (*.py)

- Select the location where you want to export the script.
- Enter a file name in the **File Name** text field.
- Click **Save**.

Running a Script

You can run either the complete script code or the selected part. To do so;

- Click the **Start Script** icon present on the toolbar to execute the entire script.
-
- Click the **Start Selected** icon present on the toolbar to execute the selected part of the script.

Verify the status of the executed scripts though **Status Indicator**. Refer to “Status Indicator” on page 102.

The output of the scripts will be displayed in the **Output** pane. However, if the scripts fails, the error will also be shown in the **Output** pane.

Stopping a Script

To stop the currently running script, click the ■ **Stop** icon present on the toolbar.

Please note that for few scripts, the **Stop** button sometimes do not respond while trying to abort script execution. This means that once the user presses the **Stop** button, the script execution stops, but the script editor gives an impression that the script is running because the progress indicator continues to run and **Stop** button is still clickable and **Start** button is disabled.

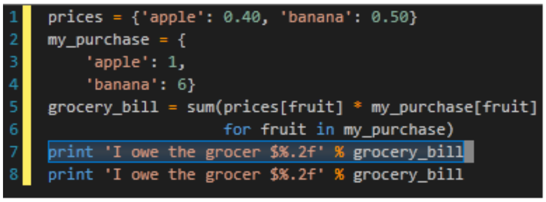
To handle such scripts, the following exception should be used:

```
    private void ExecuteScript()
    {
      try
      {
        m_Script.ExecuteCode(m_Code);
      }
      // Exceptions from in here will be ignored, otherwise
the program will close down involuntarily
      catch (Exception) { }
      finally
      {
        lock (m_AccessLocker)
        {
          m_Code = null;
        }
      }
    }
```

Adding Duplicate Row/Selection in the Code

To create a copy of the current row of the code or the selected part, click the **Duplicate Row/Selection** icon.

The following figure shows the row being duplicated using the **Duplicate Row/Selection** option.



```
prices = {'apple': 0.40, 'banana': 0.50}
my_purchase = {
    'apple': 1,
    'banana': 6}
grocery_bill = sum(prices[fruit] * my_purchase[fruit]
                   for fruit in my_purchase)
print 'I owe the grocer $%.2f' % grocery_bill
print 'I owe the grocer $%.2f' % grocery_bill
```

DUT Control Interface

The **DUT Control Interface** feature allows you to control a device under test at well defined positions within the measurements of the M8070B software. It allows you to:

- Read out error count information from devices with built-in error counters.
- Read out and write devices status registers
- Import and edit of Python scripts to access DUT from M8070B system software
- Integrate devices built-in error count information in error monitoring and jitter tolerance test routines of M8070B.

The **DUT Control Interface** editor provides inserting code snippets of the supported DUT. This allows you to tailor the supported measurements in a way that the DUT is controlled at critical points during the measurement and also provide error information for the measurement. Currently, the **DUT Control Interface** supports using a DUT's built-in error counters in the following measurements:

- Error Ratio Measurement
- Jitter Tolerance Measurement

All other measurements provide callback functions as well. These are useful to tailor the measurement to the DUT (e.g. settling times in dependency of measurement parameters), or to control the device under test during the measurement.

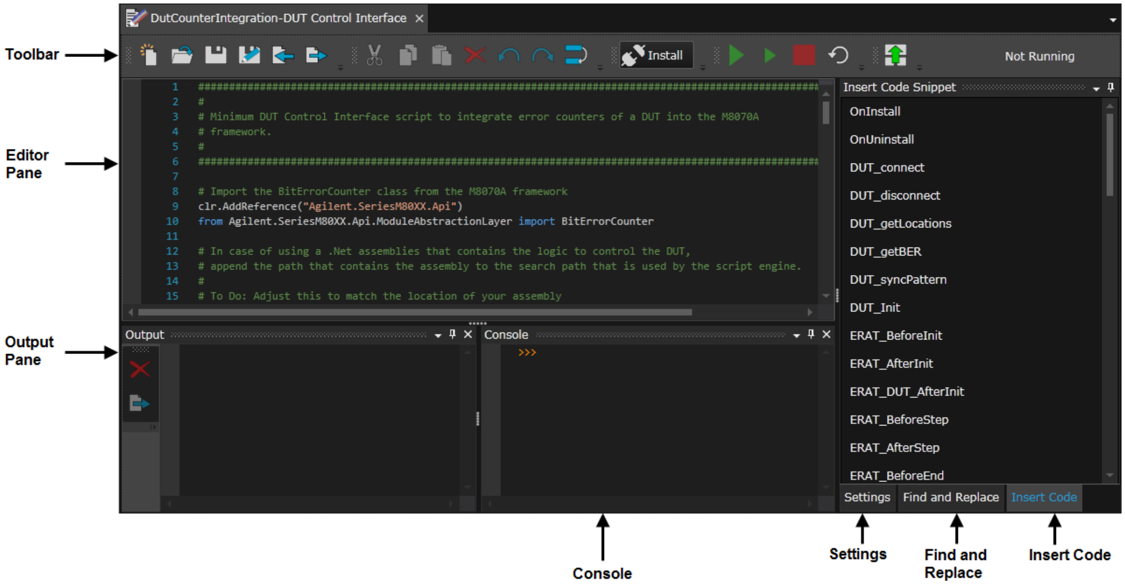
The **DUT Control Interface** feature is enabled by the license M8070ADVB-1xx.

Launching the DUT Control Interfacer

To launch the **DUT Control Interface**:

- Go to the **Menu Bar** > **Utilities** and then select **DUT Control Interface**.

The **DUT Control Interface** interface will appear as shown in the following figure:



```
##############################################################################################
#
# Minimum DUT Control Interface script to integrate error counters of a DUT into the M8070A
# framework.
#
##############################################################################################

# Import the BitErrorCounter class from the M8070A framework
clr.AddReference("Agilent.SeriesM80XX.Api")
from Agilent.SeriesM80XX.Api.ModuleAbstractionLayer import BitErrorCounter

# In case of using a .Net assemblies that contains the logic to control the DUT,
# append the path that contains the assembly to the search path that is used by the script engine.
#
# To Do: Adjust this to match the location of your assembly
```

```
>>>
```

The **DUT Control Interface** interface consists of the following GUI elements:

- Toolbar
- Editor Pane
- Output Pane
- Console
- Insert Code
- Settings Window
- Find and Replace Dialog

The listed GUI elements are described in the section that follows.

Toolbar

The toolbar provides the following convenient script editing functions:

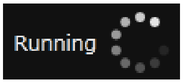
Table 19

Icon	Name	Description
🗋	New	Click this icon to open a new DUT script. For details, refer to “Creating a Script” on page 108
📂	Open	Click this icon to open an existing DUT script.
💾	Save	Click this icon to save the current DUT script. For details, refer to “Saving a Script” on page 108.
✎	Save As	Click this icon to save the current DUT script under different file name at either shared or current locations. For details, refer to “Saving a Script” on page 108.
⇐	Import	Click this icon to import external DUT script (.PY) in to the editor. For details, refer to “Importing a Script” on page 108.
⇒	Export	Click this icon to export the DUT script (.PY) to the local drive. For details, refer to “Exporting a Script” on page 109.
✂	Cut	These functions follow Microsoft Windows conventions.
⧉	Copy	
📋	Paste	
✕	Delete	
↶	Undo	Click this icon to undo your last action.

Icon	Name	Description
↷	Redo	Click this icon to redo your last action.
⮐	Duplicate Row/Selection	Click this icon to duplicate the code of the current cursor position or selected code to a new row. For details, refer to “Adding Duplicate Row/Selection in the Code” on page 111.
Install	Install	Click this button to install or uninstall the DUT control interface. Installing the DUT Control Interface script integrates the implemented functions into the M8070B software and makes the built-in error counters available for measurements and the remote interface. Editing, or interactive execution of the script is no longer possible when the script is installed.
▶	Start Script	Click this icon to execute the script. For details, refer to “Running a Script” on page 109. Running the script in the editor is useful to debug the script functionality. The individual functions can be executed from the Console after the script has been executed.
▶	Start Selected	Click this icon to execute the select portion of the script. This is useful to update the functionality available on the console after making changes to parts of the script, without executing all of the script. This avoids other side effects like resetting global variables that can be unwanted during testing the script.
■	Stop Script	Click this icon to stop the currently running script.
⟲	Reset local variables and loops	Click this icon to reset the local variable and loops in the script.
⬆	Restore default window layout	Click this icon to restore the default window layout.

Status Indicator

The **Status Indicator** indicates the current state of the script. It is located on the right side of the toolbar.



Running

It shows the following status:

- **Running** - When the script is being executed.
- **Not Running** - When the script is either not executed or the execution is completed.
- **Failed** - When the script execution fails.

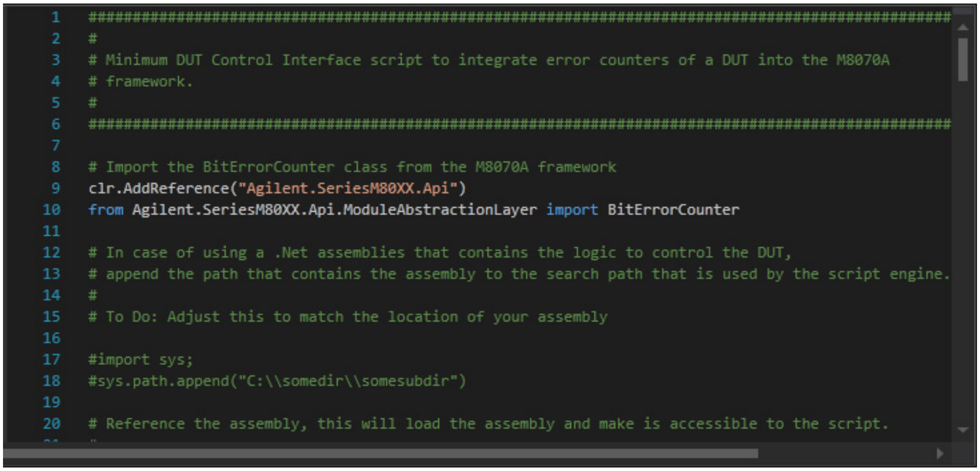
Editor Pane

The **Editor** pane allows you to:

- Write, edit and run the scripts
- Open, save, import and export the scripts
- Run the entire script or selected lines from the script
- Apply settings to the editor using the **Settings** window

The **Editor** pane is an area where you enter code and execute it immediately. It also allows the execution of a single line of script code, with the resulting messages printed to the **Output** pane. For details, refer to "Output Pane" on page 104. In addition, the **Editor** pane has many features to help you enter, edit and check syntax of the code. Beside this, it also allows you to use the Microsoft Window's standard functionally such as cut, copy, paste, undo and redo operations in the script code.

The following figure shows a simple program written in the **Editor** pane.

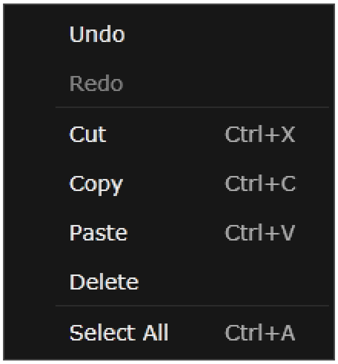


```
##################################################################################################
#
# Minimum DUT Control Interface script to integrate error counters of a DUT into the M8070A
# framework.
#
##################################################################################################

# Import the BitErrorCounter class from the M8070A framework
clr.AddReference("Agilent.SeriesM80XX.Api")
from Agilent.SeriesM80XX.Api.ModuleAbstractionLayer import BitErrorCounter

# In case of using a .Net assemblies that contains the logic to control the DUT,
# append the path that contains the assembly to the search path that is used by the script engine.
#
# To Do: Adjust this to match the location of your assembly

#import sys;
#sys.path.append("C:\\somedir\\somesubdir")

# Reference the assembly, this will load the assembly and make is accessible to the script.
```

Context Menu

The **Context** menu appears when you right-click on the editor pane. It provides the Window's standard functionally such as undo, redo, cut, copy, paste, delete and select all operations in the script code.

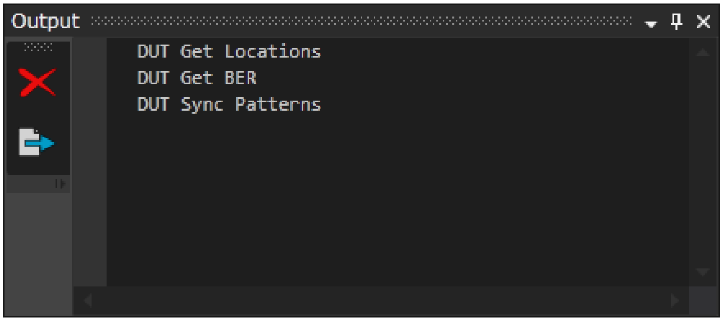
The following figure shows the options provided by the **Context** menu:



Output Pane

The **Output** pane displays the result of the executed script code. However, if the interpretor finds any error in the code, it is displayed in this pane.

The following figure shows an output of a program shown in an **Output** pane:



```
DUT Get Locations
DUT Get BER
DUT Sync Patterns
```

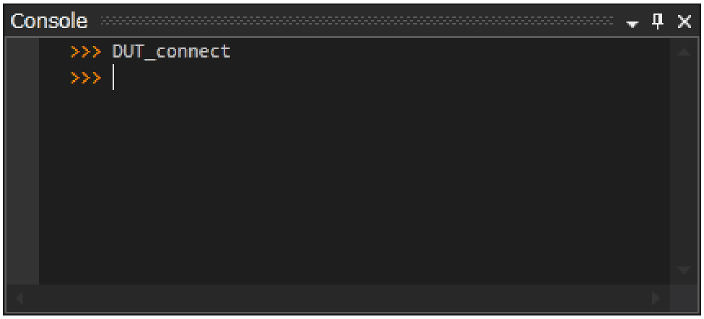
It provides the following options:

Clear Window - Click the ✕ **Clear Window** icon to clear the output pane.

Export - Click the ➡ **Export** icon to open a Window's standard **Save As** dialog to save the content of the **Output** pane in a log file.

Console

The **Console** allows you to enter, interact with and visualize data, inside a command interpreter. All the commands entered in the **Console** are executed in a separate process, thus allowing you to interrupt any process at any time.



```
>>> DUT_connect
>>> |
```

Insert Code Window

The **Insert Code** window lists code snippets that are supported by the DUT Control Interface. You can double-click on the code snippet to insert them in the **Editor** pane. The code snippets ensure correct signature of the supported functions and contain documentation of the expected functionality and the function's arguments.

Please note that you are not allowed to modify the script when the DUT control script is installed.

Also, the **Output** pane and **Console** will not be available when DUT control script is installed.

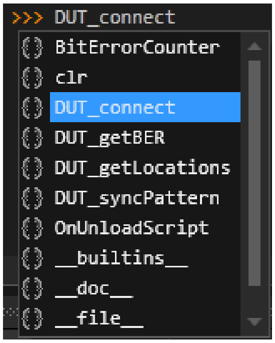
Settings Window

The **Settings** window provides settings for the **Editor** pane, **Output** pane and **Console**.

The following table lists the settings for the **Editor** pane:

Table 20

Settings	Description
Show Whitespace	Click this switch to show whitespace in the code.
Convert Tabs to Spaces	Click this switch to convert tabs to spaces in the code.
Tab Size	Use this setting to increase/decrease the tab size in the code.

Settings	Description
Word Wrapping	Click this switch to enable word wrapping in the code.
Zoom	Use this setting to zoom in or zoom out the code.
Code Completion	Click this switch to enable code completion feature. It saves the key strokes for caret movement when coding which can greatly reduce typing required for coding. The following figure shows how the code completion feature can reduce the typing while coding:  >>> DUT_connect BitErrorCounter clr DUT_connect DUT_getBER DUT_getLocations DUT_syncPattern OnUnloadScript __builtins__ __doc__ __file__

The following table lists the settings for the **Output** pane:

Table 21

Settings	Description
Word Wrapping	Click this switch to enable word wrapping.
Zoom	Use this setting to zoom in or zoom out the output of the program executed.
Autoscroll Output	Click this switch to enable autoscroll feature.

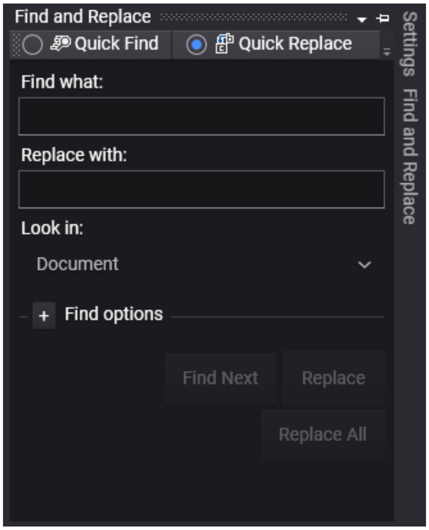
The following table lists the settings for **Console**:

Table 22

Settings	Description
Zoom	Use this setting to zoom in or zoom out.
Code Completion	Click this switch to enable code completion feature. It saves the key strokes for the caret movement when coding which can greatly reduce typing required for coding.

Find and Replace

The **Find and Replace** dialog provides the search and replace capability in the script. The **Quick Find** option allows you to locate a specified text string while the **Quick Replace** option allows you to locate and replace a specified text string.



○ Quick Find ● Quick Replace

Look in: Document

The **Find and Replace** dialog provides the following options:

- **Find What** – Displays the search string you want to locate.
- **Replace With** – Displays the string that will replace the string you searched for.

- **Look In** - Provides the search operation either in the entire document (script) or in the selection (selected part of the script).
- **Find Options** - Provides several other options (Match case, Match whole word, Search up and Search type) in the find dialog that you can use to customize your search.
- **Find Next** - Locates the next instance of the search string.
- **Replace** - Displays the string that will replace the string you searched for.
- **Replace All** - Replaces all instances where the search string is found.

Creating a New Script

To create a new script, click the **New** icon present on the toolbar. A default script will be loaded in the **Editor** pane which you can edit to create a new script.

Open a Script

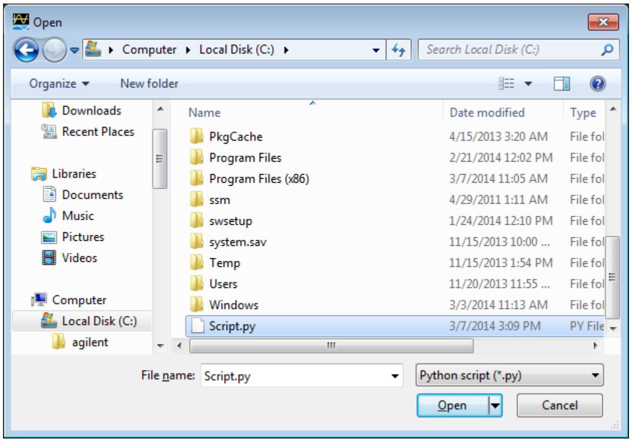
To open a saved script, click the **Open** icon present on the toolbar. A **Open Hooks Script** dialog will appear which allows you to open the DUT scripts.

Saving a Script

To save the current script, click the **Save** icon present on the toolbar. A **Save Hooks Script** dialog will appear which allows you to save the scripts on either shared or current folder.

Importing a Script

To import a script, click the **Import** icon present on the toolbar. A Window’s standard **Open** dialog will appear.



Name	Date modified	Type
PkgCache	4/15/2013 3:20 AM	File fol
Program Files	2/21/2014 12:02 PM	File fol
Program Files (x86)	3/7/2014 11:05 AM	File fol
ssm	4/29/2011 1:11 AM	File fol
swsetup	1/24/2014 12:10 PM	File fol
system.sav	11/15/2013 10:00 ...	File fol
Temp	11/15/2013 1:54 PM	File fol
Users	11/20/2013 11:55 ...	File fol
Windows	3/3/2014 11:13 AM	File fol
Script.py	3/7/2014 3:09 PM	PY File

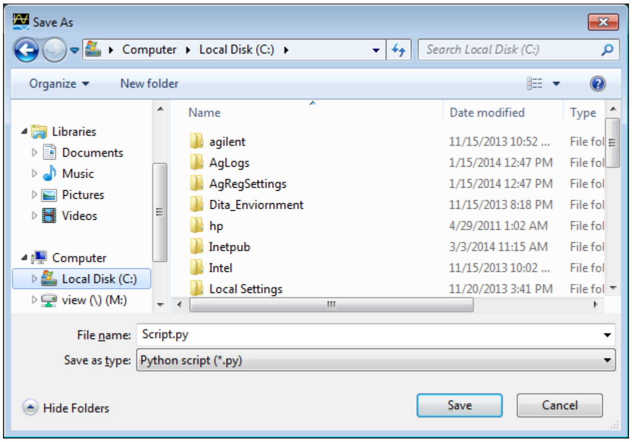
File name: Script.py

Python script (*.py)

- Locate the Python script file (.PY) you want to import.
- Click **OK**.

Exporting a Script

To export a script, click the **Export** icon present on the toolbar. The Window’s standard **Save As** dialog will appear.



Name	Date modified	Type
agilent	11/15/2013 10:52 ...	File fol
AgLogs	1/15/2014 12:47 PM	File fol
AgRegSettings	1/15/2014 12:47 PM	File fol
Dita_Enviornment	11/15/2013 8:18 PM	File fol
hp	4/29/2011 1:02 AM	File fol
Inetpub	3/3/2014 11:15 AM	File fol
Intel	11/15/2013 10:02 ...	File fol
Local Settings	11/20/2013 3:41 PM	File fol

File name: Script.py

Save as type: Python script (*.py)

- Select the location where you want to export the script.
- Enter a file name in the **File Name** text field.
- Click **Save**.

Running a Script

You can run either the complete script code or the selected part. To do so;

- Click the ▶ **Start Script** icon present on the toolbar to execute the entire script.

- Click the ▶ **Start Selected** icon present on the toolbar to execute the selected part of the script.

Verify the status of the executed scripts though **Status Indicator**. Refer to "Status Indicator" on page 102.

The output of the scripts will be displayed in the **Output** pane. However, if the scripts fails, the error will also be shown in the **Output** pane.

Stopping a Script

To stop the currently running script, click the ■ **Stop** icon present on the toolbar.

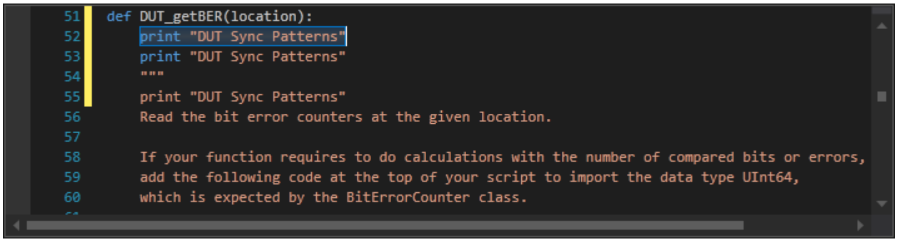
Stopping the script does not always work. If the script is using non-Python code (e.g. calling functions in .Net libraries), then the execution of these functions cannot be stopped in all cases.

Adding Duplicate Row/Selection in the Code

To create a copy of the current row of the code or the

selected part, click the **Duplicate Row/Selection** icon.

The following figure shows the row being duplicated using the **Duplicate Row/Selection** option.



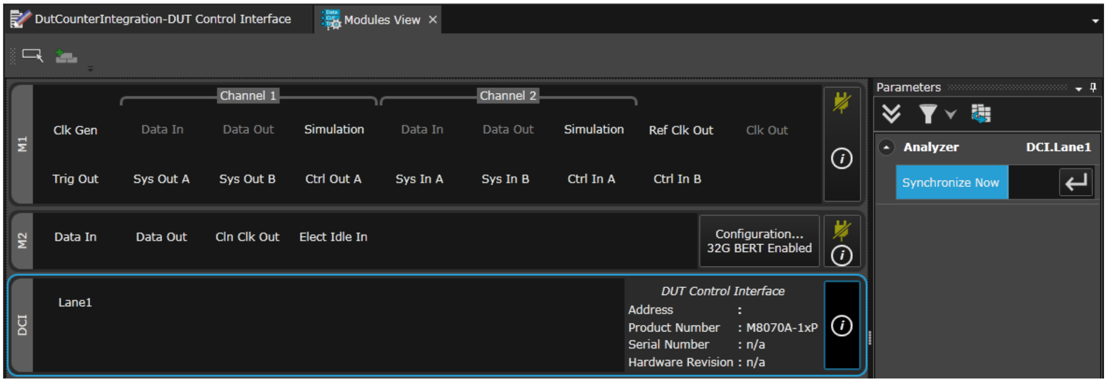
```
def DUT_getBER(location):
    print "DUT Sync Patterns"
    print "DUT Sync Patterns"
    """
    print "DUT Sync Patterns"
    Read the bit error counters at the given location.

    If your function requires to do calculations with the number of compared bits or errors,
    add the following code at the top of your script to import the data type UInt64,
    which is expected by the BitErrorCounter class.
```

Effects on M8070B GUI on Installing a DUT Control Script

On successful installation of DUT control scripts, the following changes are reflected in M8070B GUI.

Module View

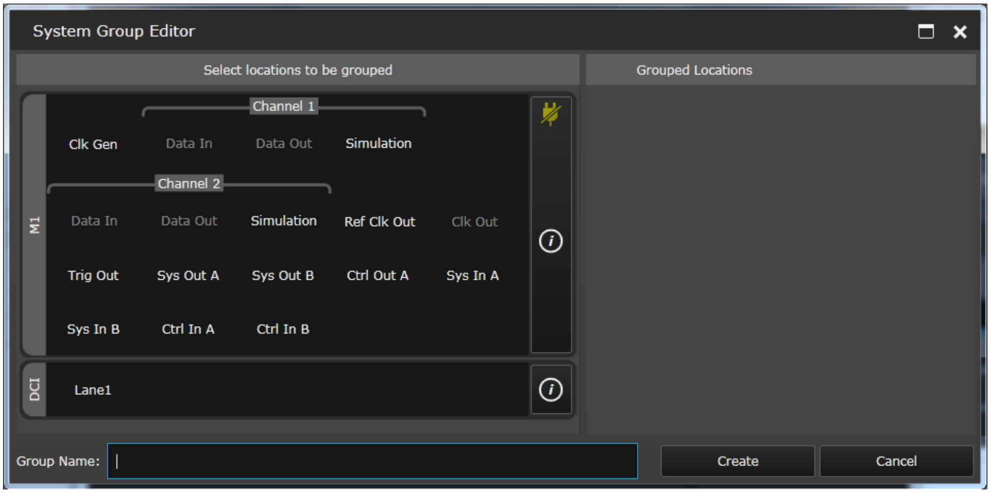
A new DUT control along with the supported parameters is now added in the **Module View**:



DUT Control Interface
Address :
Product Number : M8070A-1xP
Serial Number : n/a
Hardware Revision : n/a

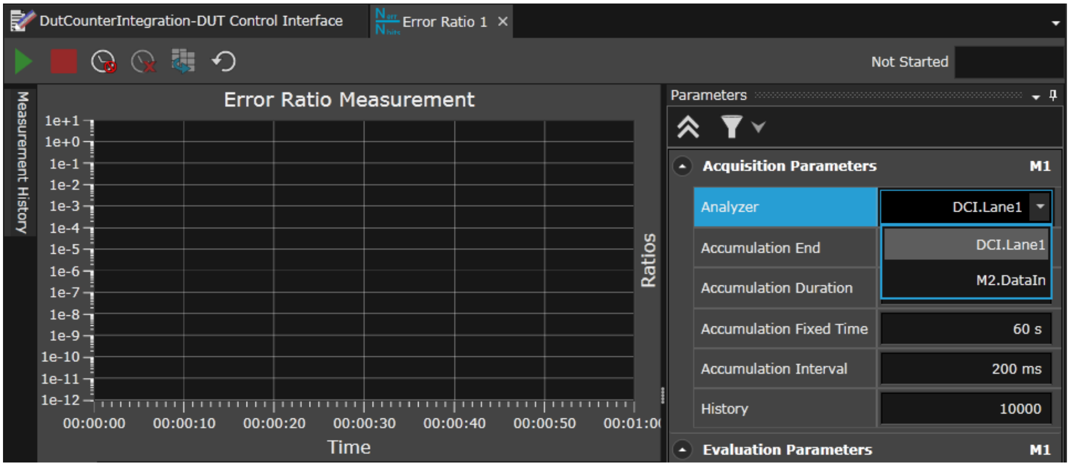
Group View

New locations will be now available to create a group in the **Group View**.



Error Ratio Measurement

A new location will be now available in the **Analyzer** parameter of the **Error Ratio Measurement**.



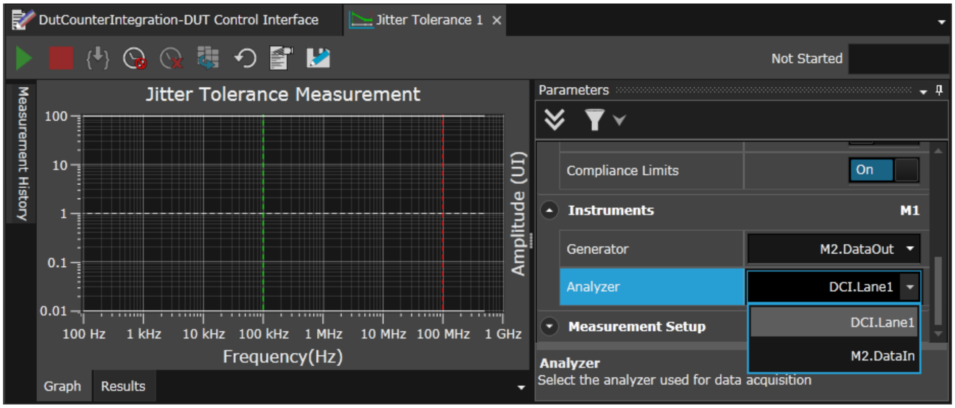
Acquisition Parameters M1

Analyzer	DCI.Lane1
Accumulation End	DCI.Lane1 M2.DataIn
Accumulation Duration	
Accumulation Fixed Time	60 s
Accumulation Interval	200 ms
History	10000

Evaluation Parameters M1

Jitter Tolerance Measurement

A new location will be now available in the **Analyzer** parameter of the **Jitter Tolerance Measurement**.



Compliance Limits	On

Instruments M1

Generator	M2.DataOut
Analyzer	DCI.Lane1 DCI.Lane1 M2.DataIn

Measurement Setup

Analyzer
Select the analyzer used for data acquisition

Writing a Script Code for DUT Control Interface

Scripting Runtime

- The underlying scripting technology uses IronPython 2.7.5
- The M8070B does not include the full IronPython 2.7.5 installation, only the scripting engine itself. When referencing the modules of an IronPython installation that is available on the computer, it must be ensured that the versions match.
- IronPython makes it very easy to use functionality provided by the Microsoft .NET framework, but it cannot directly access dynamic link libraries that are not based on .NET.
  Functions that are implemented in unmanaged DLLs (usually implemented in C or C++) do require a .NET wrapper library.
  For details see:
  https://msdn.microsoft.com/en-us/library/ms235282.aspx
  Title: Calling Native Functions from Managed Code
- Python functions do not have out parameters. If a function is using out parameters, then these will be returned as additional return values.

  Example:

  - **C# function**

  ```
  bool DoSomething(int arg1, out double resultArg1,
    double arg2, out int resultArg2,
    out string resultArg3)
  ```

  - **IronPython usage**

  ```
  (returnValue, resultArg1, resultArg2, resultArg3) =
  DoSomething(arg1, arg2)
  ```

- Remote commands to control M8070B cannot be used within the DUT Control Interface scripts.
  While this is possible in principle, it will cause problems when the functions implemented in the DUT Control Interface are being executed in the context of remote commands (e.g. query error counters via :fetc:bco?). In this case the remote interface is busy until the DUT counters have been returned by the script function DUT_getBER, but while the remote interface is busy it cannot process any further remote commands, neither internally issued nor external ones. This can very easily end up in deadlock/timeout scenarios.
- Code outside of function definitions is being executed at load time of the script.
  This can be used to initialize required variables, or establish the communication link with the DUT.

Minimum Requirements to Integrate a DUT's Error Counters

- Implement DUT_getLocations() to return a unique name for each location within the DUT that does provide an error counter (e.g. Lane1, Lane2, Lane3, Lane4)

  Example:

  The following example defines DUT_getLocations to define 4 locations named Lane to Lane 4:

  ```
  def DUT_getLocations():
    return ["Lane1", "Lane2", "Lane3", "Lane4" ]
  ```

- Implement DUT_getBER(location) to return the error counter for each of the defined locations.

  Example:

  The following example reads BER counter from one of 4 Lanes. Each Lane is assumed to be controlled by an instance of a driver class that implements the IBerReader interface (see code snippet below). The instances of the 4 lanes are held in a dictionary that allows easy lookup by the given location name.

  ```
  # create empty dictionary
  myDUTs = {}
  # instantiate the driver for each location
  myDUTs["Lane1"] = MyDutBerReader(1)
  myDUTs["Lane2"] = MyDutBerReader(2)
  myDUTs["Lane3"] = MyDutBerReader(3)
  myDUTs["Lane4"] = MyDutBerReader(4)

  def DUT_getBER(location):
    global myDUTs
    (bits, errors) = myDUTs[location].GetCounter()
    return BitErrorCounter(bits, errors)
  ```

- Implement DUT_synchPattern(location) to synchronize the error counter to the received data stream.

  Optionally reset the DUT, but be aware that this should not have side effects on other locations.

  The M8070B framework expects the DUT's error counters to be running all the time. If the synchronization occurs on a stopped to running transition, then the sop as well as the start need to be part of DUT_syncPattern(location).

```
def DUT_syncPattern(location):
  global myDUTs
  lane = myDUTs[location]
  lane.Stop()
  lane.ResetDut()
  lane.Start()
```

- Optionally implement DUT_Init(location, initArg) if it is required to configure the DUT to different modes.

  DUT_Init can be called from the remote programming interface, so the test automation is able to execute initialization code on the DUT programmatically.

  The M8070B software will not call DUT_Init unless the script is using it internally, or it is either requested from the remote programming interface, or interactively from within the DUT Control Interface script editor.

NOTE

- Making script code executable from within the remote programming interface can cause a security issue in your computer.
- The M8070B software is not able to limit the user script's functionality. It is the user‘s responsibility to ensure that no unsafe or otherwise critical code is being executed from within DUT_Init.
- Special care should be taken when using the python statements eval or exec, as these allow dynamic execution of python code.
- Keysight recommends not using eval or exec at all, except the DUT or application requires this.
- Keysight is not liable for any kind of damage caused by the use of remote executable script code (see *Limitation of Liability in End User License Agreement*).

Useful Sources of Information

- http://ironpython.net/

  Home of the IronPython project.

  Especially the IronPython .Net Integration documentation is a valuable source of information regarding access to the .Net framework (http://ironpython.net/documentation/dotnet/)
- https://www.python.org/

  Home of the python programming language. This is the preferred souorce of information regarding to the language syntax itself.

  When referring the language documentation make sure to use the Python 2.7 documentation.

  Keep in mind that Python is not the same as IronPython, so Python libraries or modules are not necessarily working with IronPython.
- https://msdn.microsoft.com/en-us/library/gg145045(v=vs.100).aspx

  Title: NET Framework Class Library

  Programming reference of the .NET framework that is accessible to IronPython scripts.

Useful Code Snippets

Importing core .NET functionality

The base .NET functionality is implemented in mscorlib.dll

Making a .NET data type available to the script requires loading the assembly and importing the data type.

Example: Importing and using System.UInt64

```
clr.AddReference("mscorlib")
from System import UInt64
comparedBits = UInt64(0)
comparedBits = comparedBits + UInt64(1e6)
```

Using .NET Generic Types

It is often required to use generic types when calling .NET code.

The C# code to instantiate a List of Strings looks like this:

```
using System;
using System.Collections.Generic;
var stringList = new List<string>()
```

The corresponding IronPython code:

```
clr.AddReference("mscorlib")
from System import String
from System.Collections.Generic import List
stringList = List[String]()
```

Using User Defined .NET Assemblies (e.g. Implementations of IBerReader)

In order to reference user defined assemblies, the search path of the scripting engine needs to be extended. After this is done the assembly reference can added and the data type imported.

```
# Add c:\MyLibraries\MyDUT to the search path import sys
sys.path.append("C:\\MyLibraries\\MyDUT")

# reference MyDutCustomBerReader.dll located in
c:\MyLibraries\MyDUT
clr.AddReferenceToFile("MyDutCustomBerReader")

# from the namespace MyNamespace.MyDutCustomBerReader
# that is defined in the previously referenced assembly,
# import the actual data type.
from MyNamespace.MyDutCustomBerReader import
MyDutBerReader

# create an instance of MyDutBerReader
myDUT = MyDutBerReader()

# call a function defined in the data type MyDutBerReader
myDUT.DoSomething()
```

Using the COM Port to Access the DUT

.NET does provide a dedicated class to work with the computer's COM ports. This class is implemented in the System assembly.

Importing the SerialPort class

```
# import SerialPort from System assembly
clr.AddReference("System")
import System.IO.Ports.SerialPort
```

```
# additional reference assembly mscorlib to have all data
types
# available that are used by SerialPort
clr.AddReference("mscorlib")
```

Open the COM Port

```
# create an instance of SerialPort for COM4
serialPort = System.IO.Ports.SerialPort("COM4")

# configure COM4 to 9600 bit/s 8N1
serialPort.BaudRate = int (9600)
serialPort.DataBits = int(8)
# available choices for Parity are
# None: System.IO.Ports.Parity.None
# Odd   : System.IO.Ports.Parity.Odd
# Even  : System.IO.Ports.Parity.Even
# Mark  : System.IO.Ports.Parity.Mark
# Space : System.IO.Ports.Parity.Space
serialPort.Parity = System.IO.Ports.Parity.None
# available choices for StopBits are
# None : System.IO.Ports.StopBits.None
# One  : System.IO.Ports.StopBits.One
# Two  : System.IO.Ports.StopBits.Two
# OnePointFive : System.IO.Ports.StopBits.OnePointFive
serialPort.StopBits = System.IO.Ports.StopBits.One
# set read and write timeouts to 500ms
serialPort.ReadTimeout = int(500)
serialPort.WriteTimeout = int(500)

# finally open the COM port
serialPort.Open()
```

Reading and Writing Data

```
# write a string
serialPort.Write("Sending string data")

# sending binary data

# prepare the data to be sent. SerialPort expects an Array
of Byte. This requires importing mscorlib!
```

```
buf = System.Array[System.Byte] ([0x5A, 0x02, 0x00,
0xaf])

# send 4 bytes beginning at offset 0
serialPort.Write(buf, 0, 4)

# reading 100 bytes
numBytes = 100
buf = System.Array.CreateInstance(System.Byte, numBytes)
serialPort.Read(buf, 0, numBytes)
```

Closing the COM Port

The following code snippet shows the required code to close the COM port and discard the SerialPort instance. This code should be called in the OnUninstall function of the DUT Control Interface in order to ensure that the COM port becomes unavailable after the currently active script becomes removed from the computer's memory.

```
if (serialPort != None):
  # check if COM port is open
  if (serialPort.IsOpen):
      serialPort.Close()
  # dispose the serialPort instance
  serialPort.Dispose()
  # ensure that the disposed object is not used in future
  serialPort = None
```

Common Pitfalls when Working with COM Ports

A COM port cannot be opened more than once. Therefore it is very important to close the port before the script is unloaded.

This does have several implications:

- If the COM port is not available when the script is being loaded (e.g. when recalling a setting), then the script will fail opening the COM port and the DUT will not be accessible.

- When working with additional tools to communicate with the DUT, e.g. a debug-tool to configure the device, only one process can have control over the COM port at a time. If the M8070B software has opened the COM port in the DUT Control Interface script, then the external tool cannot control the device and vice versa.
- Exceptions in the program flow may not prevent releasing the COM port.

Best Scripting Practices

- Always implement OnUninstall and close the COM port properly.
- Create and open the COM port as late as possible in the script.
  - Import all data types
  - Implement all driver functionality to communicate with the device.
  - Implement the required hook functions and call the device specific functions.
  - Configure and open the COM port.

    The preferred place to do this is in the DUT_connect and DUT_disconnect hooks.
  - Initialize the DUT using the successfully opened COM port.

    If there can be errors during the initialization phase, ensure that the variable that provides access to the COM port is only getting invalid after the COM port has been closed.

Communication with the M8070B Instrument Layer

Overview

The DUT Control Interface provides a functional API to allow interaction with the instrument layer of the M8070B software.

The functional API is provided via the global symbol M8000 within the DUT Control Interface script.

This is similar to the M8000.Scpi API in the Script Editor, but does not allow sending remote commands to the M8070B software. Instead it provides direct access to measurement relevant status and control of the sequencing directly.

This can be used to implement measurement hooks that take the status information of the instrument into consideration and initiate sequence branching in dependency of the reported status.

The API to control the instrument may not be used from within the DUT specific hooks. Doing so will cause error messages when executing the respective hooks at runtime.

If an API call is resulting in an error (e.g. when reading status bits that don't exist for a DUT) then an exception is thrown, which finally results in an error message on the level of the DUT Control Interface script.

Examples

An example of how this can be used to re-train a DUT back into loopback during a Jitter Tolerance measurement can be found in the Tutorials folder of the Factory provided DUT Control Interface scripts.

The script is named Retrain_DUT_in_JTOL.py and demonstrates how the Instrument API can be used to control the sequence in dependency of the reported status information and bit counter values.

API Summary

The following is giving a quick overview over Instrument control API.

Sequencing Control

M8000.Instrument.Sequence.Restart(groupOrLocationName)

Restart the sequence on the given group or location.

M8000.Instrument.Sequence.BreakAll()

Execute a sequence break, which will influence all sequencers in the system that are capable of being controlled by the break command.

M8000.Instrument.Sequence.GetBlockNumber(groupOrLocationName)

Reads the current sequence block number from the addressed locations.

Analyzer Control

M8000.Instrument.Analyzer.AutoAlign(groupOrLocationName)

Execute an auto-alignment on the addressed locations.

M8000.Instrument.Analyzer.DataCenter(groupOrLocationName)

Executes a data-center alignment on the addressed locations.

M8000.Instrument.Analyzer.ThresholdCenter(groupOrLocationName)

Executes a threshold-center alignment on the addressed locations.

M8000.Instrument.Analyzer.SyncNow(groupOrLocationName)

Executes a pattern synchronization on the addressed locations.

M8000.Instrument.Analyzer.GetBitCounters(groupOrLocationName, combineIntoOneResult)

Read the bit counters from the addressed locations and return either the sum of all addressed counters, or each addressed counter individually.

M8000.Instrument.Analyzer.GetDataLoss(groupOrLocationName, combineIntoOneResult)

Read the data loss status bit from the addressed locations and return either the logical OR of all addressed locations, or each location's status individually.

M8000.Instrument.Analyzer.GetSyncLoss(groupOrLocationName,combineIntoOneResult)

Read the sync loss status bit from the addressed locations and return either the logical OR of all addressed locations, or each location's status individually.

M8000.Instrument.Analyzer.GetSymbolLoss(groupOrLocationName,combineIntoOneResult)

Read the symbol loss status bit from the addressed locations and return either the logical OR of all addressed locations, or each location's status individually.

M8000.Instrument.Analyzer.GetCdrUnlock(groupOrLocationName, combineIntoOneResult)

Read the CDR unlock status bit from the addressed locations and return either the logical OR of all addressed locations, or each location's status individually.

API Details

Sequencing Control

M8000.Instrument.Sequence.Restart(groupOrLocationName)

Functionality	Restart the sequence on the given group or location.
Parameters	
groupOrLocationName	Addresses the locations that shall be restarted. The use of .Net regular expression patterns is allowed.
Returns	Nothing
Notes	▪ Initiates the restart, but does not wait until all sequencers have actually executed the request. ▪ Not allowed for "DUT_" hooks. ▪ Not supported for locations defined by the DUT Control Interface script.

M8000.Instrument.Sequence.BreakAll()

Functionality	Execute a sequence break, which will influence all sequencers in the system that are capable of being controlled by the break command.
Parameters	None
Returns	Nothing
Notes	▪ Initiates the break, but does not wait until the sequencers have executed the break. ▪ Not allowed for "DUT_" hooks.

M8000.Instrument.Sequence.GetBlockNumber(groupOrLocationName)

Functionality	Reads the current sequence block number from the addressed locations. The use of .Net regular expression patterns is allowed.
Parameters	
groupOrLocationName	Addresses the locations that shall be read. The use of .Net regular expression patterns is allowed.
Returns	IReadOnlyDictionary<string, int> The dictionary contains one entry for each addressed location.
Notes	▪ Not allowed for "DUT_" hooks. ▪ Not supported for locations defined by the DUT Control Interface script.

Analyzer Control

M8000.Instrument.Analyzer.AutoAlign(groupOrLocationName)

Functionality	Execute an auto-alignment on the addressed locations.
Parameters	
groupOrLocationName	Addresses the locations that shall be aligned. The use of .Net regular expression patterns is allowed.
Returns	Nothing
Notes	▪ Is blocking until the alignment has finished. ▪ Not allowed for "DUT_" hooks. ▪ Not supported for locations defined by the DUT Control Interface script.

M8000.Instrument.Analyzer.DataCenter(groupOrLocationName)

Functionality	Executes a data-center alignment on the addressed locations.
Parameters	
groupOrLocationName	Addresses the locations that shall be aligned. The use of .Net regular expression patterns is allowed.
Returns	Nothing
Notes	▪ Is blocking until the alignment has finished. ▪ Not allowed for "DUT_" hooks. ▪ Not supported for locations defined by the DUT Control Interface script.

M8000.Instrument.Analyzer.ThresholdCenter(groupOrLocationName)

Functionality	Executes a threshold-center alignment on the addressed locations.
Parameters	
groupOrLocationName	Addresses the locations that shall be aligned. The use of .Net regular expression patterns is allowed.
Returns	Nothing
Notes	▪ Is blocking until the alignment has finished. ▪ Not allowed for "DUT_" hooks. ▪ Not supported for locations defined by the DUT Control Interface script.

M8000.Instrument.Analyzer.SyncNow(groupOrLocationName)

Functionality	Executes a pattern synchronization on the addressed locations.
Parameters	
groupOrLocationName	Addresses the locations that shall be aligned. The use of .Net regular expression patterns is allowed.
Returns	Nothing
Notes	▪ Is blocking until the synchronization has finished. ▪ Not allowed for "DUT_" hooks. ▪ Supported for locations defined by the DUT Control Interface script. But requires implementation of DUT_syncPattern.

M8000.Instrument.Analyzer.GetBitCounters(groupOrLocationName, combineIntoOneResult)

Functionality	Read the bit counters from the addressed locations and return either the sum of all addressed counters, or each addressed counter individually.
Parameters	
groupOrLocationName	Addresses the locations that shall be read. The use of .Net regular expression patterns is allowed.
combineIntoOneResult	▪ True Return the sum of all addressed bit counters as a single instance of type BitErrorCounter. ▪ False Return a dictionary of type IReadOnlyDictionary<string, BitErrorCounter> that holds an entry for each addressed location.
Returns	BitErrorCounter or IReadOnlyDictionary<string, BitErrorCounter>
Notes	▪ Not allowed for "DUT_" hooks.

M8000.Instrument.Analyzer.GetDataLoss(groupOrLocationName, combineIntoOneResult)

Functionality	Read the data loss status bit from the addressed locations and return either the logical OR of all addressed locations, or each location's status individually.
Parameters	
groupOrLocationName	Addresses the locations that shall be read. The use of .Net regular expression patterns is allowed.
combineIntoOneResult	▪ True Return the logical OR of all addressed status bits. ▪ False Return a dictionary of type IReadOnlyDictionary<string, bool> that holds an entry for each addressed location.
Returns	bool or IReadOnlyDictionary<string, bool>
Notes	▪ Not allowed for "DUT_" hooks. ▪ Not supported for locations defined by the DUT Control Interface script.

M8000.Instrument.Analyzer.GetSyncLoss(groupOrLocationName,combineIntoOneResult)

Functionality	Read the sync loss status bit from the addressed locations and return either the logical OR of all addressed locations, or each location's status individually.
Parameters	
groupOrLocationName	Addresses the locations that shall be read. The use of .Net regular expression patterns is allowed.

combineIntoOneResult	▪ True Return the logical OR of all addressed status bits. ▪ False Return a dictionary of type IReadOnlyDictionary<string, bool> that holds an entry for each addressed location.
Returns	bool or IReadOnlyDictionary<string, bool>
Notes	▪ Not allowed for "DUT_" hooks. ▪ Not supported for locations defined by the DUT Control Interface script.

M8000.Instrument.Analyzer.GetSymbolLoss(groupOrLocationName,combineIntoOneResult)

Functionality	Read the symbol loss status bit from the addressed locations and return either the logical OR of all addressed locations, or each location's status individually.
Parameters	
groupOrLocationName	Addresses the locations that shall be read. The use of .Net regular expression patterns is allowed.
combineIntoOneResult	▪ True Return the logical OR of all addressed status bits. ▪ False Return a dictionary of type IReadOnlyDictionary<string, bool> that holds an entry for each addressed location.
Returns	bool or IReadOnlyDictionary<string, bool>
Notes	▪ Not allowed for "DUT_" hooks. ▪ Not supported for locations defined by the DUT Control Interface script.

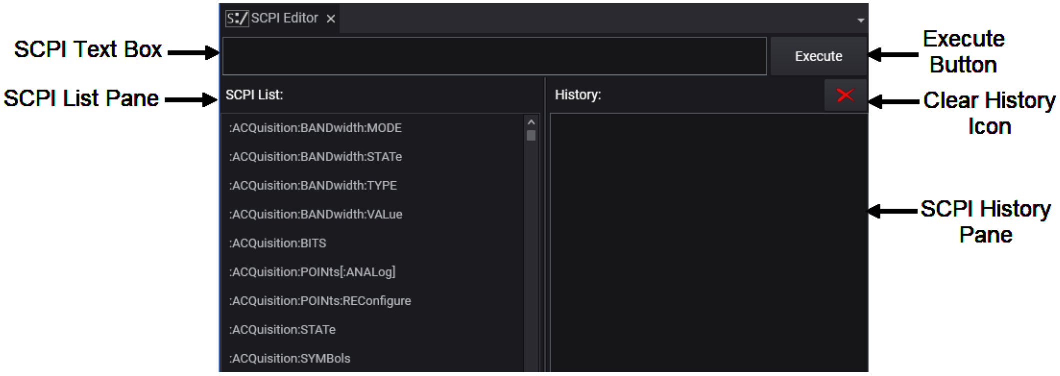
M8000.Instrument.Analyzer.GetCdrUnlock(groupOrLocationName, combineIntoOneResult)

Functionality	Read the CDR unlock status bit from the addressed locations and return either the logical OR of all addressed locations, or each location's status individually.
Parameters	
groupOrLocationName	Addresses the locations that shall be read. The use of .Net regular expression patterns is allowed.
combineIntoOneResult	▪ True Return the logical OR of all addressed status bits. ▪ False Return a dictionary of type IReadOnlyDictionary<string, bool> that holds an entry for each addressed location.
Returns	bool or IReadOnlyDictionary<string, bool>
Notes	▪ Not allowed for "DUT_" hooks. ▪ Not supported for locations defined by the DUT Control Interface script.

SCPI Editor

The **SCPI Editor** lists all SCPI that can be used to program M8020A/M8040A/M8050A and also provides a platform to execute them.

The following figure depicts the elements of the **SCPI Editor**:



```
:ACQuisition:BANDwidth:MODE
:ACQuisition:BANDwidth:STATe
:ACQuisition:BANDwidth:TYPE
:ACQuisition:BANDwidth:VALue
:ACQuisition:BITS
:ACQuisition:POINts[:ANALog]
:ACQuisition:POINts:REConfigure
:ACQuisition:STATe
:ACQuisition:SYMBols
```

The listed elements are described below:

- **SCPI Text Box** - Allows you to type the SCPI.
- **SCPI List Pane** - Lists all SCPI related to M8020A/M8040A/M8050A.
- **Execute Button** - Click this button to run the SCPI.
- **SCPI History Pane** - Maintains the history of the executed SCPI commands and also displays the invalid command errors.
- **Clear History Icon** - Clears the contents of the SCPI History Pane.

SCPI Basics

There are a number of key areas to consider when using SCPI for the first time. These are as follows:

- Instrument Model
- Command Syntax
- Optional Command Keywords
- Query Responses
- Command Separators
- SCPI Command Structure
- Invalid Commands

Instrument Model

SCPI guidelines require that the M8020A/M8040A/M8050A is compatible with an instrument model. This ensures that when using SCPI, functional compatibility is achieved between instruments that perform the same tasks. For example, if two different instruments have a programmable clock frequency setting, then both instruments would use the same SCPI commands to set their frequency. The instrument model is made up of a number of subsystems.

The sub-system defines a group of functions within a module and has a unique identifier under SCPI, which is called the **Root Keyword**.

Command Syntax

Commands may be up to twelve characters long. A short-form version is also available which has a preferred length of four characters or less. In this document the long-form and short-form versions are shown as a single word with the short-form being shown in upper-case letters.

For example, the long-form node command SOURce has the short-form SOUR. Using the short form saves time when entering a program; however, using the long form makes a program more descriptive and easier to understand.

SCPI commands may be commands only, commands and queries, or queries only. A question mark at the end of a command indicates that it is a query. If the question mark appears in brackets ([?]), the command has a command and query form.

Optional Command Keywords

Some layers in the SCPI command structure are optional. These optional keywords are indicated by square brackets ([]). A typical use for these types of keywords is with a command that is unique to one module. In this case, the top layer (Root Keyword) of the command structure may be omitted.

For example, the following command code segments are functionally identical:

```
[:SOURce]:JITTer[:GLOBal][:STATe] <ON|OFF|1|0>
:JITTer <ON|OFF|1|0>
:JITT <ON|OFF|1|0>
:jitt <ON|OFF|1|0>
```

Note that it is not necessary to include the syntax inside the square brackets ([]).

Query Responses

It is possible to interrogate the individual settings and status of a device using query commands. Retrieving data is a two-stage operation.

The query command is sent from the controller using the OUTPUT statement and the data is read from the device using the ENTER statement. A typical example is the SCPI IEEE 488.2 Common Command *IDN? which queries the identity of a device.

NOTE

When sending strings to the instrument, either the double quote (") or the single quote may be used ('), the latter being more suited to PASCAL programs, which make use of a single quote; the former being more suited to use in BASIC programs, which use a double quote as a delimiter.

Command Separators

The SCPI command structure is hierarchical and is governed by commas, semicolons and colons:

- Commas are used to separate parameters in one command.
- Colons are used to separate levels.
- Semicolons are used to send more than one command to the instrument at a time.

It is possible to send several commands in one pass, as long as the commands all belong to the same node in the SCPI tree. The commands have to be separated by semicolons.

The following SCPI commands provide examples of this.

```
SOURce:VOLTage:OFFSet ‘M2.DataOut2’,-0.99
SOURce:VOLTage:AMPLitude ‘M2.DataOut2’,1.11
```

These commands can also be sent as follows:

```
VOLT:OFFS ‘M2.DataOut2’,-0.99; ‘M2.DataOut2’,AMPL
1.11
```

SCPI Command Structure

The SCPI command structure can be best examined by means of an example. For example, the command to set the generator's output amplitude is:

```
[:SOURce]:VOLTage[:AMPLitude] ‘M1.DataOut1’,1.11
```

The structure of this command can be illustrated as follows:

- [:SOURce] - This is the top layer of the command structure and identifies the source subsystem.
- :VOLTage - This is the next layer and defines the subnode for setting a voltage level.
- [:AMPLitude] - This is the command itself for setting the output amplitude level.
- 'M1.DataOut1', 1.11 - This specifies module 1, DataOut port of channel 1, and specifies an amplitude of 1.11.

NOTE

Any optional commands are enclosed in square brackets [] and any optional characters are shown in lower case.

A colon indicates a change of level in the command hierarchy. Commands at the same level in the hierarchy may be included in the same command line, if separated by a semi-colon.

The bar symbol (|) indicates mutually exclusive commands.

To translate this syntax into a command line, follow the convention described above. Remember, however, that the command line can be created in several different ways. It can be created with or without optional keywords, and in a long or short form. The following example gives three possible forms of the command line; all are acceptable.

In long form:

```
:SOURce:VOLTage:AMPLitude 'M1.DataOut1',1.11
```

In short form:

```
:SOUR:VOLT:AMPL 'M1.DataOut1',1.11
```

With the optional commands removed:

```
:VOLT 'M1.DataOut1',1.11
```

The long form is the most descriptive form of programming commands in SCPI.

Invalid Commands

A command is invalid and will be rejected if:

- It contains a syntax error.
- It cannot be identified.
- It has too few or too many parameters.

- A parameter is out of range.
- It is out of context.

Executing SCPI Commands

To execute a SCPI command, follow the given steps:

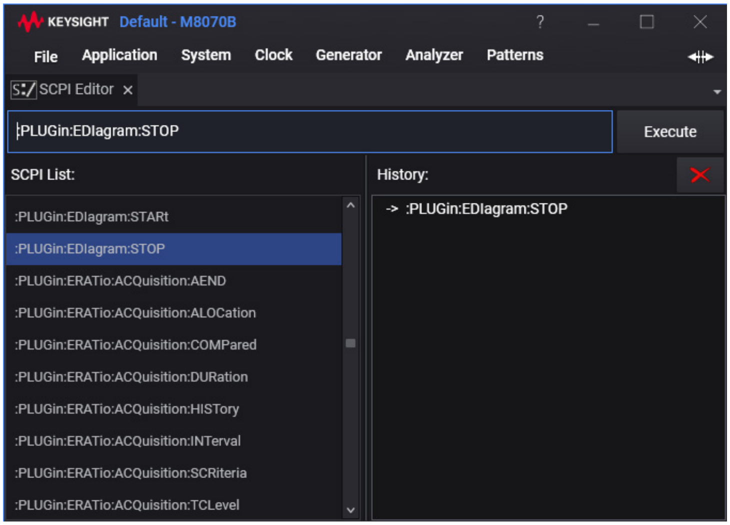
1 Select the SCPI from the given list. You can also type the SCPI in the provided text box to expedite the command search.

2 Use the proper SCPI command syntax along with the command separators. For complete details, refer to “SCPI Basics" on page 144.

The following example shows a SCPI command to enable Global Jitter State:

```
:SOURce:JITTer:GLOBal:STATe ‘M1.SYSTEM’, ON
```

3 Click **Execute**. The output of the SCPI command will be displayed in the **History** pane as shown in the following figure:



:PLUGin:EDIagram:STOP

SCPI List:

```
:PLUGin:EDIagram:STARt
:PLUGin:EDIagram:STOP
:PLUGin:ERATio:ACQuisition:AEND
:PLUGin:ERATio:ACQuisition:ALOCation
:PLUGin:ERATio:ACQuisition:COMPared
:PLUGin:ERATio:ACQuisition:DURation
:PLUGin:ERATio:ACQuisition:HISTory
:PLUGin:ERATio:ACQuisition:INTerval
:PLUGin:ERATio:ACQuisition:SCRiteria
:PLUGin:ERATio:ACQuisition:TCLevel
```

History:

```
-> :PLUGin:EDIagram:STOP
```

4 Click the ✕ **Clear History** icon to clear the contents of **History** pane.

For complete details on M8020A/M8040A/M8050A SCPI commands, refer to the *M8000 Series Programming Guide*.

4 Supported Configurations

This chapter describes the configurations supported by the M8070B to support other hardwares.

Overview

The Advanced Measurement Package allows the following instruments to be controlled from M8070B System Software:

- N1076A/77A
- N1076B/78A
- Real-Time Oscilloscope
- Enhancements made in M8070B GUI to control Real-Time Oscilloscope
- Controlling a UXR-Series Real-Time Oscilloscope from M8070B
- M8045A Automatic Deemphasis Optimization
  - DCA-X 86100D with 86108B PWA
  - DCA-X N1000A with N1060A
  - N1094A
  - N1094B
  - N1092C
  - N1092E

NOTE **Please note that the Advanced Measurement Package requires license for its activation. For details on license, see “License Information” on page 18.**

Controlling N1076A/77A from M8070B

The N1076A electrical clock recovery and N1077A electrical/optical clock recovery provide necessary clock signals to the error detector in M8070B. They take an incoming data (or clock) signal, locks onto it using a phase-locked loop (PLL) circuit, and outputs a recovered clock. The recovered clock can be used as a timing reference for oscilloscopes or M8070B.

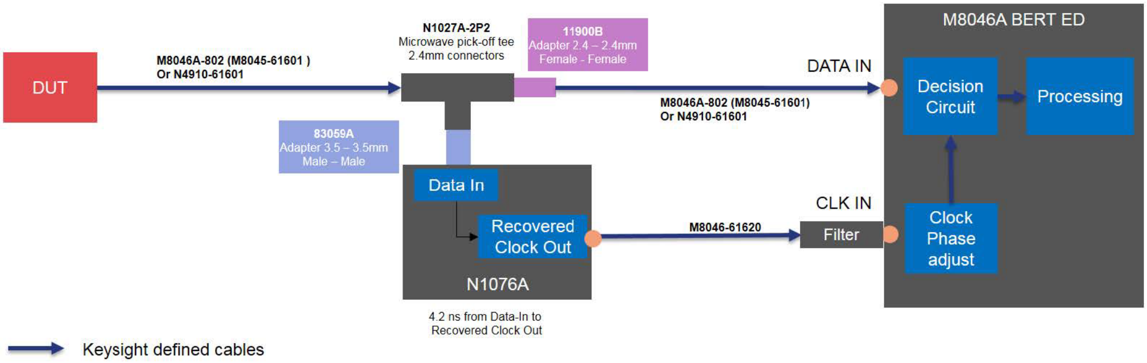
N1076A /77A cannot be controlled directly from M8070B, but requires a running instance of FlexDCA N1000-Series System Software.

NOTE

The minimum required version of the FlexDCA N1000-Series System Software is A.05.61.23.

Cabling Setup for N1076A/N1077A

The following figure shows the recommended connection setup which should be made to control N1076A/77A from M8070B system software.



NOTE

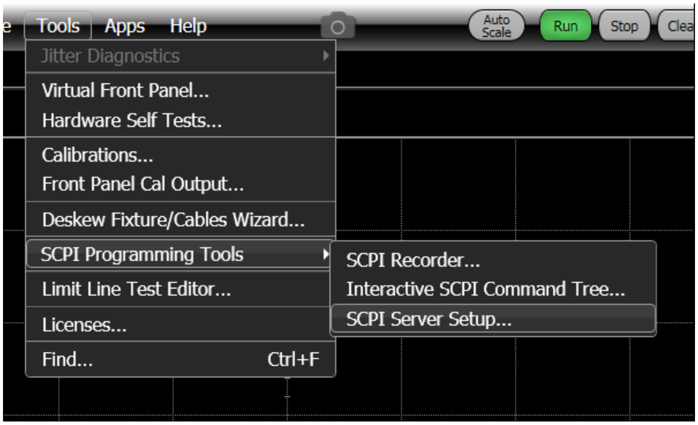
The filter at CLK IN needs to be used for symbol rates of 25 GBd and above and has to be removed to symbol rates below 25 GBd.

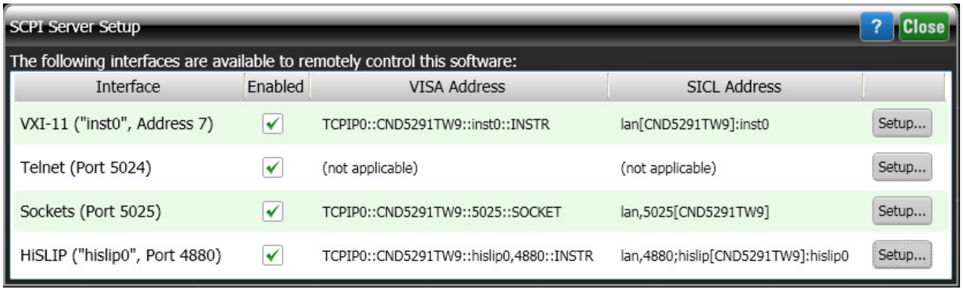
One Time Preparation

In order to make N1076A/77A accessible in M8070B please prepare the setup as follows.

1 Install FlexDCA N1000-Series System Software on the computer that the N1076A/77A is connected to via USB.In most cases this will be the same computer that executes M8070B.

2 Start FlexDCA N1000-Series System Software and enable the remote programming interface.





The following interfaces are available to remotely control this software:

Interface	Enabled	VISA Address	SICL Address	
VXI-11 ("inst0", Address 7)	☑	TCPIP0::CND5291TW9::inst0::INSTR	lan[CND5291TW9]:inst0	Setup...
Telnet (Port 5024)	☑	(not applicable)	(not applicable)	Setup...
Sockets (Port 5025)	☑	TCPIP0::CND5291TW9::5025::SOCKET	lan,5025[CND5291TW9]	Setup...
HiSLIP ("hislip0", Port 4880)	☑	TCPIP0::CND5291TW9::hislip0,4880::INSTR	lan,4880;hislip[CND5291TW9]:hislip0	Setup...

3 Manually add FlexDCA N1000-Series System Software as an instrument to the I/O Libraries Suite and define the VISA alias that enables M8070B to remote control FlexDCA.

   a Open Keysight Connection Expert.

   b Select Manual Configuration.

      i Add New Instruments/Interfaces.

      ii Select "LAN instrument":

         a. Enter the Hostname or IP Address of the computer that runs FlexDCA.

         b. Set Protocol to HiSlip and ensure the Remote Name matches the name that is shown in FlexDCA's SCPI Server Setup.

         c. Test the VISA address to verify that the communication path is setup correctly.

         d. Click "Accept".

   c Select Instruments:

      i Select the just added instrument at the left side. This may show either "Unknown Instrument" or "N1010A, Keysight Technologies FlexDCA". Verify the VISA address that is shown below the name to ensure this is the correct instrument.

      ii Click on "Add or Change Aliases" to define one of the alias names that tell M8070B to remote control this instance of FlexDCA.

         a. Click "Add New VISA Alias".

         b. Alias Name: N1076A_PROXY

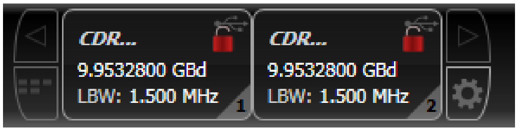
         c. VISA Address: Select the VISA ;address that matches the FlexDCA instrument.

         d. Click "Accept".

Using N1076A/77A from within M8070B

In order to use N1076A/77A from within M8070B, the FlexDCA N1000-Series System Software has to be started and fully running before starting M8070B. To ensure this, follow the following startup sequence.

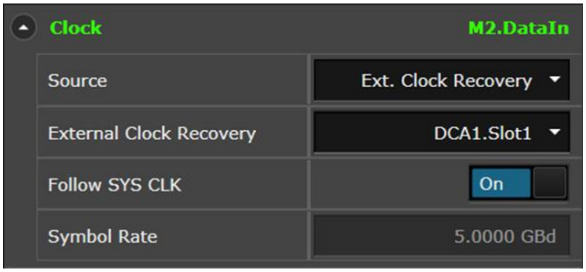
1. Power-up all N1076A/77A that are connected to the PC.
2. Start FlexDCA and wait until the software has completely started up.
3. Verify that the N1076A/77A are correctly assigned to the Extended Modules Slots and that they are connected within FlexDCA. They have to show up at the bottom of the FlexDCA window.



CDR...
9.9532800 GBd
LBW: 1.500 MHz
1

CDR...
9.9532800 GBd
LBW: 1.500 MHz
2

4. Start M8070B System Software.
5. M8070B will show a dedicated module for the FlexDCA instrument.
   - a Press the configure button to enable individual slots of FlexDCA to be controlled from M8070B.
   - b The enabled slots will show up on the module. Click on the respective button to access the clock recovery's parameters.
6. The N1076A/77A can be assigned to an M8046A error detector to be used as an external clock recovery. In this case M8046A will ensure that certain critical parameters are automatically kept in sync between M8046A and N1076A/77A.
   - a Select 'Ext. Clock Recovery' as Clock Source of M8046A.
   - b Choose one of the available external clock recoveries under "External Clock Recovery".



Clock M2.DataIn

Source: Ext. Clock Recovery
External Clock Recovery: DCA1.Slot1
Follow SYS CLK: On
Symbol Rate: 5.0000 GBd

NOTE

When the line coding “PAM4” is selected for the loopback mode, it is recommended to adjust the pattern generator data output de-emphasis for the post cursor for error free operation (e.g. +0.05 at 25 Gbaud).

As soon as FlexDCA is integrated into M8070B, it will no longer be accessible for interactive use.

All remote programming of FlexDCA must be done through M8070B. For more details on the SCPI commands, refer to *M8000 Series Programming Guide*. Since M8070B is only supporting a subset of the FlexDCA functionality, this will restrict the use of FlexDCA to the control of N1076A/77A. Any other functionality is not accessible through M8070B. These are basically the same SCPI commands that are defined by FlexDCA.

The difference to FlexDCA is that they are located under the root node :N1000, there is no suffix for :SLOT and :CREC, and that there is an extra command argument which holds the M8070B specific identifier (e.g. DCA1.Slot2). But functionality is exactly like when controlling a N1010A FlexDCA directly.

The only M8070B specific command is

```
:N1000:EMODules:SLOT:CONNect[?]<identifier>,{ON | OFF
| 1 | 0}
```

This command controls if the specific slot is being enabled to be used by M8070B. It does not establish the connection between N1010A and N1076A/77A. Connecting the N1076A/77A must be done within N1010A prior starting M8070B.

The FlexDCA SCPI Commands are documented in the N1010A FlexDCA online help.

The M8070B documentation shall only have a basic description for the FlexDCA specific commands and point the user to the detailed documentation at the FlexDCA side.

FlexDCA and N1076A/77A Documentation

Click on the following links to access the N1076A/77A related documents:

N1076A/77A Data Sheet
http://literature.cdn.keysight.com/litweb/pdf/5992-1620EN.pdf?id=2772197

N1076A/77A User Guide
http://literature.cdn.keysight.com/litweb/pdf/N1076-90001.pdf?id=2784631

N1010A FlexDCA Online Help
http://www.keysight.com/main/redirector.jspx?action=ref&cname=EDITORIAL&ckey=2038334&lc=ger&cc=DE&nfr=-11143.0.00

Offline Usage

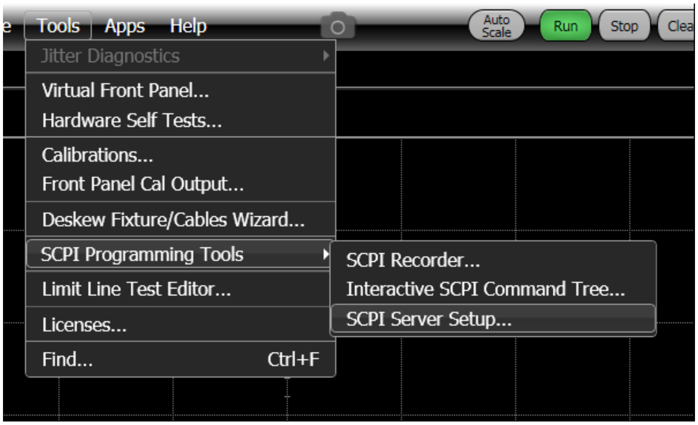
It is possible to run M8070B in an offline configuration for the external clock recovery. This can be useful to get familiar with the use model or when preparing remote programs without having the actual hardware available.

The easiest way of getting the offline configuration is to start with an online configuration and then simply leaving the cock recovery turned off, the next time M8070B is being started.

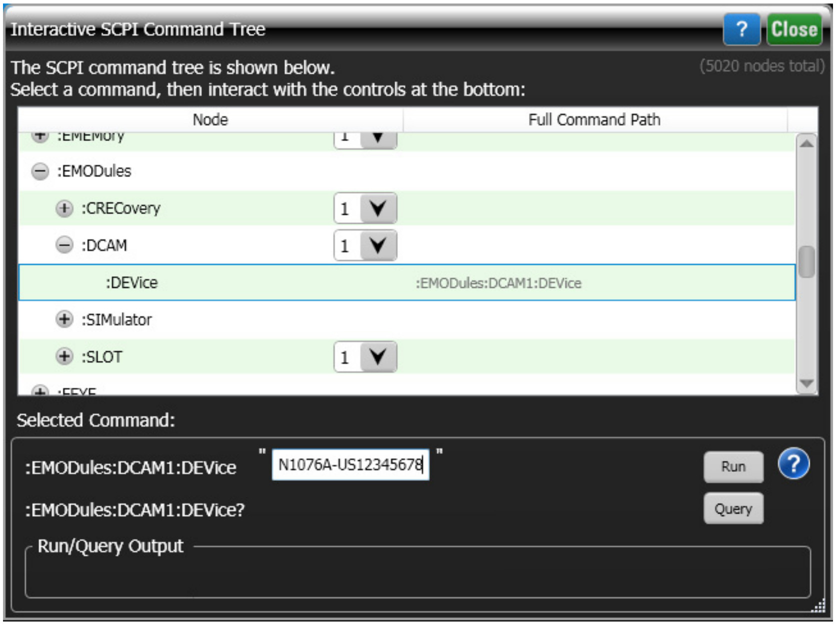
If a clock recovery is not available to prepare the offline configuration, then this can still be achieved manually.

1 Start N1010 FlexDCA Software.

2 In FlexDCA open the Interactive SCPI Command Tree.



3 Execute the command
`:EMODulesDCAM1:DEVice "N1076A-US12345678"`
Select slot number and serial number as required. The example above is using slot 1 (DCAM1) and the serial number US12345678.



Interactive SCPI Command Tree

The SCPI command tree is shown below.
Select a command, then interact with the controls at the bottom:

(5020 nodes total)

Node		Full Command Path
:EMEMory	1	
:EMODules		
:CRECovery	1	
:DCAM	1	
:DEVice		:EMODules:DCAM1:DEVice
:SIMulator		
:SLOT	1	
:EEYE		

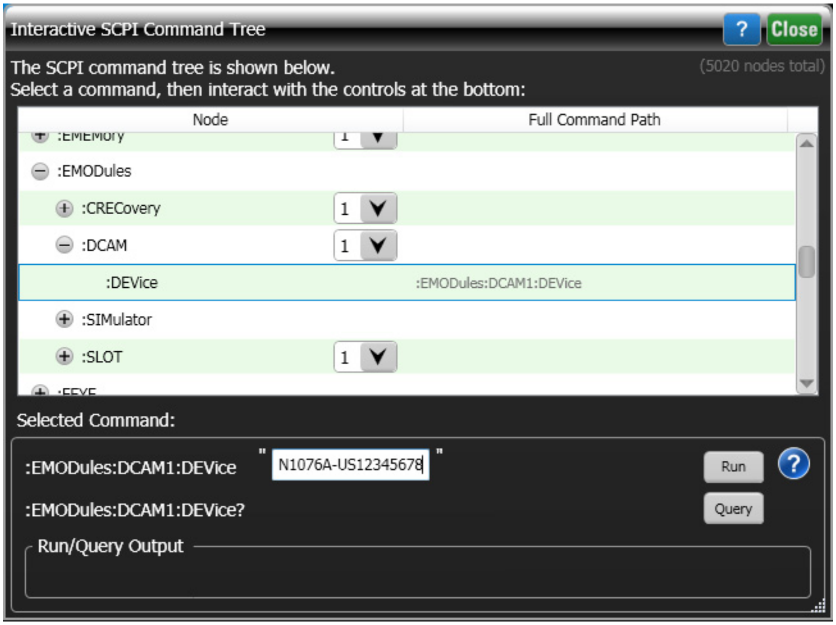
Selected Command:

:EMODules:DCAM1:DEVice " N1076A-US12345678 "

:EMODules:DCAM1:DEVice?

Run/Query Output

4 Execute the command
`:EMODules:SLOT1:SELection DCAM`
Make sure to use the same slot number (1 in this example) as for the previous command.



Interactive SCPI Command Tree

The SCPI command tree is shown below.
Select a command, then interact with the controls at the bottom:

(5020 nodes total)

Node	Full Command Path
:EMEMory 1	
:EMODules	
:CRECovery 1	
:DCAM 1	
:DEVice	:EMODules:DCAM1:DEVice
:SIMulator	
:SLOT 1	
:EEYE	

Selected Command:

:EMODules:DCAM1:DEVice "N1076A-US12345678"

:EMODules:DCAM1:DEVice?

Run/Query Output

5 Ensure that N1010A FlexDCA is configured in Keysight Command Expert and the alias is being set.

6 Start M8070B and use the clock recovery in offline mode.

The procedure for offline clock recovery configuration is similar to online one and can also be selected as clock source for a M8046A error detector. If the M8046A error detector is online, then a clock with the correct frequency must be supplied at the CLK IN connector.

In case of offline clock recovery, all commands will executed in the similar way line in online case with the exception of the lock status. The clock recovery will never report unlocked when in offline simulation mode.

Controlling N1076B/78A from M8070B

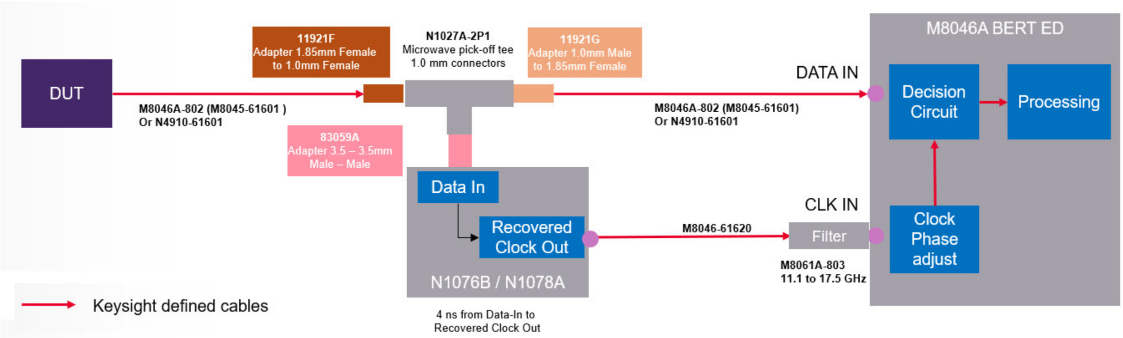
The N1076B electrical clock recovery and N1078A electrical/optical clock recovery provide necessary clock signals to the error detector in M8070B. They take an incoming data (or clock) signal, locks onto it using a phase-locked loop (PLL) circuit, and outputs a recovered clock. The recovered clock can be used as a timing reference for oscilloscopes or M8070B.

N1076B /78A cannot be controlled directly from M8070B, but requires a running instance of FlexDCA N1000-Series System Software.

NOTE **The minimum required version of the FlexDCA N1000-Series System Software is A.05.61.23.**

Cabling Setup for N1076B/N1078A

The following figure shows the recommended connection setup for PAM4 and NRZ up to 64 GBd which should be made to control N1076B/78A from M8070B system software.



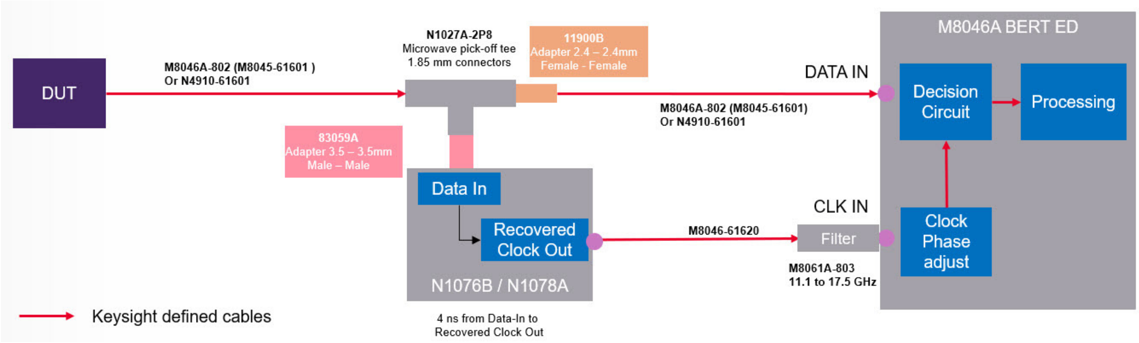
NOTE **The filter at CLK IN must be used for symbol rates above 32 GBd and up to 32.414 GBd. It may not be used above 32.414 GBd.**

NOTE

Note for direct loop-back:

For a PAM-4 loopback setup adjust the post cursor setting of the M8045A (e.g. +0.05 at 25 GBd) to compensate the fast transition times of the M8057A/B.

The following figure shows the recommended connection setup for PAM4 and NRZ up to 58 GBd which should be made to control N1076B/78A from M8070B system software.



NOTE

The filter at CLK IN must be used for symbol rates above 32 GBd and up to 32.414 GBd. It may not be used above 32.414 GBd.

NOTE

Note for direct loop-back:

For a PAM-4 loopback setup adjust the post cursor setting of the M8045A (e.g. +0.05 at 25 GBd) to compensate the fast transition times of the M8057A/B.

New/Additional features of N1076B/N1078A

The following are the new/additional features of N1076B/N1078A compared to N1076A/N1077A:

- Adjustable equalizer at the electrical inputs of the clock recovery.This can be adjusted independently to the equalizer at the data input of M8046A.
- Supports symbol rates up to 64 GBd.

One Time Preparation

In order to make N1076B/78A accessible in M8070B, please prepare the setup as described in the section "Controlling N1076A/77A from M8070B" on page 151.

Using N1076B/78A from within M8070B

In order to use N1076B/78A from within M8070B, the FlexDCA N1000-Series System Software and the M8070B System Software has to be started. Please refer to the steps described in the section "Controlling N1076A/77A from M8070B" on page 151.

N1076B/78A Documentation

Click on the following links to access the N1076B/78A and Clock recovery related documents:

N1076B

https://www.keysight.com/en/pdx-2892129-pn-N1076B/electrical-clock-recovery?nid=-33198.1234885.00&cc=IN&lc=eng

N1078A

https://www.keysight.com/en/pdx-2906605-pn-N1078A/optical-electrical-clock-recovery?nid=-33198.1238480&cc=US&lc=eng

Clock Recovery Data Sheet

https://literature.cdn.keysight.com/litweb/pdf/5992-1620EN.pdf?id=2772197

License Requirements

Using any of the external clock recoveries (N1076A, N1076B, N1077A, N1078A) does require the M8070ADVB license. Without the M8070ADVB license, the clock recoveries will not integrate into M8070B.

Controlling a Real-Time Oscilloscope from M8070B

The M8070B system software (version 6.0 and above) supports certain real-time oscilloscopes to measure BER of NRZ signals and BER as well as SER of PAM4 signals. The real-time oscilloscope is completely controlled from M8070B to capture a signal and convert into a pattern stream that is then uploaded into the M8070B for comparison with an expected pattern and provide BER/SER measurements.

Nevertheless, it is possible to stop measuring BER on the integrated oscilloscope and therefore allow interactive tweaking of oscilloscope settings before continuing the BER measurements.

Currently, the following oscilloscopes models are supported:

- DSOZ634A Infiniium Z-Series Digital Storage Oscilloscope 63GHz / 160Gsa/s
- DSAZ634A Infiniium Z-Series Digital Signal Analyzer 63GHz / 160Gsa/s
- DSOZ594A Infiniium Z-Series Digital Storage Oscilloscope 59GHz / 160Gsa/s
- DSAZ594A Infiniium Z-Series Digital Signal Analyzer 59GHz / 160Gsa/s
- DSOZ504A Infiniium Z-Series Digital Storage Oscilloscope 50GHz / 160Gsa/s
- DSAZ504A Infiniium Z-Series Digital Signal Analyzer 50GHz / 160Gsa/s
- DSOX96204Q Infiniium Q-Series Digital Storage Oscilloscope 63GHz / 160Gsa/s
- DSAX96204Q Infiniium Q-Series Digital Signal Analyzer 63GHz / 160Gsa/s
- Infiniium UXR-Series Real-Time Oscilloscopes (For detailed list, see "Controlling a UXR-Series Real-Time Oscilloscope from M8070B" on page 196).

NOTE

The M8070B software supports N7005A optical-to-electrical converter connected with real-time scope as an error detector. For more information, refer to "Supporting the N7005A Optical-to-Electrical Converter" on page 201.

Detailed description of the above oscilloscopes can be found at: http://www.keysight.com/find/oscilloscopes

The following licenses are required on the oscilloscope:

- N5384A Serial Data Analysis (SDA)
- N8827A PAM4 Measurement (PM4)
- N5461A Equalization (DEQ)

NOTE

The minimum supported Infiniium version is 06.10.00616.

All measurements will be done using the Real Edge inputs of the oscilloscope. The achievable target BER is 1e-6.

Required preparation steps

The following preparations must be made to configure the real-time oscilloscope for usage by M8070B. Once this preparation is done, simply power-on the oscilloscope and wait until it has completely started-up before launching M8070B.

- Ensure that the oscilloscope is running the minimum required version of Infiniium.
- Ensure the oscilloscope is having the options SDA and PM4 installed. DEQ is optional, and will enable the use of the oscilloscope's feed-forward equalizer in M8070B.
- Connect the oscilloscope to the computer that runs M8070B.

  Supported remote control interfaces are USB and LAN. In case of a LAN connection, only the HiSLIP protocol is supported.
- M8070B identifies the oscilloscopes to integrate by the existence of an VISA alias for the oscilloscope's the VISA address. This needs to be configured in the Keysight Connection Expert.

  The VISA alias names that will mark an oscilloscope for integration into M8070B are

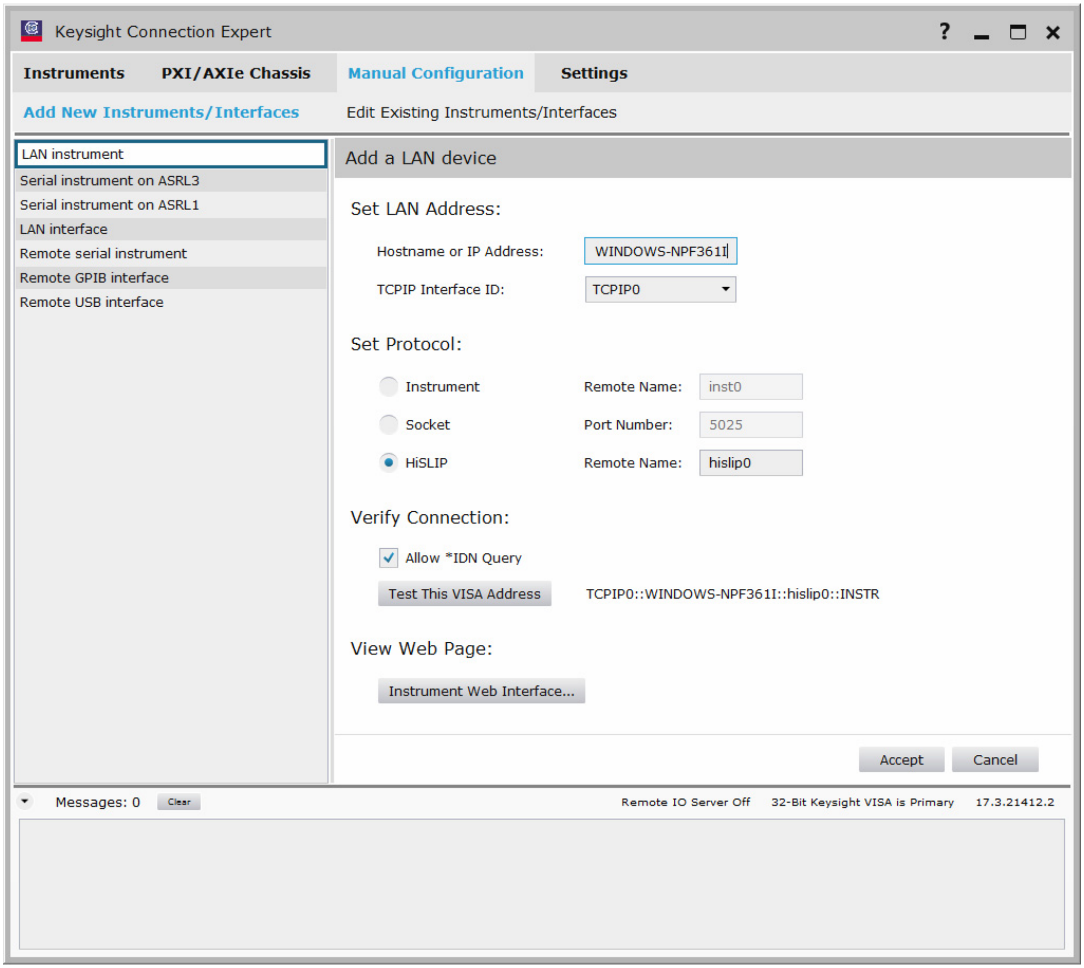
  - RTS_PROXY
  - RTS_PROXY_1
  - RTS_PROXY_2
  - RTS_PROXY_3
  - RTS_PROXY_4

Setting up a LAN controlled real-time oscilloscope for usage by M8070B

1 Add a new LAN device (this not required for USB connections)

Enter the oscilloscope's host name or IP address and select the HiSLIP protocol.

Click **Accept** when done.



Keysight Connection Expert

Add a LAN device

Set LAN Address:

Hostname or IP Address: WINDOWS-NPF361I

TCPIP Interface ID: TCPIP0

Set Protocol:

○ Instrument Remote Name: inst0

○ Socket Port Number: 5025

● HiSLIP Remote Name: hislip0

Verify Connection:

☑ Allow *IDN Query

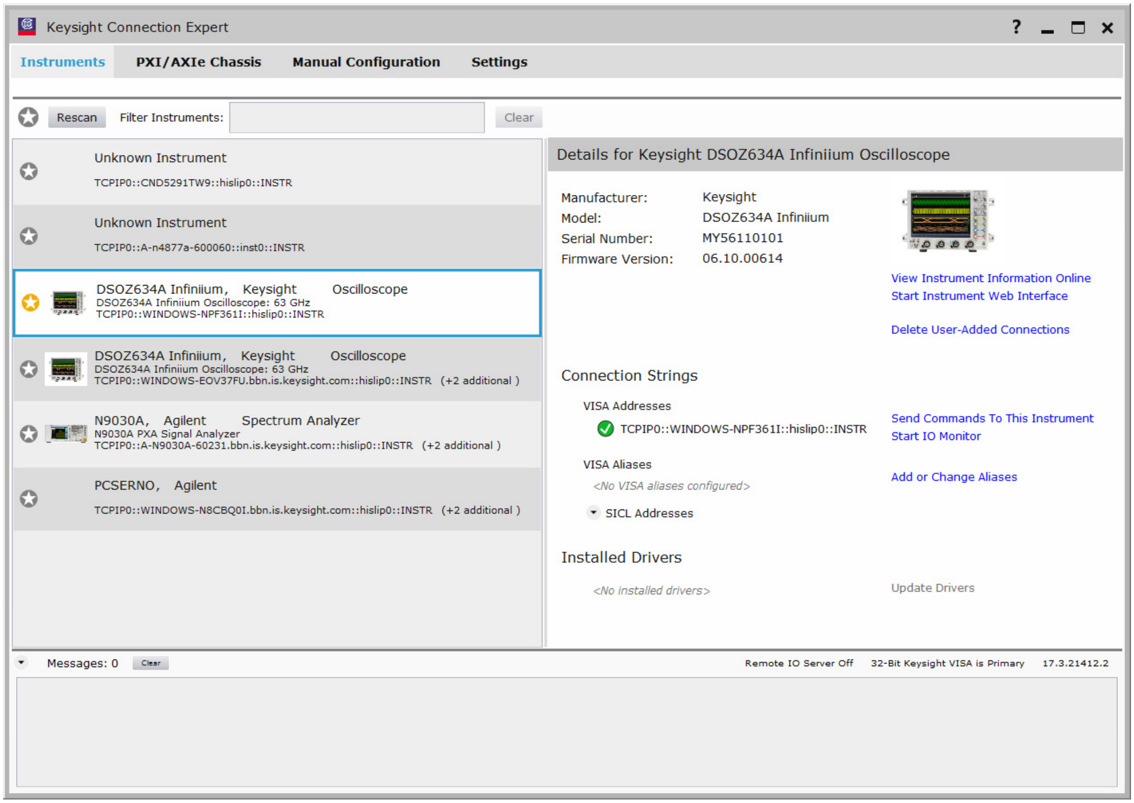
TCPIP0::WINDOWS-NPF361I::hislip0::INSTR

View Web Page:

Messages: 0

Remote IO Server Off 32-Bit Keysight VISA is Primary 17.3.21412.2

2 On the Instruments tab, select the oscilloscope.
Click '**Add or Change Aliases**' under **Instrument Details**.



Unknown Instrument
TCPIP0::CND5291TW9::hislip0::INSTR

Unknown Instrument
TCPIP0::A-n4877a-600060::inst0::INSTR

DSOZ634A Infiniium, Keysight Oscilloscope
DSOZ634A Infiniium Oscilloscope: 63 GHz
TCPIP0::WINDOWS-NPF361I::hislip0::INSTR

DSOZ634A Infiniium, Keysight Oscilloscope
DSOZ634A Infiniium Oscilloscope: 63 GHz
TCPIP0::WINDOWS-EOV37FU.bbn.is.keysight.com::hislip0::INSTR (+2 additional)

N9030A, Agilent Spectrum Analyzer
N9030A PXA Signal Analyzer
TCPIP0::A-N9030A-60231.bbn.is.keysight.com::hislip0::INSTR (+2 additional)

PCSERNO, Agilent
TCPIP0::WINDOWS-N8CBQ0I.bbn.is.keysight.com::hislip0::INSTR (+2 additional)

Details for Keysight DSOZ634A Infiniium Oscilloscope

Manufacturer:	Keysight
Model:	DSOZ634A Infiniium
Serial Number:	MY56110101
Firmware Version:	06.10.00614

View Instrument Information Online
Start Instrument Web Interface
Delete User-Added Connections

Connection Strings

VISA Addresses
✓ TCPIP0::WINDOWS-NPF361I::hislip0::INSTR

VISA Aliases
<No VISA aliases configured>

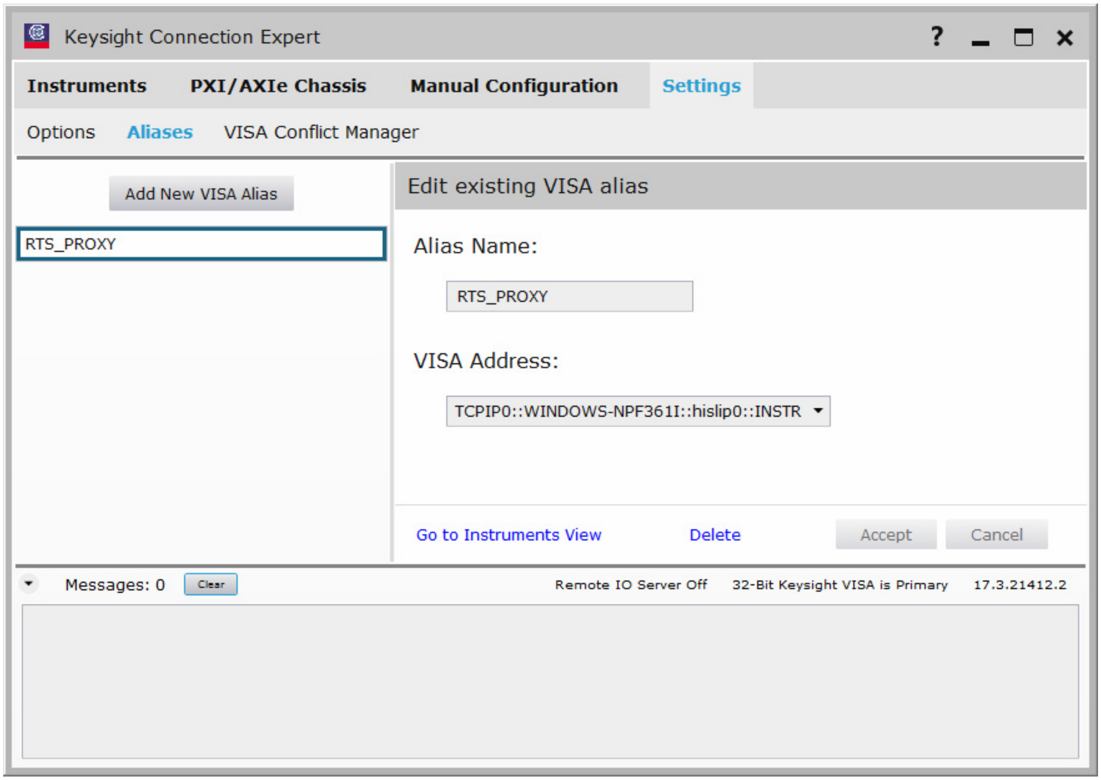
SICL Addresses

Installed Drivers
<No installed drivers>

Messages: 0 Remote IO Server Off 32-Bit Keysight VISA is Primary 17.3.21412.2

3 Ensure that the correct VISA address is selected, and enter the alias name.

Click **Accept** to assign the VISA alias.



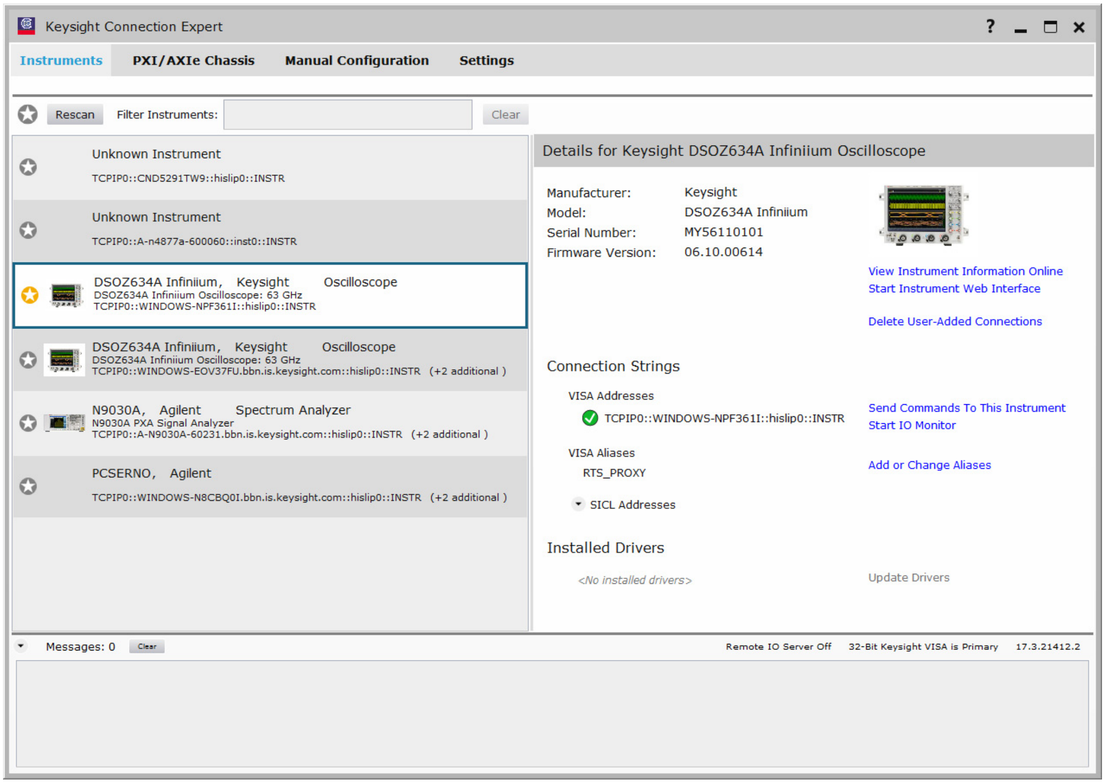
Alias Name: RTS_PROXY

VISA Address: TCPIP0::WINDOWS-NPF361I::hislip0::INSTR

Messages: 0

Remote IO Server Off 32-Bit Keysight VISA is Primary 17.3.21412.2

4 When everything is configured correctly, the Instrument Details should look like in the screen below.



Keysight Connection Expert

Unknown Instrument
TCPIP0::CND5291TW9::hislip0::INSTR

Unknown Instrument
TCPIP0::A-n4877a-600060::inst0::INSTR

DSOZ634A Infiniium, Keysight Oscilloscope
DSOZ634A Infiniium Oscilloscope: 63 GHz
TCPIP0::WINDOWS-NPF361I::hislip0::INSTR

DSOZ634A Infiniium, Keysight Oscilloscope
DSOZ634A Infiniium Oscilloscope: 63 GHz
TCPIP0::WINDOWS-EOV37FU.bbn.is.keysight.com::hislip0::INSTR (+2 additional)

N9030A, Agilent Spectrum Analyzer
N9030A PXA Signal Analyzer
TCPIP0::A-N9030A-60231.bbn.is.keysight.com::hislip0::INSTR (+2 additional)

PCSERNO, Agilent
TCPIP0::WINDOWS-N8CBQ0I.bbn.is.keysight.com::hislip0::INSTR (+2 additional)

Details for Keysight DSOZ634A Infiniium Oscilloscope

Manufacturer: Keysight
Model: DSOZ634A Infiniium
Serial Number: MY56110101
Firmware Version: 06.10.00614

View Instrument Information Online
Start Instrument Web Interface
Delete User-Added Connections

Connection Strings

VISA Addresses
✓ TCPIP0::WINDOWS-NPF361I::hislip0::INSTR

VISA Aliases
RTS_PROXY

SICL Addresses

Installed Drivers

<No installed drivers>

Messages: 0

Remote IO Server Off 32-Bit Keysight VISA is Primary 17.3.21412.2

Understanding BER measurement with the real-time oscilloscope

Despite the fact, that Infiniium is capable of measuring BER and SER under certain restrictions, these measurements will not be used when operating the oscilloscope in M8070B.

Instead, the oscilloscope is used as a digitizer and all the pattern comparison and error counting related steps are done by M8070B.

This allows to report a true error count, because M8070B compares against expected pattern and it produces the same detailed error count results as a regular error detector does.

Additionally the standard Error Ratio and Jitter Tolerance measurements of M8070B can be used, which is not possible with Infiniium's built-in BER/SER measurements.

Integrating an oscilloscope into M8070B as an error detector does introduce a few oscilloscope related terms that did not exist in the context of error detectors in the past. One important term to understand is acquisition.

A regular error detector is processing the received data in real-time and is comparing and counting each and every bit that it receives. An oscilloscope as error detector, or better digitizer, is not capable of doing this.

The oscilloscope is capturing the sampled input signal into its acquisition memory and then starts processing of the captured data in order to improve signal shape, apply clock recovery or equalizer algorithms. While the oscilloscope is doing this processing, it does not capture the received input signal. Which means that only a relatively small number of bits is being captured and finally compared with the expected pattern, relative to the time that it takes to process the bits in software.

The steps that it takes from capturing the received signal in the acquisition memory to finally updating the error counters in M8070B are summarized under the term acquisition.

In M8070B the acquisition is normally enabled, thus BER is getting measured. But it can be disabled. This leaves the error detector parameters still under control of M8070B, but additionally enables changing or tweaking oscilloscope settings locally. Therefore settings that are not directly offered by M8070B can still be optimized for the specific measurement task. For example, selecting a type of clock recovery that is not offered by M8070B, or manually fine tuning the threshold values.

While acquisition is enabled, the local user interface of the oscilloscope is locked and no local changes are possible.

The most important parameter in the context of acquisition is the number of bits that shall be processed in one acquisition cycle. This will define the required acquisition memory size on the oscilloscope. Setting the number of bits to a larger value will lower the detectable BER per acquisition. And since a single acquisition directly corresponds to an error counter update in M8070B, this defines the lowest BER value greater than 0.0 that the instantaneous BER will report.

Therefore, the number of bits per acquisition should be at least as large that the targeted minimum detectable BER can be achieved. On the other hand, a large memory size on the oscilloscope side does result in longer execution times for the signal processing. And this causes latencies between adjusting a parameter and getting the first BER result that is based on the new setting and reduces the BER update frequency.

It is best practice to configure the number of bits per acquisition to the smallest possible value that still results in the desired detectable BER. Configuring a small number of bits, will result in an instantaneous BER that becomes more and more insensitive. This is of course causing trouble when relying on the instantaneous BER result, but it is not much of a problem for accumulating error ratio measurements, under the premise that the error distribution is random.

The processing steps within a single acquisition cycle are:

1. The oscilloscope captures the waveform
2. The oscilloscope applies FFE algorithms
3. The oscilloscope applies clock recovery algorithms
4. The oscilloscope applies the decision threshold(s) and extracts digital symbol information
5. M8070B reads the captured symbols from the oscilloscope
6. M8070B aligns the captured data with the expected data
7. M8070B compares the captured data with the expected data
8. M8070B updates the accumulated and instantaneous error counters

These steps are repeated continuously, and only paused when the acquisition is stopped, or a parameter value is getting changed.

After step 8, the error count information is available in M8070B, just like for a hardware based error detector, and is available for measurements.

The currently supported measurements are:

- Error Ratio
- Jitter Tolerance

The currently supported patterns are:

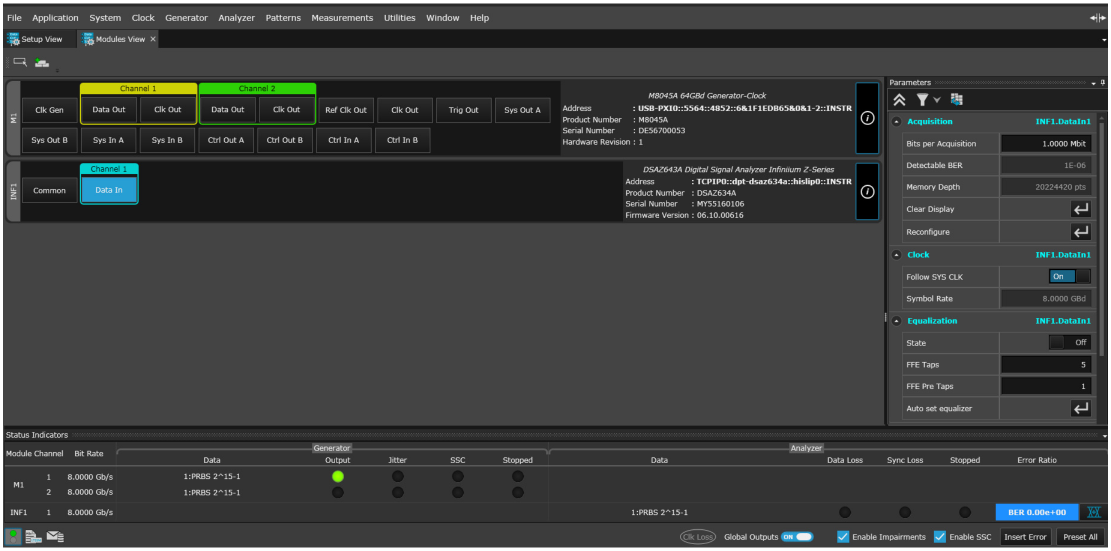
- PRBS 2^n-1
  - n = 7, 9, 10, 11, 13, 15, 23, 31, 33, 35, 39, 41, 45, 47, 49, 51
- Memory patterns
  - Maximum pattern length = 256 kbit

NOTE

Using the equalizer auto-set function when measuring a clock pattern will result in incorrect tap-coefficients. The tap optimization requires a certain amount of ISI to produce valid results. This is not the case for a clock signal. Therefore, the equalizer should not be enabled for a clock pattern.

Real-time oscilloscope in Module View

Once the real-time oscilloscope setup is done, you will see its entry in the **Module View** as shown in the following figure:



M8045A 64GBd Generator-Clock

Address : USB-PXI0::5564::4852::6&1F1EDB65&0&1-2::INSTR
Product Number : M8045A
Serial Number : DE56700053
Hardware Revision : 1

DSAZ643A Digital Signal Analyzer Infiniium Z-Series

Address : TCPIP0::dpt-dsaz634a::hislip0::INSTR
Product Number : DSAZ634A
Serial Number : MY55160106
Firmware Version : 06.10.00616

Acquisition	INF1.DataIn1
Bits per Acquisition	1.0000 Mbit
Detectable BER	1E-06
Memory Depth	20224420 pts
Clear Display	↵
Reconfigure	↵
Clock	**INF1.DataIn1**
Follow SYS CLK	On
Symbol Rate	8.0000 GBd
Equalization	**INF1.DataIn1**
State	Off
FFE Taps	5
FFE Pre Taps	1
Auto set equalizer	↵

Module	Channel	Bit Rate	Generator					Analyzer				
			Data	Output	Jitter	SSC	Stopped	Data	Data Loss	Sync Loss	Stopped	Error Ratio
M1	1	8.0000 Gb/s	1:PRBS 2^15-1	●	●	●	●					
	2	8.0000 Gb/s	1:PRBS 2^15-1	●	●	●	●					
INF1	1	8.0000 Gb/s						1:PRBS 2^15-1	●	●	●	BER 0.00e+00

Global Outputs ON ☑ Enable Impairments ☑ Enable SSC

The left side of the figure shows the connected modules and the right side shows the **Parameters** window. The corresponding real-time oscilloscope parameters can be configured through the **Parameters** window.

Differences to a hardware based error detector

Due to the post processing style of error counting, there are some differences that need to be taken into account when measuring BER.

1 The measurement duration to achieve a certain target BER with a given confidence level, or simply to compare a given number of bits, is significantly longer than with a hardware based error detector.

Therefore a meaningful target BER for a real-time oscilloscope based setup is 1e-6 or higher.

2 Elapsed time cannot be used to estimate or define the number of compared bits.

This is having impact on how measurements, especially the error ratio measurement, have to be configured when specifying the measurement duration. Always use either a pass/fail criteria or specify the number of bits to measure. Do not specify the measurement duration in terms of time.

3 Single error events may not get recognized.

The smallest recognizable BER per acquisition is defined by the number of bits per acquisition under DataIn1 -> Acquisition. Error events that occur with a lower probability will not be detected reliably if these are single events. However, when running the error ratio measurement in pass/fail mode, or until a specified number of bits has been compared, the error information will be statistically correct for randomly distributed error events. Increasing the number of bits per acquisition will not only lower the smallest detectable BER of the instantaneous BER result, but also reduce the probability that recurring error events are missed.

4 Inserting single errors, or using an error insertion ratio that is less than the detectable BER, will most likely result in not detecting the error events.

5 Auto-alignments do stop the acquisition. This can have side effects in controlling software if this is monitoring the status bits.

6 The configured pattern sequence is absolutely uncorrelated to any other data channel or sequence in M8070B. This is even true, if multiple error detectors are using the same sequence.

7 The instantaneous BER updates only after a complete acquisition cycle has finished. Depending on the configured number of bits per acquisition, symbol rate, clock recovery and equalizer configuration, this can be in the range of every second to several minutes.

Best practices

The following procedure has proven be a reliable way of configuring the real-time oscilloscope based error detector.

1. Disable acquisition
2. Configure all parameters. Especially the symbol rate and clock recovery's loop bandwidth are critical to ensure that the clock recovery will lock.
3. Ensure the signal at the inputs matches the configured parameters.

   There are cases when the oscilloscope becomes extremely unresponsive if the received signal is not matching the configured parameters. This is mostly caused by misconfiguration of the clock recovery loop bandwidth, or symbol rate.
4. Run a full auto-alignment and observe the oscilloscope display.

   If no eye diagram is shown or warning messages are displayed, then adjust parameters as suggested by Infiniium.
5. Keep the acquisition depth as low as acceptable.

   If there is no need for a specific minimum detectable BER or confidence level per acquisition, then choose the smallest allowed value, which depends on the expected pattern selection. This will ensure better responsiveness of M8070B and increases the BER update rate.
6. Enable acquisition

   Following these steps should ensure that the oscilloscope is successfully acquiring data and M8070B is therefore able to display the resulting BER. If this state is reached, then the parameter settings are good enough to adjust parameters individually. But be aware that choosing the wrong loop bandwidth setting for the clock recovery can not only result in errors due to an unlocked clock recovery, but also have severe impact on processing times of the oscilloscope.

For the best user experience and to avoid common problems that are caused by incorrect settings, please follow the given recommendations:

- Disable acquisition before doing changes to the incoming signal that do not match the parameter settings.

  Keeping the acquisition enabled while the received signal is in conflict with the oscilloscope settings, can cause the oscilloscope to get stuck in processing the signal under wrong parameter configuration.
- Run a full auto-alignment when all parameters are configured, and the signal is applied to the oscilloscope inputs. But do this before enabling acquisition on this signal for the first time.

This will allow an interactive user to observe the oscilloscope's display and adjust settings in case warning messages are shown on the oscilloscope display.

- Take advantage of the possibility to operate the oscilloscope's local user interface while the acquisition is disabled. But be aware that M8070B is not detecting local changes and is not updating its parameters accordingly. Use the Reconfigure function to re-apply the setting as defined in M8070B to synchronize the settings of M8070B and Infiniium.
- When measuring jitter tolerance, configure the number of bits per acquisition to minimum.

  The jitter tolerance measurement is always doing a pass/fail measurement and is collecting acquisitions until the required measurement depth is reached. If the error distribution can be assumed to be random, then it is safe to reduce the bits per acquisition in order to achieve a better responsiveness of the setup.
- If the measurement setup can provide a high precision reference clock (e.g. from the pattern generator being used), then better results can be achieved if this reference is used as the reference for the horizontal time base of the oscilloscope. This is especially true for high symbol rates.

  When using a M8045A as pattern generator, the Ref Clk Out can be configured to 100 MHz and connected to the 100 MHz in connector of the oscilloscope. When this is done, select the External Reference Clock as 100 MHz.

Remote controlling the real-time oscilloscope based error detector

This section provides an overview of the SCPI commands that are relevant for controlling an error detector that is based on a real-time oscilloscope. Many of the commands listed below are also valid for other error detector channels, except for those that configure the acquisition, clock recovery and equalizer parameters.

It is good practice to stop the acquisition manually, when changing multiple parameters and only enable the acquisition, once the parameters are all configured. This avoids that M8070B will automatically stop and restart the acquisition with every parameter that is getting programmed. Additionally, the remote program must to use large enough VISA timeouts that cover the time that it takes to stop acquisition, or auto-set functions that have been initiated. This is important when reprogramming parameters while the acquisition is enabled. There is no need for large timeouts if parameters are only set when the acquisition is stopped and the oscilloscope is idle.

Refer to the *M8000 Series Programming Guide* the for details regarding the commands listed in the following tables.

Acquisition related commands

Parameter	Identifier	Commands
State	INF*.Common	:ACQuisition:STATe <identifier>, {OFF|ON|0|1} :ACQuisition:STATe? <identifier>
Reconfigure	INF*.Common INF*.DataIn1	:ACQuisition:REConfigure <identifier>
Bits per Acquisition	INF*.DataIn1	:ACQuisition:BITS<identifier>,<number_of_bits> :ACQuisition:BITS? <identifier>
Symbols per Acquisition	INF*.DataIn1	:ACQuisition:SYMBols <identifier>,<number_of_UI> :ACQuisition:SYMBols? <identifier>
Memory Depth	INF*.Common INF*.DataIn1	:ACQuisition:POINts[:ANALog]? <identifier>

Clock and clock reference related commands

Parameter	Identifier	Commands
Follow SYS CLK	INF*.DataIn1	:CLOCk:TRACk[:STATe] <identifier>,{OFF|ON|0|1}
Symbol Rate	INF*.DataIn1	:SENSe:SRATe[?] <identifier>,<NRf> :SENSe:SRATe? <identifier>

External Reference Clock	INF*.Common	:TRIGger[:SOURce] <identifier>,{INTernal|REFerence}
		:TRIGger:REFerence[:FREQuency] <identifier>,{REF10|REF100}

Equalization related commands

Parameter	Identifier	Commands
Feed-Forward Equalizer (FFE) Commands		
FFE State	INF*.DataIn1	:INPut:EQUalization:FFEQualizer:STATe <identifier>,{OFF|ON|0|1} :INPut:EQUalization:FFEQualizer:STATe? <identifier>
FFE Taps	INF*.DataIn1	:INPut:EQUalization:FFEQualizer:NTAPs <identifier>,<number> :INPut:EQUalization:FFEQualizer:NTAPs? <identifier>
FFE Pre Taps	INF*.DataIn1	:INPut:EQUalization:FFEQualizer:NPRecursor <identifier>,<number> :INPut:EQUalization:FFEQualizer:NPRecursor? <identifier>
FFE Auto-set equalizer	INF*.DataIn1	:INPut:EQUalization:FFEQualizer:TAP:AUTO <identifier> :INPut:EQUalization:FFEQualizer:TAP:AUTO? <identifier>
FFE Tap Width	INF*.DataIn1	:INPut:EQUalization:FFEQualizer:TAP:WIDTh <identifier>,<width> :INPut:EQUalization:FFEQualizer:TAP:WIDTh? <identifier>
FFE Auto-align Optimizes EQ	INF*.DataIn1	:INPut:EQUalization:FFEQualizer:TAP:AUTO:INCLude <identifier>, <0|1|OFF|ON> :INPut:EQUalization:FFEQualizer:TAP:AUTO:INCLude? <identifier>
Constant Time Linear Equalizer (CTLE) Commands		
CTLE State	INF*.DataIn1	:INPut:EQUalization:CTLequalizer:STATe <identifier>, {OFF | ON | 0 | 1} :INPut:EQUalization:CTLequalizer:STATe? <identifier>
CTLE Preset	INF*.DataIn1	:INPut:EQUalization:CTLequalizer:PRESet <identifier>, “preset-name” :INPut:EQUalization:CTLequalizer:PRESet? <identifier>
Number of Poles	INF*.DataIn1	:INPut:EQUalization:CTLequalizer:NUMPoles <identifier>, { P2Z1 | P3Z1 | P3Z2 | P2ACG } :INPut:EQUalization:CTLequalizer:NUMPoles? <identifier>
CTLE AC Gain	INF*.DataIn1	:INPut:EQUalization:CTLequalizer:ACGain <identifier>,<NRf> :INPut:EQUalization:CTLequalizer:ACGain? <identifier>
CTLE DC Gain	INF*.DataIn1	:INPut:EQUalization:CTLequalizer:DCGain <identifier>,<NRf> :INPut:EQUalization:CTLequalizer:DCGain? <identifier>
CTLE Pole 1 Frequency	INF*.DataIn1	:INPut:EQUalization:CTLequalizer:P1 <identifier>, <NRf> :INPut:EQUalization:CTLequalizer:P1? <identifier>
CTLE Pole 2 Frequency	INF*.DataIn1	:INPut:EQUalization:CTLequalizer:P2 <identifier>, <NRf> :INPut:EQUalization:CTLequalizer:P2? <identifier>
CTLE Pole 3 Frequency	INF*.DataIn1	:INPut:EQUalization:CTLequalizer:P3 <identifier>, <NRf> :INPut:EQUalization:CTLequalizer:P3? <identifier>

CTLE Zero 1 Frequency	INF*.DataIn1	:INPut:EQUalization:CTLequalizer:Z1 <identifier>, <NRf> :INPut:EQUalization:CTLequalizer:Z1? <identifier>
CTLE Zero 2 Frequency	INF*.DataIn1	:INPut:EQUalization:CTLequalizer:Z2 <identifier>, <NRf> :INPut:EQUalization:CTLequalizer:Z2? <identifier>
Decision feedback Equalization (DFE) Commands		
DFE State	INF*.DataIn1	:INPut:EQUalization:DFEQualizer:STATe <identifier>, <OFF|ON|0|1> :INPut:EQUalization:DFEQualizer:STATe? <identifier>
DFE Taps	INF*.DataIn1	:INPut:EQUalization:DFEQualizer:NTAPs <identifier>,<number> :INPut:EQUalization:DFEQualizer:NTAPs? <identifier>
DFE Auto set equalizer	INF*.DataIn1	:INPut:EQUalization:DFEQualizer:TAP:AUTO <identifier>
DFE Auto-align Optimizes EQ	INF*.DataIn1	:INPut:EQUalization:DFEQualizer:TAP:AUTO:INCLude <identifier>, <0|1|OFF|ON> :INPut:EQUalization:DFEQualizer:TAP:AUTO:INCLude? <identifier>

Clock recovery related commands

Parameter	Identifier	Commands
State	INF*.DataIn1	:INPut:CDR:OJTF[:STATe][?] <identifier>,{OFF|ON|0|1}
Loop Bandwidth	INF*.DataIn1	:INPut:CDR:OJTF:SECond:LBANdwidth <identifier>,<NRf> :INPut:CDR:OJTF:SECond:LBANdwidth? <identifier>
Loop Bandwidth Divisor	INF*.DataIn1	:INPut:CDR:OJTF:SECond:LBANdwidth:DIVisor <identifier>,<NRf> :INPut:CDR:OJTF:SECond:LBANdwidth:DIVisor? <identifier>
Damping Factor	INF*.DataIn1	:INPut:CDR:OJTF:SECond:DFACtor <identifier>,<NRf> :INPut:CDR:OJTF:SECond:DFACtor? <identifier>

Line coding related commands

Parameter	Identifier	Commands
Coding	INF*.DataIn1	:DATA:LINecoding:VALue? <identifier>,<PAM4|NRZ> :DATA:LINecoding:VALue <identifier>
Symbol Mapping	INF*.DataIn1	:DATA:LINecoding:PAM4:MAPPing <identifier>,<NONE|GRAY|CUSTom> :DATA:LINecoding:PAM4:MAPPing? <identifier>
Custom Symbol Mapping	INF*.DataIn1	:DATA:LINecoding:PAM4:MAPPing:CUSTom <identifier>,<string> :DATA:LINecoding:PAM4:MAPPing:CUSTom? <identifier>

Comparator related commands

Parameter	Identifier	Commands
Compare Mode	INF*.DataIn1	:INPut:CMODe <identifier>,<DIFFerential|SINGleended> :INPut:CMODe? <identifier>
Polarity	INF*.DataIn1	:INPut:POLarity <identifier>,<NORMal|INVerted> :INPut:POLarity? <identifier>
Auto-set thresholds	INF*.DataIn1	:INPut:ALIGnment:EYE:ACEN <identifier>

Sample delay related commands

Parameter	Identifier	Commands
Delay	INF*.DataIn1	:INPut:delay <identifier>,<NRf> :INPut:delay? <identifier>
Auto-set delay	INF*.DataIn1	:INPut:ALIGnment:EYE:TCENter <identifier>

Error counter related commands

Parameter	Identifier	Commands
Instantaneous BER	INF*.DataIn1	:FETCh:IBERate? <identifier>
Accumulated BER since last acquisition start	INF*.DataIn1	:FETCh:BCOunter? <identifier>
Sync Loss Threshold	INF*.DataIn1	:DATA:SYNC:THReshold[?] <identifier>,<NRF>

Automatic parameter optimizations

Function	Identifier	Commands
Auto-alignment	INF*.DataIn1	:INPut:ALIGnment:EYE[:AUTO] <identifier>
Auto-set thresholds	INF*.DataIn1	:INPut:ALIGnment:EYE:ACEN <identifier>
Auto-set delay	INF*.DataIn1	:INPut:ALIGnment:EYE:TCENter <identifier>
Auto-set equalizer	INF*.DataIn1	:INPut:EQUalization:FFEQualizer:TAP:AUTO <identifier>

M8070B Enhanced Support for Real-Time Oscilloscopes

The steps required to control a Real-Time Oscilloscope from M8070B is similar as described in section "Controlling a Real-Time Oscilloscope from M8070B" on page 162.

The following are the enhancements introduced in M8070B release 6.5 onwards for controlling a real-time oscilloscope:

1 The M8070B system software supports for real-time oscilloscopes is now a part of Advanced Measurement Package and requires M8070ADVB license to activate. The real-time oscilloscope will not get integrated into M8070B, if the M8070ADVB license is not installed.

2 Pattern Capture

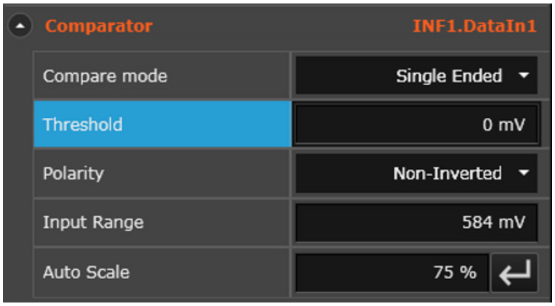
   a The pattern capture can acquire the data that is captured by the oscilloscope as an immediate triggered pattern capture.

   b The capture depth is limited to the acquisition depth configured on the oscilloscope.

   c For details refer to the feature description of "Pattern Capture for Real Time Oscilloscopes" on page 187.

3 Threshold value can be controlled from M8070B

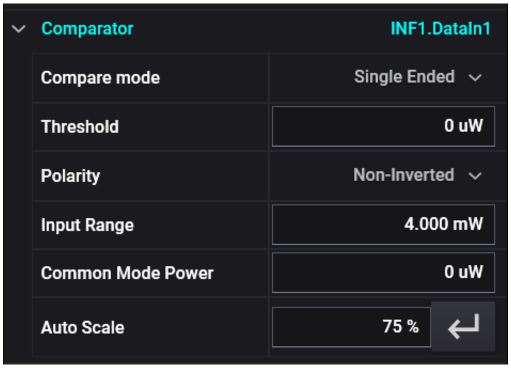
   a The following figure shows the threshold value in volts for electrical signal supported in NRZ mode:



Comparator INF1.DataIn1

Compare mode	Single Ended
Threshold	0 mV
Polarity	Non-Inverted
Input Range	584 mV
Auto Scale	75 %

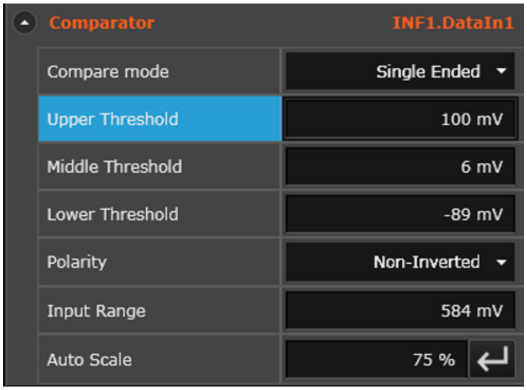
The following figure shows the threshold value in watt for optical signal supported in NRZ mode when the probe is connected. For more information, refer to “Supporting the N7005A Optical-to-Electrical Converter” on page 201.



Comparator INF1.DataIn1

Compare mode	Single Ended
Threshold	0 uW
Polarity	Non-Inverted
Input Range	4.000 mW
Common Mode Power	0 uW
Auto Scale	75 %

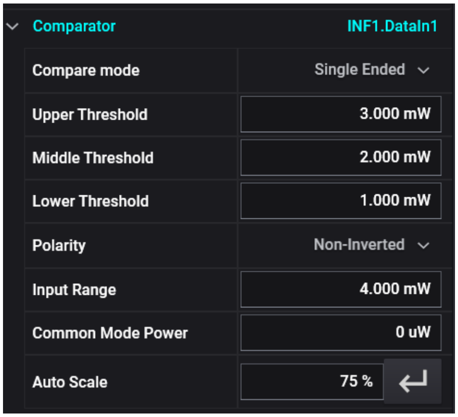
b The following figure shows the threshold value in volts for electrical signal supported in PAM4 mode:



Comparator INF1.DataIn1

Compare mode	Single Ended
Upper Threshold	100 mV
Middle Threshold	6 mV
Lower Threshold	-89 mV
Polarity	Non-Inverted
Input Range	584 mV
Auto Scale	75 %

The following figure shows the threshold value in watt for optical signal supported in PAM4 mode when the probe is connected.



Comparator	INF1.DataIn1
Compare mode	Single Ended
Upper Threshold	3.000 mW
Middle Threshold	2.000 mW
Lower Threshold	1.000 mW
Polarity	Non-Inverted
Input Range	4.000 mW
Common Mode Power	0 uW
Auto Scale	75 %

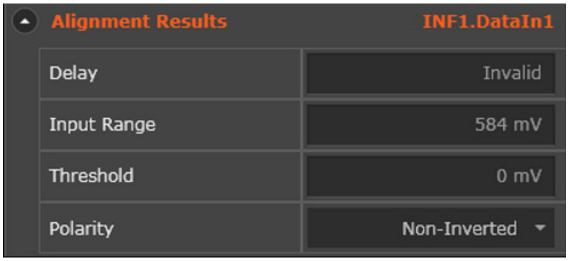
c Remote Programming (For details, refer to *M8000 Series Programming Guide*)

- `:INPut:VOLTage:NRZ:THReshold 'INF1.DataIn1',0`
- `:INPut:VOLTage:PAM4:SYMBol:THReshold3 'INF1.DataIn1',0.125`
- `:INPut:VOLTage:PAM4:SYMBol:THReshold2 'INF1.DataIn1',0`
- `:INPut:VOLTage:PAM4:SYMBol:THReshold1 'INF1.DataIn1',-0.125`

4 Auto alignment results are reported via GUI and SCPI

The Delay result is only available in PAM4 mode, because there is no adjustable clock skew available on the real-time oscilloscope in NRZ mode. Eye height and width results are unavailable for a real-time oscilloscope based error detector. The alignment algorithms for the real-time oscilloscope error detector are not doing BER measurements to center the sample delay and threshold. Therefore there is no information about a certain BER threshold available on a real-time oscilloscope.

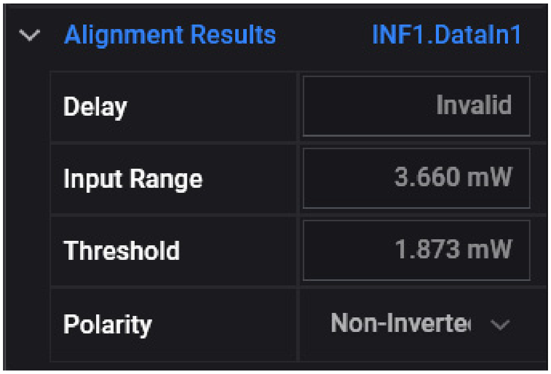
a The following figure shows the auto alignment delay results in volts for electrical signal in NRZ mode:



Alignment Results INF1.DataIn1

Delay	Invalid
Input Range	584 mV
Threshold	0 mV
Polarity	Non-Inverted

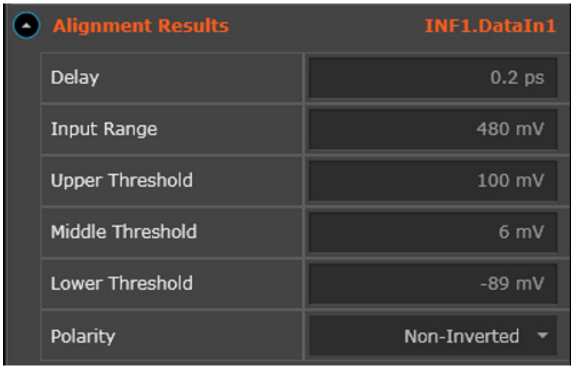
The following figure shows auto alignment delay results in watt for optical signal in NRZ mode when the probe is connected:



Alignment Results INF1.DataIn1

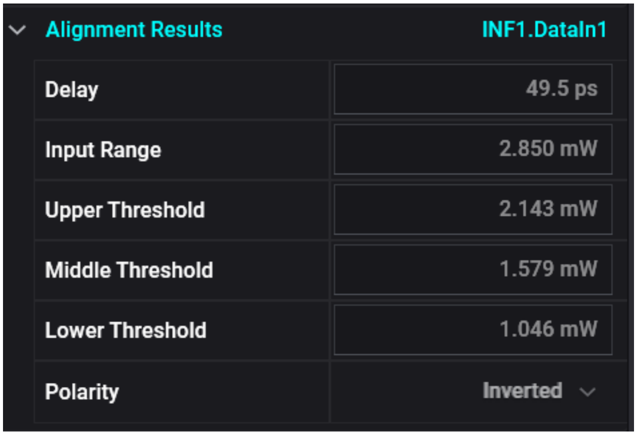
Delay	Invalid
Input Range	3.660 mW
Threshold	1.873 mW
Polarity	Non-Inverte

b The following figure shows the auto-alignment delay results in volts for electrical signal in PAM4 mode:



Alignment Results	INF1.DataIn1
Delay	0.2 ps
Input Range	480 mV
Upper Threshold	100 mV
Middle Threshold	6 mV
Lower Threshold	-89 mV
Polarity	Non-Inverted

The following figure shows auto-alignment delay results in watt for optical signal in PAM4 mode when the probe is connected:



Alignment Results	INF1.DataIn1
Delay	49.5 ps
Input Range	2.850 mW
Upper Threshold	2.143 mW
Middle Threshold	1.579 mW
Lower Threshold	1.046 mW
Polarity	Inverted

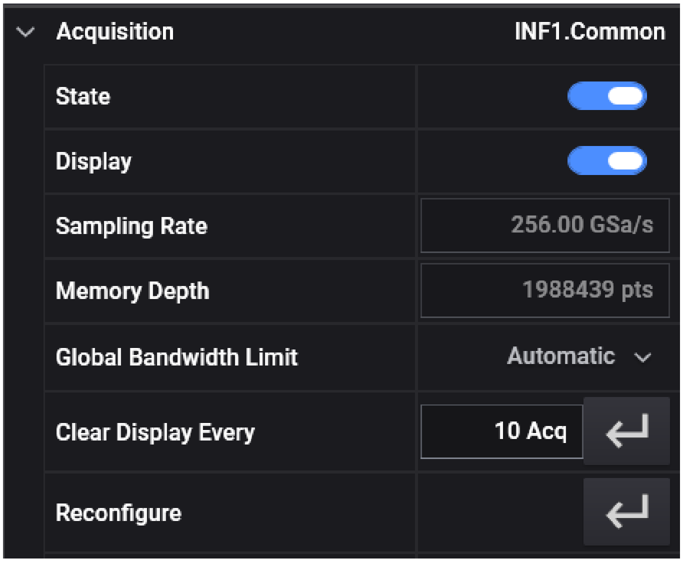
c Remote Programming (For details, refer to *M8000 Series Programming Guide*)

- `:INPut:ALIGnment:EYE:RESult:DELay? 'INF1.DataIn1'`
- `:INPut:ALIGnment:EYE:RESult:NRZ:THReshold? 'INF1.DataIn1'`
- `:INPut:ALIGnment:EYE:RESult:PAM4:THReshold3? 'INF1.DataIn1'`
- `:INPut:ALIGnment:EYE:RESult:PAM4:THReshold2? 'INF1.DataIn1'`
- `:INPut:ALIGnment:EYE:RESult:PAM4:THReshold1? 'INF1.DataIn1'`
- `:INPut:ALIGnment:EYE:RESult:POLarity? 'INF1.DataIn1'`

5 Bandwidth controls from M8070B:

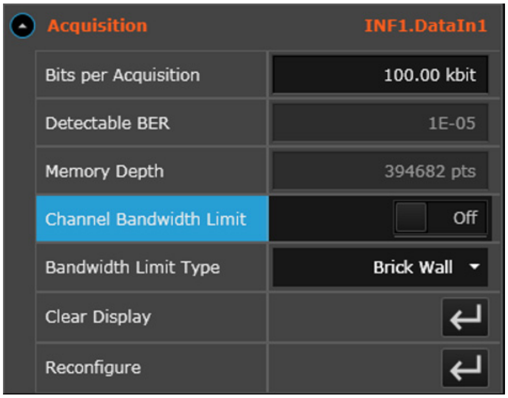
a The oscilloscopes global and channel bandwidth settings are controllable from M8070B.

b The global bandwidth is controlled from the **Acquisition** parameters.



Acquisition	INF1.Common
State	On
Display	On
Sampling Rate	256.00 GSa/s
Memory Depth	1988439 pts
Global Bandwidth Limit	Automatic
Clear Display Every	10 Acq
Reconfigure	

c The channel bandwidth settings are controlled from the Data In **Acquisition** parameters



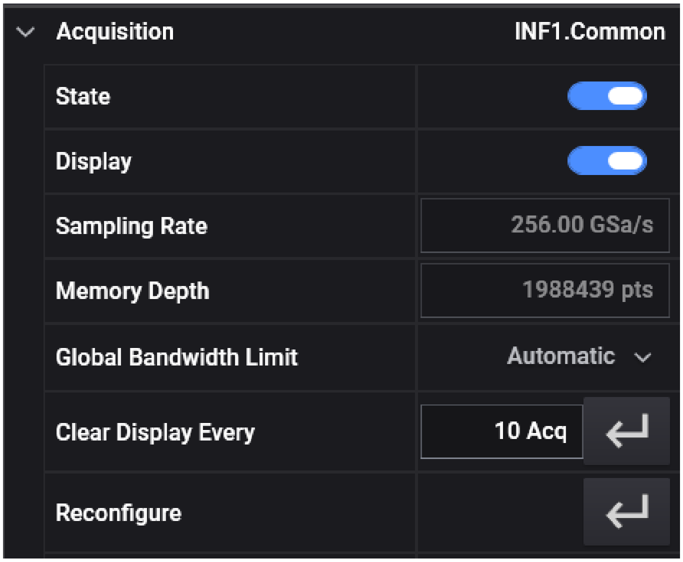
Acquisition INF1.DataIn1

Bits per Acquisition	100.00 kbit
Detectable BER	1E-05
Memory Depth	394682 pts
Channel Bandwidth Limit	Off
Bandwidth Limit Type	Brick Wall
Clear Display	
Reconfigure	

d Remote Programming (For details, refer to *M8000 Series Programming Guide*)

- `:ACQuisition:BANDwidth:MODE[?] 'INF1.Common',{AUTO|MANual|MAXimum}`
- `:ACQuisition:BANDwidth:VALue[?] 'INF1.Common',62e9`
- `:ACQuisition:BANDwidth:STATe 'INF1.DataIn1',{0|1|OFF|ON}`
- `:ACQuisition:BANDwidth:TYPE 'INF1.DataIn1',{WALL|BESSEL4}`

6 You can increase the hardware acceleration up to five times when the real time scope is used as error detector. This option is supported by M8070B 9.5.510.10 and above. You have to turn off (disable) the **Display** button in the **Acquisition** functional block in order to switch to the fast acceleration mode.



Acquisition	INF1.Common
State	●
Display	●
Sampling Rate	256.00 GSa/s
Memory Depth	1988439 pts
Global Bandwidth Limit	Automatic
Clear Display Every	10 Acq
Reconfigure	

The hardware acceleration can be achieved when the following conditions are met:

- The symbol rate should be in the range to 51.2 GBd to 120 GBd.
- In the CDR functional block, the CDR state should be enabled and the loop bandwidth should be in the range of 500 to 2656.

Remote Programming (For details, refer to *M8000 Series Programming Guide*)

- `:ACQuisition:DISPlay[?]'INF1.Common',{0|1|OFF|ON}`

Pattern Capture for Real Time Oscilloscopes

The M8070B analyzer provides capability to capture the data received from the real-time oscilloscopes. The captured data bits are displayed in the **Pattern Capture Window** in binary or 8b/10b symbol coding. The received data is compared with the expected data and the errored bits/symbols are highlighted. The captured data can be saved for post processing.

NOTE **Ensure that you have enabled the “Follow SYS CLK” under Clock, to use the same symbol rate.**

The steps to launch and capture the patterns is similar to the pattern capture for BERT Analyzers (M8041A, M8051A, M8045A and M8062A). For details, refer to the **Pattern Capture** section in the *M8000 Series User Guide*.

For description **Pattern Capture Window** and its elements, refer to the **Pattern Capture** section in the *M8000 Series User Guide.*

To to capture the data received from the real-time oscilloscopes, the **Parameters Window** provides the following acquisition and show parameters:

Acquisition Parameters

The acquisition parameters are summarized in the following list:

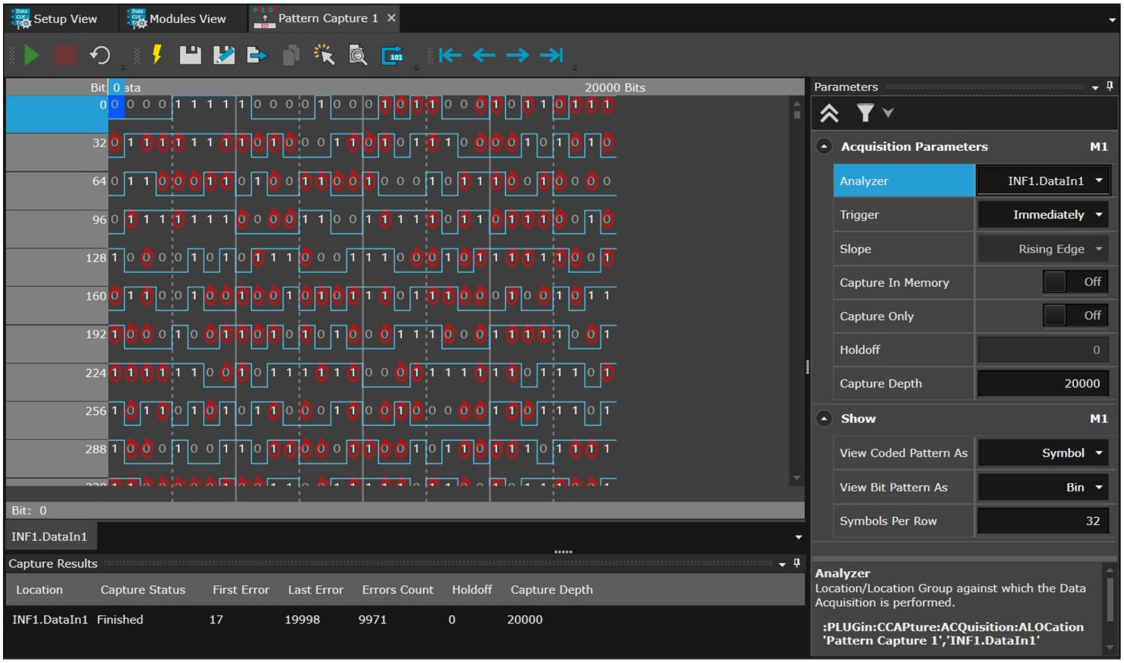
- **Analyzer** - Use this drop-down menu to select the channel against which the data capture has to be performed. For the real-time oscilloscopes, select INF1.DataIn1.
- **Trigger** - Selects the event that is triggering the captured logic. It can be captured on the following stop events:
  - **Immediately** – Starts capturing the data immediately and displays the captured data.
- **Slope** - Selects the edge (rising edge or falling edge) of CTRL IN A or CRL IN B that is triggering the captured logic. This feature is not implemented in the current version of the software.
- **Capture Memory** - Enables to capture the data in to the memory.
- **Capture Only** - Captures the data only. In this case. the errored bits in the data are not shown.

- **Holdoff** - Defines the minimum of data bits to capture before the trigger event occurs. This value is adjusted to a multiple of the current symbol granularity. This feature is not implemented in the current version of the software.
- **Capture Depth** - Defines the minimum of data bits to capture including holdoff. This value is adjusted to a multiple of the current symbol granularity. The capture depth is limited to the acquisition depth configured on the oscilloscope.

Show Parameters

- **View Bit Pattern As** - Displays the captured bit coed pattern in Binary, Hex or Symbol view.
- **View Coded Pattern As** - Display the coded pattern (8B/10B) in Binary, Hex or Symbol view.
- **Symbols Per Row** - Display the specified number of symbols in each row.

Once the parameters are specified, click **Start** button. The captured data will be displayed in the **Pattern Captured** pane. The errored bits are marked red as shown in the following figure:



```
Bit 0 ata                                                        20000 Bits
  0 0 0 0 0 1 1 1 1 1 0 0 0 0 1 0 0 0 1 0 1 1 0 0 0 1 0 1 1 0 1 1 1
 32 0 1 1 1 1 1 1 1 1 0 1 0 0 0 1 1 0 1 0 1 1 1 0 0 0 0 1 0 1 0 1 0
 64 0 1 1 0 0 0 1 1 0 1 0 0 1 1 0 0 1 0 0 0 1 0 1 1 1 0 0 1 0 0 0 0
 96 0 1 1 1 1 1 1 1 0 0 0 0 1 1 0 0 1 1 1 1 1 0 1 1 0 1 1 1 0 0 1 0
128 1 0 0 0 0 1 0 1 0 1 1 1 0 0 0 1 1 1 0 0 0 1 0 1 1 1 1 1 1 0 0 1
160 0 1 1 0 0 1 0 0 1 0 0 1 0 1 0 1 1 1 0 1 1 1 0 0 0 1 0 0 1 0 1 1
192 1 0 0 0 1 0 0 1 1 0 1 0 1 0 1 0 0 0 1 1 1 0 0 0 1 1 1 1 1 0 0 1
224 1 1 1 1 1 1 0 0 1 0 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 0 1 1 1 0 1
256 1 0 1 1 0 1 0 1 0 1 1 0 0 0 1 1 0 0 1 0 0 0 0 0 1 1 0 1 1 1 0 1
288 1 0 0 0 1 0 0 1 1 0 1 1 0 0 0 1 1 0 0 1 0 1 1 0 1 1 1 0 1 1 1 1
Bit: 0
```

INF1.DataIn1

Acquisition Parameters	M1
Analyzer	INF1.DataIn1
Trigger	Immediately
Slope	Rising Edge
Capture In Memory	Off
Capture Only	Off
Holdoff	0
Capture Depth	20000

Show	M1
View Coded Pattern As	Symbol
View Bit Pattern As	Bin
Symbols Per Row	32

Analyzer
Location/Location Group against which the Data Acquisition is performed.

```
:PLUGin:CCAPture:ACQuisition:ALOCation
'Pattern Capture 1','INF1.DataIn1'
```

Capture Results

Location	Capture Status	First Error	Last Error	Errors Count	Holdoff	Capture Depth
INF1.DataIn1	Finished	17	19998	9971	0	20000

NOTE

The captured patterns can be saved to file and analyzed later using the M8070EDAB error distribution analysis package.

The results of the patterns captured are also displayed in the **Capture Results Pane**. For description on the **Captures Results Pane**, refer to the **Pattern Capture** section in the *M8000 Series User Guide.*

PAM6 / PAM8 Support for Real Time Scopes

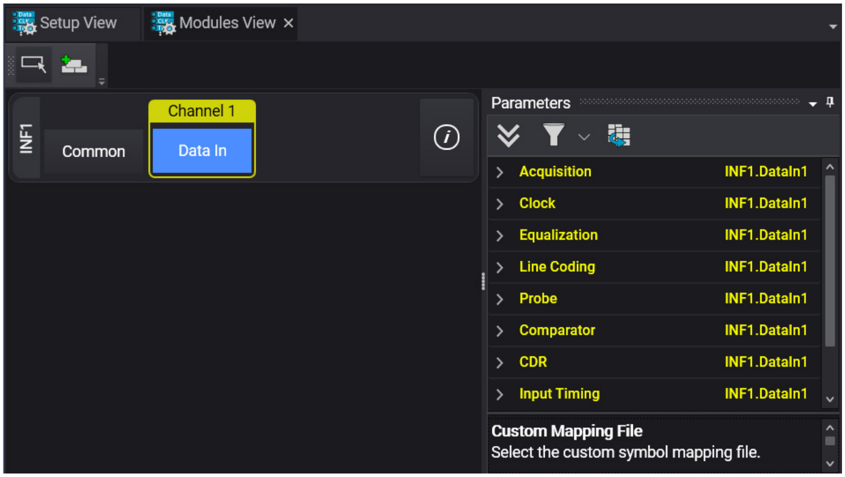
The M8070B system software (version 9.0 and above) supports real-time oscilloscopes to measure PAM6 and PAM8 line codings.

Please ensure to configure the real-time oscilloscope for usage by M8070B. Refer to the section “Controlling a Real-Time Oscilloscope from M8070B” on page 162. Once this preparation is done, simply power-on the oscilloscope and wait until it has completely started-up before launching M8070B.

Currently, the Infiniium UXR-Series Real-Time Oscilloscopes are supported (For details, see “Controlling a UXR-Series Real-Time Oscilloscope from M8070B” on page 196).

PAM6 / PAM8 Support by Module View

Once the real-time oscilloscope setup is done, you will see its entry in the Module View as shown in the following figure:



Acquisition	INF1.DataIn1
Clock	INF1.DataIn1
Equalization	INF1.DataIn1
Line Coding	INF1.DataIn1
Probe	INF1.DataIn1
Comparator	INF1.DataIn1
CDR	INF1.DataIn1
Input Timing	INF1.DataIn1

Custom Mapping File
Select the custom symbol mapping file.

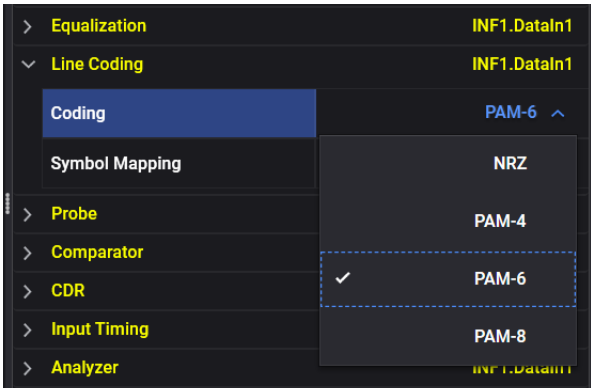
The left side of the figure shows the connected modules, and the right side shows the **Parameters** window. The corresponding real-time oscilloscope parameters can be configured through the Parameters window.

Follow the given steps to configure PAM6 or PAM8 signals:

1 Go to **Menu Bar** > **Analyzer** and select **Data In**.

2 Select **Line Coding** functional block from the **Parameters** window.

3 Select line coding options **PAM6** or **PAM8** under the '**Line Coding**' functional block.



Equalization INF1.DataIn1

Line Coding INF1.DataIn1

Coding PAM-6

Symbol Mapping

Probe

Comparator

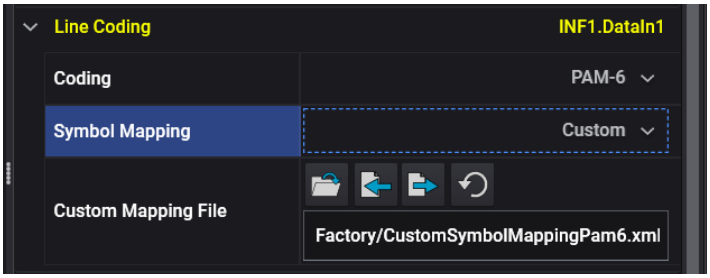
CDR

Input Timing

Analyzer INF1.DataIn1

4 Select **Symbol Mapping** mode as '**Uncoded**' or '**Custom**'.

5 If the **Symbol Mapping** mode '**Custom**' is selected you have to specify the **Custom Mapping File** for PAM6 and PAM8.



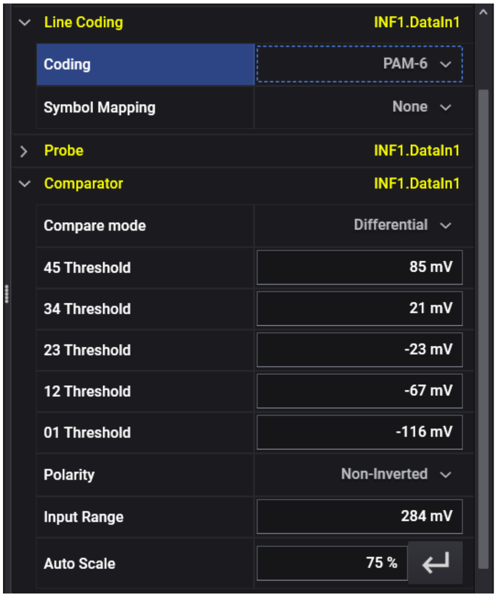
Line Coding INF1.DataIn1

Coding PAM-6

Symbol Mapping Custom

Custom Mapping File Factory/CustomSymbolMappingPam6.xml

6 For PAM6, Symbol Level 0 to 5 are visible and for PAM8 Symbol Level 0 to 7 are visible. These Symbol Level are visible under the **Comparator** functional block. The following image shows the symbol levels for PAM6:



Line Coding	INF1.DataIn1
Coding	PAM-6
Symbol Mapping	None
Probe	INF1.DataIn1
Comparator	INF1.DataIn1
Compare mode	Differential
45 Threshold	85 mV
34 Threshold	21 mV
23 Threshold	-23 mV
12 Threshold	-67 mV
01 Threshold	-116 mV
Polarity	Non-Inverted
Input Range	284 mV
Auto Scale	75 %

7 The threshold properties are also visible under the ‘**Alignment Results**’ functional block.

PAM6 / PAM8 Support by Pattern Editor

The **Pattern Editor** window allows you to allows to set the visualizations of the currently selected bit patterns in PAM6/PAM8 coding.

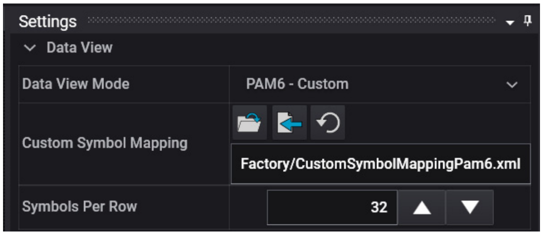
Follow the given steps to view PAM6/PAM8 coding in Pattern Editor:

1 From the Pattern Editor, go to **Settings Window** and select **Data View** functional block.

2 Select **Data View Mode** as 'PAM6 - No Coding', 'PAM6 - Custom', 'PAM8 - No Coding' or 'PAM8 - Custom coding'.

3 If the **Data View Mode** is selected as '**PAM6 - No Coding**' or '**PAM8 - No Coding**', the current pattern will be shown in PAM6 or PAM8 coding. No bit to symbol mapping is used while displaying the pattern.

4 If the **Data View Mode** is selected as '**PAM6 - Custom**' or '**PAM8 - Custom**', you have to specify the **Custom Mapping File**. The custom mapping file defines the bit to symbol mapping to be used for displaying the PAM6/PAM8 symbols. The following image shows the option to specify the custom mapping file for PAM6:



Data View Mode: PAM6 - Custom

Custom Symbol Mapping: Factory/CustomSymbolMappingPam6.xml

Symbols Per Row: 32

PAM6 / PAM8 Support by Pattern Capture

The Pattern Capture window displays the captured bit coded pattern in 'PAM6 - No Coding', 'PAM6 - Custom', 'PAM8 - No Coding' and 'PAM8 - Custom' coding.

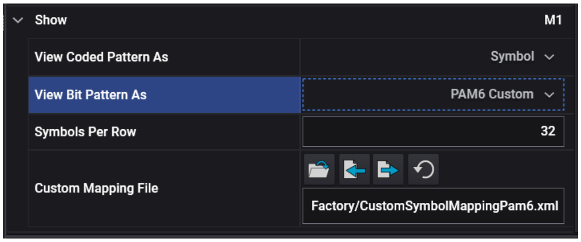
Follow the given steps to view PAM6/PAM8 coding in Pattern Capture:

1 From the **Pattern Capture**, go to **Acquisition Parameters** and select **Show** functional block.

2 Select **View Bit Pattern As** as 'PAM6 - No Coding', 'PAM6 - Custom', 'PAM8 - No Coding' or 'PAM8 - Custom coding'.

3 If the **View Bit Pattern As** is selected as '**PAM6 - No Coding**' or '**PAM8 - No Coding**', the current pattern will be shown in PAM6 or PAM8 coding. No bit to symbol mapping is used while displaying the pattern.

4 If the **View Bit Pattern As** is selected as '**PAM6 - Custom**' or '**PAM8 - Custom**', you have to specify the **Custom Mapping File**. The custom mapping file defines the bit to symbol mapping to be used for displaying the PAM6/PAM8 symbols.The following image shows the option to specify the custom mapping file for PAM6:

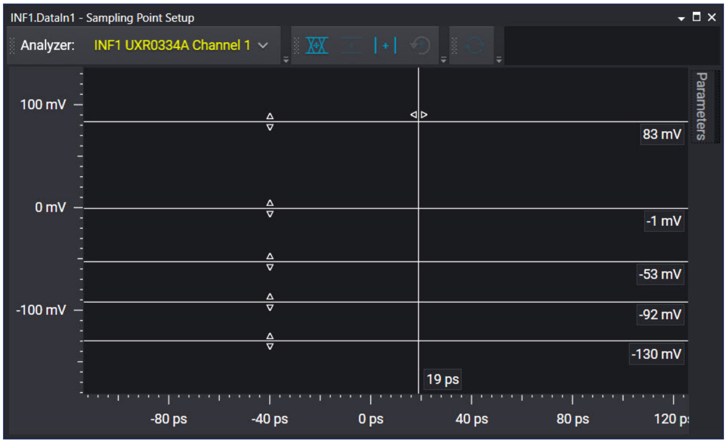


Show	M1
View Coded Pattern As	Symbol
View Bit Pattern As	PAM6 Custom
Symbols Per Row	32
Custom Mapping File	Factory/CustomSymbolMappingPam6.xml

PAM6 / PAM8 Support by Sampling Point Setup Window

In case of PAM6/PAM8 coding, the **Sampling Point Setup** window displays the PAM6 and PAM8 levels and then allows you to manually adjust the location of the sampling point.

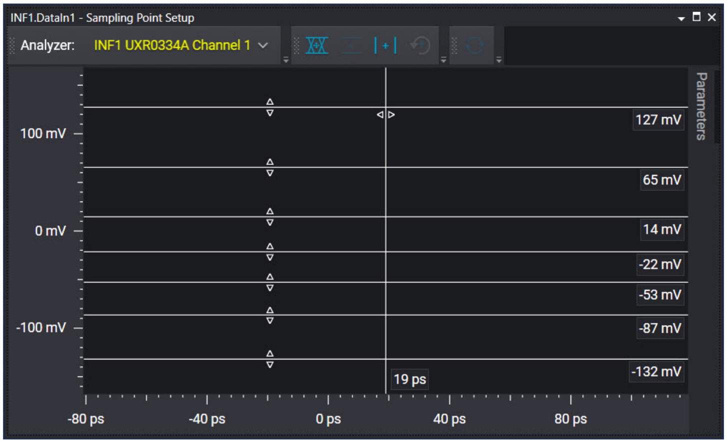
The following image shows the **Sampling Point Setup** window showing PAM6 levels:



INF1.DataIn1 - Sampling Point Setup

Analyzer: INF1 UXR0334A Channel 1

The following image shows the **Sampling Point Setup** window showing PAM8 levels:



INF1.DataIn1 - Sampling Point Setup

Analyzer: INF1 UXR0334A Channel 1

Controlling a UXR-Series Real-Time Oscilloscope from M8070B

The Advanced Measurement Package allows the UXR-Series Real-Time Oscilloscope to be controlled from M8070B System Software. It provides the following features:

- Provide support for the following UXR-Series models
  - UXR0334A/B 33GHz / 128GSa/s
  - UXR0402A/B 40GHz / 256GSa/s
  - UXR0404A/B 40GHz / 256GSa/s
  - UXR0502A/B 50GHz / 256GSa/s
  - UXR0504A/B 50GHz / 256GSa/s
  - UXR0592A/B 59GHz / 256GSa/s
  - UXR0594A/B 59GHz / 256GSa/s
  - UXR0592A/BP 59GHz / 256GSa/s
  - UXR0594A/BP 59GHz / 256GSa/s
  - UXR0702A/B 70GHz / 256GSa/s
  - UXR0704A/B 70GHz / 256GSa/s
  - UXR0702A/BP 70GHz / 256GSa/s
  - UXR0704A/BP 70GHz / 256GSa/s
  - UXR0802A/B 80GHz / 256GSa/s
  - UXR0804A/B 80GHz / 256GSa/s
  - UXR1002A/B 100GHz / 256GSa/s
  - UXR1004A/B 100GHz / 256GSa/s
  - UXR1102A/B 110GHz / 256GSa/s
  - UXR1104A/B 110GHz / 256GSa/s
- Provide support of DSAZ334A with after-market hardware upgrade to 63 GHz
- Provide support of CTLE
- Provide benefit from the hardware acceleration provided in UXR-Series starting with Infiniium 11.4 Patch 2

NOTE

As a pre-requisite for controlling the UXR from M8070B, please ensure to use a controlling PC with a native LAN connector for wired LAN connection.

Supporting the UXR-Series Senior Models

All of the UXR-Series models do have a sampling rate of 256 GSa/s with a maximum bandwidth as given by the product number. They also vary in the front-end and therefore allowed input voltage range.

NOTE

M8070B expects differential signals to be connected on Channel 1 and 3 on a four channel oscilloscope model.

If this is not the case, configure the differential mode to from "Every other channel" "Adjacent channels" directly on the oscilloscope. This setting is not part of the oscilloscope's instrument setting and is not being reset when setting the scope back to the default setting.

Support of DSAZ334A with after-market hardware upgrade to 63GHz

The Z-Series oscilloscopes offer an after-market hardware upgrade for the DSA0334A and DSO0334A oscilloscopes to support the 63 GHz bandwidth. This hardware upgrade includes adding the RealEdge inputs which are mandatory for usage with M8070B.

The upgrade effectively turns the 33 GHz oscilloscope into the 63 GHz oscilloscope which is supported by M8070B. The required change is to accept the model number of the 33 GHz scope, if the scope is also reporting the option that is indicating the hardware upgrade.

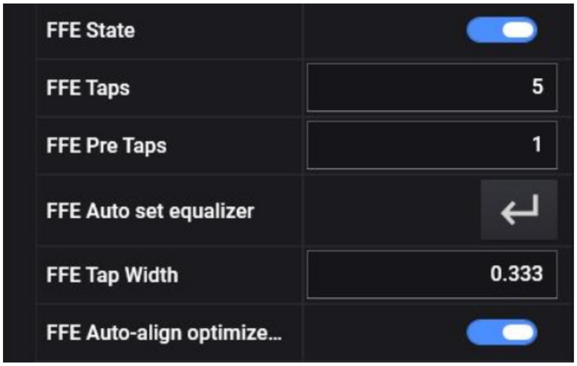
FFE, CTLE and DFE Equalization

In addition to the FFE and CTLE parameters, the DFE parameters can be fully accessed from within M8070B. The scope does not provide an auto-set function for the CTLE. You can use the following equalization options:

- CTLE
- FFE
- DFE
- CTLE + FFE
- CTLE + DFE
- FFE + DFE

FFE Parameters

This features configures a FFE (Feed-Forward Equalization) on a data input. It is available for real-time oscilloscope based error detector channels. It requires the existence for the Equalization license (option DEQ) on the real-time oscilloscope. If the license is not available, then equalization will not be offered in M8070B for the data input.



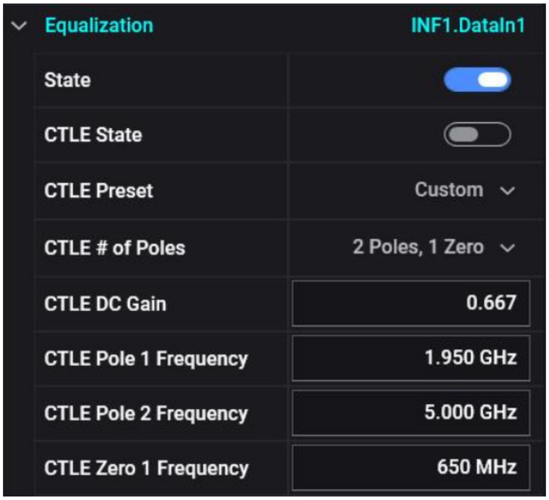
FFE State	●
FFE Taps	5
FFE Pre Taps	1
FFE Auto set equalizer	↵
FFE Tap Width	0.333
FFE Auto-align optimize...	●

It includes the following parameters:

- FEC State - Enables or disables use of the FFE algorithm on the given data input.
- FFE Taps - Sets the number of taps to be used in the FFE algorithm.
- FFE Pre Taps - Sets the number of precursor taps to be used in the FFE algorithm.
- FFE Auto set equalizer - Starts the FFE tap optimization. Ensure to first specify the number of taps.
- FFE Tap Width - The tap width sets the Eye Width field for the FFE tap optimization. Setting the width to 0.0 means the optimization is only preformed at the location of the clock. Setting the width to 1.0 means the entire acquisition is used in the optimization. The default value for FFE is 0.33. For more information on this parameter, refer to the *Infiniium Serial Data Equalization User's Guide*.
- FFE Auto-Align Optimizes EQ - Controls if the FFE tap optimization is done during a full sample point alignment. If this is set to OFF, then the tap coefficients must be set manually, or by clicking the “Auto set equalizer” button. FFE tap optimization is part of the full sample point optimization by default. Use this command to disable tap optimization in scenarios where the input signal is too distorted to produce reliable tap optimization results.

CTLE Parameters

The CTLE (Constant Time Linear Equalizer) parameters configures the use of a real-time oscilloscope that is used as an error detector in M8070B.



Equalization INF1.DataIn1

State

CTLE State

CTLE Preset Custom

CTLE # of Poles 2 Poles, 1 Zero

CTLE DC Gain 0.667

CTLE Pole 1 Frequency 1.950 GHz

CTLE Pole 2 Frequency 5.000 GHz

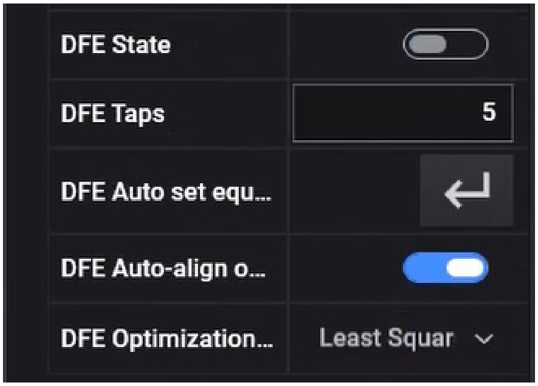
CTLE Zero 1 Frequency 650 MHz

It includes the following parameters:

- State - This is a global equalization state to enable/disable equalization. For CTLE, FFE, and DFE to work appropriately, the global equalization state must be enabled.
- CTLE State - Enables or disables use of the CTLE algorithm on the given data input.
- CTLE Preset - Selects a predefined setting of the Continuous Time Linear Equalizer (CTLE).
- CTLE Number of Poles - Selects either a 2 Pole 1 Zero, 3 Pole 1 Zero, 3 Pole 2 Zeros, 2 Poles and AC gain (for6 USB 3.1), 4 Pole 1 Zero, 6 Pole 2 Zeros, 2 Poles and 2 gains Continuous Time Linear Equalization (CTLE).
- CTLE DC Gain - Sets the DC Gain parameter for the Continuous Time Linear Equalization.
- CTLE Pole 1 Frequency - Sets the Pole 1 frequency for the Continuous Time Linear Equalization.
- CTLE Pole 2 Frequency - Sets the Pole 2 frequency for the Continuous Time Linear Equalization.
- CTLE Zero 1 Frequency - Sets first zero frequency for the 2-pole and 3-pole Continuous Time Linear Equalization.

DFE Parameters

This features configures a DFE (Decision Feedback Equalization) on a data input. It is available for real-time oscilloscope based error detector channels.



DFE State	○
DFE Taps	5
DFE Auto set equ...	↵
DFE Auto-align o...	●
DFE Optimization...	Least Squar ⌄

It includes the following parameters:

- DFE State - Enables or disables use of the DFE algorithm on the given data input.
- DFE Taps - Sets the number of taps to be used in the DFE algorithm.
- DFE Auto set equalizer - Starts the DFE tap optimization. Ensure to first specify the number of taps.
- DFE Auto-Align Optimizes EQ - Controls if the DFE tap optimization is done during a full sample point alignment. If this is set to OFF, then the tap coefficients must be set manually, or by clicking the “Auto set equalizer” button. DFE tap optimization is part of the full sample point optimization by default. Use this command to disable tap optimization in scenarios where the input signal is too distorted to produce reliable tap optimization results.
- DFE Optimization - Sets the DFE algorithms. The choices are LSEF (Least Square Error Fit) & PMAX Normalized based on pulse response (which is available only for repeating patterns).

Supporting the N7005A Optical-to-Electrical Converter

The M8070B software supports N7005A optical-to-electrical converter connected with real-time scope as an error detector.

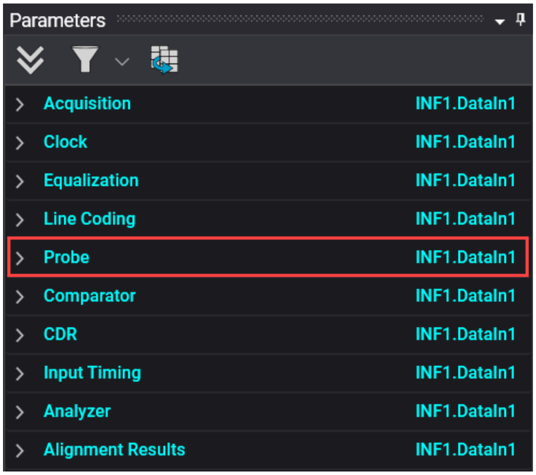
This optical-to-electrical converter directly converts optical signals into electrical signals for convenient analysis on a compatible Infiniium real-time oscilloscope.

NOTE

The minimum supported Infiniium version for this feature is 11.20.XX.

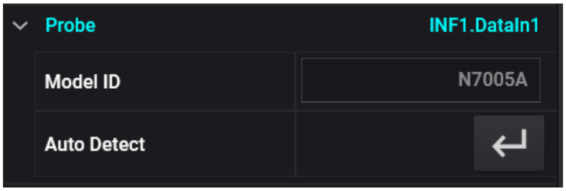
Infiniium UXR oscilloscope with bandwidth ≥ 40 GHz is supported.

The functional block **Probe** is available to support the Optical-to Electrical converter through M8070B.



Parameters

Acquisition	INF1.DataIn1
Clock	INF1.DataIn1
Equalization	INF1.DataIn1
Line Coding	INF1.DataIn1
Probe	INF1.DataIn1
Comparator	INF1.DataIn1
CDR	INF1.DataIn1
Input Timing	INF1.DataIn1
Analyzer	INF1.DataIn1
Alignment Results	INF1.DataIn1



Probe INF1.DataIn1

Model ID	N7005A
Auto Detect	↵

The following parameters are available for **Probe**:

- **Model ID**: This parameter displays the model ID of the probe connected. Currently, only N7005A probe is supported.
- **Auto Detect**: Click the **Auto Detect** ↵ button, M8070B will detect the connection change and update all the parameters accordingly.

NOTE

If the probe is connected before starting the M8070B software, then the probe will be detected automatically on startup, and all the parameters will be updated accordingly.

If the probe connection is changed in between, then M8070B will not detect the probe automatically. You have to manually click the Auto Detect ↵ button under the Probe functional block in the parameter window.

The details of N7005A Optical-to-Electrical Converter can be found at:

https://www.keysight.com/in/en/assets/9018-70016/user-manuals/9018-70016.pdf

Controlling 86108B/N1046A/N1060A/N1094A/N1094B/N1092C/N1092E from M8070B

The Auto De-emphasis feature optimizes the de-emphasis settings at the end of the cable to get the best eye performance from the instrument. Additionally, this feature allows you to de-embed/embed cable or fixture using an s-parameter file.

Prerequisites

The following are the prerequisites to use the Auto De-emphasis feature:

- M8045A module connected with the M8057A/B remote head module
- M8042A module connected with the M8058A/M8059A remote head module
- Following DCA models:
  - 86100D Infiniium DCA-X Wide-Bandwidth Oscilloscope. The 86100D Infiniium Oscilloscope should be equipped with 86108B Precision Waveform Analyzer plug-in.
  - N1000A DCA-X Wide-Bandwidth Oscilloscope Mainframe equipped with N1060A 50/85 GHz 64 GBd Precision Waveform Analyzer.
  - N1094A/B 2/4 Channel Electrical DCA-M Oscilloscope.
  - N1092C DCA-M Sampling Oscilloscope (One Optical and Two Electrical Channels)
  - N1092E DCA-M Sampling Oscilloscope (Two Optical and Two Electrical Channels)
  - N1046A 75/85/100 GHz, 1/2/4 Port, Electrical Remote Sampling Head Module
  - N1060A 50/85 GHz Precision Waveform Analyzer
  - The minimum FlexDCA revision required is A.05.80.158.

Required options (refer the tables below):

Table 23 Oscilloscope Configuration (86100D DCA-X Mainframe with 86108B)

Mainframe Model No.	Mainframe Hardware Options	Plug-In Model No.	Plug-In Model Option	Max# of Modules/Diff Channels
86100D	Required: ETR	86108B	HBW, 232, 300, 400, PTB	1/1

Table 24 Oscilloscope Configuration (N1000A DCA-X Mainframe with N1060A)

Mainframe Model No.	Mainframe Hardware Options	Plug-In Model No.	Plug-In Model Option	Max# of Modules/Diff Channels
N1000A	Required: PLK	N1060A	085, 264, PTB	1/1

Table 25 Oscilloscope Configuration (DCA-M)

DCA-M Model No.	DCA-M Options	# of Diff Channels
N1094A	050, FS1, LOJ, PLK	1
N1094B	050, FS1, LOJ, PLK	2
N1092C	050, FS1, LOJ, PLK	2
N1092E	050, FS1, LOJ, PLK	2

Table 26 Flex DCA Software Options/Package

Choose Software Option OR	
201	Equalizers (CLTE, DFE, LFE)
SIM	Waveform Transformations (embedding/de-embedding)
9FP	PAM-N Analysis
Choose Software Package	
N1010100A	Research and Development Package for FlexDCA

One Time Preparation

In order to make DCA-X/DCA-M accessible in M8070B, prepare the setup as follows.

1 Install FlexDCA N1000-Series System Software on the computer that the DCA-X/DCA-M is connected to via USB. In most cases this will be the same computer that executes M8070B.

2 Follow the steps 2 - 3 from “Controlling N1076A/77A from M8070B” “One Time Preparation” .

NOTE

Use alias name as N1076A_PROXY or N1010A_PROXY.

Only one alias must be used in the connection expert for the same Visa address.

Using FlexDCA from within M8070B

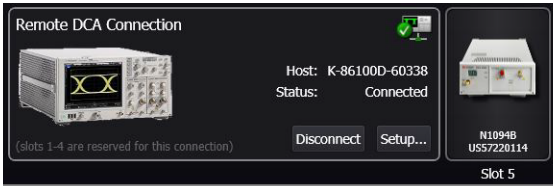
In order to use DCA-X/DCA-M from within M8070B, the FlexDCA N1000-Series System Software has to be started and fully running before starting M8070B.

To ensure this, follow the following startup sequence.

1 Power-up all DCA-X/DCA-M that are connected to the PC.

2 Start FlexDCA and wait until the software has completely started up.

3 Verify that the DCA-X/DCA-M are correctly assigned to the Extended Modules Slots and that they are connected within FlexDCA. They have to show up at the bottom of the FlexDCA window.



Remote DCA Connection

Host: K-86100D-60338

Status: Connected

4 Start M8070B System Software.

5 M8070B will show a dedicated module for the FlexDCA instrument.

   a Press the configure button to enable individual slots of FlexDCA to be controlled from M8070B.

   b The enabled slots will show up on the module. Click on the respective button to access the Deemphasis parameters.

For details on Deemphasis parameters, refer to *M8000 Series User Guide* or *M8000 Series Online Help*.

5 Remote Programming

This chapter describes the remote programming for measurements provided by Advanced Measurement Package and the instruments controlled by the M8070B system software.

Remote Programming Basics

SCPI Command Language - Introduction

The Serial BERT is compatible with the standard language for remote control of instruments. Standard Commands for Programmable Instruments (SCPI) is the universal programming language for instrument control.

SCPI can be subdivided into the following command sets:

- SCPI Common Commands
- SCPI Instrument Control Commands
- IEEE 488.2 Mandatory Commands

SCPI Common Commands

This is a common command set. It is compatible with IEEE 488.2 and contains general housekeeping commands. The common commands are always preceded by an asterisk. A typical example is the reset command:

*RST

The IEEE 488.2 command set also contains query commands. Query commands always end with a question mark.

SCPI Instrument Control Commands

The programming commands are compatible with the Standard Commands for Programmable Instruments (SCPI) standard. For more detailed information regarding the GPIB, the IEEE 488.2 standard, or the SCPI standard, refer to the following books:

- SCPI Consortium. SCPI- Standard Commands for Programmable Instruments, 1997 (http://www.scpiconsortium.org).
- International Institute of Electrical and Electronics Engineers. IEEE Standard 488.1- 1987, IEEE Standard Digital Interface for Programmable Instrumentation. New York, NY, 1987.
- International Institute of Electrical and Electronics Engineers. IEEE Standard 488.2- 1987, IEEE Standard Codes, Formats, Protocols and Common commands For Use with ANSI/IEEE Std 488.1- 1987. New York, NY, 1987.

IEEE 488.2 Mandatory Commands

In order to comply with the SCPI model as described in IEEE 488.2, the Serial BERT implements certain mandatory commands. Other commands are implemented optionally.

Overlapped and Sequential Commands

IEEE 488.2 defines the distinction between overlapped and sequential commands. A sequential command is one which finishes executing before the next command starts executing. An overlapped command is one which does not finish executing before the next command starts executing.

NOTE **It is not reliable to use wait statements in the control program to facilitate the use of overlapped commands.**

Because these commands may allow the execution of more than one command at a time, special programming techniques must be used to ensure valid results. The common commands *OPC, *WAI and *OPC? can be used for this purpose. They help synchronize a device controller with the execution of overlapped commands.

The behaviors of these commands, in brief, are as follows:

- *OPC
  The *OPC command sets the Operation Complete (OPC) bit of the Event Register when the No Operation Pending flag is TRUE (No Operation Pending flag is attached to each overlapped command). Until that time, the controller may continue to parse and execute previous commands. It is good technique, then, to periodically poll the OPC bit to determine if the overlapped command has completed.
- *WAI
  The *WAI command allows no further execution of commands or queries until the No Operation Pending flag is true, or receipt of a Device Clear (dcas) message, or a power on.
  The *WAI command can be used for overlapped commands. It stops the program execution until any pending overlapped commands have finished. Specifically, it waits until the No Operation Pending flag is TRUE, or receipt of a dcas message, or a power on.

- *OPC?
  The *OPC? query returns the ASCII character "1" in the Output Queue when the No Operation Pending flag is TRUE. At the same time, it also sets the Message Available (MAV) bit in the Status Byte Register. The *OPC? will not allow further execution of commands or queries until the No Operation Pending flag is true, or receipt of a Device Clear (dcas) message, or a power on.

NOTE

The command behaviors described above are for overlapped commands. When the same commands are used with sequential commands, the behaviors may be different.

Data Types

The M8020A/M8040A/M8050A has the capability of receiving and returning data in the following formats:

STRING

A string of human-readable ASCII characters, either quoted or nonquoted.

NUMERIC

The M8020A/M8040A/M8050A handles the following numeric formats:

- <NR1>: Integer (0, 1, 2, - 1, etc.)
- <NR2>: Number with an embedded decimal point (0.1, 0.001. 3.3, etc.)
- <NR3>: Number with an embedded decimal point and exponent (1e33, 1.3e- 12, etc.)
- <NRf>: Represents <NR1>, <NR2> and <NR3>
- Binary preceded by #b (#B010101, #b011111, etc.)
- Octal preceded by #q (#Q777111, #q7331777, etc.)
- Hex preceded by #h (#haff, #h8989fffff, etc.)

BOOLEAN

Boolean values can be sent to the M8020A/M8040A/M8050A as either ON | OFF or 0 | 1. The M8020A/M8040A/M8050A answers queries with 0 | 1.

Definite Length Arbitrary Block Data

Block data is used when a large quantity of related data is being returned. A definite length block is suitable for sending blocks of 8-bit binary information when the length is known beforehand. An indefinite length block is suitable for sending blocks of 8-bit binary information when the length is not known beforehand or when computing the length beforehand is undesirable.

It has the following format:

#<Length of length><Length of data><data>

<Length of length> is a single integer that contains the number of digits in <Length of data>, which in turn contains the length of the data. For example, a 512-byte pattern would be defined as:

#3512<data>

Important Points about SCPI

There are a number of key areas to consider when using SCPI for the first time. These are as follows:

- Instrument Model
- Command Syntax
- Optional Parts of Commands
- Sending Commands
- Command Separators
- SCPI Command Structure

Instrument Model

SCPI guidelines require that the M8020A/M8040A/M8050A is compatible with an instrument model. This ensures that when using SCPI, functional compatibility is achieved between instruments that perform the same tasks. For example, if two different instruments have a programmable clock frequency setting, then both instruments would use the same SCPI commands to set their frequency. The instrument model is made up of a number of subsystems.

The sub-system defines a group of functions within a module and has a unique identifier under SCPI, which is called the Root Keyword.

Command Syntax

Commands may be up to twelve characters long. A short-form version is also available which has a preferred length of four characters or less. In this document the long-form and short-form versions are shown as a single word with the short-form being shown in upper-case letters.

For example, the long-form node command SOURce has the short-form SOUR. Using the short form saves time when entering a program; however, using the long form makes a program more descriptive and easier to understand.

SCPI commands may be commands only, commands and queries, or queries only. A question mark at the end of a command indicates that it is a query. If the question mark appears in brackets ([?]), the command has a command and query form.

Optional Command Keywords

Some layers in the SCPI command structure are optional. These optional keywords are indicated by square brackets ([]). A typical use for these types of keywords is with a command that is unique to one module. In this case, the top layer (Root Keyword) of the command structure may be omitted.

For example, the following command code segments are functionally identical:

```
[:SOURce]:JITTer[:GLOBal][:STATe] <ON|OFF|1|0>
```

```
:JITTer <ON|OFF|1|0>
```

```
:JITT <ON|OFF|1|0>
```

```
:jitt <ON|OFF|1|0>
```

Note that it is not necessary to include the syntax inside the square brackets ([]).

Query Responses

It is possible to interrogate the individual settings and status of a device using query commands. Retrieving data is a two-stage operation.

The query command is sent from the controller using the OUTPUT statement and the data is read from the device using the ENTER statement. A typical example is the SCPI IEEE 488.2 Common Command *IDN? which queries the identity of a device.

NOTE

When sending strings to the instrument, either the double quote (") or the single quote may be used ('), the latter being more suited to PASCAL programs, which make use of a single quote; the former being more suited to use in BASIC programs, which use a double quote as a delimiter.

Command Separators

The SCPI command structure is hierarchical and is governed by commas, semicolons and colons:

- Commas are used to separate parameters in one command.
- Colons are used to separate levels.
- Semicolons are used to send more than one command to the instrument at a time.

It is possible to send several commands in one pass, as long as the commands all belong to the same node in the SCPI tree. The commands have to be separated by semicolons.

The following SCPI commands provide examples of this.

SOURce:VOLTage:OFFSet 'M2.DataOut2',-0.99

SOURce:VOLTage:AMPLitude 'M2.DataOut2',1.11

These commands can also be sent as follows:

VOLT:OFFS 'M2.DataOut2',-0.99; 'M2.DataOut2',AMPL 1.11

SCPI Command Structure Example

The SCPI command structure can be best examined by means of an example. For example, the command to set the pattern generator's output amplitude is:

[:SOURce]:VOLTage[:AMPLitude] 'M1.DataOut1',1.11

The structure of this command can be illustrated as follows:

[:SOURce]	This is the top layer of the command structure and identifies the source subsystem.
:VOLTage	This is the next layer and defines the subnode for setting a voltage level.
[:AMPLitude]	This is the command itself for setting the output amplitude level.
'M1.DataOut1',1.11	This specifies pattern generator 1, channel 1 and specifies an amplitude of 1.11.

NOTE

Any optional commands are enclosed in square brackets [ ] and any optional characters are shown in lower case.

A colon indicates a change of level in the command hierarchy. Commands at the same level in the hierarchy may be included in the same command line, if separated by a semi-colon.

The bar symbol (|) indicates mutually exclusive commands.

To translate this syntax into a command line, follow the convention described above. Remember, however, that the command line can be created in several different ways. It can be created with or without optional keywords and in a long or short form. The following example gives three possible forms of the command line; all are acceptable.

In long form:

:SOURce:VOLTage:AMPLitude 'M1.DataOut1',1.11

In short form:

:SOUR:VOLT:AMPL 'M1.DataOut1',1.11

With the optional commands removed:

:VOLT 'M1.DataOut1',1.11

The long form is the most descriptive form of programming commands in SCPI.

Sending Commands to the Instrument

A command is invalid and will be rejected if:

- It contains a syntax error.
- It cannot be identified.
- It has too few or too many parameters.
- A parameter is out of range.
- It is out of context.

Sending Commands using VISA

The following is a list of the available hardware interfaces for sending commands to the M8020A/M8040A/M8050A firmware:

SCPI Access (HiSLIP): TCPIP0::localhost::hislip0::INSTR (High-Speed LAN Instrument Protocol)

SCPI Access (VXI-11): TCPIP0::localhost::inst0::INSTR (VXI-11 is a TCP/IP instrument protocol defined by the VXIbus Consortium)

NOTE

If you use the VXI-11 (TCP/IP instrument protocol) in your test programs, you must change the resource string to the HiSLIP protocol if the software is running on Windows 8 or Windows 8.1. VXI-11 is not supported at this time on Windows 8 or Windows 8.1.

For example:

“TCPIP0::192.17.34.0::inst0::INSTR” -> “TCPIP0::192.17.34.0::hislip0::INSTR”

SCPI Access (Socket): TCPIP0::localhost::5025::SOCKET (Standard SCPI-over-sockets port)

SCPI Access (Telnet): telnet localhost 5024 (Communication with LAN instrument through SCPI Telnet port).

Command Line Arguments

(See “Communication" on page 217 for details about /Socket, /Telnet, /Inst, /AutoID, /NoAutoID, /FallBack).

Table 27 Command line arguments

Argument	Description
/AutoId	Start in auto ID mode (this is the default) [optional]
/NoAutoId	Do not start in auto ID mode - use communication parameters from command line [optional]
/FallBack	Use auto ID mode if the communication parameters from the command line don't work [optional]
/Socket <socket port>	Set the socket port for the SCPI communication (only used with /NoAutoId) [optional]
/Telnet <telnet port>	Set the telnet port for the SCPI communication (only used with /NoAutoId) [optional]
/Inst <instrument number>	Set the instrument number for VXI-11.3 and HiSLIP SCPI communication [optional]
/rcl	Recall last used setting [optional]
/rst	Reset to factory default [optional]
/IgnoreAwg	M8070B software don't grab M8195A modules [optional]

Communication

Depending on the command line arguments /Socket, /Telnet, /Inst, /AutoID, /NoAutoID, /FallBack, the M8070B software starts several servers to handle SCPI commands. (Refer to the table above.)

/Socket, /Telnet, /Inst: If -1, don't start the respective servers

“Defaults:

Socket port: 5025 (e.g. TCPIP0::localhost::5025::SOCKET)

Telnet port: 5024

HiSLIP, VXI-11.3: 0 (e.g. TCPIP0::localhost::hislip0::INSTR, TCPIP0::localhost::inst0::INSTR)

/FallBack: If starting a server fails because of a conflict, try using another port or number

"HiSLIP, VXI-11.3: increase the index until a server can be started successfully

"Socket, Telnet: start with port 60000, then increase it until the servers can be started successfully. If neither socket nor telnet is disabled the M8070B software tries to start the servers on two consecutive ports

(socket port = telnet port + 1)

/AutoID: Automatically select ports and number for the connections, which are unique per instrument.

"This is the default behavior; it is not necessary to specify this argument on the command line.

"/Socket, /Telnet, /Inst are ignored (unless they are -1 and a server is disabled)

"If the M8070B software detects more than one AXIe module, use a special mechanism to obtain a number for the HiSLIP and VXI-11.3 servers, which makes sure that the M8070B software always uses the same VISA resource string per module

"The socket and telnet port are then calculated from the HiSLIP index:

telnet port = 60000 + 2 * <HiSLIP index>

socket port = 60000 + 2 * <HiSLIP index> + 1

Note: Ports may already be in use by Windows or other applications, so they are not available for M8195A.

/NoAutoID: Do not automatically select ports and number for the connections, use the values specified with /Socket, /Telnet, /Inst or their respective default values instead.

If both /NoAutoID and /AutoID are specified, /AutoID overrides /NoAutoID.

NOTE

The first port not assigned by IANA is 49152 (IANA, Internet Assigned Numbers Authority, http://www.iana.org).

Advanced Measurement Package PLUGin Subsystem

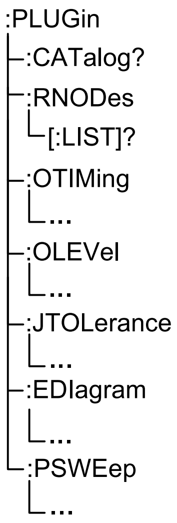
The M8020A/M8040A/M8050A platform supports a plugin interface used to implement certain interfaces recognized by the software and integrated into the M8070B GUI and instrument software. Output Timing, Jitter Tolerance, Output Level, Eye Diagram and Parameter Sweep measurements are examples of the plugin concept.

The Advanced Measurement Package provides the following measurements when it is installed on the M8070B system software:

- Output timing measurement
- Output level measurement
- Jitter tolerance measurement
- Eye diagram measurement
- Parameter sweep

NOTE **The Advanced Measurement Package requires a valid license for activation.**

This subsystem has the following SCPI structure:



```
:PLUGin
├:CATalog?
├:RNODes
│└[:LIST]?
├:OTIMing
│└...
├:OLEVel
│└...
├:JTOLerance
│└...
├:EDIagram
│└...
└:PSWEep
 └...
```

This subsystem has the following subnodes:

Table 28

Name	Description under
:CATalog?	:PLUGin:CATalog? on page 220
:RNODes[:LIST]?	:PLUGin:RNODes[:LIST]? on page 220
:OTIMing	:PLUGin:OTIMing Subnode on page 221
:OLEVel	:PLUGin:OLEVel Subnode on page 253
:JTOLerance	:PLUGin:JTOLerance Subnode on page 287
:EDIagram	:PLUGin:EDIagram Subnode on page 318
:PSWEep	:PLUGin:PSWEep Subnode on page 344

:PLUGin:CATalog?

Syntax :PLUGin:CATalog?

Description This query returns the names of all the plug-ins.

Example :PLUGin:CATalog?

"Pattern Capture","Script Editor","DUT Control Interface","Error Ratio","Jitter Tolerance","Jitter Tolerance Template Editor","Output Level","Output Timing","C-Phy Frame Generator","D-Phy Frame Generator"

:PLUGin:RNODes[:LIST]?

Syntax :PLUGin:RNODes[:LIST]?

Description This query returns a list of SCPI plugin root nodes that can be used to address different type of measurements (plugins).

Example :PLUG:RNOD:LIST?
"CCAPture","ERATio","JTOLerance","OLEVel","OTIMing","LTXGKR","EDIagram","PSWEep","OCEI","CPHYplugin"

:PLUGin:OTIMing Subnode

This is the basic measurement for BER versus the sampling point delay, which is displayed as a 'Bathtub' curve and is integrated with the M8070B software during the startup. The measurement comes with the set of acquisition and evaluation parameters which are used to configure the measurement run. The acquisition parameters can only be configured before the measurement starts (i.e., you cannot change the value of the acquisition parameter while the measurement is running); however, the evaluation parameters can be configured at any time even while the measurement is running.

This subnode has the following SCPI structure:

```
:PLUGin
└:OTIMing
  ├:ACQuisition
  │ ├:ALOCation[?]
  │ ├:DRESolution[?]
  │ ├:FTJBer[?]
  │ ├:TCLevel[?]
  │ ├:OPTimization[?]
  │ ├:COMPared[?]
  │ ├:EBEnabled[?]
  │ └:ERRored[?]
  ├:EVALuation
  │ ├:BERThresh[?]
  │ ├:MBJPartition[?]
  │ ├:ETJBer[?]
  │ └:ERRType[?]
  ├:SHOW
  │ ├:SMPoints[?]
  │ ├:UNIT[?]
  │ ├:VSCale[?]
  │ ├:SGLegends[?]
  │ └:COMment[?]
  ├:CATalog?
  ├:TABS?
  ├:EXPort
  ├:DELete
  ├:NEW
  ├:RESet
  ├:STARt
  ├:STOP
  ├:BREak
  ├:STEP
  ├:CONTinue
  ├:GRAPh?
  ├:RUN
  │ ├:HISTory
  │ │ ├[:STATe][?]
  │ │ └:CLEar
  │ ├:LOG?
  │ ├:MESSage?
  │ ├:PROGress?
  │ └[:STATus]?
  └:FETCh
    └...
```

This subnode has the following commands:

Table 29

Name	Description under
:ACQuisition:ALOCation[?]	:PLUGin:OTIMing:ACQuisition:ALOCation[?] on page 225
:ACQuisition:DRESolution[?]	:PLUGin:OTIMing:ACQuisition:DRESolution[?] on page 225
:ACQuisition:FTJBer[?]	:PLUGin:OTIMing:ACQuisition:FTJBer[?] on page 226
:ACQuisition:TCLevel[?]	:PLUGin:OTIMing:ACQuisition:TCLevel[?] on page 226
:ACQuisition:OPTimization[?]	:PLUGin:OTIMing:ACQuisition:OPTimization[?] on page 227
:ACQuisition:COMPared[?]	:PLUGin:OTIMing:ACQuisition:COMPared[?] on page 227
:ACQuisition:EBEnabled[?]	:PLUGin:OTIMing:ACQuisition:EBEnabled[?] on page 228
:ACQuisition:ERRored[?]	:PLUGin:OTIMing:ACQuisition:ERRored[?] on page 228
:EVALuation:BERThresh[?]	:PLUGin:OTIMing:EVALuation:BERThresh[?] on page 229
:EVALuation:MBJPartition[?]	:PLUGin:OTIMing:EVALuation:MBJPartition[?] on page 229
:EVALuation:ETJBer[?]	:PLUGin:OTIMing:EVALuation:ETJBer[?] on page 230
:EVALuation:ERRType[?]	:PLUGin:OTIMing:EVALuation:ERRType[?] on page 230
:SHOW:SMPoints[?]	:PLUGin:OTIMing:SHOW:SMPoints[?] on page 231
:SHOW:UNIT[?]	:PLUGin:OTIMing:SHOW:UNIT[?] on page 231
:SHOW:VSCale[?]	:PLUGin:OTIMing:SHOW:VSCale[?] on page 232
:SHOW:SGLegends[?]	:PLUGin:OTIMing:SHOW:SGLegends[?] on page 232

Name	Description under
:SHOW:COMment[?]	:PLUGin:OTIMing:SHOW:COMment[?] on page 233
:CATalog?	:PLUGin:OTIMing:CATalog? on page 233
TABS?	:PLUGin:OTIMing:TABS? on page 234
EXPort	:PLUGin:OTIMing:EXPort on page 234
:DELete	:PLUGin:OTIMing:DELete on page 235
:NEW	:PLUGin:OTIMing:NEW on page 235
:RESet	:PLUGin:OTIMing:RESet on page 236
:STARt	:PLUGin:OTIMing:STARt on page 236
:STOP	:PLUGin:OTIMing:STOP on page 236
:BREak	:PLUGin:OTIMing:BREak on page 237
:STEP	:PLUGin:OTIMing:STEP on page 237
:CONTinue	:PLUGin:OTIMing:CONTinue on page 238
:GRAPh?	:PLUGin:OTIMing:GRAPh? on page 238
:RUN:HISTory[:STATe]	:PLUGin:OTIMing:RUN:HISTory[:STATe][?] on page 239
:RUN:HISTory:CLEar	:PLUGin:OTIMing:RUN:HISTory:CLEar on page 239
:RUN:LOG?	:PLUGin:OTIMing:RUN:LOG? on page 240
:RUN:MESSage?	:PLUGin:OTIMing:RUN:MESSage? on page 240
:RUN:PROGress?	:PLUGin:OTIMing:RUN:PROGress? on page 241
:RUN[:STATus]?	:PLUGin:OTIMing:RUN[:STATus]? on page 241
:FETCh	:PLUGin:OTIMing:FETCh Subnode on page 242

:PLUGin:OTIMing:ACQuisition:ALOCation[?]

Syntax	:PLUGin:OTIMing:ACQuisition:ALOCation 'identifier', <location-string> :PLUGin:OTIMing:ACQuisition:ALOCation? 'identifier'
Input Parameters	'identifier': Specify the measurement name. <location-string>: Specify location or group name identifier.
Return Range	<location or group name>
Description	This command associates the location or group name identifier with a measurement name identifier against which the data acquisition is performed. The group name identifier is defined using the :SYSTem:INSTrument:GROup:DEFine command. The query returns the current location or group name identifier against which the measurement name identifier is configured for the data acquisition. This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.
Example	:PLUG:OTIM:ACQ:ALOC 'MyMeasurement','M1.DataIn1'

:PLUGin:OTIMing:ACQuisition:DRESolution[?]

Syntax	:PLUGin:OTIMing:ACQuisition:DRESolution 'identifier', <NRf> :PLUGin:OTIMing:ACQuisition:DRESolution? 'identifier'
Input Parameters	'identifier': Specify the measurement name. <NRf>: Set the sample delay resolution.
Return Range	1 mUI to 250 mUI
Description	This command sets the sample delay resolution in terms of UI and associates it with a specific measurement name identifier. The query returns the current sample delay resolution value in UI. This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.
Example	:PLUG:OTIM:ACQ:DRES 'MyMeasurement', 0.25

:PLUGin:OTIMing:ACQuisition:FTJBer[?]

Syntax

:PLUGin:OTIMing:ACQuisition:FTJBer 'identifier', <NRf>

:PLUGin:OTIMing:ACQuisition:FTJBer? 'identifier'

Input Parameters

'identifier': Specify the measurement name.

<NRf>: Specify the BER value for the fast total jitter measurement.

Return Range

1E-15 to 1E-9

Description

This command specifies the BER value for the fast total jitter measurement and associates it with a specific measurement name identifier. The query returns the current value at which the fast total jitter measurement was performed.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example

:PLUG:OTIM:ACQ:FTJB 'MyMeasurement', 1E-10

:PLUGin:OTIMing:ACQuisition:TCLevel[?]

Syntax

:PLUGin:OTIMing:ACQuisition:TCLevel 'identifier', <NRf>

:PLUGin:OTIMing:ACQuisition:TCLevel? 'identifier'

Input Parameters

'identifier': Specify the measurement name.

<NRf>: Specify the BER value for the fast total jitter measurement.

Return Range

0.1% to 99.9%

Description

This command sets the target confidence level when sample delay optimization is set as fast total jitter and value is in percentage for the addressed measurement name identifier. The query returns the current target confidence level.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example

:PLUG:OTIM:ACQ:TCL 'MyMeasurement', 95.1

:PLUG:OTIM:ACQ:TCL? 'MyMeasurement'

95.1

:PLUGin:OTIMing:ACQuisition:OPTimization[?]

Syntax :PLUGin:OTIMing:ACQuisition:OPTimization 'identifier', <FTJ|NONE>

:PLUGin:OTIMing:ACQuisition:OPTimization? 'identifier'

Input Parameters 'identifier': Specify the measurement name.

<FTJ|NONE>: Specify the sample delay optimization technique.

Return Range FTJ|NONE

Description This command specifies the sample delay optimization technique and associates it with a specific measurement name identifier. If NONE is selected then no optimization is performed. If FTJ is selected then the fast total jitter measurement is performed. The query returns the current value.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example :PLUG:OTIM:ACQ:OPT 'MyMeasurement', FTJ

:PLUGin:OTIMing:ACQuisition:COMPared[?]

Syntax :PLUGin:OTIMing:ACQuisition:COMPared 'identifier', <NRf>

:PLUGin:OTIMing:ACQuisition:COMPared? 'identifier'

Input Parameters 'identifier': Specify the measurement name.

<NRf>: Specify the number of compared bits.

Return Range 1E+0 to 1E+18

Description This command specifies the number of compared bits used as criteria for moving to the next measurement point when the sample delay optimization is set to NONE. This command also associates this parameter with a specific measurement name identifier. The query returns the current value.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example :PLUG:OTIM:ACQ:COMP 'MyMeasurement', 1E+9

NOTE

The BER measurement processes at least as many bits as defined by the acquisition parameters, it stops when the end criteria is reached or exceeded. Due to update period and hardware granularities, an individual BER measurement will typically process more bits than configured.

:PLUGin:OTIMing:ACQuisition:EBEnabled[?]

Syntax

:PLUGin:OTIMing:ACQuisition:EBEnabled 'identifier', <OFF|ON|0|1>

:PLUGin:OTIMing:ACQuisition:EBEnabled? 'identifier'

Input Parameters

'identifier': Specify the measurement name.

<OFF|ON|0|1>: Disable/enable errored bits as criteria to move to the next measurement point.

Return Range

0|1

Description

This command enables/disables the number of errored bits. This command also associates this parameter with a specific measurement name identifier. If enabled (1 or ON), then errored bits will be used along with the number of compared bits as the criteria for moving to the next measurement point when the sample delay optimization is set to NONE. If disabled (0 or OFF), then only the number of compared bits will be used as the criteria for moving to the next measurement point while running the measurement. The query returns the current value.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example

:PLUG:OTIM:ACQ:EBE 'MyMeasurement', 1

:PLUGin:OTIMing:ACQuisition:ERRored[?]

Syntax

:PLUGin:OTIMing:ACQuisition:ERRored 'identifier', <NRf>

:PLUGin:OTIMing:ACQuisition:ERRored? 'identifier'

Input Parameters

'identifier': Specify the measurement name.

<NRf>: Specify the number of errored bits.

Return Range

1E+0 to 1E+18

Description This command specifies the number of errored bits. This command also associates this parameter with a specific measurement name identifier. This command can only be used when the :PLUGin:OTIMing:ACQuisition:EBEnabled command is set to 1 (enabled) and the sample delay optimization is set to NONE. This is the criteria for moving to the next measurement point while running the measurement. The query returns the current value.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example :PLUG:OTIM:ACQ:ERR 'MyMeasurement', 1E+4

:PLUGin:OTIMing:EVALuation:BERThresh[?]

Syntax :PLUGin:OTIMing:EVALuation:BERThresh 'identifier', <NRf>

:PLUGin:OTIMing:EVALuation:BERThresh? 'identifier'

Input Parameters 'identifier': Specify the measurement name.

<NRf>: Specify the BER threshold value.

Return Range 1E-15 to 1E-1

Description This command specifies the BER threshold value used to calculate the measurement parameters and associates it with a specific measurement name identifier. The query returns the current BER threshold value.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example :PLUG:OTIM:EVAL:BERT 'MyMeasurement', 1E-10

:PLUGin:OTIMing:EVALuation:MBJPartition[?]

Syntax :PLUGin:OTIMing:EVALuation:MBJPartition 'identifier', <NRf>

:PLUGin:OTIMing:EVALuation:MBJPartition? 'identifier'

Input Parameters 'identifier': Specify the measurement name.

<NRf>: Specify the minimum BER for separating random and deterministic jitter components.

Return Range 1E-18 to 1E+0

Description This command specifies the minimum BER value for separating the random and deterministic jitter components and associates it with a specific measurement name identifier. In order to separate the random and deterministic jitter from the bathtub, the specified BER threshold value must be greater than the value specified by this command. The range is 1E-18 to 1E+0. The query returns the minimum value specified to separate the random and deterministic jitter components. The default is 1E-9.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example :PLUG:OTIM:EVAL:MBJP 'MyMeasurement', 1E-7

:PLUGin:OTIMing:EVALuation:ETJBer[?]

Syntax :PLUGin:OTIMing:EVALuation:ETJBer 'identifier', <EM6|...|EM21>

:PLUGin:OTIMing:EVALuation:ETJBer? 'identifier'

Input Parameters 'identifier': Specify the measurement name.

<EM6|...|EM21>: Specify the residual BER for total jitter estimation calculation.

Return Range EM6 to EM21

Description This command specifies the residual BER for total jitter estimation calculation and associates it with a specific measurement name identifier. The range is 1E-21 (EM21) to 1E-6 (EM6). The query returns the current residual BER for total jitter estimation calculation. The default is 1E-12.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example :PLUG:OTIM:EVAL:ETJB 'MyMeasurement', EM7

:PLUGin:OTIMing:EVALuation:ERRType[?]

Syntax :PLUGin:OTIMing:EVALuation:ERRType 'identifier', <AERR|ZERR|OERR>

:PLUGin:OTIMing:EVALuation:ERRType? 'identifier'

Input Parameters 'identifier': Specify the measurement name.

<AERR|ZERR|OERR>: Specify the error type to evaluate.

Return Range AERR|ZERR|OERR

Description This command specifies the error type to evaluate and associates it with a specific measurement name identifier. AERR allows evaluation of all errors. ZERR allows evaluation of errored zeroes and OERR allows evaluation of errored ones. The query returns the current value.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example :PLUG:OTIM:EVAL:ERRT 'MyMeasurement', ZERR

:PLUGin:OTIMing:SHOW:SMPoints[?]

Syntax :PLUGin:OTIMing:SHOW:SMPoints 'identifier', <OFF|ON|0|1>

:PLUGin:OTIMing:SHOW:SMPoints? 'identifier'

Input Parameters 'identifier': Specify the measurement name.

<OFF|ON|0|1>: Disable/enable viewing the measurement points.

Return Range 0|1

Description This command disables/enables viewing of the measurement points in the GUI and associates it with a specific measurement name identifier. A 0 or OFF value disables viewing and a 1 or ON value enables viewing of the measurement points. The query returns the current value.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example :PLUG:OTIM:SHOW:SMP 'MyMeasurement', 1

:PLUGin:OTIMing:SHOW:UNIT[?]

Syntax :PLUGin:OTIMing:SHOW:UNIT 'identifier', <UINT|TIME>

:PLUGin:OTIMing:SHOW:UNIT? 'identifier'

Input Parameters 'identifier': Specify the measurement name.

<UINT|TIME>: Specify the timing units.

Return Range UINT|TIME

Description This command specifies the timing units and associates it with a specific measurement name identifier. If UINT is specified then the timing units are in UI. If TIME is specified then the timing units are in seconds. The query returns the current value.

This SCPI is applicable for M8041A, M8051A, M8062A2, M8043A and M8046A.

Example :PLUG:OTIM:SHOW:UNIT 'MyMeasurement', unit

:PLUGin:OTIMing:SHOW:VSCale[?]

Syntax

```
:PLUGin:OTIMing:SHOW:VSCale 'identifier', <LOG|LIN>
```

```
:PLUGin:OTIMing:SHOW:VSCale? 'identifier'
```

Input Parameters

'identifier': Specify the measurement name.

<LOG|LIN>: Specify the vertical axis scale.

Return Range

LOG|LIN

Description

This command specifies the vertical BER axis scale and associates it with a specific measurement name identifier. If LOG is specified then the BER values are in terms of a logarithmic scale. If LIN is specified then the BER values are in terms of a linear scale. The query returns the current value.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example

```
:PLUG:OTIM:SHOW:VSC 'MyMeasurement', LOG
```

:PLUGin:OTIMing:SHOW:SGLegends[?]

Syntax

```
:PLUGin:OTIMing:SHOW:SGLegends 'Identifier', <0|1|OFF|ON>
```

```
:PLUGin:OTIMing:SHOW:SGLegends? 'Identifier'
```

Input Parameters

Identifier: 'MyMeasurement'

<0|1|OFF|ON>: Specify the show/hide state for graph legends.

Return Range

0|1

Description

This command shows/hides the graph legends.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example

```
:PLUGin:OTIMing:SHOW:SGLegends 'MyMeasurement', 1
```

:PLUGin:OTIMing:SHOW:COMment[?]

Syntax

```
:PLUGin:OTIMing:SHOW:COMment 'Identifier', <“Comment”>
:PLUGin:OTIMing:SHOW:COMment? 'Identifier'
```

Input Parameters

Identifier: Specify the measurement name. For example; 'MyMeasurement'

<“Comment”>: Enter the desired comment.

Description

This command is used to enter a comment for each measurement. This comment will be displayed on the **Measurement History** tab.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example

```
:PLUG:OTIM:SHOW:COM 'MyMeasurement', “Output Timing Example”
:PLUG:OTIM:SHOW:COM? 'MyMeasurement'
Output Timing Example
```

:PLUGin:OTIMing:CATalog?

Syntax

```
:PLUGin:OTIMing:CATalog?
```

Description

This command returns a list of all created output timing measurement names currently available for measuring.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example

Assume the following is a list of created output timing measurement names:

```
:PLUG:OTIM:NEW 'OTIM_1'
:PLUG:OTIM:NEW 'OTIM_2'
:PLUG:OTIM:NEW 'OTIM_3'
```

The command and returned list would look like the following:

```
:PLUG:OTIM:CAT?
"OTIM_1","OTIM_2","OTIM_3"
```

:PLUGin:OTIMing:TABS?

Syntax :PLUGin:OTIMing:TABS?

Description This query returns a list of all results tabs which can be exported as a CSV file.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example :PLUG:OTIM:TABS?

Output "Calculated Results","M1.DataIn1"

:PLUGin:OTIMing:EXPort

Syntax :PLUGin:OTIMing:EXPort 'Identifier', 'TabName', 'FilePath'

Description This command exports the measurement data of the selected results tab to a csv file.

Input Parameters

- 'Identifier': Specify the measurement name; for example ''Output Timing 1''.
- 'TabName' – Specify the results tab name. To fetch the results tabs for a measurement, refer to the query ":PLUGin:OTIMing:TABS? on page 234". However, if you want all tabs to be saved as a csv file, use " " (single/double quotes).
- 'FilePath' – Specify the location where the CSV file should be saved.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Examples Example 1: The following command will save the 'Calculated Results' tab of the output timing measurement to *Results.csv*:

:PLUGin:OTIM:EXPort 'Output Timing 1', 'Calculated Results', 'C:\Results.csv'

Example 2: The following command will save all results tab of the output timing measurement to *Results.csv*:

:PLUGin:OTIM:EXPort 'Output Timing 1', ' ', 'C:\Results.csv'

:PLUGin:OTIMing:NEW

Syntax

```
:PLUGin:OTIMing:NEW 'identifier'
```

Input Parameters

'identifier': Specify a new measurement name.

Description

This command creates a new output timing measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example

The following example creates an output timing measurement with the measurement name identifier called 'MyOutputTiming':

```
:PLUG:OTIM:NEW 'MyOutputTiming'
```

NOTE

Creating multiple plugins using this command may slow down the GUI operations which may also result delay in remote programming. To prevent the plugin from opening automatically in the GUI, it is recommended to use “0” as a parameter input in this command.

Example- :PLUG:OTIM:NEW 'MyOutputTiming',0

:PLUGin:OTIMing:DELete

Syntax

```
:PLUGin:OTIMing:DELete 'identifier'
```

Input Parameters

'identifier': Specify the measurement name to delete.

Description

This command deletes a previously created output timing measurement addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example

The following example deletes an output timing measurement with the measurement name identifier called 'MyOutputTiming':

```
:PLUG:OTIM:DEL 'MyOutputTiming'
```

:PLUGin:OTIMing:RESet

Syntax

:PLUGin:OTIMing:RESet 'identifier'

Input Parameters

'identifier': Specify the measurement name to reset.

Description

This command resets an output timing measurement addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example

The following example resets an output timing measurement addressed with the measurement name identifier called 'MyOutputTiming':

```
:PLUG:OTIM:RES 'MyOutputTiming'
```

:PLUGin:OTIMing:STARt

Syntax

:PLUGin:OTIMing:STARt 'identifier'

Input Parameters

'identifier': Specify the measurement name to start.

Description

This command starts an output timing measurement addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example

The following example starts an output timing measurement with the measurement name identifier called 'MyOutputTiming':

```
:PLUG:OTIM:STAR 'MyOutputTiming'
```

:PLUGin:OTIMing:STOP

Syntax

:PLUGin:OTIMing:STOP 'identifier'

Input Parameters

'identifier': Specify the measurement name to stop.

Description

This command stops an output timing measurement addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example

The following example stops an output timing measurement with the measurement name identifier called 'MyOutputTiming':

```
:PLUG:OTIM:STOP 'MyOutputTiming'
```

:PLUGin:OTIMing:BREak

Syntax

```
:PLUGin:OTIMing:BREak 'identifier'
```

Input Parameters

'identifier': Specify the measurement name to break/pause.

Description

This command breaks/pauses a measurement addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example

The following example breaks an output timing measurement with the measurement name identifier called 'MyOutputTiming':

```
:PLUG:OTIM:BRE 'MyOutputTiming'
```

:PLUGin:OTIMing:STEP

Syntax

```
:PLUGin:OTIMing:STEP 'identifier'
```

Input Parameters

'identifier': Specify the measurement name to step.

Description

This command allows you to step through an output timing measurement, addressed by the measurement name identifier, that was stopped using the BREak command.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example

The following example triggers a single output timing measurement step of the measurement addressed with the measurement name identifier called 'MyOutputTiming':

```
:PLUG:OTIM:STEP 'MyOutputTiming'
```

:PLUGin:OTIMing:CONTinue

Syntax

```
:PLUGin:OTIMing:CONTinue 'identifier'
```

Input Parameters

'identifier': Specify the measurement name to continue.

Description

This command allows you to continue an output timing measurement, addressed by the measurement name identifier, that was stopped using the BREak command.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example

The following example continues a halted output timing measurement addressed with the measurement name identifier called 'MyOutputTiming':

```
:PLUG:OTIM:CONT 'MyOutputTiming'
```

:PLUGin:OTIMing:GRAPh?

Syntax

```
:PLUGin:OTIMing:GRAPh? 'identifier'
[,DesiredWidth][,DesiredHeight][,<CURR | WHIT>][,<PNG>][,1 |
0][,'TabName']
```

Input Parameters

'identifier': Specify measurement name.

Other optional parameters are:

[,DesiredWidth]: Specify the desired width of the image.

[,DesiredHeight]: Specify the desired height of the image.

[,<CURR | WHIT>]: Specify whether the user wants to capture in current theme or wants to capture in white background.

[,<PNG>]: Specify the format of the image. The default format is PNG.

[,1 | 0]: Specify whether to capture the graph with legends or not.

[,'TabName']: Specify the tab name on which the graph is supposed to be captured.

Description

This query returns data of the image captured from the graph view of the plugin in the specified format.

Example

```
:PLUG:OTIM:GRAP? 'MyMeasurement',1000,800,CURR,PNG,1,'Graph'
```

:PLUGin:OTIMing:RUN:HISTory[:STATe][?]

Syntax :PLUGin:OTIMing:RUN:HISTory[:STATe] 'identifier',<0|1|OFF|ON>

:PLUGin:OTIMing:RUN:HISTory[:STATe]? 'identifier'

Input Parameters 'identifier': Specify the measurement name.

Return Range 0|1

Description This command enables/disables the storage of output timing measurement results addressed by the measurement name identifier.

This query returns the current setting.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example The following example enables storage of output timing measurement results with the measurement name identifier called 'MyMeasurement':

:PLUG:OTIM:RUN:HIST 'MyMeasurement', 1

:PLUGin:OTIMing:RUN:HISTory:CLEar

Syntax :PLUGin:OTIMing:RUN:HISTory:CLEar 'identifier'

Input Parameters 'identifier': Specify the measurement name.

Description This command deletes the output timing measurement history addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example The following example deletes the output timing measurement history with the measurement name identifier called 'MyMeasurement':

:PLUG:OTIM:RUN:HIST:CLE 'MyMeasurement'

:PLUGin:OTIMing:RUN:LOG?

Syntax :PLUGin:OTIMing:RUN:LOG? 'identifier'

Input Parameters 'identifier': Specify the measurement name.

Description This command returns logs for the addressed measurement.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example This command returns logs for the measurement 'MyMeasurement'.

:PLUG:OTIM:RUN:LOG? 'MyMeasurement'

The format of log returned is in the following way:

#XY Message Log

where X denotes the length of digits for Y, Y denotes the number of characters in the log then followed by the log message.

For example:

#10 means, there is 1 digit after 1 and there is 0 character in the log.

#2492 means, there are 3 digits after 2 and there are 492 characters in the log message.

:PLUGin:OTIMing:RUN:MESSage?

Syntax :PLUGin:OTIMing:RUN:MESSage? 'identifier'

Input Parameters 'identifier': Specify the measurement name.

Description This command returns a string describing the state of an output timing measurement addressed by the measurement name identifier. Possible states include NotStarted, Running, Suspended, Finished, Error, or Stopped.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example The following example returns the state of an output timing measurement with the measurement name identifier called 'MyMeasurement':

:PLUG:OTIM:RUN:MESS? 'MyMeasurement' -> Running

:PLUGin:OTIMing:RUN:PROGress?

Syntax :PLUGin:OTIMing:RUN:PROGress? 'identifier'

Input Parameters 'identifier': Specify measurement name.

Return Range 0.0 to 1.0

Description This command returns a number in the range of 0.0 to 1.0 to indicate the progress of an output timing measurement addressed by the measurement name identifier. A 0.0 indicates that the measurement has not started and 1.0 indicates the measurement is finished.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example The following example returns the progress of an output timing measurement with the measurement name identifier called 'MyMeasurement':

```
:PLUG:OTIM:RUN:PROG? 'MyMeasurement'
```

```
0.51
```

:PLUGin:OTIMing:RUN[:STATus]?

Syntax :PLUGin:OTIMing:RUN[:STATus]? 'identifier'

Input Parameters 'identifier': Specify measurement name.

Return Range 0|1

Description This command returns the running status of an output timing measurement addressed by the measurement name identifier. A 0 indicates the measurement is not running and a 1 indicates the measurement is running.

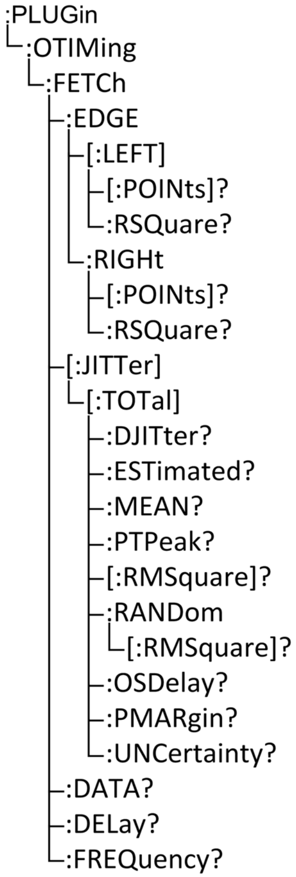
This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example The following example returns the running status of an output timing measurement with the measurement name identifier called 'MyMeasurement':

```
:PLUG:OTIM:RUN? 'MyMeasurement'
1
```

:PLUGin:OTIMing:FETCh Subnode

This subnode has the following SCPI structure:



```
:PLUGin
└:OTIMing
 └:FETCh
  ├:EDGE
  │├[:LEFT]
  ││├[:POINts]?
  ││└:RSQuare?
  │└:RIGHt
  │ ├[:POINts]?
  │ └:RSQuare?
  ├[:JITTer]
  │└[:TOTal]
  │ ├:DJITter?
  │ ├:ESTimated?
  │ ├:MEAN?
  │ ├:PTPeak?
  │ ├[:RMSquare]?
  │ ├:RANDom
  │ │└[:RMSquare]?
  │ ├:OSDelay?
  │ ├:PMARgin?
  │ └:UNCertainty?
  ├:DATA?
  ├:DELay?
  └:FREQuency?
```

This subnode has the following commands:

Table 30

Name	Description under
:EDGE[:LEFT][:POINts]?	:PLUGin:OTIMing:FETCh:EDGE[:LEFT][:POINts]? on page 244
:EDGE[:LEFT]:RSQuare?	:PLUGin:OTIMing:FETCh:EDGE[:LEFT]:RSQuare? on page 244
:EDGE:RIGHt[:POINts]?	:PLUGin:OTIMing:FETCh:EDGE:RIGHt[:POINts]? on page 245
:EDGE:RIGHt:RSQuare?	:PLUGin:OTIMing:FETCh:EDGE:RIGHt:RSQuare? on page 245
[:JITTer][:TOTal]:DJITTer?	:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:DJITTer? on page 246
[:JITTer][:TOTal]:ESTimated?	:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:ESTimated? on page 246
[:JITTer][:TOTal]:MEAN?	:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:MEAN? on page 247
[:JITTer][:TOTal]:PTPeak?	:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:PTPeak? on page 247
[:JITTer][:TOTal][:RMSquare]?	:PLUGin:OTIMing:FETCh[:JITTer][:TOTal][:RMSquare]? on page 248
[:JITTer][:TOTal]:RANDom [:RMSquare]?	:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:RANDom[:RMSquare]? on page 248
[:JITTer][:TOTal]:OSDelay?	:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:OSDelay? on page 249
[:JITTer][:TOTal]:PMARgin?	:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:PMARgin? on page 249
[:JITTer][:TOTal]:UNCertainty?	:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:UNCertainty? on page 250
:DATA?	:PLUGin:OTIMing:FETCh:DATA? on page 250
:DELay?	:PLUGin:OTIMing:FETCh:DELay? on page 251
:FREQuency?	:PLUGin:OTIMing:FETCh:FREQuency? on page 252

:PLUGin:OTIMing:FETCh:EDGE[:LEFT][:POINts]?

Syntax :PLUGin:OTIMing:FETCh:EDGE[:LEFT][:POINts]? 'identifier'[,'Location']

Input Parameters 'identifier': Specify the measurement name.

['Location']: Specify location (optional).

Return Range <NR3>

Description This command returns the number of points that have been measured between the BER Threshold and the Min. BER for RJ/DJ separation threshold associated with a specific measurement name identifier. It is calculated for left slope (left edge). This number has to be greater than 2 for the RJ, DJ and estimated TJ values to be applicable.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

A measurement run on a location identifier (for example, 'M1.DataIn1') will return following results.

Example :PLUG:OTIM:FETC:EDGE? 'MyOutputTiming','M1.DataIn1'

:PLUGin:OTIMing:FETCh:EDGE[:LEFT]:RSQuare?

Syntax :PLUGin:OTIMing:FETCh:EDGE[:LEFT]:RSQuare? 'identifier'[,'Location']

Input Parameters 'identifier': Specify the measurement name.

['Location']: Specify location (optional).

Return Range <NR3>

Description This command returns the R^2 values calculated for left slope of the bathtub curve (left edge) associated with a specific measurement name identifier. They are a measure of how well the transformed points between BER Threshold and Min. BER for RJ/DJ Separation fit to the linear regression. They have to be greater than 0.75 for the RJ, DJ and estimated TJ values to be applicable.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

A measurement run on a location identifier (for example, 'M1.DataIn1') will return following results.

Example :PLUG:OTIM:FETC:EDGE:RSQ? 'MyOutputTiming','M1.DataIn1'

:PLUGin:OTIMing:FETCh:EDGE:RIGHt[:POINts]?

Syntax :PLUGin:OTIMing:FETCh:EDGE:RIGHt[:POINts]? 'identifier'[,'Location']

Input Parameters 'identifier': Specify the measurement name.

['Location']: Specify location (optional).

Return Range <NR3>

Description This command returns the number of points that have been measured between the BER Threshold and the Min. BER for RJ/DJ separation threshold associated with a specific measurement name identifier. It is calculated for right slope (right edge). This number has to be greater than 2 for the RJ, DJ and estimated TJ values to be applicable.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

A measurement run on a location identifier (for example, 'M1.DataIn1') will return following results.

Example :PLUG:OTIM:FETC:EDGE:RIGH? 'MyOutputTiming','M1.DataIn1'

:PLUGin:OTIMing:FETCh:EDGE:RIGHt:RSQuare?

Syntax :PLUGin:OTIMing:FETCh:EDGE:RIGHt:RSQuare? 'identifier'[,'Location']

Input Parameters 'identifier': Specify the measurement name.

['Location']: Specify location (optional).

Return Range <NR3>

Description This command returns the R^2 values calculated for right slope of the bathtub curve (right edge) associated with a specific measurement name identifier. They are a measure of how well the transformed points between BER Threshold and Min. BER for RJ/DJ Separation fit to the linear regression. They have to be greater than 0.75 for the RJ, DJ and estimated TJ values to be applicable.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

A measurement run on a location identifier (for example, 'M1.DataIn1') will return following results.

Example :PLUG:OTIM:FETC:EDGE:RIGH:RSQ? 'MyOutputTiming','M1.DataIn1'

:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:DJITter?

Syntax :PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:DJITter? 'identifier'[,'Location']

Input Parameters 'identifier': Specify the measurement name.

['Location']: Specify location (optional).

Return Range <NR3>

Description This command returns the total jitter component with non-Gaussian distribution (deterministic jitter) associated with a specific measurement name identifier. After transforming a contiguous range of measured points into Q space and performing a linear regression, it is calculated as the period minus the difference between the means of the two straight lines.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

A measurement run on a location identifier (for example, 'M1.DataIn1') will return following result.

Example :PLUG:OTIM:FETC:DJIT? 'MyOutputTiming','M1.DataIn1'

:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:ESTimated?

Syntax :PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:ESTimated? 'identifier'[,'Location']

Input Parameters 'identifier': Specify the measurement name.

['Location']: Specify location (optional).

Return Range <NR3>

Description This command returns an estimate of the expected jitter for very low bit error rates associated with a specific measurement name identifier. After extrapolating the measured BER curves, it is calculated as the period minus the expected width of the eye opening.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

A measurement run on a location identifier (for example, 'M1.DataIn1') will return following result.

Example :PLUG:OTIM:FETC:EST? 'MyOutputTiming','M1.DataIn1'

:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:MEAN?

Syntax	:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:MEAN? 'identifier'[,'Location']
Input Parameters	'identifier': Specify the measurement name. ['Location']: Specify location (optional).
Return Range	<NR3>
Description	This command returns the mean value for total jitter associated with a specific measurement name identifier. This value is calculated as the weighted average of the left edge jitter histogram. This command also associates this parameter with a specific measurement name identifier. This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A. A measurement run on a location identifier (for example, 'M1.DataIn1') will return following result.
Example	:PLUG:OTIM:FETC:MEAN? 'MyOutputTiming','M1.DataIn1'

:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:PTPeak?

Syntax	:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:PTPeak? 'identifier'[,'Location']
Input Parameters	'identifier': Specify the measurement name. ['Location']: Specify location (optional).
Return Range	<NR3>
Description	This command returns the peak to peak value for total jitter associated with a specific measurement name identifier. This value is calculated as the pulse period (unit interval) minus the phase margin. This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A. A measurement run on a location identifier (for example, 'M1.DataIn1') will return following result.
Example	:PLUG:OTIM:FETC:PTP? 'MyOutputTiming','M1.DataIn1'

:PLUGin:OTIMing:FETCh[:JITTer][:TOTal][:RMSquare]?

Syntax :PLUGin:OTIMing:FETCh[:JITTer][:TOTal][:RMSquare]? 'identifier'[,'Location']

Input Parameters 'identifier': Specify the measurement name.

['Location']: Specify location (optional).

Return Range <NR3>

Description This command returns the average of the left and right jitter histogram root mean squared values associated with a specific measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

A measurement run on a location identifier (for example, 'M1.DataIn1') will return following result.

Example :PLUG:OTIM:FETC? 'MyOutputTiming','M1.DataIn1'

:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:RANDom[:RMSquare]?

Syntax :PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:RANDom[:RMSquare]? 'identifier'[,'Location']

Input Parameters 'identifier': Specify the measurement name.

['Location']: Specify location (optional).

Return Range <NR3>

Description This command returns the total jitter component with Gaussian distribution associated with a specific measurement name identifier. After transforming a contiguous range of measured points into Q space and performing a linear regression, it is calculated as the mean of the sigmas of the two straight lines. The contiguous range is limited by the BER threshold and the minimum BER for RJ/DJ separation threshold.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

A measurement run on a location identifier (for example, 'M1.DataIn1') will return following result.

Example :PLUG:OTIM:FETC:RAND? 'MyOutputTiming','M1.DataIn1'

:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:OSDelay?

Syntax	:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:OSDelay? 'identifier'[,'Location']
Input Parameters	'identifier': Specify the measurement name. ['Location']: Specify location (optional).
Return Range	<NR3>
Description	This command returns the average of the left (Ai) and right (Bi) bathtub/BER threshold intersections (optimum sample point delay) associated with a specific measurement name identifier. This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A. A measurement run on a location identifier (for example, 'M1.DataIn1') will return following result.
Example	:PLUG:OTIM:FETC:OSD? 'MyOutputTiming','M1.DataIn1'

:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:PMARgin?

Syntax	:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:PMARgin? 'identifier'[,'Location']
Input Parameters	'identifier': Specify the measurement name. ['Location']: Specify location (optional).
Return Range	<NR3>
Description	This command returns the period of time where the bit error rate is lower than the BER threshold (phase margin) associated with a specific measurement name identifier. This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A. A measurement run on a location identifier (for example, 'M1.DataIn1') will return following result.
Example	:PLUG:OTIM:FETC:PMAR? 'MyOutputTiming','M1.DataIn1'

:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:UNCertainty?

Syntax

:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:UNCertainty? 'identifier'[,'Location']

Input Parameters

'identifier': Specify the measurement name.

['Location']: Specify location (optional).

Return Range

<NR3>

Description

This command returns the total jitter uncertainty associated with a specific measurement name identifier. The returned value is only valid if the “Sample Delay Optimization” is set to “Fast Total Jitter” (FTJ) using the :PLUG:OTIM:ACQ:OPT command.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

A measurement run on a location identifier (for example, 'M1.DataIn1') will return following result.

Example

```
:PLUG:OTIM:FETC:UNC? 'MyOutputTiming','M1.DataIn1'
```

:PLUGin:OTIMing:FETCh:DATA?

Syntax

:PLUGin:OTIMing:FETCh:DATA? 'identifier'[,'Location']

Input Parameters

'identifier': Specify the measurement name.

['Location']: Specify location (optional).

Description

This command returns the raw data of the output timing measurement associated with a specific measurement name identifier.

A measurement run on a location identifier (for example, 'M1.DataIn1') will return following results.

This SCPI is applicable for M8041A, M8051A, M8043A and M8062A.

Example

```
:PLUG:OTIM:FETC:DATA? 'Output Timing 1','M1.DataIn1'
```

```
"M1.DataIn1",1420788572.96819,499999011,500000989,249999011,250
000000,-0.6035,"UNKNOWN", ...
```

The first item in the comma separated list is the name of the location for which the data is being queried. Rest of the values in the comma separated list for each measurement point are Time Stamp, Compared One Bits, Compared Zero Bits, Errored One Bits, Errored Zero Bits, Sample

Delay and Pass/Fail Results and are repeated in the same order for subsequent measured points.

In the case of Sample Delay Optimization as "None" the pass/fail result for each measurement point is returned as "UNKNOWN". However if the Fast Total Jitter Measurement is performed then the Pass/Fail results could be either "PASS", "FAIL" or "UNKNOWN"

If the location is omitted from the above example e.g. if the query is :PLUG:OTIM:FETC:DATA? 'Output Timing 1' then the query returns the comma separated list of all the analyzer location group participating in the measurement

(("M1.DataIn1",1420788572.96819,499999011,500000989,249999011,250000000,-0.6035,"UNKNOWN", ...),
("M1.DataIn2",1420788572.96819,499999011,500000989,249999011,250000000,-0.6035,"UNKNOWN",))

:PLUGin:OTIMing:FETCh:DELay?

Syntax :PLUGin:OTIMing:FETCh:DELay? 'identifier'

Input Parameters 'identifier': Specify the measurement name.

Description This command returns the currently set delay sweep in seconds. The delay sweep is set using the :INPut:DELay command. After setting the value, a data center alignment is executed followed by an output timing measurement.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example :PLUG:OTIM:FETC:DEL? 'Output Timing 1'
("M1.DataIn1",4.879243000000001E-009)

:PLUGin:OTIMing:FETCh:FREQuency?

Syntax

```
:PLUGin:OTIMing:FETCh:FREQuency? 'identifier'
```

Input Parameters

'identifier': Specify the measurement name.

Description

This command returns the currently set frequency of the measurement. The frequency is set using the :SOURce:FREQuency command.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

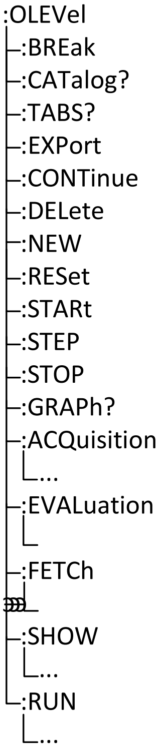
Example

```
:PLUG:OTIM:FETC:FREQ? 'Output Timing 1'
("M1.DataIn1",5.000000000000000E+009)
```

:PLUGin:OLEVel Subnode

This is the basic measurement for BER versus sample threshold and is integrated with the M8070B software during the startup. The measurement comes with the set of Acquisition and Evaluation Parameters which are used to configure the measurement run. The Acquisition parameters can only be configured before the measurement starts (i.e. during the measurement run you cannot change the value of Acquisition parameter); however the Evaluation parameters can be configured any time even after the measurement has been run.

This subnode has the following SCPI structure:



```
:OLEVel
├─:BREak
├─:CATalog?
├─:TABS?
├─:EXPort
├─:CONTinue
├─:DELete
├─:NEW
├─:RESet
├─:STARt
├─:STEP
├─:STOP
├─:GRAPh?
├─:ACQuisition
│ └...
├─:EVALuation
│ └
├─:FETCh
│ └
├─:SHOW
│ └...
└─:RUN
  └...
```

This subnode has the following commands and subnodes:

Table 31

Name	Description under
:BREak[?]	:PLUGin:OLEVel:BREak on page 255
:CATalog?	:PLUGin:OLEVel:CATalog? on page 255
:TABS?	:PLUGin:OLEVel:TABS? on page 256
:EXPort	:PLUGin:OLEVel:EXPort on page 256
:CONTinue[?]	:PLUGin:OLEVel:CONTinue on page 257
:DELete[?]	:PLUGin:OLEVel:DELete on page 257
:NEW[?]	:PLUGin:OLEVel:NEW on page 257
:RESet[?]	:PLUGin:OLEVel:RESet on page 258
:STARt[?]	:PLUGin:OLEVel:STARt on page 258
:STEP[?]	:PLUGin:OLEVel:STEP on page 259
:STOP[?]	:PLUGin:OLEVel:STOP on page 259
:GRAPh?	:PLUGin:OLEVel:GRAPh? on page 260
Subnodes	
:ACQuisition:	:PLUGin:OLEVel:ACQuisition Subnode on page 261
:EVALuation	:PLUGin:OLEVel:EVALuation Subnode on page 265
:FETCh	:PLUGin:OLEVel:FETCh Subnode on page 267
:SHOW	:PLUGin:OLEVel:SHOW Subnode on page 280
:RUN	:PLUGin:OLEVel:RUN Subnode on page 283

:PLUGin:OLEVel:BREak

Syntax :PLUGin:OLEVel:BREak 'identifier'

Input Parameters 'identifier': Specify the measurement name to break/pause.

Description This command breaks/pauses a measurement addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example The following example breaks an output level measurement with the measurement name identifier called 'MyMeasurement':

```
:PLUG:OLEV:BRE 'MyMeasurement'
```

:PLUGin:OLEVel:CATalog?

Syntax :PLUGin:OLEVel:CATalog?

Description This command returns a list of all created output level measurement names currently available for measuring.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example Assume three output level measurements named as OLEVel_1, OLEVel_2 and OLEVel_3 are created using NEW command:

```
:PLUG:OLEVel:NEW 'OLEVel_1'
:PLUG:OLEVel:NEW 'OLEVel_2'
:PLUG:OLEVel:NEW 'OLEVel_3'
```

The command and returned list would look like the following:

```
:PLUG:OLEVel:CAT?
"OLEVel_1,OLEVel_2,OLEVel_3"
```

:PLUGin:OLEVel:TABS?

Syntax :PLUGin:OLEVel:TABS?

Description This query returns a list of all results tabs which can be exported as a CSV file.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUG:OLEV:TABS?

Output "Calculated Results","M1.DataIn1"

:PLUGin:OLEVel:EXPort

Syntax :PLUGin:OLEVel:EXPort 'Identifier', 'TabName', 'FilePath'

Description This command exports the measurement data of the selected results tab to a csv file.

Input Parameters

- 'Identifier': Specify the measurement name; for example ''Output Level 1''.
- 'TabName' – Specify the results tab name. To fetch the results tabs for a measurement, refer to the query ":PLUGin:OLEVel:TABS? on page 256". However, if you want all tabs to be saved as a csv file, use " " (single/double quotes).
- 'FilePath' – Specify the location where the CSV file should be saved.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Examples Example 1: The following command will save the 'Calculated Results' tab of the output level measurement to *Results.csv*:

:PLUGin:OLEV:EXPort 'Output Level 1', 'Calculated Results', 'C:\Results.csv'

Example 2: The following command will save all results tab of the output level measurement to *Results.csv*:

:PLUGin:OLEV:EXPort 'Output Level 1', ' ', 'C:\Results.csv'

:PLUGin:OLEVel:CONTinue

Syntax :PLUGin:OLEVel:CONTinue 'identifier'

Input Parameters 'identifier': Specify the measurement name to continue.

Description This command allows you to continue an output level measurement, addressed by the measurement name identifier, which was stopped using the BREak command.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example The following example continues a halted output level measurement addressed with the measurement name identifier called 'MyMeasurement':

```
:PLUG:OLEV:CONT 'MyMeasurement'
```

:PLUGin:OLEVel:DELete

Syntax :PLUGin:OLEVel:DELete 'identifier'

Input Parameters 'identifier': Specify the measurement name to delete.

Description This command deletes a previously created output level measurement addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example The following example deletes an output level measurement addressed by the measurement name identifier called 'MyMeasurement':

```
:PLUG:OLEV:DEL 'MyMeasurement'
```

:PLUGin:OLEVel:NEW

Syntax :PLUGin:OLEVel:NEW 'identifier'

Input Parameters 'identifier': Specify the measurement name.

Description This command creates a new output level measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example

The following example creates an output level measurement name identifier called 'MyMeasurement':

:PLUG:OLEV:NEW 'MyMeasurement'

NOTE

Creating multiple plugins using this command may slow down the GUI operations which may also result delay in remote programming. To prevent the plugin from opening automatically in the GUI, it is recommended to use “0” as a parameter input in this command.

Example- :PLUG:OLEV:NEW 'MyMeasurement',0

:PLUGin:OLEVel:RESet

Syntax

:PLUGin:OLEVel:RESet 'identifier'

Input Parameters

'identifier': Specify the measurement name to reset.

Description

This command resets an output level measurement addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example

The following example resets an output level measurement addressed with the measurement name identifier called 'MyMeasurement':

:PLUG:OLEV:RES 'MyMeasurement'

:PLUGin:OLEVel:STARt

Syntax

:PLUGin:OLEVel:STARt 'identifier'

Input Parameters

'identifier': Specify the measurement name to start.

Description

This command starts an output level measurement addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example

The following example starts an output level measurement addressed with the measurement name identifier called 'MyMeasurement':

:PLUG:OLEV:STAR 'MyMeasurement'

:PLUGin:OLEVel:STEP

Syntax

:PLUGin:OLEVel:STEP 'identifier'

Input Parameters

'identifier': Specify the measurement name to step.

Description

This command allows you to step through an output level measurement, addressed by the measurement name identifier, that was stopped using the BREak command.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example

The following example triggers a single output level measurement step

of the measurement addressed with the measurement name identifier

called 'MyMeasurement':

:PLUG:OLEV:STEP 'MyMeasurement'

:PLUGin:OLEVel:STOP

Syntax

:PLUGin:OLEVel:STOP 'identifier'

Input Parameters

'identifier': Specify the measurement name to stop.

Description

This command stops an output level measurement addressed by the

measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example

The following example stops an output level measurement with the

measurement name identifier called 'MyMeasurement':

:PLUG:OLEV:STOP 'MyMeasurement'

:PLUGin:OLEVel:GRAPh?

Syntax

:PLUGin:OLEVel:GRAPh? 'identifier' [,DesiredWidth][,DesiredHeight][,<CURR | WHIT>][,<PNG>][,1 | 0][,'TabName']

Input Parameters

'identifier': Specify measurement name.

Other optional parameters are:

[,DesiredWidth]: Specify the desired width of the image.

[,DesiredHeight]: Specify the desired height of the image.

[,<CURR | WHIT>]: Specify whether the user wants to capture in current theme or wants to capture in white background.

[,<PNG>]: Specify the format of the image. The default format is PNG.

[,1 | 0]: Specify whether to capture the graph with legends or not.

[,'TabName']: Specify the tab name on which the graph is supposed to be captured.

Description

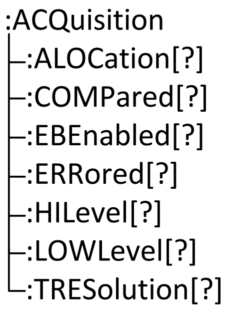
This query returns data of the image captured from the graph view of the plugin in the specified format.

Example

:PLUG:OLEV:GRAP? 'MyMeasurement',1000,800,CURR,PNG,1,'Graph'

:PLUGin:OLEVel:ACQuisition Subnode

This subnode has the following SCPI structure:



```
:ACQuisition
├:ALOCation[?]
├:COMPared[?]
├:EBEnabled[?]
├:ERRored[?]
├:HILevel[?]
├:LOWLevel[?]
└:TRESolution[?]
```

This subnode has the following commands:

Table 32

Name	Description under
:ALOCation[?]	:PLUGin:OLEVel:ACQuisition:ALOCation[?] on page 262
:COMPared[?]	:PLUGin:OLEVel:ACQuisition:COMPared[?] on page 262
:EBEnabled[?]	:PLUGin:OLEVel:ACQuisition:EBEnabled[?] on page 263
:ERRored[?]	:PLUGin:OLEVel:ACQuisition:ERRored[?] on page 263
:HILevel[?]	:PLUGin:OLEVel:ACQuisition:HILevel[?] on page 263
:LOWLevel[?]	:PLUGin:OLEVel:ACQuisition:LOWLevel[?] on page 264
:TRESolution[?]	:PLUGin:OLEVel:ACQuisition:TRESolution[?] on page 264

:PLUGin:OLEVel:ACQuisition:ALOCation[?]

Syntax :PLUGin:OLEVel:ACQuisition:ALOCation 'Identifier', < location-string>

:PLUGin:OLEVel:ACQuisition:ALOCation? 'Identifier'

Input Parameters Identifier: 'MyMeasurement'

Description This command sets the Location/Location Group string against which the Data Acquisition is performed. The query returns the current Location/Location group string against which measurement is configured for the Data Acquisition.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUG:OLEV:ACQ:ALOC 'MyMeasurement',"M1.DataIn1"

:PLUGin:OLEVel:ACQuisition:COMPared[?]

Syntax :PLUGin:OLEVel:ACQuisition:COMPared[?] 'Identifier', <NRf>

Input Parameters Identifier: 'MyMeasurement'

Return Range 1E+0 to 1E+18

Description This command allows the user to specify the number of compared bits; that would be used as criteria for moving to the next measurement point. The query returns current value.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUG:OLEV:ACQ:COMP 'MyMeasurement', 1E+9

NOTE **The BER measurement processes at least as many bits as defined by the acquisition parameters, it stops when the end criteria is reached or exceeded. Due to update period and hardware granularities, an individual BER measurement will typically process more bits than configured.**

:PLUGin:OLEVel:ACQuisition:EBEnabled[?]

Syntax :PLUGin:OLEVel:ACQuisition:EBEnabled[?] 'Identifier', <0 | 1 | ON | OFF>

Input Parameters Identifier: 'MyMeasurement'

Return Range 0|1

Description If enabled then user can specify the number of errored bits additionally with number of compared bits in order to make them criteria together to move to the next measurement point. If this command sets the value as 0 then only number of compared bits are considered to move to the next measurement point while running the measurement. The query returns current value.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUG:OLEV:ACQ:EBE 'MyMeasurement', 1

:PLUGin:OLEVel:ACQuisition:ERRored[?]

Syntax :PLUGin:OLEVel:ACQuisition:ERRored[?] 'Identifier', <NRf>

Input Parameters Identifier: 'MyMeasurement'

Return Range 1E+0 to 1E+18

Description This command allows the user to specify the number of errored bits. This command can only be used when the command PLUGin:OLEVel:ACQuisition:EBEnabled is set to 1. This would be the criteria to move to the next measurement point while the measurement is running. The query returns current value.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUG:OLEV:ACQ:ERR'MyMeasurement', 1E+4

:PLUGin:OLEVel:ACQuisition:HILevel[?]

Syntax :PLUGin:OLEVel:ACQuisition:HILevel[?] 'Identifier', <NRf>

Input Parameters Identifier: 'MyMeasurement'

Return Range -5 V to 5 V

Description This command allows the user to specify the sample threshold high level.

The query returns current value of sample threshold high Level.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUG:OLEV:ACQ:HIL 'MyMeasurement', 0.5

:PLUGin:OLEVel:ACQuisition:LOWLevel[?]

Syntax :PLUGin:OLEVel:ACQuisition:LOWLevel[?] 'Identifier', <NRf>

Input Parameters Identifier: 'MyMeasurement'

Return Range -5 V to 5 V

Description This command allows the user to specify the Sample Threshold Low Level. The query returns current value of sample threshold low level.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUG:OLEV:ACQ:LOWL 'MyMeasurement', -0.5

:PLUGin:OLEVel:ACQuisition:TRESolution[?]

Syntax :PLUGin:OLEVel:ACQuisition:TRESolution[?] 'Identifier', <NRf>

Input Parameters Identifier: 'MyMeasurement'

Return Range 1E-3 to 1

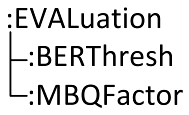
Description This command allows the user to specify the sample threshold resolution in terms of Volts. The query returns current value of sample threshold resolution in terms of Volts.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUG:OLEV:ACQ:TRES 'MyMeasurement', 0.001

:PLUGin:OLEVel:EVALuation Subnode

This subnode has the following SCPI structure:



This subnode has the following commands:

Table 33

Name	Description under
:BERThresh[?]	:PLUGin:OLEVel:EVALuation:BERThresh[?] on page 265
:MBQFactor[?]	:PLUGin:OLEVel:EVALuation:MBQFactor[?] on page 266

:PLUGin:OLEVel:EVALuation:BERThresh[?]

Syntax :PLUGin:OLEVel:EVALuation:BERThresh[?] 'Identifier', <NRf>

Input Parameters Identifier: 'MyMeasurement'

Return Range 1E-15 to 1E-1

Description This command allows the user to specify the BER Threshold value against which the measurement parameters are calculated. The query returns current value of the BER threshold.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUG:OLEV:EVAL:BERT 'MyMeasurement', 1E-10

:PLUGin:OLEVel:EVALuation:MBQFactor[?]

Syntax :PLUGin:OLEVel:EVALuation:MBQFactor[?] 'Identifier', <NRf>

Input Parameters Identifier: 'MyMeasurement'

Range 1e-15...1.0

Description This command allows the user to specify the Minimum BER for Q-factor caluclations. Please note that in order to be able to do Q-Factor calculations the specified BER Threshold value must be greater than the value specified by this command.

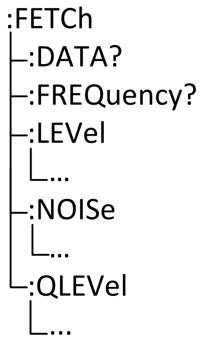
The query returns the minimum value specified to allow the Q-Factor calculations.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUG:OLEV:EVAL:MBQF 'MyMeasurement', 1E-15

:PLUGin:OLEVel:FETCh Subnode

This subnode has the following SCPI structure:



```
:FETCh
├:DATA?
├:FREQuency?
├:LEVel
│└...
├:NOISe
│└...
└:QLEVel
 └...
```

This subnode has the following commands and subnodes:

Table 34

Name	Description under
:DATA?	:PLUGin:OLEVel:FETCh:DATA? on page 268
:FREQuency?	:PLUGin:OLEVel:FETCh:FREQuency? on page 268
Subnode	
:LEVel	:PLUGin:OLEVel:FETCh:LEVel Subnode on page 269
:NOISe	:PLUGin:OLEVel:FETCh:NOISe Subnode on page 272
:QLEVel	:PLUGin:OLEVel:FETCh:QLEVel Subnode on page 275

:PLUGin:OLEVel:FETCh:DATA?

Syntax

:PLUGin:OLEVel:FETCh:DATA? 'identifier'[,'Location']

Input Parameters

'identifier': Specify the measurement name.

[,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).

Description

Return the raw data of the output level measurement.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example

A measurement run on location e.g 'M1.DataIn1' would return following results.

:PLUG:OLEV:FETC:DATA? 'MyMeasurement','M1.DataIn1'

Return: ("M1.DataIn1", ...

The first item in that comma separated list is the name of the location here "M1.DataIn1",the next values are TimeStamp, ComparedOnes, ComparedZeroes, ErroredOnes, ErroredZeroes and SampleThreshold and these 6 values will be repeated.

If the 'Location' is omitted and the measurement runs on multiple locations the response contains all results of current location.

(("M1.DataIn1", ...),("M1.DataIn2", ...))

:PLUGin:OLEVel:FETCh:FREQuency?

Syntax

:PLUGin:OLEVel:FETCh:FREQuency? 'identifier' [,'Location']

Input Parameters

'identifier': Specify the measurement name.

[,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).

Description

This command fetches frequency set for the whole module. It also prefixes the acquisition location.

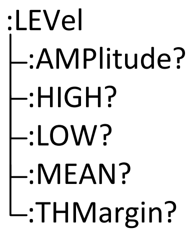
This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example

:plugin:olev:fetch:frequency? 'MyMeasurement'

:PLUGin:OLEVel:FETCh:LEVel Subnode

This subnode has the following SCPI structure:



```
:LEVel
├:AMPlitude?
├:HIGH?
├:LOW?
├:MEAN?
└:THMargin?
```

This subnode has the following commands:

Table 35

Name	Description under
:AMPlitude?	:PLUGin:OLEVel:FETCh:LEVel:AMPlitude? on page 270
:HIGH?	:PLUGin:OLEVel:FETCh:LEVel:HIGH? on page 270
:LOW?	:PLUGin:OLEVel:FETCh:LEVel:LOW? on page 270
:MEAN?	:PLUGin:OLEVel:FETCh:LEVel:MEAN? on page 271
:THMargin?	:PLUGin:OLEVel:FETCh:LEVel:THMargin? on page 271

:PLUGin:OLEVel:FETCh:LEVel:AMPlitude?

Syntax :PLUGin:OLEVel:FETCh:LEVel:AMPlitude? 'identifier' [,'Location']

Input Parameters 'identifier': Specify the measurement name.

[,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).

Description This command returns the value of amplitude calculated after a run.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUGin:OLEVel:FETCh:LEVel:AMPlitude? 'MyMeasurement'

:PLUGin:OLEVel:FETCh:LEVel:HIGH?

Syntax :PLUGin:OLEVel:FETCh:LEVel:HIGH? 'identifier' [,'Location']

Input Parameters 'identifier': Specify the measurement name.

[,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).

Description This command returns the value of the high level.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUGin:OLEVel:FETCh:LEVel:HIGH? 'MyMeasurement'

:PLUGin:OLEVel:FETCh:LEVel:LOW?

Syntax :PLUGin:OLEVel:FETCh:LEVel:LOW? 'identifier' [,'Location']

Input Parameters 'identifier': Specify the measurement name.

[,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).

Description This command returns the value of the low level.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUGin:OLEVel:FETCh:LEVel:LOW? 'MyMeasurement'

:PLUGin:OLEVel:FETCh:LEVel:MEAN?

Syntax :PLUGin:OLEVel:FETCh:LEVel:MEAN? 'identifier' [,'Location']

Input Parameters 'identifier': Specify the measurement name.

[,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).

Description This command returns the value of the mean level.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUGin:OLEVel:FETCh:LEVel:MEAN? 'MyMeasurement'

:PLUGin:OLEVel:FETCh:LEVel:THMargin?

Syntax :PLUGin:OLEVel:FETCh:LEVel:THMargin? 'identifier' [,'Location']

Input Parameters 'identifier': Specify the measurement name.

[,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).

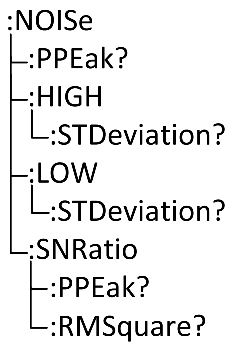
Description This command returns the value of the threshold margin.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUGin:OLEVel:FETCh:LEVel:THMargin? 'MyMeasurement'

:PLUGin:OLEVel:FETCh:NOISe Subnode

This subnode has the following SCPI structure:



```
:NOISe
├:PPEak?
├:HIGH
│└:STDeviation?
├:LOW
│└:STDeviation?
└:SNRatio
 ├:PPEak?
 └:RMSquare?
```

This subnode has the following commands:

Table 36

Name	Description under
:PPEak?	:PLUGin:OLEVel:FETCh:NOISe:PPEak? on page 273
:HIGH:STDeviation?	:PLUGin:OLEVel:FETCh:NOISe:HIGH:STDeviation? on page 273
:LOW:STDeviation?	:PLUGin:OLEVel:FETCh:NOISe:LOW:STDeviation? on page 273
:SNRation:PPEak?	:PLUGin:OLEVel:FETCh:NOISe:SNRatio:PPEak? on page 274
:SNRation:RNSquare?	:PLUGin:OLEVel:FETCh:NOISe:SNRatio:RMSquare? on page 274

:PLUGin:OLEVel:FETCh:NOISe:PPEak?

Syntax	:PLUGin:OLEVel:FETCh:NOISe:PPEak? 'identifier' [,'Location']
Input Parameters	'identifier': Specify the measurement name. [,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).
Description	This command returns the value of the peak to peak noise level. This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.
Example	:PLUGin:OLEVel:FETCh:NOISe:PPEak? 'MyMeasurement'

:PLUGin:OLEVel:FETCh:NOISe:HIGH:STDeviation?

Syntax	:PLUGin:OLEVel:FETCh:NOISe:HIGH:STDeviation? 'identifier' [,'Location']
Input Parameters	'identifier': Specify the measurement name. [,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).
Description	This command returns the value of the Q High Level Standard Deviation. This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.
Example	:PLUGin:OLEVel:FETCh:NOISe:HIGH:STDeviation? 'MyMeasurement'

:PLUGin:OLEVel:FETCh:NOISe:LOW:STDeviation?

Syntax	:PLUGin:OLEVel:FETCh:NOISe:LOW:STDeviation? 'identifier' [,'Location']
Input Parameters	'identifier': Specify the measurement name. [,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).
Description	This command returns the value of the Q Low Level Standard Deviation. This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.
Example	:PLUGin:OLEVel:FETCh:NOISe:LOW:STDeviation? 'MyMeasurement'

:PLUGin:OLEVel:FETCh:NOISe:SNRatio:PPEak?

Syntax :PLUGin:OLEVel:FETCh:NOISe:SNRatio:PPEak? 'identifier' [,'Location']

Input Parameters 'identifier': Specify the measurement name.

[,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).

Description This command returns the peak to peak value of the signal to noise ratio.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUGin:OLEVel:FETCh:NOISe:SNRatio:PPEak? 'MyMeasurement'

:PLUGin:OLEVel:FETCh:NOISe:SNRatio:RMSquare?

Syntax :PLUGin:OLEVel:FETCh:NOISe:SNRatio:RMSquare? 'identifier' [,'Location']

Input Parameters 'identifier': Specify the measurement name.

[,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).

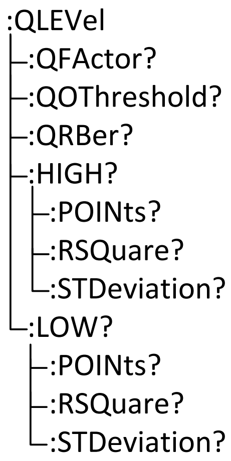
Description This command returns the value of the root mean square of signal to noise ratio.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUGin:OLEVel:FETCh:NOISe:SNRatio:RMSquare? 'MyMeasurement'

:PLUGin:OLEVel:FETCh:QLEVel Subnode

This subnode has the following SCPI structure:



```
:QLEVel
├:QFActor?
├:QOThreshold?
├:QRBer?
├:HIGH?
│├:POINts?
│├:RSQuare?
│└:STDeviation?
└:LOW?
 ├:POINts?
 ├:RSQuare?
 └:STDeviation?
```

This subnode has the following commands:

Table 37

Name	Description under
:QFActor?	:PLUGin:OLEVel:FETCh:QLEVel:QFActor? on page 276
:HIGH?	:PLUGin:OLEVel:FETCh:QLEVel:HIGH? on page 277
:HIGH:POINts?	:PLUGin:OLEVel:FETCh:QLEVel:HIGH:POINts? on page 277
:HIGH:RSQuare?	:PLUGin:OLEVel:FETCh:QLEVel:HIGH:RSQuare? on page 278
:HIGH:STDeviation?	:PLUGin:OLEVel:FETCh:QLEVel:HIGH:STDeviation? on page 278
:LOW?	:PLUGin:OLEVel:FETCh:QLEVel:LOW? on page 278

Name	Description under
:LOW:POINts?	:PLUGin:OLEVel:FETCh:QLEVel:LOW:POINts? on page 279
:LOW:RSQuare?	:PLUGin:OLEVel:FETCh:QLEVel:LOW:RSQuare? on page 279
:LOW:STDeviation?	:PLUGin:OLEVel:FETCh:QLEVel:LOW:STDeviation? on page 279

:PLUGin:OLEVel:FETCh:QLEVel:QFActor?

Syntax :PLUGin:OLEVel:FETCh:QLEVel:QFActor? 'identifier' [,'Location']

Input Parameters 'identifier': Specify the measurement name.

[,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).

Description This command returns the value of the Q factor.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUGin:OLEVel:FETCh:QLEVel:QFActor? 'MyMeasurement'

:PLUGin:OLEVel:FETCh:QLEVel:QOThreshold?

Syntax :PLUGin:OLEVel:FETCh:QLEVel:QOThreshold? 'identifier' [,'Location']

Input Parameters 'identifier': Specify the measurement name.

[,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).

Description This command returns the value of the Q Optimum threshold.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUGin:OLEVel:FETCh:QLEVel:QOThreshold? 'MyMeasurement'

:PLUGin:OLEVel:FETCh:QLEVel:QRBer?

Syntax	:PLUGin:OLEVel:FETCh:QLEVel:QRBer? 'identifier' [,'Location']
Input Parameters	'identifier': Specify the measurement name. [,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).
Description	This command returns the value of the Q Residual BER. This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.
Example	:PLUGin:OLEVel:FETCh:QLEVel:QRBer? 'MyMeasurement'

:PLUGin:OLEVel:FETCh:QLEVel:HIGH?

Syntax	:PLUGin:OLEVel:FETCh:QLEVel:HIGH? 'identifier' [,'Location']
Input Parameters	'identifier': Specify the measurement name. [,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).
Description	This command returns the value of the Q High Level. This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.
Example	:PLUGin:OLEVel:FETCh:QLEVel:HIGH? 'MyMeasurement'

:PLUGin:OLEVel:FETCh:QLEVel:HIGH:POINts?

Syntax	:PLUGin:OLEVel:FETCh:QLEVel:HIGH:POINts? 'identifier' [,'Location']
Input Parameters	'identifier': Specify the measurement name. [,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).
Description	This command returns the value of the Q High Level Points. This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.
Example	:PLUGin:OLEVel:FETCh:QLEVel:HIGH:POINts? 'MyMeasurement'

:PLUGin:OLEVel:FETCh:QLEVel:HIGH:RSQuare?

Syntax :PLUGin:OLEVel:FETCh:QLEVel:HIGH:RSQuare? 'identifier' [,'Location']

Input Parameters 'identifier': Specify the measurement name.

[,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).

Description This command returns the value of the Q High Level R^2.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUGin:OLEVel:FETCh:QLEVel:HIGH:RSQuare? 'MyMeasurement'

:PLUGin:OLEVel:FETCh:QLEVel:HIGH:STDeviation?

Syntax :PLUGin:OLEVel:FETCh:QLEVel:HIGH:STDeviation? 'identifier' [,'Location']

Input Parameters 'identifier': Specify the measurement name.

[,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).

Description This command returns the value of the Q High Level Standard Deviation.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUGin:OLEVel:FETCh:QLEVel:HIGH:STDeviation? 'MyMeasurement'

:PLUGin:OLEVel:FETCh:QLEVel:LOW?

Syntax :PLUGin:OLEVel:FETCh:QLEVel:LOW? 'identifier' [,'Location']

Input Parameters 'identifier': Specify the measurement name.

[,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).

Description This command returns the value of the Q Low Level.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUGin:OLEVel:FETCh:QLEVel:LOW? 'MyMeasurement'

:PLUGin:OLEVel:FETCh:QLEVel:LOW:POINts?

Syntax :PLUGin:OLEVel:FETCh:QLEVel:LOW:POINts? 'identifier' [,'Location']

Input Parameters 'identifier': Specify the measurement name.

[,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).

Description This command returns the value of the Q Low Level Points.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUGin:OLEVel:FETCh:QLEVel:LOW:POINts? 'MyMeasurement'

:PLUGin:OLEVel:FETCh:QLEVel:LOW:RSQuare?

Syntax :PLUGin:OLEVel:FETCh:QLEVel:LOW:RSQuare? 'identifier' [,'Location']

Input Parameters 'identifier': Specify the measurement name.

[,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).

Description This command returns the value of the Q Low Level R^2.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUGin:OLEVel:FETCh:QLEVel:LOW:RSQuare? 'MyMeasurement'

:PLUGin:OLEVel:FETCh:QLEVel:LOW:STDeviation?

Syntax :PLUGin:OLEVel:FETCh:QLEVel:LOW:STDeviation? 'identifier' [,'Location']

Input Parameters 'identifier': Specify the measurement name.

[,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).

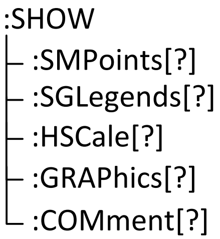
Description This command returns the value of the Q Low Level Standard Deviation.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUGin:OLEVel:FETCh:QLEVel:LOW:STDeviation? 'MyMeasurement'

:PLUGin:OLEVel:SHOW Subnode

This subnode has the following SCPI structure:



```
:SHOW
├ :SMPoints[?]
├ :SGLegends[?]
├ :HSCale[?]
├ :GRAPhics[?]
└ :COMment[?]
```

This subnode has the following commands:

Table 38

Name	Description under
:SMPoints[?]	:PLUGin:OLEVel:SHOW:SMPoints[?] on page 280
:GRAPhics[?]	:PLUGin:OLEVel:SHOW:GRAPhics[?] on page 281
:HSCale[?]	:PLUGin:OLEVel:SHOW:HSCale[?] on page 281
:SGLegends[?]	:PLUGin:OLEVel:SHOW:SGLegends[?] on page 282
:COMment[?]	:PLUGin:OLEVel:SHOW:COMment[?] on page 282

:PLUGin:OLEVel:SHOW:SMPoints[?]

Syntax :PLUGin:OLEVel:SHOW:SMPoints[?] 'Identifier', <0|1|ON|OFF>

Input Parameters Identifier: 'MyMeasurement'

Description This command allows the user to control the GUI if he/she wants to visualize the measurement points of the measurement run. Value of 0 indicates that user does not want to visualize the measurement points and

1 indicates that user wants to visualize the measurement points. The query returns current value.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUG:OLEV:SHOW:SMP 'MyMeasurement', 1

:PLUGin:OLEVel:SHOW:GRAPhics[?]

Syntax :PLUGin:OLEVel:SHOW:GRAPhics[?] 'Identifier', <BVTH | QFBV >

Input Parameters Identifier: 'MyMeasurement'

Description The command allows the user to specify the Graphics Type to Visualize. If BVTH is used then BER versus Threshold graphics is visualized and the Horizontal Axis Scale (see :PLUGin:OLEVel:SHOW:HSCale[?]) can be visualized in terms of Logarithmic or Linear Scale. If QFBV is used then QBER versus Threshold graphics is visualized and the horizontal axis scale is always linear. The query returns current value.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUG:OLEV:SHOW:GRAP 'MyMeasurement', BVTH

:PLUGin:OLEVel:SHOW:HSCale[?]

Syntax :PLUGin:OLEVel:SHOW:HSCale[?] 'Identifier', <LOG | LIN >

Input Parameters Identifier: 'MyMeasurement'

Description The command allows the user to specify the horizontal axis scale (i.e. BER scale) if BER versus Threshold graphics is visualized (see :PLUGin:OLEVel:SHOW:GRAPhics[?]), however if QBER versus threshold graphics is visualized then the Horizontal Axis Scale is always linear. If LOG is used then the BER values are in terms of Logarithmic scale and if the LIN is used then the BER values are in terms of linear scale .The query returns current value.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUGin:OLEVel:SHOW:HSC 'MyMeasurement',LOG

:PLUGin:OLEVel:SHOW:SGLegends[?]

Syntax
:PLUGin:OLEVel:SHOW:SGLegends 'Identifier', <0|1|OFF|ON>

:PLUGin:OLEVel:SHOW:SGLegends? 'Identifier'

Input Parameters
Identifier: 'MyMeasurement'

<0|1|OFF|ON>: Specify the show/hide state for graph legends.

Return Range
0|1

Description
This command shows/hides the graph legends.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example
:PLUGin:OLEVel:SHOW:SGLegends 'MyMeasurement', 1

:PLUGin:OLEVel:SHOW:COMment[?]

Syntax
:PLUGin:OLEVel:SHOW:COMment 'Identifier', <“Comment”>

:PLUGin:OLEVel:SHOW:COMment? 'Identifier'

Input Parameters
Identifier: Specify the measurement name. For example; 'MyMeasurement'

<“Comment”>: Enter the desired comment.

Description
This command is used to enter a comment for each measurement. This comment will be displayed on the **Measurement History** tab.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

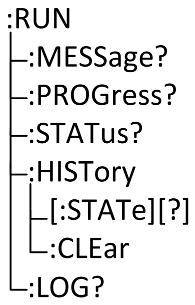
Example
:PLUG:OLEVel:SHOW:COMment 'MyMeasurement', “Output Level Example”

:PLUG:OLEVel:SHOW:COMment? 'MyMeasurement'

Output Level Example

:PLUGin:OLEVel:RUN Subnode

This subnode has the following SCPI structure:



```
:RUN
├:MESSage?
├:PROGress?
├:STATus?
├:HISTory
│├[:STATe][?]
│└:CLEar
└:LOG?
```

This subnode has the following commands:

Table 39

Name	Description under
:MESSage?	:PLUGin:OLEVel:RUN:MESSage? on page 284
:PROGress?	:PLUGin:OLEVel:RUN:PROGress? on page 284
:STATus?	:PLUGin:OLEVel:RUN:STATus? on page 284
:HISTory:[:STATe]	:PLUGin:OLEVel:RUN:HISTory[:STATe][?] on page 285
:HISTory:CLEar	:PLUGin:OLEVel:RUN:HISTory:CLEar on page 285
:LOG?	:PLUGin:OLEVel:RUN:LOG? on page 285

:PLUGin:OLEVel:RUN:MESSage?

Syntax	:PLUGin:OLEVel:RUN:MESSage? 'Identifier'
Input Parameters	Identifier: 'MyMeasurement'
Description	This command returns a string describing the state of an output level measurement addressed by the measurement name identifier. Possible states include NotStarted, Running, Suspended, Finished, Error, or Stopped. This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.
Example	:PLUGin:OLEVel:RUN:MESSage? 'MyMeasurement'

:PLUGin:OLEVel:RUN:PROGress?

Syntax	:PLUGin:OLEVel:RUN:PROGress? 'Identifier'
Input Parameters	Identifier: 'MyMeasurement'
Return Range	0 to 1
Description	This command returns a number in the range of 0.0 to 1.0 to indicate the progress of an output level measurement addressed by the measurement name identifier. A 0.0 indicates that the measurement has not started and 1.0 indicates the measurement is finished. This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.
Example	:PLUG:OLEV:RUN:PROGress? 'MyMeasurement'

:PLUGin:OLEVel:RUN:STATus?

Syntax	:PLUGin:OLEVel:RUN:STATus? 'Identifier'
Input Parameters	Identifier: 'MyMeasurement'
Return Range	0|1
Description	This command returns the running status of an output level measurement addressed by the measurement name identifier. A 0 indicates the measurement is not running and a 1 indicates the measurement is running.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUG:OLEV:RUN:STATus? 'MyMeasurement'

:PLUGin:OLEVel:RUN:HISTory[:STATe][?]

Syntax :PLUGin:OLEVel:RUN:HISTory[:STATe] 'Identifier', <0|1|OFF|ON>

:PLUGin:OLEVel:RUN:HISTory[:STATe]? 'Identifier'

Input Parameters Identifier: 'MyMeasurement'

Return Range 0|1

Description This command enables/disables the storage of output level measurement results addressed by the measurement name identifier.

This query returns the current setting.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUGin:OLEVel:RUN:HISTory[:STATe] 'MyMeasurement',1

:PLUGin:OLEVel:RUN:HISTory:CLEar

Syntax :PLUGin:OLEVel:RUN:HISTory:CLEar 'Identifier'

Input Parameters Identifier: 'MyMeasurement'

Description This command deletes the output level measurement history addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUGin:OLEVel:RUN:HISTory:CLEar 'MyMeasurement'

:PLUGin:OLEVel:RUN:LOG?

Syntax :PLUGin:OLEVel:RUN:LOG? 'Identifier'

Input Parameters Identifier: 'MyMeasurement'

Description This command returns logs for the addressed measurement.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example

```
:PLUGin:OLEVel:RUN:LOG? 'MyMeasurement'
```

Output:

The format of log returned is in the following way:

#XY Message Log

where X denotes the length of digits for Y, Y denotes the number of characters in the log then followed by the log message.

For example:

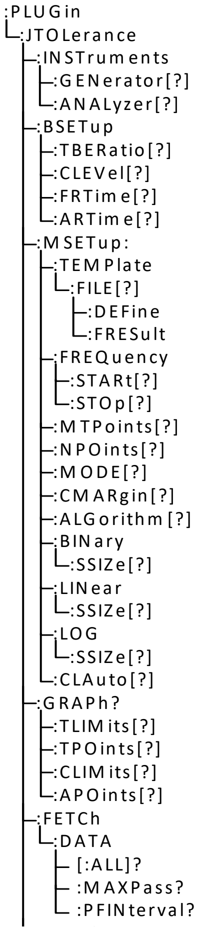
#10 means, there is 1 digit after 1 and there is 0 character in the log.

#2492 means, there are 3 digits after 2 and there are 492 characters in the log message.

:PLUGin:JTOLerance Subnode

The jitter tolerance commands are used to determine the capacity of a serial data receiver to tolerate jitter. There are two modes: compliance or characterization. Jitter frequency is set at multiple test points. At each point, the jitter amplitude is modulated using various search algorithms to determine the level of jitter which causes the device to fail. The M8040A system supports also jitter tolerance measurement using NRZ or PAM4.

This subnode has the following SCPI structure:



```
:PLUGin
└:JTOLerance
  ├:INSTruments
  │ ├:GENerator[?]
  │ └:ANALyzer[?]
  ├:BSETup
  │ ├:TBERatio[?]
  │ ├:CLEVel[?]
  │ ├:FRTime[?]
  │ └:ARTime[?]
  ├:MSETup:
  │ ├:TEMPlate
  │ │ └:FILE[?]
  │ │   ├:DEFine
  │ │   └:FRESult
  │ ├:FREQuency
  │ │ ├:STARt[?]
  │ │ └:STOp[?]
  │ ├:MTPoints[?]
  │ ├:NPOints[?]
  │ ├:MODE[?]
  │ ├:CMARgin[?]
  │ ├:ALGorithm[?]
  │ ├:BINary
  │ │ └:SSIZe[?]
  │ ├:LINear
  │ │ └:SSIZe[?]
  │ ├:LOG
  │ │ └:SSIZe[?]
  │ └:CLAuto[?]
  ├:GRAPh?
  │ ├:TLIMits[?]
  │ ├:TPOints[?]
  │ ├:CLIMits[?]
  │ └:APOints[?]
  ├:FETCh
  │ └:DATA
  │   ├ [:ALL]?
  │   ├ :MAXPass?
  │   └ :PFINterval?
  │
```

```
├:BREak
├:CATalog?
├:TABS?
├:EXPort
├:CONTinue
├:NEW
├:DELete
├:RESet
├:STARt
├:STOP
├:STEP
├:RUN
│ ├:HISTory
│ │ ├ [:STATe][?]
│ │ └ :CLEar
│ ├:LOG?
│ ├:MESSage?
│ ├:PROGress?
│ └[:STATus]?
└ :SHOW:COMment[?]
```

This subnode has the following commands:

Table 40

Name	Description under
:INSTruments:GENerator[?]	:PLUGin:JTOLerance:INSTruments:GENerator[?] on page 291
:INSTruments:ANALyzer[?]	:PLUGin:JTOLerance:INSTruments:ANALyzer[?] on page 291
:BSETup:TBERatio[?]	:PLUGin:JTOLerance:BSETup:TBERatio[?] on page 292
:BSETup:CLEVel[?]	:PLUGin:JTOLerance:BSETup:CLEVel[?] on page 292
:BSETup:FRTime[?]	:PLUGin:JTOLerance:BSETup:FRTime[?] on page 293
:BSETup:ARTime[?]	:PLUGin:JTOLerance:BSETup:ARTime[?] on page 293
:MSETup:TEMPlate:FILE[?]	:PLUGin:JTOLerance:MSETup:TEMPlate:FILE[?] on page 294
MSETup:TEMPlate:FILE:DEFine	:PLUGin:JTOLerance:MSETup:TEMPlate:FILE:DEFine on page 294

Name	Description under
:MSETup:TEMPlate:FILE:FRESult	:PLUGin:JTOLerance:MSETup:TEMPlate:FILE:FRESult on page 295
:MSETup:FREQuency:STARt[?]	:PLUGin:JTOLerance:MSETup:FREQuency:STARt[?] on page 297
:MSETup:FREQuency:STOp[?]	:PLUGin:JTOLerance:MSETup:FREQuency:STOp[?] on page 297
:MSETup:MTPoints[?]	:PLUGin:JTOLerance:MSETup:MTPoints[?] on page 298
:MSETup:NPOints[?]	:PLUGin:JTOLerance:MSETup:NPOints[?] on page 298
:MSETup:MODE[?]	:PLUGin:JTOLerance:MSETup:MODE[?] on page 298
:MSETup:CMARgin[?]	:PLUGin:JTOLerance:MSETup:CMARgin[?] on page 299
:MSETup:ALGorithm[?]	:PLUGin:JTOLerance:MSETup:ALGorithm[?] on page 300
:MSETup:BINary:SSIZe[?]	:PLUGin:JTOLerance:MSETup:BINary:SSIZe[?] on page 302
:MSETup:LINear:SSIZe[?]	:PLUGin:JTOLerance:MSETup:LINear:SSIZe[?] on page 303
:MSETup:LOG:SSIZe[?]	:PLUGin:JTOLerance:MSETup:LOG:SSIZe[?] on page 304
:MSETup:CLAuto[?]	:PLUGin:JTOLerance:MSETup:CLAuto[?] on page 304
:GRAPh?	:PLUGin:JTOLerance:GRAPh? on page 305
:GRAPh:TLIMits[?]	:PLUGin:JTOLerance:GRAPh:TLIMits[?] on page 305
:GRAPh:TPOints[?]	:PLUGin:JTOLerance:GRAPh:TPOints[?] on page 306
:GRAPh:CLIMits[?]	:PLUGin:JTOLerance:GRAPh:CLIMits[?] on page 306
:GRAPh:APOints[?]	:PLUGin:JTOLerance:GRAPh:APOints[?] on page 307
:FETCh:DATA?	:PLUGin:JTOLerance:FETCh:DATA[:ALL]? on page 307

Name	Description under
:FETCh:DATA:MAXPass?	:PLUGin:JTOLerance:FETCh:DATA:MAXPass? on page 307
:FETCh:DATA:PFINterval?	:PLUGin:JTOLerance:FETCh:DATA:PFINterval? on page 308
:BREak	:PLUGin:JTOLerance:BREak on page 309
:CATalog?	:PLUGin:JTOLerance:CATalog? on page 309
:TABS?	:PLUGin:JTOLerance:TABS? on page 310
:EXPort	:PLUGin:JTOLerance:EXPort on page 310
:CONTinue	:PLUGin:JTOLerance:CONTinue on page 311
:NEW	:PLUGin:JTOLerance:NEW on page 311
:DELete	:PLUGin:JTOLerance:DELete on page 312
:RESet	:PLUGin:JTOLerance:RESet on page 312
:STARt	:PLUGin:JTOLerance:STARt on page 312
:STOP	:PLUGin:JTOLerance:STOP on page 313
:STEP	:PLUGin:JTOLerance:STEP on page 313
:GRAPh?	:PLUGin:JTOLerance:GRAPh? on page 305
:RUN:HISTory[:STATe]	:PLUGin:JTOLerance:RUN:HISTory[:STATe][?] on page 314
:RUN:HISTory:CLEar	:PLUGin:JTOLerance:RUN:HISTory:CLEar on page 314
:RUN:LOG?	:PLUGin:JTOLerance:RUN:LOG? on page 315
:RUN:MESSage?	:PLUGin:JTOLerance:RUN:MESSage? on page 315
:RUN:PROGress?	:PLUGin:JTOLerance:RUN:PROGress? on page 316
:RUN[:STATus]?	:PLUGin:JTOLerance:RUN[:STATus]? on page 316
SHOW:COMment[?]	:PLUGin:JTOLerance:SHOW:COMment[?] on page 317

:PLUGin:JTOLerance:INSTruments:GENerator[?]

Syntax
:PLUGin:JTOLerance:INSTruments:GENerator 'identifier',<location-string>

:PLUGin:JTOLerance:INSTruments:GENerator? 'identifier'

Input Parameters
'identifier': Specify the measurement name.

<location-string>: Specify the location identifier or group name identifier.

Return Range
<location or group name>

Description
This command is used to select the pattern generator module/channel for the jitter tolerance measurement. The location-string refers to either a location name or a group name identifier associated with a measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example
:PLUG:JTOL:INST:GEN 'MyMeasurement','M1.DataOut1'

:PLUGin:JTOLerance:INSTruments:ANALyzer[?]

Syntax
:PLUGin:JTOLerance:INSTruments:ANALyzer 'identifier',<location-string>

:PLUGin:JTOLerance:INSTruments:ANALyzer? 'identifier'

Input Parameters
'identifier': Specify the measurement name.

<location-string>: Specify the location identifier or group name identifier.

Return Range
<location or group name>

Description
This command is used to select the analyzer (error detector) module/channel for the jitter tolerance measurement. The location-string refers to either a location name or a group name identifier associated with a measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example
:PLUG:JTOL:INST:ANAL 'MyMeasurement','M1.DataIn1'

:PLUGin:JTOLerance:BSETup:TBERatio[?]

Syntax

:PLUGin:JTOLerance:BSETup:TBERatio 'identifier', <NRf>

:PLUGin:JTOLerance:BSETup:TBERatio? 'identifier'

Input Parameters

'identifier': Specify the measurement name.

<NRf>: Set the target BER.

Return Range

1E-15 to 1E-1

Description

This command is used to set the target BER of the measurement specified by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example

:PLUG:JTOL:BSET:TBER 'MyMeasurement',1E-15

:PLUGin:JTOLerance:BSETup:CLEVel[?]

Syntax

:PLUGin:JTOLerance:BSETup:CLEVel 'identifier', <NRf>

:PLUGin:JTOLerance:BSETup:CLEVel? 'identifier'

Input Parameters

'identifier': Specify the measurement name.

<NRf>: Set the percent confidence interval.

Return Range

0.1 to 99.9

Description

This command is used to set the percent confidence interval of the measurement specified by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example

:PLUG:JTOL:BSET:CLEV 'MyMeasurement', 95%

:PLUGin:JTOLerance:BSETup:FRTime[?]

Syntax	:PLUGin:JTOLerance:BSETup:FRTime 'identifier', <NRf> :PLUGin:JTOLerance:BSETup:FRTime? 'identifier'
Input Parameters	'identifier': Specify the measurement name. <NRf>: Set the frequency relax time.
Return Range	0 ms to 60 s
Description	This command is used to set the frequency relax time after a change in jitter modulation frequency for the measurement specified by the measurement name identifier. This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.
Example	:PLUG:JTOL:BSET:FRT 'MyMeasurement',30s

:PLUGin:JTOLerance:BSETup:ARTime[?]

Syntax	:PLUGin:JTOLerance:BSETup:ARTime 'identifier', <NRf> :PLUGin:JTOLerance:BSETup:ARTime? 'identifier'
Input Parameters	'identifier': Specify the measurement name. <NRf>: Set the amplitude relax time.
Return Range	0 ms to 60 s
Description	This command is used to set the amplitude relax time after a change in jitter modulation amplitude for the measurement specified by the measurement name identifier. This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.
Example	:PLUG:JTOL:BSET:ART 'MyMeasurement', 30s

:PLUGin:JTOLerance:MSETup:TEMPlate:FILE[?]

Syntax

:PLUGin:JTOLerance:MSETup:TEMPlate:FILE 'identifier', <filepath-string>

:PLUGin:JTOLerance:MSETup:TEMPlate:FILE? 'identifier'

Input Parameters

'identifier': Specify the measurement name.

<filepath-string>: Specify path/filename of template to load.

Return Range

Path/filename of current measurement template.

Description

This command is used to specify the path/filename of the measurement template to load for the measurement specified by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example

:PLUGin:JTOLerance:MSETup:TEMPlate:FILE "MyMeasurement","Shared/USB_1.jtt"

:PLUGin:JTOLerance:MSETup:TEMPlate:FILE:DEFine

Syntax

:PLUGin:JTOLerance:MSETup:TEMPlate:FILE:DEFine "Identifier","templateName","templatePoints"

Input Parameters

'identifier': Specify the measurement name.

<filepath-string>: Specify path/filename of template to load.

Return Range

Path/filename of current measurement template.

Description

This command creates a template file and selects it immediately in the given jitter tolerance measurement.

The jitter amplitude values are automatically limited to a minimum of 0.1mUI.

templateName: Specifies the name and location of the template file.

Structure is "Shared|Current:name.jtt" alternatively "Shared|Current/name.jtt"

Examples:

"Shared:template1.jtt"

"Shared/template1.jtt"

"Current:template2.jtt"

"Current/template2.jtt"

Template Points

A list of tuples, containing values for frequency, min limit, compliance value and max limit with the following structure:

(tuple,tuple,...)

-

Each tuple has the following structure:

(frequencyHz,minLimitUI,complianceLevelUI,maxLimitUI)

Example: ((100,500,750,1000),(100e3,1,10,500),(500e6,1e-3,1,1))

This defines the following points:

Frequency	Minimum Limit	Compliance Value	Maximum Limit
100 Hz	500 UI	750 UI	1000 UI
100 kHz	1 UI	10 UI	500 UI
500 MHz	1 mUI	1 UI	1 UI

The frequency value are clipped to a resolution of 1Hz and the jitter amplitudes are clipped to a minimum value of 0.1mUI

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example

```
:PLUGin:JTOLerance:MSETup:TEMPlate:FILE:DEFine "jitter tolerance 1", "Shared:template.jtt","((100,500,700,1000),(100e3,0.1,1,500),(100e6,0.05, 1,1000))"
```

:PLUGin:JTOLerance:MSETup:TEMPlate:FILE:FRESult

Syntax

```
:PLUGin:JTOLerance:MSETup:TEMPlate:FILE:FRESult "Indentifier", "templateName", [[[[minMarginPct], minMarginUI], maxMarginPct], maxMarginUI]
```

Description

This command creates a template file and immediately selects it in the given jitter tolerance measurement.

The results of the last measurement run define points in the template file.

If a jitter frequency has not resulted in a passed point then the template will be set to:

Minimum	1.0 mUI
Confidence Level	1.0 UI
Maximum	Current instrument limit

If the jitter frequency has resulted in a passed point, then the template will be set to:

Minimum	The smaller value of ▪ (Maximum passed jitter amplitude) * minMarginPct/100 ▪ (Maximum passed jitter amplitude) – abs(minMarginUI)
Confidence Level	Maximum passed jitter amplitude
Maximum	The larger value of ▪ (Maximum passed jitter amplitude) * maxMarginPct/100 ▪ (Maximum passed jitter amplitude) + abs(maxMarginUI)

Defaults and value ranges for the optional arguments are:

	Default	Minimum	Maximum
minMarginPct	50	0	100
minMarginUI	0.25	0	10000
maxMarginPct	110	100	200
maxMarginUI	0.25	0	10000

The resulting limit values will be clipped to a minimum of 0.1mUI.

:PLUGin:JTOLerance:MSETup:FREQuency:STARt[?]

Syntax

```
:PLUGin:JTOLerance:MSETup:FREQuency:STARt 'identifier', <NRf>
:PLUGin:JTOLerance:MSETup:FREQuency:STARt? 'identifier'
```

Input Parameters

'identifier': Specify the measurement name.

<NRf>: Set the start frequency.

Return Range

100 Hz to 500 MHz

Description

This command is used to set the start (minimum) modulation frequency for the measurement specified by the measurement name identifier. The default is 100 kHz.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example

```
:PLUG:JTOL:MSET:FREQ:STAR 'MyMeasurement', 10kHz
```

:PLUGin:JTOLerance:MSETup:FREQuency:STOp[?]

Syntax

```
:PLUGin:JTOLerance:MSETup:FREQuency:STOp 'identifier', <NRf>
:PLUGin:JTOLerance:MSETup:FREQuency:STOp? 'identifier'
```

Input Parameters

'identifier': Specify the measurement name.

<NRf>: Set the stop frequency.

Return Range

100 Hz to 500 MHz

Description

This command is used to set the stop (maximum) modulation frequency for the measurement specified by the measurement name identifier. The default is 100 MHz.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example

```
:PLUG:JTOL:MSET:FREQ:STO 'MyMeasurement', 100MHz
```

:PLUGin:JTOLerance:MSETup:MTPoints[?]

Syntax :PLUGin:JTOLerance:MSETup:MTPoints[?] 'identifier', <NRf>

Input Parameters 'identifier': Specify the measurement name.

Return Range 0 | 1 OFF | ON

Description This command selects either template points or specified range.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example :PLUGin:JTOLerance:MSETup:MTPoints 'MyMeasurement'

:PLUGin:JTOLerance:MSETup:NPOints[?]

Syntax :PLUGin:JTOLerance:MSETup:NPOints 'identifier', <NRf>

:PLUGin:JTOLerance:MSETup:NPOints? 'identifier'

Input Parameters 'identifier': Specify the measurement name.

<NRf>: Set the number of measurement points.

Return Range 1 to 100

Description This command is used to set the number of measurement points for the measurement specified by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example :PLUG:JTOL:MSET:NPO 'MyMeasurement', 50

:PLUGin:JTOLerance:MSETup:MODE[?]

Syntax :PLUGin:JTOLerance:MSETup:MODE 'identifier', <COMPliance|CHARacterize>

:PLUGin:JTOLerance:MSETup:MODE? 'identifier'

Input Parameters 'identifier': Specify the measurement name.

<COMPliance|CHARacterize>: Specify the measurement mode.

Return Range COMPliance|CHARacterize

Description This command is used to specify compliance or characterization as the measurement mode for the measurement specified by the measurement name identifier. Compliance is used for checking that a device passes the minimum margin required for compliance to a standard. Characterization tests a device for tolerance to jitter to establish the pass/fail amplitude at each measurement point.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example :PLUG:JTOL:MSET:MODE 'MyMeasurement',CHAR

:PLUGin:JTOLerance:MSETup:CMARgin[?]

Syntax :PLUGin:JTOLerance:MSETup:CMARgin 'identifier', <NRf>

:PLUGin:JTOLerance:MSETup:CMARgin? 'identifier'

Input Parameters 'identifier': Specify the measurement name.

<NRf>: Specify the compliance margin.

Return Range -75% to 1000%

Description This command is used to specify the compliance margin for the measurement specified by the measurement name identifier. The default is 0%.

Compliance is used for checking that a device passes the minimum margin required for compliance to a standard. Characterization tests a device for tolerance to jitter to establish the pass/fail amplitude at each measurement point.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example :PLUG:JTOL:MSET:CMAR 'MyMeasurement', 10

:PLUGin:JTOLerance:MSETup:ALGorithm[?]

Syntax

:PLUGin:JTOLerance:MSETup:ALGorithm 'identifier', <BINary|BDLInear|BULInear|DLINear|DLOGarithmic|ULINear|ULOGarithmic|ULL|ABIN|AdaptiveBINary|CUSTom>

:PLUGin:JTOLerance:MSETup:ALGorithm? 'identifier'

Input Parameters

'identifier': Specify the measurement name.

<BINary|BDLInear|BULInear|DLINear|DLOGarithmic|ULINear|ULOGarithmic|ULL|ABIN|AdaptiveBINary|CUSTom>: Specify the measurement algorithm.

Return Range

BINary|BDLinear|BULinear|DLINear|DLOGarithmic|ULINear|ULOGarithmic|ULL|ABIN|AdaptiveBINary|CUSTom

Description

This command is used to specify the measurement algorithm during characterization measurements for the measurement specified by the measurement name identifier.

BINary: The Binary algorithm uses a variable step size to find the highest passing SJ amplitude at each SJ frequency in the template. Starting at the maximum SJ amplitude determined by the upper limit in the template file or the instrument limit, whichever is lower, followed by the minimum SJ amplitude determined by the lower limit in the template file, the Binary algorithm then sets subsequent SJ amplitudes at the logarithmic midpoint between the highest passing and lowest failing SJ amplitudes. A test point is determined to be a passing point if the BER measured is below the BER threshold set by the user.

BDLinear: The Binary + Down Linear algorithm is the same as the Binary algorithm followed by the Down Linear algorithm. Refer to the descriptions for Binary and Down Linear.

BULinear: The Binary + Up Linear algorithm is the same as the Binary algorithm followed by the Up Linear algorithm. Refer to the descriptions for Binary and Up Linear.

DLINear: The Down Linear search algorithm starts at the maximum jitter value determined by the template.

If a BER measurement has errors above the measurement threshold, the jitter amplitude is adjusted lower by the linear step size.

This algorithm exits when the BER is measured below the measurement threshold BER, or if the last amplitude measurement point was at the minimum jitter value in the template.

The jitter tolerance result at each jitter modulation frequency is the highest jitter amplitude at which the measurement was made below the threshold BER. However, if the device is intolerant of even the minimum jitter value, the result is not valid and is not plotted.

DLOGarithmic: The Down Logarithmic search algorithm starts at the maximum jitter value determined by the template.

If a BER measurement has errors above the measurement threshold, the jitter amplitude is adjusted lower by the coefficient calculated from the logarithmic step size. For example, if the coefficient is 10%, then the next amplitude equals the previous amplitude minus 10% of the previous amplitude.

This algorithm exits when the BER is measured below the measurement threshold BER, or if the last amplitude measurement point was at the minimum jitter value in the template.

The jitter tolerance result at each jitter modulation frequency is the highest jitter amplitude at which the measurement was made below the threshold BER. However, if the device is intolerant of even the minimum jitter value, the result is not valid and is not plotted.

ULINear: The Up Linear search algorithm starts at the minimum jitter value determined by the template.

If a BER measurement has errors below the measurement threshold, the jitter amplitude is adjusted higher by the linear step size.

This algorithm exits when BER is measured above the measurement threshold BER, or if the last amplitude measurement point was at the max jitter value in the template.

The jitter tolerance result at each jitter modulation frequency is the highest jitter amplitude at which the measurement was made below the threshold BER. However, if the device is intolerant of even the minimum jitter value, the result is not valid and is not plotted.

ULOGarithmic: The Up Logarithmic search algorithm starts at the minimum jitter value determined by the template.

If a BER measurement has errors below the measurement threshold, the jitter amplitude is adjusted higher by the coefficient calculated from the logarithmic step size. For example, if the coefficient is 10%, then the next amplitude equals the previous amplitude plus 10% of the previous amplitude.

This algorithm exits when the BER is measured above the measurement threshold BER, or if the last amplitude measurement point was at the maximum jitter value determined by the template.

Up Log + Linear: The Up Log + Linear search algorithm performs the same algorithm as Up Logarithmic, but in addition, it returns to the last passing amplitude and steps linearly up until it reaches a fail point. The algorithm continues to increase the SJ amplitude until failure, even if it surpassed the original failed point.

The jitter tolerance result at each jitter modulation frequency is the highest jitter amplitude at which the measurement was made below the threshold BER. However, if the device is intolerant of even the minimum jitter value, the result is not valid and is not plotted.

Adaptive Binary: The Adaptive Binary search algorithm is a mix of the Up-Logarithmic and Binary and works best when the jitter frequency is being swept from high to low frequencies. It will increase/decrease the jitter amplitude using the logarithmic step size until it finds a pass/fail transition. Then reduce the step size until it becomes less than the defined binary step size. The minimum value of the active template and the maximum passed amplitude of the previous jitter frequency define the initial jitter amplitude.

Custom: The Custom search algorithm requires the DUT Control Interface license (M8070B-1TP, M8070B-1NP or M8070B-1TT) and allow to define the jitter amplitude by implementing 3 functions in the DUT Control Interface.
The tutorial script 'JTOL-Custom-Algorithm-with-Virtual-DUT.py' demonstrates how to implement the algorithm correctly. This script is a part of the M8070B installation.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example :PLUG:JTOL:MSET:ALG 'MyMeasurement',BIN

:PLUGin:JTOLerance:MSETup:BINary:SSIZe[?]

Syntax :PLUGin:JTOLerance:MSETup:BINary:SSIZe 'identifier', <NRf>

:PLUGin:JTOLerance:MSETup:BINary:SSIZe? 'identifier'

Input Parameters 'identifier': Specify the measurement name.

<NRf>: Set the step size.

Return Range 2 mUI to 5 UI

Description This command is used to set the step size for the binary algorithm for the measurement specified by the measurement name identifier. The step size defines the exit criteria for the algorithm. The Binary search algorithm stops once its step size falls below this user defined Step Size.

The Binary Step Size applies to the Binary algorithm and the binary portions of the Binary + Down Linear and Binary + Up Linear algorithms.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example :PLUG:JTOL:MSET:BIN:SSIZ 'MyMeasurement', 100mUI

:PLUGin:JTOLerance:MSETup:LINear:SSIZe[?]

Syntax :PLUGin:JTOLerance:MSETup:LINear:SSIZe 'identifier', <NRf>

:PLUGin:JTOLerance:MSETup:LINear:SSIZe? 'identifier'

Input Parameters 'identifier': Specify the measurement name.

<NRf>: Set the step size.

Return Range 5 mUI to 100 UI

Description This command is used to set the step size for each step of the Up Linear and Down Linear algorithms for the measurement specified by the measurement name identifier.

When Down Linear is enabled as the search algorithm, the measurement starts from the maximum (which depends on the jitter frequency). A step size of 100 mUI, for example, may result in a sequence of 100 UI, 99.9 UI, 99.8 UI and so on. The test for one frequency stops when the BER limit is met or zero amplitude is reached.

When Up Linear is enabled as the search algorithm, the measurement starts from the minimum. A step size of 100 mUI, for example, will result in a sequence of 0.1 UI, 0.2 UI, 0.3 UI and so on. The test for one frequency stops when the BER limit is exceeded or the maximum amplitude is reached.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example :PLUG:JTOL:MSET:LIN:SSIZe 'MyMeasurement', 100mUI

:PLUGin:JTOLerance:MSETup:LOG:SSIZe[?]

Syntax

```
:PLUGin:JTOLerance:MSETup:LOG:SSIZe 'identifier', <NRf>
```

```
:PLUGin:JTOLerance:MSETup:LOG:SSIZe? 'identifier'
```

Input Parameters

'identifier': Specify the measurement name.

<NRf>: Set the step size.

Return Range

0.1% to 50%

Description

This command is used to set the step size for each step of the Up Logarithmic and Down Logarithmic algorithms for the measurement specified by the measurement name identifier.

When Down Logarithmic is enabled as the search algorithm, the measurement starts from the maximum (which depends on the jitter frequency). A percentage of 50%, for example, may result in a sequence of 100 UI, 50 UI, 25 UI and so on. The test for one frequency stops when the BER limit is met or the specified minimum amplitude is reached.

When Up Logarithmic is enabled as the search algorithm, the measurement starts from the specified minimum amplitude. A percentage of 50%, for example, may result in a sequence of 0.1 UI, 0.15 UI, 0.23 UI and so on. The test for one frequency stops when the BER limit is crossed or the maximum amplitude (which depends on the jitter frequency) is reached.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example

```
:PLUG:JTOL:MSET:LOG:SSIZe 'MyMeasurement', 25
```

:PLUGin:JTOLerance:MSETup:CLAuto[?]

Syntax

```
:PLUGin:JTOLerance:MSETup:CLAuto 'identifier', <ON|OFF|1|0>
```

```
:PLUGin:JTOLerance:MSETup:CLAuto? 'identifier'
```

Input Parameters

'identifier': 'Specify the measurement name.

<ON|OFF|1|0>: Enable/disable auto loop bandwidth.

Return Range

1|0

Description

This command sets the external CDR loop bandwidth selection during a Jitter Tolerance measurement. If AUTO state is ON, will set the CDR LBW based on the jitter frequency. If AUTO state is OFF, CDR LBW is not altered during the measurement. The query returns the current auto mode state.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example :PLUG:JTOL:MSET:CLA 'MyMeasurement',ON

:PLUGin:JTOLerance:GRAPh?

Syntax :PLUGin:JTOLerance:GRAPh? 'identifier' [,DesiredWidth][,DesiredHeight][,<CURR | WHIT>][,<PNG>][,1 | 0][,'TabName']

Input Parameters 'identifier': Specify measurement name.

Other optional parameters are:

[,DesiredWidth]: Specify the desired width of the image.

[,DesiredHeight]: Specify the desired height of the image.

[,<CURR | WHIT>]: Specify whether the user wants to capture in current theme or wants to capture in white background.

[,<PNG>]: Specify the format of the image. The default format is PNG.

[,1 | 0]: Specify whether to capture the graph with legends or not.

[,'TabName']: Specify the tab name on which the graph is supposed to be captured.

Description This query returns data of the image captured from the graph view of the plugin in the specified format.

Example :PLUG:JTOL:GRAP? 'MyMeasurement',1000,800,CURR,PNG,1,'Graph'

:PLUGin:JTOLerance:GRAPh:TLIMits[?]

Syntax :PLUGin:JTOLerance:GRAPh:TLIMits 'identifier', <ON|OFF|1|0>

:PLUGin:JTOLerance:GRAPh:TLIMits? 'identifier'

Input Parameters 'identifier': Specify the measurement name.

<ON|OFF|1|0>: Enable/disable template limit lines.

Return Range 1|0

Description This command enables/disables the template limits in the graph for the measurement specified by the measurement name identifier. These limits show the search range amplitude (UI) during characterization defined using the template editor in the M8070B software interface.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example :PLUG:JTOL:GRAP:TLIM 'MyMeasurement', ON

:PLUGin:JTOLerance:GRAPh:TPOints[?]

Syntax :PLUGin:JTOLerance:GRAPh:TPOints 'identifier', <ON|OFF|1|0>

:PLUGin:JTOLerance:GRAPh:TPOints? 'identifier'

Input Parameters 'identifier': Specify the measurement name.

<ON|OFF|1|0>: Enable/disable template test points.

Return Range 1|0

Description This command enables/disables the template measurement points in the graph for the measurement specified by the measurement name identifier. These test points are defined using the template editor in the M8070B software interface.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example :PLUG:JTOL:GRAP:TPO 'MyMeasurement', ON

:PLUGin:JTOLerance:GRAPh:CLIMits[?]

Syntax :PLUGin:JTOLerance:GRAPh:CLIMits 'identifier', <ON|OFF|1|0>

:PLUGin:JTOLerance:GRAPh:CLIMits? 'identifier'

Input Parameters 'identifier': Specify the measurement name.

<ON|OFF|1|0>: Enable/disable compliance limits.

Return Range 1|0

Description This command enables/disables the compliance limits in the graph for the measurement specified by the measurement name identifier. These limits are defined using the template editor in the M8070B software interface.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example :PLUG:JTOL:GRAP:CLIM 'MyMeasurement', ON

:PLUGin:JTOLerance:GRAPh:APOints[?]

Syntax

```
:PLUGin:JTOLerance:GRAPh:APOints 'identifier',{ON|OFF|1|0}
:PLUGin:JTOLerance:GRAPh:APOints? 'identifier'
```

Input Parameters

'identifier': Specify measurement name.

ON | OFF | 1 | 0

Description

This command enables or disables the show all measured points option on the graph.

This query returns the present settings.

Example

```
:PLUGin:JTOLerance:GRAPh:APOints 'Jitter Tolerance 1', ON
```

:PLUGin:JTOLerance:FETCh:DATA[:ALL]?

Syntax

```
:PLUGin:JTOLerance:FETCh:DATA? 'identifier'
```

Input Parameters

'identifier': Specify the measurement name.

Description

This command returns the jitter tolerance measurement results addressed by the measurement name identifier.
"M1.DataIn1",100000,0.01,3000000000,0,0,"PASS"

The first item in the comma separated list is the name of the location which is “M1.DataIn1”. The next values are Frequency, Amplitude, Number of Bits, Number of Errors, BER and Pass/Fail Results and are repeated in the same order for subsequent measured points.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example

```
:PLUG:JTOL:FETC:DATA? 'MyMeasurement'
```

:PLUGin:JTOLerance:FETCh:DATA:MAXPass?

Syntax

```
:PLUGin:JTOLerance:FETCh:DATA:MAXPass? 'Identifier'
```

Input Parameters

'identifier': Specify the measurement name.

Description

This query returns the maximum passed amplitude information for each measured jitter frequency. However, if a jitter frequency has not achieved a passed result, then the minimum failed amplitude information is being returned for the specific jitter frequency.

The response is build up as a list of location and frequency results enclosed in '(' and ')'. The frequency results are reporting the jitter frequency in Hz and a list of jitter amplitude records enclosed in '(' and ')'.

The jitter amplitude records reports the jitter amplitude in UI, number of compared bits, number of errors, bit error ratio and the PASS/FAIL indication as a comma separated list enclosed by '(' and ')'.

The response format is identical to the SCPI command :PLUGin:JTOLerance:FETCh:DATA:PFINterval with the difference that only one amplitude record is reported per jitter frequency.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example

:PLUGin:JTOLerance:FETCh:DATA:MAXPass? 'jitter tolerance 1'

Response

("M2.DataIn",(100,((100,3290274304,0,0,"PASS"))),(500000000,((0.11326875,3554441216,0,0,"PASS"))))

The tutorial script 'JTOL-parse-results.py' demonstrates how to parse the response for further processing.

:PLUGin:JTOLerance:FETCh:DATA:PFINterval?

Syntax

:PLUGin:JTOLerance:FETCh:DATA:PFINterval? 'Identifier'

Input Parameters

'identifier': Specify the measurement name.

Description

This query returns the maximum passed and minimum failed amplitude information for each measured jitter frequency. However, if a jitter frequency has not achieved a passed result, then the minimum failed amplitude information is being returned twice and if there is no failed jitter amplitude, then the maximum passed amplitude is being returned twice.

The response is build up as a list of location and frequency results enclosed in '(' and ')'. The frequency results are reporting the jitter frequency in Hz and a list of jitter amplitude records enclosed in '(' and ')'.

The jitter amplitude records reports the jitter amplitude in UI, number of compared bits, number of errors, bit error ratio and the PASS/FAIL indication as a comma separated list enclosed by '(' and ')'.

The response format is identical to the SCPI command :PLUGin:JTOLerance:FETCh:DATA:MAXPass with the difference that 2 amplitude records are reported per jitter frequency.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example :PLUGin:JTOLerance:FETCh:DATA:PFINterval? 'jitter tolerance 1'

Response ("M2.DataIn",(100,((100,3290274304,0,0,"PASS"),(100,3290274304,0,0,"PASS"))),(500000000,((0.11326875,3554441216,0,0,"PASS"),(0.151025,39507456,11,2.78428456643728E-07,"FAIL"))))

The tutorial script 'JTOL-parse-results.py' demonstrates how to parse the response for further processing.

:PLUGin:JTOLerance:BREak

Syntax :PLUGin:JTOLerance:BREak 'identifier'

Input Parameters 'identifier': Specify the measurement name to break/pause.

Description This command breaks/pauses a measurement addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example The following example breaks a jitter tolerance measurement with the measurement name identifier called 'MyMeasurement':

```
:PLUG:JTOL:BRE 'MyMeasurement'
```

:PLUGin:JTOLerance:CATalog?

Syntax :PLUGin:JTOLerance:CATalog?

Description This command returns a list of all created jitter tolerance measurement names currently available for measuring.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example Assume the following is a list of created jitter tolerance measurement names:

```
:PLUG:JTOL:NEW 'JTOL_1'
:PLUG:JTOL:NEW 'JTOL_2'
:PLUG:JTOL:NEW 'JTOL_3'
```

The command and returned list would look like the following:
:PLUG:JTOL:CAT?

"JTOL_1,JTOL_2,JTOL_3"

:PLUGin:JTOLerance:TABS?

Syntax :PLUGin:JTOLerance:TABS?

Description This query returns a list of all results tabs which can be exported as a CSV file.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example :PLUG:JTOL:TABS?

Output "Results"

:PLUGin:JTOLerance:EXPort

Syntax :PLUGin:JTOLerance:EXPort 'Identifier', 'TabName', 'FilePath'

Description This command exports the measurement data of the selected results tab to a csv file.

Input Parameters

- 'Identifier': Specify the measurement name; for example ''Jitter Tolerance 1''.
- 'TabName' – Specify the results tab name. To fetch the results tabs for a measurement, refer to the query ":PLUGin:JTOLerance:TABS? on page 310". However, if you want all tabs to be saved as a csv file, use " " (single/double quotes).
- 'FilePath' – Specify the location where the CSV file should be saved.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

Examples Example 1: The following command will save the 'Results' tab of jitter tolerance measurement to *Results.csv*:

:PLUGin:JTOL:EXPort 'Jitter Tolerance 1', 'Results', 'C:\Results.csv'

Example 2: The following command will save all results tab of jitter tolerance measurement to *Results.csv*:

:PLUGin:JTOL:EXPort 'Jitter Tolerance 1', ' ', 'C:\Results.csv'

:PLUGin:JTOLerance:CONTinue

Syntax

```
:PLUGin:JTOLerance:CONTinue 'identifier'
```

Input Parameters

'identifier': Specify the measurement name to continue.

Description

This command allows you to continue a jitter tolerance measurement, addressed by the measurement name identifier, that was stopped using the BREak command.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example

The following example continues a halted jitter tolerance measurement addressed with the measurement name identifier called 'MyMeasurement':

```
:PLUG:JTOL:CONT 'MyMeasurement'
```

:PLUGin:JTOLerance:NEW

Syntax

```
:PLUGin:JTOLerance:NEW 'identifier'
```

Input Parameters

'identifier': Specify the measurement name.

Description

This command creates a new jitter tolerance measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example

The following example creates a jitter tolerance measurement name identifier called 'MyMeasurement':

```
:PLUG:JTOL:NEW 'MyMeasurement'
```

NOTE

Creating multiple plugins using this command may slow down the GUI operations which may also result delay in remote programming. To prevent the plugin from opening automatically in the GUI, it is recommended to use “0” as a parameter input in this command.

Example- :PLUG:JTOL:NEW 'MyMeasurement',0

:PLUGin:JTOLerance:DELete

Syntax :PLUGin:JTOLerance:DELete 'identifier'

Input Parameters 'identifier': Specify the measurement name to delete.

Description This command deletes a previously created jitter tolerance measurement addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example The following example deletes a jitter tolerance measurement addressed by the measurement name identifier called 'MyMeasurement':

```
:PLUG:JTOL:DEL 'MyMeasurement'
```

:PLUGin:JTOLerance:RESet

Syntax :PLUGin:JTOLerance:RESet 'identifier'

Input Parameters 'identifier': Specify the measurement name to reset.

Description This command resets a jitter tolerance measurement addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example The following example resets a jitter tolerance measurement addressed with the measurement name identifier called 'MyMeasurement':

```
:PLUG:JTOL:RES 'MyMeasurement'
```

:PLUGin:JTOLerance:STARt

Syntax :PLUGin:JTOLerance:STARt 'identifier'

Input Parameters 'identifier': Specify the measurement name to start.

Description This command starts a jitter tolerance measurement addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example The following example starts a jitter tolerance measurement addressed with the measurement name identifier called 'MyMeasurement':

```
:PLUG:JTOL:STAR 'MyMeasurement'
```

:PLUGin:JTOLerance:STOP

Syntax
```
:PLUGin:JTOLerance:STOP 'identifier'
```

Input Parameters 'identifier': Specify the measurement name to stop.

Description This command stops a jitter tolerance measurement addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example The following example stops a jitter tolerance measurement with the measurement name identifier called 'MyMeasurement':

```
:PLUG:JTOL:STOP 'MyMeasurement'
```

:PLUGin:JTOLerance:STEP

Syntax
```
:PLUGin:JTOLerance:STEP 'identifier'
```

Input Parameters 'identifier': Specify the measurement name to step.

Description This command allows you to step through a jitter tolerance measurement, addressed by the measurement name identifier, that was stopped using the BREak command.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example The following example triggers a single jitter tolerance measurement step of the measurement addressed with the measurement name identifier called 'MyMeasurement':

```
:PLUG:JTOL:STEP 'MyMeasurement'
```

:PLUGin:JTOLerance:RUN:HISTory[:STATe][?]

Syntax

```
:PLUGin:JTOLerance:RUN:HISTory[:STATe] 'identifier',<0|1>
:PLUGin:JTOLerance:RUN:HISTory[:STATe]? 'identifier'
```

Input Parameters 'identifier': Specify the measurement name.

Return Range 0|1

Description This command enables/disables the storage of jitter tolerance measurement results addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example The following example enables storage of jitter tolerance measurement results with the measurement name identifier called 'MyMeasurement':

```
:PLUG:JTOL:RUN:HIST 'MyMeasurement', 1
```

:PLUGin:JTOLerance:RUN:HISTory:CLEar

Syntax

```
:PLUGin:JTOLerance:RUN:HISTory:CLEar 'identifier'
```

Input Parameters 'identifier': Specify the measurement name.

Description This command deletes the stored jitter tolerance measurement history addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example The following example deletes the stored jitter tolerance measurement history with the measurement name identifier called 'MyMeasurement':

```
:PLUG:JTOL:RUN:HIST:CLE 'MyMeasurement'
```

:PLUGin:JTOLerance:RUN:LOG?

Syntax

:PLUGin:JTOLerance:RUN:LOG? 'identifier'

Input Parameters

'identifier': Specify the measurement name.

Description

This command returns logs for the addressed measurement.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example

The following example returns logs for the measurement name identifier called 'MyMeasurement':

:PLUG:JTOL:RUN:LOG? 'MyMeasurement'

The format of log returned is in the following way:

#XY Message Log

where X denotes the length of digits for Y, Y denotes the number of characters in the log then followed by the log message.
For example:

#10 means, there is 1 digit after 1 and there is 0 character in the log.

#2492 means, there are 3 digits after 2 and there are 492 characters in the log message.

:PLUGin:JTOLerance:RUN:MESSage?

Syntax

:PLUGin:JTOLerance:RUN:MESSage? 'identifier'

Input Parameters

'identifier': Specify the measurement name.

Description

This command returns a string describing the state of an jitter tolerance measurement addressed by the measurement name identifier. Possible states include NotStarted, Running, Suspended, Finished, Error, or Stopped.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example

The following example returns the state of a jitter tolerance measurement with the measurement name identifier called 'MyMeasurement':

:PLUG:JTOL:RUN:MESS? 'MyMeasurement' -> Running

:PLUGin:JTOLerance:RUN:PROGress?

Syntax
:PLUGin:JTOLerance:RUN:PROGress? 'identifier'

Input Parameters
'identifier': Specify measurement name.

Return Range
0.0 to 1.0

Description
This command returns a number in the range of 0.0 to 1.0 to indicate the progress of a jitter tolerance measurement addressed by the measurement name identifier. A 0.0 indicates that the measurement has not started and 1.0 indicates the measurement is finished.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example
The following example returns the progress of a jitter tolerance measurement with the measurement name identifier called 'MyMeasurement':

```
:PLUG:JTOL:RUN:PROG? 'MyMeasurement'
0.51
```

:PLUGin:JTOLerance:RUN[:STATus]?

Syntax
:PLUGin:JTOLerance:RUN[:STATus]? 'identifier'

Input Parameters
'identifier': Specify measurement name.

Return Range
0|1

Description
This command returns the running status of a jitter tolerance measurement addressed by the measurement name identifier. A 0 indicates the measurement is not running and a 1 indicates the measurement is running.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example
The following example returns the running status of a jitter tolerance measurement with the measurement name identifier called 'MyMeasurement':

```
:PLUG:JTOL:RUN? 'MyMeasurement'
1
```

:PLUGin:JTOLerance:SHOW:COMment[?]

Syntax

:PLUGin:JTOLerance:SHOW:COMment 'identifier', <“Comment”>

:PLUGin:JTOLerance:SHOW:COMment? 'identifier'

Input Parameters

'identifier': Specify measurement name. For example; 'MyMeasurement'

<“Comment”>: Enter the desired comment.

Description

This command is used to enter a comment for each measurement. This comment will be displayed on the **Measurement History** tab.

This query returns the current settings.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example

:PLUG:JTOL:SHOW:COM 'MyMeasurement', "Jitter Tolerance Example"

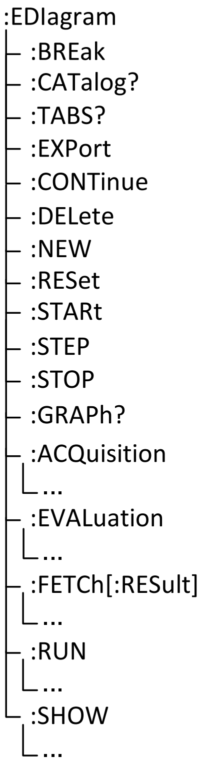
:PLUG:JTOL:SHOW:COM? 'MyMeasurement'

Jitter Tolerance Example

:PLUGin:EDIagram Subnode

The M8070B system software provides quick design analysis with the eye diagram capability. The measurement allows a quick check for the DUT's signal output and determines the signal quality. The eye contour lines display the measured eye at a deeper BER level, for accurate results. The eye diagram generates a three dimensional graph of the bit error rate (BER). This measurement helps in determining and analyzing the quality of the DUT's signal output. The eye diagram results comprise of voltage(y), time(x) and BER(z). The measurement comes with the set of Acquisition and Evaluation Parameters which are used to configure the measurement run. The Acquisition parameters can only be configured before the measurement starts (i.e. during the measurement run you cannot change the value of Acquisition parameter); however the Evaluation parameters can be configured any time even after the measurement has been run.

This subnode has the following SCPI structure:



```
:EDIagram
├─ :BREak
├─ :CATalog?
├─ :TABS?
├─ :EXPort
├─ :CONTinue
├─ :DELete
├─ :NEW
├─ :RESet
├─ :STARt
├─ :STEP
├─ :STOP
├─ :GRAPh?
├─ :ACQuisition
│  └─ ...
├─ :EVALuation
│  └─ ...
├─ :FETCh[:RESult]
│  └─ ...
├─ :RUN
│  └─ ...
└─ :SHOW
   └─ ...
```

This subnode has the following commands and subnodes:

Table 41

Name	Description under
:BREak	:PLUGin:EDIagram:BREak on page 320
:CATalog?	:PLUGin:EDIagram:CATalog? on page 320
:TABS?	:PLUGin:EDIagram:TABS? on page 320
:EXPort	:PLUGin:EDIagram:EXPort on page 321
:CONTinue	:PLUGin:EDIagram:CONTinue on page 321
:DELete	:PLUGin:EDIagram:DELete on page 322
:NEW	:PLUGin:EDIagram:NEW on page 322
:RESet	:PLUGin:EDIagram:RESet on page 323
:STARt	:PLUGin:EDIagram:STARt on page 323
:STEP	:PLUGin:EDIagram:STEP on page 323
:STOP[	:PLUGin:EDIagram:STOP on page 324
GRAPh?	:PLUGin:EDIagram:GRAPh? on page 324
Subnodes	
:ACQuisition:	:PLUGin:EDIagram:ACQuisition Subnode on page 325
:EVALuation	:PLUGin:EDIagram:EVALuation Subnode on page 328
:FETCh[:RESult]	:PLUGin:EDIagram:FETCh[:RESult] Subnode on page 330
:RUN	:PLUGin:EDIagram:RUN Subnode on page 336
:SHOW	:PLUGin:EDIagram:SHOW Subnode on page 339

:PLUGin:EDIagram:BREak

Syntax :PLUGin:EDIagram:BREak 'identifier'

Input Parameters 'identifier': Specify the measurement name to break/pause.

Description This command breaks/pauses a measurement addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example The following example breaks an eye diagram measurement with the

measurement name identifier called 'MyMeasurement':

```
:PLUG:EDI:BRE 'MyMeasurement'
```

:PLUGin:EDIagram:CATalog?

Syntax :PLUGin:EDIagram:CATalog?

Description This command returns a list of all created eye diagram measurement names currently available for measuring.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example Assume three eye diagram measurements named as EDIagram_1, EDIagram_2 and EDIagram_3 are created using NEW command:

```
:PLUG:EDI:NEW 'EDIagram_1'
```

```
:PLUG:EDI:NEW 'EDIagram_2'
```

```
:PLUG:EDI:NEW 'EDIagram_3'
```

The command and returned list would look like the following:

```
:PLUG:EDI:CAT?
```

```
"EDIagram_1","EDIagram_2","EDIagram_3"
```

:PLUGin:EDIagram:TABS?

Syntax :PLUGin:EDIagram:TABS?

Description This query returns a list of all results tabs which can be exported as a CSV file.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUG:EDI:TABS?

Output "Calculated Results"

:PLUGin:EDIagram:EXPort

Syntax :PLUGin:EDIagram:EXPort 'Identifier', 'TabName', 'FilePath'

Description This command exports the measurement data of the selected results tab to a csv file.

Input Parameters

- 'Identifier': Specify the measurement name; for example ''Eye Diagram 1''.
- 'TabName' – Specify the results tab name. To fetch the results tabs for a measurement, refer to the query ":PLUGin:EDIagram:TABS? on page 320". However, if you want all tabs to be saved as a csv file, use " " (single/double quotes).
- 'FilePath' – Specify the location where the CSV file should be saved.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Examples Example 1: The following command will save the 'Calculated Results' tab of eye diagram measurement to *Results.csv*:

:PLUGin:EDI:EXPort 'Eye Diagram 1', 'Results', 'C:\Results.csv'

Example 2: The following command will save all results tab of eye diagram measurement to *Results.csv*.

:PLUGin:EDI:EXPort 'Eye Diagram 1', '', 'C:\Results.csv'

:PLUGin:EDIagram:CONTinue

Syntax :PLUGin:EDIagram:CONTinue 'identifier'

Input Parameters 'identifier': Specify the measurement name to continue.

Description This command allows you to continue an eye diagram measurement, addressed by the measurement name identifier, which was stopped using the BREak command.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example The following example continues a halted eye diagram measurement addressed with the measurement name identifier called 'MyMeasurement':

:PLUG:EDI:CONT 'MyMeasurement'

:PLUGin:EDIagram:DELete

Syntax :PLUGin:EDIagram:DELete 'identifier'

Input Parameters 'identifier': Specify the measurement name to delete.

Description This command deletes a previously created eye diagram measurement addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example The following example deletes an eye diagram measurement addressed by the measurement name identifier called 'MyMeasurement':

```
:PLUG:EDI:DEL 'MyMeasurement'
```

:PLUGin:EDIagram:NEW

Syntax :PLUGin:EDIagram:NEW 'identifier'

Input Parameters 'identifier': Specify the measurement name.

Description This command creates a new eye diagram measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example The following example creates an eye diagram measurement name identifier called 'MyMeasurement':

```
:PLUG:EDI:NEW 'MyMeasurement'
```

NOTE Creating multiple plugins using this command may slow down the GUI operations which may also result delay in remote programming. To prevent the plugin from opening automatically in the GUI, it is recommended to use “0” as a parameter input in this command.

Example- :PLUG:EDI:NEW 'MyMeasurement',0

:PLUGin:EDIagram:RESet

Syntax :PLUGin:EDIagram:RESet 'identifier'

Input Parameters 'identifier': Specify the measurement name to reset.

Description This command resets an eye diagram measurement addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example The following example resets an eye diagram measurement addressed with the measurement name identifier called 'MyMeasurement':

```
:PLUG:EDI:RES 'MyMeasurement'
```

:PLUGin:EDIagram:STARt

Syntax :PLUGin:EDIagram:STARt 'identifier'

Input Parameters 'identifier': Specify the measurement name to start.

Description This command starts an eye diagram measurement addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example The following example starts an eye diagram measurement addressed with the measurement name identifier called 'MyMeasurement':

```
:PLUG:EDI:STAR 'MyMeasurement'
```

:PLUGin:EDIagram:STEP

Syntax :PLUGin:EDIagram:STEP 'identifier'

Input Parameters 'identifier': Specify the measurement name to step.

Description This command allows you to step through an eye diagram measurement, addressed by the measurement name identifier, that was stopped using the BREak command.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example The following example triggers a single eye diagram measurement step of the measurement addressed with the measurement name identifier called

'MyMeasurement':

:PLUG:EDI:STEP 'MyMeasurement'

:PLUGin:EDIagram:STOP

Syntax
:PLUGin:EDIagram:STOP 'identifier'

Input Parameters
'identifier': Specify the measurement name to stop.

Description
This command stops an eye diagram measurement addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example
The following example stops an eye diagram measurement with the measurement name identifier called 'MyMeasurement':

:PLUG:EDI:STOP 'MyMeasurement'

:PLUGin:EDIagram:GRAPh?

Syntax
:PLUGin:EDIagram:GRAPh? 'identifier' [,DesiredWidth][,DesiredHeight][,<CURR | WHIT>][,<PNG>][,1 | 0][,'TabName']

Input Parameters
'identifier': Specify measurement name.

Other optional parameters are:

[,DesiredWidth]: Specify the desired width of the image.

[,DesiredHeight]: Specify the desired height of the image.

[,<CURR | WHIT>]: Specify whether the user wants to capture in current theme or wants to capture in white background.

[,<PNG>]: Specify the format of the image. The default format is PNG.

[,1 | 0]: Specify whether to capture the graph with legends or not.

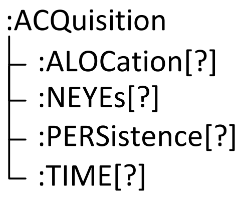
[,'TabName']: Specify the tab name on which the graph is supposed to be captured.

Description
This query returns data of the image captured from the graph view of the plugin in the specified format.

Example
:PLUG:EDIagram:GRAP? 'MyMeasurement',1000,800,CURR,PNG,1,'Graph'

:PLUGin:EDIagram:ACQuisition Subnode

This subnode has the following SCPI structure:



This subnode has the following commands:

Table 42

Name	Description under
:ALOCation[?]	:PLUGin:EDIagram:ACQuisition:ALOCation[?] on page 326
:NEYEs[?]	:PLUGin:EDIagram:ACQuisition:NEYEs[?] on page 326
:PERSistence[?]	:PLUGin:EDIagram:ACQuisition:PERSistence [?] on page 327
:TIME[?]	:PLUGin:EDIagram:ACQuisition:TIME[?] on page 327

:PLUGin:EDIagram:ACQuisition:ALOCation[?]

Syntax

:PLUGin:EDIagram:ACQuisition:ALOCation 'Identifier', < location-string>

:PLUGin:EDIagram:ACQuisition:ALOCation? 'Identifier'

Input Parameters

Identifier: 'ParameterSweep1'

<location-string>: 'M*.DataIn*'

Description

This command sets the location/location group string against which the data acquisition is performed.

This query returns the current location/location group string against which measurement is configured for the data acquisition.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example

:PLUG:EDI:ACQ:ALOC 'MyMeasurement','M1.DataIn1'

:PLUGin:EDIagram:ACQuisition:NEYEs[?]

Syntax

:PLUGin:EDIagram:ACQuisition:NEYEs 'Identifier', TDZ | ODF

:PLUGin:EDIagram:ACQuisition:NEYEs? 'Identifier'

Input Parameters

Identifier: 'MyMeasurement'

Range

TDZ | ODF

Description

This command sets the number eyes to be displayed in UI. It can be either 1.5 or 2.0.

The query returns current value.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example

:PLUG:EDI:ACQ:NEYEs 'MyMeasurement', TDZ

:PLUGin:EDIagram:ACQuisition:PERSistence[?]

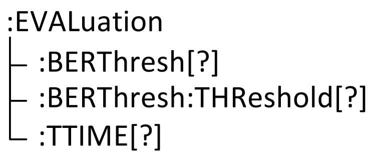
Syntax	:PLUGin:EDIagram:ACQuisition:PERSistence 'Identifier', <INF | FTIM> :PLUGin:EDIagram:ACQuisition:PERSistence? 'Identifier'
Input Parameters	Identifier: 'MyMeasurement'
Return Range	INF | FTIM
Description	This command sets the criteria for persistence. Persistence could be infinite or fixed time. The query returns the current setting. This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.
Example	:PLUG:EDI:ACQ:PERS 'MyMeasurement',INF

:PLUGin:EDIagram:ACQuisition:TIME[?]

Syntax	:PLUGin:EDIagram:ACQuisition:TIME 'Identifier', <NRf> :PLUGin:EDIagram:ACQuisition:TIME? 'Identifier'
Input Parameters	Identifier: 'MyMeasurement'
Return Range	50s to 216ks
Description	This command allows the user to specify the persistence time. The query returns current setting. This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.
Example	:PLUG:EDI:ACQ:TIME 'MyMeasurement', 80

:PLUGin:EDIagram:EVALuation Subnode

This subnode has the following SCPI structure:



```
:EVALuation
├ :BERThresh[?]
├ :BERThresh:THReshold[?]
└ :TTIME[?]
```

This subnode has the following commands:

Table 43

Name	Description under
:BERThresh[?]	:PLUGin:EDIagram:EVALuation:BERThresh[?] on page 328
:BERThresh:THReshold[?]	:PLUGin:EDIagram:EVALuation:BERThreshold[?] on page 329
:TTIMe[?]	:PLUGin:EDIagram:EVALuation:TTIMe[?] on page 329

:PLUGin:EDIagram:EVALuation:BERThresh[?]

Syntax :PLUGin:EDIagram:EVALuation:BERThresh 'Identifier', <ZERR | BTHR>

:PLUGin:EDIagram:EVALuation:BERThresh? 'Identifier'

Input Parameters Identifier: 'MyMeasurement'

Return Range ZERR | BTHR

Description This command tells how to calculate measurement results, such as, eye height, eye width and JPPeak either using '0 errors' (ZERO) or a specific 'BER Threshold' (THReshold). In the latter case, numeric value of the threshold is specified with the

:PLUGin:EDIagram:EVALuation:BERThreshold[?] on page 329 command.

The query returns either 0 errors (ZERR) or BER threshold (BTHR).

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUG:EDI:EVAL:BERT 'MyMeasurement', BTHR

:PLUGin:EDIagram:EVALuation:BERThreshold[?]

Syntax :PLUGin:EDIagram:EVALuation:BERThreshold 'Identifier', <NRf>

:PLUGin:EDIagram:EVALuation:BERThreshold? 'Identifier'

Input Parameters Identifier: 'MyMeasurement'

Range 1E-12 to 1E-1

Description This command allows the user to specify BER Threshold value.

The query returns the current BER Threshold value.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUG:EDI:EVAL:BERT 'MyMeasurement', 1E-10

:PLUGin:EDIagram:EVALuation:TTIMe[?]

Syntax :PLUGin:EDIagram:EVALuation:TTIMe 'Identifier', <TEIG | TNIN>

:PLUGin:EDIagram:EVALuation:TTIMe? 'Identifier'

Input Parameters Identifier: 'MyMeasurement'

Range TEIG | TNIN

Description This command allows the user to specify the transition time. It can be either 10/90 or 20/80.

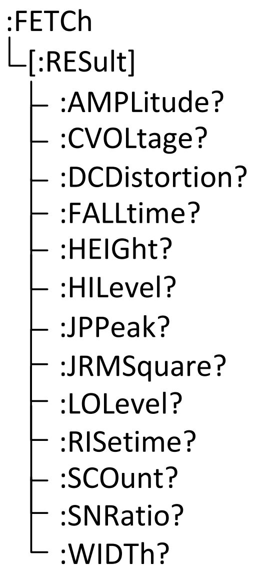
The query returns the current setting.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUG:EDI:EVAL:TTIM 'MyMeasurement', TNIN

:PLUGin:EDIagram:FETCh[:RESult] Subnode

This subnode has the following SCPI structure:



```
:FETCh
└[:RESult]
  ├ :AMPLitude?
  ├ :CVOLtage?
  ├ :DCDistortion?
  ├ :FALLtime?
  ├ :HEIGht?
  ├ :HILevel?
  ├ :JPPeak?
  ├ :JRMSquare?
  ├ :LOLevel?
  ├ :RISetime?
  ├ :SCOunt?
  ├ :SNRatio?
  └ :WIDTh?
```

This subnode has the following commands and subnodes:

Table 44

Name	Description under
AMPLitude?	:PLUGin:EDIagram:FETCh[:RESult]:AMPLitude? on page 331
:CVOLtage?	:PLUGin:EDIagram:FETCh[:RESult]:CVOLtage? on page 332
:DCDistortion?	:PLUGin:EDIagram:FETCh[:RESult]:DCDistortion? on page 332
:FALLtime?	:PLUGin:EDIagram:FETCh[:RESult]:FALLtime? on page 332
:HEIGht?	:PLUGin:EDIagram:FETCh[:RESult]:HEIGht? on page 333

Name	Description under
:HILevel?	:PLUGin:EDIagram:FETCh[:RESult]:HILevel? on page 333
:JPPeak?	:PLUGin:EDIagram:FETCh[:RESult]:JPPeak? on page 333
:JRMSquare?	:PLUGin:EDIagram:FETCh[:RESult]:JRMSquare? on page 334
:LOLevel?	:PLUGin:EDIagram:FETCh[:RESult]:LOLevel? on page 334
:RISetime?	:PLUGin:EDIagram:FETCh[:RESult]:RISetime? on page 334
:SCOunt?	:PLUGin:EDIagram:FETCh[:RESult]:SCOunt? on page 335
:SNRatio?	:PLUGin:EDIagram:FETCh[:RESult]:SNRatio? on page 335
:WIDTh?	:PLUGin:EDIagram:FETCh[:RESult]:WIDTh? on page 335

:PLUGin:EDIagram:FETCh[:RESult]:AMPLitude?

Syntax :PLUGin:EDIagram:FETCh[:RESult]:AMPLitude? 'identifier'

Input Parameters 'identifier': Specify the measurement name.

Description This command returns the eye amplitude which is the difference between the mean logic 1 level values and the mean logic 0 level values in a histogram of an eye diagram.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUG:EDI:FETC:AMPL? 'MyMeasurement'

:PLUGin:EDIagram:FETCh[:RESult]:CVOLtage?

Syntax :PLUGin:EDIagram:FETCh[:RESult]:CVOLtage? 'identifier'

Input Parameters 'identifier': Specify the measurement name.

Description This command returns the cross voltage which is the measurement of the crossing point level in relation to the logic 1 level and logic 0 level.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUG:EDI:FETC:CVOL? 'MyMeasurement'

:PLUGin:EDIagram:FETCh[:RESult]:DCDistortion?

Syntax :PLUGin:EDIagram:FETCh[:RESult]:DCDistortion? 'identifier'

Input Parameters 'identifier': Specify the measurement name.

Description This command returns the value of DC Distortion calculated after a run. This value is the difference between the period of a 1 bit and a 0 bit.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUG:EDI:FETC:DCD? 'MyMeasurement'

:PLUGin:EDIagram:FETCh[:RESult]:FALLtime?

Syntax :PLUGin:EDIagram:FETCh[:RESult]:FALLtime? 'identifier'

Input Parameters 'identifier': Specify the measurement name.

Description This command returns the value of the fall time. Fall time is a measurement of the mean transition time of the data on the downward slope of an eye diagram. This transition time is either 2080 or 1090. See :PLUGin:EDIagram:EVALuation:TTIMe[?] on page 329.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUG:EDI:FETC:FALL? 'MyMeasurement'

:PLUGin:EDIagram:FETCh[:RESult]:HEIGht?

Syntax :PLUGin:EDIagram:FETCh[:RESult]:HEIGht? 'identifier'

Input Parameters 'identifier': Specify the measurement name.

Description This command returns the value of the eye height which is a measurement of the vertical opening of an eye diagram. This opening is affected by the BER threshold that is set by the BER threshold command. See :PLUGin:EDIagram:EVALuation:BERThresh[?] on page 328

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUG:EDI:FETC:HEIG? 'MyMeasurement'

:PLUGin:EDIagram:FETCh[:RESult]:HILevel?

Syntax :PLUGin:EDIagram:FETCh[:RESult]:HILevel? 'identifier'

Input Parameters 'identifier': Specify the measurement name.

Description This command returns the measurement of the mean value of the logical 1 in the eye diagram. This is directly affected by the values set for Eye Boundaries. One Level is a measure of the mean value of the logical 1 of an eye diagram.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUG:EDI:FETC:HIL? 'MyMeasurement'

:PLUGin:EDIagram:FETCh[:RESult]:JPPeak?

Syntax :PLUGin:EDIagram:FETCh[:RESult]:JPPeak? 'identifier'

Input Parameters 'identifier': Specify the measurement name.

Description This command returns Jitter P-P value. It is the full width of the eye diagram at the eye crossing point. This is affected by the BER threshold that is set by the BER threshold command. See :PLUGin:EDIagram:EVALuation:BERThreshold[?] on page 329

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUG:EDI:FETC:JPP? 'MyMeasurement'

:PLUGin:EDIagram:FETCh[:RESult]:JRMSquare?

Syntax :PLUGin:EDIagram:FETCh[:RESult]:JRMSquare? 'identifier'

Input Parameters 'identifier': Specify the measurement name.

Description This command returns the Jitter Root-Mean Square value. It is the standard deviation of the normal distribution of random jitter. It is dependent on the BER Threshold.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUG:EDI:FETC:JRMS? 'MyMeasurement'

:PLUGin:EDIagram:FETCh[:RESult]:LOLevel?

Syntax :PLUGin:EDIagram:FETCh[:RESult]:LOLevel? 'identifier'

Input Parameters 'identifier': Specify the measurement name.

Description This command returns the measurement of the mean value of the logical 1 in the eye diagram. This is directly affected by the values set for eye boundaries.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUG:EDI:FETC:LOL? 'MyMeasurement'

:PLUGin:EDIagram:FETCh[:RESult]:RISetime?

Syntax :PLUGin:EDIagram:FETCh[:RESult]:RISetime? 'identifier'

Input Parameters 'identifier': Specify the measurement name.

Description This command returns rise time which is a measurement of the mean transition time of the data on the upward slope of an eye diagram. This transition time is either 20/80 or 10/90. See :PLUGin:EDIagram:EVALuation:TTIMe[?] on page 329

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUG:EDI:FETC:RIS? 'MyMeasurement'

:PLUGin:EDIagram:FETCh[:RESult]:SCOunt?

Syntax :PLUGin:EDIagram:FETCh[:RESult]:SCOunt? 'identifier'

Input Parameters 'identifier': Specify the measurement name.

Description This command returns the sample count.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUG:EDI:FETC:SCO? 'MyMeasurement'

:PLUGin:EDIagram:FETCh[:RESult]:SNRatio?

Syntax :PLUGin:EDIagram:FETCh[:RESult]:SNRatio? 'identifier'

Input Parameters 'identifier': Specify the measurement name.

Description This command returns signal-to-noise ratio which is a measurement of the signal difference between 1 level and 0 level in relation to the rms value of 1-level noise + rms value of 0-level noise.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUG:EDI:FETC:SNR? 'MyMeasurement'

:PLUGin:EDIagram:FETCh[:RESult]:WIDTh?

Syntax :PLUGin:EDIagram:FETCh[:RESult]:WIDTh? 'identifier'

Input Parameters 'identifier': Specify the measurement name.

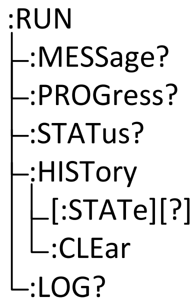
Description This command returns the calculated eye width which is horizontal measurement of the eye opening at a specified BER Threshold.It can be either at the eye crossing point (CROssing) or custom defined (CUStom).

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUG:EDI:FETC:WIDT? 'MyMeasurement'

:PLUGin:EDIagram:RUN Subnode

This subnode has the following SCPI structure:



```
:RUN
├:MESSage?
├:PROGress?
├:STATus?
├:HISTory
│ ├[:STATe][?]
│ └:CLEar
└:LOG?
```

This subnode has the following commands:

Table 45

Name	Description under
:MESSage?	:PLUGin:EDIagram:RUN:MESSage? on page 337
:PROGress?	:PLUGin:EDIagram:RUN:PROGress? on page 337
:STATus?	:PLUGin:EDIagram:RUN:STATus? on page 337
:HISTory:[:STATe]	:PLUGin:EDIagram:RUN:HISTory[:STATe][?] on page 338
:HISTory:CLEar	:PLUGin:EDIagram:RUN:HISTory:CLEar on page 338
:LOG?	:PLUGin:EDIagram:RUN:LOG? on page 338

:PLUGin:EDIagram:RUN:MESSage?

Syntax :PLUGin:EDIagram:RUN:MESSage? 'Identifier'

Input Parameters Identifier: 'MyMeasurement'

Description This command returns a string describing the state of an eye diagram measurement addressed by the measurement name identifier. Possible states include Not Started, Running, Suspended, Finished, Error, or Stopped.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUG:EDI:RUN:MESS? 'MyMeasurement'

:PLUGin:EDIagram:RUN:PROGress?

Syntax :PLUGin:EDIagram:RUN:PROGress? 'Identifier'

Input Parameters Identifier: 'MyMeasurement'

Return Range 0 to 1

Description This command returns a number in the range of 0.0 to 1.0 to indicate the progress of an eye diagram measurement addressed by the measurement name identifier. 0.0 indicates that the measurement has not started and 1.0 indicates the measurement is finished.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUG:EDI:RUN:PROG? 'MyMeasurement'

:PLUGin:EDIagram:RUN:STATus?

Syntax :PLUGin:EDIagram:RUN:STATus? 'Identifier'

Input Parameters Identifier: 'MyMeasurement'

Return Range 0|1

Description This command returns the running status of an eye diagram measurement addressed by the measurement name identifier. 0 indicates the measurement is not running and a 1 indicates the measurement is running.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUG:EDI:RUN:STAT? 'MyMeasurement'

:PLUGin:EDIagram:RUN:HISTory[:STATe][?]

Syntax :PLUGin:EDIagram:RUN:HISTory[:STATe] 'Identifier', <0|1|ON|OFF>

:PLUGin:EDIagram:RUN:HISTory[:STATe]? 'Identifier'

Input Parameters Identifier: 'MyMeasurement'

Return Range 0|1

Description This command enables/disables the storage of eye diagram measurement results addressed by the measurement name identifier.

This query returns the current setting.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUG:EDI:RUN:HIST 'MyMeasurement', 1

:PLUGin:EDIagram:RUN:HISTory:CLEar

Syntax :PLUGin:EDIagram:RUN:HISTory:CLEar 'Identifier'

Input Parameters Identifier: 'MyMeasurement'

Description This command deletes the eye diagram measurement history addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUG:EDI:RUN:HIST:CLE 'MyMeasurement'

:PLUGin:EDIagram:RUN:LOG?

Syntax :PLUGin:EDIagram:RUN:LOG? 'Identifier'

Input Parameters Identifier: 'MyMeasurement'

Description This command returns logs for the addressed measurement.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUG:EDI:RUN:LOG? 'MyMeasurement'

The format of log returned is in the following way:

#XY Message Log

where X denotes the length of digits for Y, Y denotes the number of characters in the log then followed by the log message.

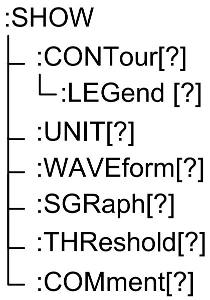
For example:

#10 means, there is 1 digit after 1 and there is 0 character in the log.

#2492 means, there are 3 digits after 2 and there are 492 characters in the log message.

:PLUGin:EDIagram:SHOW Subnode

This subnode has the following SCPI structure:



This subnode has the following commands:

Table 46

Name	Description under
:CONTour	:PLUGin:EDIagram:SHOW:CONTour[?] on page 340
:LEGend	:PLUGin:EDIagram:SHOW:CONTour:LEGend[?] on page 340
:UNIT	:PLUGin:EDIagram:SHOW:UNIT[?] on page 341

Name	Description under
:WAVEform	:PLUGin:EDIagram:SHOW:WAVeform[?] on page 341
:SGRaph	:PLUGin:EDIagram:SHOW:WAVeform:SGRaph[?] on page 342
:THReshold	:PLUGin:EDIagram:SHOW:WAVeform:THReshold[?] on page 342
COMment	:PLUGin:EDIagram:SHOW:COMment[?] on page 343

:PLUGin:EDIagram:SHOW:CONTour[?]

Syntax :PLUGin:EDIagram:SHOW:CONTour 'Identifier', <0|1|ON|OFF>

:PLUGin:EDIagram:SHOW:CONTour? 'Identifier'

Input Parameters Identifier: 'MyMeasurement'

Return Range 0|1

Description This command shows/hides the contour lines on the graph.

This query returns the current setting.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUG:EDI:SHOW:CONT 'MyMeasurement', 1

:PLUGin:EDIagram:SHOW:CONTour:LEGend[?]

Syntax :PLUGin:EDIagram:SHOW:CONTour:LEGend 'Identifier', <0|1|ON|OFF>

:PLUGin:EDIagram:SHOW:CONTour:LEGend? 'Identifier'

Input Parameters Identifier: 'MyMeasurement'

Return Range 0|1

Description The command shows/hides contour legends.

This query returns the current setting.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUG:EDI:SHOW:CONT:LEG 'MyMeasurement', 1

:PLUGin:EDIagram:SHOW:UNIT[?]

Syntax :PLUGin:EDIagram:SHOW:UNIT 'Identifier', <UNIT | TIME >

:PLUGin:EDIagram:SHOW:UNIT? 'Identifier',

Input Parameters Identifier: 'MyMeasurement'

Return Range UNIT | TIME

Description The command sets the display unit. It can be either in terms of seconds or UNIT.

This query returns the current setting.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUG:EDI:SHOW:UNIT 'MyMeasurement', UNIT

:PLUGin:EDIagram:SHOW:WAVeform[?]

Syntax :PLUGin:EDIagram:SHOW:WAVeform 'Identifier', <0|1|OFF|ON>

:PLUGin:EDIagram:SHOW:WAVeform? 'Identifier'

Input Parameters Identifier: 'MyMeasurement'

<0|1|OFF|ON>: Show/hide waveform

Return Range 0|1

Description This command shows/hides the waveform. When this feature is enabled, the waveform is displayed together with the contour lines. Disabling this feature will only display contour lines, if available.

This query returns the current setting.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUG:EDI:SHOW:WAV 'MyMeasurement', 1

:PLUGin:EDIagram:SHOW:WAVeform:SGRaph[?]

Syntax :PLUGin:EDIagram:SHOW:WAVeform:SGRaph 'Identifier', <0|1|OFF|ON>

:PLUGin:EDIagram:SHOW:WAVeform:SGRaph? 'Identifier'

Input Parameters Identifier: 'MyMeasurement'

<0|1|OFF|ON>: Smooth waveform graphics.

Return Range 0|1

Description This command smooths out the waveform graphics. When this feature is enabled, the waveform is displayed together with the contour lines. Disabling this feature will only display contour lines, if available.

This query returns the current setting.

This SCPI is applicable for M8041A, M8051A, M8062A and M8043A.

Example :PLUG:EDI:SHOW:WAV:SGR 'MyMeasurement', 1

:PLUGin:EDIagram:SHOW:WAVeform:THReshold[?]

Syntax :PLUGin:EDIagram:SHOW:WAVeform:THReshold 'Identifier', <NR3>

:PLUGin:EDIagram:SHOW:WAVeform:THReshold? 'Identifier'

Input Parameters Identifier: 'MyMeasurement'

Return Range 1E-6 to 1E-1

Description This command sets the waveform BER threshold value.

This query returns the current setting.

This SCPI is applicable for M8041A, M8051A and M8062A

Example :PLUG:EDI:SHOW:WAV:THR 'MyMeasurement', 1E-3

:PLUGin:EDIagram:SHOW:COMment[?]

Syntax

:PLUGin:EDIagram:SHOW:COMment 'Identifier', <“Commnet”>

:PLUGin:EDIagram:SHOW:COMment? 'Identifier',

Input Parameters

Identifier: Specify the plugin name. For e.g. 'MyMeasurement'

<“Commnet”> - Enter the desired comment.

Description

This command is used to enter a comment for each measurement. This comment will be displayed on the **Measurement History** tab.

This query returns the current setting.

This SCPI is applicable for M8041A, M8051A, M8062A, M8043A and M8046A.

Example

:PLUG:EDI:SHOW:COM 'MyMeasurement', "Eye Measurement Example"

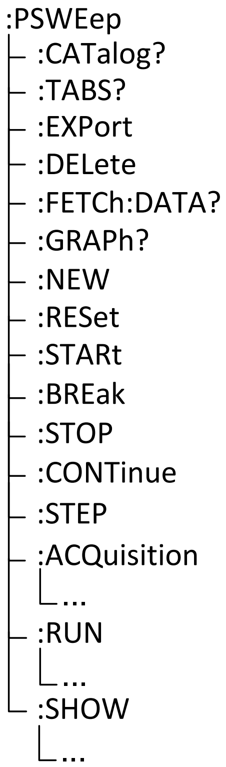
:PLUG:EDI:SHOW:COM? 'MyMeasurement'

Eye Measurement Example

:PLUGin:PSWEep Subnode

The Parameter Sweep plugin allows you to sweep the parameters of the selected location/location group to the desired analyzer in order to calculate BER. This is helpful in scenarios when a user wants to pick a property to sweep within a specified range with specified steps and measure BER with respect to that sweep. Using this plugin, a user can sweep the parameters using the acquisition parameters provided in the Parameter window and then run the measurement to calculate BER.

This subnode has the following SCPI structure:



```
:PSWEep
├─ :CATalog?
├─ :TABS?
├─ :EXPort
├─ :DELete
├─ :FETCh:DATA?
├─ :GRAPh?
├─ :NEW
├─ :RESet
├─ :STARt
├─ :BREak
├─ :STOP
├─ :CONTinue
├─ :STEP
├─ :ACQuisition
│   └─...
├─ :RUN
│   └─...
└─ :SHOW
    └─...
```

This subnode has the following commands and subnodes:

Table 47

Name	Description under
:CATalog?	:PLUGin:PSWEep:CATalog? on page 346
:TABS?	:PLUGin:PSWEep:TABS? on page 346
:EXPort	:PLUGin:PSWEep:EXPort on page 347
:DELete	:PLUGin:PSWEep:DELete on page 347
FETCh:DATA?	:PLUGin:PSWEep:FETCh:DATA? on page 348
GRAPh?	:PLUGin:PSWEep:GRAPh? on page 348
:NEW	:PLUGin:PSWEep:NEW on page 349
:RESet	:PLUGin:PSWEep:RESet on page 350
:STARt	:PLUGin:PSWEep:STARt on page 350
:BREak	:PLUGin:PSWEep:BREak on page 350
:STOP	:PLUGin:PSWEep:STOP on page 351
:CONTinue	:PLUGin:PSWEep:CONTinue on page 351
:STEP	:PLUGin:PSWEep:STEP on page 352
Subnodes	
:ACQuisition:	:PLUGin:PSWEep:ACQuisition Subnode on page 353
:RUN	:PLUGin:PSWEep:RUN Subnode on page 363
:SHOW	:PLUGin:PSWEep:SHOW Subnode on page 367

:PLUGin:PSWEep:CATalog?

Syntax :PLUGin:PSWEep:CATalog?

Description This command returns a list of all created parameter sweep names currently available for measuring.

This SCPI is applicable for M8041A, M8051A, M8062A, M8045A and M8046A.

Example Assume three parameter sweep measurements named as: 'ParameterSweep1', 'ParameterSweep2' and 'ParameterSweep3' are created using NEW command:

:PLUG:PSWE:NEW 'ParameterSweep1'

:PLUG:PSWE:NEW 'ParameterSweep2'

:PLUG:PSWE:NEW 'ParameterSweep3'

The command and returned list would look like the following:

:PLUG:PSWE:CAT?

"ParameterSweep1","ParameterSweep2","ParameterSweep3"

:PLUGin:PSWEep:TABS?

Syntax :PLUGin:PSWEep:TABS?

Description This query returns a list of all results tabs which can be exported as a CSV file.

This SCPI is applicable for M8041A, M8051A, M8062A, M8045A, M8043A and M8046A.

Example :PLUG:EDI:TABS?

Output "M1.DataIn1"

:PLUGin:PSWEep:EXPort

Syntax

```
:PLUGin:PSWEep:EXPort 'Identifier', 'TabName', 'FilePath'
```

Description

This command exports the measurement data of the selected results tab to a csv file.

Input Parameters

- 'Identifier': Specify the measurement name; for example ''Parameter Sweep 1''.
- 'TabName' – Specify the results tab name. To fetch the results tabs for a measurement, refer to the query “:PLUGin:PSWEep:TABS? on page 346”. However, if you want all tabs to be saved as a csv file, use " " (single/double quotes).
- 'FilePath' – Specify the location where the CSV file should be saved.

This SCPI is applicable for M8041A, M8051A, M8062A, M8045A, M8043A and M8046A.

Examples

Example 1: The following command will save the 'M1.DataIn1' tab of parameter sweep measurement to *Results.csv*:

```
:PLUGin:PSWE:EXPort 'Parameter Sweep 1', 'M1.DataIn1', 'C:\Results.csv'
```

Example 2: The following command will save all results tab of parameter sweep measurement to *Results.csv*.

```
:PLUGin:PSWE:EXPort 'Parameter Sweep 1', '', 'C:\Results.csv'
```

:PLUGin:PSWEep:DELete

Syntax

```
:PLUGin:PSWEep:DELete 'identifier'
```

Input Parameters

'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'

Description

This command deletes a previously created parameter sweep measurement addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A, M8045A, M8043A and M8046A.

Example

The following example deletes a parameter sweep measurement addressed by the measurement name identifier called 'Parameter Sweep 1':

```
:PLUG:PSWE:DEL 'Parameter Sweep 1'
```

:PLUGin:PSWEep:FETCh:DATA?

Syntax

```
:PLUGin:PSWEep:FETCh:DATA? 'identifier'[,'Location']
```

Input Parameters

'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'

['Location']: Specify location (optional) in case of location group

Description

This command returns the raw data of the parameter sweep measurement

associated with a specific measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A, M8045A, M8043A and M8046A.

Example

```
:PLUG:PSWE:FETC:DATA? 'ParameterSweep1', 'M1.DataIn1'
```

A measurement run on a location identifier (for example, 'M1.DataIn1') will return following result:

```
("M1.DataIn1",1491575569.21202,2500015,2499985,1250000,1249985,-
6.7E-09,1491575569.22802,2499982,2500018,1249982,1250000, ...)
```

The first item in the comma separated list is the name of the location for which the data is being queried. Rest of the values in the comma separated list for each measurement point are Sweep Value, Error Ratio, Compared Bits, Errored Bits, Errored Zero Ratio, Compared Zeros, Errored Zeros, Error One Ratio, Compared Ones and Errored Ones are repeated in the same order for subsequent measured points.

If the location is omitted from the above example e.g. if the query is :PLUG:PSWE:FETC:DATA? 'ParameterSweep1' then the query returns the comma separated list of all the analyzer location group participating in the measurement.

:PLUGin:PSWEep:GRAPh?

Syntax

```
:PLUGin:PSWEep:GRAPh? 'identifier'
[,DesiredWidth][,DesiredHeight][,<CURR | WHIT>][,<PNG>][,1 |
0][,'TabName']
```

Input Parameters

'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'

Other optional parameters are:

[,DesiredWidth]: Specify the desired width of the image.

[,DesiredHeight]: Specify the desired height of the image.

[,<CURR | WHIT>]: Specify whether the user wants to capture in current theme or wants to capture in white background.

[,<PNG>]: Specify the format of the image. The default format is PNG.

[,1 | 0]: Specify whether to capture the graph with legends or not.

[,'TabName']: Specify the tab name on which the graph is supposed to be captured.

Description This query returns data of the image captured from the graph view of the plugin in the specified format.

This SCPI is applicable for M8041A, M8051A, M8062A, M8045A, M8043A and M8046A.

Example :PLUG:PSWE:GRAP? 'Parameter Sweep 1',1000,800,CURR,PNG,1,'Graph'

:PLUGin:PSWEep:NEW

Syntax :PLUGin:PSWEep:NEW 'identifier'

Input Parameters 'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'

Description This command creates a new parameter sweep measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A, M8045A, M8043A and M8046A.

Example The following example creates a parameter sweep measurement name identifier called 'ParameterSweep1':

:PLUG:PSWE:NEW 'ParameterSweep1'

NOTE **Creating multiple plugins using this command may slow down the GUI operations which may also result delay in remote programming. To prevent the plugin from opening automatically in the GUI, it is recommended to use “0” as a parameter input in this command.**

Example- :PLUG:PSWE:NEW 'ParameterSweep1',0

:PLUGin:PSWEep:RESet

Syntax :PLUGin:PSWEep:RESet 'identifier'

Input Parameters 'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'

Description This command resets a parameter sweep measurement addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A, M8045A, M8043A and M8046A.

Example The following example resets a parameter sweep measurement addressed with the measurement name identifier called 'ParameterSweep1':

:PLUG:PSWE:RES 'ParameterSweep1'

:PLUGin:PSWEep:STARt

Syntax :PLUGin:PSWEep:STARt 'identifier'

Input Parameters 'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'

Description This command starts a parameter sweep measurement addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A, M8045A, M8043A and M8046A.

Example The following example starts a parameter sweep measurement addressed with the measurement name identifier called 'ParameterSweep1':

:PLUG:PSWE:STAR 'ParameterSweep1'

:PLUGin:PSWEep:BREak

Syntax :PLUGin:PSWEep:BREak 'identifier'

Input Parameters 'Identifier': Specify the measurement name to break/pause. For e.g. 'ParameterSweep1'

Description This command breaks/pauses a measurement addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A, M8045A, M8043A and M8046A.

Example The following example breaks a Parameter Sweep measurement with the measurement name identifier called 'ParameterSweep1':

```
:PLUG:PSWE:BRE 'ParameterSweep1'
```

:PLUGin:PSWEep:STOP

Syntax

```
:PLUGin:PSWEep:STOP  'identifier'
```

Input Parameters 'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'

Description This command stops a parameter sweep measurement addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A, M8045A, M8043A and M8046A.

Example The following example stops a parameter sweep measurement with the measurement name identifier called 'ParameterSweep1':

```
:PLUG:PSWE:STOP 'ParameterSweep1'
```

:PLUGin:PSWEep:CONTinue

Syntax

```
:PLUGin:PSWEep:CONTinue 'identifier'
```

Input Parameters 'Identifier': Specify the measurement name to continue. For e.g. 'ParameterSweep1'

Description This command allows you to continue a Parameter Sweep measurement, addressed by the measurement name identifier, which was stopped using the BREak command.

This SCPI is applicable for M8041A, M8051A, M8062A, M8045A, M8043A and M8046A.

Example The following example continues a parameter sweep measurement with the measurement name identifier called 'ParameterSweep1':

```
:PLUG:PSWE: CONT 'ParameterSweep1'
```

:PLUGin:PSWEep:STEP

Syntax

```
:PLUGin:PSWEep:STEP 'identifier'
```

Input Parameters

'Identifier': Specify the measurement name to step. For e.g. 'ParameterSweep1'

Description

This command allows you to step through a Parameter Sweep measurement, addressed by the measurement name identifier, that was stopped using the BREak command.

This SCPI is applicable for M8041A, M8051A, M8062A, M8045A, M8043A and M8046A.

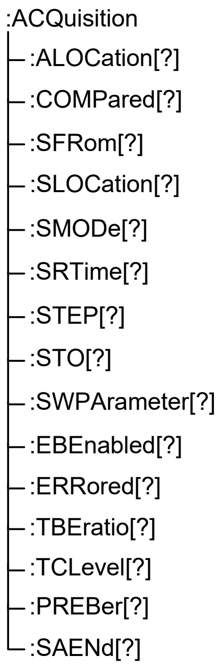
Example

The following example steps a parameter sweep measurement with the measurement name identifier called 'ParameterSweep1':

```
:PLUG:PSWE:STEP 'ParameterSweep1'
```

:PLUGin:PSWEep:ACQuisition Subnode

This subnode has the following SCPI structure:



```
:ACQuisition
├─:ALOCation[?]
├─:COMPared[?]
├─:SFRom[?]
├─:SLOCation[?]
├─:SMODe[?]
├─:SRTime[?]
├─:STEP[?]
├─:STO[?]
├─:SWPArameter[?]
├─:EBEnabled[?]
├─:ERRored[?]
├─:TBEratio[?]
├─:TCLevel[?]
├─:PREBer[?]
└─:SAENd[?]
```

This subnode has the following commands:

Table 48

Name	Description under
:ALOCation[?]	:PLUGin:PSWEep:ACQuisition:ALOCation[?] on page 354
SLOCation[?]	:PLUGin:PSWEep:ACQuisition:SLOCation[?] on page 355
:SWPArameter[?]	:PLUGin:PSWEep:ACQuisition:SWPArameter[?] on page 355

Name	Description under
::SMODe[?]	:PLUGin:PSWEep:ACQuisition:SMODe[?] on page 356
:SRTime[?]	:PLUGin:PSWEep:ACQuisition:SRTime[?] on page 356
:SFRom[?]	:PLUGin:PSWEep:ACQuisition:SFRom [?] on page 357
:STO[?]	:PLUGin:PSWEep:ACQuisition:STO[?] on page 357
:STEP[?]	:PLUGin:PSWEep:ACQuisition:STEP[?] on page 357
:COMPared[?]	:PLUGin:PSWEep:ACQuisition:COMPared[?] on page 358
:EBEnabled[?]	:PLUGin:PSWEep:ACQuisition:EBEnabled[?] on page 359
:ERRored[?]	:PLUGin:PSWEep:ACQuisition:ERRored[?] on page 359
:TBEratio[?]	:PLUGin:PSWEep:ACQuisition:TBEratio[?] on page 360
:TCLevel[?]	:PLUGin:PSWEep:ACQuisition:TCLevel[?] on page 360
:PREBer[?]	:PLUGin:PSWEep:ACQuisition:PREBer[?] on page 361
:SAENd[?]	:PLUGin:PSWEep:ACQuisition:SAENd[?] on page 362

:PLUGin:PSWEep:ACQuisition:ALOCation[?]

Syntax

:PLUGin:PSWEep:ACQuisition:ALOCation 'Identifier', <location-string>

:PLUGin:PSWEep:ACQuisition:ALOCation? 'Identifier'

Input Parameters

'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'

<location-string>: 'M*.DataIn*'

Description

This command sets the location or location group string against which the data acquisition is performed.

This query returns the current location or location group string against which measurement is configured for the data acquisition.

This SCPI is applicable for M8041A, M8051A, M8062A, M8045A, M8043A and M8046A.

Example :PLUG:PSWE:ACQ:ALOC 'ParameterSweep1', 'M1.DataIn1'

:PLUGin:PSWEep:ACQuisition:SLOCation[?]

Syntax :PLUGin:PSWEep:ACQuisition:SLOCation 'Identifier', <location-string>

:PLUGin:PSWEep:ACQuisition:SLOCation? 'Identifier'

Input Parameters 'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'

<location-string>: Specify the location/location group string against which the sweep property needs to be performed. For e.g. 'M1.DataIn1'

Description This command sets the location/location group string against which the sweep property needs to be performed.

This query returns the current sweep location/location group string from which property is configured for the sweep.

This SCPI is applicable for M8041A, M8051A, M8062A, M8045A, M8043A and M8046A.

Example :PLUG:PSWE:ACQ:SLOC 'ParameterSweep1', 'M1.DataIn1'

:PLUGin:PSWEep:ACQuisition:SWPArameter[?]

Syntax :PLUGin:PSWEep:ACQuisition:SWPArameter 'Identifier', <Sweep-Parameter>

:PLUGin:PSWEep:ACQuisition:SWPArameter? 'Identifier'

Input Parameters 'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'

<Sweep-Parameter>: Specify the sweep parameter present in current location or group.

Description This command sets the sweep parameter to be swept amongst the list of other parameters present in current location or group.

This query returns the current sweep parameter set on which sweep can be performed.

This SCPI is applicable for M8041A, M8051A, M8062A, M8045A, M8043A and M8046A.

Example :PLUG:PSWE:ACQ:SWPA 'ParameterSweep1', 'Delay'

:PLUGin:PSWEep:ACQuisition:SMODe[?]

Syntax

:PLUGin:PSWEep:ACQuisition:SMODe 'Identifier', <ABSolute | RELative>

:PLUGin:PSWEep:ACQuisition:SMODe? 'Identifier'

Input Parameters

'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'

ABS | REL

Description

This command sets the sweep mode. The choices are 'ABSolute' and 'RELative'.

In the 'ABSolute' mode, the property will sweep from the value specified in the 'Sweep From' parameter to a value specified in the 'Sweep To' parameter.

In the 'RELative' mode, the property will be sweep from a value relative to the current value specified in the 'Sweep From' parameter to a value relative to the current value specified in the 'Sweep To' parameter.

This query returns the current sweep mode.

This SCPI is applicable for M8041A, M8051A, M8062A, M8045A, M8043A and M8046A.

Example

:PLUG:PSWE:ACQ:SMOD 'ParameterSweep1', ABS

:PLUGin:PSWEep:ACQuisition:SRTime[?]

Syntax

:PLUGin:PSWEep:ACQuisition:SRTime 'Identifier', <NRf>

:PLUGin:PSWEep:ACQuisition:SRTime? 'Identifier'

Input Parameters

'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'

Description

This command sets the relaxing time between the sweep before reading the BER. The relaxing time can be between 0 sec to 60 sec.

This query returns the current set relaxing time.

This SCPI is applicable for M8041A, M8051A, M8062A, M8045A, M8043A and M8046A.

Example

:PLUG:PSWE:ACQ:SRT 'ParameterSweep1', 1E-2

:PLUGin:PSWEep:ACQuisition:SFRom[?]

Syntax :PLUGin:PSWEep:ACQuisition:SFRom 'Identifier', <NRf>

:PLUGin:PSWEep:ACQuisition:SFRom? 'Identifier'

Input Parameters 'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'

Description This command sets the value from where the sweep needs to be performed for that particular parameter.

This query returns the current sweep value.

This SCPI is applicable for M8041A, M8051A, M8062A, M8045A, M8043A and M8046A.

Example :PLUG:PSWE:ACQ:SFR 'ParameterSweep1', 1

:PLUGin:PSWEep:ACQuisition:STO[?]

Syntax :PLUGin:PSWEep:ACQuisition:STO 'Identifier', <NRf>

:PLUGin:PSWEep:ACQuisition:STO? 'Identifier'

Input Parameters 'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'

Description This command sets the value till the sweep needs to be performed for that particular parameter.

This query returns the current sweep value till the sweep is performed.

This SCPI is applicable for M8041A, M8051A, M8062A, M8045A, M8043A and M8046A.

Example :PLUG:PSWE:ACQ:STO 'ParameterSweep1', 1

:PLUGin:PSWEep:ACQuisition:STEP[?]

Syntax :PLUGin:PSWEep:ACQuisition:STEP 'Identifier', <NRf>

:PLUGin:PSWEep:ACQuisition:STEP? 'Identifier'

Input Parameters 'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'

Description This command sets the value as for how many steps user wants to sweep the parameter between the current 'Sweep From' and 'Sweep To' range. This query returns the current number of steps set for the acquisition.

This SCPI is applicable for M8041A, M8051A, M8062A, M8045A, M8043A and M8046A.

Example :PLUG:PSWE:ACQ:STEP 'ParameterSweep1', 100

:PLUGin:PSWEep:ACQuisition:COMPared[?]

Syntax :PLUGin:PSWEep:ACQuisition:COMPared 'Identifier', <NRf>

:PLUGin:PSWEep:ACQuisition:COMPared? 'Identifier'

Input Parameters 'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'

<location-string>: 'M*.DataIn*'

Description This command allows the user to specify the number of compared bits; that would be used as criteria for moving to the next measurement point in case when the sample delay optimization is 'NONE'.

This query returns the present setting.

This SCPI is applicable for M8041A, M8051A, M8062A, M8045A, M8043A and M8046A.

Example :PLUG:PSWE:ACQ:COMP 'ParameterSweep1', 1E+9

NOTE **The BER measurement processes at least as many bits as defined by the acquisition parameters, it stops when the end criteria is reached or exceeded. Due to update period and hardware granularities, an individual BER measurement will typically process more bits than configured.**

:PLUGin:PSWEep:ACQuisition:EBEnabled[?]

Syntax

```
:PLUGin:PSWEep:ACQuisition:EBEnabled 'Identifier', ON|OFF|1|0
:PLUGin:PSWEep:ACQuisition:EBEnabled? 'Identifier'
```

Input Parameters

'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'

Description

This command enabled/disables no. of errors feature. When this feature is enabled, user can specify additional number of errored bits to consider while moving to the next measurement points.

This query returns the present setting.

This SCPI is applicable for M8041A, M8051A, M8062A, M8045A, M8043A and M8046A.

Example

```
:PLUG:PSWE:ACQ:EBE 'ParameterSweep1', ON
:PLUG:PSWE:ACQ:EBE? 'ParameterSweep1' -> 1
```

:PLUGin:PSWEep:ACQuisition:ERRored[?]

Syntax

```
:PLUGin:PSWEep:ACQuisition:ERRored 'Identifier', <NRf>
:PLUGin:PSWEep:ACQuisition: ERRored? 'Identifier'
```

Input Parameters

'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'

Description

This command allows the user to specify the number of errored bits; that would be used as criteria for moving to the next measurement point in case when the sample delay optimization is 'NONE'..

This query returns the present setting.

This SCPI is applicable for M8041A, M8051A, M8062A, M8045A, M8043A and M8046A.

Example

```
:PLUG:PSWE:ACQ: ERR 'ParameterSweep1', 1E+5
:PLUG:PSWE:ACQ: ERR? 'ParameterSweep1'
1E+5
```

:PLUGin:PSWEep:ACQuisition:TBEratio[?]

Syntax

```
:PLUGin:PSWEep:ACQuisition:TBERatio 'identifier', <NRf>
:PLUGin:PSWEep:ACQuisition:TBERatio? 'identifier'
```

Input Parameters

'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'

<NRf>: Set the target BER.

Return Range

1E-18 to 1E-3

Description

This command sets the target error ratio of the accumulated results. This option is available when accumulation end is set to pass/fail.

This query returns the present setting.

This SCPI is applicable for M8041A, M8051A, M8062A, M8045A, M8043A and M8046A.

Example

```
:PLUG:PSWE:ACQ:TBER 'ParameterSweep1',1E-9
:PLUG:PSWE:ACQ:TBER? 'ParameterSweep1'
1E-9
```

:PLUGin:PSWEep:ACQuisition:TCLevel[?]

Syntax

```
:PLUGin:PSWEep:ACQuisition:TCLevel 'identifier', <NRf>
:PLUGin:PSWEep:ACQuisition:TCLevel? 'identifier'
```

Input Parameters

'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'

<NRf>: Set target confidence level.

Return Range

0.1% to 99.9%

Description

This command sets the target confidence level of the accumulated results. This option is available when accumulation end is set to pass/fail.

This query returns the current target confidence level.

This SCPI is applicable for M8041A, M8051A, M8062A, M8045A, M8043A and M8046A.

Example

```
:PLUG:PSWE:ACQ:TCL 'ParameterSweep1', 95
:PLUG:PSWE:ACQ:TCL? 'ParameterSweep1'
95
```

:PLUGin:PSWEep:ACQuisition:PREBer[?]

Syntax

:PLUGin:PSWEep:ACQuisition:PREBer 'identifier', <NONE|DAL|SDOP|TOP|FSP>

:PLUGin:PSWEep:ACQuisition:PREBer? 'identifier'

Input Parameters

'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'

NONE|DAL|SDOP|TOP|FSP

Return Value

NONE|DAL|SDOP|TOP|FSP

Description

This command sets the pre BER alignment criteria before it sweeps to next measurement point. The choices are:

- None - This parameter does not initiate any pre BER alignment.
- Data Alignment (DAL) - This parameter initiates pattern resynchronization. For details, refer to *M8000 Series Programming Guide*.
- Sample Delay Optimization (SDOP) - This parameter initiates search for the value of data/clock delay that puts the active clock edge in the center of the data eye, midway between the two relative delay points with a measured BER. For details, refer to *M8000 Series Programming Guide*.
- Threshold Optimization (TOP) - This parameter initiates search for the 0/1 threshold voltage midway between the two 0/1 threshold voltages with a measured BER. For details, refer to *M8000 Series Programming Guide*.
- Full Sample Point Optimization (FSP) - This parameter initiates auto alignment. For details, refer to *M8000 Series Programming Guide*.

This query returns the present setting.

This SCPI is applicable for M8041A, M8051A, M8062A, M8045A, M8043A and M8046A.

Example

```
:PLUG:PSWE:ACQ:PREB 'ParameterSweep1', DAL
```

```
:PLUG:PSWE:ACQ:PREB? 'ParameterSweep1'
```

```
DAL
```

:PLUGin:PSWEep:ACQuisition:SAENd[?]

Syntax

```
:PLUGin:PSWEep:ACQuisition:SAENd 'Identifier', <PFA|NOB>
```

```
:PLUGin:PSWEep:ACQuisition:SAENd? 'Identifier',
```

Input Parameters

'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'

<PFA | NOB>'

Description

This command sets the steps accumulation end parameter as Pass/Fail (PFA) or Number of Bits (NOB).

This query returns the present setting.

This SCPI is applicable for M8041A, M8051A, M8062A, M8045A, M8043A and M8046A.

Example

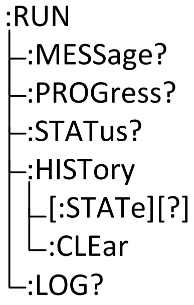
```
:PLUG:PSWE:ACQ:SAEN 'ParameterSweep1', PFA
```

```
:PLUG:PSWE:ACQ:SAEN? 'ParameterSweep1'
```

```
PFA
```

:PLUGin:PSWEep:RUN Subnode

This subnode has the following SCPI structure:



```
:RUN
├:MESSage?
├:PROGress?
├:STATus?
├:HISTory
│ ├[:STATe][?]
│ └:CLEar
└:LOG?
```

This subnode has the following commands:

Table 49

Name	Description under
:MESSage?	:PLUGin:PSWEep:RUN:MESSage? on page 364
:PROGress?	:PLUGin:PSWEep:RUN:PROGress? on page 364
:STATus?	:PLUGin:PSWEep:RUN:STATus? on page 364
:HISTory:[:STATe]	:PLUGin:PSWEep:RUN:HISTory[:STATe][?] on page 365
:HISTory:CLEar	:PLUGin:PSWEep:RUN:HISTory:CLEar on page 365
:LOG?	:PLUGin:PSWEep:RUN:LOG? on page 366

:PLUGin:PSWEep:RUN:MESSage?

Syntax :PLUGin:PSWEep:RUN:MESSage? 'Identifier'

Input Parameters 'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'

Description This command returns a string describing the state of the parameter sweep addressed by the measurement name identifier. Possible states include; Not Started, Running, Suspended, Finished, Error, or Stopped.

This SCPI is applicable for M8041A, M8051A, M8062A, M8045A, M8043A and M8046A.

Example :PLUG:PSWE:RUN:MESS? 'ParameterSweep1'

:PLUGin:PSWEep:RUN:PROGress?

Syntax :PLUGin:PSWEep:RUN:PROGress? 'Identifier'

Input Parameters 'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'

Return Range 0 to 1

Description This command returns a number in the range of 0.0 to 1.0 to indicate the progress of an parameter sweep measurement addressed by the measurement name identifier. 0.0 indicates that the measurement has not started and 1.0 indicates the measurement is finished.

This SCPI is applicable for M8041A, M8051A, M8062A, M8045A, M8043A and M8046A.

Example :PLUG:PSWE:RUN:PROG? 'ParameterSweep1'

:PLUGin:PSWEep:RUN:STATus?

Syntax :PLUGin:PSWEep:RUN:STATus? 'Identifier'

Input Parameters 'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'

Return Range 0|1

Description This command returns the running status of a parameter sweep measurement addressed by the measurement name identifier. 0 indicates the measurement is not running and a 1 indicates the measurement is running.

This SCPI is applicable for M8041A, M8051A, M8062A, M8045A, M8043A and M8046A.

Example :PLUG:PSWE:RUN:STAT? 'ParameterSweep1'

:PLUGin:PSWEep:RUN:HISTory[:STATe][?]

Syntax :PLUGin:PSWEep:RUN:HISTory[:STATe] 'Identifier', <0|1|ON|OFF>

:PLUGin:PSWEep:RUN:HISTory[:STATe]? 'Identifier'

Input Parameters 'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'

Return Range 0|1

Description This command enables/disables the storage of parameter sweep measurement results addressed by the measurement name identifier.

This query returns the current setting.

This SCPI is applicable for M8041A, M8051A, M8062A, M8045A, M8043A and M8046A.

Example :PLUG:PSWE:RUN:HIST 'ParameterSweep1', 1

:PLUGin:PSWEep:RUN:HISTory:CLEar

Syntax :PLUGin:PSWEep:RUN:HISTory:CLEar 'Identifier'

Input Parameters 'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'

Description This command deletes the parameter sweep measurement history addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A, M8045A, M8043A and M8046A.

Example :PLUG:PSWE:RUN:HIST:CLE 'ParameterSweep1'

:PLUGin:PSWEep:RUN:LOG?

Syntax :PLUGin:PSWEep:RUN:LOG? 'Identifier'

Input Parameters 'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'

Description This command returns logs for the addressed measurement.

This SCPI is applicable for M8041A, M8051A, M8062A, M8045A, M8043A and M8046A.

Example :PLUG:PSWE:RUN:LOG? 'ParameterSweep1'

The format of log returned is in the following way:

#XY Message Log

where X denotes the length of digits for Y, Y denotes the number of characters in the log then followed by the log message.

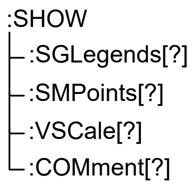
For example:

#10 means, there is 1 digit after 1 and there is 0 character in the log.

#2492 means, there are 3 digits after 2 and there are 492 characters in the log message.

:PLUGin:PSWEep:SHOW Subnode

This subnode has the following SCPI structure:



This subnode has the following commands:

Table 50

Name	Description under
:SGLegends	:PLUGin:PSWEep:SHOW:SGLegends[?] on page 367
:SMPoints	:PLUGin:PSWEep:SHOW:SMPoints[?] on page 368
:VSCale	:PLUGin:PSWEep:SHOW:VSCale[?] on page 368
COMment	:PLUGin:PSWEep:SHOW:COMment[?] on page 369

:PLUGin:PSWEep:SHOW:SGLegends[?]

Syntax

:PLUGin:PSWEep:SHOW:SGLegends 'Identifier', <0|1|ON|OFF>

:PLUGin:PSWEep:SHOW:SGLegends? 'Identifier'

Input Parameters

Identifier: Specify the plugin name. For e.g. 'ParameterSweep1'

<0|1|OFF|ON>: Specify the show/hide state for graph legends.

Return Range

0|1

Description

This command shows/hides the graph legends.

This query returns the current setting.

This SCPI is applicable for M8041A, M8051A, M8062A, M8045A, M8043A and M8046A.

Example :PLUG:PSWE:SHOW:SGL 'ParameterSweep1', 1

:PLUGin:PSWEep:SHOW:SMPoints[?]

Syntax :PLUGin:PSWEep:SHOW:SMPoints 'Identifier', <0|1|ON|OFF>

:PLUGin:PSWEep:SHOW:SMPoints? 'Identifier'

Input Parameters Identifier: Specify the plugin name. For e.g. 'ParameterSweep1'

<OFF|ON|0|1>: Disable/enable viewing the measurement points.

Return Range 0|1

Description This command disables/enables viewing of the measurement points in the GUI and associates it with a specific measurement name identifier. A 0 or OFF value disables viewing and a 1 or ON value enables viewing of the measurement points.

This query returns the current setting.

This SCPI is applicable for M8041A, M8051A, M8062A, M8045A, M8043A and M8046A.

Example :PLUG:PSWE:SHOW:SMP 'ParameterSweep1', 1

:PLUGin:PSWEep:SHOW:VSCale[?]

Syntax :PLUGin:PSWEep:SHOW:VSCale 'Identifier', <LOGarithmic | LINear >

:PLUGin:PSWEep:SHOW:VSCale? 'Identifier',

Input Parameters Identifier: Specify the plugin name. For e.g. 'ParameterSweep1'

Return Range LOG | LIN

Description This command specifies the vertical BER axis scale and associates it with a specific measurement name identifier. If LOGarithmic is specified then the BER values are in terms of a logarithmic scale. If LIN is specified then the BER values are in terms of a linear scale.

This query returns the current setting.

This SCPI is applicable for M8041A, M8051A, M8062A, M8045A, M8043A and M8046A.

Example :PLUG:PSWE:SHOW:VSC 'ParameterSweep1', LOG

:PLUGin:PSWEep:SHOW:COMment[?]

Syntax

```
:PLUGin:PSWEep:SHOW:COMment 'Identifier', <“Commnet”>
```

```
:PLUGin:PSWEep:SHOW:COMment? 'Identifier',
```

Input Parameters

Identifier: Specify the plugin name. For e.g. 'ParameterSweep1'

<“Commnet”> - Enter the desired comment.

Description

This command is used to enter a comment for each measurement. This comment will be displayed on the Measurement History tab.

This query returns the current setting.

This SCPI is applicable for M8041A, M8051A, M8062A, M8045A, M8043A and M8046A.

Example

```
:PLUG:PSWE:SHOW:COM 'ParameterSweep1', "Measurement with no. of bits"
```

```
:PLUG:PSWE:SHOW:COM? 'ParameterSweep1'
```

```
Measurement with no. of bits
```

N1000 Subsystem

The N1000 subsystem is used to control external clock recovery instruments that are themselves controlled from within the FlexDCA N1000-Series System Software.

NOTE

The minimum required version of the FlexDCA N1000-Series System Software is A.05.61.23.

It is possible to share one FlexDCA instance among multiple M8070B instances. However, to ensure correct access control, all M8070B instances have to be executed by the same PC.

All the commands used in this subsystem are basically identical with the SCPI commands of FlexDCA N1000-Series System Software. The only difference is that the M8070B does not use a suffix to address specific parts of the instrument, but instead always has the identifier parameter as first argument of the SCPI commands.

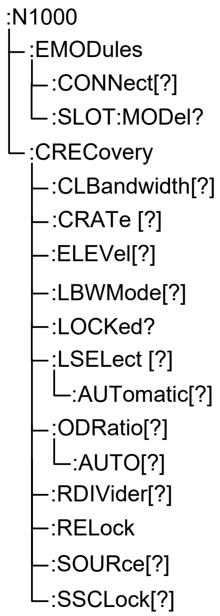
Detailed information about the features and SCPI commands of the supported clock recoveries can be found in the corresponding documentation:

N1076A Data Sheet
http://literature.cdn.keysight.com/litweb/pdf/5992-1620EN.pdf?id=2772197

N1076A User Guide
http://literature.cdn.keysight.com/litweb/pdf/N1076-90001.pdf?id=2784631

N1010A FlexDCA Online Help
http://www.keysight.com/main/redirector.jspx?action=ref&cname=EDITORIAL&ckey=2038334&lc=ger&cc=DE&nfr=-11143.0.00

This subsystem has the following SCPI structure:



```
:N1000
├─:EMODules
│  ├─:CONNect[?]
│  └─:SLOT:MODel?
└─:CRECovery
   ├─:CLBandwidth[?]
   ├─:CRATe [?]
   ├─:ELEVel[?]
   ├─:LBWMode[?]
   ├─:LOCKed?
   ├─:LSELect [?]
   │  └─:AUTomatic[?]
   ├─:ODRatio[?]
   │  └─:AUTO[?]
   ├─:RDIVider[?]
   ├─:RELock
   ├─:SOURce[?]
   └─:SSCLock[?]
```

This subsystem has the following commands:

Table 51

Name	Description under
:N1000:EMODules:SLOT:CONNect[?]	:N1000:EMODules:SLOT:CONNect[?] on page 372
:N1000:EMODules:SLOT:MODel?	:N1000:EMODules:SLOT:MODel? on page 373
:N1000:CRECovery:CLBandwidth[?]	:N1000:CRECovery:CLBandwidth[?] on page 373
:N1000:CRECovery:CRATe [?]	:N1000:CRECovery:CRATe? on page 374
:N1000:CRECovery:SOURce[?]	:N1000:CRECovery:ELEVel[?] on page 374
:N1000:CRECovery:LBWMode[?]	:N1000:CRECovery:LBWMode[?] on page 374
:N1000:CRECovery:LOCKed?	:N1000:CRECovery:LOCKed? on page 375
:N1000:CRECovery:LSELect[?]	:N1000:CRECovery:LSELect[?] on page 375

Name	Description under
:N1000:CRECovery:LSELect:AUTomatic[?]	:N1000:CRECovery:LSELect:AUTomatic[?] on page 376
:N1000:CRECovery:ODRatio[?]	:N1000:CRECovery:ODRatio? on page 377
:N1000:CRECovery:ODRatio:AUTO[?] on page 266	:N1000:CRECovery:ODRatio:AUTO? on page 377
:N1000:CRECovery:RDIVider[?]	:N1000:CRECovery:RDIVider? on page 377
:N1000:CRECovery:RELock	:N1000:CRECovery:RELock on page 378
:N1000:CRECovery:SOURce[?]	:N1000:CRECovery:SOURce[?] on page 378
:N1000:CRECovery:SSCLock[?]	:N1000:CRECovery:SSCLock [?] on page 379

:N1000:EMODules:SLOT:CONNect[?]

Syntax

:N1000:EMODules:SLOT:CONNect <identifier>,{ON | OFF | 1 | 0}

:N1000:EMODules:SLOT:CONNect? <identifier>

Input Parameters

'identifier': 'DCA*.SLOT*'

ON | OFF | 1 | 0

Return Range

ON | OFF | 1 | 0

Description

This command enables usage of the addressed slot by M8070B. In case of using multiple M8070B instances, make sure to enable a slot only in exactly one M8070B instance.

This query returns the present state.

This SCPI is applicable for M8046A.

Example

:N1000:EMOD:SLOT:CONN 'DCA1.SLOT1',ON

NOTE

This command is not the same as the similar FlexDCA command :EMODules:SLOT{n}:CONNect[?] {ON | OFF | 1 | 0}

The above command must be sent to FlexDCA prior starting up the M8070B software instances that shall control FlexDCA!

Managing the slot usage and connection state must be done using the FlexDCA commands or user interface before starting up M8070B.

:N1000:EMODules:SLOT:MODel?

Syntax :N1000:EMODules:SLOT:MODel? <identifier>

Input Parameters 'identifier': 'DCA*.SLOT*'

Description This query returns the model number of all DCA related modules which are connected to M8070B system software.

This SCPI is applicable for M8046A.

Example :N1000:EMOD:SLOT:MOD? 'DCA1.SLOT1'

:N1000:CRECovery:CLBandwidth[?]

Syntax :N1000:CRECovery:CLBandwidth <identifier>,<bandwidth>

:N1000:CRECovery:CLBandwidth? <identifier>

Input Parameters 'identifier': 'DCA*.SLOT*'

Description This command sets the fix loop bandwidth.

This query returns the present state.

This SCPI is applicable for M8046A.

Example :N1000:CREC:CLB 'DCA1.SLOT1',10e6

For detailed documentation, see :CRECovery{n}:CLBandwidth in the FlexDCA online help.

:N1000:CRECovery:CRATe?

Syntax

:N1000:CRECovery:CRATe? <identifier>

Input Parameters

'identifier': 'DCA*.SLOT*'

Description

This query returns the clock recovery's symbol rate.

This SCPI is applicable for M8046A.

Example

:N1000:CREC:CRAT? 'DCA1.SLOT1'

For detailed documentation, see :CRECovery{n}:CRATe in the FlexDCA online help.

:N1000:CRECovery:ELEVel[?]

Syntax

:N1000:CRECovery:ELEVel <identifier>,<NRf>

:N1000:CRECovery:ELEVel? <identifier>

Input Parameters

'identifier': 'DCA*.SLOT*'

Description

This command the equalizer at the electrical inputs of the clock recovery. The equalizer level range is 0 to 100.

This query returns the current settings.

This SCPI is applicable for N1076B and N1078A.

Example

:N1000:CREC:ELEVel 'DCA1.SLOT1',12

For detailed documentation, see :CRECovery{n}:ELEVel in the FlexDCA online help.

:N1000:CRECovery:LBWMode[?]

Syntax

:N1000:CRECovery:LBWMode <identifier>,{FIXed | RDEPendent}

:N1000:CRECovery:LBWMode? <identifier>

Input Parameters 'identifier': 'DCA*.SLOT*'

Description This command sets the loop bandwidth mode to either fixed or symbol rate dependent.

This query returns the present state.

This SCPI is applicable for M8046A.

Example :N1000:CREC:LBWM 'DCA1.SLOT1',RDEP

For detailed documentation, see :CRECovery{n}:LBWMode in the FlexDCA online help.

:N1000:CRECovery:LOCKed?

Syntax :N1000:CRECovery:LOCKed? <identifier>

Input Parameters 'identifier': 'DCA*.SLOT*'

Description This query displays the clock recovery’s lock status.

Response: 1 (for locked) or 0 (for unlocked)

This SCPI is applicable for M8046A.

Example :N1000:CREC:LOCK? 'DCA1.SLOT1'

For detailed documentation, see :CRECovery{n}:LOCKed the FlexDCA online help.

:N1000:CRECovery:LSELect[?]

Syntax :N1000:CRECovery:LSELect <identifier>, <LOOP1 | LOOP2 | LOOP3 | LOOP4>

:N1000:CRECovery:LSELect? <identifier>,

Input Parameters
'identifier': 'DCA*.SLOT*'

Description
This command selects the type 2 loop transition frequency.

This query returns the present state.

This SCPI is applicable for M8046A.

Example
:N1000:CREC:LSEL 'DCA1.SLOT1', LOOP1

For detailed documentation, see :CRECovery{n}:LSELect in the FlexDCA online help.

:N1000:CRECovery:LSELect:AUTomatic[?]

Syntax
:N1000:CRECovery:LSELect:AUTomatic <identifier>, <ON | OFF | 1 | 0>

:N1000:CRECovery:LSELect:AUTomatic? <identifier>

Input Parameters
'identifier': 'DCA*.SLOT*'

ON | OFF | 1 | 0

Description
This command enables or disables automatic selection of the type 2 loop transition frequency.

This query returns the present state.

This SCPI is applicable for M8046A.

Example
:N1000:CREC:LSEL:AUTO 'DCA1.SLOT1',ON

For detailed documentation, see :CRECovery{n}:LSELect:AUTO in the FlexDCA online help.

:N1000:CRECovery:ODRatio?

Syntax	:N1000:CRECovery:ODRatio? <identifier>
Input Parameters	'identifier': 'DCA*.SLOT*'
Return Range	SUB32 | SUB16 | SUB8 | SUB4 | SUB2 | UNITy | SUPer2 | SUPer4 | SUPer8
Description	This query returns the recovered clock’s output divide ratio. This SCPI is applicable for M8046A.
Example	:N1000:CREC:ODR? 'DCA1.SLOT1' For detailed documentation, see :CRECovery{n}:ODRatio in the FlexDCA online help.

:N1000:CRECovery:ODRatio:AUTO?

Syntax	:N1000:CRECovery:ODRatio:AUTO? <identifier>
Input Parameters	Input Parameters 'identifier': 'DCA*.SLOT*'
Return Range	ON | OFF | 1 | 0
Description	This query returns the present state. This SCPI is applicable for M8046A.
Example	:N1000:CREC:ODR:AUTO? 'DCA1.SLOT1' For detailed documentation, see :CRECovery{n}:ODR:AUTO in the FlexDCA online help.

:N1000:CRECovery:RDIVider?

Syntax	:N1000:CRECovery:RDIVider? <identifier>,
Input Parameters	'identifier': 'DCA*.SLOT*'
Description	This query returns the loop bandwidth symbol rate divider. This SCPI is applicable for M8046A.
Example	:N1000:CREC:RDIV? 'DCA1.SLOT1'

For detailed documentation, see :CRECovery{n}:RDIVider in the FlexDCA online help.

:N1000:CRECovery:RELock

Syntax :N1000:CRECovery:RELock <identifier>

Input Parameters 'identifier': 'DCA*.SLOT*'

Description This command initiates a re-lock of the clock recovery.

NOTE **The re-lock operation can take quite some time. Especially when the clock recovery cannot lock to the received signal. Therefore, the VISA timeout shall be increased to well above 10 seconds to ensure that the controlling program is waiting long enough for the re-lock operation to finish.**

This SCPI is applicable for M8046A.

Example :N1000:CREC:REL 'DCA1.SLOT1';*OPC?

For detailed documentation, see :CRECovery{n}:RELock in the FlexDCA online help.

:N1000:CRECovery:SOURce[?]

Syntax :N1000:CRECovery:SOURce<identifier>, {ELECtrical | EINVerted | DIFFerential}

:N1000:CRECovery:SOURce?<identifier>

Input Parameters 'identifier': 'DCA*.SLOT*'

ELECtrical | EINVerted | DIFFerential

Description This command selects the input of the clock recovery.

This query returns the present state.

This SCPI is applicable for M8046A.

Example :N1000:CREC:SOUR 'DCA1.SLOT1',DIFF

For detailed documentation, see :CRECovery{n}:SOURce in the FlexDCA online help.

:N1000:CRECovery:SSCLock [?]

Syntax

:N1000:CRECovery:SSCLock [?] <identifier>,{OFF | ON | 0 | 1}

:N1000:CRECovery:SSCLock? <identifier>

Input Parameters

'identifier': 'DCA*.SLOT*'

ELECtrical | EINVerted | DIFFerential | AUXilliary

Description

This command turns on the ability to use the nominal symbol rate when recovering the clock on Spread Spectrum Clock (SSC) signals. Normally, FlexDCA uses the average symbol rate to recovery the clock. However, when performing clock recovery on a SSC signal, using the nominal symbol rate may be required. The default setting is off.

This command requires FlexDCA revision A.05.80 and above, and is only available for clock recovery modules with Option JSA (see the Electrical and Optical Clock Data Recovery Solutions data sheet, 5992-1620EN)

This query returns the present state.

This SCPI is applicable for M8046A.

Example

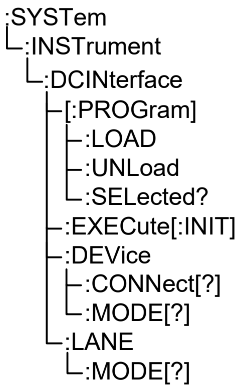
:N1000:CREC:SSCL 'DCA1.SLOT1',ON

For detailed documentation, see :CRECovery{n}:SSCLock in the FlexDCA online help.

SYSTem Subsystem

The SYSTem subsystem is used for general system functions.

This subsystem has the following SCPI structure:



```
:SYSTem
└:INSTrument
  └:DCINterface
    ├[:PROGram]
    │├:LOAD
    │├:UNLoad
    │└:SELected?
    ├:EXECute[:INIT]
    ├:DEVice
    │├:CONNect[?]
    │└:MODE[?]
    └:LANE
      └:MODE[?]
```

This subsystem has the following commands:

Table 52

Name	Description under
:DCINterface[:PROGram]:LOAD	:SYSTem:DCINterface[:PROGram]:LOAD on page 381
:DCINterface[:PROGram]:UNLoad	:SYSTem:DCINterface[:PROGram]:UNLoad on page 381
:DCINterface[:PROGram]:SELected?	:SYSTem:DCINterface[:PROGram]:SELected? on page 381
:DCINterface:EXEcute[:INIT]	:SYSTem:DCINterface:EXEcute[:INIT] on page 382
:DCINterface:DEVice:CONNect	:SYSTem:DCINterface:DEVice:CONNect[?] on page 382
:DCINterface:DEVice:MODE	:SYSTem:DCINterface:DEVice:MODE[?] on page 383
:DCINterface:LANE:MODE	:SYSTem:DCINterface:LANE:MODE[?] on page 383

:SYSTem:DCINterface[:PROGram]:LOAD

Syntax :SYSTem:DCINterface[:PROGram]:LOAD "PathToPythonProgramFile"

Description This commands loads the given Python script into the DUT control interface. This DUT control interface program can be used to read the bit error rate of an externally connected DUT and provide these to the M8070B software as if the counters are part of the instrument. These values can then be used by the built in Bit Error Ratio measurement or Jitter Tolerance measurement.

In addition the script can implement functions that extend the measurements in order to adapt the measurement to the specific test setup and device under test.

Python scripts can be part of settings and shared areas. The addressing scheme looks like:

"Factory/HookExample.py" // Read only factory shared area

"Shared/HookExample.py" // User shared area

"Current/HookExample.py" // Current setting

Example :SYSTem:DCINterface:PROGram:LOAD "factory/DutCounterIntegration.py"

:SYSTem:DCINterface[:PROGram]:UNLoad

Syntax :SYSTem:DCINterface[:PROGram]:UNLoad

Description This command unloads the currently used DUT Control Interface script and all the resources that the script integrated into the system.

Example :SYSTem:DCINterface:PROGram:UNLoad

:SYSTem:DCINterface[:PROGram]:SELected?

Syntax :SYSTem:DCINterface[:PROGram]:SELected?

Description This query returns the DUT Control Interface script that is currently in use. If the DUT Control Interface is not using a script, an empty string is returned.

Example :SYSTem:DCINterface:PROGram:SELected?

:SYSTem:DCINterface:EXEcute[:INIT]

Syntax :SYSTem:DCINterface:EXEcute[:INIT] 'Location'[,"ArgumentsForInit"]

Input Parameters 'Location': '*.Lane*'

Description This command matches the DUT Control Interface hook function DUT_Init(location, argument).

Sending this command results in executing the function DUT_Init with the given parameters if it is implemented by the script that is currently in use (see syst:dcin:load).

The Location argument shall be one of the locations defined by the scripts DUT_getLocations, but as full system location identifier.

For example, if,DUT_getLocations defines the locations "Lane1" and "Lane2", then valid Location arguments are "DCI.Lane1" and "DCI.Lane2"

The argument string is not checked anyway, but simply passed to the script implementation of DUT_Init.

Example :SYSTem:DCINterface:execute:init "DCI.Lane1"

:SYSTem:DCINterface:DEVice:CONNect[?]

Syntax :SYSTem:DCINterface:DEVice:CONNect 'identifier', <ON|OFF|1|0>

:SYSTem:DCINterface:DEVice:CONNect 'identifier'?

Input Parameters 'identifier': "DCI.Control"

Description This command relates to the DUT Control Interface hook functions DUT_connect and DUT_disconnect.

When these functions are implemented by the currently used script (see syst:dcin:load), then this command is controlling the Connection parameter under the function Device for the Control location of the device under test.

The location argument must be the Control location as full system location identifier as displayed in the parameter editor (e.g. "DCI.Control").

This query returns the current setting.

Example :SYSTem:DCINterface:DEVice:CONNect 'DCI.Control', 1

:SYSTem:DCINterface:DEVice:MODE[?]

Syntax

```
:SYSTem:DCINterface:DEVice:MODE 'identifier', 'mode'
:SYSTem:DCINterface:DEVice:MODE? 'identifier'
```

Input Parameters

'identifier': 'DCI.Control'

'mode': Individual modes or configurations.

Description

This command relates to the DUT Control Interface hook functions DUT_getDeviceModes and DUT_setDeviceMode.

When these functions are implemented by the currently used script (see syst:dcin:load), then this command is controlling the Mode parameter under the function Device for the Control location of the device under test.

The location argument must be the Control location as full system location identifier as displayed in the parameter editor (e.g. "DCI.Control").

The mode argument must be one of the strings returned by DUT_getDeviceModes.

This can be used to control configurations that are valid for the whole device under test.

When sending this command in one SCPI transaction with :syst:dcin:lane:mode, then the device mode hook (DUT_setDeviceMode) will be executed before the lane specific mode (DUT_setLaneMode).

This query returns the current setting.

Example

```
:SYSTem:DCINterface:DEVice:MODE 'DCI.Control','Configuration1'
```

:SYSTem:DCINterface:LANE:MODE[?]

Syntax

```
:SYSTem:DCINterface:LANE:MODE "Location", "mode"
:SYSTem:DCINterface:LANE:MODE? "Location"
```

Input Parameters

'Location': '*.Lane*'

'mode': Individual modes or configurations.

Description

This command relates to the DUT Control Interface hook functions DUT_getLaneModes and DUT_setLaneMode.

When these functions are implemented by the currently used script (see syst:dcin:load), then this command is controlling the Mode parameter under the function Lane for the given lane of the device under test.

The location argument must be the Control location as full system location identifier as displayed in the parameter editor (e.g. "DCI.Lane1").

The mode argument must be one of the strings returned by DUT_getLaneModes.

This can be used to control configurations that are valid for an individual lane or channel of the device under test.

When sending this command in one SCPI transaction with :syst:dcin:dev:mode, then the device mode hook (DUT_setDeviceMode) will be executed before the lane specific mode (DUT_setLaneMode).

This query returns the current setting.

Example :SYSTem:DCINterface:LANE:MODE 'DCI.Lane1','MODE1'

Index

KEYSIGHT
TECHNOLOGIES

This information is subject to
change without notice.

Edition 9.0, November 2023

M8000-91B10

www.keysight.com