M-PHY
Receiver Automated Test Solution
Printable Application Help

077-0643-03

Tektronix

M-PHY
Receiver Automated Test Solution
Printable Application Help

Supports M-PHY Receiver Automated Test Solution, release 3.1.0 and above.

www.tektronix.com
077-0643-03

Tektronix

Contacting Tektronix

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

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

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

Table of Contents

Setting up and configuring tests

Running tests and viewing results

TekExpress programmatic interface

HS testing procedures and examples

PWM testing procedures and examples

Reference

Welcome

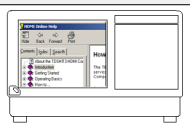
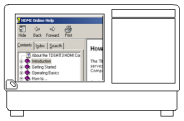
The Tektronix M-PHY Rx Automated Test software runs on Tektronix real-time oscilloscopes that are based on Windows 7 computer operating systems. M-PHY Rx provides an automated, simple, and efficient way to test M-PHY Receiver interfaces and devices consistent to the requirements of the MPHY Base Specification and Conformance Test Specification.

Getting help and support

Related documentation

The following manuals are available as part of the TekExpress M-PHY Rx Automated Solution documentation set.

Table 1: Product documentation

Item	Purpose	Location
Online Help	In-depth operation and UI help	
PDF of the Online Help	In-depth operation and UI help	

See also *Technical support*

Conventions used in help

Online Help uses the following conventions:

- The term “DUT” is an abbreviation for Device Under Test.
- The term “select” is a generic term that applies to the two mechanical methods of choosing an option: using a mouse or using the touch screen.

Technical support

Tektronix values your feedback on our products. To help us serve you better, please send us your suggestions, ideas, or comments on your application or oscilloscope. Contact Tektronix through mail, telephone, or the Web site. See *Contacting Tektronix* for more information.

When you contact Tektronix Technical Support, please include the following information (be as specific as possible):

General Information

- All instrument model numbers
- Hardware options, if any
- Probes used
- Your name, company, mailing address, phone number, FAX number
- Please indicate if you would like to be contacted by Tektronix about your suggestion or comments.

Application Specific Information

- Software version number
- Description of the problem such that technical support can duplicate the problem
- If possible, save the setup files for all the instruments used and the application.
- If possible, save the TekExpress setup files, log.xml, *.TekX (session files and folders), and status messages text file.

Getting started

Installing the software

Compatibility

The TekExpress M-PHY Rx application runs on the following Tektronix oscilloscopes:

- DPO/DSA/MSO70604 C/D M-PHYRX (only for HS-Gear 1)
- DPO/DSA/MSO71254 C/D M-PHYRX (only for HS-Gear 1 and 2)
- DPO/DSA/MSO72304 C/D M-PHYRX
- DPO/DSA/MSO71604 C M-PHYRX (only for HS-Gear 1 and 2)
- DPO/DSA/MSO72004 C M-PHYRX (only for HS-Gear 1 and 2)
- DPO/DSA/MSO72504 DX M-PHYRX
- DPO/DSA/MSO73304 D M-PHYRX
- DPO/MSO72304 DX M-PHYRX
- DPO/DSA/MSO70804 C M-PHYRX (only for HS Gear 1)

The TekExpress M-PHY Rx application can be used with the following probes:

- Tektronix P7380SMA for HS Gear 1
- P73xx SMA for HS Gear 2
- P76xx for HS Gear 3
- P73xx for all PWM Gears

See Also. *Minimum system requirements*

Minimum system requirements

The following table shows the minimum system requirements for an oscilloscope to run TekExpress.

Table 2: System requirements

System component	Requirement
Oscilloscope	■ DPO/DSA/70K C/D/DX Series, MSO 70K. Refer *Compatibility* ■ Oscilloscope with GPIB set as the controller. [1] ■ Error Detector (ERRDT) and Serial Trigger Unit (STU)
Arbitrary Waveform Generator (AWG)	■ AWG7082/AWG7102 or above, for HS-GEAR1 [2] ■ AWG7122C for up to HS-GEAR2 ■ AWG7122C with Interleave (option 06), for up to HS-GEAR3
Processor	Same as the oscilloscope
Operating System	Same as the oscilloscope: ■ Windows 7 64-bit
Memory	Same as the oscilloscope
Hard Disk	Same as the oscilloscope.
Display	Same as the oscilloscope [3]
Firmware	TekScope v7.2.0.4 or later
Software	■ Microsoft .NET 4.0 Framework ■ Microsoft Internet Explorer 6.0 SP1 or later ■ Adobe Reader 7.0 or equivalent software for viewing portable document format (PDF) files

[1] controller is the instrument which controls the scope.
[2] M-PHYRX supports non-interleave channels.
[3] If TekExpress is running on an instrument having a video resolution lower than 800x600 (for example, sampling oscilloscope), it is recommended that you connect a secondary monitor, which must be enabled before launching the application.

System component	Requirement
Probes	Tektronix P7380SMA or higher bandwidth, P73xx, and P76xx
Other Devices	■ Matched pair of SMA cables, two-set minimum for single lane ■ Two BIAS Tee (PSPL5542 or PSPL5544 with option 292JJ) for HS Gear 3 ■ Microsoft compatible mouse or compatible pointing device ■ PCI-GPIB or equivalent interface for instrument connectivity [4] ■ GPIB cable for instrument connectivity

See also. *Compatibility*

Equipment connection setup

Install the software

The TekExpress M-PHY Rx web installer page provides link to the software package which is to be installed on the oscilloscope. For a list of compatible instruments, see *Compatibility*.

- TekExpressM-PHYRXWebInstaller.exe: Installs the TekExpress M-PHY Rx application

To download and install the files:

1. Close the TekScope application.
2. Go to the www.tek.com Web site and search for M-PHY Rx to locate the installation file. Download the file TekExpressM-PHYRXWebInstaller.exe.
3. Double-click the executable file to extract the installation files.

   After extraction, the installer launches and the software automatically installs in the following location:

   - C:\Program Files\Tektronix\TekExpress\TekExpress M–PHY Rx
4. To open the application, open the TekScope application and then select **TekExpress M-PHY Rx** from the Analyze menu.

See also. *Minimum system requirements*

Compatibility

[4] If TekExpress is installed on a Tektronix oscilloscope, the virtual GPIB port will be used by TekExpress for communicating with oscilloscope applications. If external GPIB communication devices such as USB-GPIB-HS or equivalent are used for instrument connectivity, make sure that the Talker Listener utility is enabled in the DPO/DSA/MSO oscilloscope GPIB menu. For ease of use, connect to an external (secondary) monitor.

Activate the license

Activate the license using the Option Installation wizard on the oscilloscope. The oscilloscope Online Help has instructions for using the Options Installation window to activate licenses for installed applications. Follow these steps to activate the TekExpress M-PHY Rx license:

1. From the oscilloscope menu, click **Utilities** > **Option Installation**.

   The TekScope Option Installation wizard opens.

2. Press the **F1** key on the oscilloscope keyboard to open the Option Installation help topic. Follow the directions in the topic to activate the license.

See also. *View version and license information*

View license information

To view license information:

1. From the oscilloscope Help menu, select **About TekScope**.

   The Options section in the dialog box displays a list of installed options, including M-PHY Rx: M-PHY receiver solution.

2. To view the Option key, look in the Option Installation Key section.

See also. *Activate the license*

Options menu

Introduction to the application

M-PHY Rx application overview

TekExpress is the Tektronix Test Automation Framework, developed to support your current and future test automation needs. TekExpress uses a highly modular architecture that lets you deploy automated test solutions for various standards in a relatively short time.

Key Features. M-PHY Rx has the following key features:

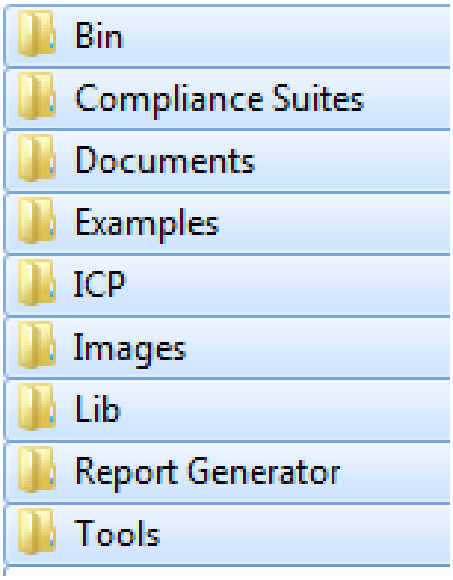
- Allows conformance testing to the Base specification version 2.0 and Compliance Test Specification (CTS v1.0)
- Automated Testing:
  - Reduces the amount of time required to conduct testing
  - Enables you to test devices faster
- Tests coverage: 9 HS measurements, 7 PWM measurements, and Margin test for HS mode.
- Auto calibration of setup for HS mode and HS Margin tests.
- Selective testing: Allows you to select individual tests or test groups in the tree-structure.
- Reliable Results: Avoids repeated testing through accurate and reliable results from a single run
- Integrated BER:

Bit Error Rate

  - Leverage Bit-Error-Rate or Error-Count testing using oscilloscope-integrated ERRDT software in the background.
  - Dialog box to support manual error entry.
  - Custom script support for reading error.
- Customize the setup: Provision to configure the test setup according to the DUT-supported configuration, such as the HS Gear A or B, and test time.
- Detailed test reporting:
  - Provides a Pass/Fail summary table. (For details, see *Results panel*)
  - Provides a consolidated report for all tests
  - Provides additional information such as test setup hardware and software details, signal type selected, measured value, execution time, and user-comments for each measurement

Application directories and usage

The application directory and associated files are organized as follows:



The following table lists the default directory names and their usage:

Table 3: Application directories and usage

Directory names	Usage
InstallDir\TekExpress\TekExpress MIPI M-PHY Rx	Contains the application and associated files
TekExpress M-PHY Rx\Bin	Contains miscellaneous M-PHY Rx application libraries
TekExpress M-PHY Rx\Compliance Suites	Contains compliance-specific files
TekExpress M-PHY Rx\Images	Contains Tektronix logo images used for reports
TekExpress M-PHY Rx\Documents	Contains the technical documentation for the M-PHY Rx application
TekExpress M-PHY Rx\Examples	Contains various support files
TekExpress M-PHY Rx\ICP	Contains instrument and M-PHY Rx application-specific interface libraries
TekExpress M-PHY Rx\Lib	Contains utility files specific to the M-PHY Rx application
TekExpress M-PHY Rx\Report Generator	Contains Excel Active X interface Library for Report Generation
TekExpress M-PHY Rx\Tools	Contains instrument and M-PHY Rx application-specific files

See also. *View test-related files*

File name extensions

The TekExpress M-PHY Rx application uses the following file name extensions:

File name extension	Description
.TekX	Session files are saved in this format but the extensions may not be displayed.
.py	The test sequence file
.xml	The encrypted XML file that contains the test-specific configuration information The log file extension is also xml
.wfm	The test waveform file
.mht	Test result reports are saved in this format by default. Test reports can also be *saved in HTML format*
.pdf	Application help document

See also. *View test-related files*

Application directories and usage

Operating basics

Start the application

When you open the application after installation, the application checks for a file called Resources.xml located in the My TekExpress folder. If this file is not found, instrument discovery is performed before launching M-PHY Rx. The Resources.xml file contains information regarding instruments that are available on your network. If the application license was not installed using the TekScope menu **Utilities > Option Installation** selection, the application will allow 10 free launches. Each time you open the application without supplying a valid license key, one of the free trials is used.

To run the M-PHY Rx application, do any of the following:

- From the TekScope Analyze menu, select **Analyze** > **TekExpress M-PHY Rx**.
- Double-click any saved session file.

See also *Activate the license*

Exit the application

Use the following method to exit the application:

NOTE. *Using other methods to exit the application results in abnormal termination of the application.*

1. Click  on the application title bar.
2. Do one of the following:
   - If you have an unsaved session or test setup open, you are asked to save it before exiting. To save it, click **Yes**. Otherwise click **No**. The application closes.
   - A message box appears asking if you really want to exit TekExpress. To exit, click **Yes**.

Application controls and menus

Application controls

Table 4: Application controls descriptions

Item	Description
Options menu	Opens the Options menu for access to global controls
Panels	Visual frames with sets of related options
Command buttons	Buttons that initiate an immediate action such as the Start, Stop, Pause, Continue, and Clear command buttons
Start button	 Use the Start button to continuously acquire and accumulate measurements. If prior acquired measurements have not been cleared, the new measurements are added to the existing set.
Stop button	 Use the Stop button to abort the test.
Pause \ Continue button	 Use the Pause button to temporarily interrupt the current acquisition. When a test is paused, the button name changes to Continue.
Clear button	 Use the Clear button to clear all existing measurement results. Adding or deleting a measurement, or changing a configuration parameter of an existing measurement, also clears measurements. This is to prevent the accumulation of measurement statistics or sets of statistics that are not coherent. This button is available only on the *Results panel*.
Application window move icon	 Place the cursor over the three-dot pattern in the upper left corner of the application window. When the cursor changes to a hand, drag the window to the desired location.

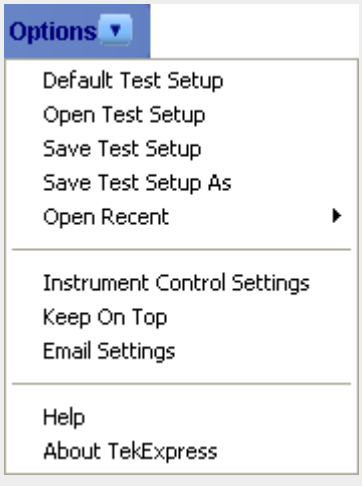
Options menu

The Options menu is located in the upper right corner of the application.

The Options menu has the following selections:

Menu	Function
Default Test Setup	Opens an untitled test setup with defaults selected
Open Test Setup	Opens a saved test setup
Save Test Setup	Saves the current test setup selections
Save Test Setup As	Creates a new test setup based on an existing one
Open Recent	Displays a menu of recently opened test setups to select from
Instrument control settings	Shows the list of instruments connected to the test setup and allows you to locate and refresh connections to connected instruments
Keep On Top	Keeps the TekExpress M-PHY Rx utility on top of other open windows on the desktop
Email settings	Use to configure email options for test run and results notifications
Help	Displays TekExpress Help
About TekExpress	■ Displays application details such as software name, version number, and copyright ■ Provides access to *license information* for your M-PHY Rx installation ■ Provides a link to the Tektronix Web site

Options menu

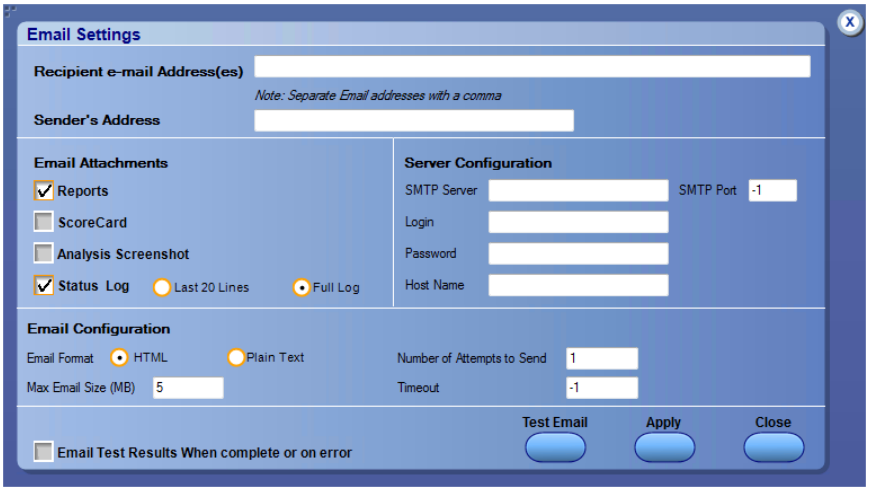


See also. *Application controls*

Email settings

Use the Email Settings utility to *configure email notifications* if you want M-PHY Rx to notify you when a test completes, produces an error, or fails. Select the type of test run information to include in the notification, such as test reports and test logs, the email message format, and the email message size limit.

NOTE. *Recipient email address, sender's address, and SMTP Server are mandatory fields.*



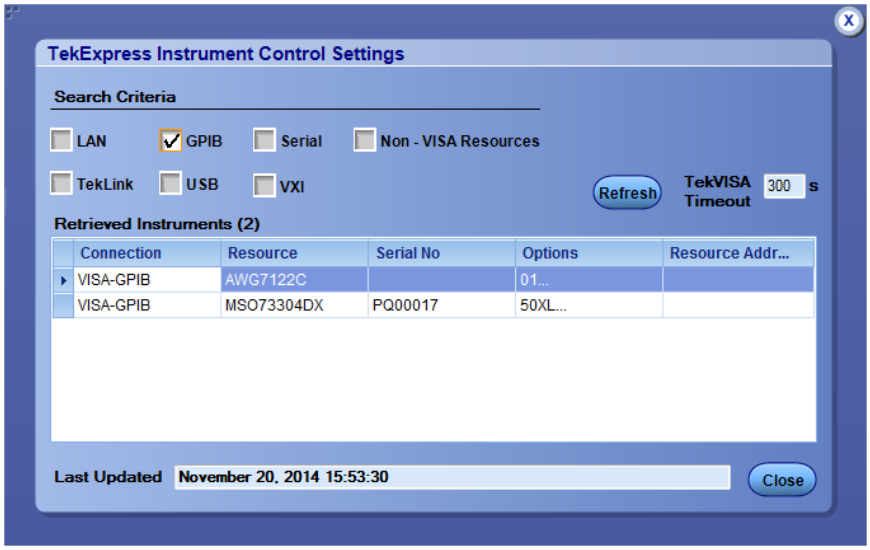
See Also. *Options menu*

Select test notification preferences

Instrument control settings

The Instrument Control Settings dialog box shows the list of resources found on different connections. It serves two purposes:

- Discovers the connected instruments
- Confirms the instrument connection setup



Use the Instrument Control Settings feature to *search for connected instruments* and view instrument connection details. Connected instruments displayed here can be selected for use in the Global Settings tab in the configuration section. See step 1 of *Configure tests* for details.

See also. *Options menu*

Panels

About panels. TekExpress M-PHY Rx has the following main panels:

Table 5: Application panels

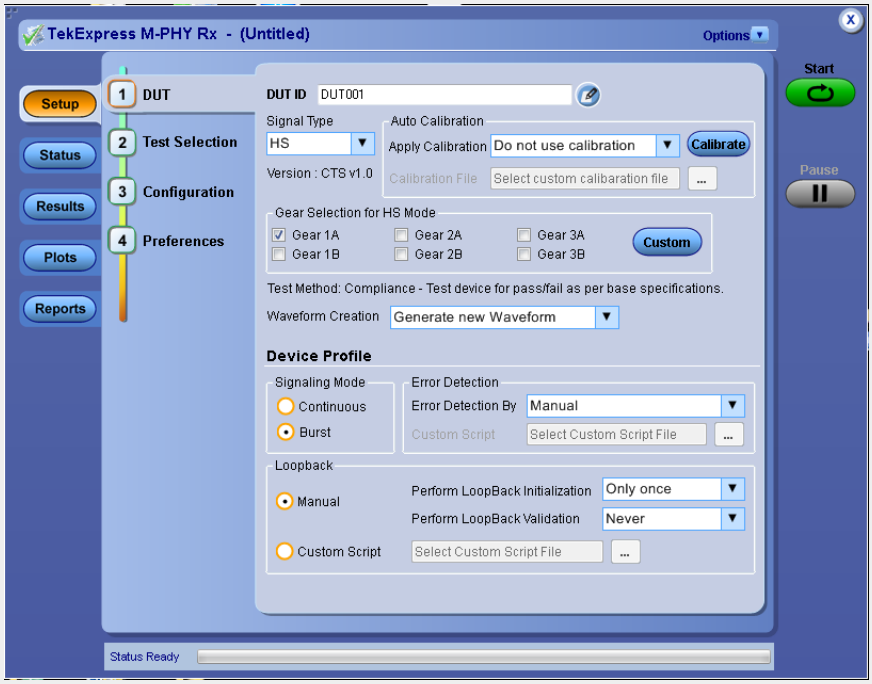
Panel Name	Purpose
Setup	The Setup panel allows you to configure the test setup. Use this panel to: ■ *Select the device paramaters*. ■ *Select the test(s)*. ■ *Configure the selected tests*. ■ *Select test notification preferences*.
Status	View the progress and analysis status of the selected tests, and view test logs.
Results	View a summary of test results and select results viewing preferences.
Reports	Browse for reports, save reports as specific file types, specify report naming conventions, select report content to include (such as summary information, detailed information, user comments, setup configuration, application configuration), and select report viewing options.

See also. *Application controls*

Setup panel. The Setup panel has numbered tabs to guide you through the test setup process:

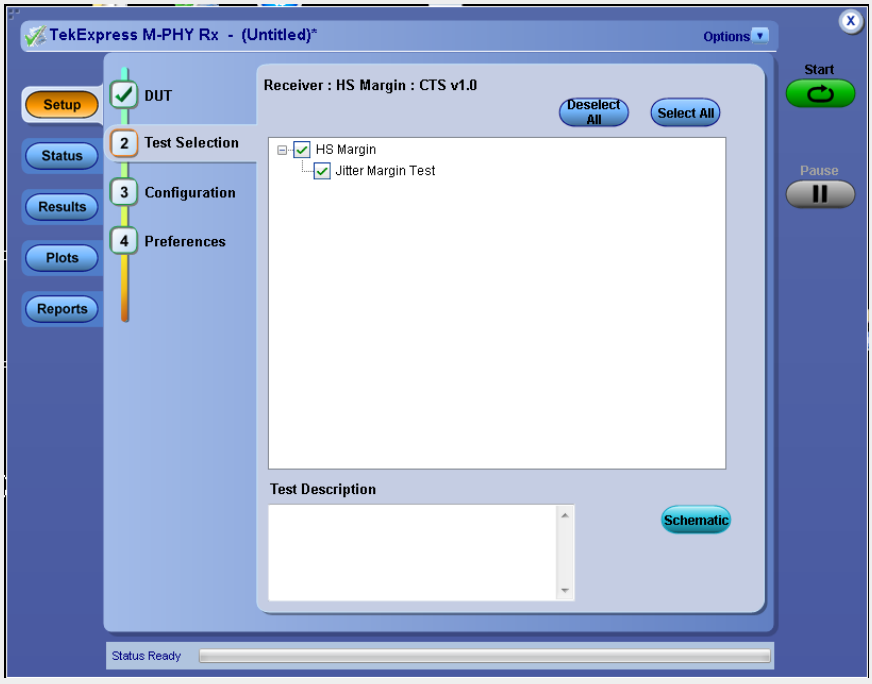
1. DUT

Setup Panel DUT tab – HS Signals

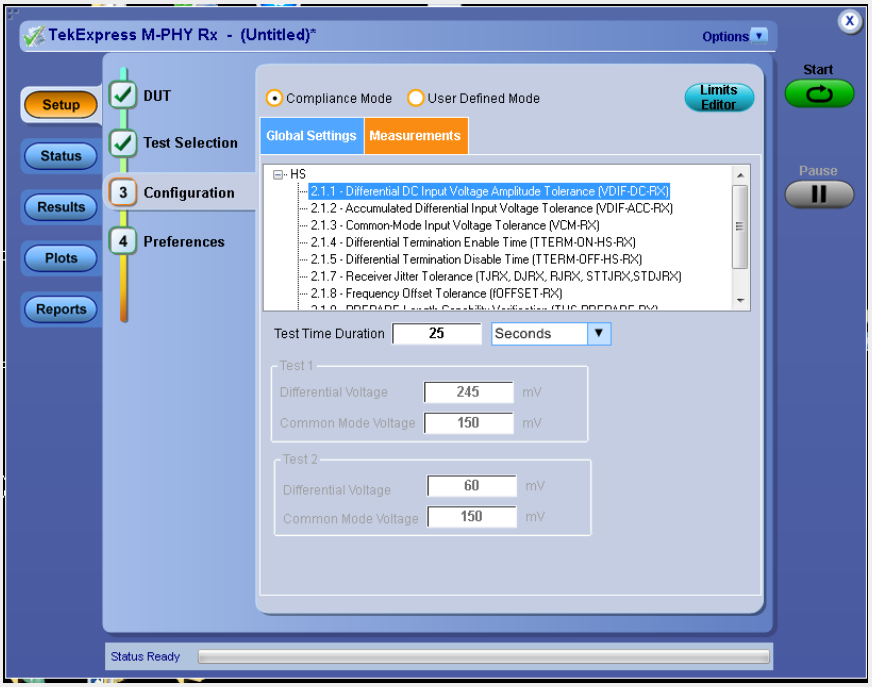


2. Test Selection and Configuration

Setup Panel Test Selection tab – HS Margin

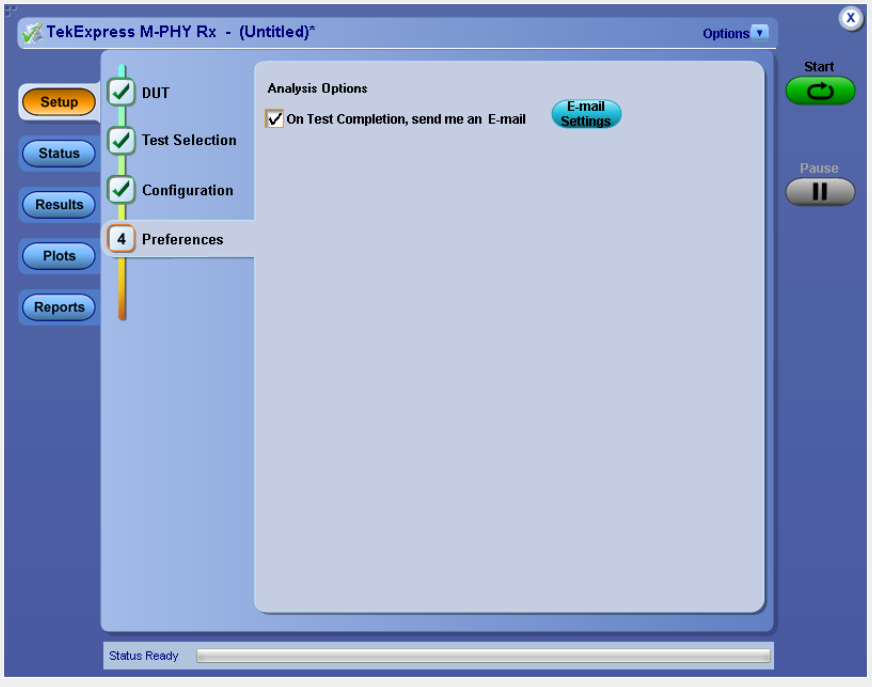


Setup Panel Configuration tab – HS 2.1.1



3. Preferences

Setup Panel Preferences tab

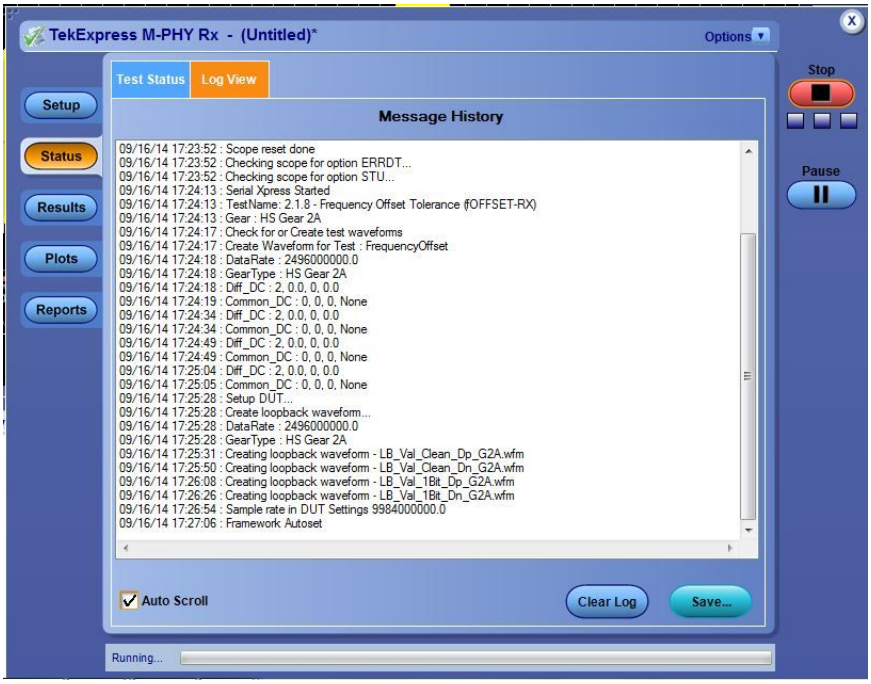


Options selected in a preceding tab affect options available in the next tab down. However, you can switch between the tabs in any order to modify your test parameters. For more information on using the Setup panel, see *About setting up tests*.

See also. *About panels*

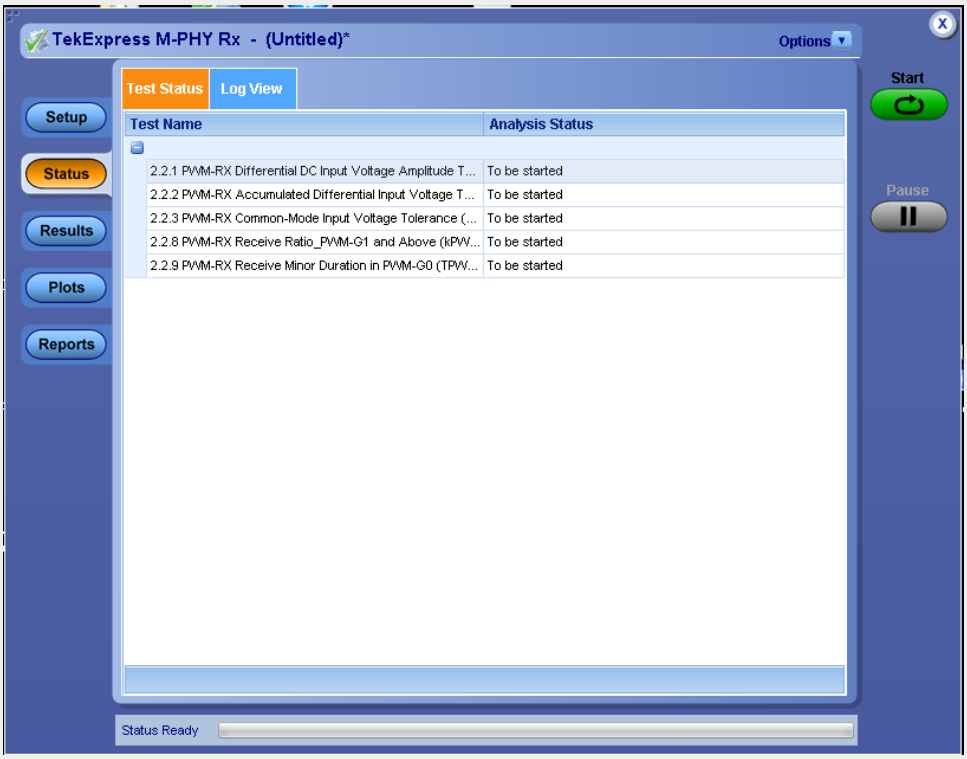
Status panel. The Status panel has two tabs: the Test Status tab displays the analysis status of the tests as they are being executed; the Log View tab (shown below) displays a view of the test log.

For more information on using the Status panel, see *View the progress of analysis*.



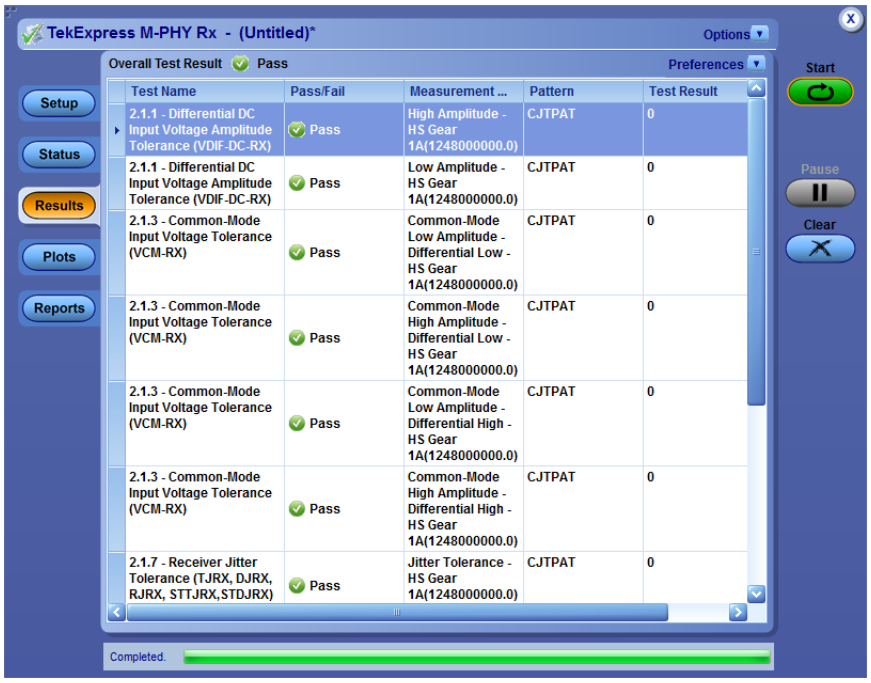
See also. *About panels*

Test Status view



Results panel. When a test finishes, the application switches to the Results panel to display a summary of test results. Set viewing preferences for this panel from the Preferences menu in the upper right corner. Viewing preferences include showing whether a test passed or failed, summary results or detailed results, and enabling wordwrap. For information on using this panel, see

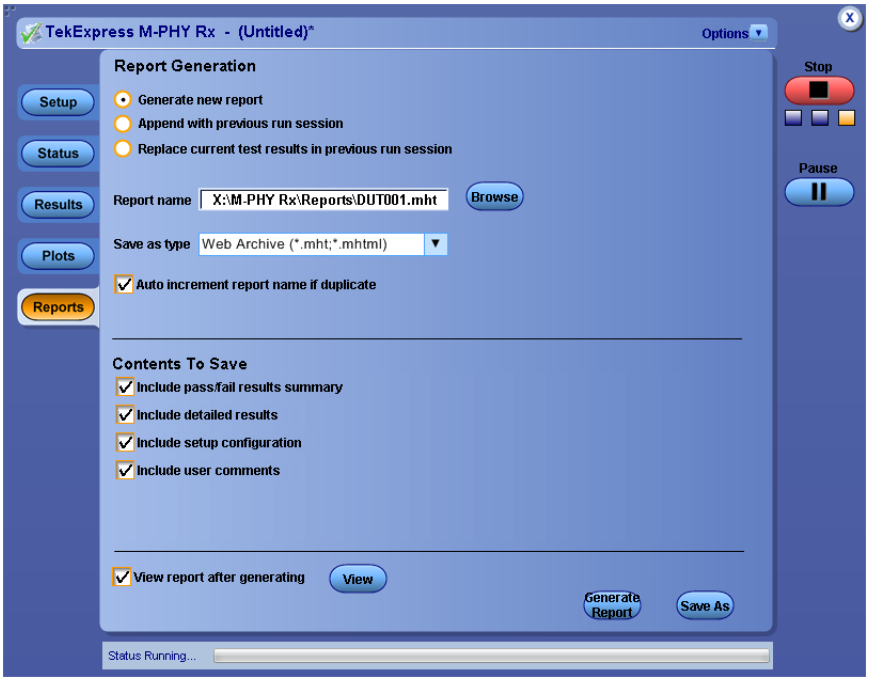
View test results.



See also. *About panels*

Reports panel. Use the Reports panel to browse for reports, name and save reports, select report content to include, and select report viewing options.

For information on setting up reports, see *Select report options*. For information on viewing reports, see *View the report*.



See Aaso

About panels

Saving and recalling test setups

About test setups

TekExpress M-PHY Rx opens with the default setup selected. Run a test before or after saving a setup. When you save a setup, the selected oscilloscope, general parameters, measurement limits, test, and other configuration settings are all saved under the setup name. You can run a saved test without having to do any other setup except ensuring that the oscilloscope is ready. Open the setup and click **Start**.

See also

About setting up tests

Save a test setup

Recall a saved test setup

Create a new test setup based on an existing one

Save a test setup

Save a test setup before or after running a test using the parameters you want saved. Create a new test setup from any open setup or from the default setup. When you select the default test setup, all parameters are returned to their defaults. The following instructions start from the default test setup:

1. From the Options menu, select **Default Test Setup**.
2. Select the desired options in the Setup panel.
3. Select the desired *report options*.
4. If desired, run the test to be sure that it captures the information you want. If it does not, edit the parameters.
5. From the Options menu, select **Save Test Setup**.

See also *About setting up tests*

About configuring tests

Test setup overview

Recall a saved test setup

These instructions are for recalling saved test setups:

1. From the Options menu, select **Open Test Setup**.
2. In the File Open dialog box, select the desired setup from the list and then click **Open**.

See also *About test setups*

Create a new test setup based on an existing one

Create a new test setup based on an existing one

Use this method to create a variation on a test setup without having to create the setup from the beginning.

1. From the Options menu, select **Open Test Setup**.
2. In the File Open dialog box, select the desired setup from the list and then click **Open**.
3. Modify the parameters as desired.
4. From the Options menu, select **Save Test Setup As**.
5. In the File Save As dialog box, enter a test setup name and then click **Save**.

See also *About test setups*

Test setup overview

Setting up and configuring tests

About setting up tests

Set up tests using the tabs in the Setup panel. Settings in the DUT tab use a top-down, left-to-right logic flow, so that any parameter that affects or acts as a filter for other parameters appears either to the top of or to the left of the affected parameters.

Tests are saved when you save a test setup. To avoid overwriting test results, remember to assign a unique name to the test either before running it or immediately after.

The application supports the following tests in Compliance Mode only:

- *2.1.1 HS-RX Differential DC input voltage amplitude tolerance (V_{DIF-RX})*
- *2.1.2 HS-RX Accumulated diff input voltage ($V_{DIF-ACC-HS-RX}$)*
- *2.1.3 HS-RX Common-mode input voltage tolerance (V_{CM-RX})*
- *2.1.4 HS-RX Differential termination enable time ($T_{TERM-ON-HS-RX}$)*
- *2.1.5 HS-RX Differential termination disable time ($T_{TERM-OFF-HS-RX}$)*
- *2.1.7 HS-RX Receiver jitter tolerance (TJ_{RX}, DJ_{RX}, RJ_{RX}, $STTJ_{RX}$, $STDJ_{RX}$)*
- *2.1.8 HS-RX Frequency offset tolerance ($f_{OFFSET-RX}$)*
- *2.1.9 HS-RX PREPARE length capability verification ($T_{HS-PREPARE-RX}$)*
- *2.1.10 HS-RX Sync length capability verification ($T_{SYNC-RX}$)*
- *HS Margin*
- *2.2.1 PWM-RX Differential DC input voltage amplitude tolerance ($V_{DIF-DC-PWM-RX}$)*
- *2.2.2 PWM-RX Accumulated differential input voltage tolerance ($V_{DIF-ACC-PWM-RX}$)*
- *2.2.3 PWM-RX Common-mode input voltage tolerance (V_{CM-RX})*
- *2.2.4 PWM-RX Differential termination enable time ($T_{TERM-ON-PWM-RX}$)*
- *2.2.5 PWM-RX Differential termination disable time ($T_{TERM-OFF-PWM-RX}$)*
- *2.2.8 PWM-RX Receive ratio PWMG1 and above (k_{PWM-RX})*
- *2.2.9 PWM-RX Receive minor duration in PWMG0 ($T_{PWM-MINOR-G0–RX}$)*

See also

About test setups

Before You click start

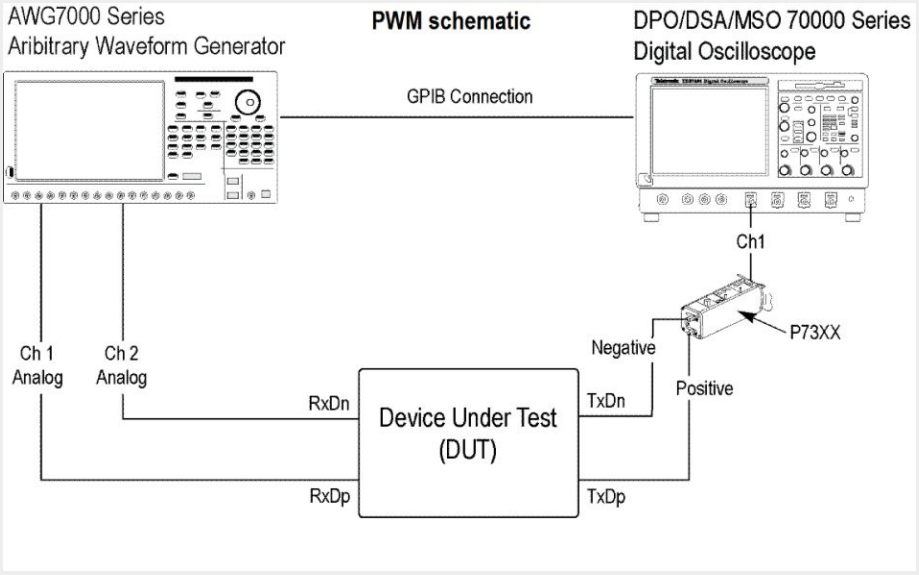
About running tests

Equipment connection setup

To run tests, you need the following equipment (for details, see *Minimum system requirements*):

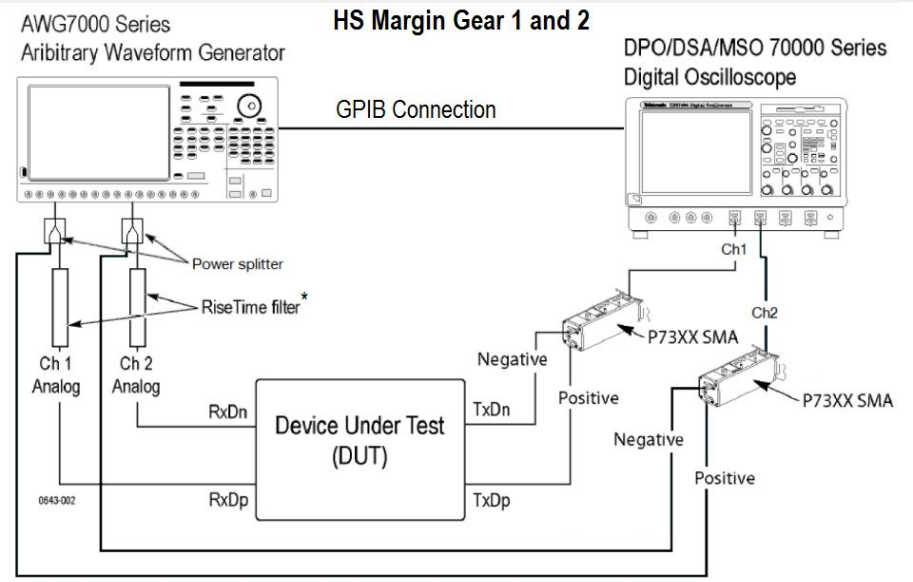
- A *supported Tektronix oscilloscope*
- Arbitrary Waveform Generator (AWG)
- Differential probe
- Device under test
- Filters

Connection diagrams



NOTE. *The above setup is applicable only for 2.2.4 PWM-RX differential termination enable time ($T_{TERM-ON-PWM-RX}$) and 2.2.5 PWM-RX differential termination disable time ($T_{TERM-OFF-PWM-RX}$). For all other PWM tests, probe connection is not required.*

Figure 1: PWM schematics



* Gear 1 should be connected with 240ps Rise time filters, and Gear 2 should be connected with 120 ps Rise time filters.

Figure 2: HS Margin schematic for Gear 1 and Gear 2

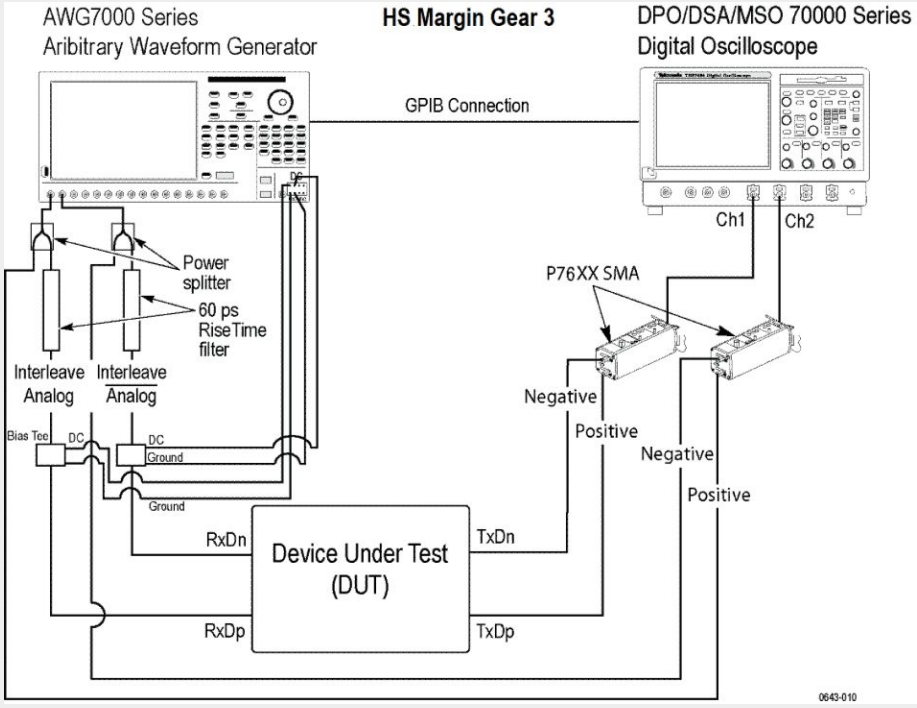
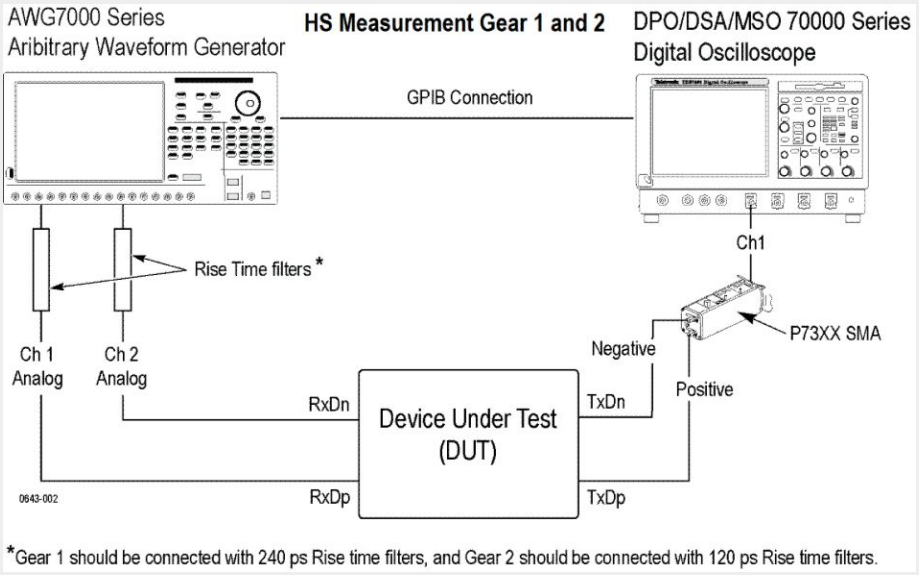
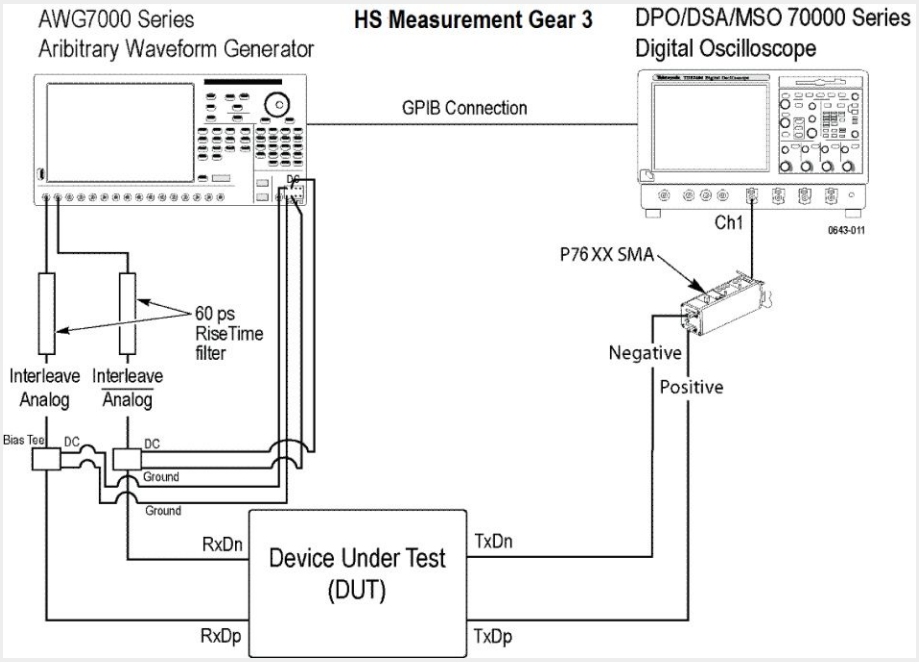


Figure 3: HS Margin schematic for Gear 3



NOTE. *The above setup is applicable only for continuous mode tests. For Burst mode tests, probe connection is not required. Users are expected to use IBER for counting errors.*

Figure 4: HS Measurement Gear 1 and Gear 2



NOTE. *The above setup is applicable only for continuous mode tests. For Burst mode, probe connection is not required. Users are expected to use IBER for counting errors.*

Figure 5: HS Measurement Gear 3

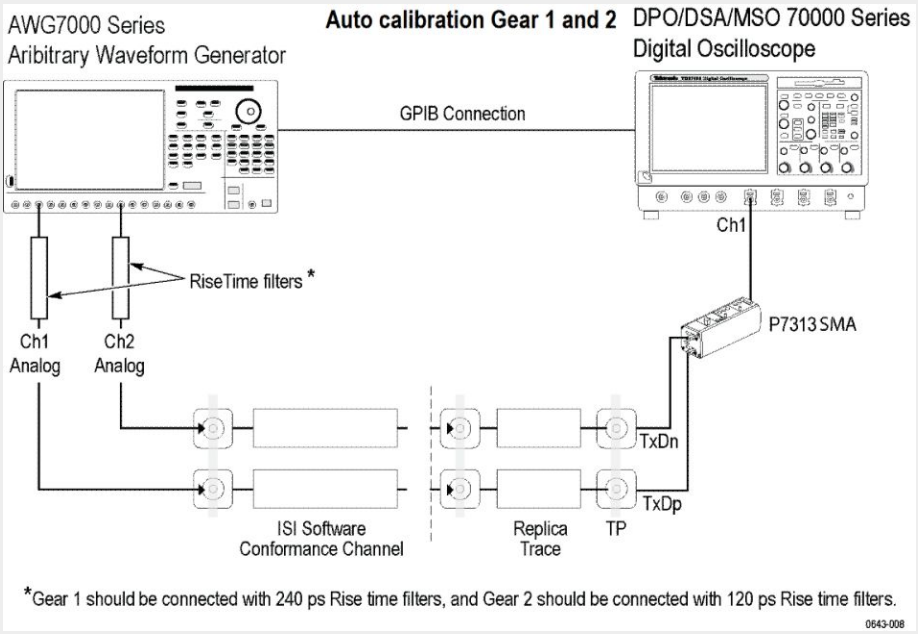


Figure 6: Auto calibration Gear 1 and Gear 2 schematic

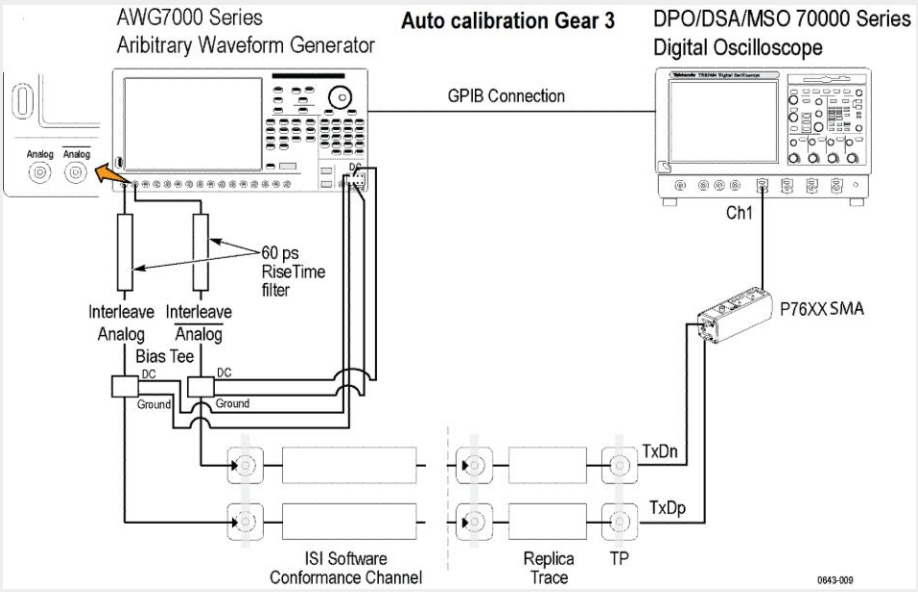


Figure 7: Auto calibration Gear 3 schematic

See also

View connected instruments

About setting up tests

Receiver testing measurement procedure

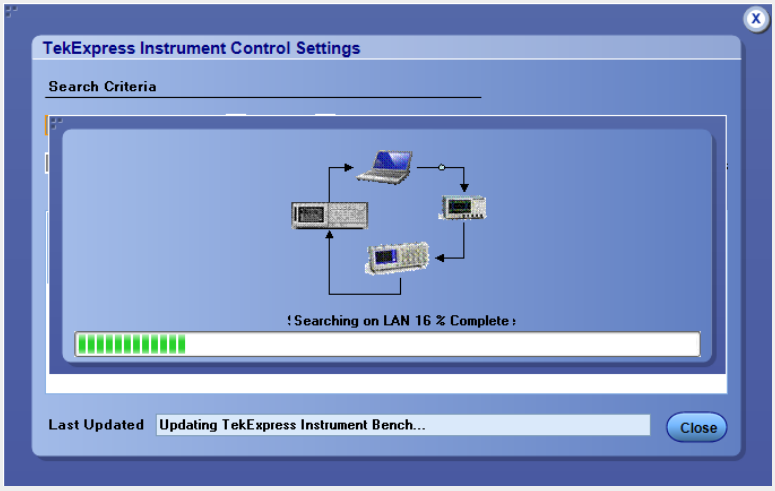
View connected instruments

Use the Instrument Control Settings dialog box to view or search for connected instruments required for the tests. The application uses TekVISA to discover the connected instruments.

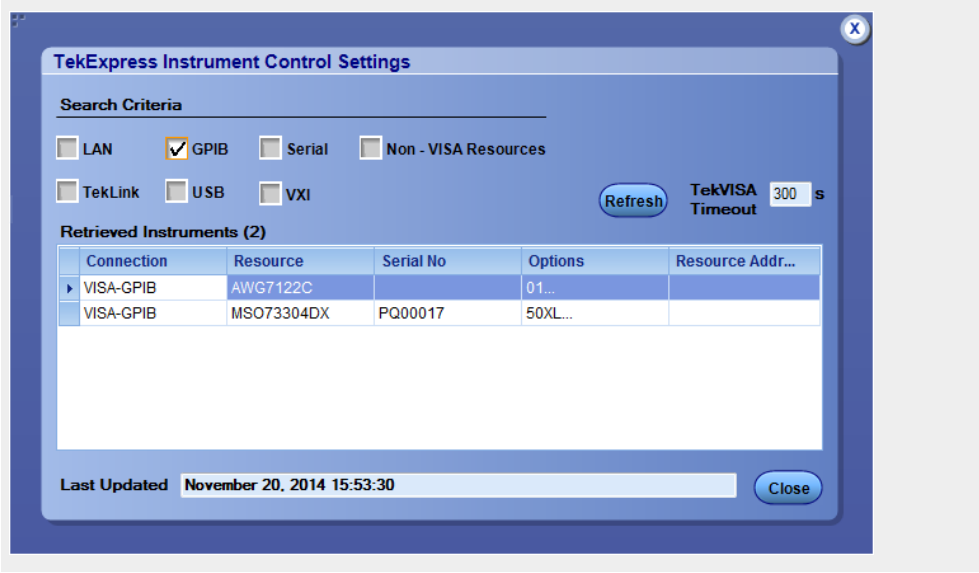
To refresh the list of connected instruments:

1. From the Options menu, select **Instrument Control Settings**.
2. In the Search Criteria section of the Instrument Control Settings dialog box, select the connection types of the instruments to search for.

   Instrument search is based on the VISA layer but different connected cables determine the resource type, such as GPIB. If the search does not find any instruments that match a selected resource type, a message appears telling you that no such instruments were found.
3. Click **Refresh**. TekExpress searches for connected instruments.



4. After discovery, the dialog box lists the instrument-related details based on the search criteria you selected. For example, if you selected GPIB as the search criteria, the application checks for the availability of instruments over GPIB.



The details of the instruments are displayed in the Retrieved Instruments table. The time and date of instrument refresh is displayed in the Last Updated field.

See also

Configure tests

Equipment setup

Test setup overview

Test setup includes configuration parameters and report options. Use the options in the *Setup panel* and *Reports panel* to select and configure tests.

1. *Select the device parameters*.
2. *Select one or more tests*.
3. *Configure tests*.
4. *Select test notification preferences*.
5. *Select report options*.

See also

About test setups

Pre-run checklist

Before you click start

About running tests

Select device parameters

Use the Setup panel DUT tab settings to select parameters for the device under test. Options selected here are global settings that apply to all tests for the current session. Options here also affect the list of available tests in the Test Selection tab.

1. In the Setup panel, click the **DUT** tab.
2. If needed, in the DUT ID field, enter the ID for the device. The default value is DUT001. The name you enter here appears on reports.
3. (Optional) To add comments that will appear at the top of the test report for the selected DUT, click the note pad icon ( ) to the right of the DUT ID field and specify a comment up to 256 characters. (To enable or disable comments appearing on the test report, see *Select report options*.)
4. Select from the following options if available. (Settings that do not apply to compliance testing cannot be changed and are grayed out.)

Table 6: DUT settings

Setting	Description
Use Pre-Defined Pattern	Uses the built-in pre-defined pattern associated with the selected test. This cannot be changed in Compliance Mode.
Version	Select the M-PHY Rx specification version. The latest version is the default.
Error Detection Using	■ Scope: TekExpress uses the oscilloscope for error testing. This cannot be changed in Compliance Mode. ■ Manual: User enters the error count using a dialog box. ■ Custom Script: User can use a custom script to read error. ***NOTE.*** *A custom script should read the error and write to a text file parallel to the script file with same file name. For example, if you have ABC.exe in the directory C:\MIPI\, after reading the error from the DUT, ABC.exe should write the error value in a text file named ABC.txt in the same C:\MIPI\ directory.*

Setting	Description
LoopBack Initialization	■ This is set up manually. The application prompts you to set this value before executing a test. ■ Custom Script: User can put the DUT into loopback using a custom script. The application will run the custom script when it needs to put the DUT into loopback mode.
Perform LoopBack Initialization	Determines how frequently loopback initialization occurs. ■ Always: Do loopback initialization before starting every test. ■ Only Once: Do loopback initialization only once before starting the first measurement in the current test session.
Perform LoopBack Validation	Determines when loopback validation occurs. This process validates the DUT. ■ Only Once: Do loopback validation only once before starting the first measurement of the current session. ■ Always: Do loopback validation at the beginning of every test. ***NOTE.*** *This may require you to be available to put the DUT in loopback mode several times during testing, depending on the number of tests and the test options selected.* ■ Never: Do not do loopback validation.
Signal Type	■ HS: High Speed ■ PWM: Pulse Width Modulation and HS-Margin
Gear Selection	Select the gear and then select the variation of the gear that is supported by the DUT.

See also *About setting up tests*

Loopback validation

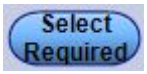
Select a test

Use these instructions to select the tests to run on the connected DUT.

1. In the Setup panel, click the **Test Selection** tab.
2. All tests are selected by default. Click **Deselect All** if you do not want to run all tests.
3. Select the desired test(s):
   - To select all tests in the list, click the **Select All** button.
   - To select only the pre-defined tests, click **Select Required** button
   - To select individual tests, select only the check boxes for the desired tests.

Test selection controls

Table 7: Setup panel test controls

Button	Description
	Displays the schematic document for the selected test. Use to verify the test setup before running the test
	Deselects all tests in the table
	Selects all tests in the table
	Selects only pre-defined tests from Test Selection panel

See also *About setting up tests*

Configuring tests

About configuring tests

Use configuration settings to view the measurement parameters for selected tests. Some settings can be changed.

Configuration settings consist of Global parameters and Measurement parameters. Global parameters are common for all tests. Measurement parameters are specific to the test selected in the tree view section of the configuration settings.

See also. *Configure tests*

About running tests

Configure tests

Use these instructions for configuring tests. To return to test selection from the Configuration section, click the **Test Selection** button.

NOTE. *You cannot change test parameters that are grayed out.*

1. Modify *Global settings* as desired:
   - To select the instruments for testing, click **Global Settings**. In the Instruments Detected section, click in the shaded areas to activate the drop-down lists and then select the desired instruments for Signal Generator and Error Detector. If you do not see the desired instrument in the list, *refresh the list*.
   - To change the Trigger source for the current test session, under Scope Settings, select the desired channel from the drop-down list.
2. To modify any individual test measurement settings, click **Measurements**, select the test in the tree view and change the settings.

See also. *About setting up tests*

Common test parameters

The following table lists the settings and parameters common to all tests.

Table 8: Common parameters and values

Parameter type	Parameter
Mode	■ Compliance Mode is selected by default and cannot be deselected. Test parameters that fall outside of compliance values are disabled. ■ User Defined Mode is selected, the test parameters are editable.
Global Settings	These settings apply to all tests selected for the current session. You can change only some of these settings. ■ Signal Generator: Specifies the instrument to use for generating the signals to test. This can be changed. ■ Error Detector: Specifies the instrument to use for detecting errors in the signal. This can be changed. ■ Scope Settings: Specifies the instrument channel to use as the source for the trigger. This can be changed. ■ Signal Source Settings: ■ Test Pattern: CJTPAT ■ Channel Amplitude: The value of the amplitude to be set for the selected channel.
Measurements	These settings apply to the test selected in the tree view of the configuration section.

See also. *Configure tests*

Configure email notification

To be notified by email when a test completes, fails, or produces an error, configure the email settings.

1. From the Options menu, select **Email Settings** to open the Email settings dialog box.
2. (Required) For Recipient email Address(es), enter your email address. To include multiple addresses, separate the addresses with commas.
3. (Required) For Sender's Address, enter the email address used by the instrument. This address consists of the instrument name, followed by an underscore, followed by the instrument serial number, the @ symbol and then the email server used. For example: DPO72004C_B130099@yourcompany.com.
4. (Required) In the Server Configuration section, type the SMTP Server address of the Mail server configured at the client location, and the SMTP Port number, in the corresponding fields.

   If this server requires password authentication, enter a valid login name, password, and host name in the corresponding fields.

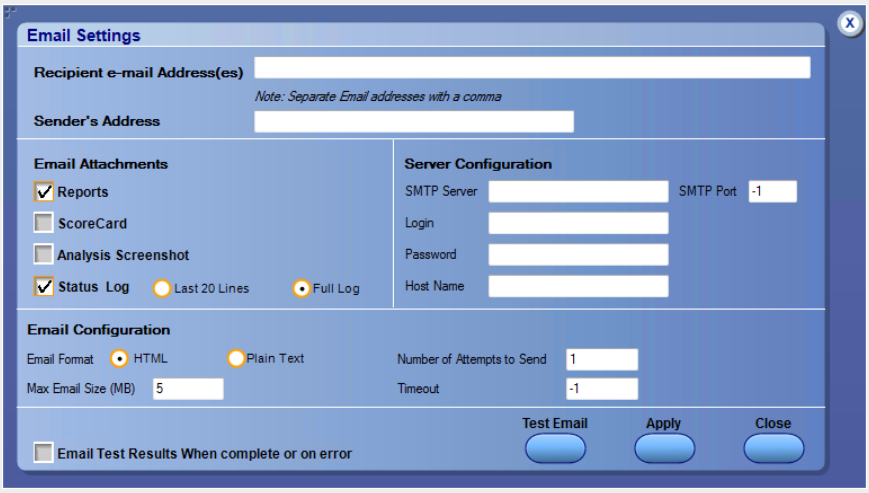
   NOTE. *If any of the above required fields are left blank, the settings will not be saved and email notifications will not be sent.*

5. In the Email Attachments section, select from the following options:
   - **Reports**: Select to receive the test report with the notification email.
   - **Status Log**: Select to receive the test status log with the notification email. If you select this option, then also select whether you want to receive the full log or just the last 20 lines.

     NOTE. *The ScoreCard and Analysis Screenshot options are not available in M-PHY Rx.*

6. In the Email Configuration section, select as desired:
   - Select the message file format to send: HTML (the default) or plain text.
   - Enter a maximum file size for the email message. Messages with attachments larger than this limit will not be sent. The default is 5 MB.
   - To limit the number of attempts that the system makes to send a notification, enter the number in the Number of Attempts to Send field. The default is 1. You can also specify a timeout.
7. Select the **Email Test Results When complete or on error** check box. Use this check box to quickly enable or disable email notifications.
8. To test your email settings, click **Test Email**.
9. To apply your settings, click **Apply**.
10. Click **Close** when finished.

Email Settings



Select test notification preferences

Select how you want to be notified by the system when a test measurement fails:

1. In the Setup panel, click the **Preferences** tab.
2. To receive an email when testing completes, select the **On Test Completion, send me an email** check box.

   The application will send you an email when the current test completes. If you select this option, select the **Email Test Results when complete or on error** check box in the Email Settings. Click **Email Settings** to *configure the email settings*.

See also *About setting up tests*

Select report options

Select report options

Use the *Reports panel* to select which test information to include in the report, and the naming conventions to use for the report. For example, always give the report a unique name or select to have the same name increment each time you run a particular test. Generally, you would select report options before running a test or when creating and saving test setups. Report settings are included in saved test setups.

In the Reports panel, select from the following options:

Table 9: Report options

Setting	Description
Report Name	Displays the name and location where the report will be saved when generated. The report and related files are stored in the Untitled Session folder located by default at \My TekExpress\ M-PHY Rx \Untitled Session. The report file in this folder gets overwritten each time you run a test unless you specify a unique name or select to auto increment the report name. Change the report name or location. Do one of the following: ■ In the Report Name field, type over the current folder path and name. ■ Double-click in the Report Name field. Make selections from the popup keyboard and then click the **Enter** button. Be sure to include the entire folder path, the file name, and the file type. For example: C:\Documents and Settings\your user name\My Documents \My TekExpress\M-PHY Rx\DUT001_Test_211.mht. ***NOTE.*** *You cannot change the file location using the Browse button.* Open an existing report. Click **Browse**, locate and select the report file and then click **View** at the bottom of the panel.
Save As Type	Saves a report in a file type different from the default. Lists supported file types to choose from. ***NOTE.*** *If you select a file type different from the default, be sure to change the report file name extension in the Report Name field to match.*
Auto increment report name if duplicate	If the application finds a report with the same name as the one being generated, the application automatically increments the name of the report. For example: DUT001, DUT002, DUT003. This option is enabled by default.
Include Pass/Fail Results Summary	The color block labeled Test Result indicating whether the test passed or failed will be included in the report. For details, see Report Contents in *View a report*.
Include Detailed Results	The report will include parameter limits, execution time, and any test-specific comments generated during the test.

Setting	Description
Include Setup Configuration	Information about hardware and software used in the test will be included in the summary box at the top of the report. Information includes: the oscilloscope model and serial number, probe model and serial number, AWG model and serial number, firmware version for the oscilloscope and AWG, SPC and factory calibration status, and software versions for applications used in the measurements.
Include Complete Application Configuration	A table listing general, common, and acquired parameters used in the test will be included in the report. This option is disabled by default.
Include User Comments	Select to include any comments about the test that you or another user added in the DUT tab of the Setup panel. Comments appear in the Comments section under the summary box at the beginning of each report.
View Report After Generating	Automatically opens the report in your Web browser when the test completes. This option is selected by default.

See also

View a report

About setting up tests

Auto calibration

Auto calibration

The auto calibration function addresses receiver calibration requirements for the M PHY standard. Auto calibration compensates the patterns for specific jitter parameters (ISI, Random Jitter - RJ, Sinusoidal Jitter - SJ and Voltage).

The procedure sequences through all the patterns, with each pattern calibrated independently. These values are used for the jitter-controlled generation of patterns which will be injected into the DUT during loopback.

The design engineer needs to ensure that the amount of jitter components (also known as target impairments) meets the compliance test specification.

To perform the calibration, the application varies the target parameter through a predefined range and a uniform step size and computes the transfer function between the measured and input values.

The signal generation for calibration is done using an AWG. The measurement methodology is DPOJET as per the CTS.

The calibration results can be viewed at any time as values or graphical plots. The application uses an appropriate polynomial fit algorithm for all the target values which gives the characteristic curve. The respective calibrated values are derived from the characteristic curve.

The orange square icons on the calibration graph plot represent the measured jitter values.

See also. *Auto calibration setup diagrams*

Auto calibration wizard overview

Auto calibration wizard overview

The Auto calibration wizard lets you:

- Auto-calibrate the M PHY RX signal with the help of an AWG and oscilloscope running TekExpress
- Specify the calibration constants, standards, and custom parameters
- Perform auto calibration for Inter Symbol Interference (ISI), Random Jitter (RJ), Periodic Jitter (PJ) or Sinuosoidal jitter (SJ) and voltage
- Generate and view the calibration report

Table 10: Calibration table parameters

Parameter	Description
ISI	For the 2.1.2 HS-RX Accumulated Differential Input Voltage Tolerance test, the test signal is expected to have 0.2 UI of ISI. This is realized in most part by the traces to the Rx pins. If additional software ISI is needed, this parameter can be used.
RJ	The random jitter component is calibrated using this parameter. Two RJ components are indirectly specified: Random Jitter and short term RJ.
SJ	The sinusoidal jitter component is calibrated using this parameter. Deterministic Jitter (DJ_{RX}) and Short Term Total Jitter ($STTJ_{RX}$) are calibrated using SJ at various frequencies.
Voltage	This entry represents the parameter Accumulated Differential Input Voltage Tolerance ($V_{DIF-ACC-HS-RX}$).

Filter. This setting specifies the frequency in MHz which is applicable to the following parameters:

- Periodic Jitter: Specifies the jitter frequency.
- Random Jitter: Specifies the High frequency limit of the RJ components; this is useful in specifying STRJ.

Pattern. This setting specifies the test pattern (CJTPAT), to be used for calibrating this parameter.

Jitter specification. The HS-RX jitter specification is shown in the following table.

Table 11: Jitter specifications

Parameter	Max value
$STRJ_{RX}$	0.1 UI
RJ_{RX}	0.17 UI
$STDJ_{RX}$	0.20 UI
DJ_{RX}	0.35 UI
$V_{DIF-ACC-HS-RX}$	40 mV

See also. *Perform ISI auto calibration*

Perform RJ auto calibration

Perform SJ auto calibration

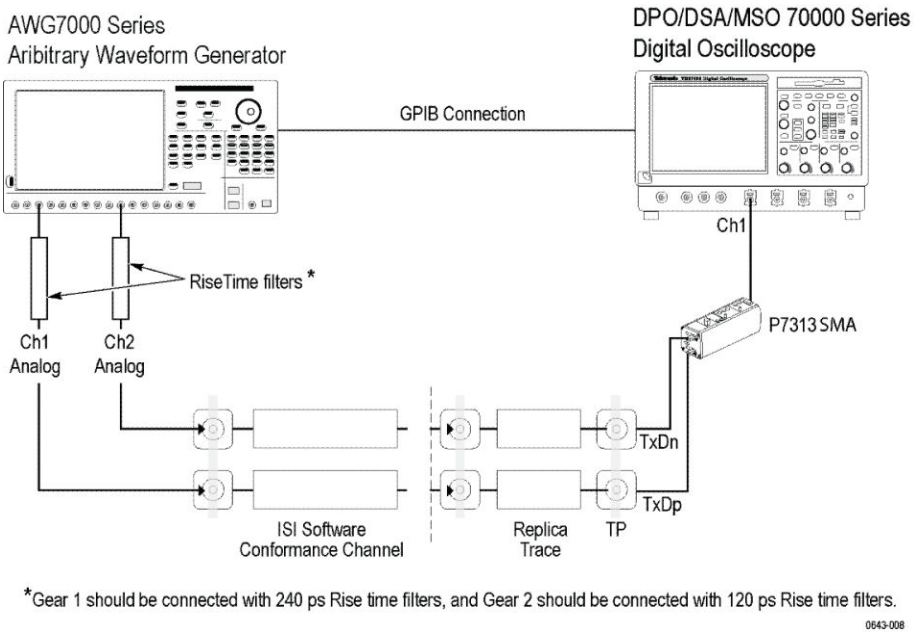
Perform voltage auto calibration

View auto calibration results

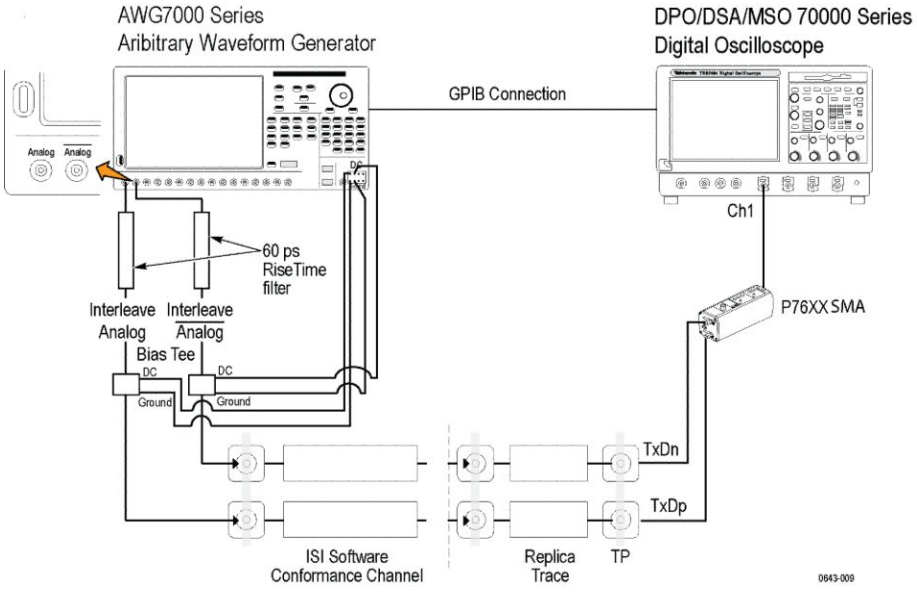
View auto calibration parameter details

Auto calibration setup diagrams

Auto calibration setup diagram for Gear 1 and Gear 2

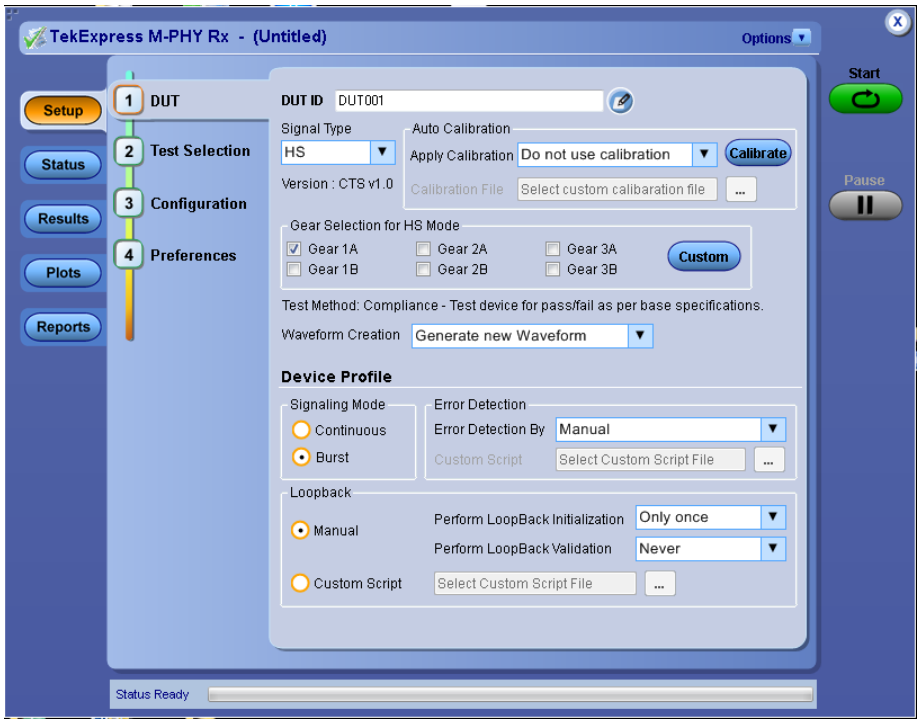


Auto calibration setup diagram for Gear 3



Start the auto calibration wizard

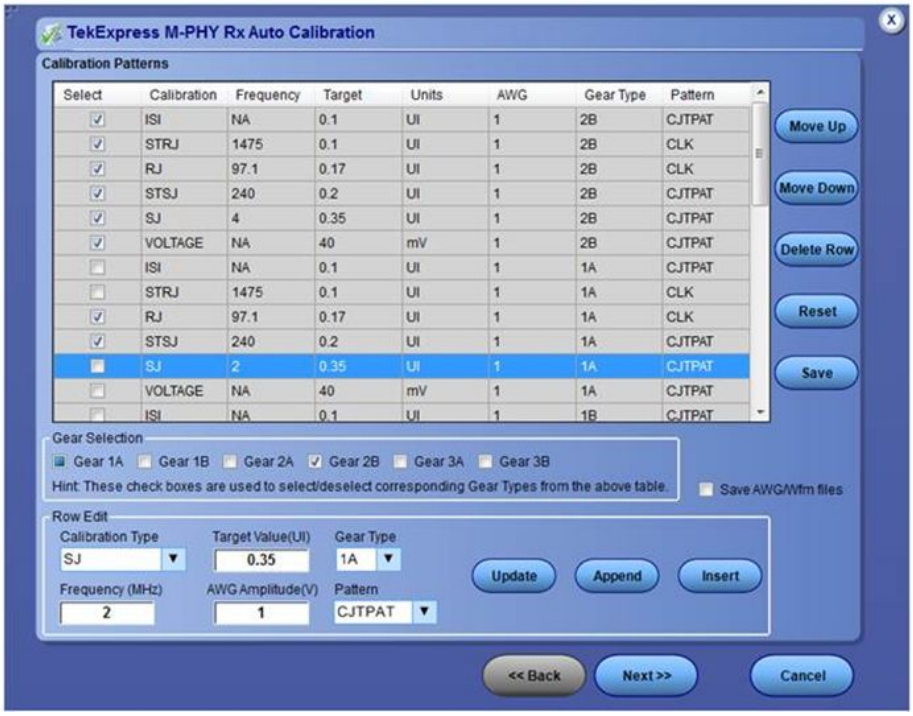
Click the Calibration button (in the Setup tab of the main MPHY Rx application) to open the Auto Calibration wizard.



Click **Calibrate** button to *Set auto calibration pattern parameters*.

Set auto calibration pattern parameters

The Calibration Patterns screen lets you to select the calibration parameters and the corresponding gear types from the table below.



1. Select or deselect any desired calibration parameters to enable or disable it.
2. Select or deselect the gear in the **Gear Selection** group, which will automatically select or deselect all the calibration parameters corresponding to it.
3. Select **AWG/Wfm** files to save the AWG and waveform files at the end of the calibration.
4. Click **Save** to save the calibration type and parameters.

To create custom calibration, set the variables in the Row Edit group:

- Select the Calibration Type and Pattern from the respective fields.

  NOTE. *'n' number of SJ's can be added, and the calibrated SJ values are applied in 2.1.7 (HS-RX receiver jitter tolerance (TJ_{RX}, DJ_{RX}, RJ_{RX}, $STTJ_{RX}$, $STDJ_{RX}$)*

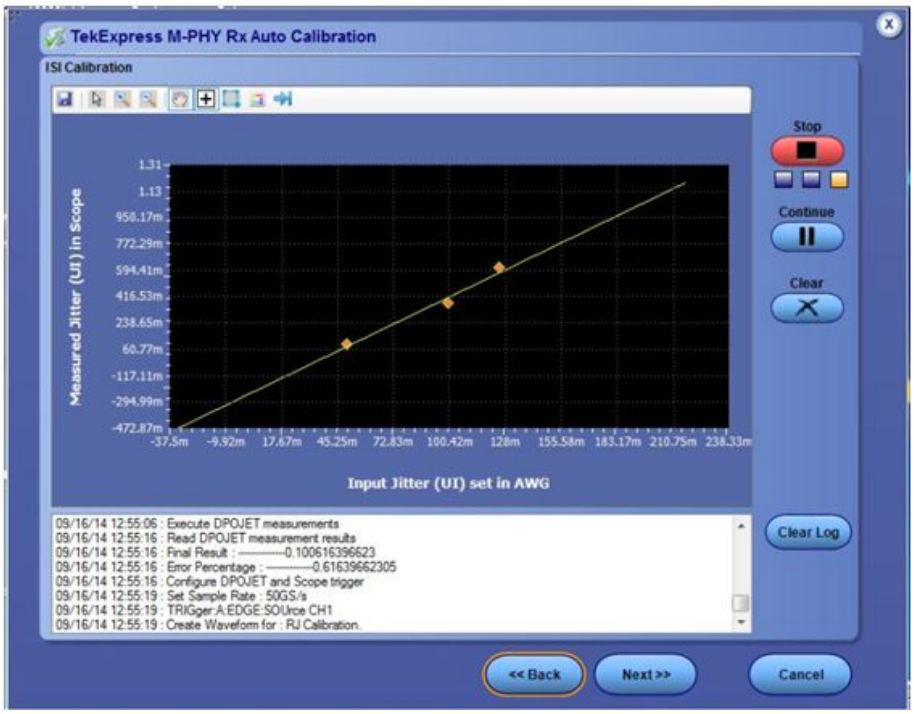
- Enter the Target Value, Frequency, and AWG Amplitude in their respective fields.
- Select the gear in the **Gear Type** drop down.
- Click **Update** after updating the customized row in the calibration table.
- Click **Append** to add the default or customized SJ values at the end of the table.

- Click **Insert** to insert the default SJ value or customized value into the calibration table.
- Click **Next** to display the *ISI Calibration screen*.

Perform ISI auto calibration

Use the ISI calibration screen to perform the ISI calibration. The ISI calibration screen plots the measured versus input jitter (ISI) values.

1. Click **Start** button to run the ISI calibration. The process runs until the calibration is completed.



2. Click **Stop** to stop the calibration process.
3. Click **Continue** and **Pause** button to continue and pause the calibration process
4. Click **Clear** to clear the generated, plot.
5. Click **Clear Log** to clear the details of the user log.

NOTE. *If the plots have high error percentage, it dynamically changes the points close to the target value.*

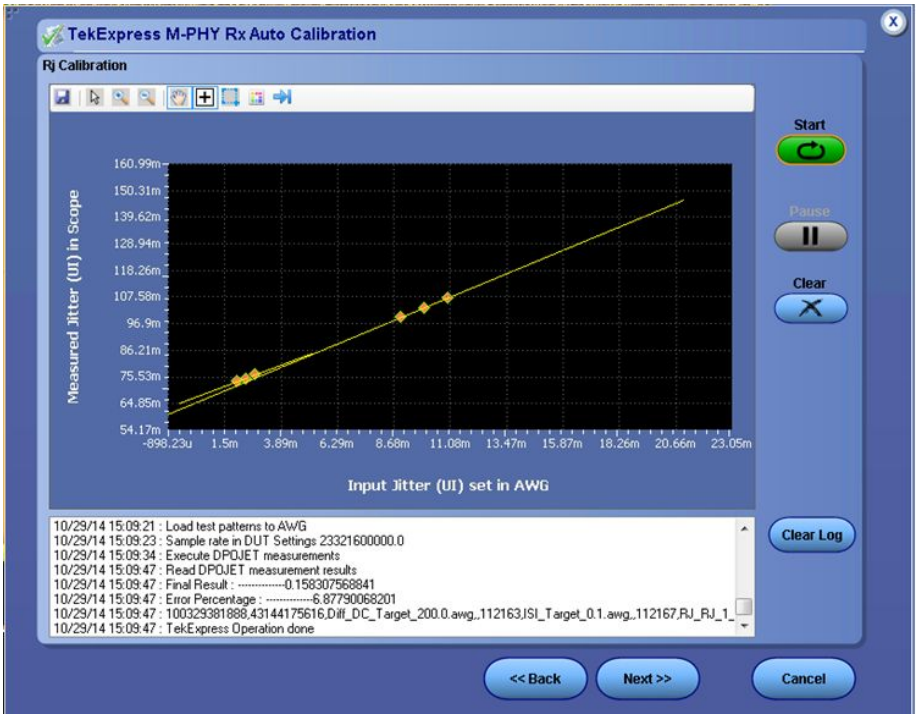
---

6. Click **Back** to return to the Calibration pattern screen.
7. Click **Cancel** to cancel the ISI calibration.
8. Click **Next** to display the *RJ Calibration screen*.

See also. *View auto calibration parameter details*

Perform RJ auto calibration

The RJ calibration screen plots the input random jitter (RJ) versus measured values.



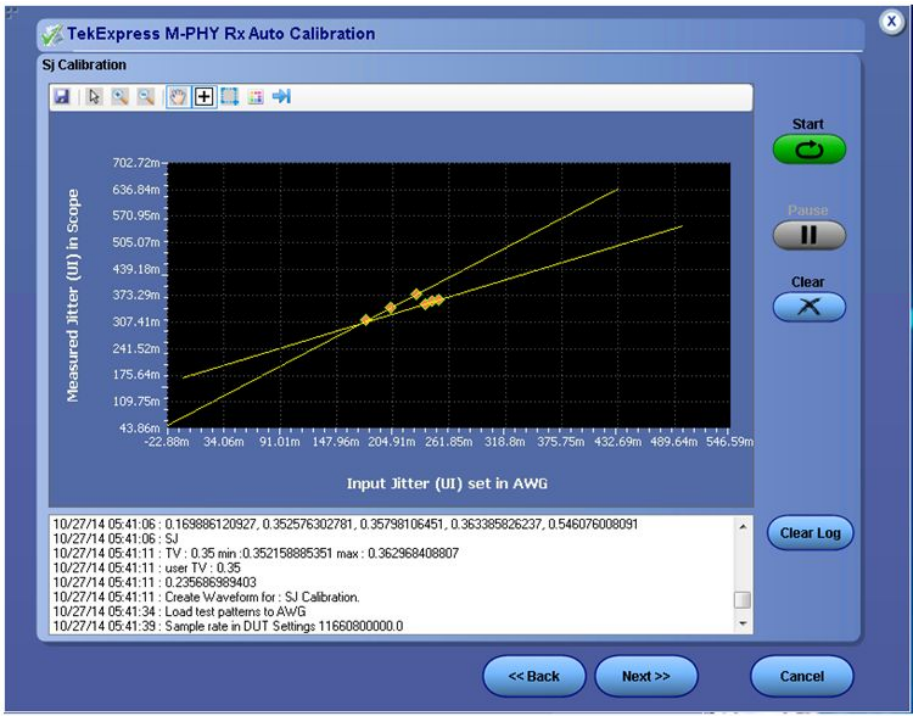
The operation and plot are similar to the ISI calibration screen. The two plots are for STRJ jitter and RJ jitter.

Click **Next** to display the *SJ Calibration screen*.

See also. *View auto calibration parameter details*

Perform SJ auto calibration

The SJ Calibration screen plots the sinusoidal jitter (SJ) versus measured values.



The operation and plot are similar to the ISI calibration screen. The two plots displayed are the STSJ jitter and SJ jitter.

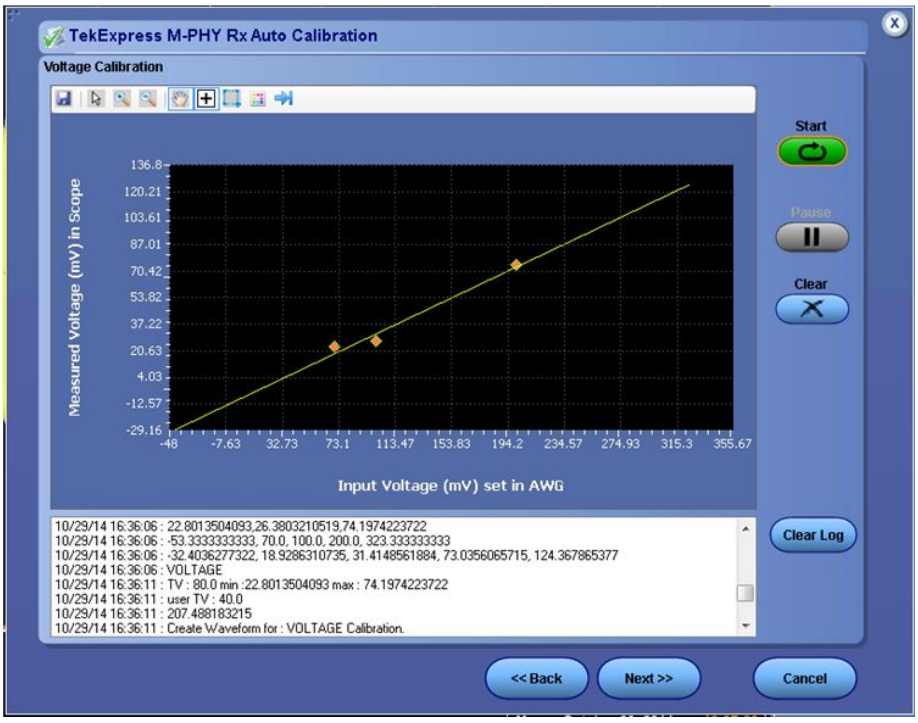
NOTE. *Recommended to run SJ calibration, as per the frequency and amplitude used in HS-RX receiver jitter tolerance (TJRX, DJRX, RJRX, STTJRX, STDJRX) test. e.g., For Gear 2B the frequency is 0.4 MHz, 4 MHz, 10 MHZ and 97.1 MHz, and the target value is 0.35 UI.*

Click Next to display the *Voltage Calibration screen*.

See also. *View auto calibration parameter details*

Perform voltage auto calibration

The Voltage calibration screen plots the measured versus input voltage values.



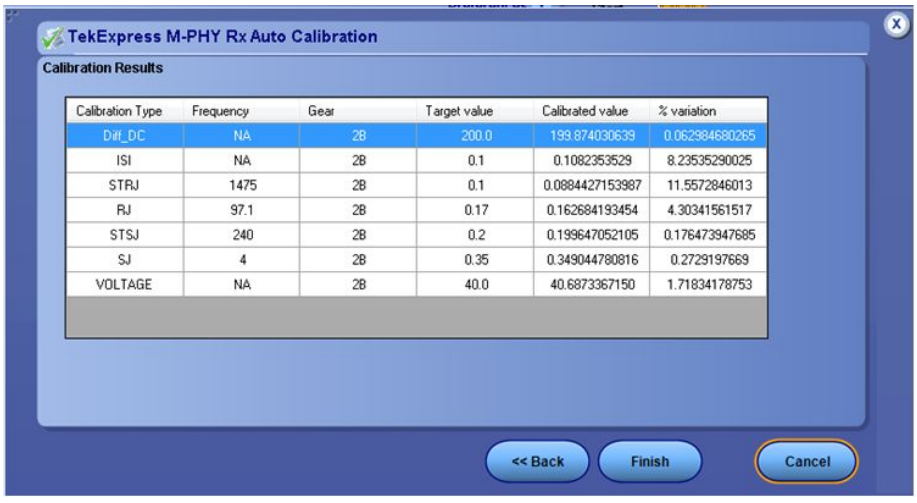
All the calibrated values (ISI, RJ and SJ) are applied during Voltage calibration to measure the eye height.

Click **Next** to display the *Calibration Results screen*.

See also. *View auto calibration parameter details*

View auto calibration results

The Calibration Results panel shows the result of each calibration step, along with the percentage variation with respect to the expected target value.



Click **Finish** to save the session to the specified file and exit the Auto calibration wizard.

View auto calibration parameter details

You can view the parameters of the generated signal in SerialXpress.

- The settings for base pattern file and the data rate for the selected gear in the Base Pattern tab
- The ISI calibration parameters in Channel/Cable tab
- The RJ and SJ calibration parameters in Transmitter tab

NOTE. *The setting of Base pattern is the same for all the parameters (ISI, RJ, SJ and voltage). By default Gear 2B values are displayed.*

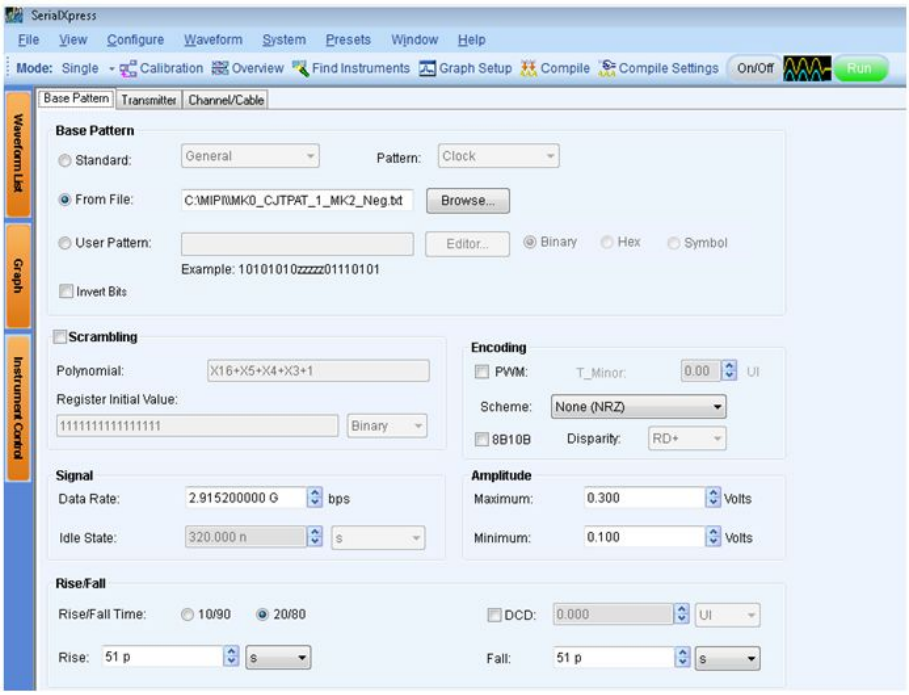


Figure 8: Base Pattern setting

NOTE. *During SJ calibration, use STSJ and SJ values. During RJ calibration, use STRJ and RJ values.*

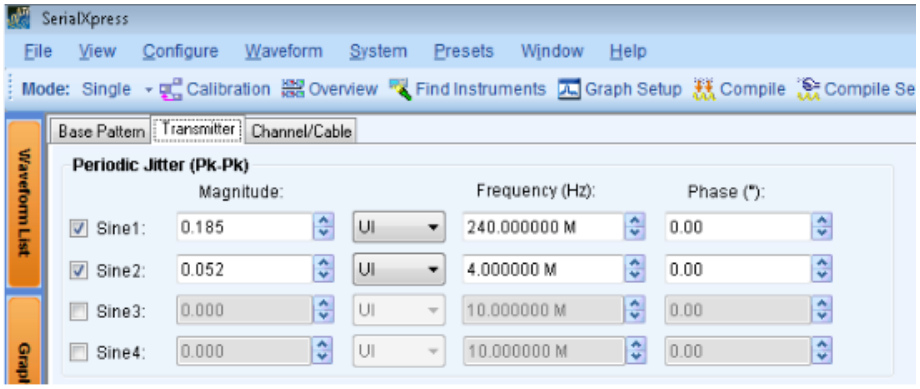


Figure 9: SJ calibration

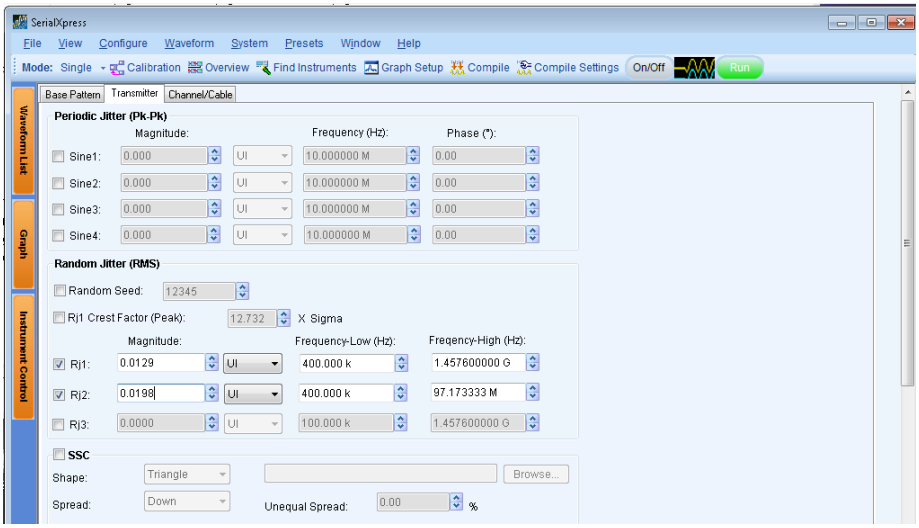


Figure 10: RJ calibration

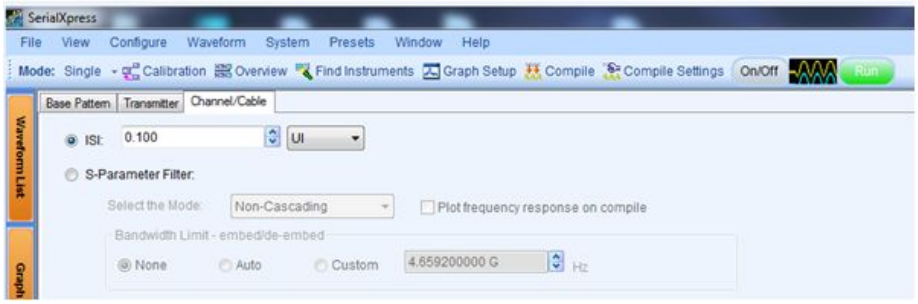


Figure 11: ISI calibration

NOTE.

The ISI, RJ and SJ calibrated values are applied in 2.1.7 (HS-RX receiver jitter tolerance (TJ_{RX}, DJ_{RX}, RJ_{RX}, $STTJ_{RX}$, $STDJ_{RX}$)).

View the measured jitter for the ISI, RJ and SJ calibration parameters in DPOJET.

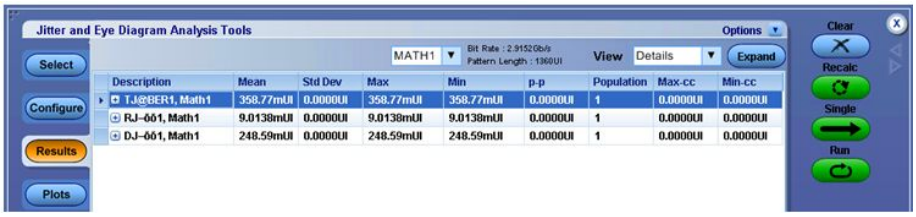


Figure 12: Jitter for ISI, RJ and SJ

Running tests and viewing results

About running tests

After selecting and configuring the test, *review the pre-run checklist* and then click **Start** to run the tests. While tests are running, you cannot access the Setup or Reports panels. To monitor the test progress, switch back and forth between the Status panel and the Results panel.

The application displays a report when the tests are complete. While the tests are running, other applications may display windows in the background. The TekScope application takes precedence over other applications, but you can switch to other applications by using the **Alt** + **Tab** key combination. To keep the TekExpress M-PHY Rx application on top, select **Keep On Top** from the M-PHY Rx Options menu.

See also

About configuring tests

About setting up tests

Before you click start

Before you click start

Before you run tests for the first time, do the following:

1. Understand where your test files are stored on the instrument.

After you install and launch TekExpress M-PHY Rx, it creates the following folders on the oscilloscope:

CAUTION.

Do not modify any of the session files or folders because this may result in loss of data or corrupted session files. Each session has multiple files associated with it. When you save a session, a .TekX file, and a folder named for the session that contains associated files, is created on the oscilloscope X: drive.

- \Program Files\Tektronix\TekExpress**TekExpress M-PHY Rx**

  NOTE. *This application will work only on machines with the Windows 7 operating system.*

- \My Documents\My TekExpress**M-PHY Rx**
- \My Documents\My TekExpress\M-PHY Rx**Untitled Session**

Every time you launch TekExpress M-PHY Rx, an Untitled Session folder is created in the M-PHY Rx folder. The Untitled Session folder is automatically deleted when you exit the M-PHY Rx application.

2. *Map the shared My TekExpress folder* as X: (X drive) on all instruments used in test setup running Microsoft Windows Operating System.

   The My TekExpress folder has the shared name format <domain><user ID>My TekExpress. Or, if the instrument is not connected to a domain, then the shared name format is <instrument name><user ID>My TekExpress. This shared folder is used to save the test session files and is used during any other file transfer operations.

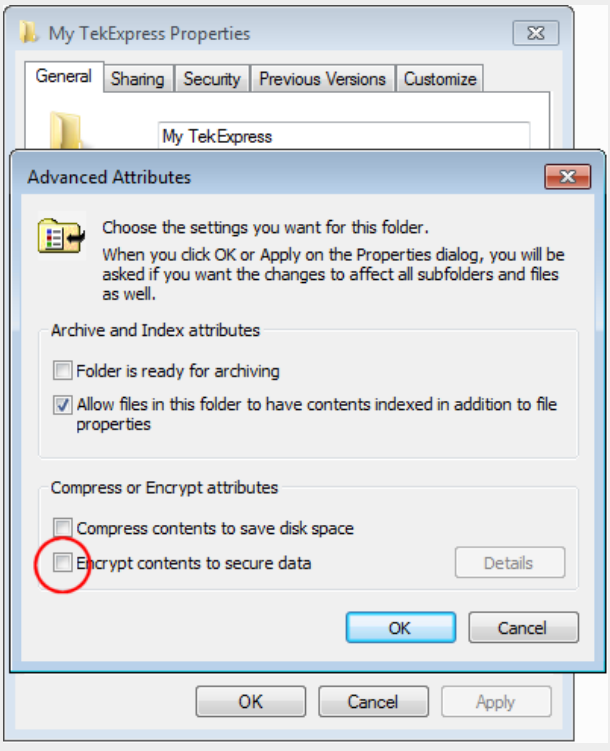
   NOTE. *If the X: drive is mapped to any other shared folder, the application will display a warning message asking you to disconnect the X: drive manually.*

3. Make sure that the My TekExpress folder has read and write access and that the contents are not set to be encrypted:

   a. Right-click the folder and select **Properties**.

   b. Select the **General** tab and then click **Advanced**.

   c. In the Advanced Attributes dialog box, make sure that the option **Encrypt contents to secure data** is NOT selected. Example



4. Review the *pre-run checklist* before you run a test.

See also *View test-related files*

Application directories and usage

File name extensions

Pre-Run checklist

Do the following before you click Start to run a test. If this is the first time you are running a test for a setup, refer to the information in *Before you click start*.

1. Make sure that all the required instruments are properly warmed up (approximately 20 minutes).
2. Perform Signal Path Compensation (SPC).
   a. On the oscilloscope main menu, select the **Utilities** menu.
   b. Select **Instrument Calibration**.
3. Perform deskew on any cables.
4. Verify that the application is able to find the DUT. If it cannot, *perform a search for connected instruments*
   a. Launch the M-PHY Rx application.
   b. Select the **Setup** panel, click the **Test Selection** tab. Select any test and then click **Configure**.
   c. In the Configuration section, click **Global Settings**.
   d. In the Instruments Detected section, click the drop-down arrow to the right of the listed instruments and make sure that the DUT is listed.

See also *Equipment connection setup*

View the progress of analysis

The *Status panel* displays a record of the test as it is executed. By default, the application switches to this panel after you click the Start button to run a test. Choose from the following two views by selecting the named tab, even while a test is in progress.

Test status

The Test Status tab presents a collapsible table with information about each test as it is running.

Table 12: Status tab table

Column	Description
Test Name	Name of the test
Analysis Status	Progress state of the analysis: ■ To be started ■ In progress ■ Completed ■ Aborted

Log view

The Log View tab provides a list of actions that happen as the test executes. Use this tab to review or troubleshoot tests.

Table 13: Log View options

Item	Description
Message History	This window timestamps and displays all run messages.
Show Detailed Log	Select this check box to record a detailed history of test execution.
Auto Scroll	Select this check box to have the program automatically scroll down as information is added to the log during the test.
Clear Log	Click this button to clear all messages in the Message History window.
Save	Use to save the log file as a text file for examination. Displays a standard Save File window and saves the status messages in the file that you specify.

See also *View test results*

View test results

When a test finishes, the application switches to the *Results panel*, which displays a summary of test results. The Overall Test Result is displayed at the top left of the Results table. If all of the tests for the session pass, the overall test result will be Pass. If even one test out of multiple tests fails, the overall test result will show Fail.

Each test result occupies a row in the Results table. By default, results are displayed in summary format with the measurement details collapsed and with the Pass/Fail column visible. Change the view in the following ways:

- To expand all tests listed, select **View Results Details** from the Preferences menu in the upper right corner.
- To expand and collapse tests, click the plus and minus buttons.
- To collapse all expanded tests, select **Preferences > View Results Summary**.
- To remove or restore the Pass/Fail column, select **Preferences > Show Pass/Fail**.
- To enable or disable the wordwrap feature, select **Preferences > Enable Wordwrap**.
- To expand the width of a column, place the cursor over the vertical line that separates the column from the column to the right. When the cursor changes to a double-ended arrow, hold down the mouse button and drag the column to the desired width.
- To sort the test information by column, click the column head. When sorted in ascending order, a small up arrow is displayed. When sorted in descending order, a small down arrow is displayed.
- To clear all test results displayed, click **Clear**.

See also *View a report*

View the progress of analysis

View a report

After the analysis, a report is automatically generated. By default, the report is displayed in your Web browser unless you cleared the **View Report After Generating** check box in the Reports panel before running the test. If you cleared this check box, or if you want to view a different test report, do the following:

1. In the Reports panel, click the **Browse** button and locate the desired report.

   The path to the report displays in the Report Name field.

   NOTE. *If you did not save the test setup after running the report and you either closed the application or you ran another report, the report file was not saved.*

2. At the bottom of the Reports panel, click **View**.

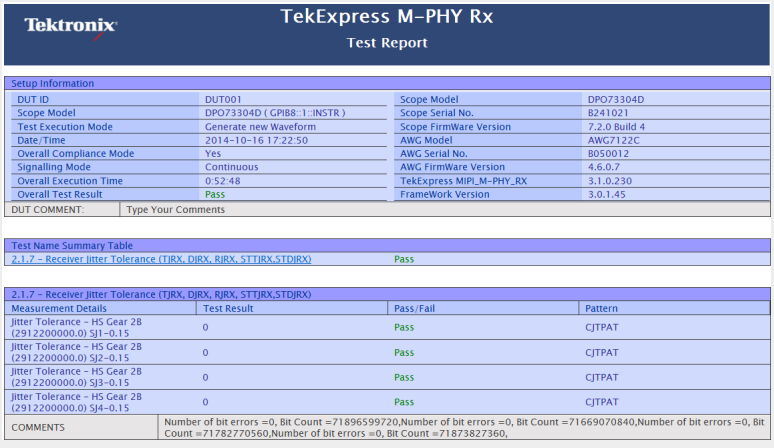
For information on changing the file type, file name, and other report options, see *Select report options*.

Report Contents

In the report, a table shows the test name, measurement details, test result (pass/fail), mode, and analysis time.

Setup configuration information

Setup configuration information is listed in the summary box at the beginning of the report. This information includes the oscilloscope model and serial number, probe model and serial number, and software versions. To exclude this information from the report, clear the **Include Setup Configuration** check box in the Reports panel before running the test.



Tektronix

TekExpress M-PHY Rx

Test Report

Setup Information			
DUT ID	DUT001	Scope Model	DPO73304D
Scope Model	DPO73304D (GPIB8::1::INSTR)	Scope Serial No.	B241021
Test Execution Mode	Generate new Waveform	Scope FirmWare Version	7.2.0 Build 4
Date/Time	2014-10-16 17:22:50	AWG Model	AWG7122C
Overall Compliance Mode	Yes	AWG Serial No.	B050012
Signalling Mode	Continuous	AWG FirmWare Version	4.6.0.7
Overall Execution Time	0:52:48	TekExpress MIPI_M-PHY_RX	3.1.0.230
Overall Test Result	Pass	FrameWork Version	3.0.1.45
DUT COMMENT:	Type Your Comments		

Test Name Summary Table	
2.1.7 – Receiver Jitter Tolerance (TJRX, DJRX, RJRX, STTJRX,STDJRX)	Pass

2.1.7 – Receiver Jitter Tolerance (TJRX, DJRX, RJRX, STTJRX,STDJRX)			
Measurement Details	Test Result	Pass/Fail	Pattern
Jitter Tolerance – HS Gear 2B (2912200000.0) SJ1-0.15	0	Pass	CJTPAT
Jitter Tolerance – HS Gear 2B (2912200000.0) SJ2-0.15	0	Pass	CJTPAT
Jitter Tolerance – HS Gear 2B (2912200000.0) SJ3-0.15	0	Pass	CJTPAT
Jitter Tolerance – HS Gear 2B (2912200000.0) SJ4-0.15	0	Pass	CJTPAT
COMMENTS	Number of bit errors =0, Bit Count =71896599720,Number of bit errors =0, Bit Count =71669070840,Number of bit errors =0, Bit Count =71782770560,Number of bit errors =0, Bit Count =71873827360,		

DUT comments

If you selected to include comments in the test report, any comments you added in the DUT tab of the Setup panel appear in the Comments section directly below the summary box.

Setup Information			
DUT ID	DUT001	Scope Model	DPO73304D
Scope Model	DPO73304D (GPIB8::1::INSTR)	Scope Serial No.	B241021
Test Execution Mode	Generate new Waveform	Scope FirmWare Version	7.2.0 Build 4
Date/Time	2014-10-16 17:22:50	AWG Model	AWG7122C
Overall Compliance Mode	Yes	AWG Serial No.	B050012
Signalling Mode	Continuous	AWG FirmWare Version	4.6.0.7
Overall Execution Time	0:52:48	TekExpress MIPI_M-PHY_RX	3.1.0.230
Overall Test Result	Pass	FrameWork Version	3.0.1.45
DUT COMMENT:	Type Your Comments		

Pass/Fail summary

The Pass/Fail Summary column indicates whether a test passed or failed. If the test passed, the column cell is green. If the test failed, it is red. To exclude this information from the report, clear the **Include Pass/Fail Results Summary** check box in the Reports panel before running the test.

Test Name Summary Table	
2.1.7 – Receiver Jitter Tolerance (TJRX, DJRX, RJRX, STTJRX,STDJRX)	Pass

2.1.7 – Receiver Jitter Tolerance (TJRX, DJRX, RJRX, STTJRX,STDJRX)			
Measurement Details	Test Result	Pass/Fail	Pattern
Jitter Tolerance – HS Gear 2B (2912200000.0) SJ1–0.15	0	Pass	CJTPAT
Jitter Tolerance – HS Gear 2B (2912200000.0) SJ2–0.15	0	Pass	CJTPAT
Jitter Tolerance – HS Gear 2B (2912200000.0) SJ3–0.15	0	Pass	CJTPAT
Jitter Tolerance – HS Gear 2B (2912200000.0) SJ4–0.15	0	Pass	CJTPAT

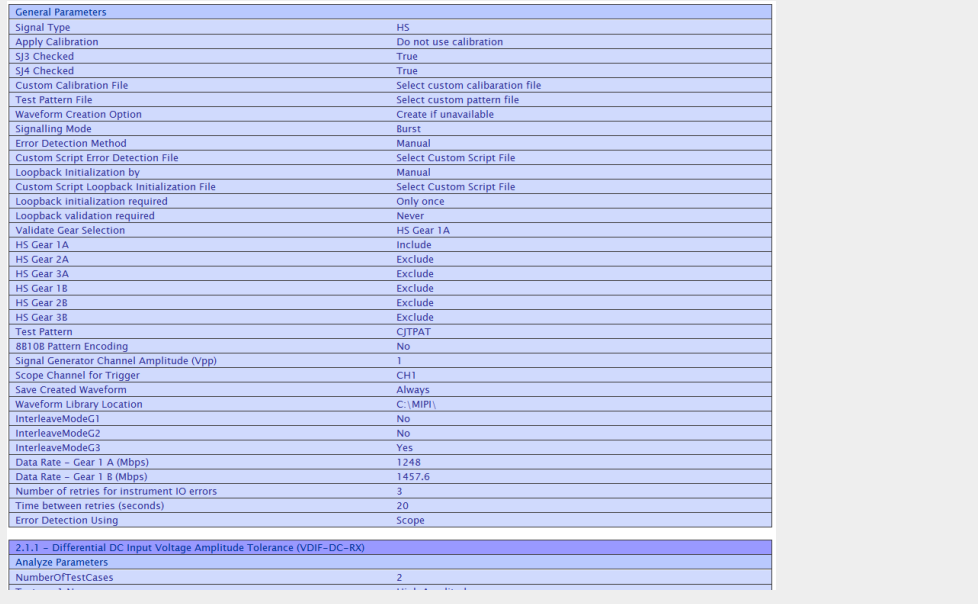
Detailed results

Includes detailed measurement information, as shown in the example. The Comments column is for test-specific information; for example, if the test had to be aborted, the reason is listed in the Comments column. To exclude this information from the report, clear the **Include Detailed Results** check box in the Reports panel before running the test.

2.1.7 – Receiver Jitter Tolerance (TJRX, DJRX, RJRX, STTJRX,STDJRX)			
Measurement Details	Test Result	Pass/Fail	Pattern
Jitter Tolerance – HS Gear 2B (2912200000.0) SJ1–0.15	0	Pass	CJTPAT
Jitter Tolerance – HS Gear 2B (2912200000.0) SJ2–0.15	0	Pass	CJTPAT
Jitter Tolerance – HS Gear 2B (2912200000.0) SJ3–0.15	0	Pass	CJTPAT
Jitter Tolerance – HS Gear 2B (2912200000.0) SJ4–0.15	0	Pass	CJTPAT
COMMENTS	Number of bit errors =0, Bit Count =71896599720,Number of bit errors =0, Bit Count =71669070840,Number of bit errors =0, Bit Count =71782770560,Number of bit errors =0, Bit Count =71873827360,		

Application configuration

This table lists general, common, and acquired parameters used in the test. General parameters are those selected to be included in all tests (where applicable). General parameters are set in the DUT tab. By default, this is not selected for inclusion in the test. To include this information in the report, select the **Include Complete Application Configuration** check box before running the test.



General Parameters	
Signal Type	HS
Apply Calibration	Do not use calibration
SJ3 Checked	True
SJ4 Checked	True
Custom Calibration File	Select custom calibaration file
Test Pattern File	Select custom pattern file
Waveform Creation Option	Create if unavailable
Signalling Mode	Burst
Error Detection Method	Manual
Custom Script Error Detection File	Select Custom Script File
Loopback Initialization by	Manual
Custom Script Loopback Initialization File	Select Custom Script File
Loopback initialization required	Only once
Loopback validation required	Never
Validate Gear Selection	HS Gear 1A
HS Gear 1A	Include
HS Gear 2A	Exclude
HS Gear 3A	Exclude
HS Gear 1B	Exclude
HS Gear 2B	Exclude
HS Gear 3B	Exclude
Test Pattern	CJTPAT
8B10B Pattern Encoding	No
Signal Generator Channel Amplitude (Vpp)	1
Scope Channel for Trigger	CH1
Save Created Waveform	Always
Waveform Library Location	C:\MIPI\
InterleaveModeG1	No
InterleaveModeG2	No
InterleaveModeG3	Yes
Data Rate – Gear 1 A (Mbps)	1248
Data Rate – Gear 1 B (Mbps)	1457.6
Number of retries for instrument IO errors	3
Time between retries (seconds)	20
Error Detection Using	Scope

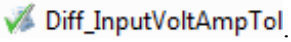
2.1.1 – Differential DC Input Voltage Amplitude Tolerance (VDIF-DC-RX)	
Analyze Parameters	
NumberOfTestCases	2

See also *View test results*

View test-related files

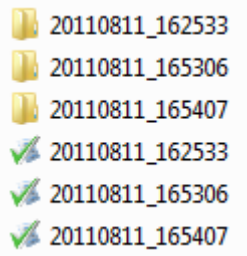
View test-related files

Files related to tests are stored in the M-PHY Rx folder under the My TekExpress shared folder. In the M-PHY Rx folder, each test setup has a test setup file and a test setup folder, both with the test setup name. The test setup file is preceded by the M-PHY Rx icon and usually has no visible file extension. For example,



Inside the test setup folder is another folder named for the DUT ID used in the test sessions. The default is DUT001.

Inside the DUT001 folder are the session folders and files. Each session has a folder and file pair, both named for the test session using the naming convention (date)_(time). Each session file is stored outside its matching session folder.



The first time you run a new, unsaved session, the session files are stored in the Untitled Session folder located at ..\My TekExpress\M-PHY Rx. When you name and save the session, the files are placed in a folder with the name that you specify.

See also *File name extensions*

TekExpress programmatic interface

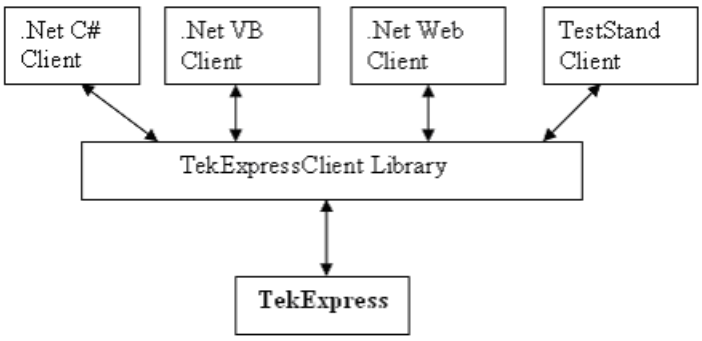
About the programmatic interface

The Programmatic interface allows you to seamlessly integrate the TekExpress test automation application with the high-level automation layer. This also allows you to control the state of the TekExpress application running on a local or a remote computer.

For simplifying the descriptions, the following terminologies are used in this section:

- **TekExpress Client:** A high-level automation application that communicates with TekExpress using TekExpress Programmatic Interface.
- **TekExpress Server:** The TekExpress application when being controlled by TekExpress Client.

TekExpress leverages .Net Marshalling to enable the Programmatic Interface for TekExpress Client. TekExpress provides a client library for TekExpress clients to use the programmatic interface. The TekExpress client library is inherited from .Net MarshalByRef class to provide the proxy object for the clients. The TekExpress client library maintains a reference to the TekExpress Server and this reference allows the client to control the server state.



See also

Requirements for developing tekExpress client

Remote proxy object

Client proxy object

Requirements for developing TekExpress client

While developing the TekExpress Client, use the TekExpressClient.dll. The client can be a VB .Net, C# .Net, TestStand or Web application. The examples for interfaces in each of these applications are in the Samples folder.

References required

- *TekExpressClient.dll* has an internal reference to *IIdlglib.dll* and *IRemoteInterface.dll*.
- *IIdlglib.dll* has a reference to *TekDotNetLib.dll*.
- *IRemoteInterface.dll* provides the interfaces required to perform the remote automations. It is an interface that forms the communication line between the server and the client.
- *IIdlglib.dll* provides the methods to generate and direct the secondary dialog messages at the client-end.

NOTE. *The end-user client application does not need any reference to the above mentioned DLL files. It is essential to have these DLLs (IRemoteInterface.dll, IIdlglib.dll and TekDotNetLib.dll) in the same folder as that of TekExpressClient.dll.*

Required steps for a client

The following steps are used by the client to programmatically control the server using TekExpressClient.dll:

Develop a client UI to access the interfaces exposed through the server. This client loads TekExpressClient.dll to access the interfaces. After TekExpressClient.dll is loaded, the client UI can call the specific functions to run the operations requested by the client. When the client is up and running, it does the following to run a remote operation:

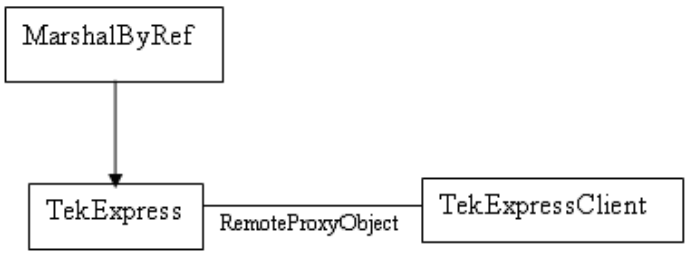
1. To connect to the server, the client provides the IP address of the PC where the server is running.
2. The client locks the server application to avoid conflict with any other Client that may try to control the server simultaneously. “Lock” would also disable all user controls on the server so that server state cannot be changed by manual operation.

   If any other client tries to access a server that is locked, it will receive a notification that the server is locked by another client.
3. When the client has connected to and locked the server, the client can access any of the programmatic controls needed to run the remote automations.
4. After the client operations finish, the client unlocks the server.

See also *About M-PHY Rx application commands*

Remote proxy object

The server exposes a remote object to let the remote client access and perform the server-side operations remotely. The proxy object is instantiated and exposed at the server-end through marshalling.



The following is an example:

RemotingConfiguration.RegisterWellKnownServiceType (typeof (TekExpressRemoteInterface), “TekExpress Remote interface”, WellKnownObjectMode.Singleton);

This object lets the remote client access the interfaces exposed at the server side. The client gets the reference to this object when the client gets connected to the server.

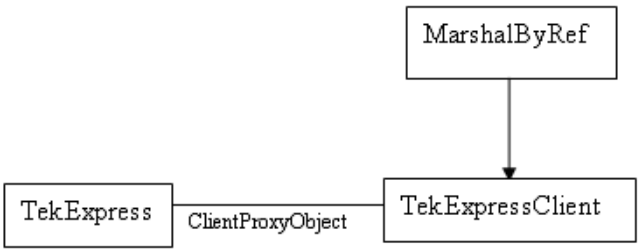
For example,

//Get a reference to the remote object

remoteObject = (IRemoteInterface)Activator.GetObject(typeof(IRemoteInterface), URL.ToString());

Client proxy object

Client exposes a proxy object to receive certain information.



For example,

//Register the client proxy object

WellKnownServiceTypeEntry[] e = RemotingConfiguration.GetRegisteredWellKnownServiceTypes();

clientInterface = new ClientInterface();

RemotingConfiguration.RegisterWellKnownServiceType(typeof(ClientInterface), “Remote Client Interface”, WellKnownObjectMode.Singleton);

//Expose the client proxy object through marshalling

RemotingServices.Marshal(clientInterface, "Remote Client Inteface");

The client proxy object is used for the following:

- To get the secondary dialog messages from the server.
- To get the file transfer commands from the server while transferring the report.

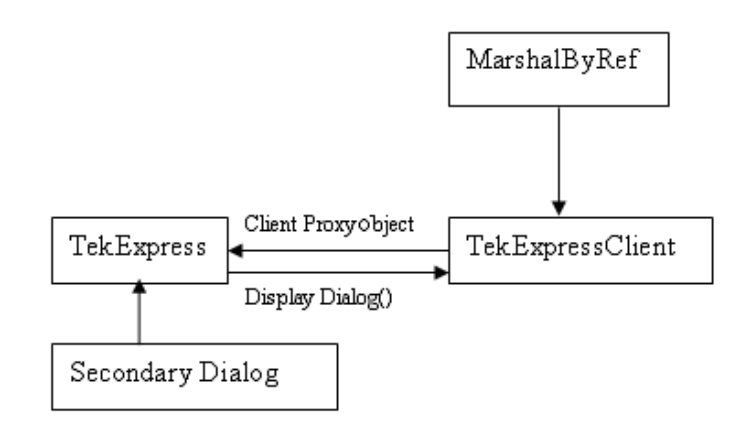
Examples

```
clientObject.clientIntf.DisplayDialog(caption, msg,iconType, btnType);

clientObject.clientIntf.TransferBytes(buffer, read, fileLength);
```

For more information, click the following links:

Secondary dialog message handling



The secondary dialog messages from the Secondary Dialog library are redirected to the client-end when a client is performing the automations at the remote end.

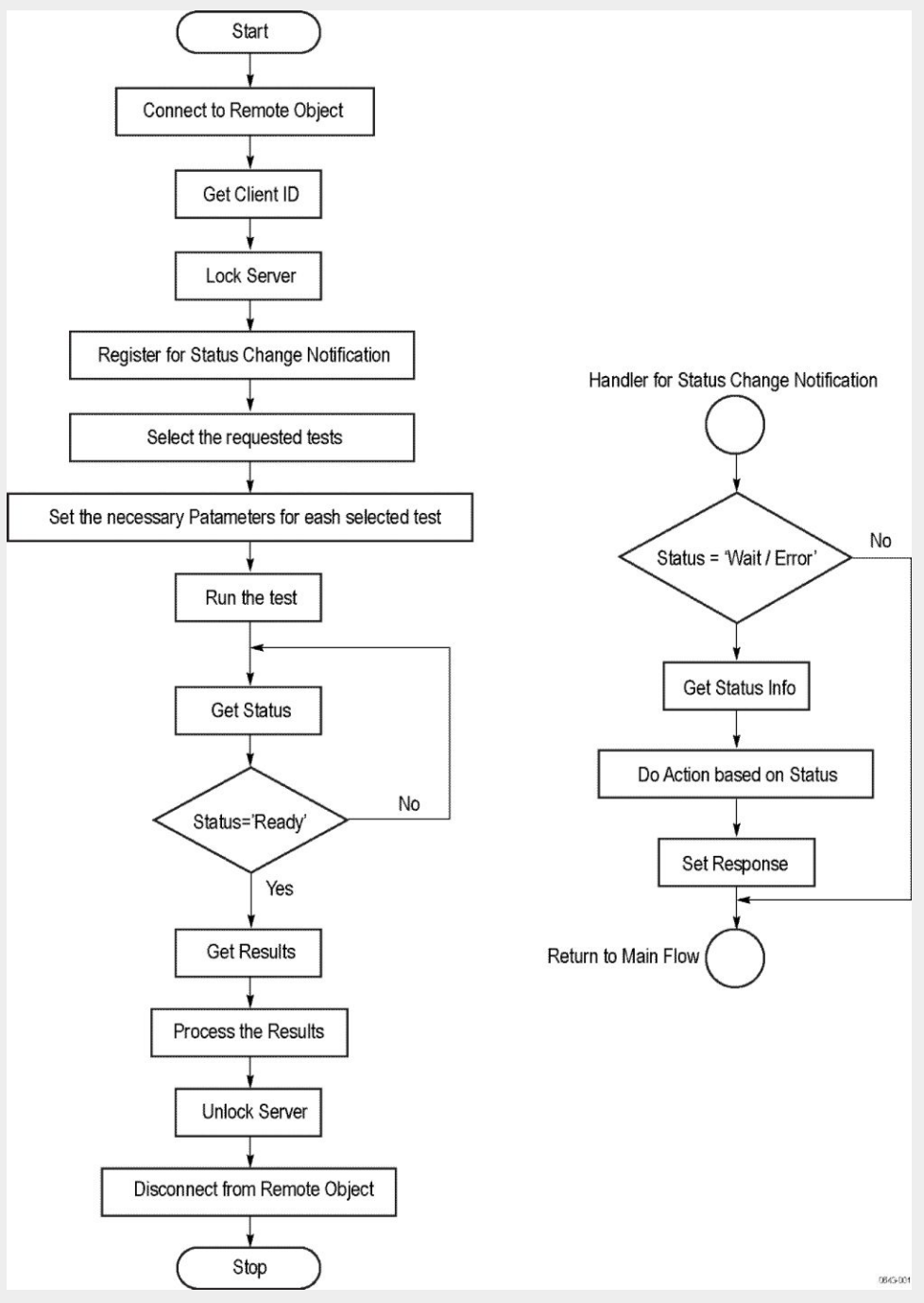
In the secondary dialog library, the assembly that is calling for the dialog box to be displayed is checked and if a remote connection is detected, the messages are directed to the remote end.

File transfer events

When the client requests the transfer of the report, the server reads the report and transfers the file by calling the file transfer methods at the client-end.

Client programmatic interface example

An example of the client programmatic interface is described and shown as follows:



1. Connect to a server or remote object using the programmatic interface provided.
2. Get the client ID that is created when connecting to the remote object. This client ID is one of the required parameters to communicate with the server.

NOTE. *The server identifies the client with this ID only and rejects any request if the ID is invalid.*

3. Lock the server for further operations. This disables the application interface.

   NOTE. *You can get values from the server or set values from the server to the client only if the application is locked.*

4. Register for receiving notifications on status change events on the server. To register you need to give a handler as a parameter. For details, see.*Handler of status change notification*

   NOTE. *Whenever there is a change in the status of the server, all the clients registered with the server receive a notification from the server.*

5. Select the tests to run through the programmatic interface.
6. Set the necessary parameters for each test.
7. Run the tests.
8. Poll for the status of the application.

   NOTE. *Skip this step if you are registered for the status change notification and the status is Ready.*

9. After completing the tests, get the results.
10. Create a report or display the results and verify or process the results.
11. Unlock the server after completing all the tasks.
12. Disconnect from the remote object.

Handler of Status Change Notification

1. Get the status. If the status is Wait or Error, get the information that contains the title, message description, and the expected responses for the status.
2. Perform the actions based on the status information.
3. Set the response as expected.

See also

About M-PHY Rx application commands

Program example

Program example

This program example shows how to communicate between a PC and TekExpress M-PHY Rx remotely.

A typical application does the following:

1. Start the application.
2. Connect through an IP address.

   m_Client.Connect("localhost") 'True or False

   clientID = m_Client.getClientID
3. Lock the server.

   m_Client.LockServer(clientID)
4. Disable the Popups.

   m_Client.SetVerboseMode(clientID, false)
5. Set the Dut ID.

   m_Client.SetDutId(clientID, "DUT_Name")
6. Select a measurement.

   mClient.SelectTest(clientID, "Receiver", "HS", "2.1.1 - Differential DC Input Voltage Amplitude Tolerance (VDIF-DC-RX)", true)
7. Select a channel.

   mClient.SetGeneralParameter(clientID, "Receiver", "HS", "2.1.1 - Differential DC Input Voltage Amplitude Tolerance (VDIF-DC-RX)", "Scope Channel for Trigger$CH1")
8. Configure the selected measurement.

   mClient.SetGeneralParameter(clientID, "Receiver", "HS", "2.1.1 - Differential DC Input Voltage Amplitude Tolerance (VDIF-DC-RX)", Loopback validation required$Always)'Set loopback validation to Always

   mClient.SetGeneralParameter(clientID, "Receiver", "HS", "2.1.1 - Differential DC Input Voltage Amplitude Tolerance (VDIF-DC-RX)", HS Gear 1A$Include) 'Select Gear 1A
9. Run with set configurations.

   m_Client.Run(clientID)
10. Wait for the test to complete.

    Do

    Thread.Sleep(500)

    m_Client.Application_Status(clientID)

    Select Case status

    Case "Wait"

'Get the Current State Information

mClient.GetCurrentStateInfo(clientID, WaitingMsbBxCaption, WaitingMsbBxMessage, WaitingMsbBxButtontexts)

'Send the Response

mClient.SendResponse(clientID, WaitingMsbBxCaption, WaitingMsbBxMessage, WaitingMsbBxResponse)

End Select

Loop Until status = "Ready"

11. After the Test is Complete.

    'Save all results values from folder for current run

    m_Client.TransferResult(clientID, logDirname)

12. Unlock the server.

    m_Client.UnlockServer(clientID)

13. Disconnect from server.

    m_Client.Disconnect()

14. Exit the application.

M-PHY Rx application commands

About M-PHY Rx application commands

Click a client action below to see the command name, description, parameters, return value, and an example, associated with the action.

Connect through an IP address

Lock the server

Disable the popups

Set or get the DUT ID

Set the configuration parameters for a suite or measurement

Query the configuration parameters for a suite or measurement

Select a measurement

Select a suite

Select a channel

Configure the selected measurement

Handle Error Codes

Get or set the timeout value

Wait for the test to complete

After the test is complete

Save, recall, or check if a session is saved

Unlock the server

Disconnect from server

string id

Name	Type	Direction	Description
id	string	IN	Identifier of the client performing the remote function

Ready: Test configured and ready to start

Running: Test running

Paused: Test paused

Wait: A popup that needs your inputs

Error: An error is occurred

string dutName

Name	Type	Direction	Description
dutName	string	IN	The new DUT ID of the setup

out bool saved

Name	Type	Direction	Description
saved	bool	OUT	Boolean representing whether the current session is saved

This parameter is used as a check in SaveSession() and SaveSessionAs() functions.

string ipAddress

Name	Type	Direction	Description
ipAddress	string	IN	The ip address of the server to which the client is trying to connect. This is required to establish the connection between the server and the client.

out string clientID

Name	Type	Direction	Description
clientid	string	OUT	Identifier of the client that is connected to the server clientId = unique number + ipaddress of the client. For example, 1065–192.157.98.70

NOTE. *If the dutName parameter is null, the client is prompted to provide a valid DUT ID.*

NOTE. *The server must be active and running for the client to connect to the server. Any number of clients can be connected to the server at a time.*

NOTE. *When the client is disconnected, it is unlocked from the server and then disconnected. The id is reused.*

string dutId

Name	Type	Direction	Description
dutId	string	OUT	The DUT ID of the setup

The dutId parameter is set after the server processes the request.

string device

Name	Type	Direction	Description
device	string	IN	Specifies the name of the device

string suite

Name	Type	Direction	Description
suite	string	IN	Specifies the name of the suite

string test

Name	Type	Direction	Description
test	string	IN	Specifies the name of the test to obtain the pass or fail status

string parameterString

Name	Type	Direction	Description
parameterString	string	IN	Selects or deselects a test

int rowNr

Name	Type	Direction	Description
rowNr	int	IN	Specifies the zero based row index of the sub-measurement for obtaining the result value

NOTE. *When the client tries to lock a server that is locked by another client, the client gets a notification that the server is already locked and it must wait until the server is unlocked. If the client locks the server and is idle for a certain amount of time then the server is unlocked automatically from that client.*

out string[] status

Name	Type	Direction	Description
status	string array	OUT	The list of status messages generated during the run

string name

Name	Type	Direction	Description
name	string	IN	The name of the session being recalled

The name parameter cannot be empty. If it is empty, the client is prompted to provide a valid name.

NOTE. *When the run is performed, the status of the run is updated periodically using a timer.*

string name

Name	Type	Direction	Description
name	string	IN	The name of the session being saved

The name parameter cannot be empty. If it is empty, the client is prompted to provide a valid name.

Once the session is saved under 'name' you cannot use this method to save the session in a different name. Use SaveSessionAs instead.

string name

Name	Type	Direction	Description
name	string	IN	The name of the session being recalled

The same session is saved under different names using this method. The name parameter cannot be empty. If it is empty, the client is prompted to provide a valid name.

bool isSelected

Name	Type	Direction	Description
isSelected	bool	IN	Selects or deselects a test

string time

Name	Type	Direction	Description
time	string	IN	The time in seconds that refers to the timeout period

The time parameter gives the timeout period, which is the time the client is allowed to be locked and idle. After the timeout period if the client is still idle, it gets unlocked.

The time parameter should be a positive integer; otherwise, the client is prompted to provide a valid timeout period.

bool_verbose

Name	Type	Direction	Description
_verbose	bool	IN	Specifies whether the verbose mode should be turned ON or OFF

NOTE. *When the session is stopped, the client is prompted to stop the session and is stopped at the consent.*

string filePath

Name	Type	Direction	Description
filePath	string	IN	The location where the report must be saved in the client

NOTE. *If the client does not provide the location to save the report, the report is saved at C:\ProgramFiles.*

NOTE. *When the client is disconnected, the client is unlocked automatically.*

out string caption

Name	Type	Direction	Description
caption	string	OUT	The wait state or error state message sent to you

out string message

Name	Type	Direction	Description
message	string	OUT	The wait state/error state message sent to you

out string[] buttonTexts

Name	Type	Direction	Description
buttonTexts	string array	OUT	An array of strings containing the possible response types that you can send

string response

Name	Type	Direction	Description
response	string	IN	A string containing the response type that you can select (it must be one of the strings in the string array buttonTexts)

out string clientID

Name	Type	Direction	Description
clientID	string	OUT	Identifier of the client that is connected to the server clientID = unique number + ipaddress of the client. For example, 1065–192.157.98.70

Connect through an IP address

Command name	Parameters	Description	Return value	Example

Connect()	string ipAddress
	out string clientID
		This method connects the client to the server. Note The client provides the IP address to connect to the server.
		The server provides a unique client identification number when connected to it.
			Return value is either True or False	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.
				returnval as boolean
				returnval = m_Client.Connect(ipaddress,m_clientID)

string ipAddress

Name	Type	Direction	Description
ipAddress	string	IN	The ip address of the server to which the client is trying to connect. This is required to establish the connection between the server and the client.

out string clientID

Name	Type	Direction	Description
clientid	string	OUT	Identifier of the client that is connected to the server clientId = unique number + ipaddress of the client. For example, 1065-192.157.98.70

NOTE. *The server must be active and running for the client to connect to the server. Any number of clients can be connected to the server at a time.*

NOTE. *The Fail condition for PI commands occurs in any of the following cases:*

The server is LOCKED and the message displayed is "Server is locked by another client".

The session is UNLOCKED and the message displayed is "Lock Session to execute the command".

The server is NOTFOUND and the message displayed is "Server not found...Disconnect!".

When none of these fail conditions occur, then the message displayed is "Failed...".

Lock the server

Command name	Parameters	Description	Return value	Example

LockSession()	string clientID	This method locks the server. Note The client must call this method before running any of the remote automations. The server can be locked by only one client.	String value that gives the status of the operation after it has been performed
			The return value is "Session Locked...” on success.
				m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.
				returnval as string
				returnval = m_Client.LockServer(clientID)

NOTE. *The Fail condition for PI commands occurs in any of the following cases:*

The server is LOCKED and the message displayed is "Server is locked by another client".

The session is UNLOCKED and the message displayed is "Lock Session to execute the command".

The server is NOTFOUND and the message displayed is "Server not found...Disconnect!".

When none of these fail conditions occur, then the message displayed is "Failed...".

out string clientID

Name	Type	Direction	Description
clientID	string	OUT	Identifier of the client that is connected to the server clientID = unique number + ipaddress of the client. For example, 1065+192.157.98.70

NOTE. *When the client tries to lock a server that is locked by another client, the client gets a notification that the server is already locked and it must wait until the server is unlocked. If the client locks the server and is idle for a certain amount of time then the server is unlocked automatically from that client.*

Disable the popups

Command name	Parameters	Description	Return value	Example

SetVerboseMode()	string clientID
	bool _verbose
		This method sets the verbose mode to either true or false.
		When the value is set to true, any message boxes that appear during the application will be routed to the client machine that is controlling TekExpress.
		When the value is set to false, all the message boxes are shown on the server machine.
			String that gives the status of the operation after it has been performed
			When Verbose mode is set to true, the return value is “Verbose mode turned on. All dialog boxes will be shown to client”.
			When Verbose mode is set to false, the return value is “Verbose mode turned off. All dialog boxes will be shown to server”.
				m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.
				returnval as string
				Verbose mode is turned on
				return=m_Client.SetVerboseMode(clientID, true)
				Verbose mode is turned off
				returnval=m_Client.SetVerboseMode(clientID, false)

out string clientID

Name	Type	Direction	Description
clientID	string	OUT	Identifier of the client that is connected to the server clientID = unique number + ipaddress of the client. For example, 1065+192.157.98.70

bool_verbose

Name	Type	Direction	Description
_verbose	bool	IN	Specifies whether the verbose mode should be turned ON or OFF

NOTE. *The Fail condition for PI commands occurs in any of the following cases:*

The server is LOCKED and the message displayed is "Server is locked by another client".

The session is UNLOCKED and the message displayed is "Lock Session to execute the command".

The server is NOTFOUND and the message displayed is "Server not found...Disconnect!".

When none of these fail conditions occur, then the message displayed is "Failed...".

Set or get the DUT ID

Command name	Parameters	Description	Return value	Example

SetDutId()	string clientID
	string dutName
		This method changes the DUT ID of the setup. The client must provide a valid DUT ID.	String that gives the status of the operation after it has been performed
			Return value is “DUT Id Changed” on success
				m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.
				returnval as string
				return=m_Client.SetDutId(clientID,desiredDutId) Note

GetDutId()	string clientID
	string dutId
		This method gets the DUT ID of the current setup.	String that gives the status of the operation after it has been performed	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.
				returnval as string
				return=m_Client.GetDutid(clientID, out DutId)

out string clientID

Name	Type	Direction	Description
clientID	string	OUT	Identifier of the client that is connected to the server clientID = unique number + ipaddress of the client. For example, 1065 + 192.157.98.70

string dutName

Name	Type	Direction	Description
dutName	string	IN	The new DUT ID of the setup

out string clientID

Name	Type	Direction	Description
clientID	string	OUT	Identifier of the client that is connected to the server clientID = unique number + ipaddress of the client. For example, 1065 + 192. 157.98.70

string dutID

Name	Type	Direction	Description
dudID	string	OUT	The DUT ID of the setup

NOTE. *If the dutName parameter is null, the client is prompted to provide a valid DUT ID.*

NOTE. *The Fail condition for PI commands occurs in any of the following cases:*

The server is LOCKED and the message displayed is "Server is locked by another client".

The session is UNLOCKED and the message displayed is "Lock Session to execute the command".

The server is NOTFOUND and the message displayed is "Server not found...Disconnect!".

When none of these fail conditions occur, then the message displayed is "Failed...".

Set the configuration parameters for a suite or measurement

Command name	Parameters	Description	Return value	Example

SetGeneralParameter	string clientID
	string device
	string suite
	string test
	string parameterString
		This method sets the number of video lanes for the selected measurement.
		NOTE. *Use this command to select a lane, channel, or source type.*
			String that gives the status of the operation after it has been performed
			The return value is "" (an empty String) on success.
				m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.
				returnval as string
				Select Channel
				Select Measurement
				Method

SetAnalyzeParameter()	string clientID
	string device
	string suite
	string test
	string parameterString
		This method sets the configuration parameters in the Analyze panel of the Configuration Panel dialog box for a given suite or measurement.	The return value is "" (an empty String) on success.	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL .
				returnval as string

SetAcquireParameter()	string clientID string device string suite string test string parameterString	This method sets the configuration parameters in the Acquire panel of the Configuration Panel dialog box for a given suite or measurement.	returnVal = remoteObject.SetAcquireParameter(id, device, suite, test, parameterString) if ((OP_STATUS) returnVal != OP_STATUS.SUCCESS) return CommandFailed(returnVal)	

out string clientID

Name	Type	Direction	Description
clientID	string	OUT	Identifier of the client that is connected to the server clientID = unique number + ipaddress of the client. For example, 1065 + 192.157.98.70

string device

Name	Type	Direction	Description
device	string	IN	Specifies the name of the device

string suite

Name	Type	Direction	Description
suite	string	IN	Specifies the name of the suite

string test

Name	Type	Direction	Description
test	string	IN	Specifies the name of the test to obtain the pass or fail status

string parameterString

Name	Type	Direction	Description
parameterString	string	IN	Selects or deselects a test

out string clientID

Name	Type	Direction	Description
clientID	string	OUT	Identifier of the client that is connected to the server clientID = unique number + ipaddress of the client. For example, 1065 + 192.157.98.70

string device

Name	Type	Direction	Description
device	string	IN	Specifies the name of the device

string suite

Name	Type	Direction	Description
suite	string	IN	Specifies the name of the suite

string test

Name	Type	Direction	Description
test	string	IN	Specifies the name of the test to obtain the pass or fail status

string parameterString

Name	Type	Direction	Description
parameterString	string	IN	Selects or deselects a test

out string clientID

Name	Type	Direction	Description
clientID	string	OUT	Identifier of the client that is connected to the server clientID = unique number + ipaddress of the client. For example, 1065 + 192.157.98.70

string device

Name	Type	Direction	Description
device	string	IN	Specifies the name of the device

string suite

Name	Type	Direction	Description
suite	string	IN	Specifies the name of the suite

string test

Name	Type	Direction	Description
test	string	IN	Specifies the name of the test to obtain the pass or fail status

string parameterString

Name	Type	Direction	Description
parameterString	string	IN	Selects or deselects a test

NOTE. *The Fail condition for PI commands occurs in any of the following cases:*

The server is LOCKED and the message displayed is "Server is locked by another client".

The session is UNLOCKED and the message displayed is "Lock Session to execute the command".

The server is NOTFOUND and the message displayed is "Server not found...Disconnect!".

When none of these fail conditions occur, then the message displayed is "Failed...".

Select Channel Example

returnval=mClient.SetGeneralParameter(clientID, "Receiver", "HS", "2.1.1 - Differential DC Input Voltage Amplitude Tolerance (VDIF-DC-RX)", "Scope Channel for Trigger$CH1")

Select Loopback Validation Method Example

returnval=mClient.SetGeneralParameter(clientID, "Receiver", "HS", "2.1.1 - Differential DC Input Voltage Amplitude Tolerance (VDIF-DC-RX)", Loopback validation required$Always)

Query the configuration parameters for a suite or measurement

Command name	Parameters	Description	Return value	Example

GetGeneralParameter()	string clientID
	string device
	string suite
	string test
	string parameterString
		This method gets the general configuration parameters for a given suite or measurement.	The return value is the general configuration parameter for a given suite or measurement that is set.	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.
				returnval as string
				Query Channel
				Query Loopback Validation Method

GetAnalyzeParameter()	string clientID
	string device
	string suite
	string test
	string parameterString
		This method gets the configuration parameters set in the Analyze panel of the Configuration Panel dialog box for a given suite or measurement.	The return value is the configuration parameter set in the Analyze panel of the Configuration Panel dialog box for a given suite or measurement.	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL .
				returnval as string

GetAcquireParameter()	string clientID string device string suite string test string parameterString	This method gets the configuration parameters set in the Acquire panel for a given suite or measurement.	The return value is the configuration parameter set in the Acquire panel for a given suite or measurement.	

out string clientID

Name	Type	Direction	Description
clientID	string	OUT	Identifier of the client that is connected to the server clientID = unique number + ipaddress of the client. For example, 1065 + 192.157.98.70

string device

Name	Type	Direction	Description
device	string	IN	Specifies the name of the device

string suite

Name	Type	Direction	Description
suite	string	IN	Specifies the name of the suite

string test

Name	Type	Direction	Description
test	string	IN	Specifies the name of the test to obtain the pass or fail status

string parameterString

Name	Type	Direction	Description
parameterString	string	IN	Selects or deselects a test

out string clientID

Name	Type	Direction	Description
clientID	string	OUT	Identifier of the client that is connected to the server clientID = unique number + ipaddress of the client. For example, 1065 + 192.157.98.70

string device

Name	Type	Direction	Description
device	string	IN	Specifies the name of the device

string suite

Name	Type	Direction	Description
suite	string	IN	Specifies the name of the suite

string test

Name	Type	Direction	Description
test	string	IN	Specifies the name of the test to obtain the pass or fail status

string parameterString

Name	Type	Direction	Description
parameterString	string	IN	Selects or deselects a test

out string clientID

Name	Type	Direction	Description
clientID	string	OUT	Identifier of the client that is connected to the server clientID = unique number + ipaddress of the client. For example, 1065 + 192.157.98.70

string device

Name	Type	Direction	Description
device	string	IN	Specifies the name of the device

string suite

Name	Type	Direction	Description
suite	string	IN	Specifies the name of the suite

string test

Name	Type	Direction	Description
test	string	IN	Specifies the name of the test to obtain the pass or fail status

string parameterString

Name	Type	Direction	Description
parameterString	string	IN	Selects or deselects a test

NOTE. *The Fail condition for PI commands occurs in any of the following cases:*

The server is LOCKED and the message displayed is "Server is locked by another client".

The session is UNLOCKED and the message displayed is "Lock Session to execute the command".

The server is NOTFOUND and the message displayed is "Server not found...Disconnect!".

When none of these fail conditions occur, then the message displayed is "Failed...".

Query Channel for Trigger Example

returnval = mClient.GetGeneralParameter(clientID, "Receiver", "HS", "2.1.1 - Differential DC Input Voltage Amplitude Tolerance ($V_{DIF-DC-RX}$)", "Scope Channel for Trigger")

Query Loopback Validation Method Example

returnval=mClient.GetGeneralParameter(clientID, "Receiver", "HS", "2.1.1 - Differential DC Input Voltage Amplitude Tolerance ($V_{DIF-DC-RX}$)", Loopback validation required)

Select a measurement

Command name	Parameters	Description	Return value	Example

SelectTest()	string clientID
	string device
	string suite
	string test
	bool isSelected
		This method selects or deselects a given test.
		If this Setting parameter is set to true, you can select a measurement.
		If this Setting parameter is set to false, you can deselect a measurement.
			String that displays the status of the operation after it has been performed
			The return value is "" (an empty String) on success.
				m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL
				returnval as string
				Select Measurement 2.1.1 – Differential Input Voltage Amplitude Tolerance (VDIF-RX):
				returnval = mClient.SelectTest(clientID, "Receiver", "HS", "2.1.1 - Differential DC Input Voltage Amplitude Tolerance ($V_{DIF-DC-RX}$)", true)

out string clientID

Name	Type	Direction	Description
clientID	string	OUT	Identifier of the client that is connected to the server clientID = unique number + ipaddress of the client. For example, 1065 + 192.157.98.70

string device

Name	Type	Direction	Description
device	string	IN	Specifies the name of the device

string suite

Name	Type	Direction	Description
suite	string	IN	Specifies the name of the suite

string test

Name	Type	Direction	Description
test	string	IN	Specifies the name of the test to obtain the pass or fail status

bool isSelected

Name	Type	Direction	Description
isSelected	bool	IN	Selects or deselects a test

NOTE. *The Fail condition for PI commands occurs in any of the following cases:*

The server is LOCKED and the message displayed is "Server is locked by another client".

The session is UNLOCKED and the message displayed is "Lock Session to execute the command".

The server is NOTFOUND and the message displayed is "Server not found...Disconnect!".

When none of these fail conditions occur, then the message displayed is "Failed...".

Select a suite

Command name	Parameters	Description	Return value	Example

SelectSuite()	string clientID
	string device
	string suite
	bool isSelected
		This method selects or deselects a given suite.
		When this parameter is set to true, you can select a suite.
		When this parameter is set to false, you can deselect a suite.
			String that gives the status of the operation after it has been performed
			The return value is "" (an empty String) on success.
				m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.
				returnval as string
				Select Suite (Default):
				returnval=mClient.SelectSuitet(clientID, "Receiver", "HS", true)

out string clientID

Name	Type	Direction	Description
clientID	string	OUT	Identifier of the client that is connected to the server clientID = unique number + ipaddress of the client. For example, 1065 + 192.157.98.70

string device

Name	Type	Direction	Description
device	string	IN	Specifies the name of the device

string suite

Name	Type	Direction	Description
suite	string	IN	Specifies the name of the suite

bool isSelected

Name	Type	Direction	Description
isSelected	bool	IN	Selects or deselects a test

NOTE. *The Fail condition for PI commands occurs in any of the following cases:*

The server is LOCKED and the message displayed is "Server is locked by another client".

The session is UNLOCKED and the message displayed is "Lock Session to execute the command".

The server is NOTFOUND and the message displayed is "Server not found...Disconnect!".

When none of these fail conditions occur, then the message displayed is "Failed...".

Select a channel

Command name	Parameters	Description	Return value	Example

SetGeneralParameter()	string clientID
	string device
	string suite
	string test
	string parameterString
		This method sets the parameters that are not specific to any given test.
		NOTE. *Using this command we can select a lane, channel, or source type.*
			String that gives the status of the operation after it has been performed
			The return value is "" (an empty String) on success.
				m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.
				returnval as string
				Select Channel for Trigger
				Select Loopback Validation Method

SetAnalyzeParameter()	string clientID
	string device
	string suite
	string test
	string parameterString
		This method sets the configuration parameters in the Analyze panel of the Configuration Panel dialog box for a given suite or measurement.	The return value is "" (an empty String) on success.	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.
				returnval as string

SetAcquireParameter()	string clientID string device string suite string test string parameterString	This method sets the configuration parameters in the Acquire panel of the Configuration Panel dialog box for a given suite or measurement.	returnVal = remoteObject.SetAcquireParameter(id, device, suite, test, parameterString) if ((OP_STATUS) returnVal != OP_STATUS.SUCCESS) return CommandFailed(returnVal)	

out string clientID

Name	Type	Direction	Description
clientID	string	OUT	Identifier of the client that is connected to the server clientID = unique number + ipaddress of the client. For example, 1065 + 192.157.98.70

string device

Name	Type	Direction	Description
device	string	IN	Specifies the name of the device

string suite

Name	Type	Direction	Description
suite	string	IN	Specifies the name of the suite

string test

Name	Type	Direction	Description
test	string	IN	Specifies the name of the test to obtain the pass or fail status

string parameterString

Name	Type	Direction	Description
parameterString	string	IN	Selects or deselects a test

out string clientID

Name	Type	Direction	Description
clientID	string	OUT	Identifier of the client that is connected to the server clientID = unique number + ipaddress of the client. For example, 1065 + 192.157.98.70

string device

Name	Type	Direction	Description
device	string	IN	Specifies the name of the device

string suite

Name	Type	Direction	Description
suite	string	IN	Specifies the name of the suite

string test

Name	Type	Direction	Description
test	string	IN	Specifies the name of the test to obtain the pass or fail status

string parameterString

Name	Type	Direction	Description
parameterString	string	IN	Selects or deselects a test

out string clientID

Name	Type	Direction	Description
clientID	string	OUT	Identifier of the client that is connected to the server clientID = unique number + ipaddress of the client. For example, 1065 + 192.157.98.70

string device

Name	Type	Direction	Description
device	string	IN	Specifies the name of the device

string suite

Name	Type	Direction	Description
suite	string	IN	Specifies the name of the suite

string test

Name	Type	Direction	Description
test	string	IN	Specifies the name of the test to obtain the pass or fail status

string parameterString

Name	Type	Direction	Description
parameterString	string	IN	Selects or deselects a test

NOTE. *The Fail condition for PI commands occurs in any of the following cases:*

The server is LOCKED and the message displayed is "Server is locked by another client".

The session is UNLOCKED and the message displayed is "Lock Session to execute the command".

The server is NOTFOUND and the message displayed is "Server not found...Disconnect!".

When none of these fail conditions occur, then the message displayed is "Failed...".

Select Channel for Trigger Example

returnval = mClient.SetGeneralParameter(clientID, "Receiver", "HS", "2.1.1 - Differential DC Input Voltage Amplitude Tolerance ($V_{DIF-DC-RX}$)", "Scope Channel for Trigger$CH1")

Select Loopback Validation Method Example

returnval=mClient.SetGeneralParameter(clientID, "Receiver", "HS", "2.1.1 - Differential DC Input Voltage Amplitude Tolerance ($V_{DIF-DC-RX}$)", Loopback validation required$Always)

Configure the selected measurement

Command name	Parameters	Description	Return value	Example
SetAnalyzeParameter()	string clientID string device string suite string test string parameterString	This method sets the Analyze parameters (Configuration parameters) for a given test.	The return value is "" (an empty String) on success.	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string

out string clientID

Name	Type	Direction	Description
clientID	string	OUT	Identifier of the client that is connected to the server clientID = unique number + ipaddress of the client. For example, 1065 + 192.157.98.70

string device

Name	Type	Direction	Description
device	string	IN	Specifies the name of the device

string suite

Name	Type	Direction	Description
suite	string	IN	Specifies the name of the suite

string test

Name	Type	Direction	Description
test	string	IN	Specifies the name of the test to obtain the pass or fail status

string parameterString

Name	Type	Direction	Description
parameterString	string	IN	Selects or deselects a test

NOTE. *The Fail condition for PI commands occurs in any of the following cases:*

The server is LOCKED and the message displayed is "Server is locked by another client".

The session is UNLOCKED and the message displayed is "Lock Session to execute the command".

The server is NOTFOUND and the message displayed is "Server not found...Disconnect!".

When none of these fail conditions occur, then the message displayed is "Failed...".

Run with set configurations or stop the run operation

Command name	Parameters	Description	Return value	Example
Run()	string clientID	Runs the selected tests Note After the server is set up and configured, run it remotely using this function.	String that gives the status of the operation after it has been performed. The return value is "Run started..." on success.	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnval=m_Client.Run(clientID)
Stop()	string clientID	Stops the currently running tests Note	String that gives the status of the operation after it has been performed The return value is "Stopped..." on success.	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnval=m_Client.Stop(clientID)

out string clientID

Name	Type	Direction	Description
clientID	string	OUT	Identifier of the client that is connected to the server clientID = unique number + ipaddress of the client. For example, 1065 + 192.157.98.70

out string clientID

Name	Type	Direction	Description
clientID	string	OUT	Identifier of the client that is connected to the server clientID = unique number + ipaddress of the client. For example, 1065 + 192.157.98.70

NOTE. *When the run is performed, the status of the run is updated periodically using a timer.*

NOTE. *When is the session is stopped, the client is prompted to stop the session and is stopped at the consent.*

NOTE. *The Fail condition for PI commands occurs in any of the following cases:*

The server is LOCKED and the message displayed is "Server is locked by another client".

The session is UNLOCKED and the message displayed is "Lock Session to execute the command".

The server is NOTFOUND and the message displayed is "Server not found...Disconnect!".

When none of these fail conditions occur, then the message displayed is "Failed...".

Handle error codes

The return value of the remote automations at the server-end is OP_STATUS, which changes to a string value depending on its code, and returned to the client. The values of OP_STATUS are as follows:

Value	Code	Description
FAIL	−1	The operation failed.
SUCCESS	1	The operation succeeded.
NOTFOUND	2	Server not found
LOCKED	3	The server is locked by another client, so the operation cannot be performed.
UNLOCK	4	The server is not locked. Lock the server before performing the operation.
NULL	0	Nothing

NOTE. *The Fail condition for PI commands occurs in any of the following cases:*

The server is LOCKED and the message displayed is "Server is locked by another client".

The session is UNLOCKED and the message displayed is "Lock Session to execute the command".

The server is NOTFOUND and the message displayed is "Server not found...Disconnect!".

When none of these fail conditions occur, then the message displayed is "Failed...".

Get or set the timeout value

Command name	Parameters	Description	Return value	Example

GetTimeOut()	string clientID	Returns the current timeout period set by the client	String that gives the status of the operation after it has been performed
			The default return value is 1800000.
				m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.
				returnval as string
				returnval=m_Client.GetTimeOut()

SetTimeOut()	string clientID
	string time
		Sets a timeout period specified by the client. After this timeout period expires, the server is unlocked automatically.	String that gives the status of the operation after it has been performed
			On success the return value is "TimeOut Period Changed".
				m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.
				returnval as string
				returnval=m_Client.SetTimeOut(clientID, desiredTimeOut)

out string clientID

Name	Type	Direction	Description
clientID	string	OUT	Identifier of the client that is connected to the server clientID = unique number + ipaddress of the client. For example, 1065 + 192.157.98.70

out string clientID

Name	Type	Direction	Description
clientID	string	OUT	Identifier of the client that is connected to the server clientID = unique number + ipaddress of the client. For example, 1065 + 192.157.98.70

string time

Name	Type	Direction	Description
time	string	IN	The time in seconds that refers to the timeout period

The time parameter gives the timeout period, which is the time the client is allowed to be locked and idle. After the timeout period if the client is still idle, it gets unlocked.

The time parameter should be a positive integer; otherwise, the client is prompted to provide a valid timeout period.

NOTE. *The Fail condition for PI commands occurs in any of the following cases:*

The server is LOCKED and the message displayed is "Server is locked by another client".

The session is UNLOCKED and the message displayed is "Lock Session to execute the command".

The server is NOTFOUND and the message displayed is "Server not found...Disconnect!".

When none of these fail conditions occur, then the message displayed is "Failed...".

Wait for the test to complete

The commands in this group execute while tests are running. The GetCurrentStateInfo() and SendResponse() commands are executed when the application is running and in the wait state.

Command name	Parameters	Description	Return value	Example

ApplicationStatus()	string clientID	This method gets the status of the server application. The states at a given time are Ready, Running, Paused, Wait, or Error.	String value that gives the status of the server application	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.
				returnval as string
				returnval=m_Client.ApplicationStatus(clientID)

QueryStatus()	string clientID
	out string[] status
		An interface for the user to transfer Analyze panel status messages from the server to the client	String that gives the status of the operation after it has been performed
			On success the return value is “Transferred...”.
				m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.
				returnval as string
				returnVal=m_Client.QueryStatus(clientID, out statusMessages)
				if ((OP_STATUS)returnVal == OP_STATUS.SUCCESS)
				return “Status updated...”
				else
				return CommandFailed(returnVal)

GetCurrentStateInfo()
NOTE. *This command is used when the application is running and is in the wait or error state.*
	string clientID
	out string caption
	out string message
	out string[] buttonTexts
		This method gets the additional information of the states when the application is in Wait or Error state. Except client ID, all the others are Out parameters.	This command does not return any value.
			This function populates the Out parameters that are passed when invoking this function.
				m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL
				m_Client.GetCurrentStateInfo(clientID, caption,message, buttonTexts)

Command name	Parameters	Description	Return value	Example

SendResponse()
NOTE. *This command is used when the application is running and is in the wait or error state.*
	string clientID
	out string caption
	out string message
	string response
		After receiving the additional information using the method GetCurrentStateInfo(), the client can decide which response to send and then send the response to the application using this function. The response should be one of the strings that was received earlier as a string array in the GetCurrentStateInfo function. The _caption and _message should match the information received earlier in the GetCurrentStateInfo function.	This command does not return any value.	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL
				m_Client.SendResponse(clientID, caption,message, response)

out string clientID

Name	Type	Direction	Description
clientID	string	OUT	Identifier of the client that is connected to the server clientID = unique number + ipaddress of the client. For example, 1065 + 192.157.98.70

out string clientID

Name	Type	Direction	Description
clientID	string	OUT	Identifier of the client that is connected to the server clientID = unique number + ipaddress of the client. For example, 1065 + 192.157.98.70

out string[] status

Name	Type	Direction	Description
status	string array	OUT	The list of status messages generated during the run

out string clientID

Name	Type	Direction	Description
clientID	string	OUT	Identifier of the client that is connected to the server clientID = unique number + ipaddress of the client. For example, 1065 + 192.157.98.70

out string caption

Name	Type	Direction	Description
caption	string	OUT	The wait state or error state message sent to you

out string message

Name	Type	Direction	Description
message	string	OUT	The wait state/error state message sent to you

out string[] button Texts

Name	Type	Direction	Description
button Texts	string array	OUT	An array of strings containing the possible response types that you can send

out string clientID

Name	Type	Direction	Description
clientID	string	OUT	Identifier of the client that is connected to the server clientID = unique number + ipaddress of the client. For example, 1065 + 192.157.98.70

out string caption

Name	Type	Direction	Description
caption	string	OUT	The wait state or error state message sent to you

out string message

Name	Type	Direction	Description
message	string	OUT	The wait state/error state message sent to you

string response

Name	Type	Direction	Description
response	string	IN	A string containing the response type that you can select (it must be one of the strings in the string array buttonTexts)

Ready: Test configured and ready to start

Running: Test running

Paused: Test paused

Wait: A popup that needs your inputs

Error: An error is occured

NOTE. *The Fail condition for PI commands occurs in any of the following cases:*

The server is LOCKED and the message displayed is "Server is locked by another client".

The session is UNLOCKED and the message displayed is "Lock Session to execute the command".

The server is NOTFOUND and the message displayed is "Server not found...Disconnect!".

When none of these fail conditions occur, then the message displayed is "Failed...".

After the test is complete

Command name	Parameters	Description	Return value	Example
GetPassFailStatus()	string clientID string device string suite string test	This method gets the pass or fail status of the measurement after test completion. ***NOTE.*** *Execute this command after completing the measurement.*	String that gives the status of the operation after it has been performed Returns the pass or fail status in the form of a string	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnval=m_Client.GetPassFailStatus(clientID, device, suite, ”testname”) //Pass or Fail
GetResultsValue()	string clientID string device string suite string test string parameterString	This method gets the result values of the measurement after the run.	String that gives the status of the operation after it has been performed Returns the result value in the form of a string	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as srting returnval=m_Client.GetResultsValue(clientID,"Receiver", "HS", "2.1.1 - Differential DC Input Voltage Amplitude Tolerance (VDIF-DC-RX)", "Bit Error")

Command name	Parameters	Description	Return value	Example

GetResultsValueForSubMeasurements()	string clientID
	string device
	string suite
	string test
	string parameterString
	int rowNr
		This method gets the result values for individual sub-measurements after the run.	String that gives the status of the operation after it has been performed
			Returns the result value in the form of a string
				m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.
				returnval as string
				2.1.1 - Differential DC Input Voltage Amplitude Tolerance (VDIF-DC-RX)
				returnval=m_Client.GetResultsValueForSubMeasurements(clientID, "Receiver", "HS", "2.1.1 - Differential DC Input Voltage Amplitude Tolerance (VDIF-DC-RX)", "Measured Value", 0
				returnval=m_Client.GetResultsValueForSubMeasurements(clientID, "Receiver", "HS", "2.1.1 - Differential DC Input Voltage Amplitude Tolerance (VDIF-DC-RX)", "Measured Value", 1

Command name	Parameters	Description	Return value	Example

GetReportParameter()	string clientID
	string device
	string suite
	string test
	string parameterString
		This method gets the general report details such as oscilloscope model, TekExpress version, and M-PHY Rx version.	The return value is the oscilloscope model, TekExpress version, and M-PHY Rx version.	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.
				returnval as string
				Oscilloscope Model
				returnval=m_Client.GetReportParameter(clientID,"Scope Model")
				TekExpress Version
				returnval=m_Client.GetReportParameter(clientID,"TekExpress Version")
				M-PHY Rx Version
				returnval=m_Client.GetReportParameter(clientID,"Application Version")

Command name	Parameters	Description	Return value	Example

TransferReport()	string clientID
	string filePath
		This method transfers the report generated after the run.
		The report contains the summary of the run.
		The client must provide the location where the report is to be saved at the client-end.
			String that gives the status of the operation after it has been performed
			Transfers all the result values in the form of a string
				m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.
				returnval as string
				returnval=m_Client.TransferReport(clientID,"C:\Report")

TransferImages()	string clientID od
	string filePath
		This method transfers all the images (screen shots) from the folder for the current run (for a given suite or measurement).
		NOTE. *Every time you click Start, a folder is created in the X: drive. Transfer the waveforms before clicking Start .*
			String that gives the status of the operation after it has been performed
			Transfers all the images in the form of a string
				m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.
				returnval as string
				returnval=m_Client.TransferImages(clientID, "C:\Waveforms")

out string clientID

Name	Type	Direction	Description
clientID	string	OUT	Identifier of the client that is connected to the server clientID = unique number + ipaddress of the client. For example, 1065 + 192.157.98.70

string device

Name	Type	Direction	Description
device	string	IN	Specifies the name of the device

string suite

Name	Type	Direction	Description
suite	string	IN	Specifies the name of the suite

string test

Name	Type	Direction	Description
test	string	IN	Specifies the name of the test to obtain the pass or fail status

out string clientID

Name	Type	Direction	Description
clientID	string	OUT	Identifier of the client that is connected to the server clientID = unique number + ipaddress of the client. For example, 1065 + 192.157.98.70

string device

Name	Type	Direction	Description
device	string	IN	Specifies the name of the device

string suite

Name	Type	Direction	Description
suite	string	IN	Specifies the name of the suite

string test

Name	Type	Direction	Description
test	string	IN	Specifies the name of the test to obtain the pass or fail status

string parameterString

Name	Type	Direction	Description
parameterString	string	IN	Selects or deselects a test

out string clientID

Name	Type	Direction	Description
clientID	string	OUT	Identifier of the client that is connected to the server clientID = unique number + ipaddress of the client. For example, 1065 + 192.157.98.70

string device

Name	Type	Direction	Description
device	string	IN	Specifies the name of the device

string suite

Name	Type	Direction	Description
suite	string	IN	Specifies the name of the suite

string test

Name	Type	Direction	Description
test	string	IN	Specifies the name of the test to obtain the pass or fail status

string parameterString

Name	Type	Direction	Description
parameterString	string	IN	Selects or deselects a test

int rowNr

Name	Type	Direction	Description
rowNr	int	IN	Specifies the zero based row index of the sub-measurement for obtaining the result value

out string clientID

Name	Type	Direction	Description
clientID	string	OUT	Identifier of the client that is connected to the server clientID = unique number + ipaddress of the client. For example, 1065 + 192.157.98.70

string device

Name	Type	Direction	Description
device	string	IN	Specifies the name of the device

string suite

Name	Type	Direction	Description
suite	string	IN	Specifies the name of the suite

string test

Name	Type	Direction	Description
test	string	IN	Specifies the name of the test to obtain the pass or fail status

string parameterString

Name	Type	Direction	Description
parameterString	string	IN	Specifies the oscilloscope model, TekExpress version, and M-PHY Rx version

out string clientID

Name	Type	Direction	Description
clientID	string	OUT	Identifier of the client that is connected to the server clientID = unique number + ipaddress of the client. For example, 1065 + 192.157.98.70

string filePath

Name	Type	Direction	Description
filePath	string	IN	The location where the report must be saved in the client

NOTE. *If the client does not provide the location to save the report, the report is saved at C:\ProgramFiles.*

out string clientID

Name	Type	Direction	Description
clientID	string	OUT	Identifier of the client that is connected to the server clientID = unique number + ipaddress of the client. For example, 1065 + 192.157.98.70

string filePath

Name	Type	Direction	Description
filePath	string	IN	The location where the report must be saved in the client

NOTE. *If the client does not provide the location to save the report, the report is saved at C:\ProgramFiles.*

NOTE. *The Fail condition for PI commands occurs in any of the following cases:*

The server is LOCKED and the message displayed is "Server is locked by another client".

The session is UNLOCKED and the message displayed is "Lock Session to execute the command".

The server is NOTFOUND and the message displayed is "Server not found...Disconnect!".

When none of these fail conditions occur, then the message displayed is "Failed...".

Save, recall, or check if a session is saved

Command name	Parameters	Description	Return value	Example

CheckSessionSaved()	string clientID
	out bool saved
		This method checks whether the current session is saved.	Return value is either True or False	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.
				returnval as string
				returnval=m_Client.CheckSessionSaved(m_clientID, out savedStatus)

RecallSession()	string clientID
	string name
		Recalls a saved session. The client provides the session name.	String that gives the status of the operation after it has been performed
			The return value is “Session Recalled...”
				m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.
				returnval as string
				returnval=m_Client.RecallSession(clientID, savedSessionName)

SaveSession()	string clientID
	string name
		Saves the current session. The client provides the session name.	String that gives the status of the operation after it has been performed
			The return value is "Session Saved..."/"Failed..."
				m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.
				returnval as string
				returnval=m_Client.SaveSession(clientID, desiredSessionName)

SaveSessionAs()	string clientID
	string name
		Saves the current session under a different name every time this method is called. The client provides the session name.	String that gives the status of the operation after it has been performed
			The return value is "Session Saved..."
				m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.
				returnval as string
				returnval=m_Client.SaveSessionAs(clientID, desiredSessionName)

out string clientID

Name	Type	Direction	Description
clientID	string	OUT	Identifier of the client that is connected to the server clientID = unique number + ipaddress of the client. For example, 1065 + 192.157.98.70

out bool saved

Name	Type	Direction	Description
saved	bool	OUT	Boolean representing whether the current session is saved

This parameter is used as a check in SaveSession() and SaveSessionAs() functions.

out string clientID

Name	Type	Direction	Description
clientID	string	OUT	Identifier of the client that is connected to the server clientID = unique number + ipaddress of the client. For example, 1065 + 192.157.98.70

string name

Name	Type	Direction	Description
name	string	IN	The name of the session being recalled

The name parameter cannot be empty. If it is empty, the client is prompted to provide a valid name.

out string clientID

Name	Type	Direction	Description
clientID	string	OUT	Identifier of the client that is connected to the server clientID = unique number + ipaddress of the client. For example, 1065 + 192.157.98.70

string name

Name	Type	Direction	Description
name	string	IN	The name of the session being saved

The name parameter cannot be empty. If it is empty, the client is prompted to provide a valid name.

Once the session is saved under 'name' you cannot use this method to save the session in a different name. Use SaveSessionAs instead.

out string clientID

Name	Type	Direction	Description
clientID	string	OUT	Identifier of the client that is connected to the server clientID = unique number + ipaddress of the client. For example, 1065 + 192.157.98.70

string name

Name	Type	Direction	Description
name	string	IN	The name of the session being recalled

The same session is saved under different names using this method. The name parameter cannot be empty. If it is empty, the client is prompted to provide a valid name.

NOTE. *The Fail condition for PI commands occurs in any of the following cases:*

The server is LOCKED and the message displayed is "Server is locked by another client".

The session is UNLOCKED and the message displayed is "Lock Session to execute the command".

The server is NOTFOUND and the message displayed is "Server not found...Disconnect!".

When none of these fail conditions occur, then the message displayed is "Failed...".

Unlock the server

Command name	Parameters	Description	Return value	Example
UnlockSession()	string clientID	This method unlocks the server from the client. The ID of the client to be unlocked must be provided. Note	String that gives the status of the operation after it has been performed The return value is "Session Un-Locked..."	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnval=m_Client.UnlockServer(clientID)

out string clientID

Name	Type	Direction	Description
clientID	string	OUT	Identifier of the client that is connected to the server clientID = unique number + ipaddress of the client. For example, 1065 + 192.157.98.70

NOTE. *When the client is disconnected, the client is unlocked automatically.*

NOTE. *The Fail condition for PI commands occurs in any of the following cases:*

The server is LOCKED and the message displayed is "Server is locked by another client".

The session is UNLOCKED and the message displayed is "Lock Session to execute the command".

The server is NOTFOUND and the message displayed is "Server not found...Disconnect!".

When none of these fail conditions occur, then the message displayed is "Failed...".

Disconnect from the server

Command name	Parameters	Description	Return value	Example

Disconnect()	string clientID	This method disconnects the client from the server. Note	Integer value that gives the status of the operation after it has been performed
			1 for Success
			–1 for Failure
				m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.
				returnval as string
				returnval=m_Client.Disconnect(m_clientID)

out string clientID

Name	Type	Direction	Description
clientID	string	OUT	Identifier of the client that is connected to the server clientID = unique number + ipaddress of the client. For example, 1065 + 192.157.98.70

NOTE. *When the client is disconnected, it is unlocked from the server and then disconnected. The ID is reused.*

NOTE. *The Fail condition for PI commands occurs in any of the following cases:*

The server is LOCKED and the message displayed is "Server is locked by another client".

The session is UNLOCKED and the message displayed is "Lock Session to execute the command".

The server is NOTFOUND and the message displayed is "Server not found...Disconnect!".

When none of these fail conditions occur, then the message displayed is "Failed...".

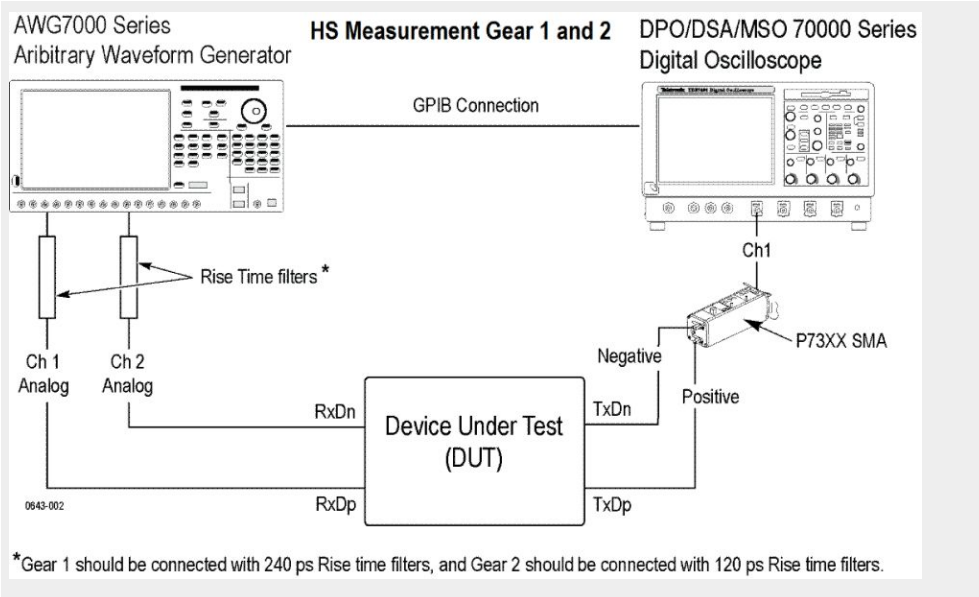
HS testing procedures and examples

Receiver testing measurement procedure

Tests are conducted with 8b/10b encoded continuous CJTPAT.

The general procedure for executing receiver tests is as follows:

1. Connect CH1 and CH2 of the AWG to the RxDp and RxDn pins of the DUT using rise time filters and a matched pair of SMA cables. For guidance, see the connection diagram.



2. Connect the TxDp and TxDn pins of the DUT to the oscilloscope using a matched pair of SMA cables and a differential probe.
3. Set the DUT in loopback mode: Configure the DUT so that its Rx port and Tx port have the same data transfer mode, signaling type, gear, and drive strength settings.
4. For generating a signal using the AWG, generate the MIPI M-PHY single-ended signals with the characteristics given in the Signal Characteristics tables for each test.
5. After the oscilloscope synchronizes to the signal and loopback is validated, the TekScope error detector script runs. The script validates the received bits and reports the number of bit errors and the BER.

See also *Loopback validation*

High-Speed mode measurements

The following are the parameters for high-speed mode measurements:

- Pattern Differential Amplitude: Low, Average, High
- Common-Mode Amplitude: Low, Average, High

NOTE. *For tests that include the Average Common Mode Voltage or the Average Differential Voltage, you can omit the Average parameter by clearing the check box in the test Measurement configuration settings.*

- 8b/10b encoded: YES or NO
- Sinusoidal Jitter:
  - Frequency: Absolute or range
  - Amplitude: Absolute value in terms of UI

Unit Interval

- PREPARE length: Absolute value in terms of SI

Symbol Interval

- STALL length: Absolute value in terms of SI

  For each test, these parameters are set to the default measurement values for Compliance Mode. These parameters cannot be changed in Compliance Mode. For specific parameters for each test, refer to the individual test topic.

See also *Receiver testing measurement procedure*

Test pattern details for continuous mode

The pattern is CJTPAT without PREPARE and STALL regions.

Test pattern details for burst mode

For all the tests (unless noted otherwise) in this mode, the pattern is CJTPAT with PREPARE and STALL regions of 14 SI (Symbol Intervals).

For Test - HS-RX PREPARE Length Capability Verification will be done using a waveform with prepare length based on the user input, based on the DUT capability register.

Test procedure

Unless explicitly stated, for all HS tests use the following test procedure:

1. The test procedure involves receive 3E10 observable bits from the DUT loopback Tx output, and count errors.
2. The measurement is performed on all selected Gears/rate series.
3. Test are applicable only for the Terminated case.
4. The bit errors are counted by Scope Error Detector or using a custom script written by the user. The user can read the error value and enter it manually through a pop-up.

Loopback validation

Use this test to verify that DUT loopback has occurred successfully and that the DUT and the oscilloscope error detector are synchronized. Clean signals and with no impairments are used for this validation. If the DUT loopback passes, then the test solution executes tests.

Table 14: Parameters

Parameter name	Value	Min	Max
Differential voltage – Low (V)	0.06	0.06	0.245
Differential voltage – High (V)	0.244	0.06	0.245
Common-mode voltage (V)	0.18	0.025	0.33
Test time duration (sec)	10	1	120
HS_PREPARE_length	30	0	15
STALL length	30	0	250

Table 15: Oscilloscope error detector settings

Parameter	Value
Error Detector Type	Bit
Error Detector Standard	ANY8B10B
Synchronization Pattern	K28.5
Data Rate	Gear 1A/B
Error Limit	1e-10

See also *Receiver testing measurement procedure*

About the oscilloscope error detector

Performing error detection using the oscilloscope requires the following:

- A multi-channel, programmable lab-grade signal source capable of generating appropriate High Speed and LP signaling.
- A *compatible oscilloscope* with ERRDET and STU options enabled.
- A DUT that supports loopback mode.
- Error Detector (ERRDT) and Serial Trigger Unit (STU) options enabled on the oscilloscope. If needed, *you can verify that these options are enabled*.

TekExpress M-PHY Rx uses the oscilloscope-based bit error detector. This bit error detector requires two input parameters: the bit sequence (SOF), which it uses for synchronization, and the bit length. Bit length specifies the number of bits that need to be stored in memory and later used for comparison and error detection. The synchronization pattern does not have to be at the beginning of the signal test pattern, just somewhere in the signal test pattern, because the fixed-length signal test pattern is sent repeatedly by the signal generator.

2.1.1 HS-RX differential DC input voltage amplitude tolerance (V_{DIF-RX}) test

This test verifies that the high-speed receiver is able to successfully receive high-speed signaling that meets the maximum and minimum conformance requirements for Differential Input Voltage Amplitude (V_{DIF_RX}). This measurement is performed for both large and small amplitude drive strengths.

Measurement parameters

The following table lists parameters for the test. Unless specified, these parameters cannot be configured in Compliance Mode.

Table 16: Parameters for test 2.1.1 (V_{DIF-RX})

Parameter name	Value	Min	Max
Differential amplitude - T1 (mV)	245	60	490
Common mode amplitude - T1 (mV)	150	25	330
Differential amplitude - T2 (mV)	60	60	490
Common mode amplitude - T2 (mV)	150	25	330
HS_PREPARE_length	1	0	15
STALL length	30	0	250
Test time duration	25	1	120

Test cases. The CTS defines two test cases for this test, with the following Rx Common-Mode Voltage and Rx Differential Voltage parameters:

Table 17: Differential DC input voltage amplitude tolerance ($V_{DIF-DC-HS-RX}$) test cases

Test case	Rx Common-Mode Voltage (V_{CM-RX})	Rx Differential Voltage (V_{DIF-RX})
1	150 mV	245 mV
2	150 mV	60 mV

Oscilloscope error detector settings

Table 18: Oscilloscope error detector settings for test 2.1.1 (V_{DIF-RX})

Parameter	Value
Error Detector Type	Bit
Error Detector Standard	ANY8B10B
Data Rate	Gear 1A/B, 2A/B, and 3A/B
Error Limit	1e-10

See also

Common test parameters

Receiver testing measurement procedure

About the oscilloscope error detector

2.1.2 HS-RX accumulated diff input voltage ($V_{DIF-ACC-HS-RX}$) test

This test verifies that the accumulated differential input voltage ($V_{DIF-ACC-HS-RX}$) of the high-speed receiver (HS-RX) is within conformance limits.

Measurement parameters

The following table lists parameters for the test. Unless specified, only Gear 1 type, Data Rate, and Test Time Duration parameters can be configured in Compliance Mode.

Table 19: Parameters for test 2.1.2 ($V_{DIF-ACC-HS-RX}$)

Parameter name	Value	Min	Max
Differential amplitude - T1 (mV)	200	60	490
Common mode amplitude - T1 (mV)	150	25	330
ISI	0.2	0	0.4
HS_PREPARE_length	1	0	15
STALL length	30	0	250
Test time duration	25	1	120

This test is performed with an ISI of only 0.2 UI and no additional jitter. For this test, an appropriate Rx Differential Voltage is chosen to meet the 40 mV of Accumulated Differential Input Voltage.

Table 20: Accumulated differential input voltage tolerance ($V_{DIF-ACC-HS-RX}$) test case

Test case	Rx common-mode voltage (V_{CM-RX})	Accumulated differential Rx input voltage ($V_{DIF-ACC-RX}$)
1	150 mV	40 mV

Oscilloscope error detector settings

Table 21: Oscilloscope error detector settings for test 2.1.2 ($V_{DIF-ACC-HS-RX}$)

Parameter	Value
Error Detector Type	Bit
Error Detector Standard	ANY8B10B
Data Rate	Gear 1A/B, 2A/B, and 3A/B
Error Limit	1e-10

See also *Common test parameters*

Receiver testing measurement procedure

About the oscilloscope error detector

2.1.3 HS-RX Common-Mode input voltage tolerance (V_{CM-RX}) test

This test verifies that the high-speed receiver (HS-RX) is able to successfully receive high-speed signaling that meets the maximum and minimum conformance requirements for common-mode voltage amplitude.

Measurement parameters

The following table lists parameters for the test. Unless specified, only the Gear 1 type, Data Rate, Average amplitude setting, and Test Time Duration parameters can be configured in Compliance Mode.

Table 22: Parameters for test 2.1.3 (V_{CM-RX})

Parameter name	Value	Min	Max
Differential amplitude - T1 (mV)	150	60	490
Common mode amplitude - T1 (mV)	330	25	330
Differential amplitude - T2 (mV)	150	60	490
Common mode amplitude - T2 (mV)	25	25	330
Differential amplitude - T3 (mV)	245	60	490
Common mode amplitude - T3 (mV)	330	25	330
Differential amplitude - T4 (mV)	60	60	490
Common mode amplitude - T4 (mV)	25	25	330
HS_PREPARE_length	1	0	15
STALL length	30	0	250
Test time duration	25	1	120

Test cases. The CTS defines four test cases for this test, with the following Rx Common-Mode Voltage and Rx Differential Voltage parameters:

Table 23: Common-mode input voltage tolerance (V_{CM-RX}) test cases

Test case	Rx Common-Mode Voltage (V_{CM-RX})	Rx Differential Voltage (V_{DIF-RX})
1	330 mV	150 mV
2	25 mV	150 mV
3	330 mV	245 mV
4	25 mV	60 mV

Oscilloscope error detector settings

Table 24: Oscilloscope error detector settings for test 2.1.3 (V_{CM-RX})

Parameter	Value
Error Detector Type	Bit
Error Detector Standard	ANY8B10B
Data Rate	Gear 1A/B, 2A/B, and 3A/B
Error Limit	1e-10

See also

Common test parameters

Receiver testing measurement procedure

About the oscilloscope error detector

2.1.4 HS-RX differential termination enable time ($T_{TERM-ON-HS-RX}$) test

This test verifies the HS-RX switch to termination within the time period specified in the capability attribute - RX_HS_Gx_PREPARE_LENGTH_Capability.

Measurement parameters

The following table lists parameters for the test. Unless specified, only the Gear 1 type, Data Rate, Average amplitude setting, and Test Time Duration parameters can be configured in Compliance Mode.

Table 25: Parameters for test 2.1.4 ($V_{TERM-ON-HS-RX}$)

Parameter name	Value	Min	Max
Differential amplitude - T1 (mV)	200	60	490
Common mode amplitude - T1 (mV)	200	25	330
HS_PREPARE_length	15	0	15
STALL length	30	0	250

Parameter name	Value	Min	Max
Test time duration	25	1	120
RX_HS_Gx_PREPARE_LENGTH_Capability	15	0	15

The following additional parameters are used for the test:

- Burst-Mode CJTPAT signal with a maximum allowed prepare length is used for this test.
- Nominal Rx common-mode voltage and Rx differential voltage is used (V_{CM-TX} = 200 mV, V_{DIF-TX} = 200 mV).
- RX_HS_Gx_PREPARE_LENGTH_Capability: This register information about the DUT needs to be specified in the application.

Test procedure

The following procedure is used for the test:

1. Using a high-impedance probe at the DUT's RX pins (or as close to the pins as possible), capture the burst-mode signal and measure the $T_{TERM-ON-HS-RX}$ value.
2. Identify RX_HS_Gx_PREPARE_LENGTH_Capability prior to the measurement. Convert this into seconds (for example, PREPARE_sec).
3. Over at least one burst Measure $T_{TERM-ON-HS-RX}$.
4. Maximum $T_{TERM-ON-HS-RX}$ value must be less than PREPARE_sec.

Oscilloscope error detector settings

Table 26: Oscilloscope error detector settings for test 2.1.4 ($V_{TERM-ON-HS-RX}$)

Parameter	Value
Error Detector Type	Bit
Error Detector Standard	ANY8B10B
Data Rate	Gear 1A/B, 2A/B, and 3A/B
Error Limit	1e-10

See also *Common test parameters*

Receiver testing measurement procedure

About the oscilloscope error detector

2.1.5 HS-RX differential termination disable time ($T_{TERM-OFF-HS-RX}$) test

This test verifies the HS-RX disable termination within the time period specified in its related capability attribute.

Measurement parameters

The following table lists parameters for the test. Unless specified, only the Gear 1 type, Data Rate, Average amplitude setting, and Test Time Duration parameters can be configured in Compliance Mode.

Table 27: Parameters for test 2.1.5 ($V_{TERM-OFF-HS-RX}$)

Parameter name	Value	Min	Max
Differential amplitude - T1 (mV)	200	60	490
Common mode amplitude - T1 (mV)	200	25	330
STALL length	15	0	250
Test time duration	25	1	120
RX_Min_STALL_NoConfig_Time_Capability	255	1	255

The following additional parameters are used for this test:

- Burst-Mode CJTPAT signal with a maximum allowed prepare length is used for this test.
- Nominal Rx common-mode voltage and Rx differential voltage is used (V_{CM-TX} = 200 mV, V_{DIF-TX} = 200 mV).
- RX_Min_STALL_NoConfig_Time_Capability: This register information about the DUT needs to be specified in the application.

Test procedure

The following procedure is used for the test:

1. Using a high-impedance probe at the DUT's RX pins (or as close to the pins as possible), capture the burst-mode signal and measure the $T_{TERM-OFF-HS-RX}$ value.
2. Identify RX_Min_STALL_NoConfig_Time_Capability prior to the measurement. Convert this into seconds (for example, PREPARE_sec).
3. Over at least one burst Measure $T_{TERM-OFF-HS-RX}$.
4. Maximum $T_{TERM-OFF-HS-RX}$ value must be less than PREPARE_sec.

Oscilloscope error detector settings

Table 28: Oscilloscope error detector settings for test 2.1.5 ($V_{TERM-OFF-HS-RX}$)

Parameter	Value
Error Detector Type	Bit
Error Detector Standard	ANY8B10B
Data Rate	Gear 1A/B, 2A/B, and 3A/B
Error Limit	1e-10

See also

Common test parameters

Receiver testing measurement procedure

About the oscilloscope error detector

2.1.7 HS-RX receiver jitter tolerance (TJ_{RX}, DJ_{RX}, RJ_{RX}, $STTJ_{RX}$, $STDJ_{RX}$) test

This test verifies that the high-speed receiver is able to successfully receive high-speed signaling that meets the conformance requirements of both low-frequency and short-term jitter.

Measurement parameters

The following table lists parameters for the test. Unless specified, only the Gear 1 type, Data Rate, and Test Time Duration parameters can be configured in Compliance Mode.

Table 29: Parameters for test 2.1.7 (TJ_{RX}, DJ_{RX}, RJ_{RX}, $STTJ_{RX}$, $STDJ_{RX}$)

Parameter name	Value	Min	Max
Differential amplitude - T1 (mV)	150	60	490
Common mode amplitude - T1 (mV)	25	25	330
STRJ (UI)	0	0	0.5
STDJ (UI)	0.2	0	0.5
DJ (UI)	0.35	0	0.5
Test time duration	25	1	120
Frequency f_SJ1	240	0.1	250
Frequency f_SJ2	0.4	0.1	250
Frequency f_SJ3	10	0.1	250
Frequency f_SJ4	91	0.1	250

The following additional parameters are used with this test:

- The following is the RX jitter parameter that needs to be met to test the tolerance:

Table 30: Parameters for RX jitter

Parameter	Max value
Transmitter Total Jitter (TJ_{RX})	0.52 UI
Short Term Total Jitter ($STTJ_{RX}$)	0.30 UI
Deterministic Jitter (DJ_{RX})	0.35 UI
Short Term Deterministic Jitter ($STDJ_{RX}$)	0.30 UI

- A Rx differential voltage is chosen to meet the 40 mV of accumulated differential input voltage.
- Rx common-mode voltage is set to 25 mV.
- To meet the deterministic jitter values, sinusoidal jitter at a maximum of four frequencies can be used.

Test procedure

The following procedure is used for the test:

1. The jitter parameters are specified at the Rx pins. So taking into account the traces on the board up to the pin, a calibrated waveform will used for this test.
2. The number of received bit errors should be zero.

Oscilloscope error detector settings

Table 31: Oscilloscope error detector settings for test 2.1.7 (TJ_{RX}, DJ_{RX}, RJ_{RX}, $STTJ_{RX}$, $STDJ_{RX}$)

Parameter	Value
Error Detector Type	Bit
Error Detector Standard	ANY8B10B
Data Rate	Gear 1A/B, 2A/B, and 3A/B
Error Limit	1e-10

See also *Common test parameters*

Receiver testing measurement procedure

About the oscilloscope error detector

2.1.8 HS-RX frequency offset tolerance ($f_{OFFSET-RX}$) test

This test verifies the DUT's HS-RX ability to receive HS signalling having the worst-case frequency offset characteristics.

Measurement parameters

The following table lists parameters for the test. Unless specified, only the Gear 1 type, Data Rate, Average amplitude setting, and Test Time Duration parameters can be configured in Compliance Mode.

Table 32: Parameters for test 2.1.8 ($f_{OFFSET-RX}$)

Parameter name	Value	Min	Max
Differential amplitude - T1 (mV)	200	60	490
Common mode amplitude - T1 (mV)	200	25	330
Differential amplitude - T2 (mV)	200	60	490
Common mode amplitude - T2 (mV)	200	25	330
Frequency offset (+/-)	2000	-2000	2000

The following additional parameters are used for the test:

- Nominal Rx common-mode voltage and Rx differential voltage is used (V_{CM-TX} = 200 mV, V_{DIF-TX} = 200 mV).
- Maximum Frequency offset of 2000 ppm is allowed.
- Frequency offset of +2000 ppm and -2000 ppm with respect to the nominal bit rate used in the test.

Oscilloscope error detector settings

Table 33: Oscilloscope error detector settings for test 2.1.8 ($f_{OFFSET-RX}$)

Parameter	Value
Error Detector Type	Bit
Error Detector Standard	ANY8B10B
Data Rate	Gear 1A/B, 2A/B, and 3A/B
Error Limit	1e-10

See also *Common test parameters*

Receiver testing measurement procedure

About the oscilloscope error detector

2.1.9 HS-RX PREPARE length capability verification ($T_{HS-PREPARE-RX}$) test

This test verifies the DUT's HS-RX capablity of receiving bursts as specified in its related capability attribute.

Measurement parameters

The following table lists parameters for the test. Unless specified, only the Gear 1 type, Data Rate, Average amplitude setting, and Test Time Duration parameters can be configured in Compliance Mode.

Table 34: Parameters for test 2.1.9 ($T_{HS-PREPARE-RX}$)

Parameter name	Value	Min	Max
Differential amplitude - T1 (mV)	200	60	490
Common mode amplitude - T1 (mV)	200	25	330
Test time duration	25	1	120
Frequency offset	2000	-2000	2000

The following additional parameters are used for the test:

- Burst-Mode CJTPAT signal with a prepare length in accordance with the DUT's capability register is used for this test.
- Nominal Rx common-mode voltage and Rx differential voltage is used (V_{CM-TX} = 200 mV, V_{DIF-TX} = 200 mV).
- RX_HS_Gn_PREPARE_LENGTH_Capability: This register information about the DUT needs to be specified in the application.
- Frequency offset of +2000 ppm with respect to the nominal bitrate used in the test.

Oscilloscope error detector settings

Table 35: Oscilloscope error detector settings for test 2.1.9 ($T_{HS-PREPARE-RX}$)

Parameter	Value
Error Detector Type	Bit
Error Detector Standard	ANY8B10B
Data Rate	Gear 1A/B, 2A/B, and 3A/B
Error Limit	1e-10

See also

Common test parameters

Receiver testing measurement procedure

About the oscilloscope error detector

2.1.10 HS-RX sync length capability verification ($T_{SYNC-RX}$) test

This test verifies the DUT's HS-RX capability of receiving SYNC length as specified in its related capability attribute.

Measurement parameters

The following table lists parameters for the test. Unless specified, only the Gear 1 type, Data Rate, Average amplitude setting, and Test Time Duration parameters can be configured in Compliance Mode.

Table 36: Parameters for test 2.1.10 ($T_{SYNC-RX}$)

Parameter name	Value	Min	Max
Differential amplitude - T1 (mV)	200	60	490
Common mode amplitude - T1 (mV)	200	25	330
Test time duration	25	1	120
RX_HS_Gx_SYNC_LENGTH_Capability	15	1	255

The following additional parameters are used for the test:

- Burst-Mode CJTPAT signal with a SYNC length in accordance with the DUT's capability register is used for this test.
- Nominal Rx common-mode voltage and Rx differential voltage is used (V_{CM-TX} = 200 mV, V_{DIF-TX} = 200 mV).
- RX_HS_Gx_SYNC_LENGTH_Capability: This register information about the DUT needs to be specified in the application.
- Frequency offset of +2000 ppm with respect to the nominal bitrate used in the test.

Oscilloscope error detector settings

Table 37: Oscilloscope error detector settings for test 2.1.10 ($T_{SYNC-RX}$)

Parameter	Value
Error Detector Type	Bit
Error Detector Standard	ANY8B10B
Data Rate	Gear 1A/B, 2A/B, and 3A/B
Error Limit	1e-10

See also

Common test parameters

Receiver testing measurement procedure

About the oscilloscope error detector

HS margin tests

The HS margin testing suite enables you to check the limit of the HS Receiver. The receiver is stressed using Sinusoidal Jitter tones with varying Amplitudes and Frequencies. This suite provides two Scan methods to configure the Sinusoidal Jitter tones.

Measurement parameters

The following table lists parameters for the test. Unless specified, only the Gear 1 type, Data Rate, Average amplitude setting, and Test Time Duration parameters can be configured in Compliance Mode.

Table 38: Parameters for HS margin test

Parameter name	Value	Min	Max
Scan Method	Linear Pass to Fail	Linear Pass to Fail, Define Scan Parameter Preset	–
Jitter	true, 23, 30, 40, 2 true, 25, 30, 40, 1.5 true, 27, 30, 40, 1	–	–
Start Amplitude	0.01	0	2
End Amplitude	0.15	0	2
Incremental Step Amplitude	0.01	0.01	0.1
Start Frequency	1	1	200
End Frequency	100	1	200
Incremental Step Frequency	10	1	20
Test time duration	25	1	120

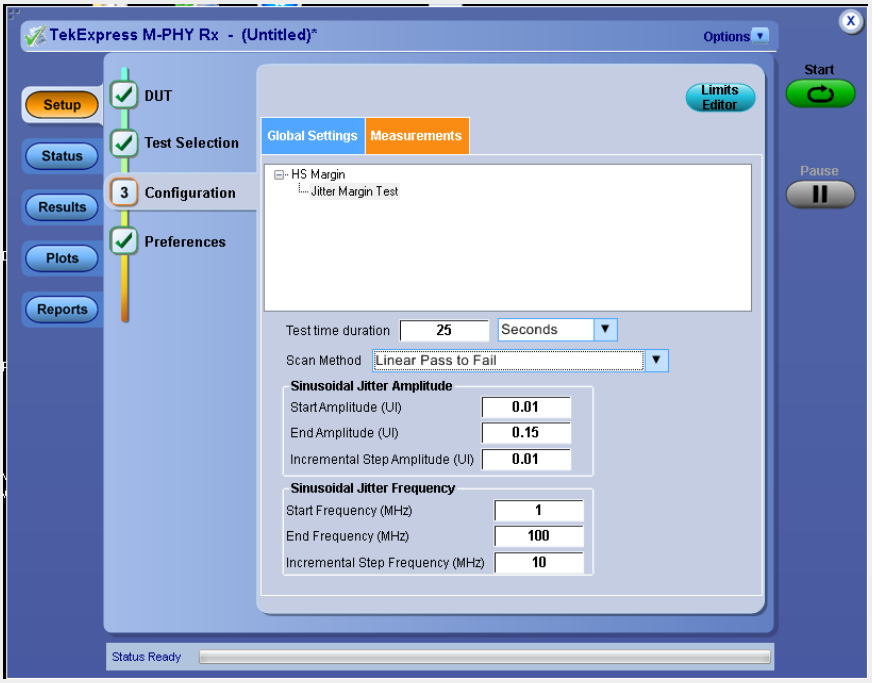
The following additional parameters are used for the test:

- The Sinusoidal jitter amplitude is to be specified in Unit Interval (UI).
- The Sinusoidal jitter Frequency is to be specified in MHz.

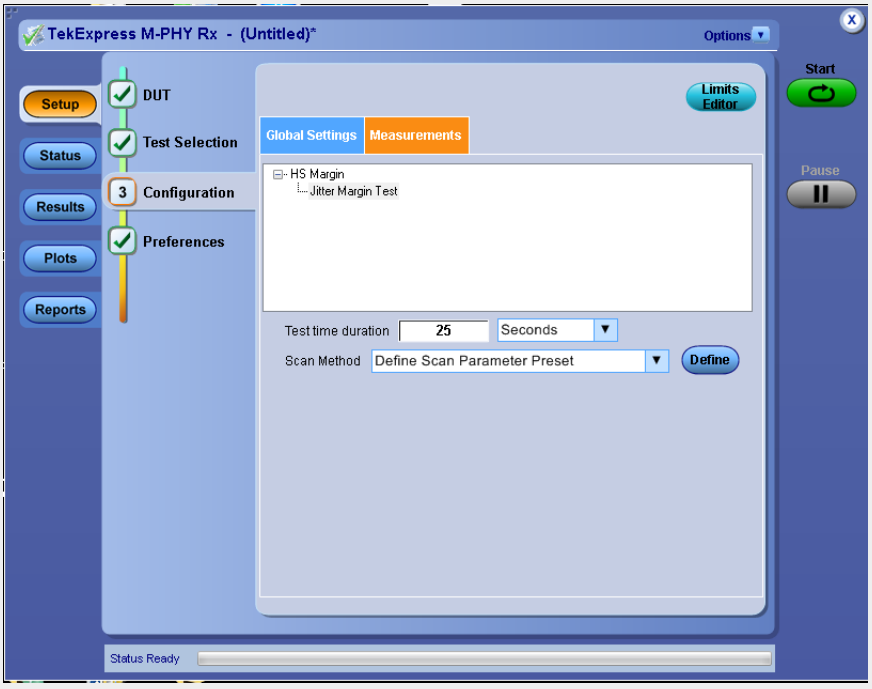
Scan methods

This suite provides two Scan methods to configure the Sinusoidal Jitter tones used for the test. Use the Scan Method drop-down to select between the two methods.

Method 1 - Linear Pass to Fail. Select this method to specify the range of amplitude and frequencies to be used and to specify the incremental step size for both amplitude and frequency. The incremental step setting controls the number of discrete amplitude/frequency points in the test.



Method 2 - Scan parameter preset. Select this method to specify a different amplitude range for each frequency of Sinusoidal Jitter as shown below.



Oscilloscope error detector settings

Table 39: Oscilloscope error detector settings for HS margin test

Parameter	Value
Error Detector Type	Bit
Error Detector Standard	ANY8B10B
Data Rate	Gear 1A/B, 2A/B, and 3A/B
Error Limit	1e-10

See also

Common test parameters

Receiver testing measurement procedure

About the oscilloscope error detector

PWM testing procedures and examples

Test pattern details for continuous mode

The pattern is CJTPAT without PREPARE and STALL regions.

Test pattern details for burst mode

For all the tests in this mode, the pattern is CJTPAT with PREPARE and STALL regions of 14 SI (Symbol Intervals).

Test procedure

For all PWM tests, use the following test procedure:

1. The measurement is performed for All Gears/rate series/Lanes.
2. DUT needs to support Internal BER (IBER) counter. The mechanism to read this internal error counter is not included in this software.
3. DUTs which implement only loopback mode of receiver testing cannot be addressed with this software since the oscilloscope error detector does not support error detection on PWM signaling. In this case, an alternative error detection mechanisms or devices could be employed.

2.2.1 PWM-RX differential DC input voltage amplitude tolerance ($V_{DIF-DC-PWM-RX}$) test

This test verifies the maximum and minimum conformance requirements for differential DC input voltage amplitude at the M-RX pins.

Measurement parameters

The following table lists parameters for the test. Unless specified, only the Gear 1 type, Data Rate, Average amplitude setting, and Test Time Duration parameters can be configured in Compliance Mode.

Table 40: Parameters for test 2.2.1 ($V_{DIF-DC-PWM-RX}$)

Parameter name	Value	Min	Max
Differential amplitude - T1 (mV)	490	60	490
Common mode amplitude - T1 (mV)	150	25	330
Differential amplitude - T2 (mV)	120	60	490
Common mode amplitude - T2 (mV)	150	25	330
Test time duration	25	1	120

Test cases. The CTS defines two test cases for this test, with the following Rx Common-Mode Voltage and Rx Differential Voltage parameters.

Table 41: Differential DC input voltage amplitude tolerance ($V_{DIF-DC-PWM-RX}$) test cases

Test case	Rx Common-Mode Voltage (V_{CM-RX})	Rx Differential Voltage (V_{DIF-RX})
1	150 mV	490 mV [1]
2	150 mV	120 mV

Oscilloscope error detector settings

[1] While using the non-interleave output, the maximum of 490 mV with a common mode of 150 mV will not be achievable. As a work-around, try the test with a lower common mode voltage. Otherwise, use bias-tee for common mode.

Table 42: Oscilloscope error detector settings for test 2.2.1 ($V_{DIF-DC-PWM-RX}$)

Parameter	Value
Error Detector Type	Bit
Error Detector Standard	ANY8B10B
Data Rate	Gear 1A/B, 2A/B, and 3A/B
Error Limit	1e-10

See also *Common test parameters*

Receiver testing measurement procedure

About the oscilloscope error detector

2.2.2 PWM-RX accumulated differential input voltage tolerance ($V_{DIF-ACC-PWM-RX}$) test

This test verifies that the DUT's M-RX is able to successfully receive HS signaling that meets the minimum Accumulated Differential Input Voltage Amplitude.

Measurement parameters

The following table lists parameters for the test. Unless specified, only the Gear 1 type, Data Rate, Average amplitude setting, and Test Time Duration parameters can be configured in Compliance Mode.

Table 43: Parameters for test 2.2.2 ($V_{DIF-ACC-PWM-RX}$)

Parameter name	Value	Min	Max
Differential amplitude - T1 (mV)	40	60	490
Common mode amplitude - T1 (mV)	150	25	330
ISI	0.2	0	0.4
Test time duration	25	1	120

This test is performed with an ISI of only 0.2 UI and no additional jitter. For this test, an appropriate Rx Differential Voltage is chosen to meet the 40 mV of Accumulated Differential Input Voltage.

Table 44: Accumulated Differential Input Voltage Tolerance ($V_{DIF-ACC-PWM-RX}$) test cases

Test case	Rx Common-Mode Voltage (V_{CM-RX})	Accumulated Differential Rx Input Voltage ($V_{DIF-ACC-RX}$)
1	150 mV	40 mV

Oscilloscope error detector settings

Table 45: Oscilloscope error detector settings for test 2.2.2 ($V_{DIF-ACC-PWM-RX}$)

Parameter	Value
Error Detector Type	Bit
Error Detector Standard	ANY8B10B
Data Rate	Gear 1A/B, 2A/B, and 3A/B
Error Limit	1e-10

See Also

Common test parameters

Receiver testing measurement procedure

About the oscilloscope error detector

2.2.3 PWM-RX Common-Mode input voltage tolerance (V_{CM-RX}) test

This test verifies maximum and minimum conformance requirements for common-mode input voltage amplitude at the M-RX pins.

Measurement parameters

The following table lists parameters for the test. Unless specified, only the Gear 1 type, Data Rate, Average amplitude setting, and Test Time Duration parameters can be configured in Compliance Mode.

Table 46: Parameters for test 2.2.3 (V_{CM-RX})

Parameter name	Value	Min	Max
Differential amplitude - T1 (mV)	490	60	490
Common mode amplitude - T1 (mV)	330	25	330
Differential amplitude - T2 (mV)	120	60	490
Common mode amplitude - T2 (mV)	25	25	330
Test time duration	25	1	120

Test cases. The CTS defines two test cases for this test, with the following Rx Common-Mode Voltage and Rx Differential Voltage parameters.

Table 47: Common-mode input voltage tolerance (V_{CM-RX}) test cases

Test case	Rx Common-Mode Voltage (V_{CM-RX})	Rx Differential Voltage (V_{DIF-RX})
1	330 mV	490 mV [2]
2	25 mV	120 mV

Oscilloscope error detector settings

Table 48: Oscilloscope error detector settings for test 2.2.3 (V_{CM-RX})

Parameter	Value
Error Detector Type	Bit
Error Detector Standard	ANY8B10B
Data Rate	Gear 1A/B, 2A/B, and 3A/B
Error Limit	1e-10

See also

Common test parameters

Receiver testing measurement procedure

About the oscilloscope error detector

[2] While using the non-interleave output, the maximum of 490 mV with a common mode of 150 mV will not be achievable. As a work-around, try the test with a lower common mode voltage. Otherwise, use bias-tee for common mode.

2.2.4 PWM-RX differential termination enable time ($T_{TERM-ON-PWM-RX}$) test

This test verifies the HS-RX switch to termination within the time period specified in the capability attribute - RX_LS_PREPARE_LENGTH_Capability.

Measurement parameters

The following table lists parameters for the test. Unless specified, only the Gear 1 type, Data Rate, Average amplitude setting, and Test Time Duration parameters can be configured in Compliance Mode.

Table 49: Parameters for test 2.2.4 ($T_{TERM-ON-PWM-RX}$)

Parameter name	Value	Min	Max
Differential amplitude - T1 (mV)	200	60	490
Common mode amplitude - T1 (mV)	200	25	330
Test time duration	25	1	120
PWM_PREPARE_length	15	0	15
RX_LS_PREPARE_LENGTH_Capability	15	0	15

The following additional parameters are used for the test:

- Burst-Mode CJTPAT signal with a maximum allowed prepare length is used for this test.
- Nominal Rx Common-Mode Voltage and Rx Differential Voltage is used (V_{CM-TX} = 200 mV, V_{DIF-TX} = 200 mV).
- RX_LS_PREPARE_LENGTH_Capability: This register information about the DUT needs to be specified in the application.

Test procedure

The following procedure is used for the test:

1. Using a high-impedance probe at the DUT's RX pins (or as close to the pins as possible), capture the burst-mode signal and measure the $T_{TERM-ON-PWM-RX}$ value.
2. Identify RX_LS_PREPARE_LENGTH_Capability prior to the measurement. Convert this into seconds (for example, PREPARE_sec).
3. Over at least one burst Measure $T_{TERM-ON-PWM-RX}$.
4. Maximum $T_{TERM-ON-PWM-RX}$ value must be less than PREPARE_sec.

Oscilloscope error detector settings

Table 50: Oscilloscope error detector settings for test 2.2.4 ($T_{TERM-ON-PWM-RX}$)

Parameter	Value
Error Detector Type	Bit
Error Detector Standard	ANY8B10B
Data Rate	Gear 1A/B, 2A/B, and 3A/B
Error Limit	1e-10

See also *Common test parameters*

Receiver testing measurement procedure

About the oscilloscope error detector

2.2.5 PWM-RX differential termination disable time ($T_{TERM-OFF-PWM-RX}$) test

This test verifies the HS-RX disable termination within the time period specified in its related capability attribute - RX_Min_SLEEP_NoConfig_Time_Capability.

Measurement parameters

The following table lists parameters for the test. Unless specified, only the Gear 1 type, Data Rate, Average amplitude setting, and Test Time Duration parameters can be configured in Compliance Mode.

Table 51: Parameters for test 2.2.5 ($T_{TERM-OFF-PWM-RX}$)

Parameter name	Value	Min	Max
Differential amplitude - T1 (mV)	490	60	490
Common mode amplitude - T1 (mV)	150	25	330
Test time duration	25	1	120
STALL length	15	0	15
RX_Min_SLEEP_NoConfig_Time_Capability	15	0	15

The following additional parameters are used for this test:

- Burst-Mode CJTPAT signal with a maximum allowed prepare length is used for this test.
- Nominal Rx common-mode voltage and Rx differential voltage is used (V_{CM-TX} = 200 mV, V_{DIF-TX} = 200 mV).
- RX_Min_SLEEP_NoConfig_Time_Capability: This register information about the DUT needs to be specified in the application.

Test procedure

The following procedure is used for the test:

1. Using a high-impedance probe at the DUT's RX pins (or as close to the pins as possible), capture the burst-mode signal and measure the $T_{TERM-OFF-PWM-RX}$ value.
2. Identify RX_Min_SLEEP_NoConfig_Time_Capability prior to the measurement. Convert this into seconds (for example, PREPARE_sec).
3. Over at least one burst Measure $T_{TERM-OFF-PWM-RX}$.
4. Maximum $T_{TERM-OFF-PWM-RX}$ value must be less than PREPARE_sec.

Oscilloscope error detector settings

Table 52: Oscilloscope error detector settings for test 2.2.5 ($T_{TERM-OFF-PWM-RX}$)

Parameter	Value
Error Detector Type	Bit
Error Detector Standard	ANY8B10B
Data Rate	Gear 1A/B, 2A/B, and 3A/B
Error Limit	1e-10

See also

Common test parameters

Receiver testing measurement procedure

About the oscilloscope error detector

2.2.8 PWM-RX receive ratio PWM-G1 and above (k_{PWM-RX}) test

This test verifies that the Receive Ratio tolerance (k_{PWM-RX}) of the DUT's PWM-RX is within the conformance limits.

Measurement parameters

The following table lists parameters for the test. Unless specified, only the Gear 1 type, Data Rate, Average amplitude setting, and Test Time Duration parameters can be configured in Compliance Mode.

Table 53: Parameters for test 2.2.8 (k_{PWM-RX})

Parameter name	Value	Min	Max
Differential amplitude - T1 (mV)	200	60	490
Common mode amplitude - T1 (mV)	200	25	330
PWM receive ratio	1.5	1.5	3
Test time duration	25	1	120

The following additional parameters are used for this test:

- PWM signal with two receive ratios (0.60/0.40 and 0.75/0.25) are used for this test.
- Nominal Rx common-mode voltage and Rx differential voltage is used (V_{CM-TX} = 200 mV, V_{DIF-TX} = 200 mV).

Oscilloscope error detector settings

Table 54: Oscilloscope error detector settings for test 2.2.8 (k_{PWM-RX})

Parameter	Value
Error Detector Type	Bit
Error Detector Standard	ANY8B10B
Data Rate	Gear 1A/B, 2A/B, and 3A/B
Error Limit	1e-10

See also *Common test parameters*

Receiver testing measurement procedure

About the oscilloscope error detector

2.2.9 PWM-RX receive minor duration in PWM-G0 ($T_{PWM-MINOR-G0-RX}$) test

This test verifies that the Receive Minor Duration tolerance ($T_{PWM-MINOR-G0-RX}$) of the DUT's PWM-RX is within the conformance limits.

Measurement parameters

The following table lists parameters for the test. Unless specified, only the Gear 1 type, Data Rate, Average amplitude setting, and Test Time Duration parameters can be configured in Compliance Mode.

Table 55: Parameters for test 2.2.9 ($T_{PWM-MINOR-G0-RX}$)

Parameter name	Value	Min	Max
Differential amplitude - T1 (mV)	200	60	490
Common mode amplitude - T1 (mV)	200	25	330
PWM Ratio - T1	37	37	111
Test time duration	25	1	120

The following additional parameters are used for this test:

- PWM signal with two Minor durations (37 nsec and 111 nsec) are used for this test.
- Nominal Rx Common-Mode Voltage and Rx Differential Voltage is used (V_{CM-TX} = 200 mV, V_{DIF-TX} = 200 mV).

Oscilloscope error detector settings

Table 56: Oscilloscope error detector settings for test 2.2.9 ($T_{PWM-MINOR-G0-RX}$)

Parameter	Value
Error Detector Type	Bit
Error Detector Standard	ANY8B10B
Data Rate	Gear 1A/B, 2A/B, and 3A/B
Error Limit	1e-10

See also

Common test parameters

Receiver testing measurement procedure

About the oscilloscope error detector

Reference

Map the My TekExpress folder

To map the My TekExpress folder on the instruments, follow these steps:

1. Open Windows Explorer.
2. From the Windows Explorer menu, click **Computer**.
3. In the menu bar, click **Map network drive**.
4. Select the Drive letter as **X:** (if there is any previous connection on X:, disconnect it first through **Tools > Disconnect Network drive** menu of Windows Explorer. Windows 7 users: if you do not see the Tools menu, press the **Alt** key).
5. In the Folder field, enter the remote My TekExpress folder path (for example, \\192.158.97.65\ My TekExpress).
6. Click **Finish**.

To determine the IP address of the instrument where the My TekExpress folder exists, do the following:

1. On the instrument where the My TekExpress folder exists, click **Start** and select **Run**.
2. Type "cmd" and then press **Enter**.
3. At the command prompt, type "ipconfig" and then press **Enter**.

Enable the oscilloscope error detector

The ERRDT and STU options are installed on the oscilloscope as part of the TekScope software but need to be enabled with a license before use.

To enable the error detector options:

1. After purchasing a license, from the TekScope menu select **Utilities > Option Installation**.
2. After the TekScope Options Installation wizard opens, press **F1** key to view step-by-step instructions on using the wizard to install options.

To verify that the ERRDT and STU options are enabled:

1. From the TekScope Help menu, select **About TekScope**.
2. In the Options window, "ERRDET: Error Detector" and "STU: 6.25 Gb/s Serial Pattern Trigger" should appear in the list. If they do not, they are not enabled.

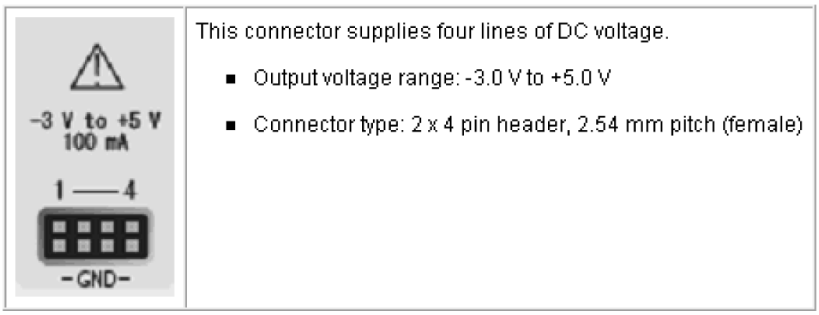
See Also *About the oscilloscope error detector*

Use a bias tee for common mode DC

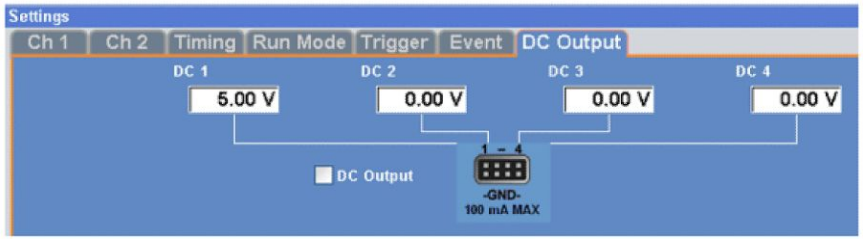
The Bias Tee is used to introduce Common mode DC (for HS Gear 3 testing) when interleave channel outputs are used.

AWG DC Output Connector (front panel)

The Arbitrary Waveform Generator (AWG) has four lines of DC output on the front panel.



Of these four lines, only the first two DC outputs (DC1 and DC2) are used.



Index

I

K

L

M

O

P

Q

R

S

T

U