Tektronix®

# TekExpress Automotive Ethernet (100/1000BASE-T1) Compliance Analysis Solution

## Printable Application Help

077-1460-00

Tektronix®

# TekExpress Automotive Ethernet (100/1000BASE-T1) Compliance Analysis Solution

## Printable Application Help

www.tek.com
077-1460-00

### Contacting Tektronix

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

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

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

# Table of Contents

## Running tests



## Viewing plots



## Viewing results



## Configuring and viewing reports



## Saving and recalling test setups



## Automotive Ethernet measurements

## Measurement error messages



## TekExpress programmatic interface

# Welcome

The TekExpress Automotive Ethernet (100/1000BASE-T1) is a compliance test solution for performing Transmitter electrical specification tests and MDI electrical specification Return Loss test of the transmitter as per OPEN Alliance BroadR-Reach (OABR) specification version 3.2. The IEEE P802.3bw™ D3.3 is for 100 BASE-T1 and IEEE P802.3bp™ is for 1000BASE-T1. These specifications have a stated objective to provide electrical interoperability for 100/1000 Mbps client and the same electrical test requirements to the Physical Medium Attachment (PMA) transmitter electrical specifications.

The Automotive Ethernet (100/1000BASE-T1) is an Ethernet-based point-to-point technology for automotive applications.

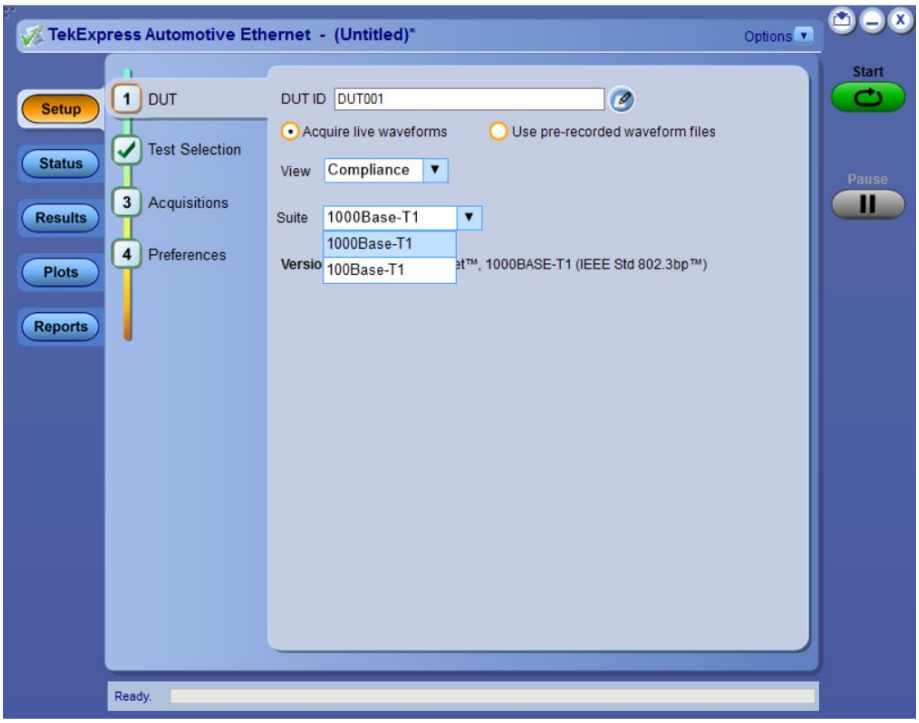



**Figure 1: Setup panel**

## Supported Tests

The following tests are supported for both 100/1000BASE-T1. The Transmitter Timing Jitter-MDI Jitter is a new measurement which is supported for 1000BASE-T1.

- *Transmitter Clock Frequency*
- *Transmitter Timing Jitter-MDI Jitter (1000Base-T1)*
- *Transmitter Timing Jitter Master and Slave*
- *Transmitter Output Droop*
- *Transmitter Power Spectral Density*
- *Transmitter Peak Differential Output*
- *Transmitter Distortion*
- *Return Loss*

## Test modes

**Table 1: Test modes**

| Required test mode | | Automotive Ethernet measurement |
|---|---|---|
| 100BASE-T1 | 1000BASE-T1 | |
| 1 | 6 | Transmitter Output Droop |
| | | Transmitter Timing Jitter |
| 3 | 1 | Transmitter Timing Jitter-Slave Jitter |
| 2 | 1 | Transmitter Timing Jitter-Master Jitter |
| | | Transmitter Clock Frequency and MDI Jitter |
| 2 | 2 | Transmitter Clock Frequency |
| NA | 2 | Transmitter Timing Jitter-MDI Jitter |
| 4 | 4 | Transmitter Distortion<br>Return Loss |
| 5 | 5 | Transmitter Power Spectral Density |
| 5 | 5 | Transmitter Peak Differential Output |

## See also:

*Application Basics*

*Install the Software*

*Application Directories and Usage*

*File Name Extensions*

# Introduction

## Related documentation

The following information is available as part of the TekExpress Automotive Ethernet documentation (100/1000BASE-T1) set.

**Table 2: Related documentation**

| Item | Purpose | Location |
|---|---|---|
| Help | Application operation and User Interface help | Application Help menu |
| PDF of the help | Printable version of the compiled help | PDF file that ships with TekExpress Automotive Ethernet software. Downloadable from *http://www.tek.com/* |

## Abbreviation and conventions

The online help uses the following conventions:

- When steps require a sequence of selections using the software interface, the ">" delimiter marks each transition between a menu and an option. For example, **File > Save**.
- DUT refers to the Device Under Test.
- The terms "waveform" and "signal" are used interchangeably.
- The term AWG refers to a Tektronix Arbitrary Waveform Generator.

# 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, *www.tektronix.com*.

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.
- If possible, save the waveform on which you are performing the measurement as a .wfm file.

# Operating basics

## AWG/AFG automation

AWG/AFG automation is supported in the TekExpress Automotive Ethernet (100/1000BASE-T1) application for the following measurements. Before selecting the AWG/AFG automation feature, ensure that a GPIB connection is established between the oscilloscope and the AWG/AFG. The AFG automation depends on the measurement used for testing.

- *Transmitter Distortion with disturbing signal*: AWG/AFG is used to transmit a disturbing signal. CH1 and CH1_inverted are used.
- *Return Loss measurement and calibration*: AWG/AFG is used to transmit a wide band signal. CH1 and CH1_inverted are used for transmission of the signal. A marker signal has to be connected to the auxiliary channel of the oscilloscope (used as trigger source).

### How Automotive Ethernet automates AWG/AFG

Automation of the AWG/AFG is taken care in the TekExpress Automotive Ethernet application in the following way:

1. Refresh the Instruments Control Settings to observe the connected AWG/AFG through GPIB.

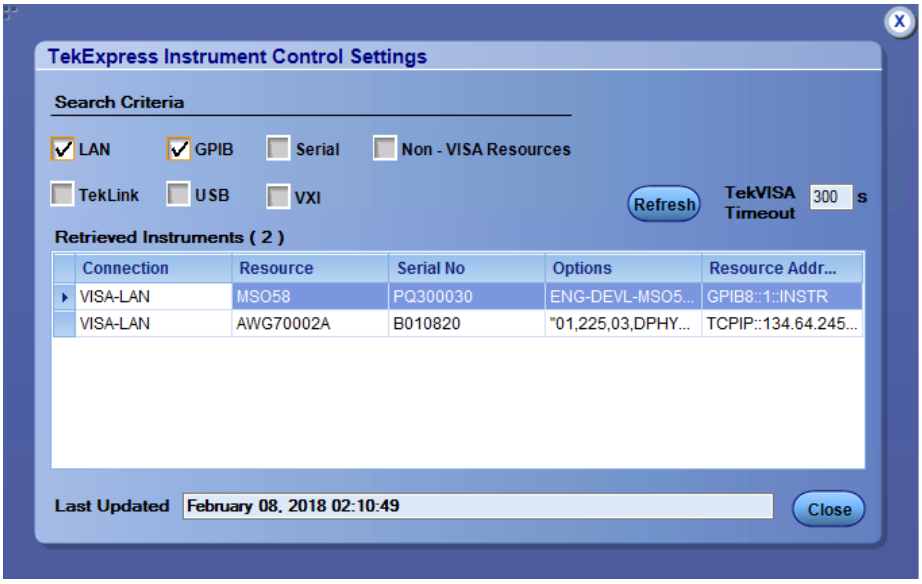



**Figure 2: Instrument Control Settings**

2. Once the AWG/AFG is listed as shown above, go to the Global Settings tab from the Configuration menu of the Tests. If the AWG/AFG is supported, it will be listed as a drop-down menu option next to the Automate with AWG/AFG label. By default, the application will consider the connected AWG to be used for automation.

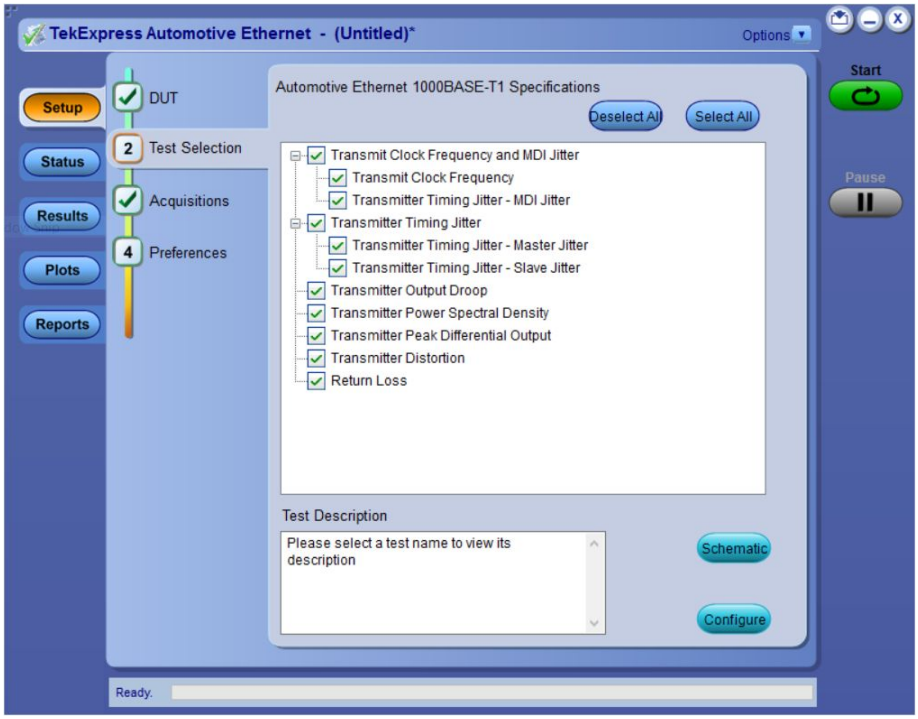



**Figure 3: Test Selection tab**

3. If you do not want to automate the use of an AWG/AFG, select the **Do not use** option in the drop-down menu for that parameter, and then start the testing.

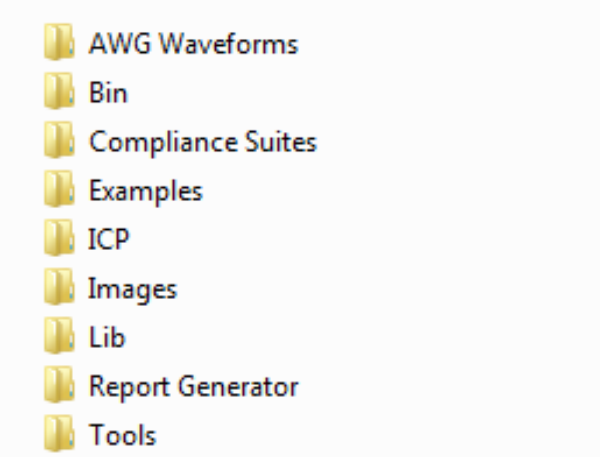
---

***NOTE.*** *If Automate with AWG/AFG is set as Do not use, you will need to manually copy the AWG/AFG waveforms from the oscilloscope to the AWG/AFG. Waveforms are available at C:\Program Files\Tektronix\TekExpress \TekExpress Automotive-Ethernet\AWG Waveforms.*

---

## Application directories and usage

The application directory and associated files are organized in the C:\Program Files\Tektronix\TekExpress\TekExpress Automotive-Ethernet.




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

**Table 3: Application directories and usage**

| Directory names | Application specific information |
|---|---|
| InstallDir\TekExpress\TekExpress Automotive-Ethernet | It contains the application and associated files. |
| TekExpress Automotive-Ethernet\AWG Waveforms | It contains AWG waveforms (AWG 5K, 7K and AFG waveforms) that get used in return loss and distortion with disturbing signal . |
| TekExpress Automotive-Ethernet\Bin | It contains miscellaneous TekExpress Automotive-Ethernet libraries. |
| TekExpress Automotive-Ethernet\Compliance Suites | It contains compliance-specific files and filter files. |
| TekExpress Automotive-Ethernet | It contains the Manuals. |
| TekExpress Automotive-Ethernet\Examples | It contains various support files. |
| TekExpress Automotive-Ethernet\ICP | It contains instrument and TekExpress Automotive-Ethernet-specific interface libraries. |
| TekExpress Automotive-Ethernet\Images | It contains images required for the application. |
| TekExpress Automotive-Ethernet\Lib | It contains utility files specific to the TekExpress Automotive-Ethernet. |
| TekExpress Automotive-Ethernet\Report Generator | It contains style sheets for report generation. |
| TekExpress Automotive-Ethernet\Tools | It contains instrument and TekExpress Automotive-Ethernet-specific files. |

**See also**

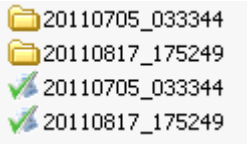
*View Test-Related Files*

## View test-related Files

Files related to TekExpress Automotive Ethernet tests are stored in the X:\Automotive Ethernet\Untitled Session shared folder. In the Automotive Ethernet 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 Automotive Ethernet icon and usually has no file extension displayed.

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 of its matching session folder.




Each session folder contains image files of any plots generated by the test session and any waveform files if prerecorded waveform files were used during the session.

The first time you run a new, unsaved session, the session files are stored in the X:\Automotive Ethernet\Untitled Session folder. Once you name and save the session, the Untitled Session folder name is changed to the one you specified.

***NOTE.*** *By default, test report files are saved in the session folder. You can change the report file location for a specific test.*

**See also** *File name extensions*

## File name extensions

The TekExpress Automotive Ethernet software uses the following file name extensions:

**Table 4: 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<br>The log file extension is also xml |
| .wfm | The test waveform file |
| .mht<br>.pdf | Test result reports are saved in this format by default. Test reports can also be saved in MHTML format and .pdf. |
| .cal | Calibration file used with transmitter tests |

**See also** *Application Directories and Usage*

# Getting started

## Compatibility

### Supported Oscilloscopes

The TekExpress Automotive Ethernet application runs on the following Tektronix oscilloscopes:

- MSO54, MSO56, and MS058

***NOTE.*** *Minimum 2 GHz oscilloscope required for 1000BASE-T1 test and 1 GHz oscilloscope for 100BASE-T1 test measurements.*

### Supported AWG/AFGs

The TekExpress Automotive Ethernet application supports the following Tektronix arbitrary waveform and arbitrary function generators:

**100BASE-T1**

- AWG7102 (Option 2): 10 GS/s
- AWG5014: 1.2 GS/s, 14-bit AWG
- AWG7122C: 12 GS/s
- AFG3252: 2 GS/s, 14-bit AFG
- AFG3102: 1 GS/s, 14-bit AFG
- AWG7122B
- AFG3022C
- AFG3052C
- AWG7082
- AWG7082B
- AWG7082C
- AWG5014B
- AWG5012C
- AWG5012B
- AWG5002C
- AWG5002B
- AWG5204
- AFG3252C

**1000BASE-T1**

- AWG5204
- AWG5202
- AFG3152C
- AFG3252C
- AFG3252

***NOTE.***

- *You may use AWG automation for Return Loss and Transmitter Distortion with Disturbing Signal measurements.*
- *Return Loss measurement is supported only on AWG5202 and AWG5204, with High Power option (AWG5200-2HV/AWG5200-4HV: High Amplitude DC coupled outputs) for AWG5200 S/N B020000 and above.*
- *For 1000Base-T1, use the Signal Generator, which can generate the 1.8 Volts Peak-to-Peak.*

## Probes

The TekExpress Automotive Ethernet application supports the following Tektronix differential probes:

**100BASE-T1**

- Return Loss Measurement: TDP500, TDP1000 and TDP1500
- All other measurements: P6247, P6248, P6330, TDP1500, TDP1000

**1000BASE-T1**

- All other measurements: TDP3500 ±15 V DC + pk AC input voltage

## Test Fixtures

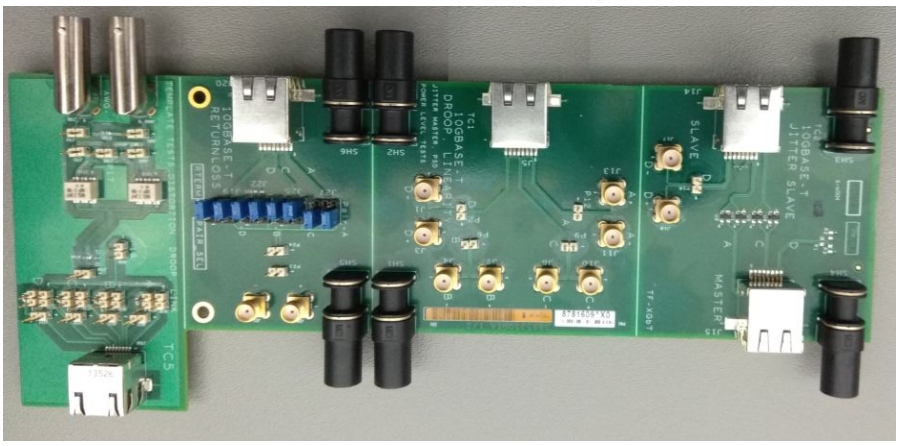


**Figure 4: TF-XGbT Ethernet fixture**

**1000BASE-T1**

The TekExpress Automotive Ethernet application supports the TF-XGbT test fixture.

**100BASE-T1**

The TekExpress Automotive Ethernet application supports the TF-XGbT test fixture.

**See Also** *Minimum system requirements*

## Minimum system requirements

The following table shows the minimum system requirements for an oscilloscope to run TekExpress Automotive Ethernet (100\1000BASE-T1).

**Table 5: System requirements**

| | |
|---|---|
| **Oscilloscope** | For a list of compatible oscilloscopes, see *Compatibility*. |
| **Firmware** | ■ TekScope 1.4.9 and above |
| **Software** | ■ Software version 1.2.0.x |
| **Probes** | Following differential probes are supported:<br>■ **100BASE-T1**:TDP1500, TDP3500, P6248, P6247add TDP3500 differential probes<br>■ **1000BASE-T1**:TDP3500 ±15 V DC + pk AC input voltage differential probe |
| **Automotive Ethernet Fixtures** | TF-XGbT for 1000BASE-T1 and 100BASE-T1 |
| **Other Devices** | |
| **Frequency Divider** | |
| 100BASE-T1 | The TF-BRR-CFD test fixture allows to phase lock frequency between DUT Transmitter Clock, oscilloscope , and AWG/AFG sources.<br>Converts DUT Transmitter Clock from 66.666 MHz to 10 MHz. |
| 1000BASE-T1 | The TF-BRR-CFD fixtures allows to phase lock frequency between DUT Transmitter Clock, oscilloscope, and AWG/AFG sources.<br>Converts DUT Transmitter Clock from 125 MHz to 10 MHz. |
| **Cables** | |
| 100BASE-T1 | Six same length BNC cables/SMA cables with BNC to SMA connectors: one TCA-BNC or TCA-SMA |
| 1000BASE-T1 | Six same length BNC cables/SMA cables with BNC to SMA connectors: one TCA-BNC or TCA-SMA |

**See also**

*Compatibility*

## Install the software

Complete the following steps to download and install the latest TekExpress Automotive Ethernet application. See Minimum system requirements for compatibility.

1. Close all applications (including the TekScope application).
2. Go to the www.tek.com.
3. Click **Downloads**. In the Download menu, select DOWNLOAD TYPE as Software and enter TekExpress Automotive Ethernet in the MODEL OR KEYWORD field and click **SEARCH**
4. Select the latest version of software and follow the instructions to download. Copy the executable file to the oscilloscope
5. Double-click the executable and follow the on-screen instructions. The software is installed at C:\Program Files\Tektronix\TekExpress\TekExpress Automotive-Ethernet
6. Select **Application**>**TekExpress Automotive Ethernet** from the TekScope menu to launch the application.

**See also**

*Minimum System Requirements*

## Activate the license

Follow these steps to activate the TekExpress Automotive Ethernet license:

- From the MSO 5 Series Oscilloscope menu bar, click **Help** > **About**.
- Click Install License, and then select the .Lic file.
- Follow the prompts of the oscilloscope to active the license.
- After activating the license, reboot the oscilloscope.

***NOTE.*** *Check the Oscilloscope help for the steps to activate the license. Click F1 key to open the Option Installation topic and follow the steps to activate the license. Press the **F1** key on the oscilloscope keyboard to open the Option Installation help topic and follow the steps to activate the license.*

**See also**

*View Version and License Information*

## View version and license information

Use the following instructions to view application version information and license information for the application modules.

To view version information:

1. From the Options menu, select **About TekExpress**.
2. Click the **View Version Details** link to check the version numbers of the installed test suites. Close the dialog box when finished.

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 TekExpress Automotive Ethernet solution.
2. To view the Option key, look in the Option Installation Key section. When finished, click **OK** to close the dialog box.

**See also** *Activate the License*

*Options Menu*

## Start the application

To run the TekExpress Automotive Ethernet application from the oscilloscope menu, select **Application**>**TekExpress Automotive Ethernet**.

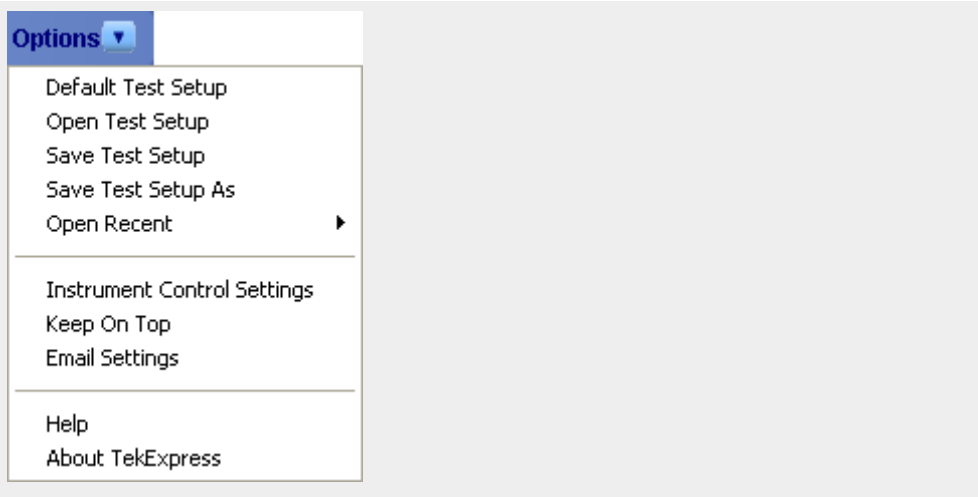
While running the application, you can switch between the oscilloscope screen and TekExpress Automotive Ethernet application by clicking the desired window. To keep the application window on top, select **Keep On Top** from the Options menu.

***NOTE.*** *If the application was not terminated properly during the last use, a dialog box prompts you to recall the previously unsaved session.*

# Options menu

## Options menu

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




It has the following selections:

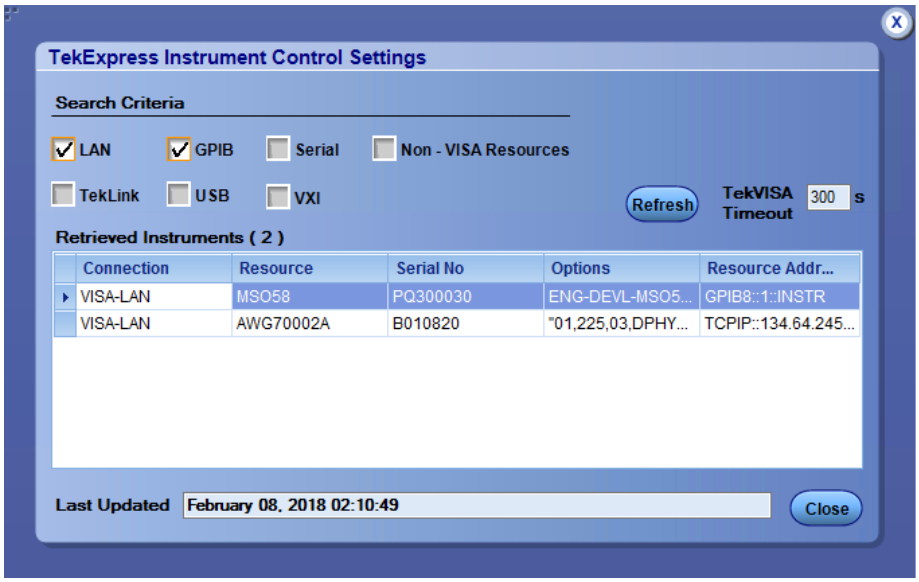
**Table 6: Options menu**

| 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 | It displays a menu of recently opened test setups to select from. |
| *Instrument Control Settings* | It shows the list of instruments connected to the test setup and allows you to locate and refresh connections to connected instruments. |
| Keep On Top | It keeps the TekExpress Automotive Ethernet application on top of other open windows on the desktop. |
| *Email settings* | It configures email options for test run and results notifications. |
| Help | It displays TekExpress Help. |
| About TekExpress | ■ It displays application details such as software name, version number, and copyright.<br>■ It provides access to *license information* for your TekExpress Automotive Ethernet application installation.<br>■ It provides a link to the Tektronix Website. |

## View connected instruments: the instrument control settings

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.

**Figure 5: Instrument Control Settings**



To refresh the list of connected instruments:

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

   Instrument search is based on the VISA layer but different connected cables determine the resource type, such as LAN, GPIB, USB. For example, if you choose LAN, you can search for all the instruments supported by TekExpress that are communicating over the LAN. 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. Click **OK** to close the message window.

   ***NOTE.*** *Ensure that the GPIB option is enabled to detect the real time oscilloscope in the Global Settings tab.*

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 LAN and GPIB as the search criteria, the application checks for the availability of instruments over LAN, then 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.

# Application controls

**Table 7: Application controls descriptions**

| Item | Description |
|---|---|
| Options menu | It opens the Options menu for access to global controls. |
| Panels | Visual frames with sets of related options. Some panels are further divided into tabs and other sections. |
| Start button |  <br>Use the Start button to acquire and accumulate measurements. If prior acquired measurements have not been cleared, the new measurements are added to the existing set. |
| Stop button |  <br>Use the Stop button to abort a test in progress. |
| Pause \ Continue button |  <br>Use the Pause button to interrupt the current acquisition. When a test is paused, the button name changes to Continue. |
| Clear button |  <br>Available only on the *Results panel*.<br>Use the Clear button to clear all existing measurement results.<br>***NOTE.*** *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.* |
| Application window move icon |  <br>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. |
| Mini view/ Normal view |  <br>Toggles the application between Mini view and Normal view. Mini view displays the run messages with the time stamp, progress bar, Start/Stop button, and Pause/Continue button.<br>The application moves to Mini view when you click the Start button. |

**See also**

*Options menu*

## 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:
   - A message box appears asking if you really want to exit TekExpress. To exit, click **Yes**; otherwise, click **No**.
   - If you have an unsaved session or test setup open, you are prompted to save it before exiting. To save it, click **Yes**. If you do not wish to save, click **No**. To remain in the session, click **Cancel.**.

# Application basics

## Application basics

The TekExpress Automotive Ethernet software user interface is intuitive and easy to use. In addition to the UI, a *programmatic interface* is available.

The user interface has four main panels, which allows you do the following:

- *Setup panel*
- *Before running the tests for the first time*
- *Status panel*
- *Results panel*
- *Save and recall test setups*

**See also** *Overview and key specifications*

# Setting up tests

## Setting Up Tests: the Setup Panel

The Setup panel guides you through the Automotive Ethernet test setup process using tabs. The options selected in a tab will have an effect on the options available in the next tab down.

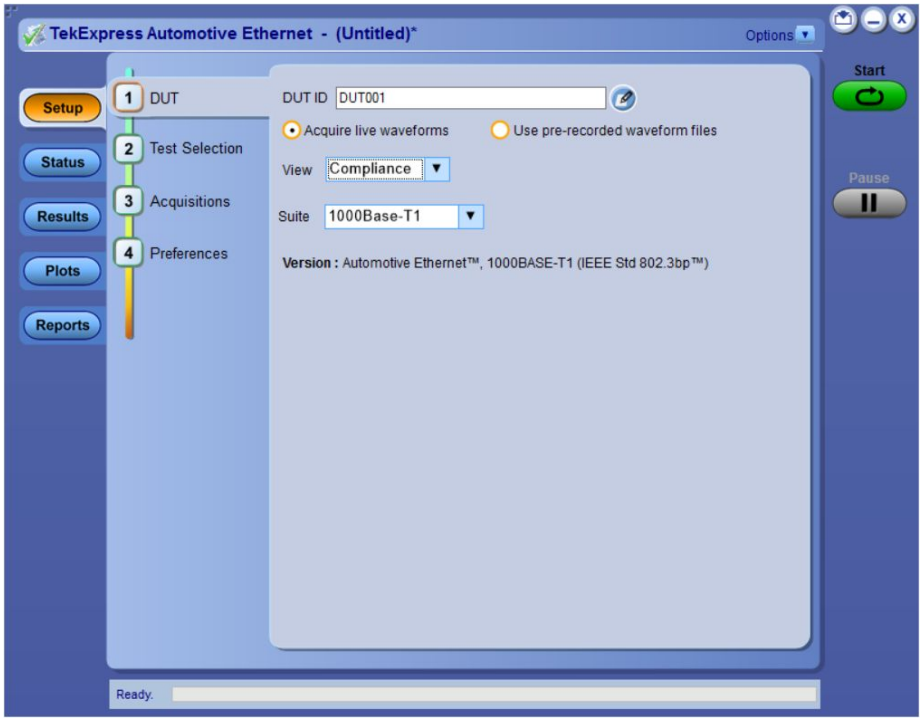



**Figure 6: Setup panel**

- *DUT tab*

  Specify the DUT ID and the view settings. These settings apply to all tests for the current session, and affect the test list in the Test Selection tab.

- *Test Selection tab*

  Select tests individually or by group, view a short description of a selected test, and view a schematic showing appropriate device connections.

- *Acquisitions tab*

  View configurations for attached probes, view a *list of signals and sources*, and view the Acquisitions table.

- *Configuration tab* (Displays only when **View: Advanced** is selected in the DUT tab.)

Select either **Compliance Mode** or **User Defined Mode**, view Global Settings, select measurement settings, and view or edit test parameter limits.

- Preferences tab

  Specify that an email be sent to you upon test failure.

**See also**

*Saving a Test Setup*

*Running the Tests and Viewing their Progress in the Status Panel*

*Viewing Test Results in the Results Panel*

## Selecting device parameters: the DUT tab

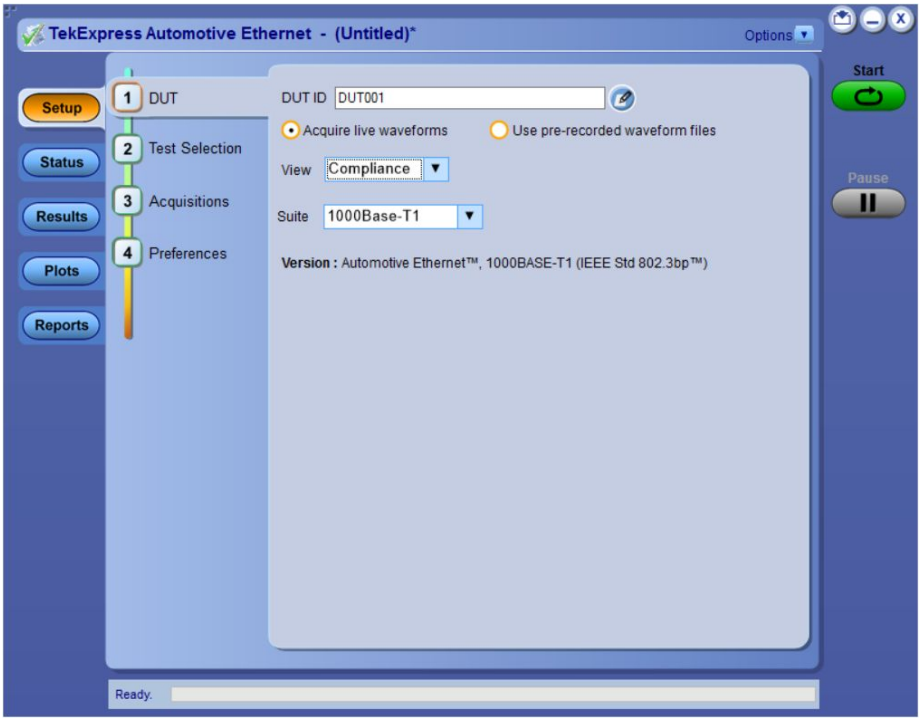



**Figure 7: Setup panel**

1. Click the DUT tab on the Setup panel and specify the DUT ID and the view settings.

These settings apply to all current tests, and affect the test list in the Test Selection tab.

2. Enter the device **ID** (default value is DUT001). This ID appears on test reports. To add comments to the top of a report, click and enter up to 256 characters.
3. For the View option, select either **Compliance** or **Advanced**.
   - If you select the **Advanced** view, an additional Configuration tab appears.
   - To access configuration options for Compliance view, click the **Configure** button on the Test Selection tab.

**See also**

*Setting up tests: the Setup panel*

*Choosing tests: the Test Selection tab*

*Acquiring waveforms: the Acquisitions tab*

*Configuring tests: the Configure button or Configuration tab*

## Selecting tests: the Test Selection tab

Use this tab to select the tests to run on the connected DUT. The tests that you select here impact the parameters available in the Acquisitions tab.

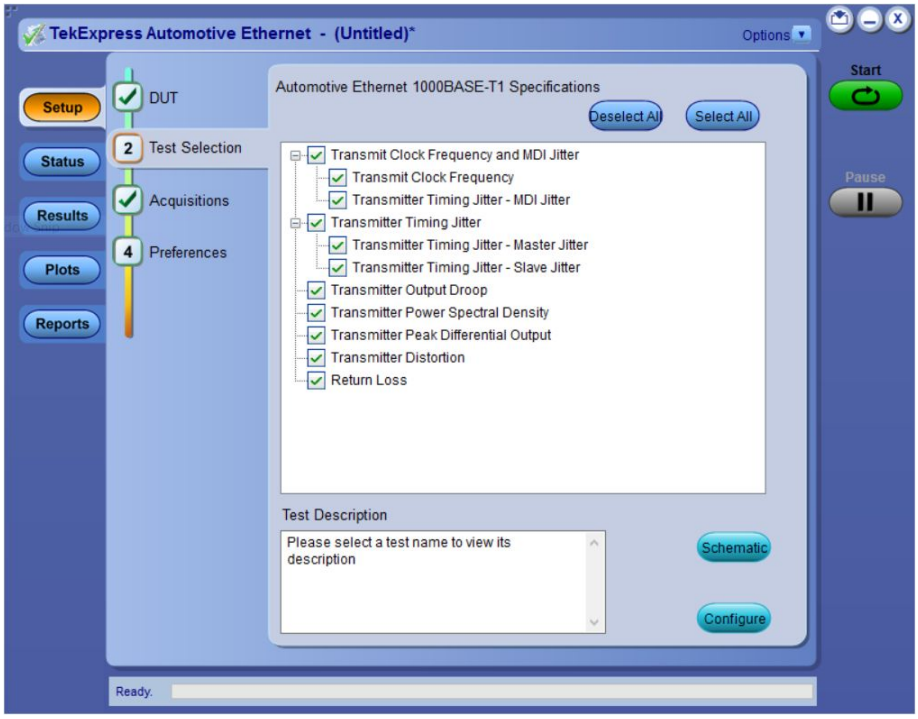



**Figure 8: Test Selection tab**

To select a test, do the following:

1. In the Setup panel, click the **Test Selection** tab.
2. Select the desired tests. For a list of supported tests, *click here*.
   - Select the check boxes of individual tests or of entire groups of tests. To select all tests in the list, click the **Select All** button and click **Deselect All** unless you want to run all tests.
   - Click on the **Schematic** button to display the schematic document for the selected test. Use to verify the test setup before running the test.

Once you have selected the tests, then you can *configure the tests*.

**See also**

*Setup panel*

*Acquiring waveforms*

*Configuration tab*

*Status panel*

## Selecting Acquisitions

### Acquiring waveforms: the Acquisitions tab

The Acquisitions tab in the Setup panel is used to view and set acquisitions parameters for the selected tests. Before you can do this, you must first:

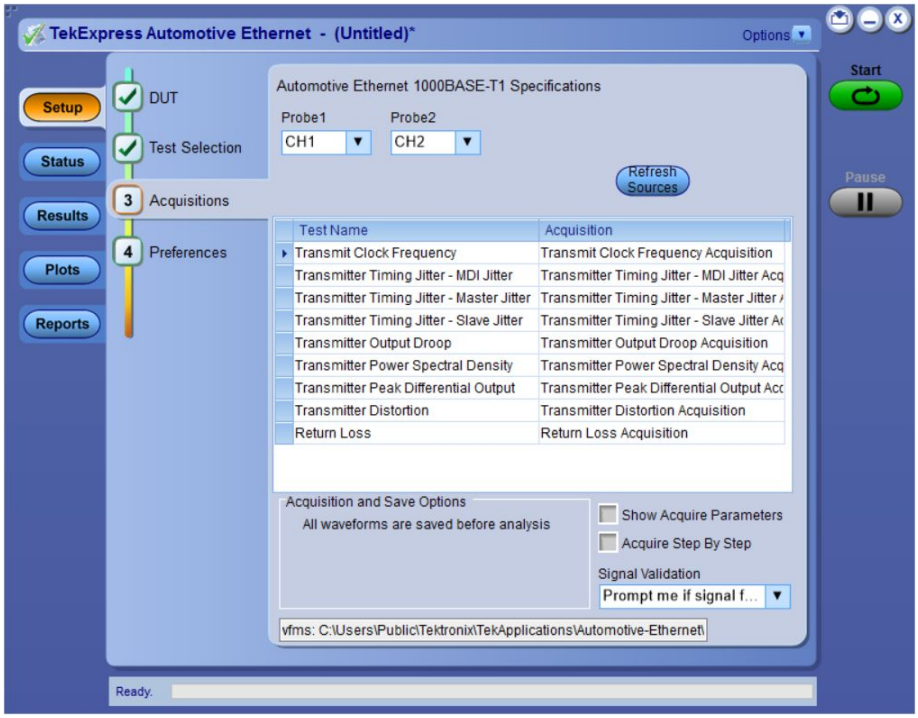



**Figure 9: Acquisitions tab with Acquire live waveforms selected**

- Select either **Acquire Live Waveforms** or **Use Pre-recorded waveform files** on the DUT tab of the Setup panel.
- The acquisition parameters displayed on the Acquisitions tab will differ depending on your choice on the DUT tab.

If you select **Acquire Live Waveforms** on the DUT tab, do the following:

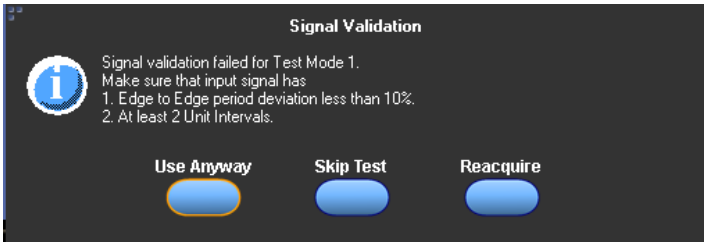
1. In the Setup panel, click the **Acquisitions** tab.
2. Select the probe source channel for each listed signal in the Probe selection drop-down menu.
3. You may select to view the probe configuration for each channel used.
4. You will see the Acquisitions table. For information about the possible parameter columns displayed in the Acquisitions table, click here.

The following table below lists the parameter columns displayed in the Acquisitions table in the Acquisitions tab of the Setup panel. Columns displayed depend on the tests selected in the Test Selection tab, and whether or not the Show Acquire Parameters check box has been selected.

**Table 8: Acquisitions parameters**

| Column name | Description |
|---|---|
| Test Name | Displays the name of the selected test for performing acquisitions. One or more tests can perform the same acquisitions. |
| Acquisition | Displays the named acquisition. |
| Record Length | Displays the size of the record in samples. |
| Disturbing signal | Displays whether or not a disturbing signal is excluded. |
| Average | Displays the number of averages that will be used with average mode acquisition. |
| AWG Series | Displays the AWG series model, if present. |
| TX_TCLK | Displays whether or not a TX_TCLK is excluded. |
| Hi resolution | Displays the HiRes(Vertical resolution) factor. |
| WaveForm FileName | Displays the name and location of the waveform file to be used for the measurement. Applies only to testing using prerecorded waveforms. |

5. Underneath the table, you may select the **Show Acquire Parameters** check box. When selected, the acquisition parameters for each test displays an additional columns in the Acquisitions table.
6. You may select the **Acquire Step by Step** check box. When selected, the application prompts you to continue after each phase of test completes.
7. Select a Signal validation parameter (Signal validation is valid only for Live acquisitions):
   - **Prompt me if signal fails**: Select to prompt if signal fails.




   - **Use Anyway**: Click to Run the test on the current acquired signal in spite of failed signal.
   - **Skip Test**: Click to skip the current test after signal validation fails.
   - **Reacquire**: Click to reacquire the signal for the test being Run.
   - **Use signal as is-Don't check**: Select to perform the test without signal validation.
   - **Skip test if signal fails**:Select to skip the test for which signal validation fails.

1. In the Setup panel, click the **Acquisitions** tab.

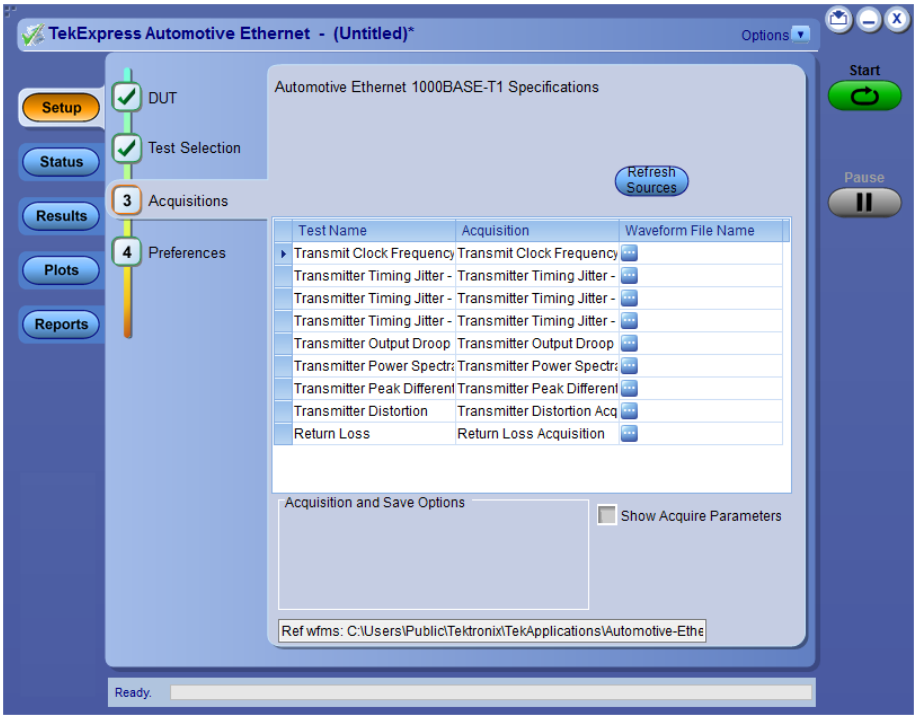



**Figure 10: Acquisitions tab with Use pre-recorded waveform files selected**

2. You will see the Acquisitions table. Locate the row for the desired test and then click the ellipsis button ( ) in the Waveform FileName column. Select a file. You can select more than one file for each test. For information about the possible parameter columns displayed in the Acquisitions table, click here.
3. Underneath the Acquisitions table, you may select the **Show Acquire Parameters** check box. When selected, the acquisition parameters for each test are displayed in additional columns in the Acquisitions table.

**See also.** *Use prerecorded waveforms for analysis*

*Acquire live waveforms for analysis*

*Setting up tests the Setup panel*

## Acquire live waveforms for analysis

Use these instructions to set up acquisition for a live waveform.

1. *Open a saved test setup* or *create a new one*.
2. In the Setup panel, select the **DUT** tab, enter the desired DUT ID, and then select the **Acquire live waveforms** check box.
3. *Select other DUT options* as desired.
4. In the Test Selection tab, *select the desired test(s)*.
5. In the Acquisitions tab, select or view the desired parameters in the Acquisitions Table.
6. *Configure the tests* if you have not done so already, and then click **Start** to run the test.

**See also.** *Acquiring waveforms the acquisitions tab*

*Use prerecorded waveforms for analysis*

## Use pre-recorded waveforms for analysis

You can use pre-recorded waveforms for analysis.

***NOTE.*** *If you are using the pre-recorded waveform files option, it is recommended that you use a waveform file (.wfm) that was captured from a Tektronix oscilloscope. This eliminates the need to use an oscilloscope. You can manually select waveforms and perform the tests by clicking the* ***Start*** *button. The reference waveforms are provided with the installer at C:\Users\Public\Tektronix\TekApplications\TekExpress Automotive-Ethernet.*

1. *Open a saved test setup* or *create a new one*.
2. In the Setup panel, select the DUT tab, select the desired DUT, and then select **Use pre-recorded waveform files** check box.
3. In the Test Selection tab, *select the desired tests*.
4. In the Acquisitions tab, locate the row for the desired test in the Acquisitions Table, and then In the **Waveform FileName** column for that test, click the ellipses button (  ) and select the desired waveform file.
5. Perform any additional desired test setup, such as *test configuration*, and then click **Start** to run the test.

You can select a different waveform file for each test, clear a test from the current run, or add a test to the current run. Once you click **Start**, data corresponding to that test run populates the other panels (such as the test status and test results).

**See also.** *Acquiring waveforms the Acquisitions tab*

*Acquire live waveforms for analysis*

### Acquire step by step

The Acquire Step By Step option is available in the Acquisitions panel. This is a global parameter that is applied to all tests when selected. By default, this option is deselected.

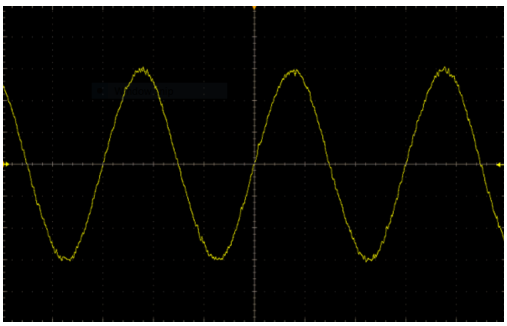
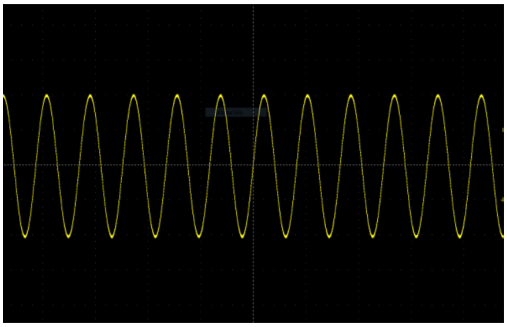
When selected, this parameter allows for display of the reference input waveform of the selected measurement. This helps to compare the input waveform coming from DUT with the typical reference waveform (snap shot), allowing you to change the setup before acquiring the waveforms. The following table gives different reference waveform snap shots that appear for different tests.

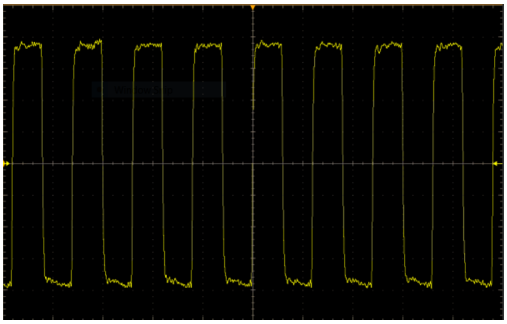
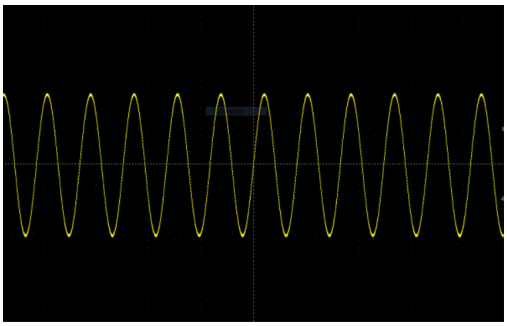
---

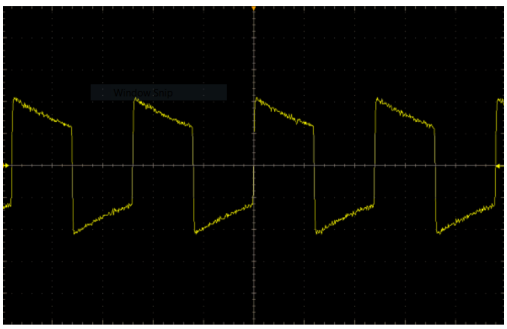
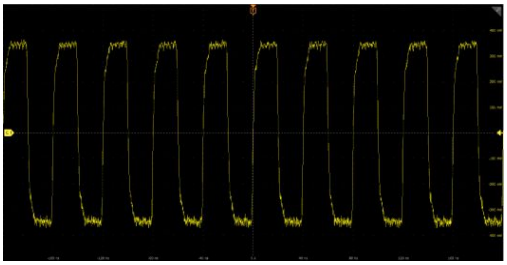
***NOTE.*** *When using prerecorded waveform files, the* ***Acquire Step By Step*** *option is not available.*

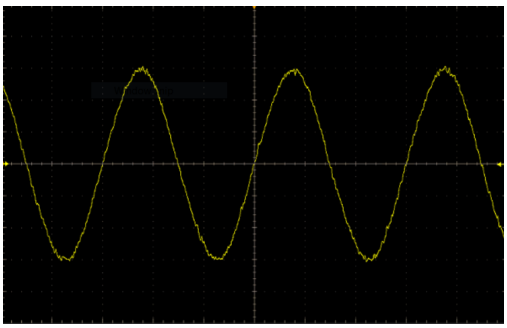
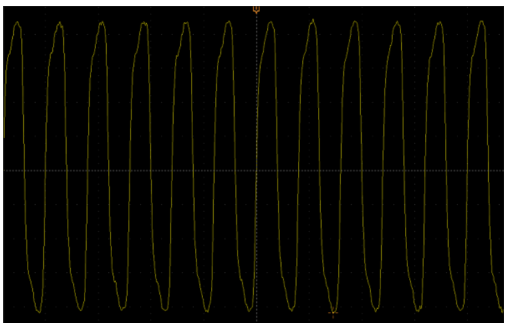
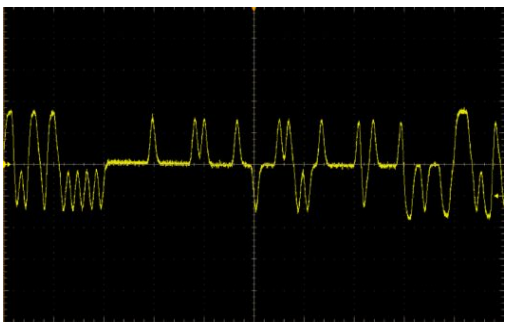
---

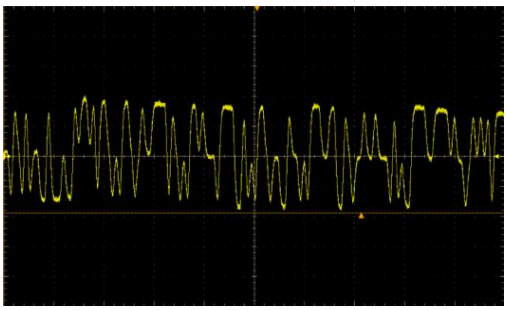
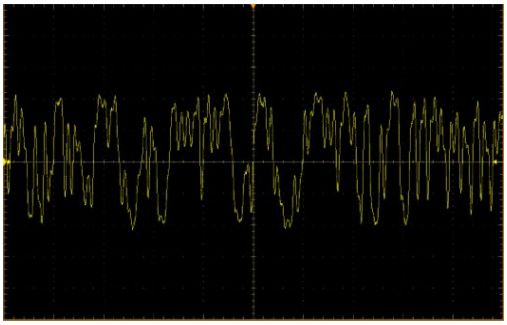
**Table 9: Reference waveforms**

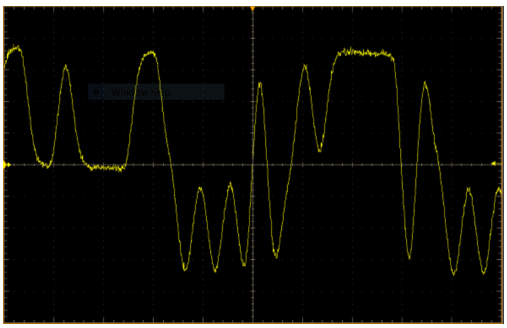
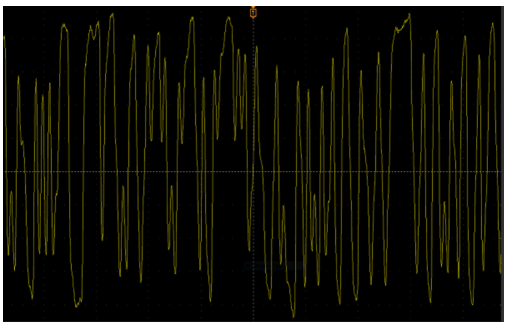
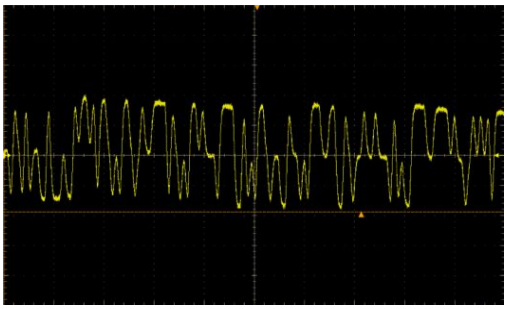
| Test name | Reference waveform |
|---|---|
| Transmit Timing Jitter – Master Jitter | 100BASE-T1<br>  <br>1000BASE-T1<br>  |

| Test name | Reference waveform |
| --- | --- |
| Transmit Timing Jitter – Slave Jitter | 100BASE-T1<br>  <br>1000BASE-T1<br>  |

| Test name | Reference waveform |
|---|---|
| Transmitter Output Droop | 100BASE-T1<br>  <br>1000BASE-T1<br>  |

| Test name | Reference waveform |
|---|---|
| Transmit Clock Frequency | 100BASE-T1<br>  <br>1000BASE-T1<br>  |
| Transmitter Distortion | 100BASE-T1<br>  |

| Test name | Reference waveform |
| --- | --- |
| Transmitter Power Spectral Density | 100BASE-T1<br>  <br>1000BASE-T1<br>  |

| Test name | Reference waveform |
|---|---|
| Transmitter Peak Differential Output | 100BASE-T1<br>  <br>1000BASE-T1<br>  |
| Return Loss | 100BASE-T1<br>  |

## Configuring tests: the Configuration screen

Once you've selected tests, you may use the Configuration screen. There are two ways to reach the screen: the Configuration tab or the Configure button.

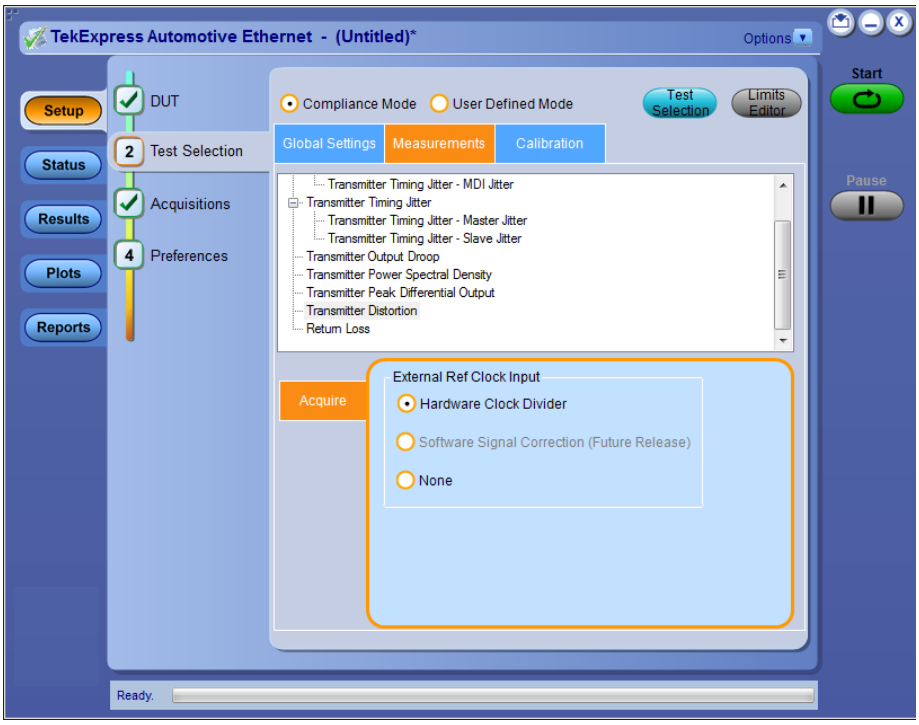



**Figure 11: Configure button on Test Selection tab**

The TekExpress Automotive Ethernet solution you are using determines how you may reach the Configuration screen.

- If you selected **Compliance** View in the Setup panel of DUT tab, click the **Test Selection** tab and select the desired test. Then click the **Configure** button in the lower right corner of the Test Selection tab.
- If you selected **Advanced** View in the Setup panel of DUT tab, the Configuration tab will appear on the Setup panel – click on it.

The Configuration screen shows Global parameters, which are common for all tests, and Measurement parameters, which are specific to selected tests, including acquisition, analysis, and limit parameters.

***NOTE.*** *Test parameters that are grayed cannot be changed.*

**Settings**

- *Compliance Mode or User Defined Mode*
- Global Settings tab

In the Instruments Detected section, click in the shaded area opposite Real Time Scope and select the desired instrument from the drop-down list. If you don't see the instrument, refresh the list by using the *Instrument Control Settings*.

- Measurements tab

Lists all tests for the selected measurement type. Click on a measurement to view the available parameters in the tabbed filed below the list. The parameters and parameter type tabs shown depend on the selected test. To edit the test parameters, select the **User Defined Mode** check box.

- You may change test parameters. Select the test whose parameters you would like to change. The options below the test list change to reflect the selected test.

  To modify the parameters, select the desired tab and parameters. For parameter details, see *Measurement parameter descriptions*.

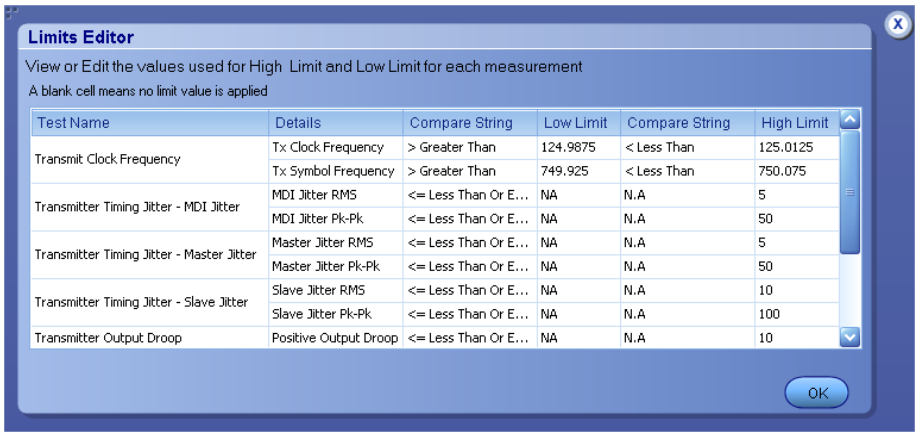



**Figure 12: Limits Editor**

- Use the Limits Editor button in the upper right corner to view or change the High Limit and Low Limit values used for each measurement.

Next, *Prepare to run the tests.*

**See Also**

*Setting up tests: the Setup panel*

*Selecting device parameters: the DUT tab*

*Choosing tests: the Test Selection tab*

*Acquiring waveforms: the Acquisitions tab*

*About saving and recalling test setups*

## Compliance Mode or User Defined Mode

From the Configuration screen, you will have the option to select either Compliance Mode or User Defined Mode.

- **Compliance Mode**: Select to use Compliance Mode values. You cannot change most test parameters in Compliance Mode, but you can view the compliance parameters.
- **User Defined Mode**: Select to run tests using custom parameters. You may change parameters that are not grayed out.

***NOTE.*** *These modes are not to be confused with the two views available from the DUT tab:* ***Compliance View*** *and* ***Advanced View****.*

### Pairing Modes and Views for Test Operation

Your selections of Modes and Views will change the way the tests operate. These pairings of views and modes are possible:

- **Compliance View** selected with **Compliance Mode**: Tests will run automatically with little or no user intervention. You will not be able to change test parameters to anything that deviates from the compliance standards. To view configuration options, click the **Test Selection** tab of the Setup panel, and then click the **Configure** button.
- **Compliance View** selected with **User Defined Mode**: Tests will run automatically but you will be able to change test parameters before starting the tests. To view configuration options, click the **Test Selection** tab of the Setup panel, and then click the **Configure** button.
- **Advanced View** selected with **Compliance Mode**: Tests will run automatically with little or no user intervention. You will not be able to change test parameters to anything that deviates from the compliance standards. To view configuration options, click the **Configuration** tab of the Setup panel.
- **Advanced View** selected with **User Defined Mode**: Tests will run automatically but you will be able to change test parameters before starting the tests. To view configuration options, click the **Configuration** tab of the Setup panel.

## Measurement parameter descriptions

View or change measurement parameters in the Configuration tab of the Setup panel. Measurement parameters are displayed for the test selected in the tree view section. Not all of the parameters listed apply to all tests, and some are only available when running tests in **User Defined** Mode. You cannot change any parameters if you selected **Compliance** Mode.

### 1000BASE-T1

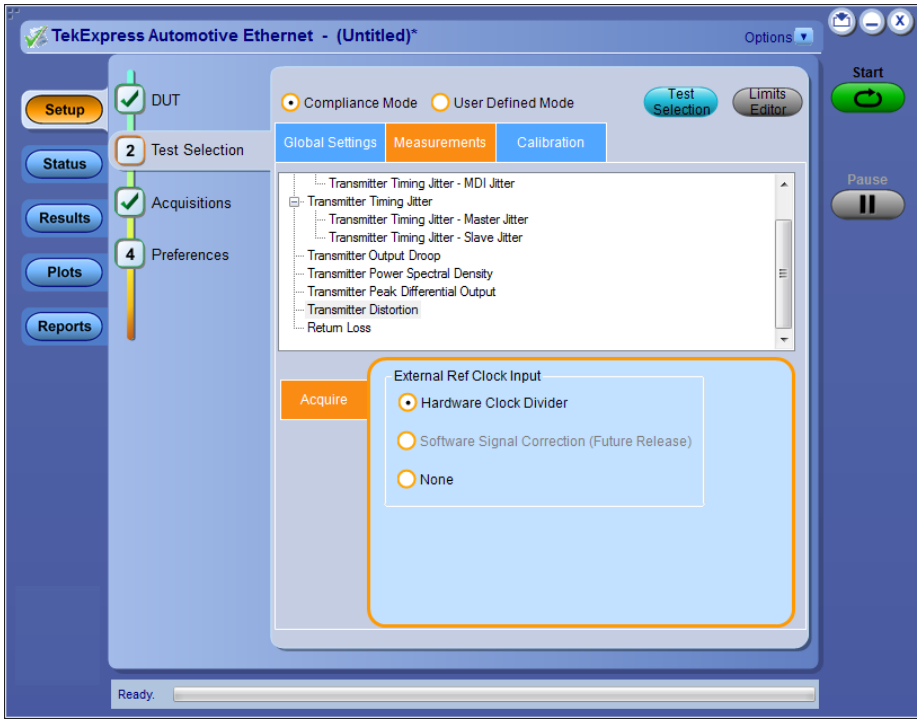



**Figure 13: Measurements tab**

**Table 10: Measurement parameters for 1000BASE-T1**

| Name | Unit | Range/ Allowable values | Default | Description | Applies to |
|---|---|---|---|---|---|
| Record Length | M | 1 to 20 | 2.75 M | Sets the record length to use. | Transmit Clock Frequency |
| | | | 12.5M | | Transmitter Timing Jitter-MDI<br>Transmitter Timing Jitter (Master Jitter)<br>Transmitter Timing Jitter (Slave Jitter) |

| Name | Unit | Range/ Allowable values | Default | Description | Applies to |
|---|---|---|---|---|---|
| Averages | NA | 2 to 100 | 2 | Sets the number of averages (number of acquisitions) for average mode acquisition. | Transmitter Output Droop |
| | | 2 to 1000 | 500 | | Return Loss |
| No of Acquisitions | NA | 1 to 100 | 5 | Sets the number of acquisitions | Transmitter Peak Differential Output |
| Hardware Clock Divider | NA | NA | Hardware Clock Divider | Select to use the external hardware clock divider unit (125 MHz Tx_CLK to 10 MHz) to synchronize the DUT, disturber signal source (AWG/ AFG), and the oscilloscope. | Transmitter Distortion |
| None | NA | NA | NA | Select to check the behavior of the DUT without any Ref clock (External or Internal). | Transmitter Distortion |
| Edge<br>■ Falling<br>■ Rising | NA | NA | Rising | Used to select the type of edges on which RMS Jitter and Peak-to-Peak Jitter will be calculated. | Transmitter Timing Jitter-MDI Jitter Transmitter Timing Jitter (Master Jitter) Transmitter Timing Jitter (Slave Jitter) |

| Name | Unit | Range/ Allowable values | Default | Description | Applies to |
|---|---|---|---|---|---|
| Hysteresis | % | 1 to 10 | 5% | Sets the hysteresis in percentage that gets used during edge finding. | Transmitter Timing Jitter (Master Jitter) Transmitter Timing Jitter (Slave Jitter) Transmitter Timing Jitter - MDI Jitter |
| None | NA | NA | NA | NA | Transmitter Power Spectral Density |
| Smooth | NA | 0 to 10 | 7 | Sets the number of samples that will be used while smoothing the return loss waveform; sets the averaging filter length. | Return Loss |

**100BASE-T1**

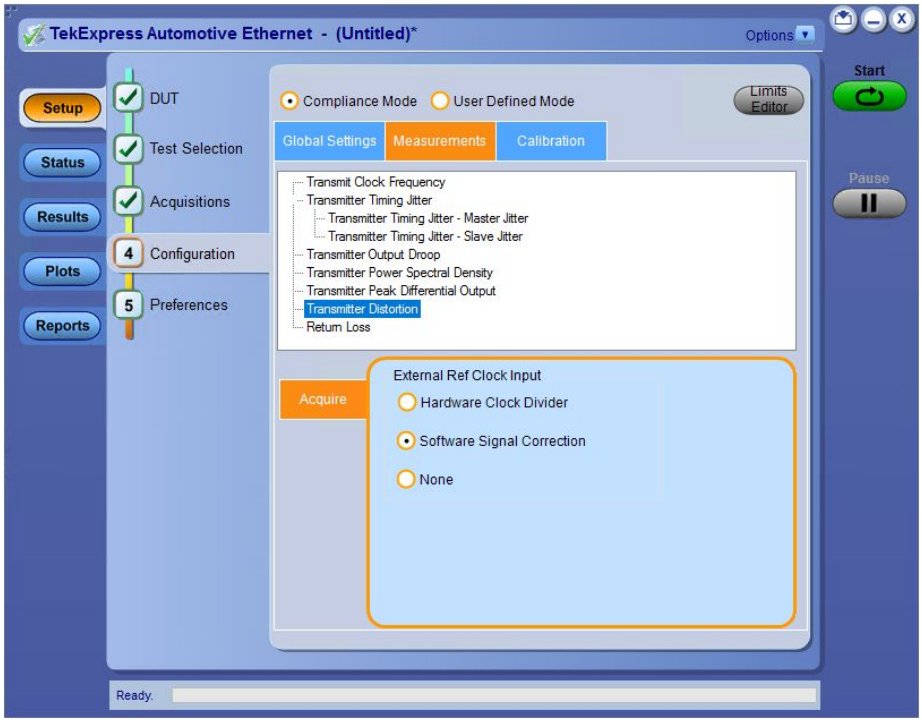


**Table 11: Measurement parameters for 100BASE-T1**

| Name | Unit | Range/ Allowable values | Default | Description | Applies to |
|---|---|---|---|---|---|
| Record Length | M | 1 to 20 | 0.5M | Sets the record length to use. | Transmit Clock Frequency |
| | | | 12.5M | | Transmitter Timing Jitter (Master Jitter) Transmitter Timing Jitter (Slave Jitter) |
| Averages | NA | 2 to 100 | 16 | Sets the number of averages (number of acquisitions) for average mode acquisition. | Transmitter Output Droop |
| | | 2 to 512 | 500 | | Return Loss |
| Spectral Average | NA | 2 to 256 | 2 | Sets the number of spectral averages | Transmitter Power Spectral Density |
| Hardware Clock Divider | NA | NA | NA | Select to use the external hardware clock divider unit (66.66 MHz Tx_CLK to 10 MHz) to synchronize the DUT, disturber signal source (AWG/ AFG), and the oscilloscope. | Transmitter Distortion |
| Software Signal Correction | NA | NA | NA | Select to use the internal software signal correction to synchronize the DUT, disturber signal source (AFG/ AWG), and the oscilloscope. | Transmitter Distortion |

| Name | Unit | Range/ Allowable values | Default | Description | Applies to |
|---|---|---|---|---|---|
| None | NA | NA | NA | Select to check the behavior of the DUT without any Ref clock (External or Internal). | Transmitter Distortion |
| Edge<br>■ Falling<br>■ Rising | NA | NA | Rising | Used to select the type of edges on which RMS jitter will be calculated. | Transmitter Timing Jitter (Master Jitter)<br>Transmitter Timing Jitter (Slave Jitter) |
| Hysteresis | % | 1 to 10 | 5% | Sets the hysteresis in percentage that gets used during edge finding. | Transmitter Timing Jitter (Master Jitter)<br>Transmitter Timing Jitter (Slave Jitter) |
| RBW | KHz | 1 to 100 | 10KHz | Sets the resolution bandwidth. This controls the bandwidth of the spectral analyzer filters. | Transmitter Power Spectral Density |
| Center Frequency | MHz | 50 to 500 | 100.5MHz | Sets the center of the frequency span over which spectral analysis is done. | Transmitter Power Spectral Density |
| Frequency Span | MHz | 100 to 500 | 201MHz | Sets the range of frequencies over which spectral analysis is done | Transmitter Power Spectral Density |
| Smooth | NA | 0 to 10 | 7 | Sets the number of samples that will be used while smoothing the return loss waveform; sets the averaging filter length. | Return Loss |

**See also** *Configuring tests: the Configure button or Configuration tab*

## Calibration

Calibration can be performed on live waveforms or by using the already-calibrated waveforms.

**Measurements**

Following are the measurements which need calibration before running the test.

*Transmitter Distortion*

In Transmitter distortion, calibration is done to effectively remove the disturbing signal and compensate for non-linearity in the disturber and test fixture.


**Figure 14: Calibration tab - 1000BASE-T1**


**Figure 15: Calibration tab - 100BASE-T1**

*Return Loss*

- **Live calibration**


**Figure 16: Live calibration**

***NOTE.*** *The live Calibration files are saved in X:\Automotive Ethernet\Calibration folder.*

- **Use pre-recorded files**

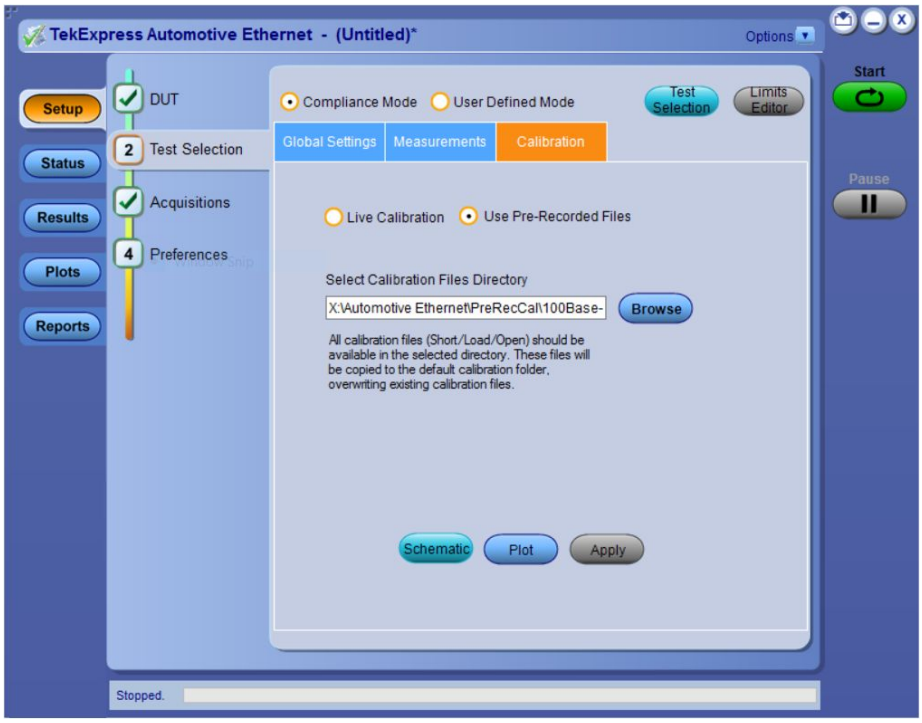



**Figure 17: Use pre-recorded files**

**Table 12: Calibration table parameters**

| Parameter | Description |
|---|---|
| Live Calibration | Sets the live calibration process. |
| Use Pre-recorded Calibration | Sets the calibration process with prerecorded calibrated waveforms and allows you to browse and select the calibrated waveforms. The path for selecting the calibrated waveforms is C:\Users\Public\Tektronix\TekApplications\Automotive-Ethernet. |
| Channel drop down | Allows you to select relevant channel and probes.<br>***NOTE.*** *when you change the input sources (Channel) other than calibrated sources, you need to re-calibrate with latest sources.* |
| Cal Type | Displays the type of calibration: Load, Open, and Short. |

| Parameter | Description |
|---|---|
| Cal Status | Displays the status of the calibration: Pending, Done. |
| Cal Time | Displays the previous calibration time: Date, Month, and Year . |
| Schematic button | Click to view the schematic. |
| Plot button | Click to view the plot. |
| Apply button | Click to apply the configured parameters to calibration. |
| Run button | Click to run the process of calibration. |
| Default button | Click to perform calibration by using default values.<br>***NOTE.*** *This button is applicable only for Transmitter distortion. The expected and measured values for Transmitter distortion on Calibration tab will be initially empty, when you launch the application for the first time and they get populated once you click the* ***Default*** *button .* |
| To manually set up signal source: Click here and copy files to USB drive and recall on connected signal source. | In case of manual signal source (AFG/AWG) setup, click the link and copy the relevant folder and recall the setup on connected signal source. |

# Running tests

## Before you click start

**Before running a test for the first time**

Before you run a test for the first time, review these steps:

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

After you install and launch TekExpress Automotive Ethernet, it creates the following folders on the Oscilloscope:

- \Program Files\Tektronix\TekExpress\TekExpress Automotive-Ethernet
- \My Documents\My TekExpress\Automotive Ethernet
- \My Documents\My TekExpress\Automotive Ethernet\Untitled Session

  Every time you launch TekExpress Automotive Ethernet.exe, an Untitled Session folder is created in the Automotive Ethernet folder. The Untitled Session folder is automatically deleted when you exit the Automotive Ethernet application.

***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 the associated files, is created on the oscilloscope X: drive.*

2. Ensure 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, ensure that the option Encrypt contents to secure data is NOT selected.

Example:

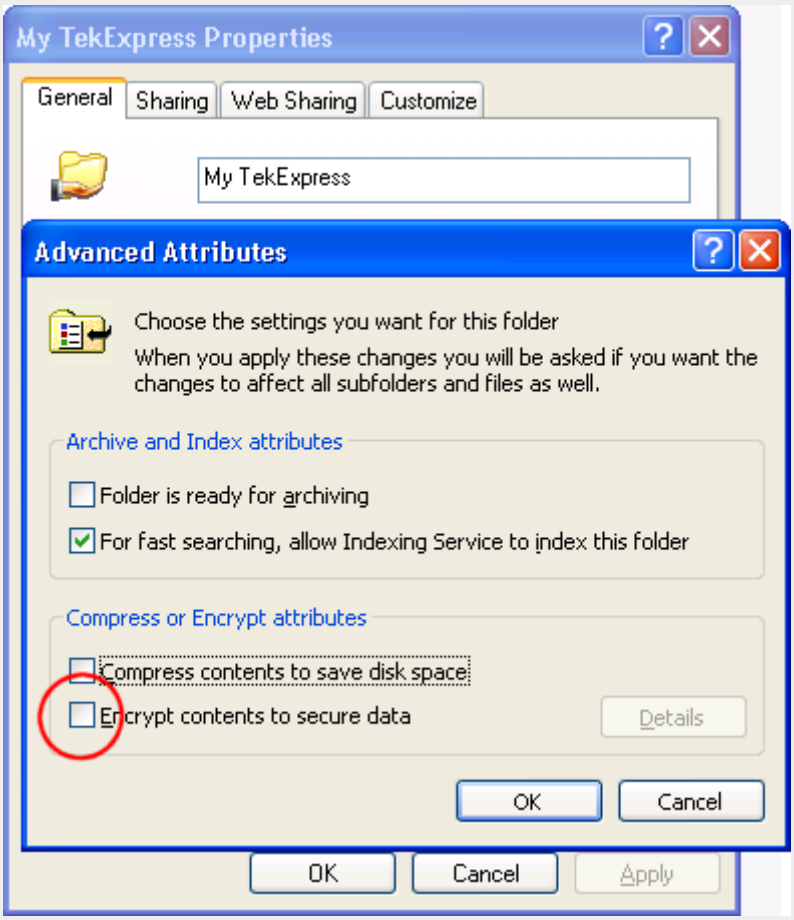



**Figure 18: Advanced Attributes dialog box**

**Before running any test**

1. Review the *Pre run check list*.
2. Configure the *Email notification options* if you want the application to notify you by email when a test completes or produces an error. Access the email options either from the Options menu in the upper right corner, or from the Preferences tab on the Setup panel.
3. Select the Report options.

**See Also**

*Prerun check list*

## Pre-run check list

Do the following before you click **Start** to run a test. If this is the first time you are running a test on a setup, then refer also to the guidelines above.

1. Ensure that all the required instruments are properly warmed up (about 20 minutes).
2. Perform the Signal Path Compensation (SPC).
   a. On the oscilloscope main menu, select the **Utilities** menu.
   b. Select **Instrument Calibration**.
3. Deskew any cables.
4. Ensure that the application is able to find the DUT. If it cannot, *perform a search for connected instruments*.

To find the DUT:

a. Launch the TekExpress Automotive Ethernet application.
b. Select the **Setup** panel and then click the **DUT** tab.
c. Click the **Test Selection** tab. Select any test, and then click **Configure** button.
d. In the Configuration section, click **Global Settings**.
e. In the Instruments Detected section, click in the shaded area opposite Real Time Scope and ensure that the oscilloscope with the (GPIB8::1::INSTR) designation is in the drop-down list.

**See also**

*Before you click start*

*Configure email notification*

*Running the tests and viewing their progress*

## Configure email notification

Set up these email settings if you want the application to notify you by email when a test completes or produces an error. Configure email from the Options menu.

1. From the Options menu in the upper right corner, select Email Settings to open the Email Settings dialog box, or click the Preferences tab on the Setup panel.

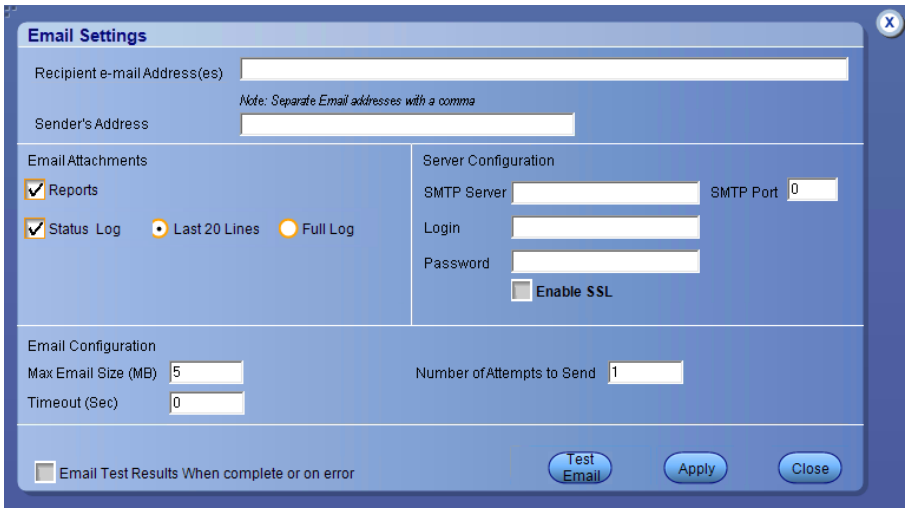



**Figure 19: Email Settings dialog box**

2. (Required) For Recipient email Address(es), enter your email address. You can include multiple addresses as long as you 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 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 Automotive Ethernet.*

6. In the Email Configuration section, select as desired:
   - 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 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** button.
10. Click **Close** button to exit the Email Settings dialog box.

**See also**

*Before you click start*

*Pre-run check list*

*Running the tests and viewing their progress*

## Running the tests and viewing their progress: the Status panel

Once you have configured the tests and gone through the *Pre-run check list*, from any screen, click the green **Start** button. The application acquires and analyzes the data, then displays a report when the tests are completed.

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 using the Alt+Tab key combination. To keep the TekExpress Automotive Ethernet application on top, select the **Keep On Top** from the Options menu in the upper right corner.

## Viewing 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. You can choose from the following two views by selecting the named tab, even while a test is in progress:

### Test Status view

**Test Status tab**

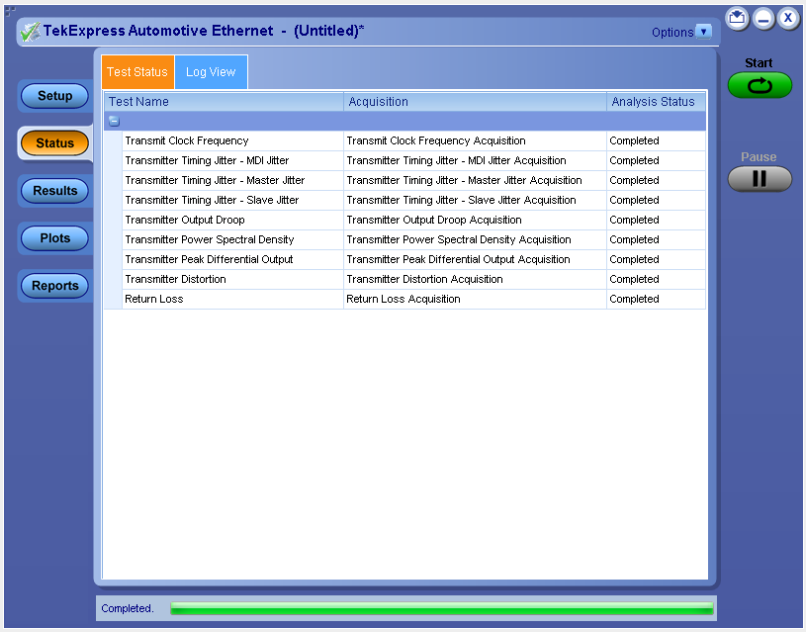



**Figure 20: Test status tab**

The Test Status tab presents a collapsible table with information about each test as it is running. To collapse and expand the table rows, click the expand (+)/(−) collapse button.

**Table 13: Status tab table**

| Column | Description |
|---|---|
| Test Name | Name of the test |
| Acquisition | Describes the type of data being acquired |
| Acquire Status | Progress of the acquisition:<br>■ To be started<br>■ Started Acquisition<br>■ Completed Acquisition |
| Analysis Status | Progress state of the analysis:<br>■ To be started<br>■ In Progress<br>■ Completed<br>■ Aborted<br>■ Skipped |

## Log View

### Log View tab

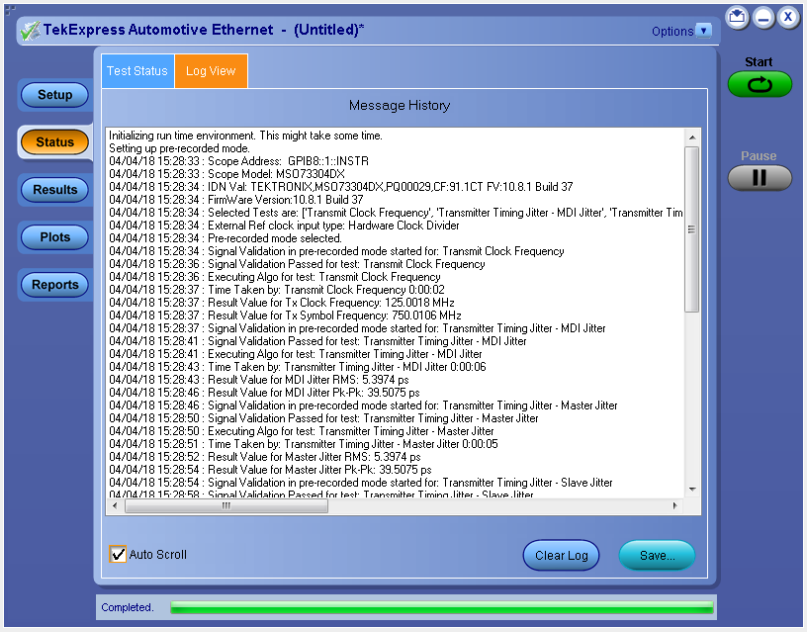



**Figure 21: Log view**

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

**Table 14: Log view options**

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

**See also**

*Before you click start*

*Pre-run check list*

*Configure email notification*

# Viewing plots

## Viewing plots

The Plots panel displays a summary of plot generated during run. The plots have zoom, cursors, save, dock/undock, and select test features.

**Types of plots**

Following are different plots generated during run, based on the measurement(s) selected.

- **Transmit Clock Frequency**

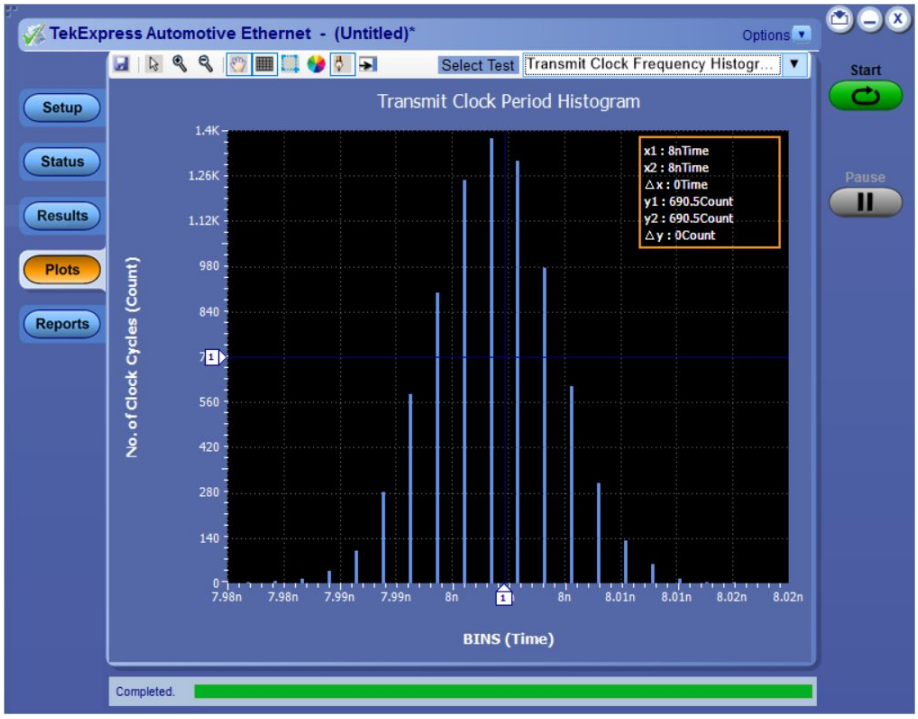



**Figure 22: Transmit Clock Frequency - 1000BASE-T1 Histogram Plot**

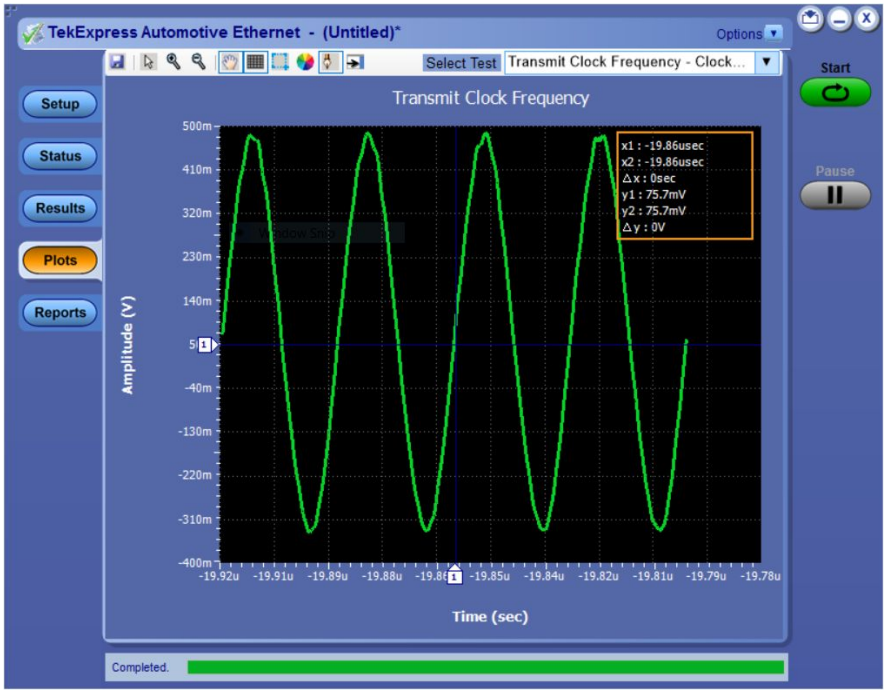



**Figure 23: Transmit Clock Frequency - 1000BASE-T1**

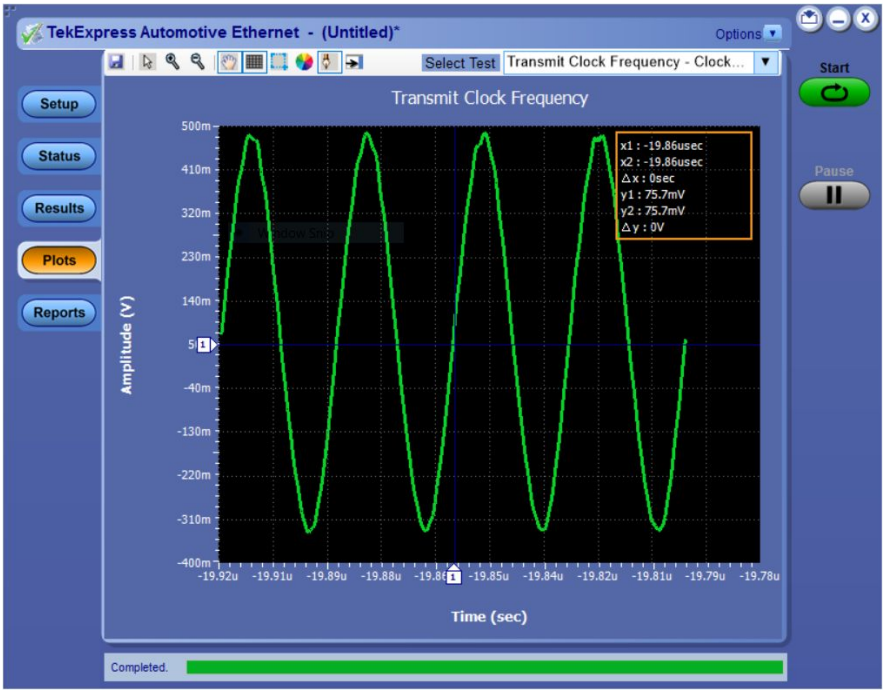



**Figure 24: Transmit Clock Frequency - 100BASE-T1**

- **Transmitter Timing Jitter**

  **Master Jitter**

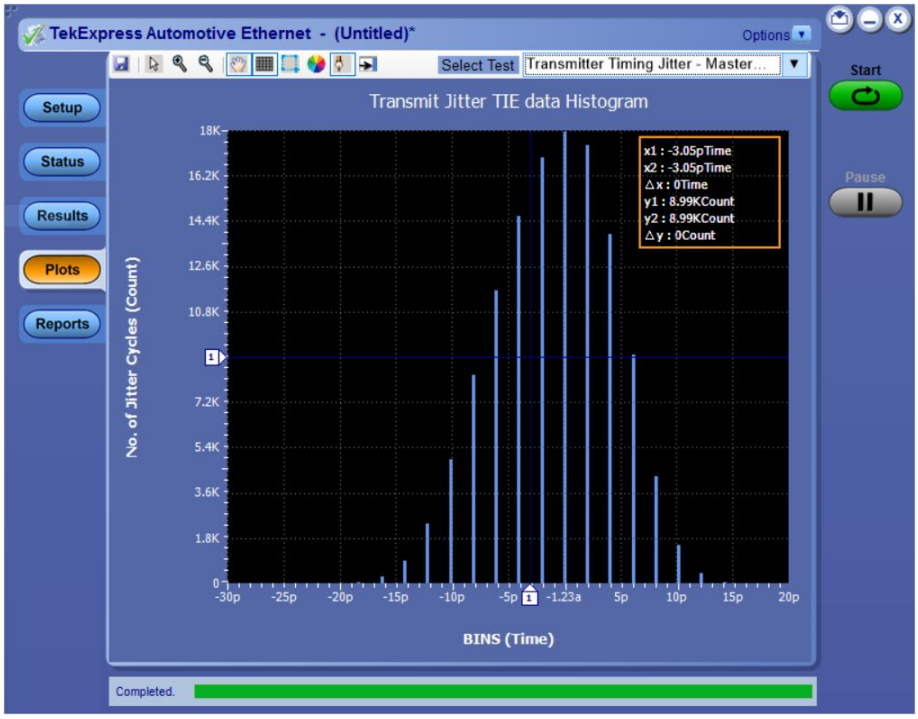



**Figure 25: Master Jitter - 1000BASE-T1 Histogram plot**

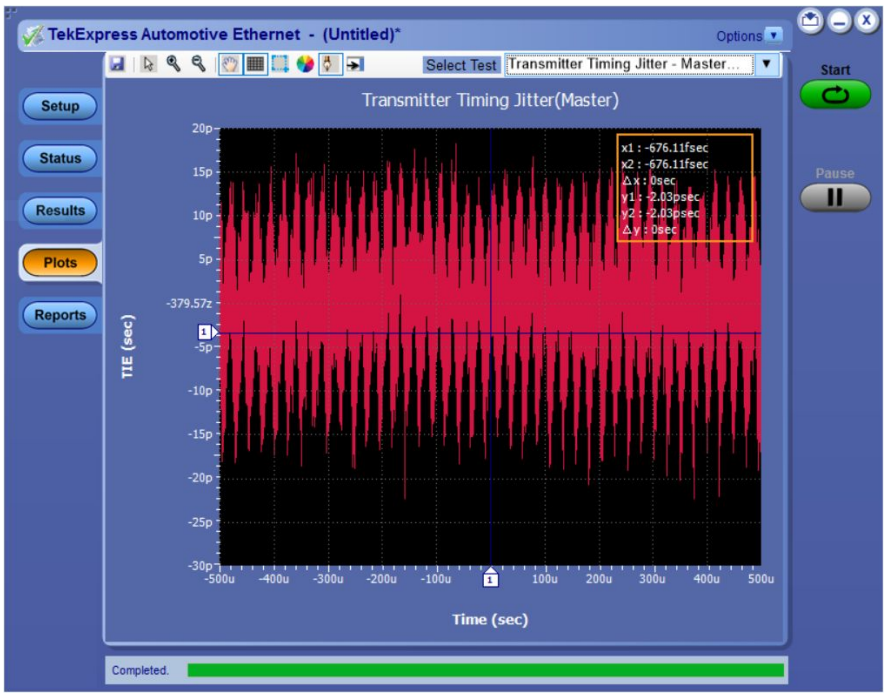



**Figure 26: Master Jitter - 1000BASE-T1**

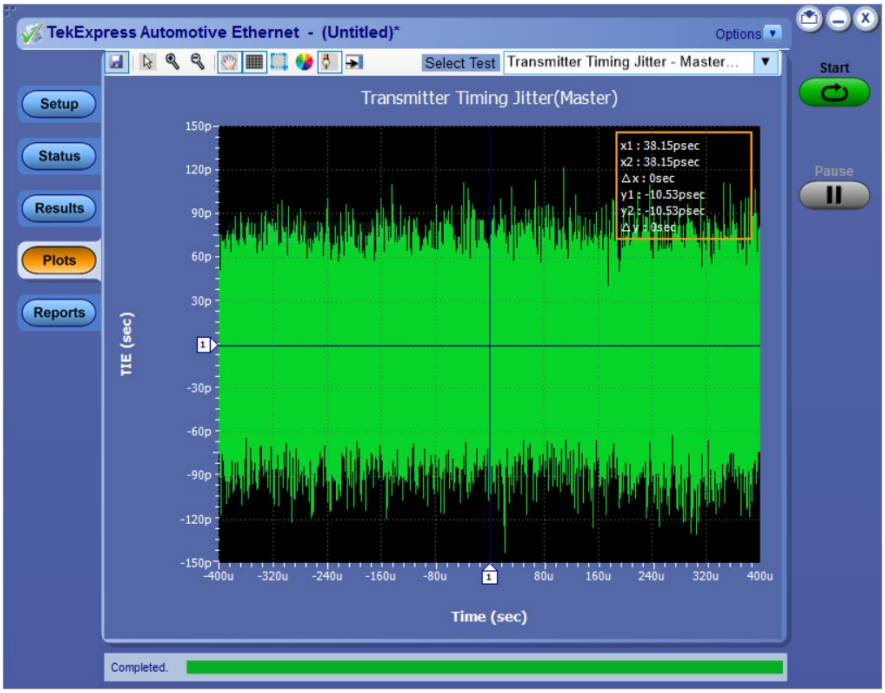



**Figure 27: Master Jitter - 100BASE-T1**

### Slave Jitter

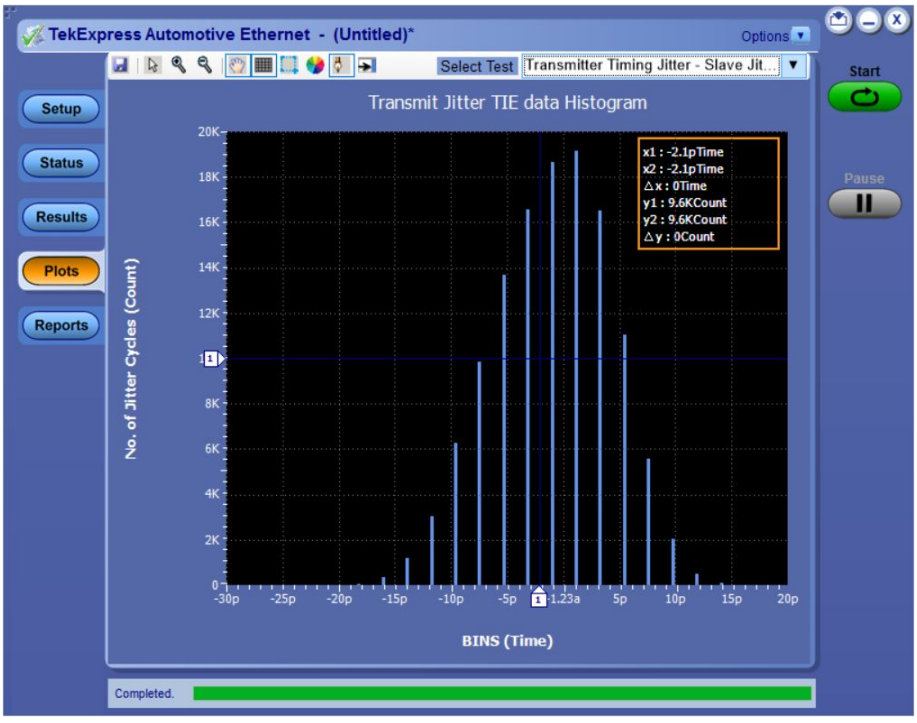



**Figure 28: Slave Jitter - 1000BASE-T1 Slave Histogram plot**

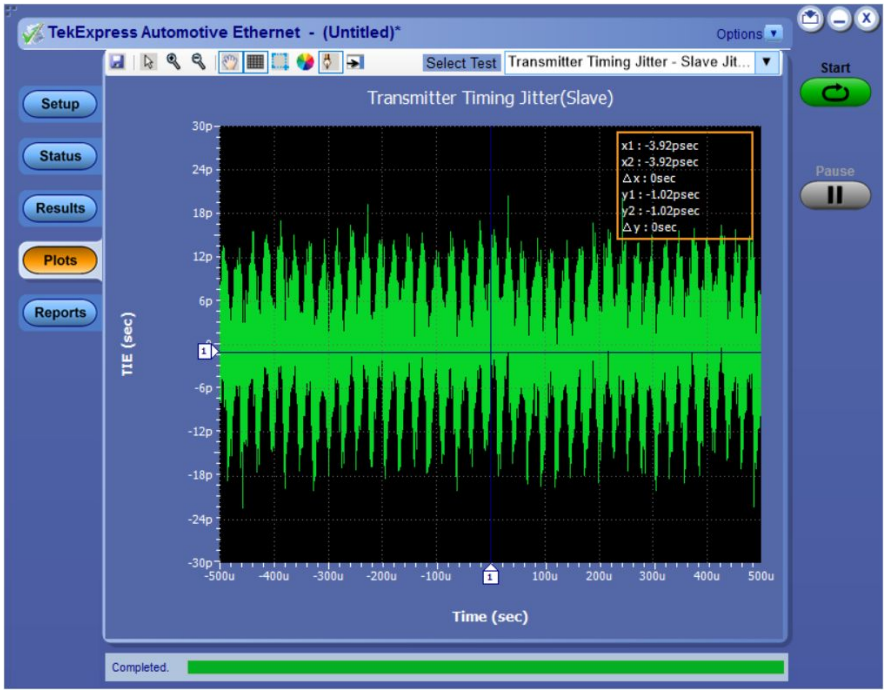



**Figure 29: Slave Jitter - 1000BASE-T1**

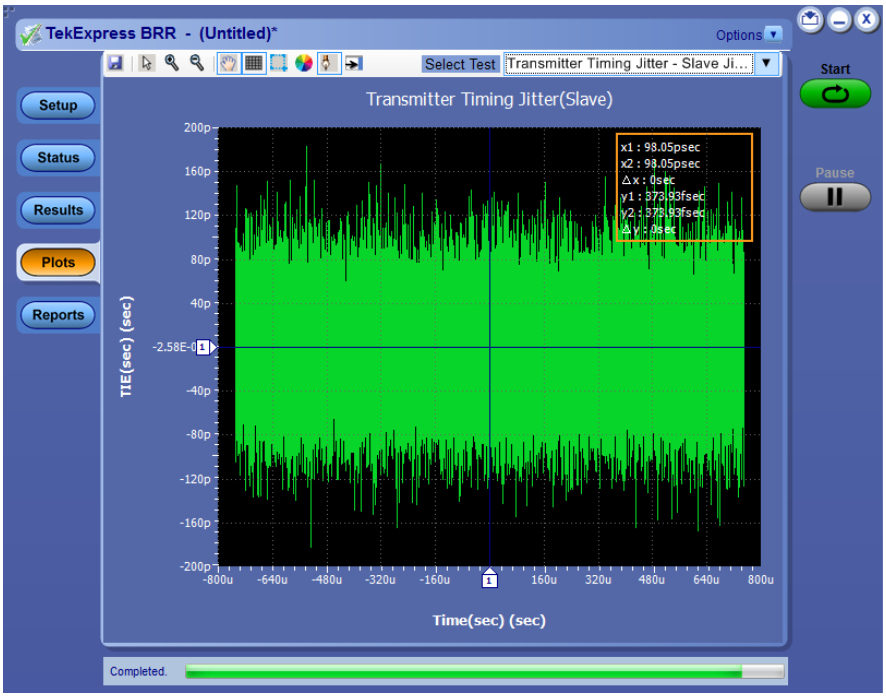



**Figure 30: Slave Jitter - 100BASE-T1**

- **Transmitter Output Droop**

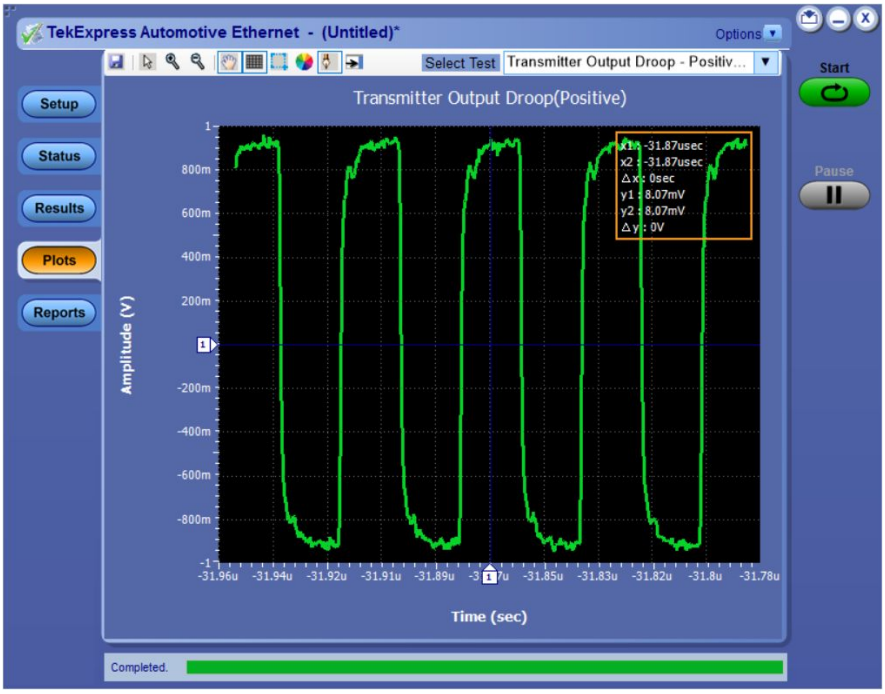



**Figure 31: Transmitter Output Droop (Negative and Positive) - 1000BASE-T1: Positive**

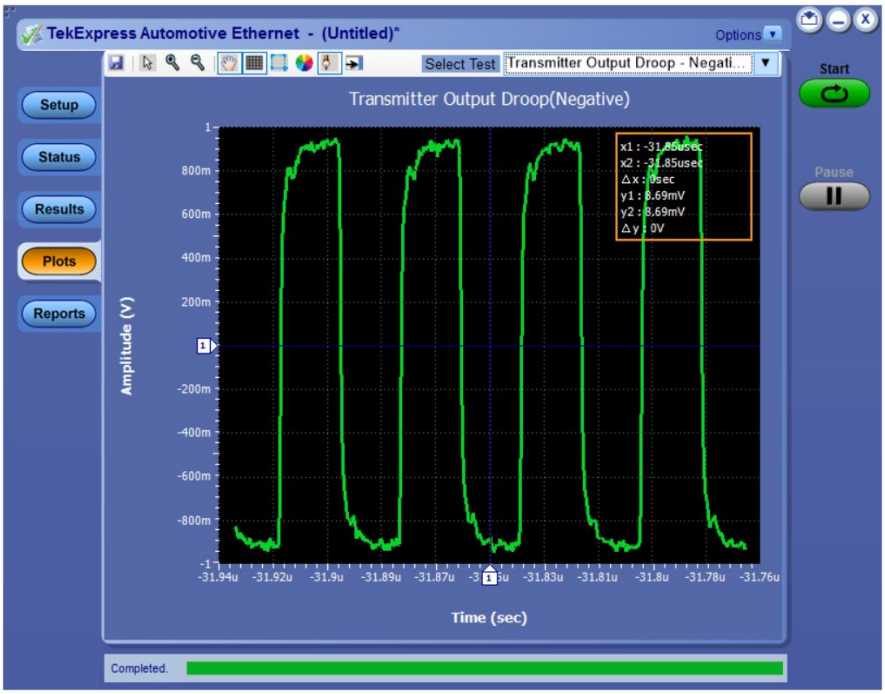



**Figure 32: Transmitter Output Droop (Negative and Positive) - 1000BASE-T: Negative**

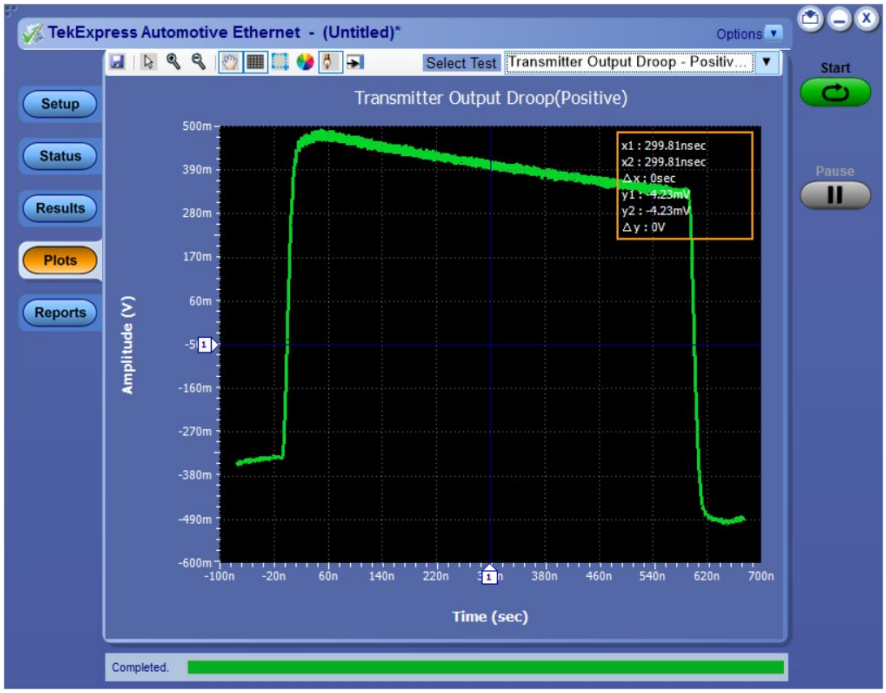



**Figure 33: Transmitter Output Droop (Negative and Positive) - 100BASE-T1: Positive**

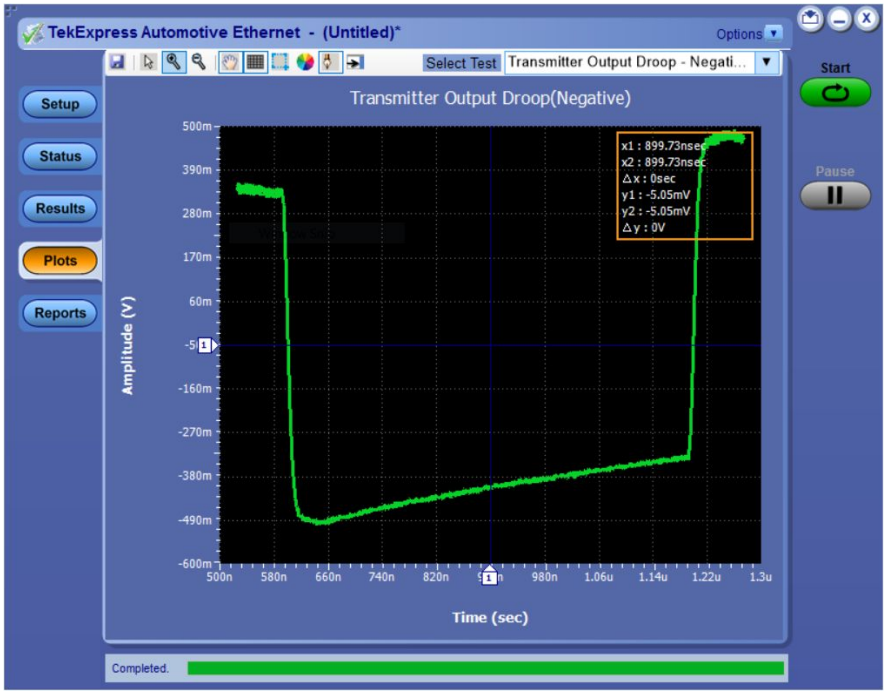



**Figure 34: Transmitter Output Droop (Negative and Positive) - 100BASE-T1: Negative**

- **Transmitter Power Spectral Density**

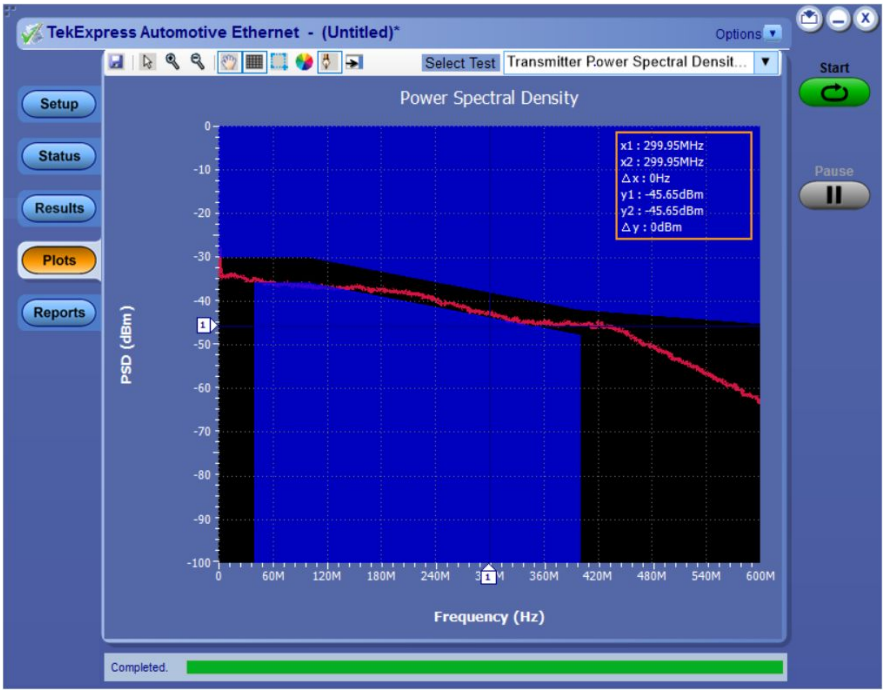



**Figure 35: Transmitter Power Spectral Density - 1000BASE-T1**

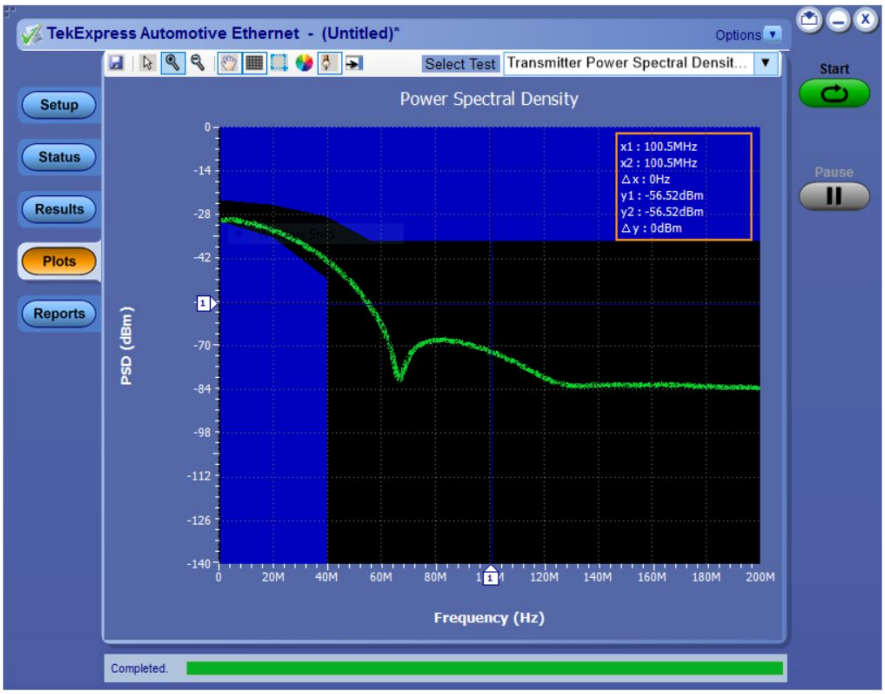



**Figure 36: Transmitter Power Spectral Density - 100BASE-T1**

- **Transmitter Distortion**

  **Error value**

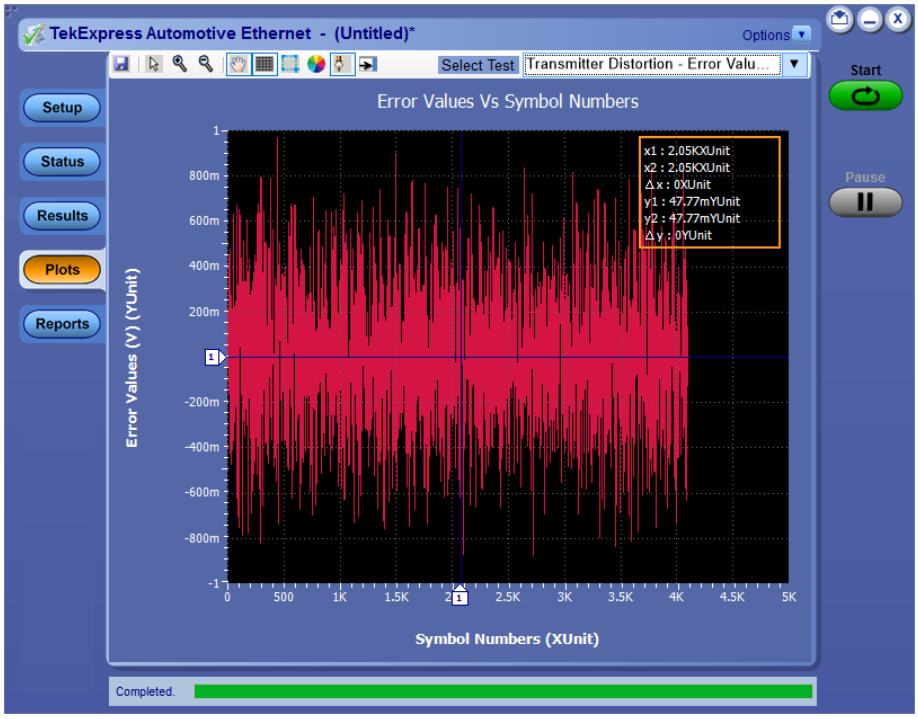



**Figure 37: Error Value - 1000BASE-T1**

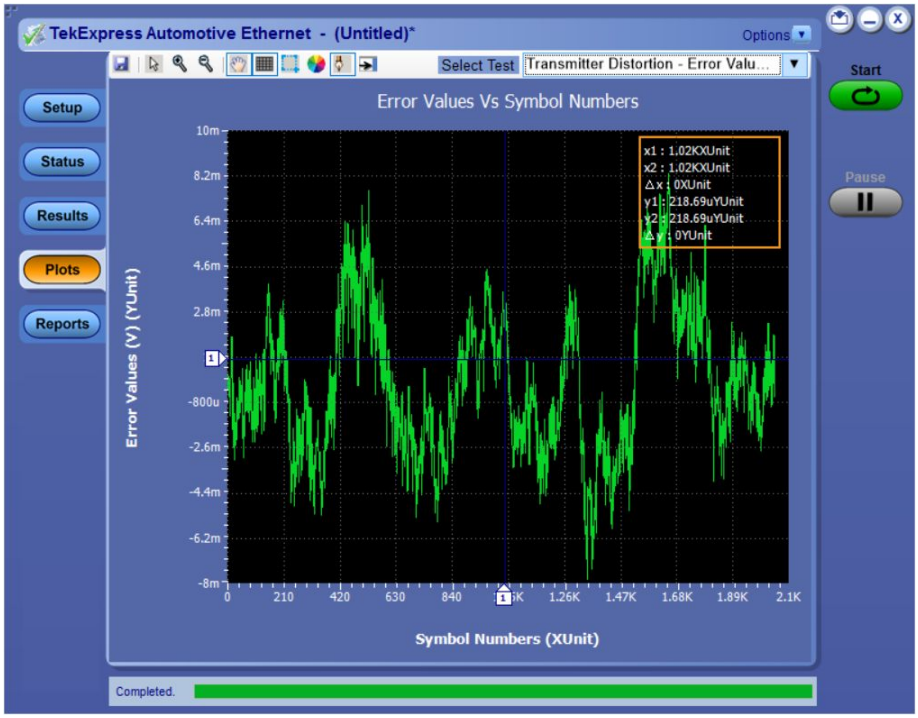



**Figure 38: Error Value - 100BASE-T1**

### Peak Distortion

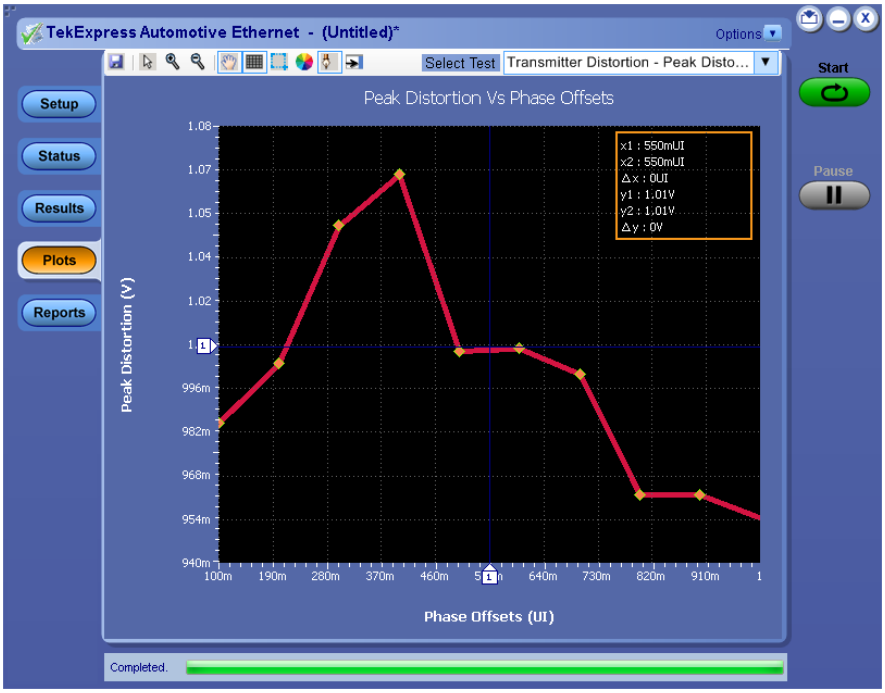



**Figure 39: Peak Distortion - 1000BASE-T1**

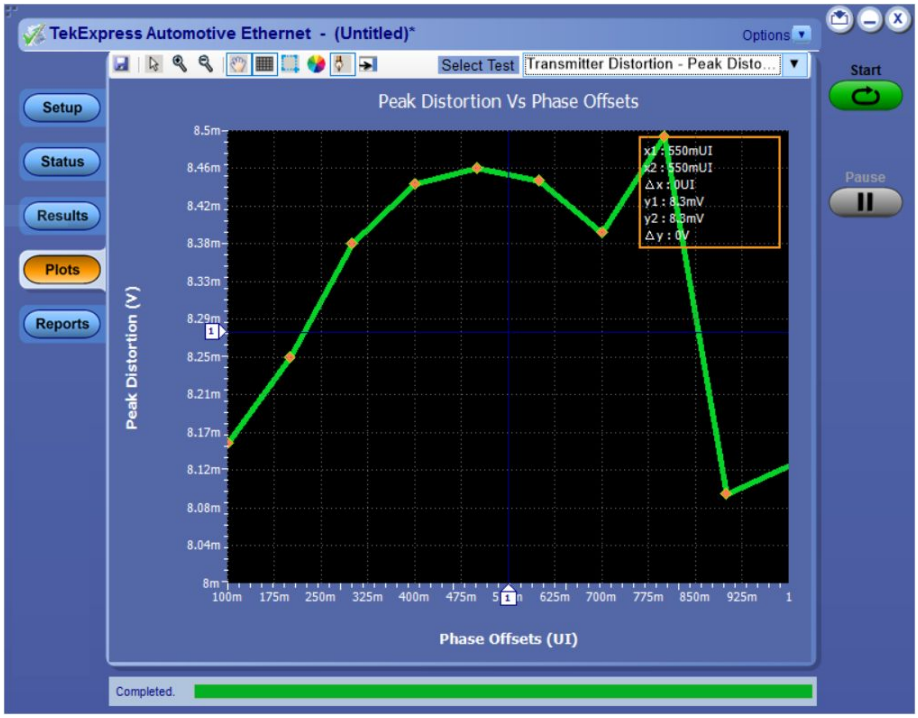



**Figure 40: Peak Distortion - 100BASE-T1**

- **Return Loss**

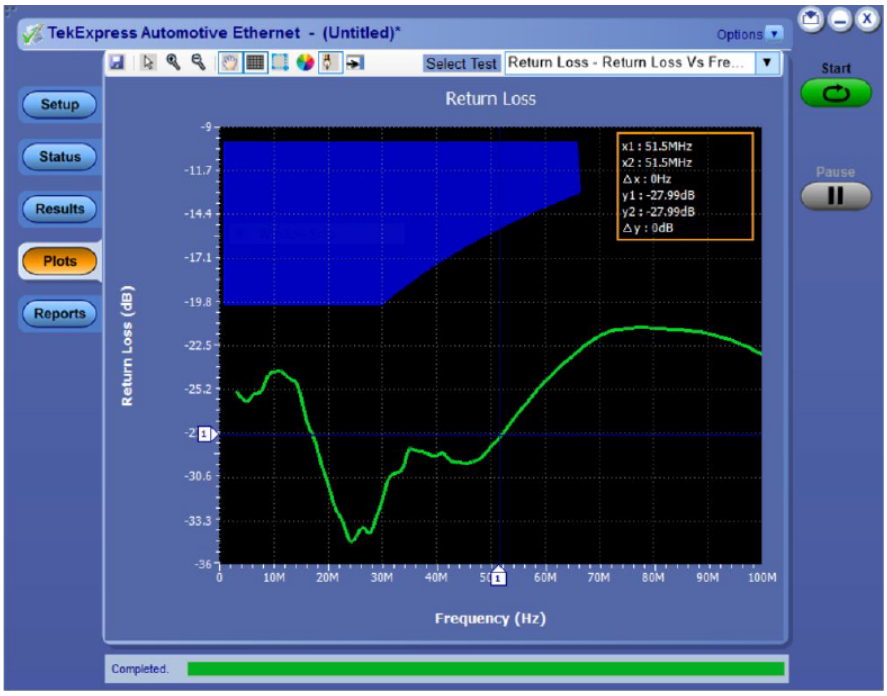



**Figure 41: Return Loss - 1000BASE-T1**

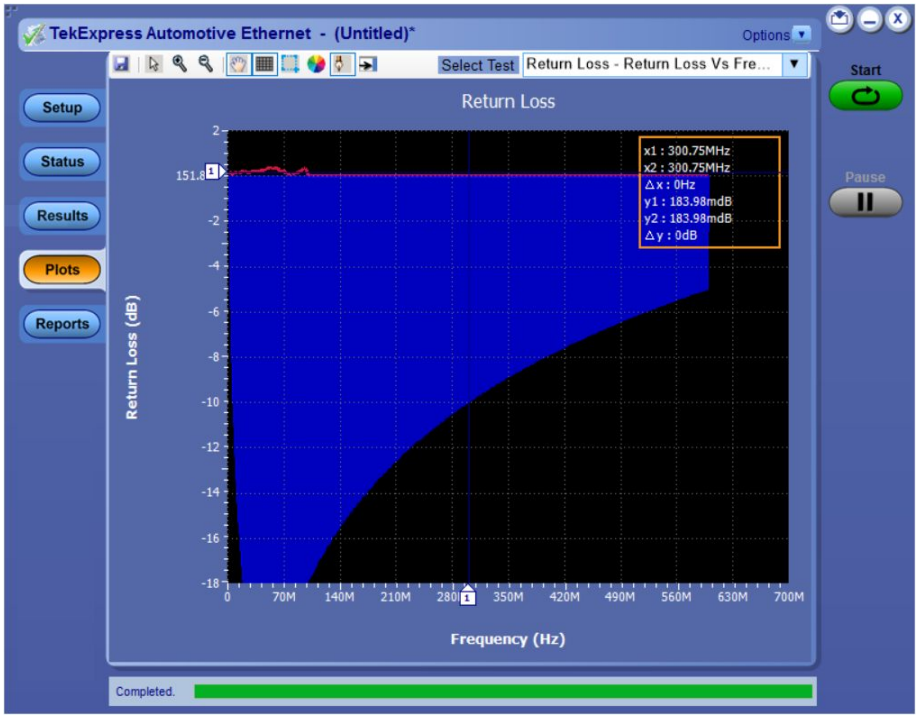



**Figure 42: Return Loss - 100BASE-T1**

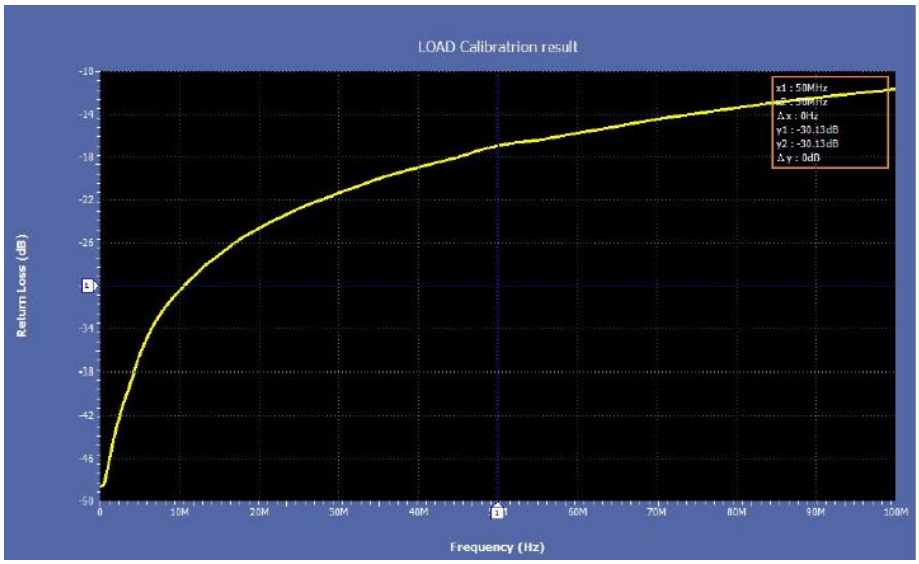



**Figure 43: 100Base-T1 Return Loss Load Calibration**

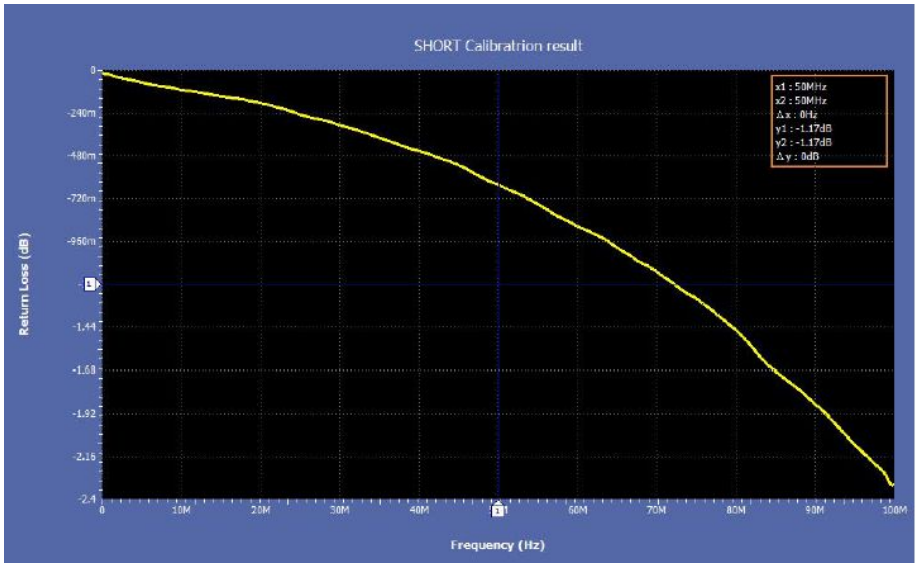



**Figure 44: 100BASE-T1 Return Loss Short Calibration**

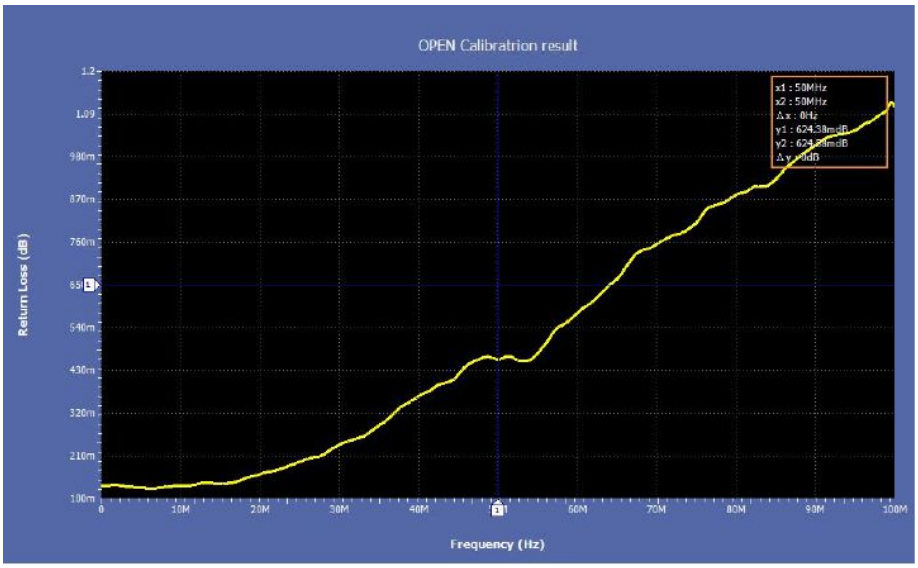



**Figure 45: 100BASE-T1 Return Loss Open Calibration**

**See also** *Known limitations*

# Viewing results

## Viewing test results: the Results panel

When a test completes running, the application switches to the Results Panel.

Set result viewing preferences from the Preferences menu in the upper right corner.

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. You can change the display view.

- To expand all test rows listed, from the **Preferences** menu in the upper right corner, select **View Results Details**.
- To expand a collapsed test row, click the plus button ( ) to the left of the test row.
- To collapse all expanded test rows, select **Preferences > View Results Summary**.
- To collapse a single expanded test row, click the minus button ( ) to the left of the test row.
- 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 one 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** 

button.

### Results preferences


**Figure 46: Results preferences**

**See also** *Setting up tests the Setup panel*

*Running the tests and viewing their progress: the Status panel*

# Configuring and viewing reports

## Configuring and viewing reports: the Reports panel

Use the Report panel to browse for name, and save reports, select report viewing options, and to view reports.

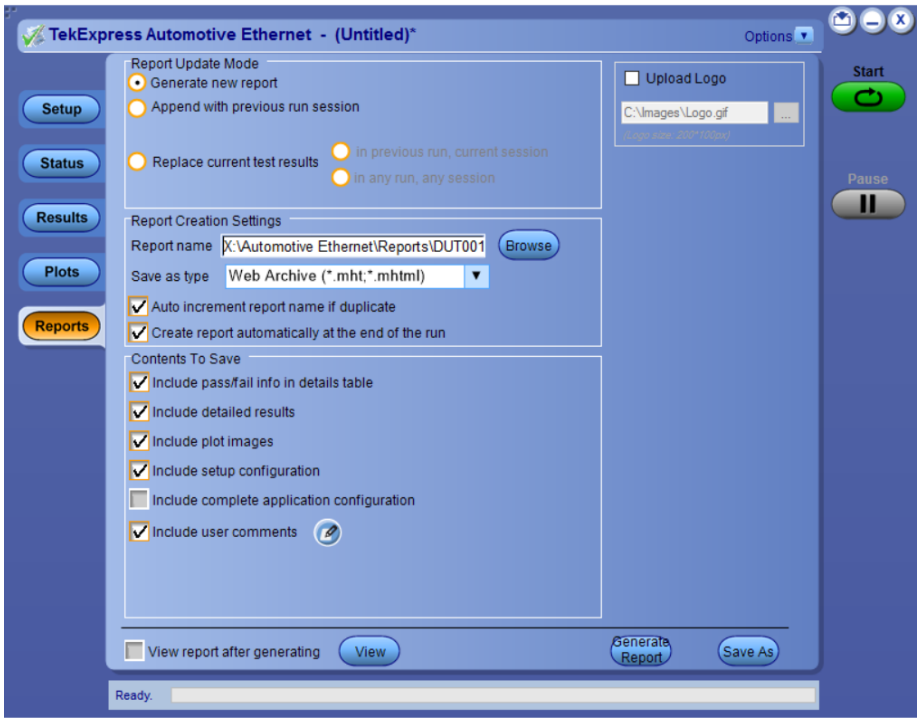



**Figure 47: Reports panel**

Naming a report

Use the Reports Panel to select the naming convention to use for the report. By default, the test report file is located in the session folder to which it belongs, and gets overwritten each time you run the test under the same device name if you don't change the test report name before running the test.

If you do not want your test results to be overwritten each time you run any test, always give the report a unique name, or select to have the 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.

Selecting report options

- Report name

Displays the default name and location where the report will be saved when generated.

To change the report name or location, type over the current folder path and name and then save the test setup. Be sure to include the entire folder path, the file name, and the file type. For example: C:\Documents and Settings\your user name \Documents\My TekExpress\Automotive Ethernet\DUT001_group1.mht.

To open an existing report, click **Browse** button, locate the report file and then click **View** button.

***NOTE.*** *You cannot change the file location using the Browse button.*

- Save as type

Saves a report in a file type different from the default. Lists supported file types to choose from (Web Archive (*mht,*mhtm), PDF (*pdf) ), CSV (*csv) ) .

***NOTE.***

- *The CSV file will open Excel, if available. Otherwise, it will open in Notepad.*
- *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

- 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.

- Append reports

This option adds new report data to the end of an existing report of the same name. This option is deselected by default.

- View report after generating

Automatically opens the report in your Web browser when the test completes. This option is selected by default.

***NOTE.*** *If you unchecked this option before running a test but now would like to view the report, then when analysis is complete, click the* ***Browse*** *button at the top of the Reports panel and navigate to the report file.*

Viewing a report

You can view any report by locating and opening the report file, which ends in .mht unless you changed the file type before running the report.

***NOTE.*** *The application automatically zooms the test report on the scope along with waveform details, after completion of test.*

The top of the report displays information about the instruments and probes used, the duration of the test, software versions, and some summary test information. Below that is a table that shows the test name, measurement details, various measurements, test result (pass/fail), Compliance Mode status (Yes/No), and analysis time. Additional test parameter information that does not fall into the other columns is put in the Comments column.

Report example

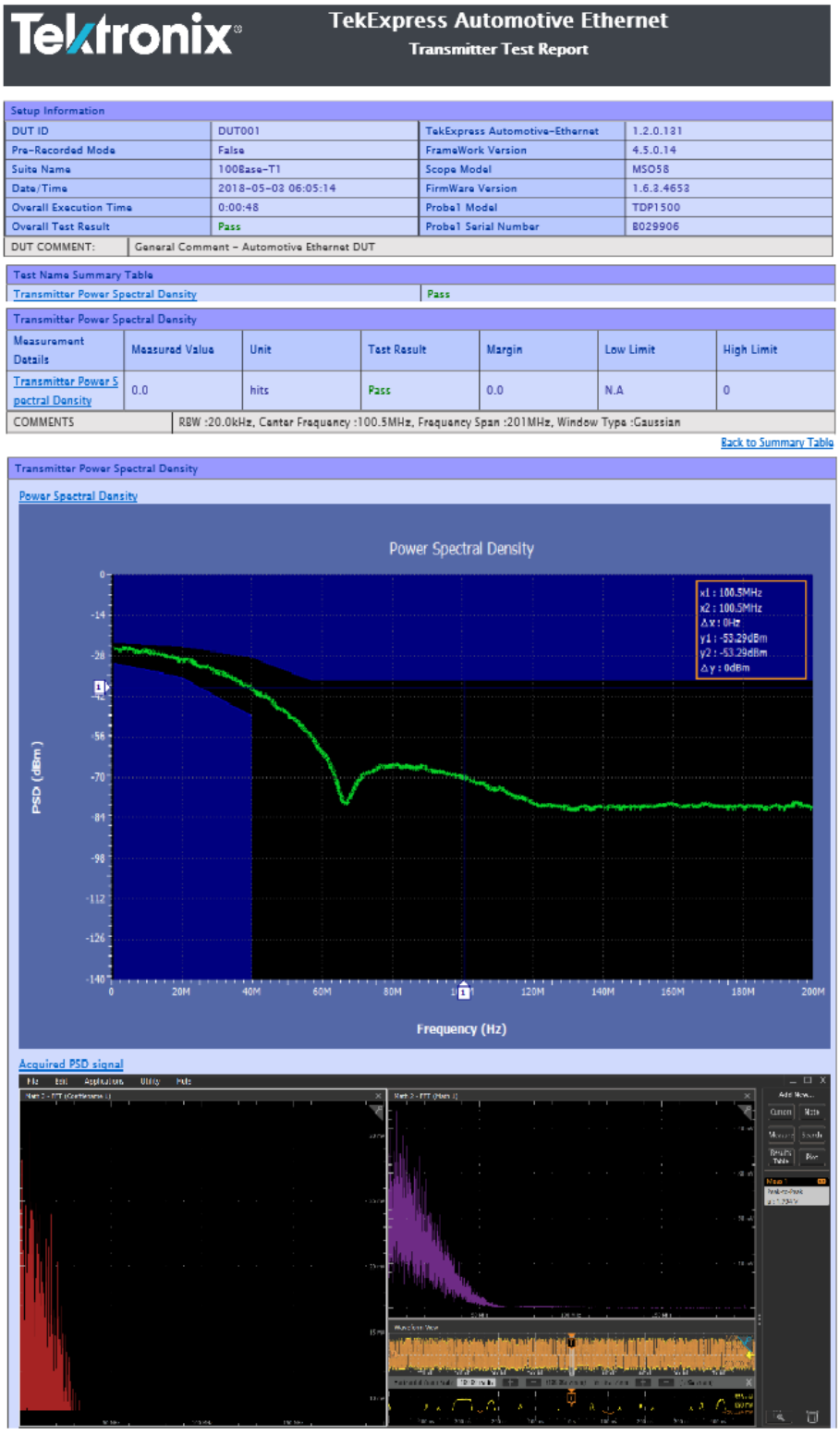



**Figure 48: Report example**

**See Also** *Setting up tests: the Setup panel*

# Saving and recalling test setups

## About saving and recalling test setups

TekExpress Automotive Ethernet application opens with the default test setup selected. Any time you want to create a new test setup, you can select the default test setup to clear the previous test setup selections and take the settings back to their defaults.

You can run a test before or after saving a setup. When you save a setup, the selected oscilloscope, general parameters, acquisition parameters, measurement limits, prerecorded waveform files (if applicable), test, and other configuration settings are all saved under the setup name. You can open a setup and click **Start** button without having to do any other setting up except ensuring that the oscilloscope is detected and ready. For details, see *Before you click start*.

**See Also**

*Save a test setup*

*Recall a saved test setup*

*Create a new test setup based on an existing one*

*Delete a test setup*

*Setting up tests: the Setup panel*

## Saving a test setup

Save a test setup before or after running a test using the parameters you want saved. You can create a new setup from any setup you have open 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 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 ensure that it captures the information you want. If it does not, edit the parameters.
5. From the Options menu, select **Save Test Setup**.
6. Name the test and, then click **Save** button.

**See Also**

*About test setups*

*Recall a saved test setup*

*Create a new test setup based on an existing one*

*Delete a test setup*

## Recalling 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** button.

**See also**

*About Test Setups*

*Save a Test Setup*

*Create a New Test Setup Based on an Existing One*

*Delete a Test Setup*

## Creating 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*

*Save a Test Setup*

*Recall a Saved Test Setup*

*Delete a Test Setup*

## Deleting a test setup

If you no longer need a test setup, delete it from the test setup list in the Options menu using these instructions.

Each saved test setup consists of two main parts, the test setup file and the test setup folder, both named for the test session.

1. Ensure the setup you want to delete is not currently selected in TekExpress Automotive Ethernet.
2. Navigate to the Automotive Ethernet folder where test setup files are stored. For example, X:\Automotive Ethernet\(test setup name).
3. Locate the test setup file, and then delete it.

   This removes the setup from the list in the Options menu.
4. Locate the test setup folder. If you want to keep any of the session files, move them out of the test setup folder and then delete the test setup folder.

**See also**

*About test setups*

*Save a test setup*

*Recall a saved test setup*

*Create a new test setup based on an existing one*

# Automotive Ethernet measurements

## Transmitter Clock Frequency

Transmitter Clock Frequency measurement is applicable for both 100BASE-T1 and 1000BASE-T1

### Test setup for Transmitter Clock Frequency

### Specification

**1000BASE-T1**

- Physical Layer Specifications (IEEE Std 802.3bp™-2016) a parameter for 1 Gb/s Operation over a Single Twisted-Pair Copper Cable section 97.5.3.6 Transmitter clock frequency test

**100BASE-T1**

- BroadR-Reach Physical Layer Transceiver Specification for Automotive Applications, version 3.2, section 5.1.3 or IEEE P802d3bw™ (100 BASE T1) section 96.5.4.5 transmitter specifications
- BroadR-Reach Physical Media Attachment Test Suite version 2.0, section 5.5

### Required test equipment for 1000BASE-T1/100BASE-T1

In addition to the DUT and oscilloscope, you will need the following:

- One supported differential probe.
- Short RJ45 connector
- Test fixture: TF-XGbT

### Test setup procedure for Transmitter Clock Frequency for 1000BASE-T1/100BASE-T1 using TF-XGbT fixture

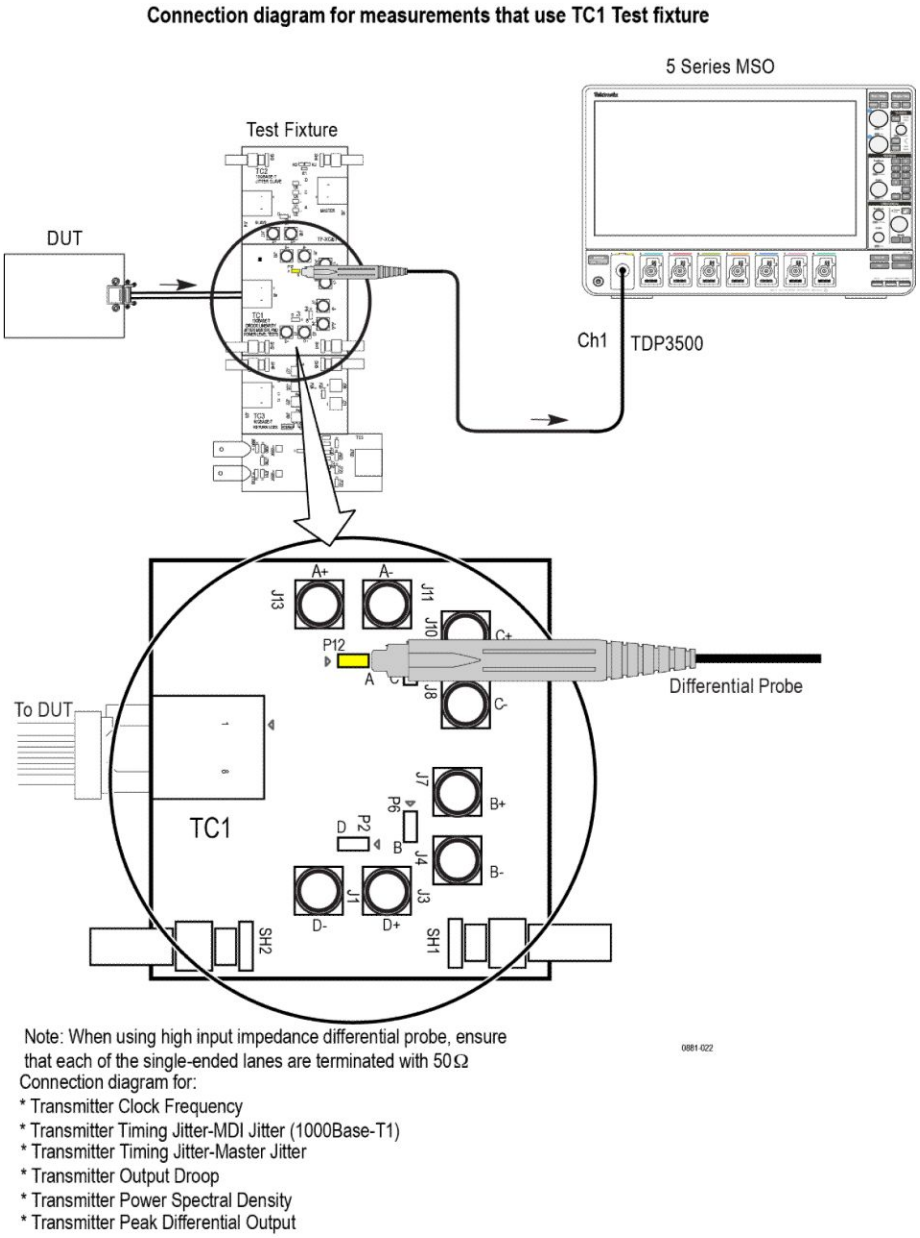
The Transmitter Clock Frequency measurement setup involves the DUT, test fixture, and the oscilloscope.

Connect the equipment as shown in the below connection diagram and as explained in the following procedure:

1. Select the Suite type on the DUT panel.
2. Set the DUT to generate and transmit test mode 2 signal for 1000BASE-T1 and 100BASE-T1.
3. Connect the DUT Ethernet cable to J5 Ethernet port (TC1 segment) of TF-XGbT test fixture.
4. Connect the differential probe to P12 and the configured channel of the oscilloscope and also terminate the SMA pins J11 and J13 with 50 terminators.

## Test setup procedure

**Figure 49: Connection diagram for Test setup procedure for Transmitter Clock Frequency, Transmitter Timing Jitter-MDI Jitter, Transmitter Timing Jitter-Master Jitter, Transmitter Timing Jitter-Slave Jitter, Transmitter Output Droop, and Transmitter Power Spectral Density**

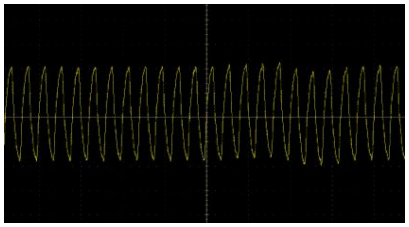
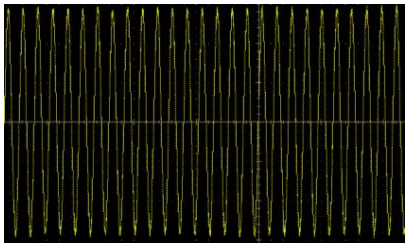



***NOTE.***

*Ensure that the "+" on the probe tip aligns with ▶ on the text fixture board. This takes care of the polarity being not reversed.*

## Measurement setup and algorithm for 1000BASE-T1/100BASE-T1

**Table 15: Measurement setup and algorithm**

| Item | Requirements |
|---|---|
| Configuration parameters | 1. Record Length |
| Signal type | Test mode 2 for 1000BASE-T1 (125 MHz +/- 100 ppm)<br>  <br>Test mode 2 for 100BASE-T1(66.667 MHz +/- 100 ppm)<br>  |
| Measurement algorithm outputs | **100BASE-T1**:<br>■ Clock frequency<br>■ Number of unit intervals<br>**1000BASE-T1**<br>■ Transmit clock frequency<br>■ Transmit symbol frequency<br>■ Number of unit intervals |
| Measurement algorithm inputs | Test mode 2 signal captured in differential form |

| Item | Requirements |
|---|---|
| Limits | **1000BASE-1**:<br>Transmit Clock Frequency<br>■ Lower limit: 124.9875 MHz<br>■ Upper limit: 125.0125 MHz<br>Transmit symbol frequency<br>■ Lower limit: 749.9250 MHz<br>■ Upper limit: 750.0750 MHz<br>**100BASE-T1**<br>■ Lower limit: 66.6603 MHz<br>■ Upper limit: 66.6736 MHz |
| Plots | *Plot* showing multiple cycles of clock frequency and Transmit Clock Period Histogram plot (for 100BASE-T1 only) |

## Measurement algorithm

TekExpress Automotive Ethernet Compliance Solution automatically executes the calculations described below for the Transmit Clock Frequency measurement.

**1000BASE-T1**

1. Configure DUT to Test mode 2. The test mode 2 signal is a clock signal where the PHY shall transmit a continuous pattern of three {+1} symbols followed by three {–1} symbols with the transmitted symbols timed from its local clock source of 750 MHz. 1000BASE-T1 PHY shall provide access to a frequency reduced version of the transmit symbol clock or TX_TCLK125. This 125 MHz test clock is one sixth frequency divided version of TX_TCLK that times the transmitted symbols.
2. Compute and verify the clock frequency between the High and Low limits as mentioned in the table above.

**100BASE-T1**

1. Configure DUT to Test mode 2. The test mode 2 signal is a clock signal where the PHY shall transmit the data symbol sequence (+1, -1) repeatedly on all channels.
2. Compute and verify the clock frequency between the High and Low limits as mentioned in the table above.

**See also** *Plots*

## Transmitter Timing Jitter-MDI Jitter

Transmitter Timing Jitter_MDI Jitter measurement is applicable for 1000BASE-T1 Suite.

**Transmitter Timing Jitter-MDI Jitter**

**Specification**

**1000BASE-T1**

- Physical Layer Specifications (IEEE Std 802.3bp™-2016) a parameter for 1 Gb/s Operation over a Single Twisted-Pair Copper Cable section 97.5.3.3 Transmitter Timing Jitter-MDI Jitter

**Required test equipment**

In addition to the DUT and oscilloscope, you will need the following:

**1000BASE-T1**

- One supported differential probe.
- Short RJ45 connector
- Test fixture: TF-XGbT

**Test setup procedure for Transmitter Clock Frequency for 1000BASE-T1 using TF-XGbT fixture**

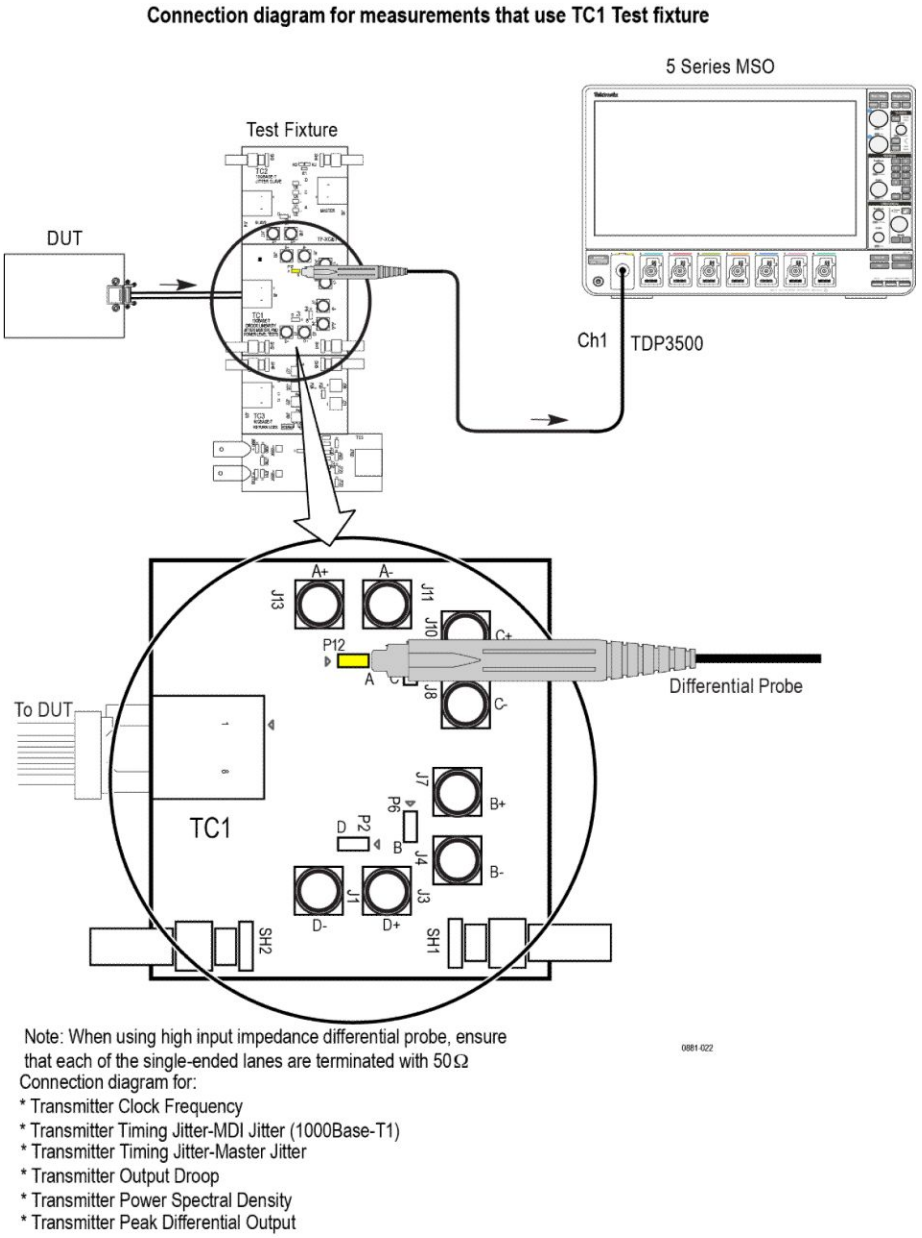
The Transmitter Clock Frequency measurement setup involves the DUT, test fixture, and the oscilloscope.

Connect the equipment as shown in below connection diagram and as explained in the following procedure:

1. Select the Suite type on the DUT panel.
2. Set the DUT to generate and transmit test mode 2 signal for 1000BASE-T1.
3. Connect the DUT Ethernet cable to J5 Ethernet port (TC1 segment) of TF-XGbT test fixture.
4. Connect the differential probe to P12 and the configured channel of the oscilloscope and also terminate the SMA pins J11 and J13 with 50 Ω terminators.

## Test setup procedure

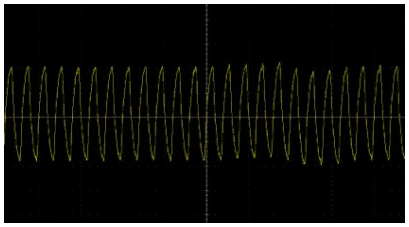
**Figure 50: Connection diagram for Test setup procedure for Transmitter Clock Frequency, Transmitter Timing Jitter-MDI Jitter, Transmitter Timing Jitter-Master Jitter, Transmitter Timing Jitter-Slave Jitter, Transmitter Output Droop, and Transmitter Power Spectral Density**




***NOTE.*** *Ensure that the "+" on the probe tip aligns with ▶ on the text fixture board. This takes care of the polarity being not reversed.*

## Measurement setup and algorithm

**Table 16: Measurement setup and algorithm**

| Item | Requirements |
|---|---|
| Configuration parameters | 1. Record Length<br>2. Edge Type<br>3. Hysteresis |
| Signal type | Test mode 2<br>  |
| Measurement algorithm outputs | ■ MDI Jitter RMS<br>■ MDI Jitter peak to peak |
| Measurement algorithm inputs | Test mode 2 signal captured in differential form |
| Limits | MDI Jitter RMS:<br>■ Lower Limit: NA<br>■ Upper Limit: 5 pico seconds<br>MDI Jitter peak to peak:<br>■ Lower Limit: NA<br>■ Upper Limit: 50 pico seconds |
| Plots | *Plot* showing Transmit Timing Jitter-MDI Master Jitter, Transmit MDI Jitter TIE data Histogram |

## Measurement algorithm

TekExpress Automotive Ethernet Compliance Solution automatically executes the calculations described below for the measurement Transmitter Timing Jitter-MDI Jitter.

1. Test mode 2 is for Transmitter Jitter testing on MDI when transmitter is in MASTER timing mode. When test mode 2 is enabled, the 1000BASE-T1 PHY shall transmit a continuous pattern of three {+1} symbols followed by three {–1} symbols with the transmitted symbols timed from its local clock source of 750 MHz. The transmitter output is a 125 MHz signal.
2. Compute and verify the clock frequency between the High and Low limits as mentioned in the above table.

**See also** *Plots*

## Transmitter Timing Jitter Master and Slave

This test confirms that the transmitter timing jitter of the PMA is within conformance limits.

**Test setup for (Master Jitter)**

### Specification

**1000BASE-T1**

- Physical Layer Specifications (IEEE Std 802.3bp™-2016) a parameter for 1 Gb/s Operation over a Single Twisted-Pair Copper Cable section 97.5.3.3 Transmitter Timing Jitter-MDI Jitter

**100BASE-T1**

- BroadR-Reach Physical Layer Transceiver Specification for Automotive Applications, version 3.2, section 5.4.3 or IEEE P802d3bwTM (100 BASE T1) section 96.5.4.3 transmitter specifications
- BroadR-Reach Physical Media Attachment Test Suite version 2.0, section 5.3

### Required test equipment

In addition to the DUT and oscilloscope, you will need the following:

**1000BASE-T1/100BASE-T1**

- One supported differential probe.
- Short RJ45 connector
- Test Fixture: TF-XGbT

### Test setup procedure for Timing Jitter (Master) test

The Timing Jitter (Master) measurement setup involves the DUT, test fixture, and the oscilloscope.

Connect the equipment as shown in the below connection diagram and as explained in the following procedure:

1. Select the Suite type on the DUT panel.
2. Set the DUT to generate and transmit a test mode 2 signal for 100BASE-T1 and test mode 1 signal for 1000BASE-T1.
3. Connect the DUT Ethernet cable to J5 Ethernet port (TC1 segment) of TF-XGbT test fixture.
4. Connect the differential probe to P12 and the configured channel of the oscilloscope and also terminate the SMA pins J11 and J13 with 50 Ω terminators.

## Test setup (Slave Jitter)

### Specifications

**1000BASE-T1**

- Physical Layer Specifications (IEEE Std 802.3bp™-2016) a parameter for 1 Gb/s Operation over a Single Twisted-Pair Copper Cable section in 97.5.3.3 Transmitter Timing Jitter-MDI Jitter

**100BASE-T1**

- BroadR-Reach Physical Layer Transceiver Specification for Automotive Applications, version 3.2, section 5.4.3 or IEEE P802d3bwTM (100 BASE T1) section 96.5.4 transmitter specifications
- BroadR-Reach Physical Media Attachment Test Suite version 2.0, section 5.3

### Required test equipment

In addition to the DUT and oscilloscope, you will need the following:

**1000BASE-T1/100BASE-T1**

- One supported differential probe
- Short RJ45 connector
- Test fixture: TF-XGbT
- Link partner (This is needed to put the DUT to slave mode)

### Test setup procedure for Timing Jitter (Slave) test

The Timing Jitter (Slave) measurement setup involves the DUT, test fixture, link partner, and oscilloscope.

Connect the equipment as shown in this *connection diagram* and as explained in the following procedure:

1. Select the Suite type on the DUT panel
2. Set the DUT to generate and transmit a test mode 1 signal for 1000BASE-T1 and test mode 3 for 100BASE-T1.
3. Connect the DUT to the link partner and establish a normal Ethernet connection.
4. Use the appropriate cable or probe to bring the 100BASE-T1 DUT SLAVE TX_TCLK or 1000BASE-T1 DUT SLAVE TX_TCLK125 to the configured channel of the oscilloscope.

---

***NOTE.*** *You will need to choose either an SMA cable, a BNC cable, or a differential probe based on the interface available on the DUT for TX_TCLK.*

*Slave transmitter timing jitter can only performed when the 100BASE-T1 DUT SLAVE TX_TCLK or 1000BASE-T1 DUT SLAVE TX_TCLK125 is exposed and accessible.*

---

## Test setup procedure

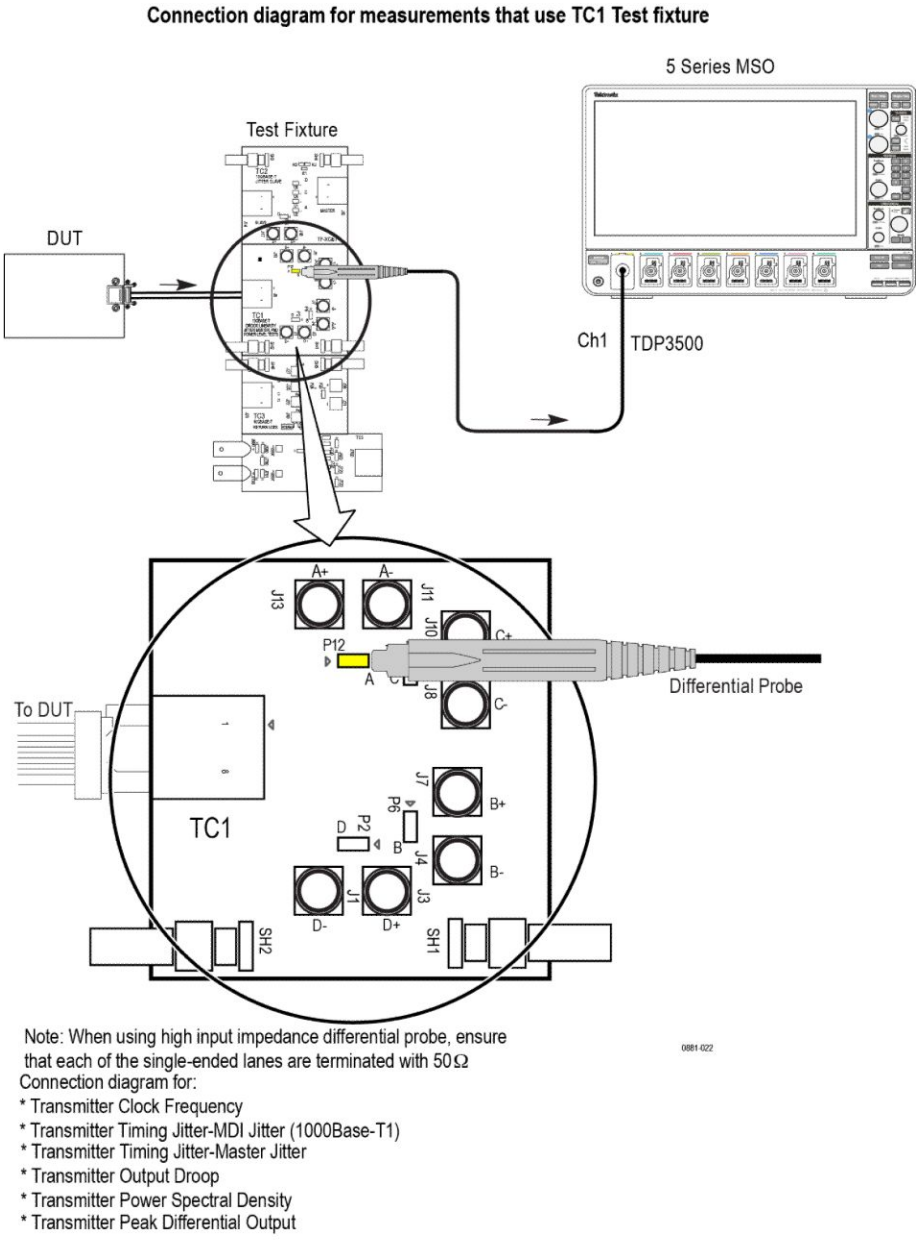



**Figure 51: Connection diagram for Transmitter Clock Frequency, Transmitter Timing Jitter-Master Jitter, Transmitter Timing Jitter-Slave Jitter, Transmitter Output Droop, and Transmitter Power Spectral Density**

---

***NOTE.*** *Ensure that the "+" on the probe tip aligns with ▸ on the text fixture board. This takes care of the polarity being not reversed.*

---

## Timing Jitter (Slave) test

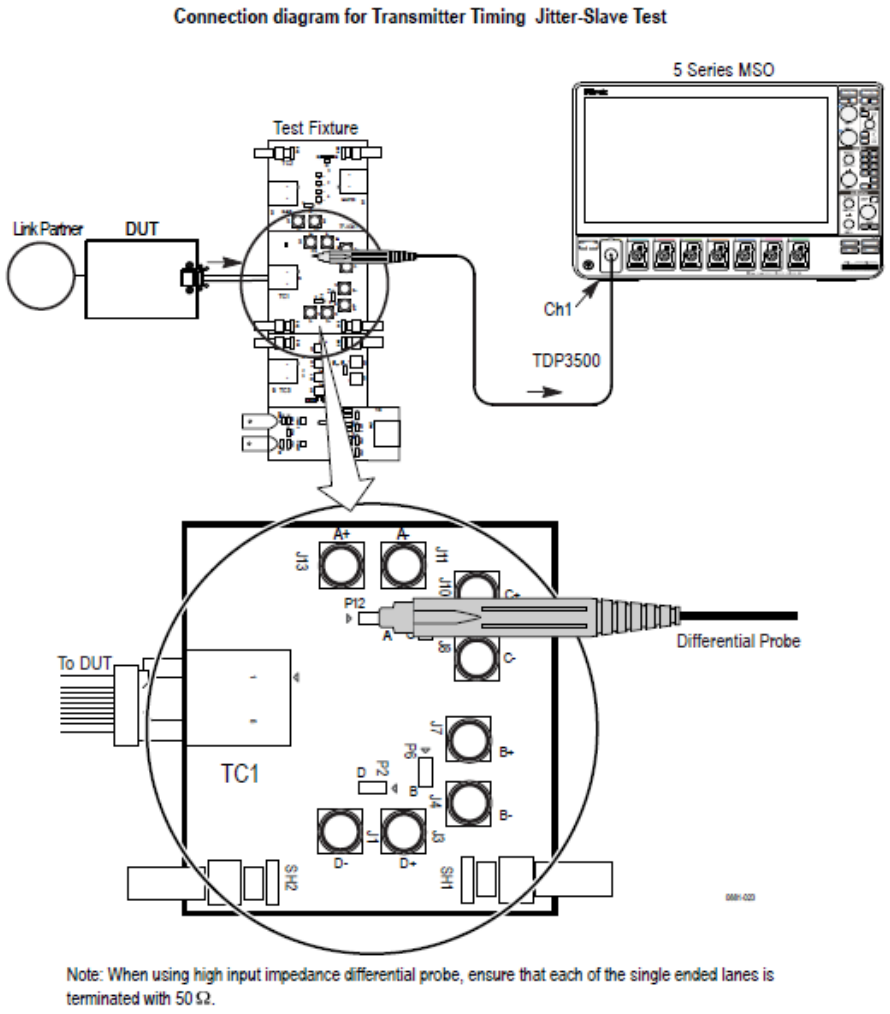
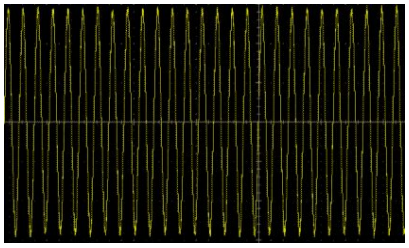
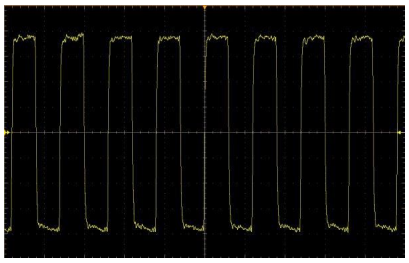
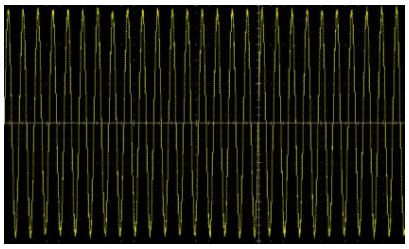
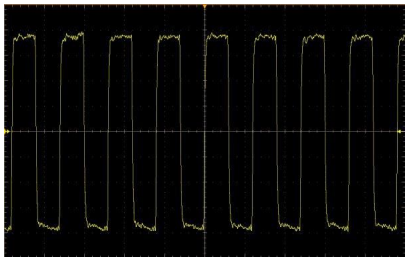



**Figure 52: Connection diagram for Timing Jitter (Slave) test**

## Measurement setup and algorithm

**Table 17: Measurement setup and algorithm**

| Item | Requirements |
|---|---|
| Configuration parameters | Applicable for both 100 and 1000BASET-T1<br>a. Record Type<br>b. Hysteresis<br>c. Edge Type |

| Item | Requirements |
|---|---|
| Signal type | Test mode 1 (master) and SLAVE TK_TCLK125 (slave) for 1000BASE-T1<br>  <br>  <br>Test mode 2 (master) and SLAVE TK_TCLK125 (slave) for 100BASE-T1<br>  <br>  |
| Measurement algorithm outputs | |

| Item | | Requirements |
|---|---|---|
| | MASTER | For 1000BASE-T1:<br>■ Master Jitter RMS: RMS TIE between test mode 1 waveform and unjittered reference<br>■ Master Jitter peak to peak : Peak-to-Peak TIE between test mode 1 waveform and unjittered reference<br>For 100BASE-T1:<br>■ Master Jitter RMS: RMS TIE between test mode 2 waveform and unjittered reference |
| | SLAVE | For 100BASE-T1:<br>■ Slave Jitter RMS: RMS TIE between SLAVE TX_TCLK waveform and unjittered reference<br>For 1000BASE-T1:<br>■ Slave Jitter RMS: RMS TIE between SLAVE TX_TCLK125 waveform and unjittered reference<br>■ Slave Jitter peak to peak: Peak-to-Peak TIE between SLAVE TX_TCLK125 waveform and unjittered reference |
| Measurement algorithm inputs | | |
| | MASTER | For 1000BASE-T1:<br>■ Minimum 1 ms duration test mode 1 signal captured in differential form.<br>■ Hysteresis<br>■ Edge Type<br>For 100BASE-T1:<br>■ Minimum 1 ms duration test mode 2 signal captured in differential form.<br>■ Hysteresis<br>■ Edge Type |

| Item | | Requirements |
|---|---|---|
| | SLAVE | For 1000BASE-T1:<br>■ Minimum 1 ms duration TX_TCLK signal captured in differential form.<br>■ Hysteresis<br>■ Edge Type<br>For 100BASE-T1:<br>■ Minimum 1 ms duration test mode 1 signal captured in differential form.<br>■ Hysteresis<br>■ Edge Type |
| Limits | | |
| | MASTER | For 1000BASE-T1:<br>■ Master Jitter RMS:<br>■ Lower limit: NA<br>■ Upper limit: 50 ps<br>■ Master Jitter peak to peak:<br>■ Lower limit: NA<br>■ Upper limit: 50 ps<br>For 100BASE-T1:<br>■ Lower limit: NA<br>■ Upper limit: 50 ps |

| Item | | Requirements |
|---|---|---|
| | SLAVE | For 1000BASE-T1:<br>■ Slave Jitter RMS:<br>■ Lower limit: NA<br>■ Upper limit: 10 ps<br>■ Slave Jitter peak to peak:<br>■ Lower limit: NA<br>■ Upper limit: 100 ps<br>For 100BASE-T1:<br>■ Slave Jitter peak to peak:<br>■ Lower limit: NA<br>■ Upper limit: 150 ps |
| Plots | | For 1000BASE-T1:<br>■ Master Jitter and Slave Jitter:<br>■ Showing Time trend of TIE, TIE data Histogram<br>For 100BASE-T1:<br>■ Time trend of TIE<br>■ Master Jitter and Slave Jitter |

### Measurement algorithm (Master and Slave)

TekExpress Automotive Ethernet Compliance Solution automatically executes the calculations as described below:

1. Edge locations in test mode 2 (100BASE-T1) / test mode 1 (1000BASE-T1) for Master waveform, SLAVE TX_TCLK (100BASE-T1) / SLAVE TX_TCLK125 (1000BASE-T1) for Slave waveform, (captured in differential form) are determined with middle level percentage as 50%. Hysteresis and EdgeType are entered by user in GUI.
2. The least square method is used for straight line fit, with Y as the edge locations and X as the edge indices. Only edges that match the EdgeType entered by the user in the GUI are considered for fitting. Slope and intercept are determined as the output of the least square method. Using the intercept and slope, reconstructed edge locations are determined (un-jittered reference).

   reconstructedTime(I) = intercept + I * slope, I = 0 to num_edges(num_edges depend on EdgeType)
3. Compute TIE on EdgeType entered by user in GUI.

   TIE(I) = reconstructedTime(I) – EdgePosition(I), I = 0 to num_edges(num_edges depend on EdgeType).

   Compute rms value of TIE, which is reported as the result.

**See also** *Plots*

## Transmitter Power Spectral Density

This test confirms that the transmitter power spectral density is within conformance limits.

**Test setup for Power Spectral Density for 1000BASE-T1**

**Specification**

**1000BASE-T1**

- Physical Layer Specifications (IEEE Std 802.3bp™-2016) a parameter for 1 Gb/s Operation over a Single Twisted-Pair Copper Cable section 97.5.3.4 Transmitter Power Spectral Density

**100BASE-T1**

- BroadR-Reach Physical Layer Transceiver Specification for Automotive Applications, version 3.2, section 5.4.4 or IEEE P802d3bwTM (100 BASE T1) section 96.5.4.4 transmitter specifications
- BroadR-Reach Physical Media Attachment Test Suite version 2.0, section 5.4

**Required test equipment**

In addition to the DUT and oscilloscope, you will need the following:

**1000BASE-T1/100BASE-T1**

- One supported differential probe
- Short RJ45 connector
- Test fixture: TF-XGbT

**Test setup procedure for Power spectral density test for 1000BASE-T1/100BASE-T1**

The Power Spectral Density test involves the DUT, test fixture, and the oscilloscope.

Connect the equipment as shown in this *connection diagram* and as explained in the following procedure:

1. Select the Suite type on the DUT panel.
2. Set the DUT to generate and transmit test mode 5 signal.
3. Connect the DUT Ethernet cable to J5 Ethernet port (TC1 segment) of TF-XGbT test fixture.
4. Connect the differential probe to P12 and the configured channel of the oscilloscope and also terminate the SMA pins J11 and J13 with 50 Ω terminators.
5. Ensure that the "+" point of differential probe should be near ▸.

## Power spectral density

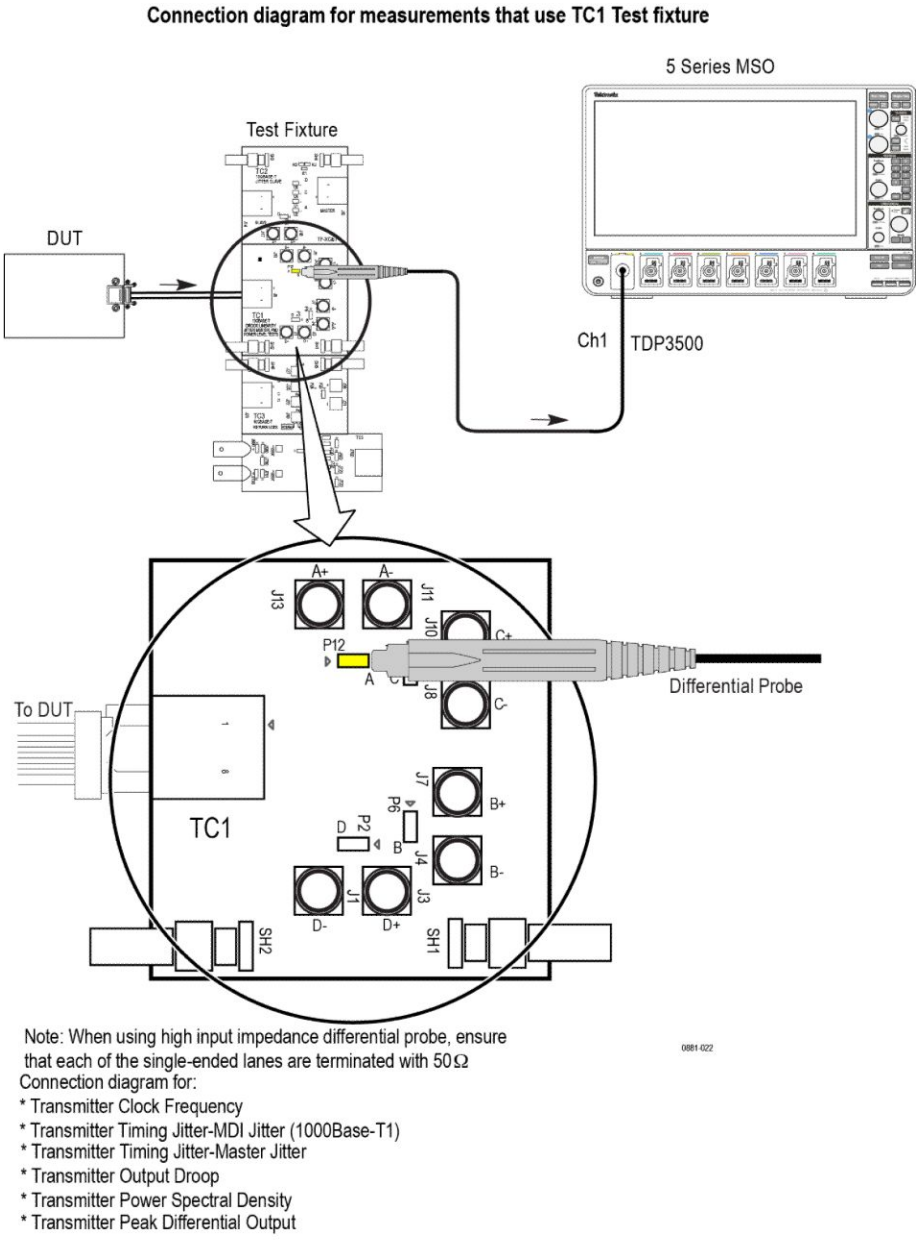



**Figure 53: Connection diagram for Power spectral density**

***NOTE.*** *Ensure that the "+" on the probe tip aligns with ▶ on the text fixture board. This takes care of the polarity being not reversed.*

## Power spectrum curve mask information

**Table 18: Power spectrum curve mask information**

| Frequency | PSD Upper Bound (dBm) | PSD Lower Bound (dBm) |
|---|---|---|
| at 1 MHz | –23.3 | –30.9 |
| at 20 MHz | –24.8 | –35.8 |
| at 40 MHz | –28.5 | –40.2 |
| 57 MHz - 200 MHz | –36.5 | – NA |

Upper and lower limits/masks are piece-wise linear masks connecting points given in the above table.

**Settings.** RBW = 10 KHz, VBW = 30 KHz, sweep time > 1 min, RMS detector, sweep time 3.275 seconds

**Mask**

**Figure 54: Mask for 1000BaseT1 PSD**

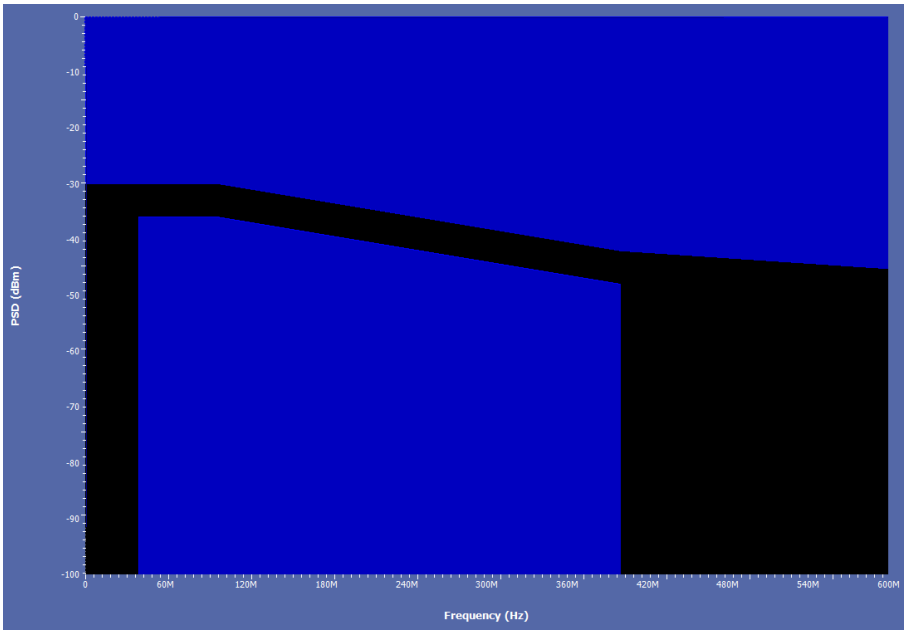



**Figure 55: Mask for 100BASET1 PSD**


**Settings**: RBW = 10 KHz, VBW = 30 KHz, sweep time > 1 min, RMS detector, sweep time 3.275 seconds

### Measurement algorithm

TekExpress Automotive Ethernet Compliance Solution automatically executes the calculations as described below for the power spectral density measurement.

1. The test mode 5 signal feed as input is filtered using a band limiting filter with cut off frequency of 600 MHz.
2. Spectral functions (SpectralMag) present in the oscilloscope MATH sub system are used for computing the spectrum of the filtered test mode 5 signal.
3. The output of "SpectralMag" is averaged over "Averages" times. "Averages" is an acquisition parameter present in the GUI (No entries in GUI for 1000BASE-T1). The default value of this parameter is 64.
4. "SpectralMag" is executed with the following configuration settings:
   - For 100BASE-T1: Center frequency: 100.5 MHz, For 1000BASE-T1: Center frequency: 300 MHz
   - For 100BASE-T1: Frequency span: 201 MHz, For 1000BASE-T1: Frequency span: 600 MHz
   - Window type: Gaussian
   - Vertical axis: scale —> Linear
   - Gating duration: 200 micro seconds
   - Gating duration = Record length / sampling rate
     - 2.5 Mega samples / 12.5 GS/sec = 200 micro seconds
   - Gating position: 0
   - R = 100 Ohm
5. Averaged SpectralMag is saved and used in postprocessing.
6. In postprocessing, the averaged spectral output is filtered using a moving average filter (smoothing).
7. Filtered result is X(k). From X(k), compute the power in dBm as given below:

   Power = (V_rms^2)/R

   Power in frequency domain = |X(k)|^2/R where k = 0 to N-1 are frequency bins

   Power in dB = 20*log10(|X(k)|) – 10*log10(R)

   Power in dBm = Power in dB + 10*log10(1000)

   Power in dBm = Power in dB + 10*log10(1000)

   20*log10(|X(k)|) – 10*log10(R) + 10*log10(1000)

   If R = 100 Ohm,

   Power in dBm = 20*log10(|X(k)|) – 20 + 30

   Power in dBm = 20*log10(|X(k)|) – (20 - 30)

   Power in dBm = 20*log10(|X(k)|) – Offset

   Where Offset = (20 – 30)
8. Computed power in dBm is compared with limits given in the specification, which gives limits only at certain frequencies. Linear interpolation is used to find values for intermediate frequencies.

Power(dBm) = Power (dBm/Hz) + 10 x log 10(BWHz) where, BWHz - Bandwidth resolution

**See also** *Plots*

## Transmitter Output Droop

This test confirms that the transmitter output level does not decay faster than the maximum specified rate.

**Test setup for Transmitter Output Droop**

**Specification**

**1000BASE-T1**

- Physical Layer Specifications (IEEE Std 802.3bp™-2016) a parameter for 1 Gb/s Operation over a Single Twisted-Pair Copper Cable section 97.5.3.1 Transmitter Output Droop

**1000BASE-T1**

- BroadR-Reach Physical Layer Transceiver Specification for Automotive Applications, version 3.2, section 5.4.1 or IEEE P802d3bwTM (100 BASE T1) section 96.5.4.1 transmitter specifications
- BroadR-Reach Physical Media Attachment Test Suite version 2.0, section 5.1

**Required test equipment**

In addition to the DUT and oscilloscope, you will need the following:

**1000BASE-T1/100BASE-T1**

- One supported differential probe
- Short RJ45 connector
- Test fixture: TF-XGbT

**Test setup procedure for Transmitter Output Droop for 1000BASE-T1/100BASE-T1**

The Transmitter Output Droop measurement setup involves the DUT, test fixture, and the oscilloscope.

Connect the equipment as shown in the connection diagram and as explained in the following procedure:

1. Select the Suite type on the DUT panel.
2. Set the DUT to generate and transmit test mode 1 for 100BASE-T1 and test mode 6 signal for 1000BASE-T1.

3. Connect the DUT Ethernet cable to J5 Ethernet port (TC1 segment) of TF-XGbT test fixture.
4. Connect the differential probe to P12 and the configured channel of the oscilloscope and also terminate the SMA pins J11 and J13 with 50 Ω terminators.

## Measurement setup and algorithm for 1000BASE-T1/100BASE-T1

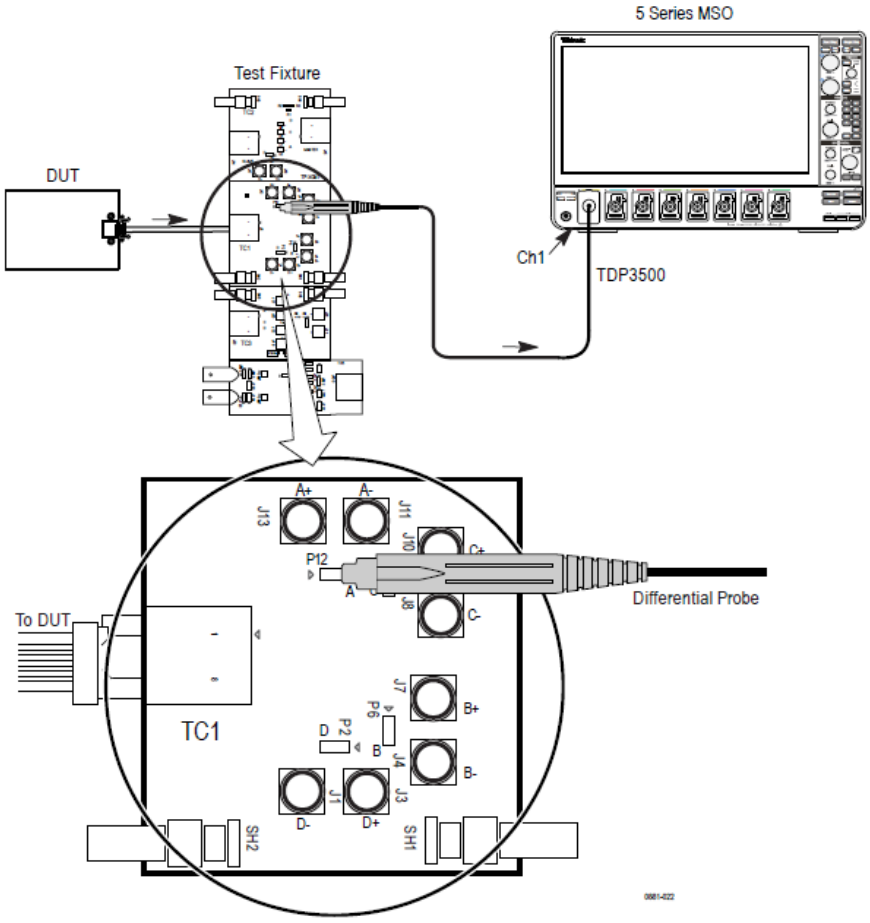
**Table 19: Measurement setup and algorithm**

| Item | Requirements |
|---|---|
| Configuration parameters | 1. Averages |
| Signal type | Test mode 1 for 100BASE-T1<br>Test mode 6 for 1000BASE-T1 |
| Measurement algorithm outputs | 1. Positive Droop in %<br>2. Negative Droop in % |
| Measurement algorithm inputs | Test mode 1 signal captured in differential form for 100BASE-T1<br>Test mode 6 signal captured in differential form for 100BASE-T1 |
| Limits | For 1000BASE-T1 Positive and Negative Output Droop Percentage<br>■ Lower limit: NA<br>■ Upper limit: 10%<br>For 100BASE-T1 Positive and Negative Output Droop Percentage<br>■ Lower limit: NA<br>■ Upper limit: 45% |

## Transmitter output droop

**Figure 56: Connection diagram for Transmitter output droop**

**Connection diagram for measurements that use TC1 segment in Test fixture**




Note: When using high input impedance differential probe, ensure that each of the single-ended lanes are terminated with 50 Ω.

Connection diagram for:

* Transmitter Clock Frequency
* Transmitter Timing Jitter-MDI Jitter (1000Base-T1)
* Transmitter Timing Jitter-Master Jitter
* Transmitter Output Droop
* Transmitter Power Spectral Density
* Transmitter Peak Differential Output

**Measurement algorithm**

TekExpress Automotive Ethernet Compliance Solution executes the calculations described as below for the transmitter output droop measurement.

**1000BASE-T1**

1. Check if the input waveform pattern is a square wave test pattern or not.
2. Check the input square wave test pattern, whether positive to negative width is , check whether positive and negative widths are greater than 12ns.
3. From zero crossing 4ns of each rising edge of each rising edge, compute the maximum voltage ($V_{p\_positive}$) and compute the voltage ($V_{droop\_positive}$) 16 ns after that maximum voltage index (12 ns period).

   $V_d = (V_{p\_positive} - V_{droop\_positive})$

   Positive Droop = $(V_d/V_{p\_positive}) * 100$
4. From zero crossing 4ns of each falling edge, compute the minimum voltage ($V_{p\_negative}$) and compute the voltage ($V_{droop\_nrgative}$) 16 ns after that maximum voltage index (12 ns period).

   $V_d = (V_{p\_negative} - V_{droop\_negative})$

   Negative Droop = $(V_d/V_{p\_negative}) * 100$

**See also** *Plots*

## Transmitter Peak Differential Output

**1000BASE-T1**

When measured with 100 Ohm termination, transmit differential signal at MDI shall be less than 1.30 V peak-to- peak.

**100BASE-T1**

When measured with 100 Ohm termination, this test confirms that the transmitter peak differential output is within 2.2 V peak-to-peak limits.

**Test setup for Transmitter Peak Differential Output for 1000BASE-T1/100BASE-T1**

### Specification

**1000BASE-T1**

- Physical Layer Specifications (IEEE Std 802.3bp™-2016) a parameter for 1 Gb/s Operation over a Single Twisted-Pair Copper Cable section 97.5.3.5 Transmitter Peak Differential Output

**100BASE-T1**

- IEEE Draft P802.3bw/ D3.3, section 96.5.6 transmitter specifications
- BroadR-Reach Physical Media Attachment Test Suite version 1.0, section 5.8

### Required test equipment

**1000BASE-T1/100BASE-T1**

In addition to the DUT and oscilloscope, you will need the following:

- One supported differential probe
- Short RJ45 connector
- Test fixture: TF-XGbT

### Test setup procedure for Transmitter Peak Differential Output test for 1000BASE-T1/100BASE-T1

The Transmitter Peak Differential Output test involves the DUT, test fixture, and the oscilloscope.

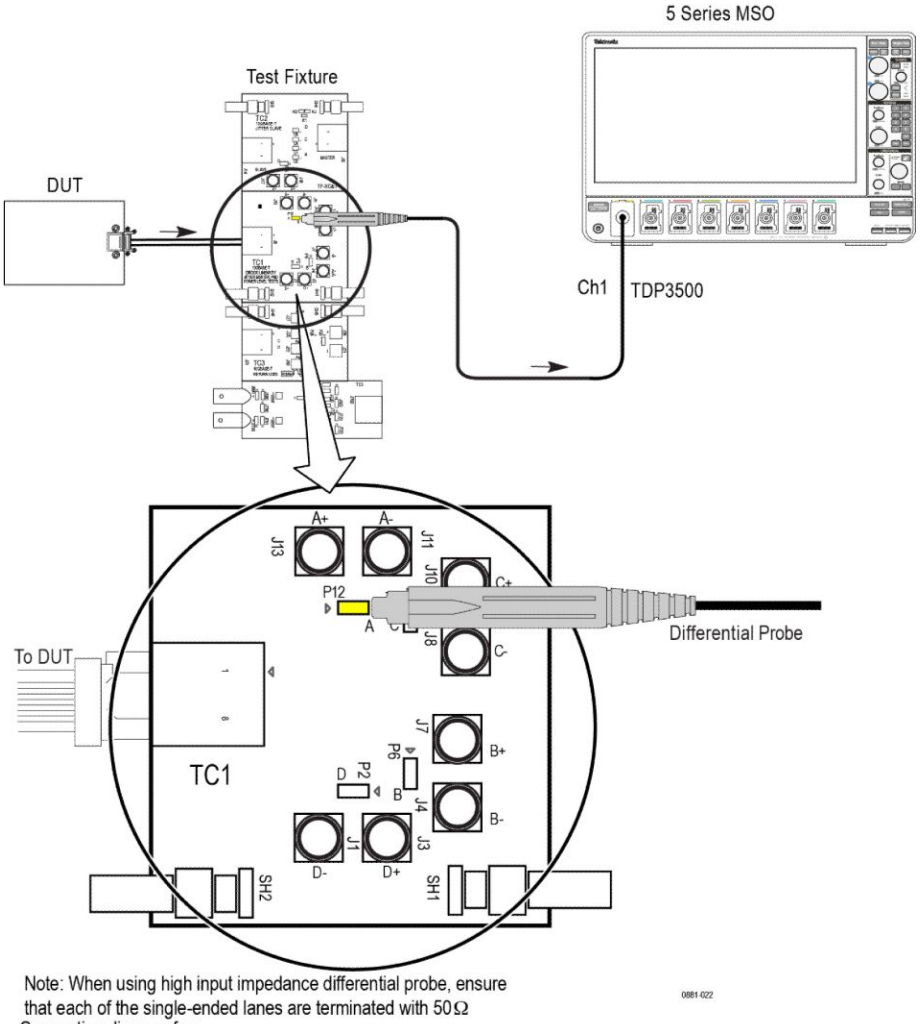
Connect the equipment as shown in the connection diagram and as explained in the following procedure:

1. Select the Suite type on the DUT panel.
2. Set the DUT to generate and transmit test mode 5 signal.
3. Connect the DUT Ethernet cable to J5 Ethernet port (TC1 segmant) of TF-XGbT test fixture.
4. Connect the differential probe to P12 and the configured channel of the oscilloscope and also terminate the SMA pins J11 and J13 with 50 terminators.
5. Ensure that the "+" point of differential probe should be near ►.

### Peak Differential Output

**Figure 57: Connection diagram for Peak Differential Output**

**Connection diagram for measurements that use TC1 Test fixture**




Note: When using high input impedance differential probe, ensure that each of the single-ended lanes are terminated with 50Ω

Connection diagram for:

* Transmitter Clock Frequency
* Transmitter Timing Jitter-MDI Jitter (1000Base-T1)
* Transmitter Timing Jitter-Master Jitter
* Transmitter Output Droop
* Transmitter Power Spectral Density
* Transmitter Peak Differential Output

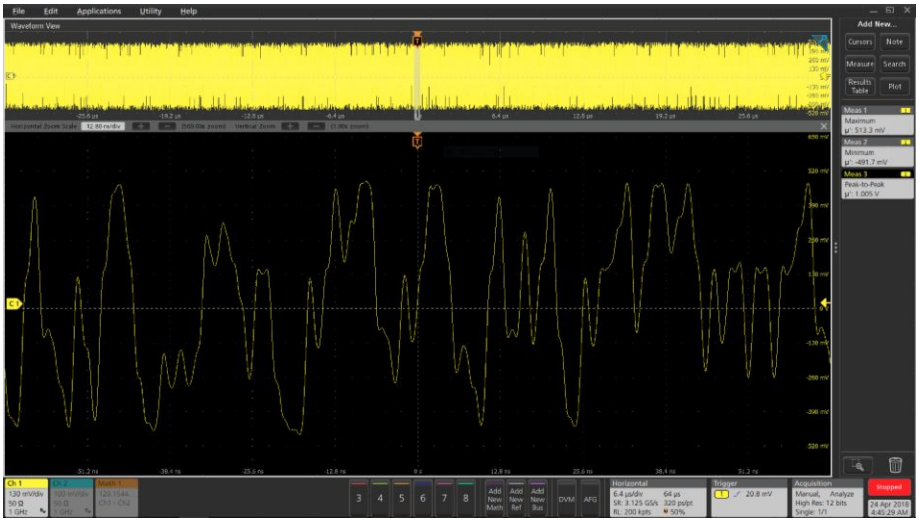
***NOTE.*** *Ensure that the "+" on the probe tip aligns with ► on the text fixture board. This takes care of the polarity being not reversed.*

## Measurement algorithm for 1000BASE-T1/100BASE-T1

TekExpress Automotive Ethernet Compliance Solution automatically executes the calculations as described below for the Transmitter Peak Differential Output measurement.

1. This measurement uses the test mode 5 signal for Peak Differential Output measurement.
2. Measurement acquires multiple test signals, which are defined on the Configuration tab. You can modify the values in the User Define Mode.
3. This test uses the Histogram peak-to-peak measurement to compute the peak-to-peak value. The histogram is being built over the defined number of acquisitions and compute the results.

**1000BASE-T1**



**100BASE-T1**

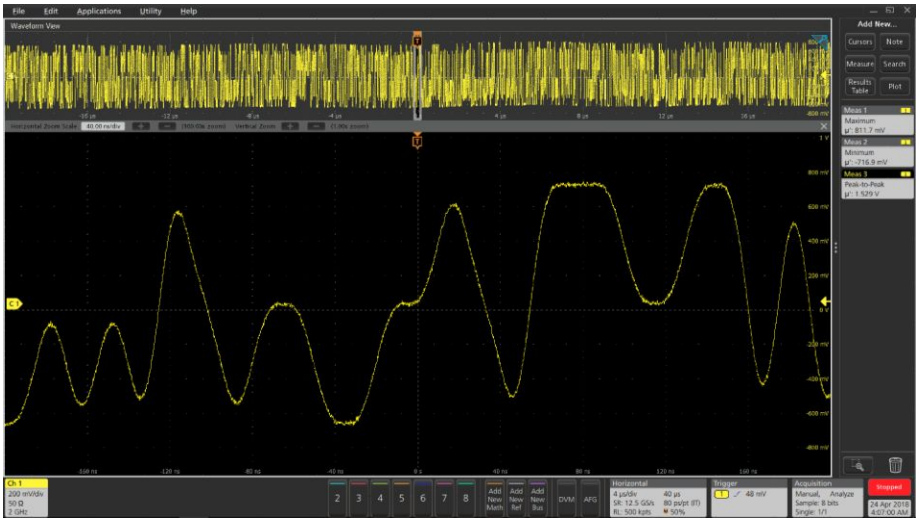


**Figure 58: Vertical Histogram for Peak Differential Output**

# Transmitter Distortion

This test confirms that the peak transmitter distortion is less than 15 mV for all 10 UIs within the eye opening.

## Test setup for Transmitter Distortion for 1000BASE-T1/100BASE-T1

### Specification

**1000BASE-T1**

- Physical Layer Specifications (IEEE Std 802.3bp™-2016) a parameters for 1 Gb/s Operation over a Single Twisted-Pair Copper Cable section 97.5.3.2 Transmitter Distortion test

**100BASE-T1**

- BroadR-Reach Physical Layer Transceiver Specification for Automotive Applications, version 3.2, section 5.4.2 or IEEE P802d3bwTM (100 BASE T1) section 96.5.4.2 transmitter specifications
- BroadR-Reach Physical Media Attachment Test Suite version 2.0, section 5.2

### Required test equipment 1000/100BASE-T1

In addition to the DUT and the oscilloscope, you will need the following:

***NOTE.*** *You will need to complete the calibration procedure before doing the Transmitter Distortion measurement with disturbing signal. The calibration procedure is used to effectively remove the disturbing signal and compensate for non-linearity in the disturber and test fixture.*

- One supported differential probe
- Four BNC cables (for connecting AFG or AWG5K to fixture) or two SMA cables with two BNC to SMA connectors (for connecting AWG7K to fixture)

  ***NOTE.*** *Use the Signal Generator for 1000BASE-T1 which supports the 125 MHz sine wave differential peak to peak 3.6 volts.*

- Short RJ45 connector
- GPIB cable (required if you use *AWG automation*, connects AWG and oscilloscope)
- Test fixtures: TF-XGbT
- TF-BRR-CFD (Clock Frequency Divider Unit): This is used to synchronize oscilloscope and signal source with the DUT Transmit_CLK.

The test setup procedure for Transmitter Distortion is divided into two parts as below:

1. **Calibration**
2. **Measurement Run**

### Calibration

*Calibration* is used to effectively remove the disturbing signal and compensate for nonlinearity in the disturber and test fixture. Calibration is used only when measurements are executed with a disturbing signal.

Design of the transmitter to tolerate the presence of the remotely driven signal with acceptable distortion or other changes in performance is a critical issue and must be addressed by the implementer. A disturbing signal is used to simulate the presence of a remote transmitter. The disturbing signal is defined as a sine wave generator that simulates the potential interfering effect of another transmitter.

Automotive Ethernet (1000/100BASE T1) measurements that require test mode 4 have to be done with a disturbing signal. Characteristics of disturbing signal are given in the following table.

The calibration has three parts/steps to calibrate the fixture path:

- A.Disturber Signal setup (1 of part 3 )
- B.DUT Signal Compensation Signal setup (2 of part 3)
- C.DUT Signal Compensation Signal setup ( Part 3 of 3)

### Test setup procedure for Transmitter Distortion 1000/100BASE-T1

1000BASE-T1


100BASE-T1


**Figure 59: Disturber compensation**

### A. Disturber Signal setup (1 of part 3) (AWG)

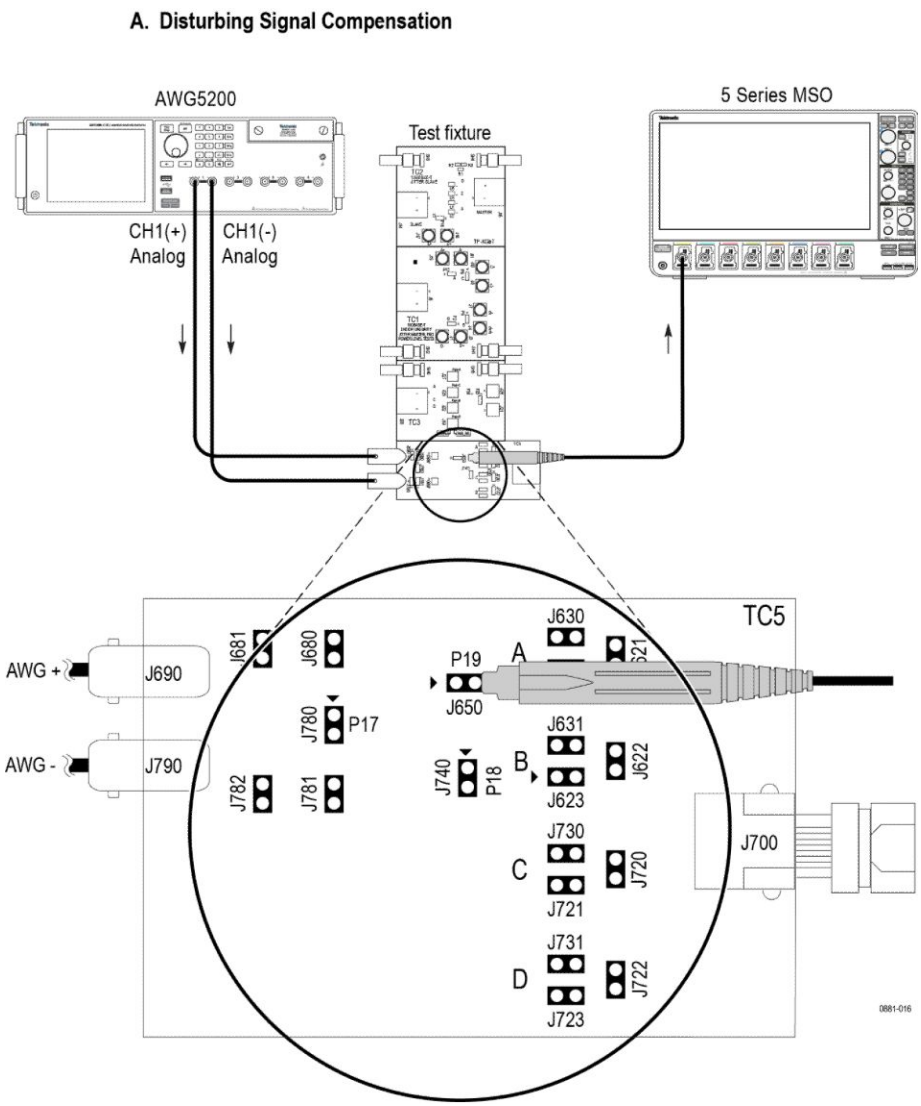



**Figure 60: Disturbing signal compensation (AWG)**

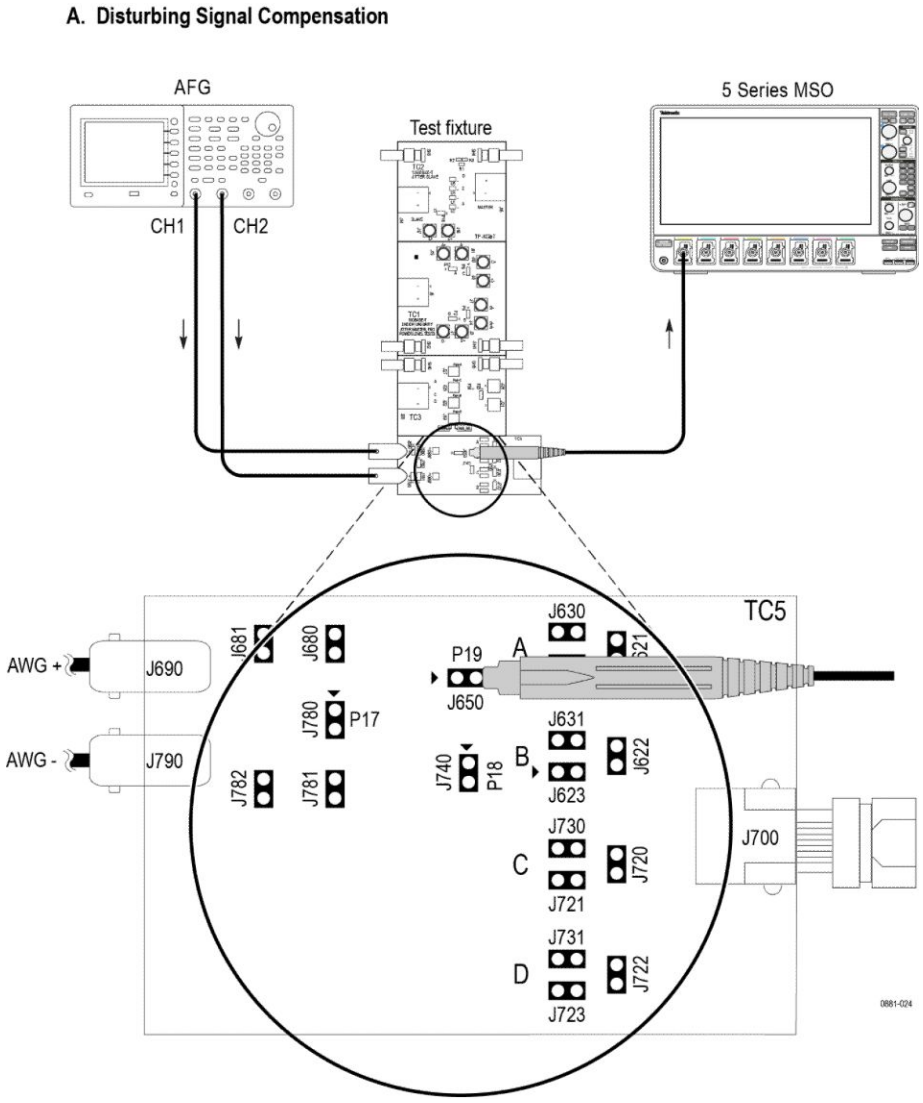



**Figure 61: Disturbing signal compensation (AFG)**

Make the connections as shown above:

1. Select the Signal Generator source instrument in Global settings.

   Follow the Instrument step mention on How to connect the Signal Generator to oscillocope.

   ***NOTE.*** *Use the Signal Generator which can generate the (3.6 )/2 volte peak to peak for 1000BASE-T1.*

2. Connect DUT as shown in the above connection diagram.
3. Take two equal length BNC cable.
4. Connect the 1st BNC Cable to (AFG/AWG) + (J690) and the other end to Channel 1 of AFG.
5. Connect the 1st BNC Cable to (AFG/AWG) + (J690) and other end to Channel 1 of AFG.
6. Connect the 2nd BNC Cable to (AWG/AFG) – (J790) and other end to Channel 2 of AFG. In case AWG, connect the other end of BNC to CH1_inverted.
7. Short the Pins J621, J630, J623, J721, J723, J680, and J781 using jumpers as shown in the connection diagram.
8. Disconnect the DUT from J700, if DUT is connected.
9. GO to the Measurement Configuration Tab and Press the RUN button of the step 1 of 3.

   The application will automatically configure the selected Signal Generator to a specified frequency and its amplitude level as defined in respective specification.

***NOTE.***

- *For 100 BASE-T1, set the attenuator factor to 10X on the probe if you are using P6247\6248.*
- *Without an amplifier, the AWG7000 series generates a maximum voltage of 2 Volt peak-to-peak signal, which cannot meet the 5.4 Volt requirement for the disturber signal for Transmitter Distortion test.*
- *Ensure that the "+" on the probe tip aligns with ► on the text fixture board. This takes care of the polarity being not reversed.*

### B. Disturbing signal compensation (2 of part 3) (AWG)

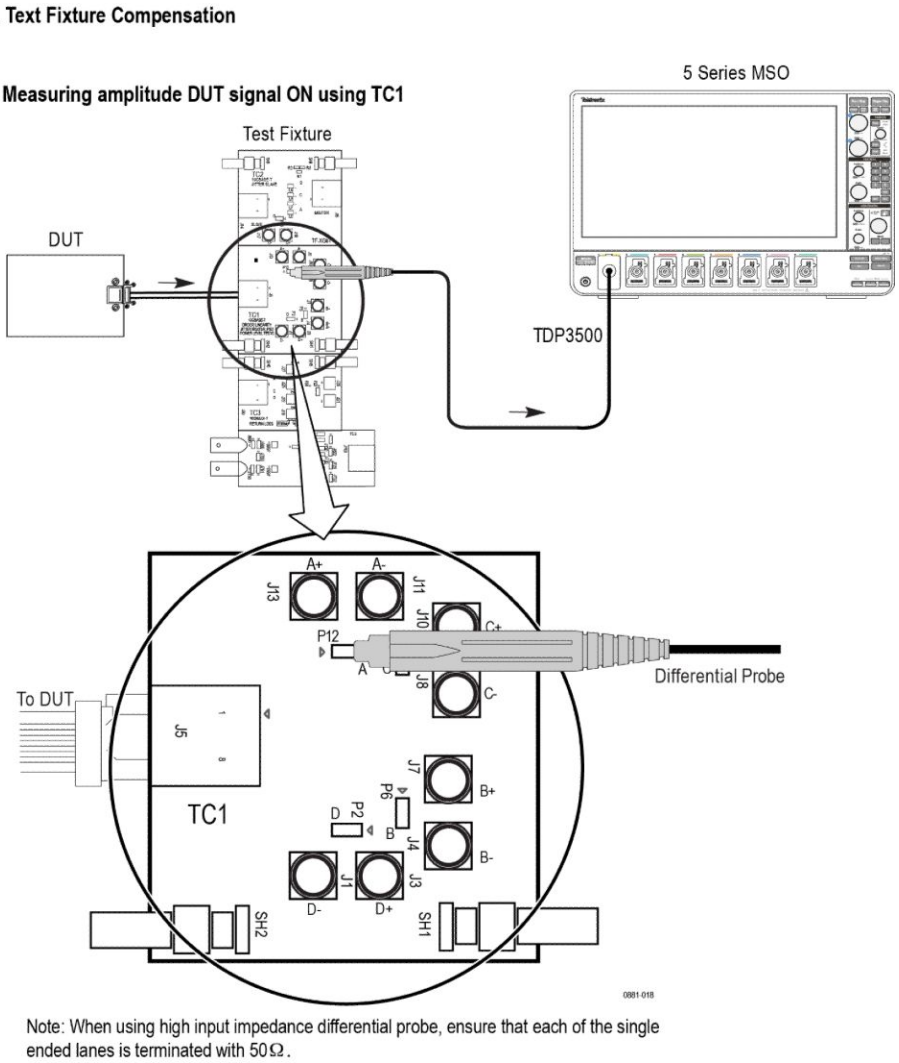



**Figure 62: Disturbing signal compensation (AWG)**

---

***NOTE.*** *When using high input impedance differential probe, ensure that each of the single ended lanes is terminated with 50* **Ω***.*

---

Make the connection as shown above:

1. Connect the DUT to RJ45 of TC1, J5 port.
2. Turn on DUT and set the DUT to run Test Mode 4 signal.
3. Connect the Differential probe tip to P12 pin.

   ---

   ***NOTE.*** *indicate the positive probe tip to be connected on test fixture. As per your DUT single twisted pair output, you can get the signal from one of the P2, P6, or P9 pin.*

   ---

4. Close J11 and J13 with the 50 Ω terminator cap, which is with respect to P12 pin.
5. From the Measurement Configuration tab > Select Calibration tab > Under step 2 of 3, click **RUN** button and wait to display the measured value in the table.

## C. Text fixture compensation (3 of 3)

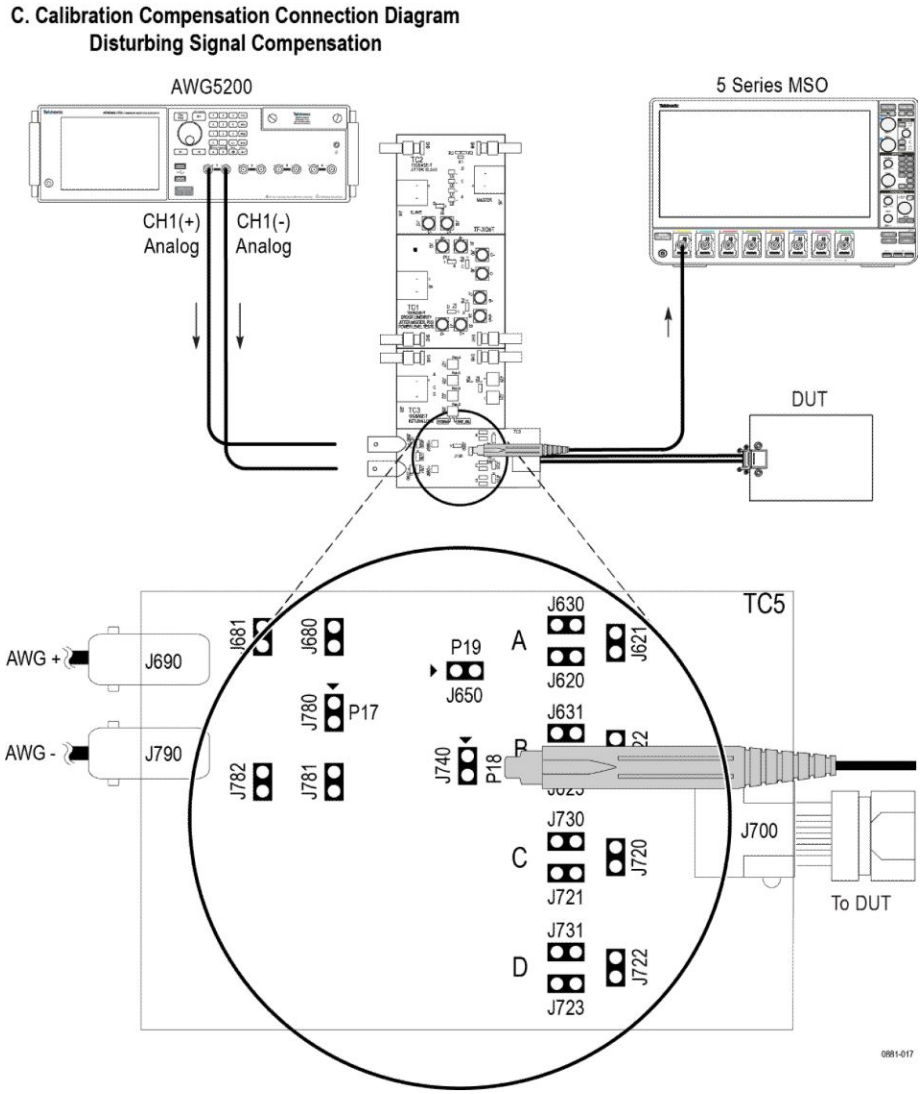



**Figure 63: Disturbing Signal Compensation (AWG)**

C. Calibration Compensation Connection Diagram

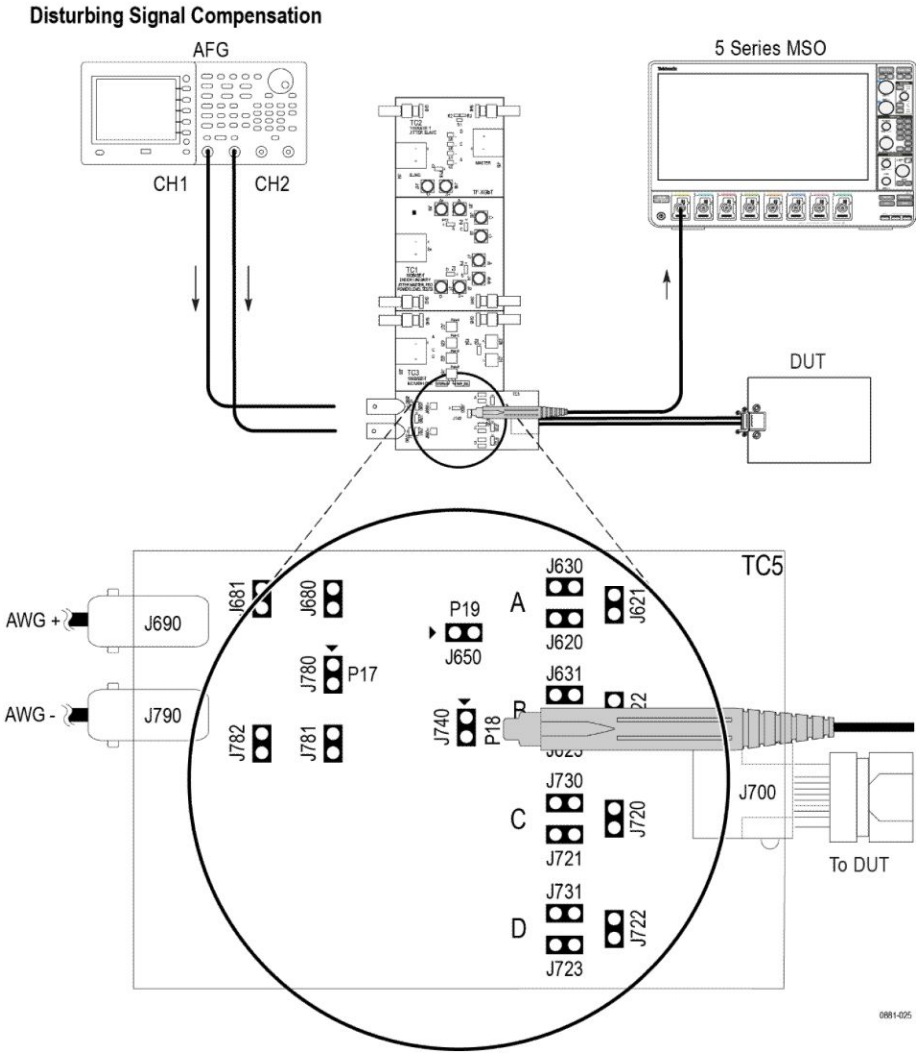



**Figure 64: Disturbing Signal Compensation (AFG)**

Make the connection as shown above:

1. Connect the DUT as shown in the connection above diagram.
2. Short the Pins J621, J630, J623, J721, J723, J680, and J781 using jumpers as shown in the connection diagram.
3. Disconnect the BNC cable J790 and J690, if it is connected.
4. From the Measurement Configuration tab > Select Calibration tab > Under step 3 of 3, click **RUN** button.

   The Automotive Ethernet Software Solution will automatically configure the selected Signal Generator to the specified frequency and its amplitude level as defined in the respective specification.

***NOTE.*** *Ensure "+" on the probe tip aligns with ► on the text fixture board. This takes care of the polarity being not reversed.*

### 100BASE-T1 AFG Configuration

The AFG setup happens automatically when you click the **Run** button at Step-1 of Disturber Compensation on Calibration tab and also sets the Signal Generator on Global Settings tab.

Following parameters are set for AFG:

1. Resets the AFG.
2. Sets amplitude to 5.4 Vpp.
3. Sets frequency to 11.111 MHz.
4. Sets the external Ref clock.
5. Turns on CH1 and CH2.

Point 1, 2, and 3 are applicable for both CH1 and CH2 and the phase set to CH1 is 0 degrees and for CH2, it is 180 degrees. For more details, refer to connection diagram (Test Fixture Compensation (2 of part 3)) and Measuring amplitude with disturbing signal OFF and DUT signal ON (3 of part 3)).

***NOTE.***

- *Ensure that before clicking the **Start** button the calibration steps are performed.*
- *During calibration, the signal source AFG/AWG will be setup automatically from the application by loading the disturber pattern.*
- *When you click the **Start** button, the AFG/AWG does not setup, only oscilloscope will setup because the application will not disturb the calibration done using AFG/AWG.*

The application automatically controls the AWG 5K and 7K similarly to that of AFG as described above.

### 100BASE-T1 Software signal correction or None method for 1000/100BASE-T1

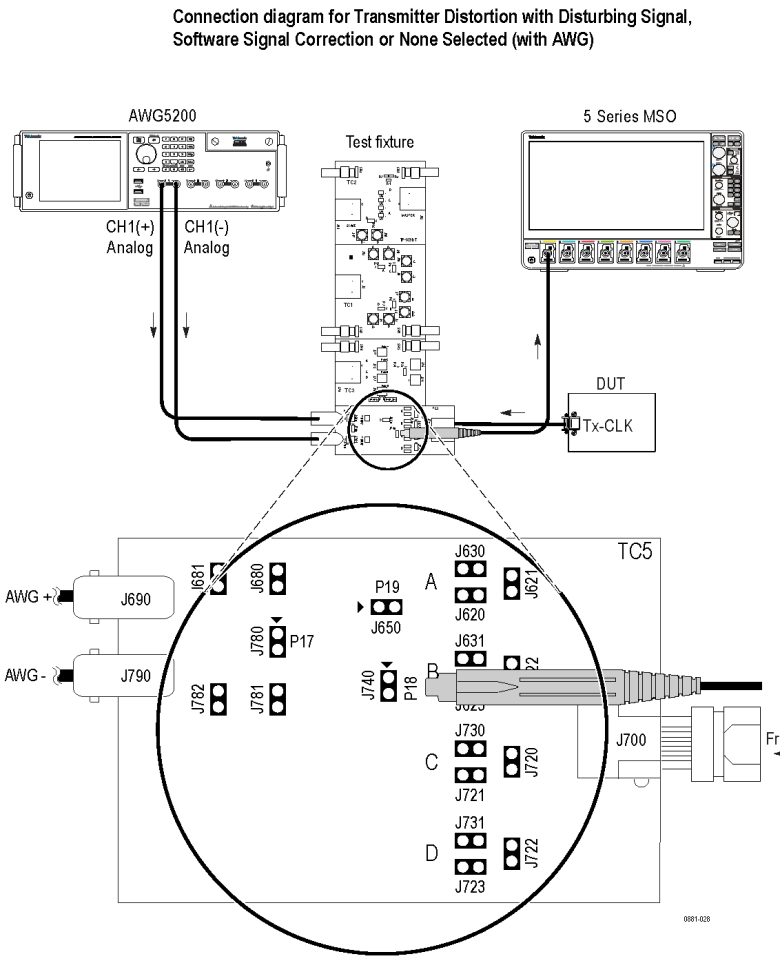



**Figure 65: Connection diagram for Software Signal Correction or None method-AWG**

Make the connection as above:

1. Click **Apply** after completing the calibration.
2. Click Measurement tab, and then select one of the options from the Configuration tab.
3. Ensure that the Jumper and the Probe setting as mention in Step 1 of 3 ( Refer A.Disturber Signal setup (1 of part 3) (AWG) section).
4. Select the Software Clock Divider or None method.

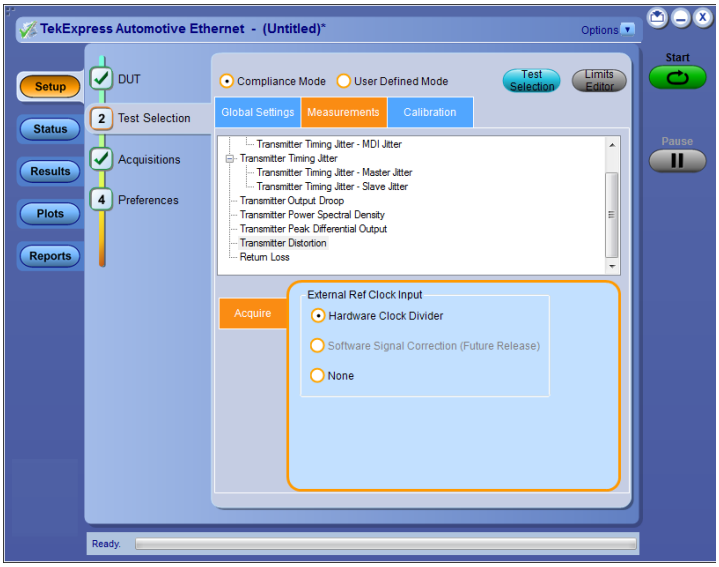
   The Software Signal Correction is selected by default and available for 100Base-T1.
5. Make the connection setup without the clock divider.

   For the Software Signal Correction option, the measurement corrects the acquired signal and adjusts the phase like a hardware clock divider unit, and then performs the peak distortion

### Hardware Clock divider method for 1000/100BASE-T1

1. Click **Apply** after completing the calibration.
2. Click **Measurement** tab, and then select one of the options from the Configuration tab.

**a.** Hardware Clock Divider option



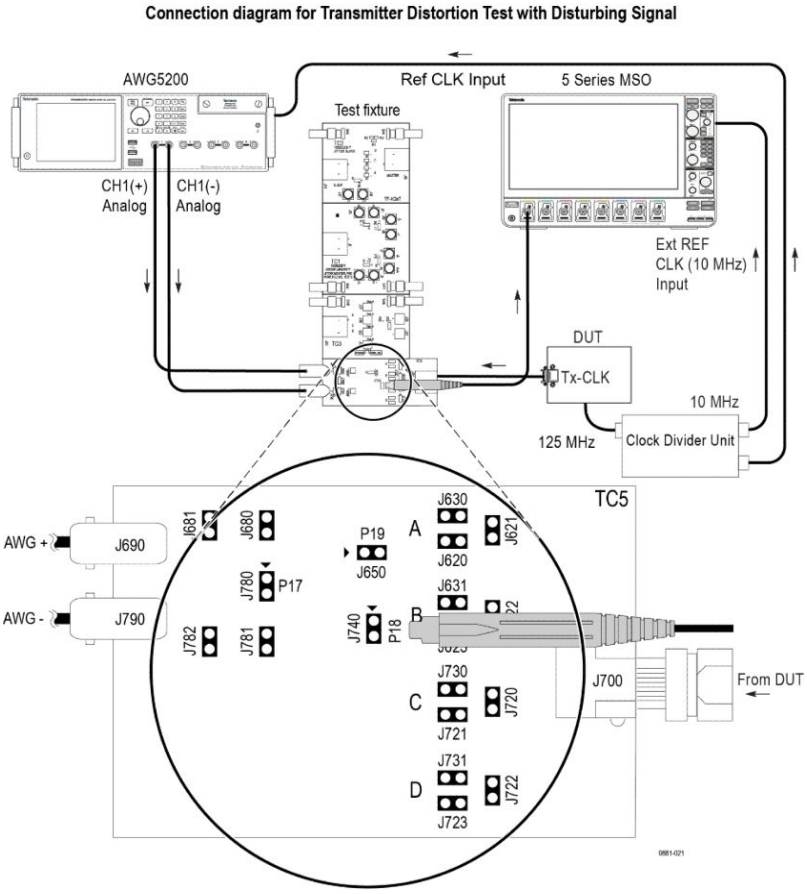




**Figure 66: Hardware clock divider**

- Select **Hardware Clock Divider**.
- Make the connection setup with the clock divider as show above. The clock divider synchronizes the oscilloscope and the disturber source.

***NOTE.***

- *In case the automaton with AFG/AWG is selected in the Global setting tab, the disturber pattern is automatically loaded during the disturber compensation process.*
- *In case of an unstable 10 MHz clock output from the Clock Divider Unit, restart the Clock Divider unit.*
- *Check the amplitude of the 10 MHz clock coming from Clock Divider Unit; it should not exceed the limit mentioned on the oscilloscope and AFG/AFG for 10 MHz reference input signal.*
- *If the External Reference Clock signal is stable (10 MHz), the oscilloscope will phase-lock with the external reference clock. This can be seen in the following image as "XRef", which indicates acquired signal is properly locked. The oscilloscope displays "NoRef", in case of unlocked phase. It indicates that the External Reference Clock is not proper and can result in incorrect results.*

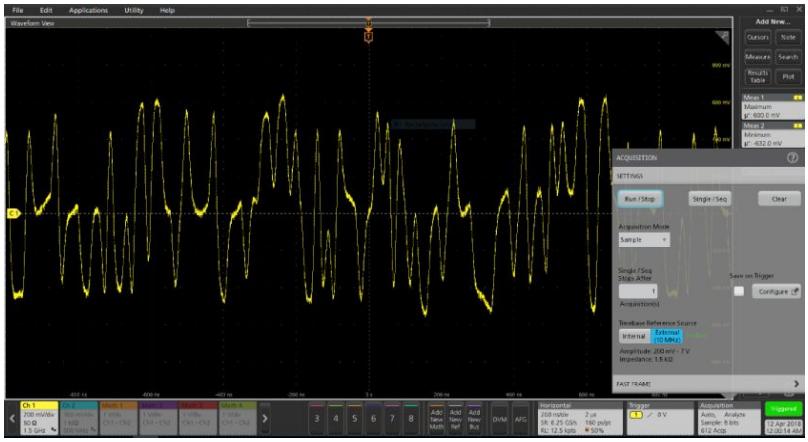
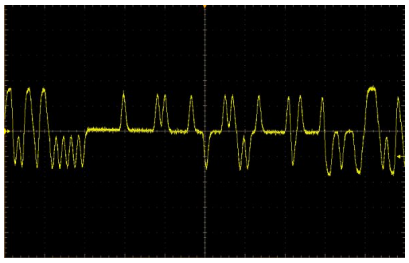


**Figure 67: XRef**

- Click **Start** to start the test and then make the connections as shown in the above schematic; click **OK** to continue the measurement.

## Measurement setup and algorithm

**Table 20: Measurement setup and algorithm for 1000/100BASE-T1**

| Item | Requirements |
|---|---|
| Signal type | Test mode 4 for 100BASE-T1  |
| Measurement algorithm outputs | All the computed 10 peaks must be less than 15 mV. |
| Measurement algorithm inputs | Test mode 4 signal captured in differential form<br>Disturbing Signal |
| Limits | Lower limit: NA<br>Upper limit: 15 mV |

TekExpress Automotive Ethernet Compliance Solution executes the calculations described below for the transmitter distortion measurement:

1. Compute the unit interval of the acquired test mode 4 signal.
2. Determine the number of frames and the start and end of the frames (each frame is 2047 bits long) in the acquired signal. Neglect the residual signal present at both the start and the end of the acquired signal.

   Ensure that all further processing happens from start of the $k^{th}$ frame to end of $n^{th}$ frame.
3. Compute the DC offset and remove the DC offset from the signal. Filter the signal using an HPF with a cutoff frequency of 1.068 MHz for 100BASE-T1 and 12 MHz for 1000BASE-T1 If LPF is enabled, then filter the signal using an LPF with a cutoff frequency of 33 1/3 MHz for 100BASE-T1 and 375 MHz for 1000BASE-T1.
4. Using the start and end frames information, do averaging across frames and finally get an averaged frame containing 2047 symbols or (2047*k) samples where $k$ is number of samples per symbol.
5. Do single acquisition and collect 2047 samples corresponding to a phase offset. These 2047 samples are picked from 2047 symbols and so each sample represents a symbol.

   The application removes the disturber signal as per the IEEE 802.3 MATLAB code.
6. Repeat the previous two steps for phase offsets 0 to 1 UI in steps of 0.1 UI (10 phase offsets). A total of 10 peak errors/peak distortions are obtained.

**See also** *Plots*

*Know limitations*

## Return Loss

This test confirms that the Return Loss of the DUT is within conformance limits.

### Test setup: Return Loss calibration and Return Loss measurement for 1000BASE-T1/100BASE-T1

### Specification

**1000BASE-T1**

Physical Layer Specifications (IEEE Std 802.3bp™-2016) a parameter for 1 Gb/s Operation over a Single Twisted-Pair Copper Cable section 97.7.2.1 MDI Return Loss not Transmitter Clock Frequency test.

**100BASE-T1**

- BroadR-Reach Physical Layer Transceiver Specification for Automotive Applications, version 3.2, section 96.8.2.1
- BroadR-Reach Physical Media Attachment Test Suite version 2.0, section 5.6

### Required test equipment for 1000/100BASE-T1

In addition to the DUT and oscilloscope, you will need the following:

- Two supported differential probes
- Three BNC cables (for connecting AFG or AWG to fixture) or three SMA cables with two BNC to SMA connectors (for connecting AWG7K to fixture) and TCA-BNC or TCA-SMA adapters (auxiliary)
- 1000 BASE-T1 MDI Return Loss need AWG5200 ( Bandwidth >= 600 MHz and with high AC voltage option)
- Four inch short RJ45 connector
- GPIB cable (required if you use *AWG automation*, connects AWG and oscilloscope)
- Test fixtures: TF-XGbT and fixture with loads (Open, Short, and Load)

### Test setup procedure for Return Loss test for 1000/100BASE-T1

The Return Loss measurement setup involves the DUT, test fixture, AWG/AFG, and the oscilloscope. Return Loss has two steps to run the test:

- Fixture Calibration using Open, Load and Short circuit
- Perform the test after calibration

### Fixture Calibration using Open, Load, and Short circuit

Connect the equipment as shown in the below connection diagram and as explained in the following procedure:

1. Connect one end of the 4" Ethernet small cable to the XGbT fixture TC3, J20 Ethernet port.
2. Connect the other end of the 4" Ethernet small cable to open J29 RJ45 port.
3. Take three equal length SMA cables. Connect the first SMA cable to SMA(+) on TC3 XGbT fixture and other end of the cable to CH1(+) to AWG5200.
4. Connect the second SMA cable to SMA(-) on TC3 XGbT fixture and other end of the cable to CH1(-) to AWG5200.
5. For 1000BASE-T1, connect the third SMA cable to CH1 marker on back side of the AWG5200 and for 100BASE-T1, connect the marker1 to the auxiliary channel of oscilloscope.

***NOTE.***

*If TF-XGbT is being used, ensure that Channel 1 is connected to P23 of TC3 and Channel 2 to P24 of TC3. Do not reverse Channel 1 to P2 and Channel 2 to P1; this will affect the machine.*

*If TF-GBE-BTP is being used, ensure that Channel 1 is connected to P1 of TC1 and Channel 2 to P2 of TC1. Do not reverse Channel 1 to P2 and Channel 2 to P1; this will affect the machine.*

6. Use Lane A, if the DUT is connected to this pair of Ethernet.
7. Follow the Jump connection J27 as shown in the following connection diagram. If your DUT is connected to other pair, follow the same jumper connection to the respective Lane.
8. Connect one differential probe to P23 on TC3 and configure the same in the Calibration tab

9. Connect the second differential probe to P24 on TC3 and configure same in the Calibration tab.

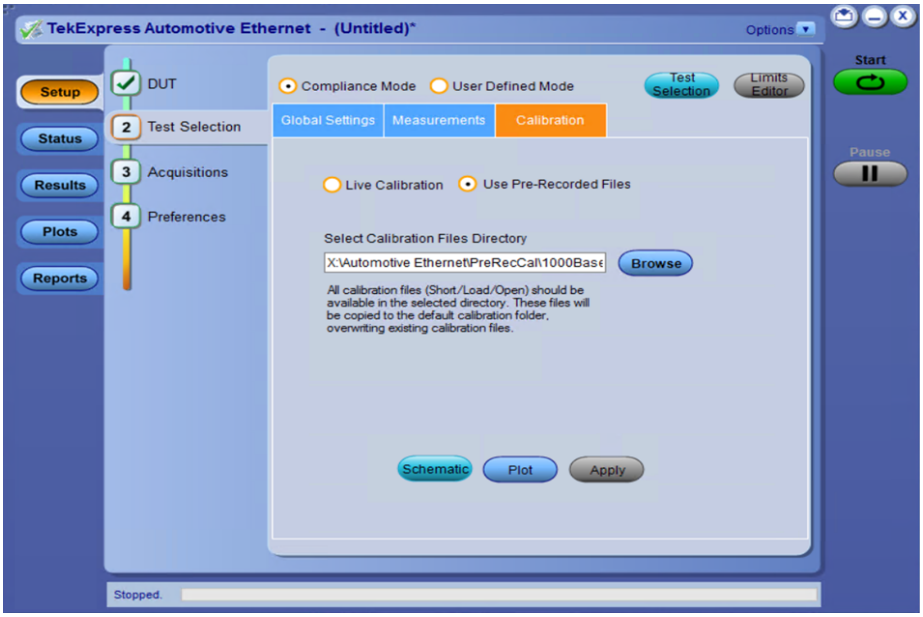

10. Connect the SMA/BNC cable to user selected AUX channel (if available) and, then select the same in SYNC INPUT.
11. Open the Instrument Control settings from the Option menu.
12. Select the options from the Search Criteria and click **Refresh** button.

    Wait for the signal source to display in list. If you are using LAN connect, follow the LAN configuration steps before doing these steps.
13. Click the **Global Settings** tab on the Test selection panel to view the listed instruments.
14. Click **Calibration** tab and then press the RUN button corresponding to Open / Load / Short calibration; then the selected AWG instrument automatically load with calibration pattern.

### Perform the test after calibration

1. After the completion of the Load, Open, and Short calibration in LIVE mode, click the **Apply** button in the Calibration tab. If the Calibration is Run with just near time, use the Use Pre-Recorded Files and select the File path and click **Apply**.
2. Connect the DUT setup as shown in below figure.
3. Connect the DUT with one end of 4-inch Ethernet cable and another end to RJ45 J20 of TC3. Connection for the AWG is as show in below images
4. Click the **Run** button.

   The measurement will Run and navigate to the Results panel.




### Test setup procedure for Return Loss test for 100BASE-T1/1000BASE-T1

The Return Loss measurement setup involves the DUT, test fixture, AWG/AFG, and the oscilloscope.

Connect the equipment as shown in the below connection diagram and as explained in the following procedure.

1. Make the connections as shown in the below connection diagram.
2. Perform the Load, Open, and Short *Calibration*.
3. Set the DUT to generate a test mode signal 4.
4. Connect a BNC cable/SMA cable to (AWG/AFG)+ terminal on TC3 and Channel 1 of AWG/AFG.
5. Connect a BNC cable/SMA cable to (AWG/AFG)– and $\overline{CH1}$ (CH1 inverted ) of the AWG/CH2 of AFG.

***NOTE.*** *Ensure Channel 1 is connected to P23 of TC3 of XGbT and Channel 2 to P24 of TC3. Do not reverse Channel 1 to P24 and Channel 2 to P23. This will affect the machine.*

6. Connect the marker1 to the user configured sync channel of the oscilloscope.
7. Connect one differential probe to P23 of TC3 segment in the fixture and another differential probe to P23 of TC3 segment in fixture.
8. Connect the GPIB/LAN/USB cable between AWG/AFG and the oscilloscope.
9. Select the Signal Generator option from Global Settings Tab.

---

***NOTE.*** *List of signal generators are shows in Global Settings Tab based on TekExpress Instrument Control Settings. Here you can select the connected signal source in Search Criteria and then click* ***Refresh*** *button. The connected signal sources will automatically display in Retrieved Instruments list and Global Settings's Tab under 'Signal Generator' list.*

---

***NOTE.*** *To setup the signal source manually, copy the related pattern from C:\Program Files\Tektronix\TekExpress\TekExpress Automotive-Ethernet\AWG Waveforms\Return Loss to USB memory stick. Recall this setup file on connected signal source.*

---

10. Click **Run** button.

    It will automatically transfer the Return loss pattern data to AWG/AFG, whichever is been selected, in the Signal generator of Global setting tab.

---

***NOTE.***

- *Ensure that the "+" on the probe tip aligns with ▸ on the text fixture board. This takes care of the polarity being not reversed.*
- *Set the attenuator factor to 1X on the probe.*
- *Step 8 and 9 are applicable and followed for Return Loss Load, Short and Open.*
- *Performing Calibration is recommended before executing the Return Loss measurement, and Ensure that you click the* ***Apply*** *button on the Calibration tab for execution of Return Loss measurement.*
- *Ensure differential high impedance probe heads and probe wires should not touch each other as it effects the Calibration results in Return Loss measurement.*
- *During calibration, the signal source AFG/AWG will be setup automatically from the application, by loading the disturber pattern.*
- *When you click on the Start button, the AFG/AWG does not setup, only oscilloscope will setup because the application will not disturb the calibration done using AFG/AWG.*

---

## Return Loss test (AWG/ AFG)

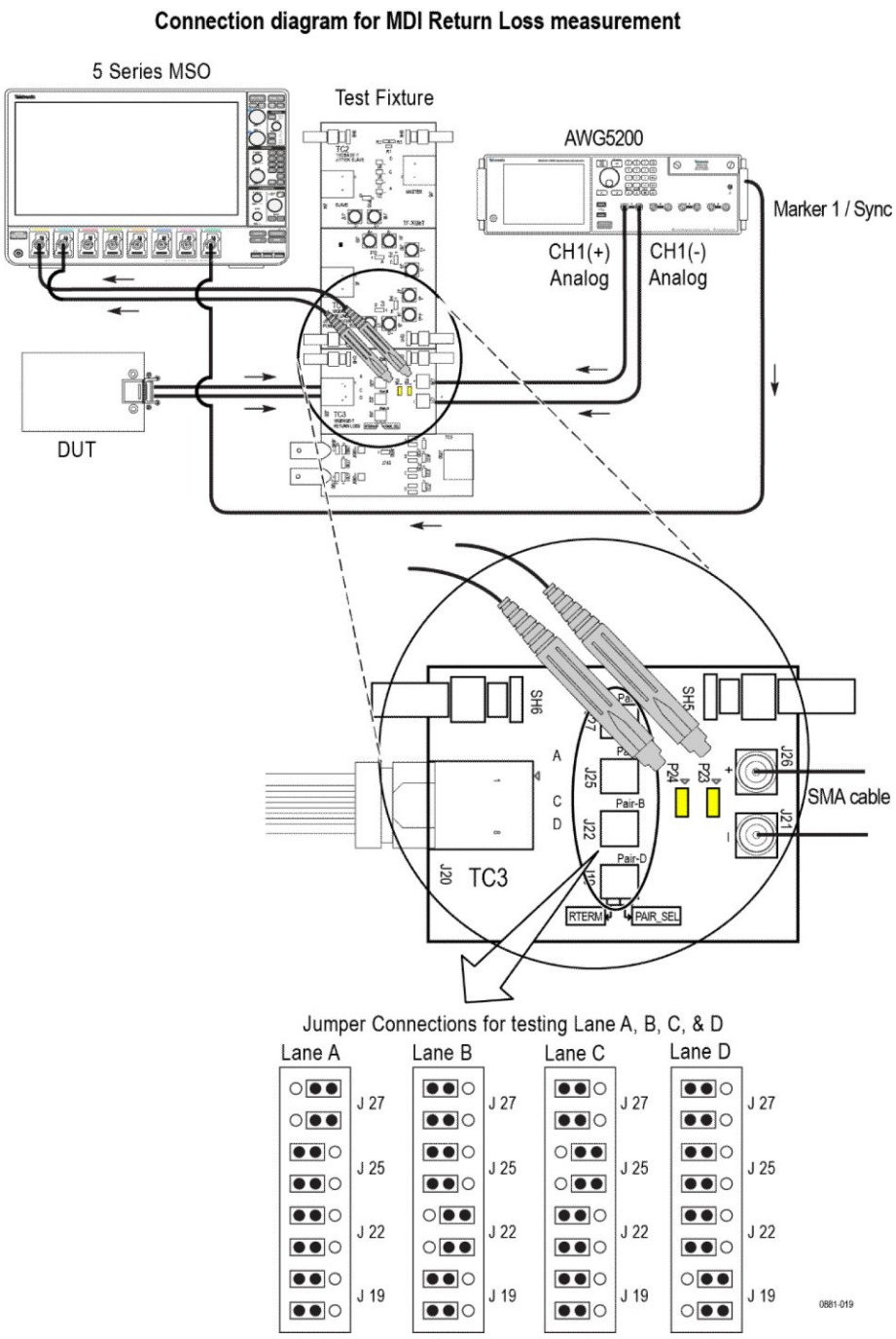
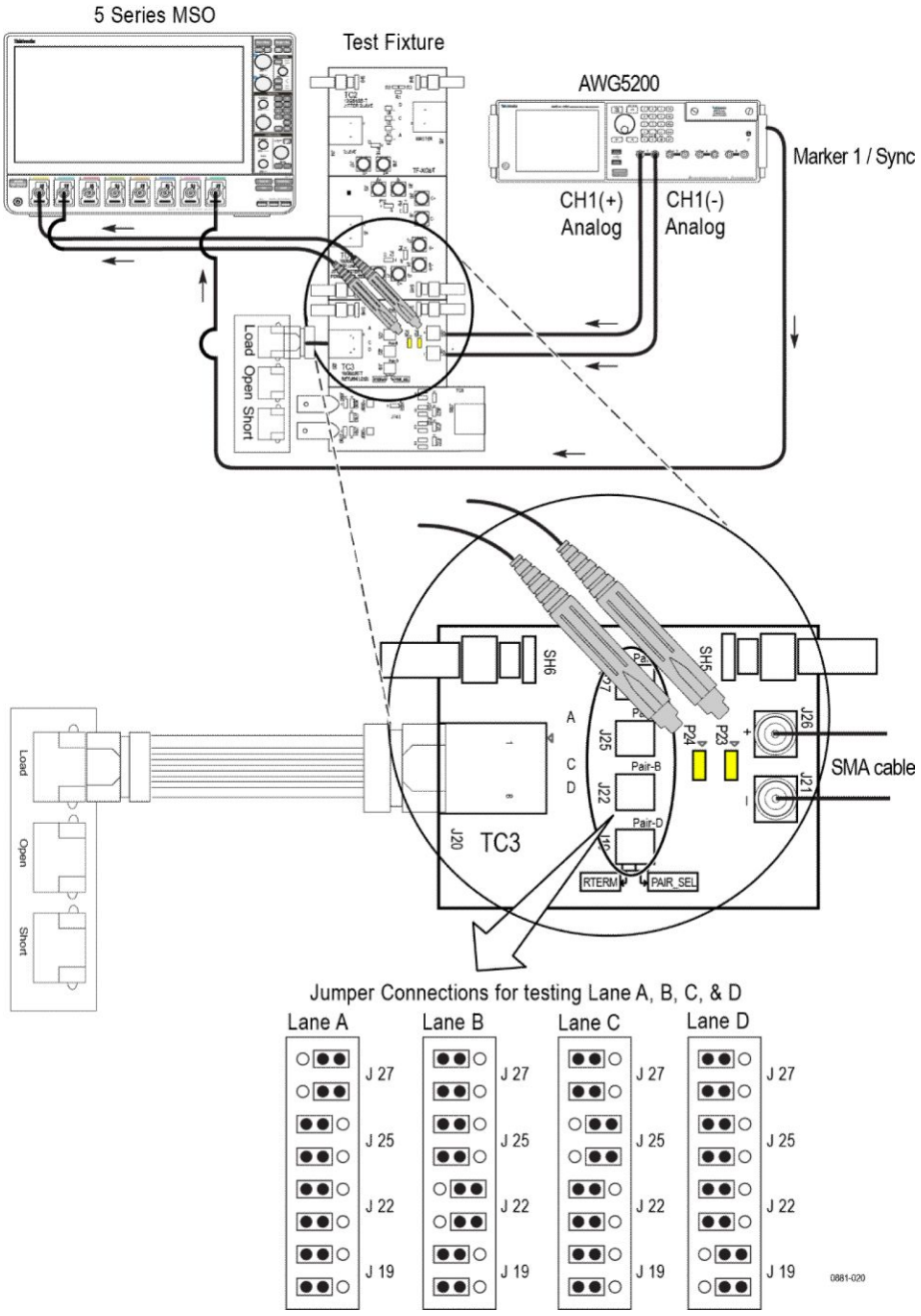



**Figure 68: Connection diagram for Return Loss test (AWG)**

## Return Loss calibration: Open, Load or Short

**Connection diagram for Return Loss measurement for OPEN, LOAD, or SHORT**

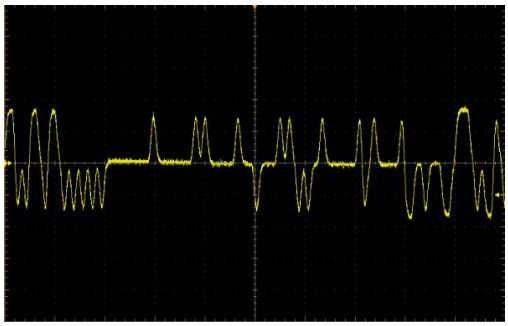


## Measurement setup and algorithm

The Return Loss measurement is divided into the following three steps:

1. Return Loss calibration
2. Computation of error coefficients
3. Computation of cable return loss

**Table 21: Measurement setup and algorithm**

| Item | | Requirements |
|---|---|---|
| Configuration parameters | | 1. Smoothing factor |
| Signal type | | Test mode 4 for 100BASE-T1  |
| Measurement algorithm outputs | | Return loss in dB for frequency ranging from 3 MHz to 100 MHz for 100BASE-T1 and 2 MHz to 600 MHz for 1000BASE-T1. |
| | Calibration | Gamma_Open, Gamma_Short and Gamma_Load corresponding to open, short and load terminations |
| | Computation of coefficients | Error coefficients a, b, and c |
| | Computation of cable loss return | 1. Return_loss_100 Ohm loads |
| Measurement algorithm inputs | | |
| | Calibration | Differential waveforms captured at probe point P23 and P24<br>Load Type |
| | Computation of coefficients | Gamma_Open, Gamma_Short and Gamma_Load computed during calibration |
| | Computation of cable loss return | Differential waveforms captured at probe point P23 and P24<br>Load Type |

| Item | Requirements |
|---|---|
| Limits | For 100BASE-T1:<br>Lower limit: NA<br>Upper limit: Return Loss (f) = 20 (in dB) for f = 1 to 30 MHz<br>= 20 – 20 * $\log_{10}$(f/30) (in dB) for f = 30 to 66 MHz<br>For 1000BASE-T1:<br>Lower limit: NA<br>Upper limit: Return Loss (f) = 18-18log10(20/f) for f = 2 to 20 MHz<br>= 18 for f = 20 to 100 MHz<br>18 - 16.7$\log_{10}$(f/100) for f = 100 MHz to 600 MHz |
| Plots | A plot containing return loss versus frequency and mask obtained with above limits |

## Measurement algorithm

### Calibration

The calibration procedure is required to correct for probe and fixture loading in the final measured result. Calibration is done using three loads: open, short, and 100 Ω load.

TekExpress Automotive Ethernet Compliance Solution executes the calculations described below for the return loss calibration:

1. Multiply the acquired signals using Gaussian window (convolving in frequency domain) and then compute the reflected and incident voltages.

   Reflected voltage = (2*P1 – P2)

   Incident voltage = P1

2. Return loss is computed as follows:

   Return loss = FFT(Reflected voltage)/ FFT(Incident voltage)

   This return loss is computed for three types of loads: Open, Short, 100 Ω load. Let the return loss computed be Gamma_Open, Gamma_Short, and Gamma_Load for open, short, and 100 Ω load, respectively. The Gamma_Open, Gamma_Short,and Gamma_Load are dumped in csv files.

3. For plotting purpose, Gamma_Open, Gamma_Short, and Gamma_Load are converted to dB scale, interpolated, and smoothed. Interpolated and smoothed Gamma_Open, Gamma_Short, and Gamma_Load are plotted for visual representation.

### Computing error coefficients

Automotive Ethernet Automated Compliance Solution executes the calculations described below for the return loss error coefficient calculation.

1. Gamma_Open, Gamma_Short, and Gamma_Load are used for computing the following three coefficient that are required for computing cable return loss:

   **a.** b – Gamma_Short + c * Gamma_Short

   **b.** Gamma_Load

   **c.** (Gamma_Short + Gamma_Open – 2 * b) / (Gamma_Short – Gamma_Open)

### Computing Return Loss

TekExpress Automotive Ethernet Automated Compliance Solution executes the calculations described below for the return loss:

1. Multiply the acquired signals using Gaussian window (convolving in frequency domain) and then compute the reflected and incident voltages.

   Reflected voltage = (2*P1 – P2)

   Incident voltage = P1

2. Return loss is computed as follows:

   Return loss = FFT(Reflected voltage)/ FFT(Incident voltage)

   Let the computed return loss be GammaValue.

3. Read the error coefficients a, b, and c from the dumped csv files. Compute the corrected return loss using a, b, and c.

   Corrected Return loss = (GammaValue – b)/(a – c * GammaValue)

   This return loss is for 100 Ω load. Let this be return_loss_100.

4. For plotting purposes, Return_Loss_100 is converted to dB scale, interpolated, and smoothed. Interpolated and smoothed Return_Loss_100 is plotted for visual representation.

   Return_loss_100 is compared with the specification given limit.

### Hints when Return Loss fails

Ensure you:

- Performed Scope SPC calibration, ensure that it passed.
- Used 6" Ethernet interconnect cable, a different one than earlier tests.
- Ran with default configuration (like 200 avg count).

### See also

*Plots*

*Know limitations*

## Known Limitations

The following are the know limitations:

1. Return Loss and Transmitter Distortion measurements needs Calibration before RUN. Calibration is recommended when there is a change in the setup.
2. In Return Loss measurement, when you select 'Use pre-recorded waveform' option for Calibration for the first time, then an error message is displayed as 'Calibration files not present'. It is recommended to RUN the live Calibration before you select the pre-recorded option.
3. When you recall the Return Loss measurement session file, the Calibration files are not recalled. You need to set/load the Calibration files manually.
4. You cannot stop or pause Calibration RUN during its execution. (This is applicable for both Return loss and Transmitter Distortion measurement)
5. The Plots are different when you RUN the Return Loss in pre-recorded mode for the first time, when compared to subsequent RUNS.

   Wait ( max of 3-4 minutes) till the **RUN** button is enable to use. Do not press **RUN** button till application is ready.
6. When the application is not able to work with LAN connection for AFG, use USB-GPIB cable to connect AFG to scope. Ensure that instrument refresh is performed in the TekExpress application, before configuring signal source in the Global settings tab.

# Measurement error messages

## Measurement error messages

The following table lists all of the error messages associated with TekExpress Automotive Ethernet measurements and their definitions.

**Table 22: Measurement error messages**

| Error message | Description |
|---|---|
| **Transmitter Output Droop measurement (Test mode 1)** | |
| Signal Validation failed for test mode 1. Make sure that input signal has:<br>■ Edge to Edge period deviation is less than 10%<br>■ At least 2 Unit Intervals | This error occurs if the input signal does not meet the requirements given in specification for test mode 1 signal. Edge to Edge period 33 unit intervals (500 ns) |
| Signal validation failed for test mode 1.Make sure that input signal has:<br>■ Rise to Rise period deviation is less than 10%. | This error occurs if the input signal has Rise to Rise period deviation of more than 10%. Rise to Rise period value (500 ns) |
| Signal validation failed for test mode 1.Make sure that input signal has:<br>■ Fall to Fall period deviation is less than 10%. | This error occurs if the input signal has Fall to Fall period deviation of more than 10%. Fall to Fall edge period value (500 ns) |
| Signal validation failed for test mode 1.Make sure that input signal has:<br>■ Edge to Edge period is greater than 500 nano seconds | This error occurs if the input signal does not meet the Edge to Edge period greater than 500 nano seconds. |
| **Transmitter Clock Frequency measurement and Transmitter Timing Master Jitter measurements (Test mode 2)** | |
| Signal Validation failed for test mode 2. Make sure that:<br>■ For 100BASE-T1: The unit interval/frequency of the signal does not deviate beyond ± 100 ppm from 66(2/3) MHz.<br>■ For 1000BASE-T1: The unit interval/frequency of the signal does not deviate beyond ± 100 ppm from 125 MHz. | For 100BASE-T1: This error occurs if the input frequency of the acquired test mode 2 signal is varies beyond 66(2/3) MHz ± 100 ppm.<br>For 1000BASE-T1: This error occurs if the input frequency of the acquired test mode 2 signal is varies beyond 125 MHz ± 100 ppm. |
| Signal Validation failed for test mode 2. Ensure that input signal has:<br>■ Rise to Rise period deviation is less than 10%. | This error occurs if the input signal has Rise to Rise period deviation of more than 10%. |

| Error message | Description |
|---|---|
| Signal Validation failed for test mode 2. Ensure that input signal has:<br>■ Fall to Fall period deviation is less than 10%. | This error occurs if the input signal has Fall to Fall period deviation of more than 10%. |
| Signal validation failed for test mode 2.Make sure that input signal has:<br>■ For 100BASE-T1: Edge to Edge period deviation is less than 10% compared to 66(2/3) MHz Tx_Clk.<br>For 1000BASE-T1: Edge to Edge period deviation is less than 10% compared to 125 MHz Tx_Clk.<br>■ At least 2 Unit Intervals,<br>■ Input is not clock signal (test mode 2). | This error occurs if the input signal has:<br>■ For 100BASE-T1: Edge to Edge period deviation is more than 10% compared to 66(2/3) MHz Tx_Clk.<br>For 1000BASE-T1: Edge to Edge period deviation is more than 10% compared to 125 MHz Tx_Clk.<br>■ More than two cycles are need to execute the measurement.<br>■ Input signal is not a clock signal (test mode 2). |
| **Transmitter Timing Slave Jitter Measurement (Test mode 3)** | |
| Signal validation failed for TX_TCLK signal. Make sure that:<br>■ The Edge to Edge deviation is less than 10%. | This error occurs if the input signal does not meet the requirements given in specification for TX_TCLK signal. |
| Signal validation failed for TX_TCLK signal. Make sure that input signal has:<br>■ Rise to Rise period deviation is less than 10%. | This error occurs if the input signal does not meet the requirements given in specification for TX_TCLK signal. |
| Signal validation failed for TX_TCLK signal. Make sure that input signal has:<br>■ Fall to Fall period deviation is less than 10%. | This error occurs if the input signal does not meet the requirements given in specification for TX_TCLK signal. |
| **Transmitter Distortion and Return Loss Measurement (Test mode 4)** | |
| Signal Validation failed for test mode 4. Ensure that input signal:<br>■ Is PAM3 modulated<br>■ Has at least 3 frames of test mode 4 (2047 bits).<br>■ The common possible cause could be polarity of the probing points being reversed. Ensure Probe tip '+' aligns with '<' on the test fixture board. | This error will occur if the input signal does not meet the requirements given in specification for test mode 4 signal. |

| Error message | Description |
|---|---|
| Signal Validation failed for test mode 4. Ensure that:<br>■ For 100BASE-T1: Interval/frequency of the Clock signal does not deviate beyond ± 100 ppm from 66(2/3) MHz.<br>For 1000BASE-T1: Interval/frequency of the Clock signal does not deviate beyond ± 100 ppm from 125 MHz. | This error occurs if the input signal does not meet the requirements given in specification for test mode 4 signal. Follow the instruction and step 3. For more information, see *Table 20: Measurement setup and algorithm*. |
| **Return Loss and Power Spectral Density Measurement (Test mode 5)** | |
| Calibration files are not present for return loss measurement. Either Deselect the Measurement or Stop the Execution, Perform Calibration and Re-Run the test. | This error occurs if return loss measurement is executed without calibration files. Always run the calibration and then execute return loss measurement. If calibration is done, return loss measurement uses latest available calibration files. |
| **Error message applicable to all tests** | |
| Not Enough Edges in the Waveform. Acquire the waveform for a longer duration. | This error occurs If the input signal does not have enough rise to fall and fall to rise transitions. |
| Data points acquired is insufficient for moving average filtering. Set the start frequency, stop frequency and RBW such that (stop frequency - start frequency)/RBW is greater than 3. | User should check the start frequency, stop frequency and RBW setting. This error will occur If the waveform does not have enough frequencies between the start and stop frequency. |
| Captured Signal has less than 3 segments, Please increase Record Length. | This error occurs If the input signal has less than three segments of test mode 4 signal (2047 bits). |
| Unable to run because return loss measurement is not calibrated. Calibrate and apply the return loss measurement, and then click on start. | This error occurs if user tries to execute return loss measurement without applying calibration (computation of error coefficients). |
| Invalid Signal at CH1. Please check DUT connections and re-run the test. | This error occurs if an invalid signal is fed as input to return loss measurement (wrong connection). |
| | |
| Signal validation failed for Test Mode.Input waveform is not found. | This error occurs when waveform is not found for signal validation. |

# TekExpress programmatic interface

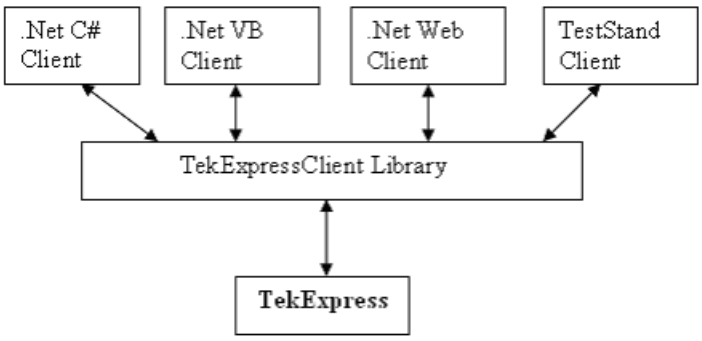
## About the Programmatic Interface

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 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 automation. 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 are the steps that a client needs to follow to use TekExpressClient.dll to programmatically control the server:

A client UI must be developed to access the interfaces exposed through the server. This client needs to load 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 must do the following to run a remote operation:

1. The client needs to provide the IP address of the PC at which the server is running in order to connect to the server.
2. The client needs to lock 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. Note that this does not lock the UI.

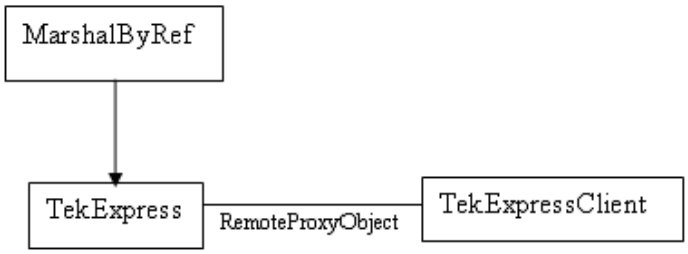
   If any other client tries to access a server that is locked, it will get a notification that the server is locked by another client.
3. When the client has connected to and locked the server, the client can access any of the programmatic controls to run the remote automations.
4. After the client operations are completed, the server needs to be unlocked by the client.

**See also**

*About TekExpress Automotive Ethernet 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.

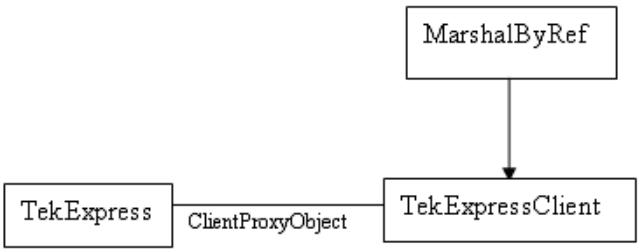
For example,

//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:

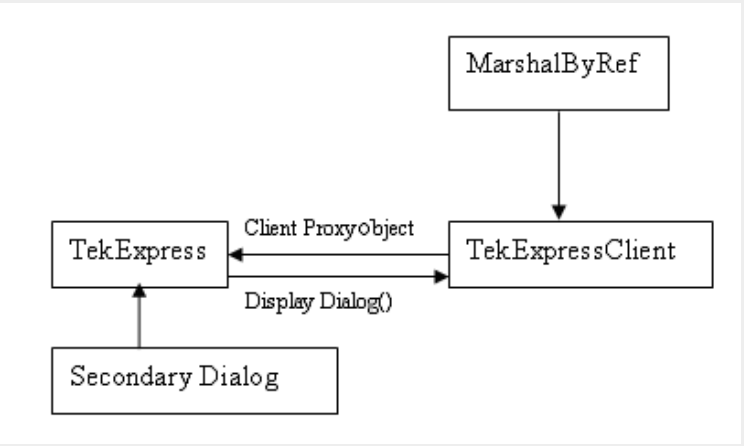
- To get the secondary dialog messages from the server.
- To get the file transfer commands from the server while transferring the report.

Examples

```
clientObject.clientIntf.DisplayDialog(caption, msg,iconType, btnType);
clientObject.clientIntf.TransferBytes(buffer, read, fileLength);
```

For more information, click the topic links listed below.

**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.

## Program examples

The following program examples show how to communicate between a PC and TekExpress Automotive Ethernet remotely, using typical steps.

For detailed information about each command, see the *TekExpress Automotive Ethernet Application Commands* section.

**Table 23: Remote access code example 1**

| Task | Code |
|---|---|
| Start the application | |
| Connect through an IP address. | {'Set String Details<br>string devicename = ”Automotive Ethernet”<br>string suitename = ”100BASE-T1/1000BASE-T1”<br>m_Client.Connect("localhost")'True or False<br>clientID = m_Client.getClientID } |
| Lock the server | m_Client.LockServer(clientID) |
| Disable the Popups | m_Client.SetVerboseMode(clientID, false) |
| Set the DUT ID | m_Client.SetDutId(clientID, "DUT_Name") |

| Task | Code |
|---|---|
| Select channels | m_Client.SetGeneralParameter(clientID, devicename, suitename, Testname, "Source Data$CH1");<br>m_Client.SetGeneralParameter(clientID, devicename, suitename, Testname, "Source Data$CH2");<br>m_Client.SetGeneralParameter(clientID, devicename, suitename, Testname, "Probe2$C2");"This Probe2 is for selecting Source 2 for return Loss" |
| Select a measurement | m_Client.SelectTest(clientID, devicename, suitename, "Return Loss", True) |
| Configure the selected measurement (Acquire Parameters) | m_client.SetAcquireParameter(clientID, devicename, suitename,"Transmit Clock Frequency","Average$60");<br>m_client.SetAcquireParameter(clientID, devicename, suitename," Transmit Clock Frequency","Record Length$10" );<br>m_client.SetAcquireParameter(clientID, devicename, suitename," Transmitter Distortion","TX_TCLK$Included" );<br>m_client.SetAcquireParameter(clientID, devicename, suitename," Transmitter Distortion","Average$100" );<br>m_client.SetAcquireParameter(clientID, devicename, suitename," Transmitter Distortion","Disturbing Signal$False" );<br>m_client.SetAcquireParameter(clientID, devicename, suitename," Transmitter Distortion","High Resolution$50" ); |
| Configure the selected measurement (Analyze Parameters) | m_client.SetAnalyzeParameter(clientID, devicename, suitename," Transmitter Distortion","LP Filter$Included")<br>m_client.SetAnalyzeParameter(clientID, devicename, suitename," Transmitter Distortion","RBW$50")<br>m_client.SetAnalyzeParameter(clientID, devicename, suitename," Transmitter Distortion","Center Frequency$75")<br>m_client.SetAnalyzeParameter(clientID, devicename, suitename," Transmitter Distortion","Frequency Span$200") |
| Run with set configurations | m_Client.Run(clientID) |

| Task | Code |
|---|---|
| Wait for the test to complete | Do<br>Thread.Sleep(500)<br>m_Client.Application_Status(clientID)<br>Select Case status<br>Case "Wait"<br>'Get the Current State Information<br>mClient.GetCurrentStateInfo(clientID, WaitingMsbBxCaption, WaitingMsbBxMessage, WaitingMsbBxButtontexts)<br>'Send the Response<br>mClient.SendResponse(clientID, WaitingMsbBxCaption, WaitingMsbBxMessage, WaitingMsbBxResponse)<br>End Select<br>Loop Until status = "Ready" |
| After the test is complete | 'Save all results values from folder for current run<br>m_Client.TransferResult(clientID, logDirname)<br>'Save all waveforms from folder for current run<br>m_Client.TransferWaveforms(clientID, logDirname)<br>'Save all images from folder for current run<br>m_Client.TransferImages(clientID, logDirname) |
| Unlock the server | m_Client.UnlockServer(clientID) |
| Disconnect from server | m_Client.Disconnect() |
| Exit the application | |

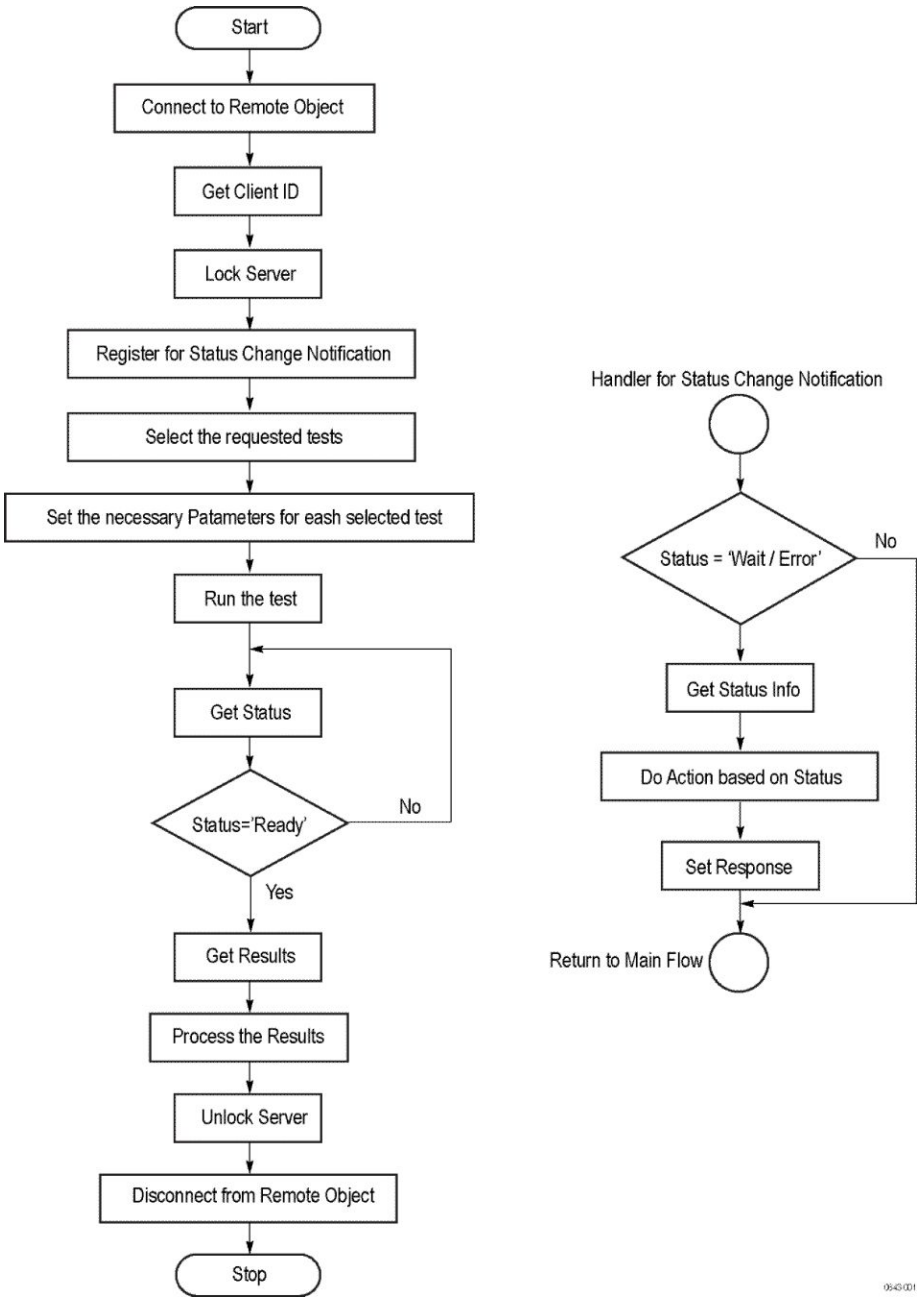
**Table 24: Remote access code example 2**

| Task | Code |
|---|---|
| Start the application | |
| Connect through an IP address. | m_Client.Connect("localhost") 'True or False<br>clientID = m_Client.getClientID |
| Lock the server | m_Client.LockServer(clientID) |
| Disable the Popups | m_Client.SetVerboseMode(clientID, false) |
| Set the DUT ID | m_Client.SetDutId(clientID, "DUT_Name") |
| Select a test | mClient.SelectsingleTest(clientID, "TekExpress Automotive", "100BASE-T1/1000BASE-T1", “Spec 1.0”, "Transmitter Distortion", true) |
| Set Disturbing Signal | mClient.SetAcquireParameter(clientID, "TekExpress Automotive", "100BASE-T1/1000BASE-T1", "Transmitter Distortion","Disturbing Signal$False") |
| Set Record Length | mClient.SetAcquireParameter(clientID, "TekExpress Automotive", "100BASE-T1/1000BASE-T1", "Transmitter Distortion","Record Length$10") |

| Task | Code |
|---|---|
| Run with set configurations | m_Client.Run(clientID) |
| Wait for the test to complete. | Do<br>Thread.Sleep(500)<br>m_Client.Application_Status(clientID)<br>Select Case status<br>Case "Wait" |
| Get the current state information | mClient.GetCurrentStateInfo(clientID, WaitingMsbBxCaption, WaitingMsbBxMessage, WaitingMsbBxButtontexts) |
| Send the response | mClient.SendResponse(clientID, WaitingMsbBxCaption, WaitingMsbBxMessage, WaitingMsbBxResponse)<br>End Select<br>Loop Until status = "Ready" |
| Save results | Save all results values from folder for current run<br>m_Client.TransferResult(clientID, logDirname) |
| Unlock the server | m_Client.UnlockServer(clientID) |

## 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.*** *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.

   ***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 that you want 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 when 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 you complete 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 BRR Application Commands*

*Program Example*

# TekExpress Automotive Ethernet Application Commands

## About TekExpress Automotive Ethernet 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

| Command name | Parameters | Description | Return value | Example |
|---|---|---|---|---|
| Connect() | string ipAddress<br>out string clientID | This method connects the client to the server. Note The client provides the IP address to connect to the server.<br>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.<br>returnval as boolean<br>returnval = m_Client.Connect(ipaddress,m_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...".*

**string ipAddress**

| Name | Type | Direction | Description |
|---|---|---|---|
| ipAddress | string | IN | The ip address of the server to which the client is trying to connect to. 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<br>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.*

## Lock the Server

***NOTE.*** *This method does not lock the UI, but you need this method to set the value that gives the status of the operation after it has been performed.*

| Command name | Parameters | Description | Return value | Example |
|---|---|---|---|---|
| LockSession() | out 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<br>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<br>bool _verbose | This method sets the verbose mode to either True or False.<br>When the value is set to True, any of the message boxes appearing during the application run will be routed to the client machine that is controlling TekExpress.<br>When the value is set to False, then all the message boxes are shown on the server machine. | String that displays the status of the operation after it has been performed.<br>When Verbose mode is set to True, the return value is “Verbose mode turned on. All dialog boxes will be shown to client ...”.<br>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.<br>returnval as string<br>**Verbose mode is turned on**<br>return=m_Client.SetVerboseMode(clientID, True)<br>**Verbose mode is turned off**<br>returnval=m_Client.SetVerboseMode(clientID, False) |

***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<br>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 |

## Set or Get the DUT ID

| Command name | Parameters | Description | Return value | Example |
|---|---|---|---|---|
| SetDutId() | string clientID<br>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<br>string dutId | This method gets the DUT ID of the current set up. | 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<br>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 |

**string dutId**

| Name | Type | Direction | Description |
|---|---|---|---|
| dutId | 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.*

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

***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<br>string device<br>string suite<br>string test<br>string parameterString | This method sets the general parameters. | String that displays the status of the operation after it has been performed.<br>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.<br>returnval as string<br>returnval = m_Client.SetGeneralParameter(clientID, devicename, suitename, string.Empty, parameterstring); |
| SetAnalyzeParameter() | string clientID<br>string device<br>string suite<br>string test<br>string parameterString | This method sets the parameter values in the Ref Levels and Clock Settings tabs in the test configuration section. | String that displays the status of the operation after it has been performed.<br>The return value is "" (an empty String) on success. | mClient = new Client() //m_Client is a reference to the Client class in the Client DLL<br>returnval as string<br>Select Analyze parameter:<br>returnval = mClient.SetAnalyzeParameter(clientID, devicename, suitename, test, parameterstring) |
| SetAcquireParameter() | string clientID<br>string device<br>string suite<br>string test<br>string parameterString | This method sets the parameter values in the Vertical Setup and the Scope Settings tabs in the test configuration section. | String that displays the status of the operation after it has been performed.<br>The return value is ” (an empty string) on success. | mClient = new Client() //m_Client is a reference to the Client class in the Client DLL<br>returnval as string<br>Set Acquire Parameter:<br>returnval = mClient.SetAcquireParameter(clientID, devicename, suitename, test, parameterstring) |

**out string clientID**

| Name | Type | Direction | Description |
|---|---|---|---|
| clientID | string | OUT | Identifier of the client that is connected to the server<br>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 the Configuration Parameters for a Suite or Measurement

| Command name | Parameters | Description | Return value | Example |
|---|---|---|---|---|
| GetGeneralParameter() | string clientID<br>string device<br>string suite<br>string test<br>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.<br>returnval as string |
| GetAnalyzeParameter() | string clientID<br>string device<br>string suite<br>string test<br>string parameterString | This method queries the parameter values in the Ref Level and Clock Settings tabs in the test configuration section. | The return value is the configuration parameter for a given suite or measurement. | mClient = new Client() //m_Client is a reference to the Client class in the Client DLL<br>returnval as string<br>Get Analyze parameter:<br>returnval = mClient.GetAnalyzeParameter(clientID, devicename, suitename, test, parameterstring)<br>GetAnalyzeParameter Examples |
| GetAcquireParameter() | string clientID<br>string device<br>string suite<br>string test<br>string parameterString | This method queries the parameter values in the Vertical Setup and Scope Settings tabs in the test configuration section. | The return value is the configuration parameter for a given suite or measurement. | mClient = new Client() //m_Client is a reference to the Client class in the Client DLL<br>returnval as string<br>Get Acquire Parameter:<br>returnval = mClient.GetAcquireParameter(clientID, devicename, suitename, test, parameterstring)<br>GetAcquireParameter Examples |

**out string clientID**

| Name | Type | Direction | Description |
|---|---|---|---|
| clientID | string | OUT | Identifier of the client that is connected to the server<br>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...".*

**GetAcquireParameter Examples**

This example uses BRR test Transmitter Distortion.

| Parameter | Example |
|---|---|
| TX_TCLK | returnval = mClient.GetAcquireParameter(clientID, devicename, suitename,"Transmitter Distortion","Transmitter Distortion Acquisition $TX_TCLK") |
| Average | returnval = mClient.GetAcquireParameter(clientID, devicename, suitename,"Transmitter Distortion","Transmitter Distortion Acquisition $Average") |
| Disturbing Signal | returnval = mClient.GetAcquireParameter(clientID, devicename, suitename,"Transmitter Distortion","Transmitter Distortion Acquisition $Disturbing Signal") |
| Hi Resolution | returnval = mClient.GetAcquireParameter(clientID, devicename, suitename,"Transmitter Distortion","Transmitter Distortion Acquisition$Hi Resolution") |
| Probe Point Amplitude in V | returnval = mClient.GetAcquireParameter(clientID, devicename, suitename,"Transmitter Distortion","Transmitter Distortion Acquisition $Probe Point Amplitude in V") |
| Distortion Amplitude in V | returnval = mClient.GetAcquireParameter(clientID, devicename, suitename,"Transmitter Distortion","Transmitter Distortion Acquisition $Distortion Amplitude in V") |
| Distortion Frequency in MHz | returnval = mClient.GetAcquireParameter(clientID, devicename, suitename,"Transmitter Distortion","Transmitter Distortion Acquisition $Distortion Frequency in MHz") |
| DUT Amplitude in V | returnval = mClient.GetAquireParameter(clientID, devicename, suitename,"Transmitter Distortion","Transmitter Distortion Acquisition $DUT Amplitude in V") |

**GetAnalyzeParameter Examples**

This example uses BRR test Transmitter Distortion.

| Parameter | Example |
|---|---|
| LP Filter | returnval =mClient.GetAnalyzeParameter(clientID, devicename, suitename,"Transmitter Distortion","LP Filter") |

## Select a Measurement

| Command name | Parameters | Description | Return value | Example |
|---|---|---|---|---|
| SelectTest() | string clientID<br>string device<br>string suite<br>string test<br>bool isSelected | This method selects or deselects a given test. | String that displays the status of the operation after it has been performed.<br>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<br>returnval as string<br>**Select test: (test name)**<br>returnval=mClient.SelectTest(clientID, devicename, suitename, testname, isSelected);)<br>Select a specific test. |

### Select a Specific Test

| Test | Command |
|---|---|
| Transmit Clock Frequency | For selecting the test:<br>returnval=m_Client.SelectTest(clientID, devicename, suitename, "Transmit Clock Frequency", True);<br>For deselecting the test:<br>returnval=m_Client.SelectTest(clientID, devicename, suitename, "Transmit Clock Frequency", False); |
| Transmitter Master Timing Jitter | For selecting the test:<br>returnval=m_Client.SelectTest(clientID, devicename, suitename, "Transmitter Timing Jitter - Master Jitter", True);<br>For deselecting the test:<br>returnval=m_Client.SelectTest(clientID, devicename, suitename, "Transmitter Timing Jitter - Master Jitter", False) |
| Transmitter Slave Timing Jitter | For selecting the test:<br>returnval=m_Client.SelectTest(clientID, devicename, suitename, "Transmitter Timing Jitter - Slave Jitter", True);<br>For deselecting the test:<br>returnval=m_Client.SelectTest(clientID, devicename, suitename, "Transmitter Timing Jitter - Slave Jitter", False) |

| Test | Command |
|---|---|
| Transmitter Output Droop | For selecting the test:<br>returnval=m_Client.SelectTest(clientID, devicename, suitename, "Transmitter Output Droop", True);<br>For deselecting the test:<br>returnval=m_Client.SelectTest(clientID, devicename, suitename, "Transmitter Output Droop", False) |
| Transmitter Power Spectral Density | For selecting the test:<br>returnval=m_Client.SelectTest(clientID, devicename, suitename, "Transmitter Power Spectral Density", True);<br>For deselecting the test:<br>returnval=m_Client.SelectTest(clientID, devicename, suitename, "Transmitter Power Spectral Density", False); |
| Transmitter Distortion | For selecting the test:<br>returnval=m_Client.SelectTest(clientID, devicename, suitename, "Transmitter Distortion", True);<br>For deselecting the test:<br>returnval=m_Client.SelectTest(clientID, devicename, suitename, "Transmitter Distortion", False); |
| Return Loss | For selecting the test:<br>returnval=m_Client.SelectTest(clientID, devicename, suitename, "Return Loss", True);<br>For deselecting the test:<br>returnval=m_Client.SelectTest(clientID, devicename, suitename, "Return Loss", False); |

***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<br>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 |

## Select a Suite

| Command name | Parameters | Description | Return value | Example |
|---|---|---|---|---|
| SelectSuite() | string clientID<br>string device<br>string suite<br>bool isSelected | This method selects or deselects a given suite.<br>Setting parameter is selected to True, you can select a suite.<br>Setting parameter is selected to False, you can deselect a suite. | String that gives the status of the operation after it has been performed.<br>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.<br>returnval as string<br>**Select Suite (Default):**<br>returnval=m_Client.SelectTest(clientID, "BRR", "BRR", True) |

***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<br>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 |

## Run with Set Configurations or Stop the Run Operation

| Command name | Parameters | Description | Return value | Example |
|---|---|---|---|---|
| Run() | string clientID | Runs the selected tests. Note Once the server is set up and is configured, it can be run 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_Clientis 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) |

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

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

## 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 client. After expiry of this timeout period, the server is automatically unlocked. | 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) |

***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 |

**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. Else, the client is prompted to provide a valid timeout period.

## Wait for the Test to Complete

The commands in this group are executed while tests are running. The GetCurrentStateInfo() and SendResponse() commands are executed when application is running and in 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.<br>returnval as string<br>returnval=m_Client.ApplicationStatus(clientID) |
| QueryStatus() | string clientID<br>out string[] status | This is 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.<br>On success the return value is “Transferred...”. | m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.<br>returnval as string<br>returnVal=m_Client.QueryStatus(clientID, out statusMessages)<br>if ((OP_STATUS)returnVal == OP_STATUS.SUCCESS)<br>return “Status updated...”<br>else<br>return CommandFailed(returnVal) |
| GetCurrentStateInfo()<br>***NOTE.*** *This command is used when the application is running and is in the wait or error state.* | string clientID<br>out string caption<br>out string message<br>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.<br>This function fills up 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.<br>m_Client.GetCurrentStateInfo(clientID, caption,message, buttonTexts) |

| Command name | Parameters | Description | Return value | Example |
|---|---|---|---|---|
| SendResponse()<br>***NOTE.*** *This command is used when the application is running and is in the wait or error state.* | string clientID<br>out string caption<br>out string message<br>string response | After receiving the additional information using the method GetCurrentStateInfo(), the client can decide on the response to send and send the response to the application using this function. The response should be one of the strings that was earlier received 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) |

***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 NOT FOUND and the message displayed is "Server not found...Disconnect!".*

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

Ready: Test configured and ready to start

Running: Test running

Paused: Test paused

Wait: A popup that needs your inputs

Error: An error has occurred

**out string clientID**

| Name | Type | Direction | Description |
|---|---|---|---|
| clientID | string | OUT | Identifier of the client that is connected to the server<br>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 run |

**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 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) |

## After the test is complete

| Command name | Parameters | Description | Return value | Example |
|---|---|---|---|---|
| GetPassFailStatus() | string clientID<br>string device<br>string suite<br>string test | This method gets the pass or fail status of the measurement after test completion.<br>***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.<br>returnval as string<br>returnval=m_Client.GetPassFailStatus(clientID, device, suite, "Transmitter Output Droop") // Pass or Fail |
| GetResultsValue() | string clientID<br>string device<br>string suite<br>string test<br>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.<br>returnval as srting<br>returnval=m_Client.GetResultsValue(clientID, "TekExpress Automotive Ethernet", "TekExpress Automotive Ethernet", "Transmitter Output Droop", "Measured Value") |

| Command name | Parameters | Description | Return value | Example |
|---|---|---|---|---|
| GetResultsValueForSubMeasurements() | string clientID<br>string device<br>string suite<br>string test<br>string parameterString<br>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.<br>returnval as string<br>**Single-ended Low Level Voltages DP (nS)**<br>returnval=m_Client.GetResultsValueForSubMeasurements(clientID, "TekExpress Automotive Ethernet", "TekExpress Automotive Ethernet", "Transmitter Output Droop", "Transmitter Output Droop", "Measured Value", 0) //For DP wfm<br>returnval=m_Client.GetResultsValueForSubMeasurements(clientID, "TekExpress Automotive Ethernet", "TekExpress Automotive Ethernet", "Transmitter Output Droop", "Measured Value", 1) //For DN wfm |

| Command name | Parameters | Description | Return value | Example |
|---|---|---|---|---|
| GetReportParameter() | string clientID<br>string device<br>string suite<br>string test<br>string parameterString | This method gets the general report details such as oscilloscope model, TekExpress version, and TekExpress Automotive Ethernetversion. | The return value is the oscilloscope model, TekExpress version, and TekExpress Automotive Ethernet version. | m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.<br>returnval as string<br>**Oscilloscope Model**<br>returnval= m_Client.GetReportParameter(client_Id,device, suite,'','Scope Model')<br>**TekExpress Version**<br>returnval= m_Client.GetReportParameter(client_Id,device,suite,'','TekExpress Version')<br>**TekExpress Automotive Ethernet Version**<br>returnval= m_Client.GetReportParameter(client_Id,device,suite,'','Application Version') |
| TransferReport() | string clientID<br>string filePath | This method transfers the report generated after the run.<br>The report contains the summary of the run.<br>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.<br>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.<br>returnval as string<br>returnval=m_Client.TransferReport(clientID,"C:\Report”) |

| Command name | Parameters | Description | Return value | Example |
|---|---|---|---|---|
| TransferWaveforms() | string clientID<br>string filePath | This method transfers all the waveforms from the folder for the current run.<br>***NOTE.*** *For each click of Run button, a folder is created in the X: drive. Transfer the waveforms before clicking the Run button.* | String that gives the status of the operation after it has been performed. Transfers all the waveforms in the form of a string. On success the return value is “Transferred...”. | m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.<br>returnval as string returnval=m_Client.TransferWaveforms(clientID,"C:\Waveforms”) |
| TransferImages() | string clientID<br>string filePath | This method transfers all the images (screenshots) from the folder for the current run (for a given suite or measurement).<br>***NOTE.*** *For each click of Run button, a folder is created in the X: drive. Transfer the waveforms before clicking the Run button.* | 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.<br>returnval as string returnval=m_Client.TransferImages(clientID, "C:\Waveforms”) |

***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<br>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 |

**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.*

## Save Recall or Check if a Session is Saved

| Command name | Parameters | Description | Return value | Example |
|---|---|---|---|---|
| CheckSessionSaved() | string clientID<br>out bool saved | This method is called when a check is to be made to know if 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.<br>returnval as string<br>returnval=m_Client.CheckSessionSaved(m_clientID, out savedStatus) |
| RecallSession() | string clientID<br>string name | Recalls a saved session. The name of the session is provided by the client. | String that gives the status of the operation after it has been performed.<br>The return value is “Session Recalled...”. | m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.<br>returnval as string<br>returnval=m_Client.RecallSession(clientID, savedSessionName) |
| SaveSession() | string clientID<br>string name | Saves the current session. The name of the session is provided by the client. | String that gives the status of the operation after it has been performed.<br>The return value is "Session Saved..."/"Failed...". | m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.<br>returnval as string<br>returnval=m_Client.SaveSession(clientID, desiredSessionName) |
| SaveSessionAs() | string clientID<br>string name | Saves the current session in a different name every time this method is called. The name of the session is provided by the client. | String that gives the status of the operation after it has been performed.<br>The return value is "Session Saved...". | m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.<br>returnval as string<br>returnval=m_Client.SaveSessionAs(clientID, desiredSessionName) |

***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<br>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.

**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.

**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.

## 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) |

***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<br>clientID = unique number + ipaddress of the client. For example, 1065–192.157.98.70* |

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

## Disconnect from the Server

| Command name | Parameters | Description | Return value | Example |
|---|---|---|---|---|
| Disconnect() | string clientID | This method disconnects the client from the server it is connected to. Note | Integer value that gives the status of the operation after it has been performed.<br>1 for Success<br>–1 for Failure | m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.<br>returnval as string<br>returnval=m_Client.Disconnect(m_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<br>clientID = unique number + ipaddress of the client. For example, 1065–192.157.98.70 |

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

## Handle Error Codes

The return value of the remote automations at the server-end is OP_STATUS, which is changed 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...".*

# Index

## L



## M



## O



## P



## Q



## R



## S



## T

## U



## V



## W