



400G-M4
Application Help
Printable Application Help



077-1471-01



400G-M4
Application Help
Printable Application Help

Copyright © Tektronix. All rights reserved. Licensed software products are owned by Tektronix or its subsidiaries or suppliers, and are protected by national copyright laws and international treaty provisions.

Tektronix products are covered by U.S. and foreign patents, issued and pending. Information in this publication supersedes that in all previously published material. Specifications and price change privileges reserved.

TEKTRONIX and TEK are registered trademarks of Tektronix, Inc.

This document supports 400G-M4 software version 1.0.X and greater, for use with the DSA8300 oscilloscope.

Application help system part number: 076-0423-01

Printable version of the application Help (PDF): 077-1471-01

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

Introduction

Welcome	1
Computer requirements	2
DSA8300 sampling oscilloscope requirements	3
TekVISA software	3
Option key	3
Documentation	3

Display elements

Elements of the display	5
-------------------------------	---

Menu bar operations

File menu	7
Scope menu	7
Plot menu	9
Help menu	9

Horizontal Settings

Horizontal settings	11
---------------------------	----

Measurement selections

Measurement selections	13
Measurement configuration	
TDECQ configuration	13

Mode selection

Mode selection	15
----------------------	----

Bandwidth enhancement

Bandwidth enhancement (BWE)	17
-----------------------------------	----

Clear button

Clear button	19
--------------------	----

Run button

Run button	21
------------------	----

Offline mode

Offline mode	23
--------------------	----

Online mode

Online mode.....	25
------------------	----

Results

Results elements	27
Symbol level measurement	27
Eye results	28
TDECQ and PAM4 results	29

Oscilloscope connection tips

Check network access.....	31
TekVISA setup and troubleshooting.....	31
Firewall exceptions.....	33

Remote control

Remote control introduction.....	35
Handshaking protocol.....	35
Setting up the PI environment.....	36
Variable:Value Commands	
Syntax.....	37
Variable name arguments and queries	38
Programming examples	
Perl program example: configure and operate 400G-M4	51
Python example	53

Feedback

Feedback.....	55
---------------	----

Index

Welcome

The 400G-M4 analysis application provides PAM4 signaling analysis, including TDECQ (Transmitter and Dispersion Eye Closure Quaternary) measurement. The application brings together PAM4 optical measurements in a simple and easy to use application.

The application is intended to be installed on a PC and can operate in either offline or online mode.

- **Offline mode:** allows you to analyze a waveform file saved from the 80SJNB application (running on the DSA8300 sampling oscilloscope) or a waveform file saved from the 400G-M4 application acquired from a connected DSA8300 sampling oscilloscope.
- **Online mode:** allows you to acquire directly from a connected DSA8300 sampling oscilloscope and analyze acquired waveform data.

NOTE. *Online mode requires the installation of an option key on the target DSA8300 oscilloscope to enable establishing a connection from the 400G-M4 application to the DSA8300. Refer to the documentation provided with the DSA8300 for instructions of adding a new option key.*

Key features

The 400G-M4 application performs comprehensive analysis of PAM4 signals, including TDECQ measurements and other optical measurements.

400G-M4 provides the following features:

- Show results as numeric and graphical display
- Perform TDECQ (Transmitter and Dispersion Eye Closure Quaternary)
- PAM4 signal characterization measurement such as level and eye measurements
- Standard IEEE TDECQ Measurements (802.3bs, Section 121.8.5)
- Acquire complete pattern waveform with user defined Samples/UI
- Display 2-D eye diagrams
- Save acquisition results to a data file

See also:

[Computer requirements \(see page 2\)](#)

[DSA8300 sampling oscilloscope requirements \(see page 3\)](#)

[Option key \(see page 3\)](#)

Computer requirements

The 400G-M4 application is designed to run on a PC running the Windows operating system. A high performance PC is recommended to minimize computation time to analyze the waveform data.

These are the minimum requirements to successfully run the 400G-M4 application. A PC with higher performance results in faster measurements.

- **Operating system:** Microsoft Windows 10 (64 bit) operating system.
- **Screen resolution:** 1920 x 1080.

NOTE. *Installing the 400G-M4 application directly on a DSA8300 sampling oscilloscope is not permitted.*

Software requirements

The 400G-M4 application requires the following software to be installed on the PC.

- **TekVISA:** TekVISA version 4.2.0.10 is recommended. Installation is available during 400G-M4 installation.
- **Matlab runtime:** Matlab 2017b version 9.3 is required. The Matlab runtime is available from MathWorks® (<https://www.mathworks.com>).

See also:

[TekVISA software \(see page 3\)](#)

The 400G-M4 application is installed on a PC and can operate in either offline or online mode.

- **Offline mode:** allows you to analyze a file saved from the 80SJNB application (available for use with the DSA8300 sampling oscilloscopes) or a saved file from the 400G-M4 application acquired from a connected DSA8300 sampling oscilloscope.
- **Online mode:** allows you to acquire directly from a connected DSA8300 sampling oscilloscope and analyze acquired waveform data.

To operate in online mode, the target DSA8300 oscilloscope must have the 400G-M4 option key installed (to allow a connection). Refer to [Option key \(see page 3\)](#) for more information.

DSA8300 sampling oscilloscope requirements

The 400G-M4 application requires the following software to be installed on the connected DSA8300 sampling oscilloscope.

- **Scope Noise Characteristics:** This application is provided with the 400G-M4 installation. It is a separate installation file (bundled with the 400G-M4 download package) that is to be installed on the connected DSA8300 to enable the 400G-M4 application to automatically obtain the scope noise value of the DSA8300.
- The option key for the 400G-M4 application must be installed on the DSA8300 to which the application connects for online mode of operation.

TekVISA software

TekVISA is preinstalled on the DSA8300 sampling oscilloscopes, but to use this protocol to connect and communicate with a DSA8300 sampling oscilloscope, TekVISA must also be installed on the PC (where the 400G-M4 application resides).

TekVISA is available with the 400G-M4 application installation file or can be downloaded for free from the Tektronix website (www.tek.com). Search for TekVISA Connectivity software.

Option key

To enable the 400G-M4 application to connect directly to a DSA8300 sampling oscilloscope (online mode), the target DSA8300 must have a 400G-M4 option key installed.

Contact Tektronix to purchase a 400G-M4 option key.

Refer to the DSA8300 application help system for instructions to install option keys.

Documentation

The application help system is available from the Help menu.

In addition to the help system, a printable version of the help is provided as a PDF in the following folder:

C:\Users\Public\Tektronix\TekApplications\400G-M4\PIExamples.

Elements of the display

Connect to oscilloscope Show plot Configure measurements

Horizontal settings

Choose measurements

Mode select (offline/online)

Enable bandwidth enhancement

Measurement results of selected measurements

Status

Clear all measurement results Start analysis (offline or online)

Tektronix 400G-M4 (TDECQ Analysis)

File Scope Plot Help

Horizontal Settings
Baud Rate: 53.125 GBd Pattern Length: 8191 Wfm Samples/UI: 10

Measurements
 TDECQ Eye Symbol Level **Config**

Off Line Ch1 Ch2 Ch3 Ch4

Load Waveform: **Browse** Scope Noise: 6 uW
examplePRBS13.mat

BW Enhancement
 BWE Enable Bandwidth: HW only BT 4th Order BW: 19.9219 GHz S-parameter: **Browse**
Load S-parameter file

Results

Symbol level	Mean	StdDev	P-P	Eye	Thresh	Offset	H_eye	V_eye
V_D(3)	920.3 uW	9.042 uW	61.34 uW	Upper	790.7 uW	83.73 fs	8.457 ps	197.5 uW
V_C(2)	661.9 uW	7.65 uW	56.2 uW	Middle	527.2 uW	-94.2 fs	8.855 ps	216.2 uW
V_B(1)	393.7 uW	6.968 uW	53.52 uW	Lower	260.3 uW	205.5 fs	7.947 ps	210.1 uW
V_A(0)	134 uW	7.373 uW	45.96 uW					

Power (mW) vs Time (UI) plot showing eye diagram with red grid lines.

Measurement	Value
TDECQ	0.3941 dB
OMAAouter	-1.017 dBm
ER	8.444 dB
AOP	-2.773 dBm
RLM	97.18 %
Level Deviation	1.544 %
Level Thickness	1.892 %

FFE Tap Values: -0.029017 1.0598 -0.042904 0.012969 -0.00085498

Total measurement time is 7.8345 second. **Clear** **Run**

File menu

The file menu provides the following operations:

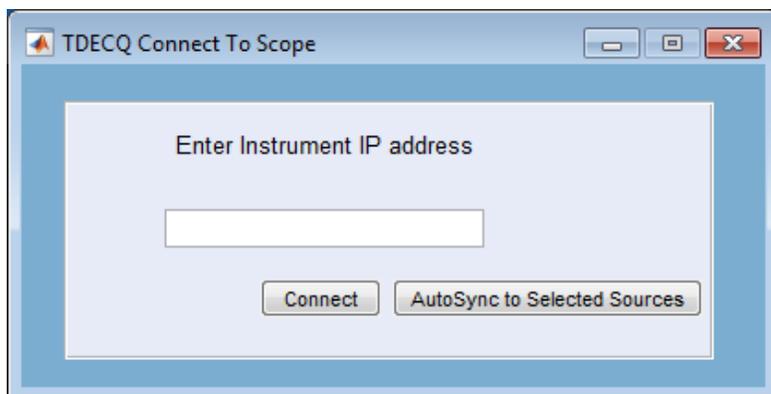
Table 1: File menu

Setting	Description
Save Waveform	<p>Saving waveforms is only available when using the online mode.</p> <p>You can save the acquired waveform (or waveforms if multiple channels are enabled) from the connected DSA8300 oscilloscope. A dialog screen opens to allow you to navigate to name and save the waveform.</p> <p>Use the dialog screen to provide a base filename. The filename will be appended with the channel source of the waveform.</p>
Save Setup	<p>Use the dialog screen to navigate to a location to save the setup file to recall at a later time. Setup files use a file extension of .gm4.</p> <p>Setup files do not save information relating to the oscilloscope or modules, such as:</p> <ul style="list-style-type: none"> ■ Scope Noise ■ Vertical setting ■ Channels enabled ■ Module related information
Recall Setup	Use the dialog screen to navigate to a saved setup file to restore the application to a known setup.
Exit	Exits the application.

The File menu operations are unavailable if an analysis is currently running.

Scope menu

Selecting Scope opens the TDECQ Connect To Scope dialog screen.



Enter the IP address or the computer name of the oscilloscope you want to connect to. The IP address can be found by viewing the Windows network connection status. The computer name can be found in the System properties.

Connect. Select the Connect button to make the connection. (The Connect button changes to Disconnect if a connection is established.)

The Status area at the bottom of the screen also indicates a successful connection.

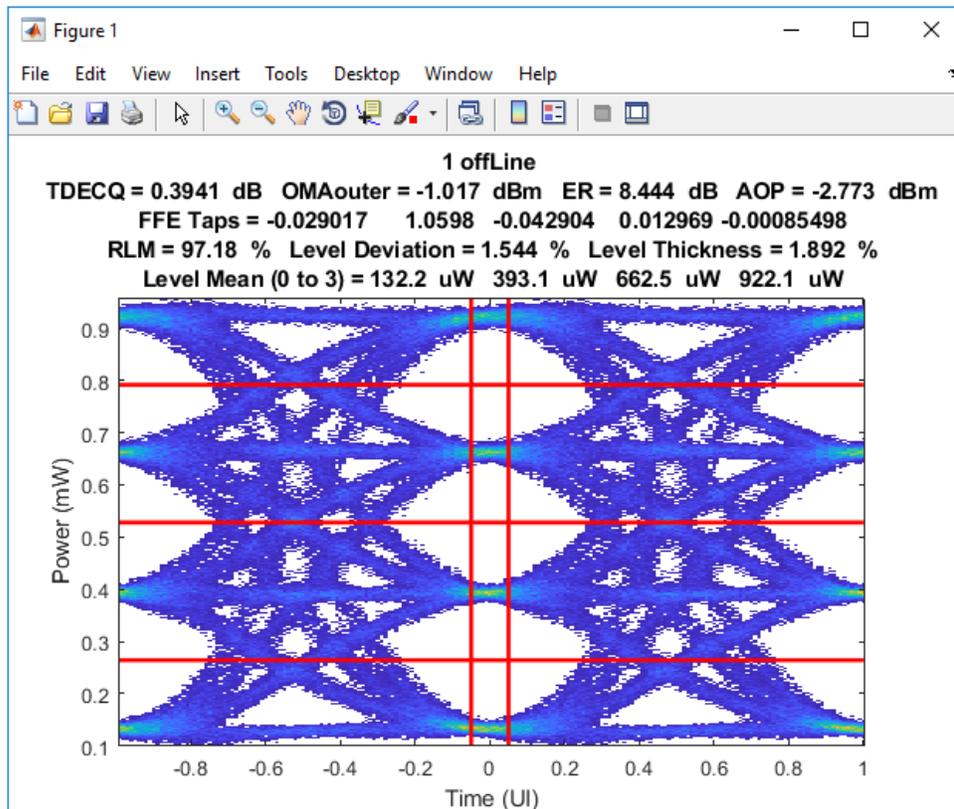
AutoSync to Selected Sources. During the connection process, the 400G-M4 application detects the modules installed in the oscilloscope along with their characteristics (such as the module type, serial number, filters or bandwidth). If you change a module or change a module's characteristics, use the AutoSync to Selected Sources button to easily update the 400G-M4 application without having to reconnect.

If you are having difficulty connecting to your oscilloscope, see the section [Connection help and tips \(see page 31\)](#).

NOTE. *The 400G-M4 application can only connect to one DSA8300 at a time.*

Plot menu

The Plot menu allows you to open the TDECQ eye plot display in Matlab.



Help menu

In addition to accessing the application help system, use the help button to obtain the application software version number.

This can help when contacting Tektronix about the application and to check if there are newer versions available from the Tektronix website (www.tek.com).

A printable version of the help system is provided as a PDF in the following folder:

C:\Program Files\TekApplications\400G-M4.

Horizontal settings

The horizontal settings are only applicable when using the online mode (connected to an oscilloscope). The settings are global for all online channels. In offline mode, these settings are automatically extracted from the waveform file.

Set the values as appropriate for the type of optical signal you are analyzing.



The screenshot shows a 'Horizontal Settings' panel with three input fields: 'Baud Rate' set to 53.125 GBd, 'Pattern Length' set to 8191, and 'Wfm Samples/UI' set to 10.

Table 2: Horizontal settings

Setting	Description
Baud Rate	Enter the GigaBaud rate of the signal you intend to acquire.
Pattern Length	Enter the Pattern Length of the signal you intend to acquire.
Wfm Samples/UI	Enter the number of waveform samples per user interval.

Measurement selections

The measurement selection includes all the available optical measurements.

The selected measurements apply to the offline mode (file analysis) and to the online mode for all channels.



Table 3: Measurement selections

Measurements	Description
TDECQ	<p>Includes the following measurements to the results section.</p> <ul style="list-style-type: none"> ■ TDECQ: Transmitter and dispersion eye closure for PAM4 defined by IEEE 802.3bs (draft 3.5, dated 10-October-2017). ■ OMAouter: The outer Optical Modulation Amplitude. ■ ER: The extinction ratio of the highest and lowest optical power levels of a PAM4 optical signal. ■ AOP: The average optical power of a PAM4 optical signal.
Eye	<p>Includes the following measurements to the results section.</p> <ul style="list-style-type: none"> ■ RLM: Level separation mismatch ratio. ■ Level Deviation: The average deviation of level spacing from the ideal spacing. ■ Level Thickness: The averaged, normalized level standard deviation at minimum inter-symbol interference. ■ The Threshold/Offset/Horizontal Eye (H_eye)/Vertical Eye (V_eye) for each PAM4 eye.
Symbol Level	<p>Includes all the Symbol level measurement results.</p> <ul style="list-style-type: none"> ■ Symbol Levels: Provides the Mean, Standard Deviation, and Pk-Pk values of the V_D(3), V_C(2), V_B(1), V_A (0) symbol levels.

TDECQ configuration

The measurement configuration applies to the TDECQ measurement selection to further define the measurement parameters, such as the FFE taps.

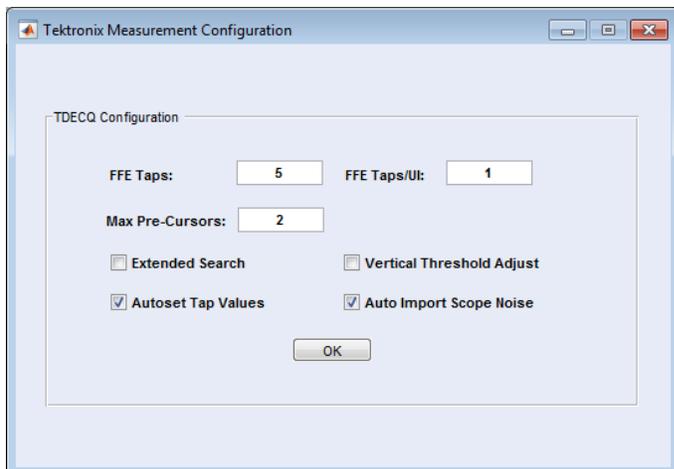
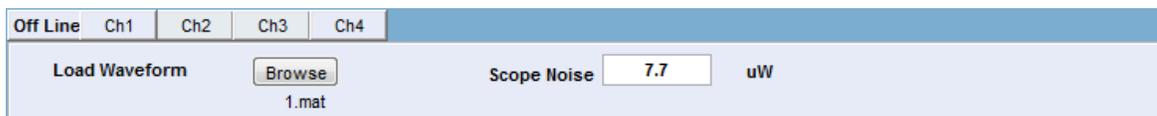


Table 4: Measurement configuration

Measurements	Description
FFE taps	Enter the number of taps. The number of taps must be an odd value.
FFE Taps/UI	Enter the number of taps per unit interval (symbol) for FFE. The range of the value is 1 – 10.
Max Pre-Cursors	Enter the maximum number of pre-cursor taps for FFE. The value must be less than the number of FFE taps.
Extended Search	This is disabled by default. When enabled, the algorithms perform an extended search for the optimal FFE taps to minimize the TDECQ value (but increases the time to complete). When disabled, the FFE taps are adapted faster but may be less optimal.
Vertical Threshold Adjust	This is disabled by default. When enabled, the sub-eye threshold levels are allowed to adjust by $\pm 1\%$ of the OMAouter. IEEE 802.3cd allows this option to be selected for TDECQ measurement. When disabled, the sub-eye threshold levels are determined by OMAouter and the average optical power.
Autoset Tap Values	When enabled, the application automatically calculates optimized FFE taps to minimize the TDECQ value. When disabled, the current FFE taps are used. You can manually input the FFE tap values via the FFE Tap Values area at the bottom of the application main screen. This is enabled by default.
Auto Import Scope Noise	This is enabled by default. When enabled, the application automatically retrieves the measured oscilloscope noise from the connected DSA8300 oscilloscope when in online mode. The Scope Noise Characteristics application must be installed and run prior to importing the oscilloscope noise.

Mode selection



The mode selection allows you to either analyze a saved data file (offline mode) or make a live acquisition (online mode) directly from one (or all) of the four optical channel selections available on the DSA8300.

The offline mode requires you to navigate to a saved data file for analysis.

The online mode requires a connection to a DSA8300 oscilloscope.

For more information, see the topics [Offline mode \(see page 23\)](#) and [Online mode \(see page 25\)](#).

Bandwidth enhancement (BWE)

Bandwidth enhancement can be used whether analyzing a saved file (offline mode) or when connected to an instrument (online mode). The feature operates the same for either mode.

When using in online mode, the feature can be enabled or disabled independently for each channel.

Table 5: Measurement configuration

Item	Description
BWE Enable	<p>Enables the bandwidth enhancement feature for the selected file (Offline mode) or the channel (Online mode).</p> <p>The Bandwidth Enhancement controls allow you to use DSP processing to achieve more accurate oscilloscope channel response. For example, the BWE can make the oscilloscope channel response to be ORR compliant even if the hardware is not.</p>
Bandwidth	<p>Select the bandwidth filter to use during the analysis.</p> <ul style="list-style-type: none"> ■ HW only: uses the instrument hardware. ■ 0.5 Baud rate: sets the bandwidth of the 4th order Bessel-Thomson filter to be $0.5 \times$ Baud Rate. ■ 0.75 Baud rate: sets the bandwidth of the 4th order Bessel-Thomson filter to be $0.75 \times$ Baud Rate. ■ User: This allows a user to dial in the bandwidth of a 4th order 4th order Bessel-Thomson filter.
BT 4th Order BW	Enter the Bessel -Thomson reference receiver filter when the Bandwidth is set to User.
S-parameter	<p>You can load a scattering parameter (S-parameter) file that characterizes the optical sampling module available for the DSA8300.</p> <p>NOTE. An S-parameter file is supplied with the 80C20 and 80C21 optical sampling modules. These files are unique to each module.</p> <p>Use the Browse button to navigate to a saved S-parameter file. The S-parameter file name contains the module type, serial number, and hardware filter information.</p>

Clear button

The Clear button removes all measurement results, including the TDECQ plot display.

Run button

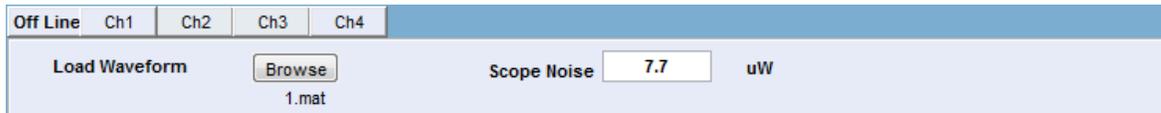
Press the Run button to start the analysis cycle. At least one measurement must be selected.

When in offline mode, a file must be loaded before pressing the Run button.

When in online mode, a connection must be established to a DSA8300, containing supported modules. The analysis cycle includes acquiring waveforms and performing the selected measurements.

Offline mode

Using the application in Offline mode simply means you can analyze a saved waveform file without connecting to a sampling oscilloscope.



Click the Off Line tab and use the Browse button to navigate to waveform file (saved from the 80SJNB application or the 400G-M4 application) for analysis.

TDECQ measurement requires the input of scope noise, σ_s , defined as the standard deviation of the noise of the optical probe and oscilloscope combined. It is calibrated with no optical input signal and the same settings used to capture the optical signal, after the application of applicable ORR.

Loading a waveform file automatically inputs the scope noise value associated with the loaded file if available.

Once the file is loaded, select the measurements you want returned, and then press the Run button. The various result panels will be populated based on the selected measurements.

Online mode

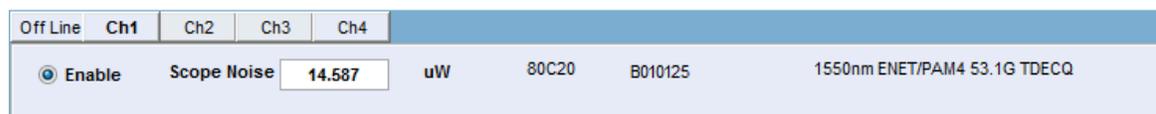
Online mode allows you to acquire waveforms directly from the connected oscilloscope, analyze the data, and display the results. (A connection to the oscilloscope must be made prior to selecting the Run button. See the [Scope menu \(see page 7\)](#).)

The 400G-M4 applications currently supports the following optical modules (installed in a DSA8300 mainframe):

- 80C17
- 80C18
- 80C20
- 80C21
- 80C10C

NOTE. *If you would like to use a non-supported module, add it to "SupportedModules.xml" file located in the installation directory. Performance and measurement accuracy are not guaranteed for non-supported modules.*

Selecting any of the channel tabs enters the Online mode.



With a channel tab open, select Enable to enable acquiring waveforms from that channel of the oscilloscope. With a connection established, the information about the optical module installed in that channel is displayed.

NOTE. *If the module installed in the channel is not supported, the Enable button is inactive.*

There are four channels available on the DSA8300 oscilloscope. You can enable any number of channels. Each channel must be enabled in its channel tab.

TDECQ measurement requires the input of scope noise, σ_s , defined as the standard deviation of the noise of the optical probe and oscilloscope combined. It is calibrated with no optical input signal and the same settings used to capture the optical signal, after the application of applicable ORR. You must set the scope noise value for each channel intended to use for testing. This value can be automatically imported from the connected DSA8300. See Scope Noise auto import below.

When the analysis is started, the waveforms are acquired from each enabled channel.

The results displayed are for the current channel. Use the channel tabs to display the results for each channel.

Scope noise auto import

The Scope Noise value can be imported directly from the connected DSA8300 if the following conditions are met:

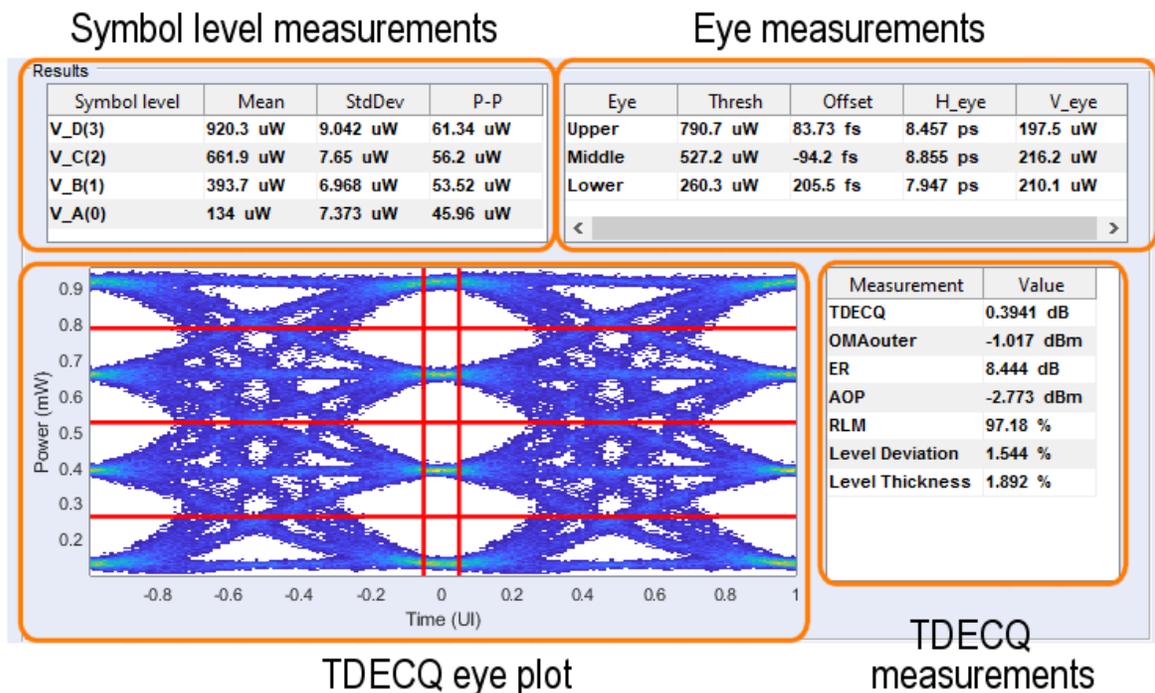
- The Scope Noise Characteristics application must be installed on the connected DSA8300.
- The Scope Noise Characteristics application had been run on the DSA8300 (with modules installed) prior to starting the analysis cycle of the 400G-M4 application.
- The Measurement Configuration must have Auto Import Scope Noise enabled.

Results elements

The results area is populated with the latest measurements from the latest analysis. The areas populated with results depend on the selected measurements to make during analysis.

When in offline mode, the results display the measurements for the waveform file loaded.

When in online mode, the results displayed are for the selected channel tab. Use the channel tabs to see results for the various channels. (There are no results for a channel if the channel was not enabled prior to starting the analysis.)



Symbol level measurement

When the Symbol Level measurement is selected, the Symbol Levels result area is populated for the following four symbol levels:

- **V_D(3):** For the top symbol level, this is the voltage at the center of the unit interval across all instances of symbol 3s in the complete waveform. The mean, standard deviation and peak-to-peak range are shown. Only the mean value is used in the computation of RLM .
- **V_C(2):** Same as V_D(3), except for the symbol 2's.

- **V_B(1):** Same as V_D(3), except for the symbol 1's.
- **V_A(0):** Same as V_D(3), except for the symbol 0's.

Table 6:

Result	Description
Mean	Mean value (watts) of the measurements.
Standard deviation	Standard deviation of the measurements.
P-P	Peak-to-peak value of the measurements.

Eye results

When the Eye measurement is selected, the Eye result area is populated with the eye measurements for all three eye openings (Upper, Middle, Lower).

Table 7:

Result	Description
Threshold	The reference voltage levels for each PAM4 TDECQ eye.
Offset	The horizontal offset, in time, from the center of the sub-eye to the measured center of the eye as rendered, at the corresponding reference voltage.
H_eye (Width)	The eye width of the corresponding (upper, middle or lower) eye.
V_eye (Height)	The eye height of the corresponding (upper, middle or lower) eye.

TDECQ and PAM4 results

When the TDECQ measurement is selected, the TDECQ result is provided along with various PAM4 measurements.

Table 8:

Result	Description
TDECQ	As defined by IEEE 802.3bs, Section 121.8.5.
OMA _{outer}	Outer Optical Modulation Amplitude.
ER	Extinction Ratio
AOP	Average Optical Power The average power of a PAM4 optical signal, reported in dBm.
RLM	Level separation mismatch ratio.
Level Deviation	A measure of the deviation of the vertical intervals between levels from perfectly equal spacing, where 0% represents perfect spacing. The measurement is computed as: $\text{Level Deviation} = \frac{1}{3} \sum_{ij=10,21,32} \frac{\left d_{ij} - \frac{(P_k - P_k)}{3} \right }{\frac{(P_k - P_k)}{3}} \times 100\%$
Level Thickness	An overall measure of the vertical thickness of the symbol levels in the correlated waveform, where an ideal signal with maximally open eyes would have a thickness of 0%. The measurement is computed as: $\text{Level Thickness} = \frac{1}{4} \sum_{i=0}^3 \frac{\sigma_i}{(P_k - P_k) \div 2} \times 100\%$

Check network access

The first step to troubleshoot a connection problem is to check the network access.

Ensure that the DSA8300 is connected to the network, and the TekScope application is running.

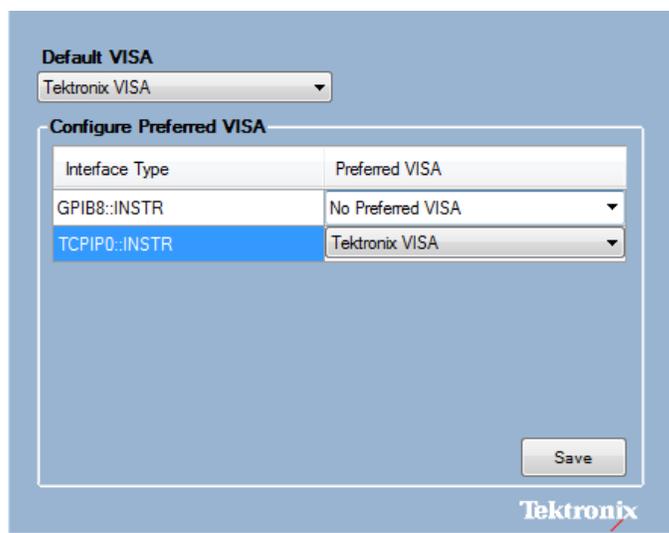
TekVISA setup and troubleshooting

- Verify that TekVISA is installed on the PC.
 - TekVISA is required to be running on both the PC and the target oscilloscope.

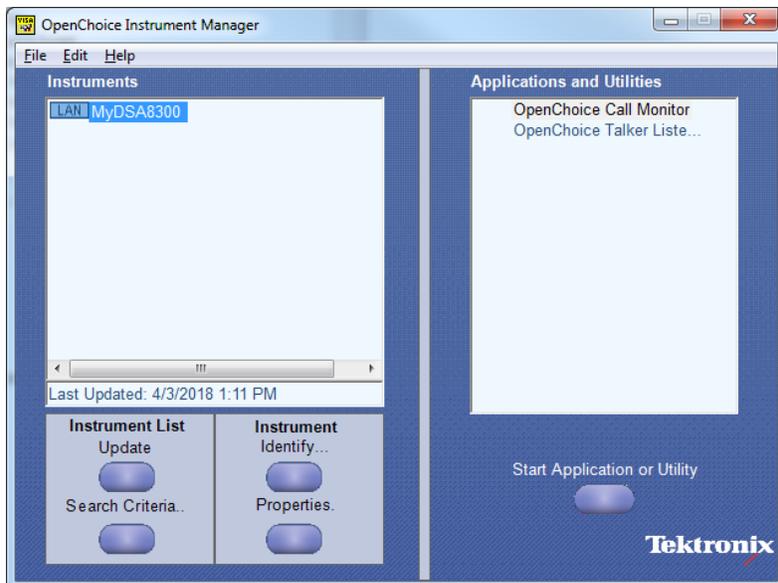
Look for the TekVISA icon in task bar.



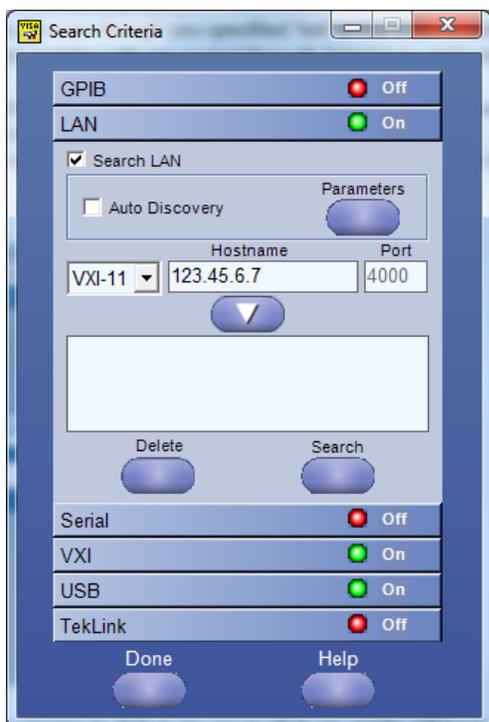
- If other VISA software is installed on the PC, verify that TekVISA is the default VISA. Use the OpenChoice VISA64 Conflict Manger (available in the list of applications under the TekVISA program folder).



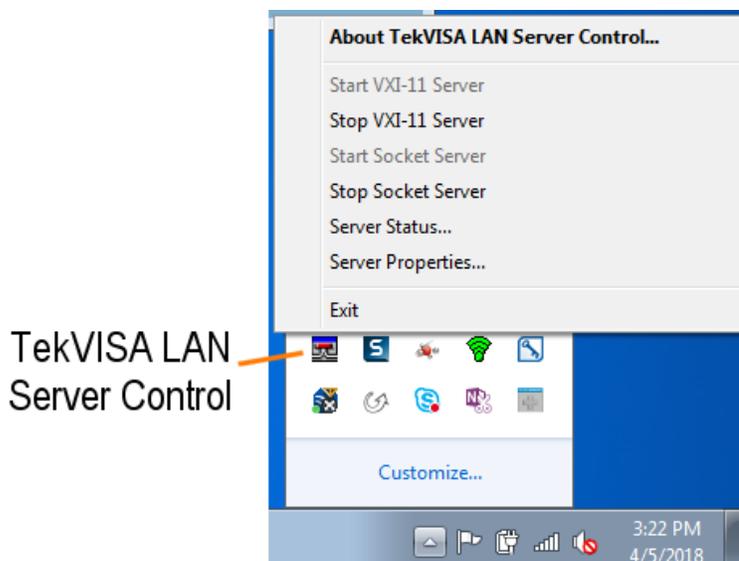
- Make sure that the instrument is detected in the TekVISA Open Choice Instrument Manager (available in the list of applications under the TekVISA program folder).



To detect an instrument for through TekVISA, select the Search Criteria button and then select LAN. Search on the IP address.



- Make sure that the VXI-11 server under TekVISA LAN Server is started on the DSA8300. The server is turned off after the instrument is rebooted, so user needs to start it manually.



Firewall exceptions

- Make sure that TekVISA inbound and outbound traffic are allowed through the Windows Firewall on both the DSA8300 instrument and the PC running 400G-M4 application.
- Make sure that 400G-M4 application is added to the Windows Firewall exception list on the PC.

Open the "Windows Firewall" from "Control Panel", and follow instructions under "Allow an app or feature through Windows Firewall".

Allow programs to communicate through Windows Firewall

To add, change, or remove allowed programs and ports, click Change settings.

[What are the risks of allowing a program to communicate?](#)

[Change settings](#)

i For your security, some settings are managed by your system administrator.

Allowed programs and features:				
Name	Domain	Home/Work (Pri...	Public	Group Policy
<input checked="" type="checkbox"/> 400G-M4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	No

Remote control introduction

The 400G-M4 software application can be controlled programmatically through the programmatic interface. Communication with the application is accomplished using VARIABLE:VALUE remote GPIB commands.

For information on how to operate the sampling oscilloscope and use its application-specific GPIB commands, refer to the programmers guide for your sampling oscilloscope.

Your program should comply with the following guidelines:

- The application startup must complete before sending additional GPIB commands to the application.
- The measurements cycle must complete before you query data.

See also:

[Variable:Value Commands \(see page 37\)](#)

[Program example configure and operate the 400G-M4 \(see page 51\)](#)

Handshaking protocol

The application handles GPIB communications through its own protocol handshaking.

The requirements for GPIB communications with a controller are as follows:

1. Once the application has started, it writes an "OK" status to the application handshake variable. This tells the controller that it may now write a valid commands into the "tdecq" variable.
2. The GPIB controller polls the handshake variable (VARIABLE:VALUE? "tdecq") until it detects the OK status.
3. The GPIB controller writes a command string into the application handshake variable. For example, sending the command VARIABLE:VALUE "tdecq", "single" writes the string "single" into the variable "tdecq".
4. The application GPIB function polls the handshake variable, reads the command string and interprets it as a command. If the command is not understood, it writes an ERROR handshake value to the variable.
5. A good command is parsed and executed. On successful execution, the application writes an OK to the handshake variable. When the GPIB controller reads the OK status, it may send a new command string.

Setting up the PI environment

To help users get started with using GPIB commands to operate the 400G-M4 Analysis application, examples of automatic testing script written in Perl and Python 3.5 is included. These are examples only. Users are free to use any programming language of their choice as long as the handshaking protocol is followed.

Start Client with Perl

The prerequisites are:

- Matlab runtime needs to be installed and in the user's path
- Perl v5.24.3 or later needs to be installed and in the user's path

When the 400G-M4 is running, you can start the client with the following Perl script.

```
perl TDECQ_PI_Client.pl
```

When the 400G-M4 is not running, the script needs to start the 400G-M4 as well as set up the environment by using:

```
perl TDECQ_PI_Client.pl -startApp
```

NOTE. You may need to use the *-filePath* “<pathToTDECQexe>” to tell the client where to connect to the currently running Applications' process.

Start Client with Python

The prerequisites are:

- Matlab runtime needs to be installed and in the user's path
- Python 3.5 or above needs to be installed and in the user's path

When the 400G-M4 is running, you can start the client with the following Python script.

```
python TDECQ_PI_Client.py
```

When the 400G-M4 is not running, the script needs to start the 400G-M4, as well as set up the environment by using:

NOTE. You may need to use the *-filePath* “<pathToTDECQexe>” to tell the client where to connect to the currently running Applications' process.

Running Multiple Instances

Use the following to start the main UI program with assigned instance id of <instanceId>. The default <instanceId> is 1. See the [Example Main Program Execution \(see page 52\)](#) for more details and examples.

```
400G-M4.exe <instanceId>
```

Running Using TCP/IP Address

For an IP connect, use the following command

```
perl TDECQ_PI_Client.pl -cmdfilepath=\\ip address\c$\Program Files\TekApplications\400G-M4
```

This starts 400G-M4.exe at C:\Program Files\TekApplications\400G-M4\400G-M4.exe. The TCPIP requires that the C: drive on the target running the 400G-M4 application is publicly accessible. In the example, it shares as "c\$".

Syntax

The VARIABLE:VALUE command accepts string arguments for a control or data variable and a value to which to set the argument.

To set a variable to a value, use the syntax:

```
VARIABLE:VALUE "tdecq", "<variableName>:<variableValue>"
```

For example, the following commands sets the BWE mode to ON.

```
VARIABLE:VALUE "tdecq", "bwe:on"
```

To query the value in a variable:

```
VARIABLE:VALUE? "tdecq", "<variableName>"
```

For example, the following command queries the state of the BWE mode.

```
VARIABLE:VALUE? "tdecq", "bwe"
```

NOTE. The arguments <variableName> and <variableValue> are required in the order indicated, no spaces, and use of proper capitalization.

Your program will not operate correctly if you do not follow these requirements.

See also:

[Variable:Value Command Arguments and Queries \(see page 38\)](#)

[Program example configure and operate 400GM4 \(see page 51\)](#)

Variable name arguments and queries

Table 9: Control commands and queries

Commands	Description
:variable:value "tdecq","activate:<NR1>"	Command only. Starts the 400G-M4 application. Required before any further processing or setup commands can be executed. Use the command :variable:value? "tdecq","instanceId" to return the instance ID.
Syntax	:variable:value "tdecq","activate:<NR1>" <NR1> = instance ID.
Example	:variable:value "tdecq","activate" starts the 400G-M4 application.
:variable:value? "tdecq","instanceId"	Query only. Returns the instance ID.
Syntax	:variable:value? "tdecq","instanceId"
Example	:variable:value? "tdecq","instanceId" returns the ID.
Return Example	Returns: \$response = '1'
:variable:value "tdecq","connect:<ip_address>"	The set form provides the IP address or the computer name of the oscilloscope to connect to the application. The query form returns the IP address of the connected oscilloscope if it exists. Make sure TekVISA is turned on and the instrument can be found via the TekVISA before setting the IP address. For help with connection issues, refer to Oscilloscope connection tips (see page 31) .
Syntax	:variable:value "tdecq","connect:<ip_address>"
Example	:variable:value "tdecq","connect:134.62.9.4" connects to the instrument with the given IP address. :variable:value? "tdecq","connect" returns the IP address of a connected instrument.
Return Example	Returns: \$response = '134.62.9.4'
:variable:value "tdecq","disconnect"	Command only. Disconnects the application from the oscilloscope in online mode (returns to offline mode).
Syntax	:variable:value "tdecq","disconnect"
Example	:variable:value "tdecq","disconnect" disconnects the application from the connected oscilloscope.

Table 9: Control commands and queries (cont.)

Commands	Description
<code>:variable:value? "tdecq","online"</code>	Query only. Checks if the application is connected to an oscilloscope and has a valid license.
Syntax	<code>:variable:value? "tdecq","online"</code>
Example	<code>:variable:value? "tdecq","online"</code> verifies an oscilloscope connection and a valid license.
Return Example	Returns: \$response = '1' indicates connected. Returns: \$response = '0' indicates no connection.
<code>:variable:value "tdecq","autosync"</code>	Command only. Loads the module information from the connected oscilloscope (when in online mode). Use this command to re-sync module information if any changes have been made to the modules after the initial connection to the oscilloscope. See queries for "various module data fields" to retrieve module information after sending this command.
Syntax	<code>:variable:value "tdecq","autosync"</code>
Example	<code>:variable:value "tdecq","autosync"</code> synchronize the module information with the application.
<code>:variable:value "tdecq","analyze"</code>	Command only. Starts the analysis measurements on the currently loaded data. This is equivalent to pressing the "Run" button. This is required before querying any measurement results. NOTE. After analysis is complete, results of all active channels are also available in <code>%TEMP%\TDECQ_ResultsFile.txt</code> . For TDECQ instances other than the default, results are written to <code>%TEMP%\TDECQ_ResultsFile<instanceId>.txt</code> .
Syntax	<code>:variable:value "tdecq","analyze"</code>
Example	<code>:variable:value "tdecq","analyze"</code> starts the analysis process.
<code>:variable:value "tdecq","clear"</code>	Command only. Clears all measurement results.
Syntax	<code>:variable:value "tdecq","clear"</code>
Example	<code>:variable:value "tdecq","clear"</code> clears all result measurements.
<code>:variable:value? "tdecq","status"</code>	Query only. Returns the contents of the status line.
Syntax	<code>:variable:value? "tdecq","status"</code>

Table 9: Control commands and queries (cont.)

Commands	Description
Example	<code>:variable:value? "tdecq", "status"</code> returns the current string from the status line.
Return Example	Returns: \$response = 'No waveform File'
:variable:value "tdecq","exit"	Command only. Exits the 400G-M4 application. If and analysis is in process, the program exits after the analysis is complete.
Syntax	<code>:variable:value "tdecq","exit"</code>
Example	<code>:variable:value "tdecq", "exit"</code> exits the 400G-M4 application.
:variable:value "tdecq","save-waveform:<filename>"	Command only. Saves the acquired live channel waveform into files for ch1 to ch 4. The files are saved in the specified directory.
Syntax	<code>:variable:value "tdecq","savewaveform:<filename>"</code> <filename> must include the filepath and filename.
Example	<code>:variable:value "tdecq", "savewaveform:c:\test\waveform"</code> saves the file named "waveform" to the specified directory.
:variable:value "tdecq","savesetup:<setup-name>"	Command only. Saves the current settings to a setup file. The setup file is saved in the specified directory. Setup files use the file extension .gm4.
Syntax	<code>:variable:value "tdecq","savesetup:<setupname>"</code> <setupname> must include the filepath and filename.
Example	<code>:variable:value "tdecq", "savesetup:C:\Users\Public\Tektronix\TekApplications\400G-M4\setup1.gm4"</code> saves the setup file named "setup1" to the specified directory.
:variable:value "tdecq","recallsetup:<setupname>"	Command only. Recalls the named setup file from the specified directory. Setup files use the file extension .gm4.
Syntax	<code>:variable:value "tdecq","recallsetup:<setupname>"</code> <setupname> must include the filepath and filename.
Example	<code>:variable:value "tdecq", "recallsetup:C:\Users\Public\Tektronix\TekApplications\400G-M4\setup1.gm4"</code> recalls the setup file named "setup1" from the specified directory.

Table 10: Horizontal settings commands and queries

Commands	Description
:variable: value "tdecq", "patternLength: <Integer>"	Sets or returns the horizontal pattern length value.
Syntax	:variable: value "tdecq", "patternLength: <Integer>" <Integer> = value from 21 to 10E3
Example	:variable: value "tdecq", "patternLength: 8191" sets the pattern length to 8191. :variable: value? "tdecq", "patternLength" returns the current pattern length.
Return Example	Returns: \$response = '8191'
:variable: value "tdecq", "baudRate: <Integer>"	Sets or returns the horizontal baud rate value (GHz) in symbols/second.
Syntax	:variable: value "tdecq", "baudRate: <Integer>" <Integer> = value from 10 to 100.
Example	:variable: value "tdecq", "baudRate: 20" sets the baud rate to 20 G symbols/second. :variable: value? "tdecq", "baudRate" returns the current symbols/second value.
Return Example	Returns: \$response = '20000000000'
:variable: value "tdecq", "samplesPerBit: <Integer>"	Sets or returns the horizontal waveform samples per bit of the signal.
Syntax	:variable: value "tdecq", "samplesPerBit: <Integer>" <Integer> = value from 5 to 100.
Example	:variable: value "tdecq", "samplesPerBit: 10" sets the waveform samples per bit to 10. :variable: value? "tdecq", "samplesPerBit" returns the waveform samples per bit value.
Return Example	Returns: \$response = '10'

Table 11: Measurement select commands and queries

Commands	Description
:variable: value "tdecq", "tdecqMeas: 1 0"	Sets or returns the state (enabled or disabled) of the TDECQ measurement. This is a global setting for both offline and online modes.
Syntax	:variable: value "tdecq", "tdecqMeas: 1 0" 1 enables the TDECQ measurement. 0 disables the TDECQ measurement.

Table 11: Measurement select commands and queries (cont.)

Commands	Description
Example	<code>:variable:value "tdecq", "tdecqMeas:1"</code> enables the TDECQ measurement.
Return Example	Returns: \$response = '1'
<code>:variable:value "tdecq", "levelMeas:1 0"</code>	Sets or returns the state (enabled or disabled) of the Symbol Level measurement. This is a global setting for both offline and online modes.
Syntax	<code>:variable:value "tdecq", "levelMeas:1 0"</code> 1 enables the Symbol Levels measurements. 0 disables the Symbol Levels measurements.
Example	<code>:variable:value "tdecq", "levelMeas:1"</code> enables the Symbol Level measurements.
Return Example	Returns: \$response = '1'
<code>:variable:value "tdecq", "eyeMeas:1 0"</code>	Sets or returns the state (enabled or disabled) of the Eye measurements. This is a global setting for both offline and online modes.
Syntax	<code>:variable:value "tdecq", "eyeMeas:1 0"</code> 1 enables the Eye measurements. 0 disables the Eye measurements.
Example	<code>:variable:value "tdecq", "eyeMeas:1"</code> enables the Eye measurements.
Return Example	Returns: \$response = '1'

Table 12: Configuration commands and queries

Commands	Description
<code>:variable:value "tdecq", "config:ExtendedSearch:0 1"</code>	Sets or returns the state of the Extended Search (enabled or disabled). When enabled, the whole waveform is searched.
Syntax	<code>:variable:value "tdecq", "config:ExtendedSearch:0 1"</code> 1 enables the Extended search capability. 0 disables the Extended search capability.
Example	<code>:variable:value "tdecq", "config:ExtendedSearch:1"</code> enables the Extended search capability. <code>:variable:value? "tdecq", "config:ExtendedSearch"</code> returns the state.
Return Example	Returns: \$response = '1'

Table 12: Configuration commands and queries (cont.)

Commands	Description
:variable:value "tdecq","config:VerticalThresholdFlex:0 1"	Sets or returns the state of the Vertical Threshold Adjust (enabled or disabled) for the TDECQ measurement configuration.
Syntax	:variable:value "tdecq","config:VerticalThresholdFlex:0 1" 1 enables the Vertical Threshold Adjust capability. 0 disables the Vertical Threshold Adjust capability.
Example	:variable:value "tdecq","config:VerticalThresholdFlex:1" enables the Vertical Threshold Adjust capability. :variable:value? "tdecq","config:VerticalThresholdFlex" returns the state.
Return Example	Returns: \$response = '1'
:variable:value "tdecq","config:AutoSetTapValues:0 1"	Sets or returns the state of the Auto Tap Values (enabled or disabled) for the TDECQ measurement configuration.
Syntax	:variable:value "tdecq","config:AutoSetTapValues:0 1" 1 enables the Auto Tap Values capability. 0 disables the Auto Tap Values capability.
Example	:variable:value "tdecq","config:AutoSetTapValues:1" enables the Auto Tap Values capability. :variable:value? "tdecq","config:AutoSetTapValues" returns the Auto Tap Values state.
Return Example	Returns: \$response = '1'
:variable:value "tdecq","config:AutoImportNoise:0 1"	Sets or returns the state of the Auto Import Noise (enabled or disabled) for the TDECQ measurement configuration.
Syntax	:variable:value "tdecq","config:AutoImportNoise:0 1" 1 enables the Auto Import Noise capability. 0 disables the Auto Import Noise capability.
Example	:variable:value "tdecq","config:AutoImportNoise:1" enables the Auto Import Noise capability. :variable:value? "tdecq","config:AutoImportNoise" returns the Auto Import Noise state.
Return Example	Returns: \$response = '1'
:variable:value "tdecq","config:NumTaps:<Integer>"	Sets or returns the FFE tap length of the equalizer.
Syntax	:variable:value "tdecq","config:NumTaps:<Integer>" <Integer> = value from 1 to 99 and must be an odd number.

Table 12: Configuration commands and queries (cont.)

Commands	Description
Example	<code>:var:value "tdecq", "config:NumTaps:31"</code> sets the FFE tap length to 31. <code>:var:value? "tdecq", "config:NumTaps"</code> returns the FFE tap length.
Return Example	Returns: <code>\$response = '31'</code>
:variable:value "tdecq", "config:SamplesPerUI:<Integer>"	Sets or returns the samples per FFE tap for the equalizer.
Syntax	<code>:variable:value "tdecq", "config:SamplesPerUI:<Integer>"</code> <code><Integer> = value from 1 to 10</code>
Example	<code>:variable:value "tdecq", "config:SamplesPerUI:3"</code> sets the samples per FFE tap to 3. <code>:variable:value? "tdecq", "config:SamplesPerUI"</code> returns the samples per FFE tap.
Return Example	Returns: <code>\$response = '3'</code>
:variable:value "tdecq", "config:MaxPreCursor:<Integer>"	Sets or returns the maximum pre cursor for the FFE equalizer.
Syntax	<code>:variable:value "tdecq", "config:MaxPreCursor:<Integer>"</code> <code><Integer> = value from 0 to the length of the FFE taps minus 1.</code>
Example	<code>:variable:value "tdecq", "config:MaxPreCursor:3"</code> sets the maximum pre cursor to 3. <code>:variable:value? "tdecq", "config:MaxPreCursor"</code> returns the maximum pre cursor.
Return Example	Returns: <code>\$response = '3'</code>
:variable:value "tdecq", "ffetapvalue[:(offline ch<x>)]:<float>"	Sets or returns the FFE tap values for the offline mode or the specified channel. The input needs to be float, and the taps must match the FFE tap length. See <code>:variable:value "tdecq", "config: NumTaps:<Integer>"</code>
Syntax	<code>:variable:value "tdecq", "ffetapvalue[:(offline ch<x>)]:<float>"</code> Offline or <code>ch<x></code> are optional. If not specified, the currently selected mode or channel is used. <code><float> = string of tap values.</code>
Example	<code>:variable:value "tdecq", "ffetapvalue:offline:0.1 0.2 0.3 0.4 0.5"</code> sets the tap value for 5 FFE taps. <code>:variable:value? "tdecq", "ffetapvalue:offline"</code> returns the current tap values.
Return Example	Returns: <code>\$response = '0.1 0.2 0.3 0.4 0.5 '</code>

Table 13: Offline and Online mode commands and queries

Commands	Description
<code>:variable:value "tdecq","channel:(offline ch<x>)"</code>	The set form selects either the offline mode or one of the channels in the online mode. The query form returns either the offline mode or a channel (online mode).
Syntax	<code>:variable:value "tdecq","channel:(offline ch<x>)"</code> <x> = 1 to 4.
Examples	<code>VARIABLE:VALUE "tdecq","channel:offline"</code> selects the offline mode. <code>VARIABLE:VALUE "tdecq","channel:ch2"</code> selects the channel 2 online mode. <code>VARIABLE:VALUE? "tdecq","channel1"</code> returns the current mode or channel.
Return Example	Returns: <code>\$response = 'offline'</code>
<code>:variable:value "tdecq","select[:(offline ch<x>)]:ON OFF 0 1>"</code>	Sets or returns the Enabled stated (enables or disabled) of the specified channel or offline mode.
Syntax	<code>:variable:value "tdecq","select[:(offline ch<x>)]:ON OFF 0 1>"</code> <x> value ranges from 1 to 4. On or 1 enables the offline mode or channel. Off or 0 disables the offline mode or channel.
Example	<code>:variable:value "tdecq","select:ch1:1"</code> enables channel 1. <code>:variable:value? "tdecq","select:ch1"</code> returns the enabled state for channel 1.
Return Example	Returns: <code>\$response = '1'</code>
<code>:variable:value "tdecq","noise[:(offline ch<x>)]:<float>"</code>	Sets or returns the scope noise value (in W) for the specified channel or offline mode.
Syntax	<code>:variable:value "tdecq","noise[:(offline ch<x>)]:<float>"</code> Offline or ch<x> are optional. If not specified, the currently selected mode or channel is used. <x> = 1 to 4. <float> = 0 to 1.
Example	<code>:variable:value "tdecq","noise:ch1:5e-6"</code> sets the channel 1 scope noise to 5 μ W. <code>:variable:value? "tdecq","noise:ch1"</code> returns the scope noise value for channel 1.
Return Example	Returns: <code>\$response = '5e-6'</code>
<code>:variable:value "tdecq","jnb_filename:<filename>"</code>	The set form loads the specified JNB filename for offline measurement analysis. The query form returns the currently loaded filename and filepath.

Table 13: Offline and Online mode commands and queries (cont.)

Commands	Description
Syntax	:variable: value "tdecq", "jnb_filename: <filename> <filename> must include the filepath and filename.
Example	:var: value "tdecq", "jnb_filename: c:\test\jnbTest.mat" loads the file named jnbTest.mat. :var: value? "tdecq", "jnb_filename" returns the currently loaded filename and filepath.
Return Example	Returns: \$response = 'c:\test\jnbTest.mat'
:variable: value? "tdecq", "module: [(offline ch<x>)]"	Query only. Returns the module data fields of the module located in the specified channel of the DSA8300 oscilloscope when in online mode.
Syntax	:variable: value? "tdecq", "module: [(offline ch<x>)]" Offline or ch<x> are optional. If not specified, the currently selected mode or channel is used.
Example	:variable: value? "tdecq", "module: ch4" returns the information of the module that is installed in channel 4 of the DSA8300.
Return Example	Returns: \$response = '80C21'
:variable: value? "tdecq", "serialnum: [(offline ch<x>)]"	Query only. Returns the serial number of the module located in the specified channel of the DSA8300 oscilloscope when in online mode.
Syntax	:variable: value? "tdecq", "serialnum: [(offline ch<x>)]" Offline or ch<x> are optional. If not specified, the currently selected mode or channel is used.
Example	:variable: value? "tdecq", "serialnum: ch4" returns the serial number of the module installed in channel 4 of the DSA8300.
Return Example	Returns: \$response = 'B010564'
:variable: value? "tdecq", "modfilter: [(offline ch<x>)]"	Query only. Returns the filter selection of the module located in the specified channel of the DSA8300 oscilloscope when in online mode.
Syntax	:variable: value? "tdecq", "modfilter: [(offline ch<x>)]" Offline or ch<x> are optional. If not specified, the currently selected mode or channel is used.
Example	:variable: value? "tdecq", "modfilter: ch4" returns the filter selection of the module installed in channel 4 of the DSA8300.
Return Example	Returns: \$response = 'none'

Table 13: Offline and Online mode commands and queries (cont.)

Commands	Description
<code>:variable:value?</code> <code>"tdecq","modbw[:(offline ch<x>)]"</code>	Query only. Returns the bandwidth selection of the module located in the specified channel of the DSA8300 oscilloscope when in online mode.
Syntax	<code>:variable:value? "tdecq","modbw[:(offline ch<x>)]"</code> Offline or ch<x> are optional. If not specified, the currently selected mode or channel is used.
Example	<code>:variable:value? "tdecq","modbw:ch4"</code> returns the bandwidth selection of the module installed in channel 4 of the DSA8300.
Return Example	Returns: <code>\$response = '49.999E+9'</code>

Table 14: BW Enhancement commands and queries

Commands	Description
<code>:variable:value "tdecq","bwe[:(offline ch<x>)]:ON OFF"</code>	Sets or returns the BWE state (enabled or disabled) to the offline mode or the specified channel. When BWE is enabled, the selection of the "bwe:select" value can be made.
Syntax	<code>variable:value "tdecq","bwe[:(offline ch<x>)]:ON OFF"</code> Offline or ch<x> are optional. If not specified, the currently selected mode or channel is used. ON enables the BWE mode. OFF disables the BWE mode.
Example	<code>VARIABLE:VALUE "tdecq","bwe:ch3:ON"</code> enables BWE on channel 3. <code>VARIABLE:VALUE? "tdecq","bwe:ch3"</code> returns the BWE state for channel 3.
Return Example	Returns: <code>\$response = 'ON'</code> Use to <code>:variable:value "tdecq","bwe[:(offline ch<x>)]:ON OFF"</code>
<code>:variable:value "tdecq","bwe[:(offline ch<x>)]:select:<NR1>"</code>	Sets or returns the BWE filter selection type for the offline mode or the specified channel. NOTE. BWE must be enabled before selecting a filter. See the command <code>:variable:value "tdecq","bwe[:(offline ch<x>)]:ON OFF"</code>
Syntax	<code>:variable:value "tdecq","bwe[:(offline ch<x>)]:select:<NR1>"</code> Offline or ch<x> are optional. If not specified, the currently selected mode or channel is used. <NR1> = an integer between 1 and 4. 1 = HW only 2 = 0.5 X Baud rate 3 = 0.75 X Baud rate 4 = User selected

Table 14: BW Enhancement commands and queries (cont.)

Commands	Description
Example	<pre>:variable:value "tdecq", "bwe:ch3:select:1" sets the BWE filter for channel 3 to HW only.</pre> <pre>:variable:value? "tdecq", "bwe:ch3:select" returns the BWE filter selection for channel 3.</pre>
Return Example	Returns: \$response = '1'
:variable:value "tdecq", "bwe:[offline ch<x>]:btbw:<NR1>"	<p>Sets or returns the "BT 4th Order BW" user supplied filter value (in Hz) for the offline mode or the specified channel.</p> <p>NOTE. BWE must be enabled and the filter type set to User.</p> <p>See the commands:</p> <pre>:variable:value "tdecq", "bwe:[offline ch<x>]:ON OFF"</pre> <pre>:variable:value "tdecq", "bwe:[offline ch<x>]:select:<NR1>"</pre>
Syntax	<pre>:variable:value "tdecq", "bwe:[offline ch<x>]:btbw:<NR1>"</pre> <p>Offline or ch<x> are optional. If not specified, the currently selected mode or channel is used.</p> <p><NR1> = value from 13.2813e9 to 100e9 and must be larger than ½ baud rate.</p>
Example	<pre>:variable:value "tdecq", "bwe:ch3:btbw:10e9" sets the Bessel-Thomson 4th Order bandwidth to 10 GHz.</pre> <pre>:variable:value? "tdecq", "bwe:ch3:btbw" returns the Bessel-Thomson 4th Order bandwidth setting.</pre>
Return Example	Returns: \$response = '10000000000'
:variable:value "tdecq", "sparam_filename:[offline ch<x>]:<filename>"	<p>The set form loads the specified S-parameter filename and filepath for the offline mode or the specified channel for measurement analysis.</p> <p>The query form returns the filename and filepath of the S-parameter file.</p>
Syntax	<pre>:variable:value "tdecq", "sparam_filename:[offline ch<x>]:<filename>"</pre> <p>Offline or ch<x> are optional. If not specified, the currently selected mode or channel is used.</p> <p><filename> must include the filepath and filename.</p>
Example	<pre>:variable:value "tdecq", "sparam_filename:ch1:c:\test\ch1sparameter.s1p" loads the ch1sparameter.s1p file.</pre> <pre>:variable:value? "tdecq", "sparam_filename:ch1" returns the S-parameter filename and filepath for channel 1.</pre>
Return Example	Returns: \$response = 'c:\test\ch1sparameter.s1p'

Table 15: Measurement result queries

Commands	Description
:variable:value? "tdecq","TDECQ"	Query only. Returns the TDECQ measurement result for the active mode or channel.
Syntax	:variable:value? "tdecq","TDECQ"
Example	:variable:value? "tdecq","TDECQ" returns the TDECQ result for the active mode or channel.
Return Example	Returns: \$response = 'TDECQ:3.0726'
:variable:value? "tdecq","OMA"	Query only. Returns the OMA measurement result for the active mode or channel.
Syntax	:variable:value? "tdecq","OMA"
Example	:variable:value? "tdecq","OMA" returns the OMA measurement result for the active mode or channel.
Return Example	Returns: \$response = 'OMA:0.0013304'
:variable:value? "tdecq","ER"	Query only. Returns the ER measurement result for the active mode or channel.
Syntax	:variable:value? "tdecq","ER"
Example	:variable:value? "tdecq","ER" returns the ER measurement result for the active mode or channel.
Return Example	Returns: \$response = 'ER:14.891'
:variable:value? "tdecq","AOP"	Query only. Returns the AOP measurement result for the active mode or channel.
Syntax	:variable:value? "tdecq","AOP"
Example	:variable:value? "tdecq","AOP" returns the AOP measurement result for the active mode or channel.
Return Example	Returns: \$response = 'AOP:0.00068321'
:variable:value? "tdecq","RLM"	Query only. Returns the RLM measurement result for the active mode or channel.
Syntax	:variable:value? "tdecq","RLM"
Example	:variable:value? "tdecq","RLM" returns the RLM measurement result for the active mode or channel.
Return Example	Returns: \$response = 'RLM:0.94991'

Table 15: Measurement result queries (cont.)

Commands	Description
:variable:value? "tdecq","Deviation"	Query only. Returns the Deviation measurement result for the active mode or channel.
Syntax	:variable:value? "tdecq","Deviation"
Example	:variable:value? "tdecq","Deviation" returns the Deviation measurement result for the active mode or channel.
Return Example	Returns: \$response = 'levelDeviation:0.027654'
:variable:value? "tdecq","Thickness"	Query only. Returns the Thickness measurement result for the active mode or channel.
Syntax	:variable:value? "tdecq","Thickness"
Example	:variable:value? "tdecq","Thickness" returns the Thickness measurement result for the active mode or channel.
Return Example	Returns: \$response = 'levelThickness:0.070283'
:variable:value? "tdecq","AllMeas[:(offline ch<x> active)]"	Query only. Returns all available measurements from the most recent analysis from the offline mode, specified channel, or simply the currently active mode/channel.
Syntax	:variable:value? "tdecq","AllMeas[:(offline ch<x> active)]" offline returns all measurements from the offline analysis run. ch<x> returns all measurements from the specified channel source. active returns all measurements across all active online sources. Each "<measTypeX>" field will be prefixed with the specific source's name. e.g., "ch1:TDECQ","7.3698+13.6438i","ch1:OMA","-0.0041289", Offline, ch<x>, or active are optional. If not specified, the currently selected mode or channel is used.

Table 15: Measurement result queries (cont.)

Commands	Description
Example	<p><code>:variable:value? "tdecq", "AllMeas:active"</code> returns all measurements from the all enabled channels.</p> <p>The response format is a list of "<measType1>","<measValue1>","<measType2>","<measValue2>"... This may provide more data than the individual measurement queries, as well as space-delimited arrays of values for particular measurement types.</p> <p>A sample response might look like the following:</p> <pre>"TDECQ", "3.5884", "OMA", "0.0013259", "ER", "14.9446", "AOP", "0.0006836", "v:mean", "6.8129e-07 0.00044641 0.00091906 0.0013689", "v:std", "3.8602e-05 4.7552e-05 4.0125e-05 4.9945e-05", "v:pk2pk", "0.00027391 0.00036793 0.00028668 0.00030382", "RLM", "0.95458", "levelThick- ness", "0.064398", "levelDeviation", "0.024212", "eye:Up- per:Thresh", "0.0011315", "eye:Upper:Offset", "1.0756e- 13", "eye:Upper:H_eye", "3.6034e-12", "eye:Up- per:V_eye", "0.00014137", "eye:Middle:Thresh", "0.000 70909", "eye:Middle:Offset", "3.0882e-13", "eye:Mid- dle:H_eye", "4.1471e-12", "eye:Middle:V_eye", "0.000127 24", "eye:Lower:Thresh", "0.00021014", "eye:Lower:Off- set", "-2.2588e-13", "eye:Lower:H_eye", "3.6894e- 12", "eye:Lower:V_eye", "0.0001365"</pre>
Return Example	<p>Returns:</p> <pre>"TDECQ", "3.5884", "OMA", "0.0013259", "ER", "14.9446", "AOP", "0.0006836", "V- :mean", "6.8129e-07 0.00044641 0.00091906 0.0013689", "V:std", "3.8602e-05 4.7552e-05 4.0125e-05 4.9945e-05", "V:pk2pk", "0.00027391 0.00036793 0.00028668 0.00030382", "RLM", "0.95458", "levelThickness", "0.064398", "lev- elDeviation", "0.024212", "eye:Upper:Thresh", "0.0011315", "eye:Up- per:Offset", "1.0756e-13", "eye:Upper:H_eye", "3.6034e-12", "eye:Up- per:V_eye", "0.00014137", "eye:Middle:Thresh", "0.00070909", "eye:Mid- dle:Offset", "3.0882e-13", "eye:Middle:H_eye", "4.1471e-12", "eye:Mid- dle:V_eye", "0.00012724", "eye:Lower:Thresh", "0.00021014", "eye:Lower:Off- set", "-2.2588e-13", "eye:Lower:H_eye", "3.6894e-12", "eye:Lower:V_eye", "0.0001365"</pre>

Perl program example: configure and operate 400G-M4

With available offline analysis, 400G-M4 offers a programmatic interface similar to that of TekScope, JNB, or other on-scope applications. A client needs to know how to interface and synchronize with the application. The Perl script below gives an example of such a client (similar clients could be implemented in Python, .NET, etc.). A file interface is used to communicate between 400G-M4 and the client.

NOTE. *The examples presume that you are familiar with designing GPIB programs.*

All words within quotes are case sensitive and must be entered exactly as shown, including spaces and quotes.

Prerequisites

- Matlab runtime needs to be installed and in the user's path
- Perl needs to be installed and in the user's path
- The client expects to be able to write "CommandFile.txt" and "TDECQ_LogFile.txt" to the current working directory (C:\Users\\AppData\Local\Temp) for communication and logging. For multiple instances, the files "CommandFile<instanceId>.txt" and "TDECQ_LogFile<instanceId>.txt" are used. See below for examples.

The measurement results are captured in TDECQ_ResultsFile.txt at the same directory.

- The client by default enters an 'interactive' mode, reading from STDIN (which times out and exits after 5 minutes). At this point you can send any of the specified commands and queries.

Example Main Program Execution

```
400G-M4.exe <instanceId>
```

Starts the main UI program with assigned instance id of <instanceId>. The default instanceId is 1. For each instance other than 1, a separate client may connect to each unique instance. Each separate client for instanceIds other than 1 can connect to that particular <instanceId> via the corresponding "CommandFile<instanceId>.txt". e.g., upon execution of the following:

```
400G-M4.exe 1
```

```
400G-M4.exe 2
```

```
400G-M4.exe 3
```

, client 1 may connect to instance 1 via %TEMP%\CommandFile.txt, and client 2 may connect to instance 2 via %TEMP%\CommandFile2.txt, and client 3 may connect to instance 3 via %TEMP%\CommandFile3.txt.

NOTE. For the Example Client Execution below, you would need to use the "--appInstance=<instanceId>" to tell the client to connect to the specific <instanceId> UI execution. See the example using the "--appInstance" flag below.

Example Client Execution

```
perl TDECQ_PI_Client.pl
```

Starts the client in interactive mode, which reads the commands/queries from stdin. This is ideal if the UI is already running.

NOTE. You may need to use the -filePath "<pathToTDECQexe>" to tell the client where to connect to the currently running Application's process).

```
perl TDECQ_PI_Client.pl -startApp
```

Starts the client and the main application (ideal if the UI is not running yet) which in turn reads the set of commands/queries from stdin.

Once stdin is ready to read commands, an example set to send is:

```
:variable:value "tdecq","activate"
:variable:value "tdecq","select:offline"
:variable:value "tdecq","jnb_filename:1.mat"
:variable:value "tdecq","analyze"
:variable:value? "tdecq","TDECQ"
:variable:value? "tdecq","OMA"
:variable:value? "tdecq","ER"
:variable:value "tdecq","exit"

perl TDECQ_PI_Client.pl -startApp < TDECQ_PI_Client_Example_CmdSet.txt
```

Starts the client which in turn reads the set of commands/queries from the .txt file.

Example output from similar commands is:

```
...
> :variable:value? "tdecq","tdecq" $response = 'TDECQ:5.8421'

> :variable:value "tdecq","exit" Beginning shutdown...

...shutdown complete.
```

```
perl TDECQ_PI_Client.pl -startApp -appInstance=2
```

Starts the client and the main application with the instance ID 2, which in turn reads the set of commands/queries from stdin. A query of :var:val? "tdecq","instanceid" at this point should return the value of "2".

NOTE. The log file at “%TEMP%\TDECQ_LogFile.txt” will contain trace and debug information from any client run, including the TDECQ MatLab processing output after exiting the client.

Many other behaviors may be configured by the client, such as file paths and run modes. See the “\$configParams” section for details.

Python example

A simple Python programming example file is provided.

The example establishes a connection to the 400G-M4 application and provides a simple interface to send commands to the application.

NOTE. *Python is required to be installed on the PC.*

1. Open a DOS command window.
2. Enter the Python example file name “TDECQ_PIClient.py”.
3. Wait for the program to run and for the UI window to open to send commands.

The actual Python example file is located at C:\Users\Public\Tektronix\TekApplications\400G-M4\PIExamples.

Feedback

Tektronix values your feedback on our products. To help us serve you better, please send us suggestions, ideas, or other comments you may have about your application or oscilloscope. Send your feedback to techsupport@tektronix.

Please be as specific as possible and include the following information:

General information

- Oscilloscope model number, firmware version number, and hardware/software options, if any.
- Module and probe configuration. Include model numbers and the channel/slot location.
- Serial data standard.
- Signaling rate.
- Pattern type and length.
- Your name, company, mailing address, phone number, FAX number.

NOTE. *Please indicate if you would like Tektronix to contact you regarding your suggestion or comments.*

Application-specific information

- 400G-M4 application version number.
- Description of the problem such that technical support can duplicate the problem.
- Configuration of the 400G-M4 application. Use screen shots to capture application parameters.
- If possible, save the acquired oscilloscope waveform file. (**File > Save Waveform**).
- If possible, save the DSA8300 settings to a .stp file.

Once you have gathered this information, contact technical support by phone or through email. If using email, be sure to enter “400G-M4 Problem” in the subject line, and attach supporting files.

Index

Symbols and Numbers

400G-M4 application
description, 1

A

Application
software version, 9
Arguments and queries, 38
Auto import scope noise, 14
Autoset tap values, 14
AutoSync to Selected Sources, 8

B

Bandwidth enhancement
BWE, 17
Baud Rate, 11
BT 4th Order BW, 17
BWE
bandwidth enhancement, 17

C

Clear button, 19
Command Syntax, 37
Commands, 38
Computer requirements, 2
Configuration
TDECQ, 13
Connect to scope, 7

D

Display elements, 5
DSA8300 requirements, 3

E

Example client execution, 52
Example main program
execution, 52
Extended search, 14
Eye, 13
Eye results, 28

F

Features, 1
Feedback, 55
FFE taps, 14
FFE Taps/UI, 14
File menu, 7

G

GPIO, 35
GPIO Commands, 35
GPIO Guidelines, 35
GPIO Program, 35
GPIO Program Examples, 51
GPIO Programs, 51

H

Handshaking protocol, 35
Horizontal settings, 11

I

IP address, 8

M

Matlab, 9
Matlab runtime, 2
Max pre-cursors, 14
Measurement configuration
TDECQ, 13
Measurement selections, 13
Eye, 13
symbol levels, 13
TDECQ, 13
Mode selection
offline, online, 15
Modes
offline, 23
online, 25
Modules
supported, 25

O

Offline mode, 23
Online mode, 25
Operating system, 2
Option key, 3
Oscilloscope connection, 31

P

Pattern Length, 11
Plot menu, 9
Product description, 1
Product Description, 1
Program Examples, 51
Programming
Example client execution, 52
Example main program
execution, 52
Start client with Perl, 36
Start client with Python, 36

Q

Queries and arguments, 38

R

Recall setup, 7
Remote control, 35
Requirements
computer, 2
DSA8300, 3
Results
clear, 19
Results elements, 27
Run button, 21

S

S-parameter, 17
samples per user interval, 11
Save setup, 7
Save waveform, 7
Scope menu, 7
scope noise, 23

- Scope noise, 14
 - application, 3
 - auto import, 26
- Setup files, 7
- Software, 55
- Software requirements
 - Matlab runtime, 2
 - TekVISA, 2
- Software version, 9
- Start analysis, 21
- Start client with Perl, 36
- Start client with Python, 36
- Supported
 - modules, 25

- Symbol Level, 13
- Symbol level measurement, 27
- Syntax, 37

T

- Tap values, 14
- TDECQ, 13
- TDECQ and PAM4 results, 29
- TDECQ configuration, 13
- Techsupport@tektronix.com, 55
- TekVISA, 3
- TekVISA setup, 31
- TekVISA troubleshooting, 31

- Troubleshooting connection, 31

V

- Variable Value, 37
- Vertical threshold adjust, 14

W

- Wfm Samples/UI, 11