Quarch Technology Ltd

Cable Tester

Technical Manual

For use with:

QTL2250 – PCIe 4.0 / SAS 4.0 Cable Tester

Change History

1.0	May 2020	Initial release
1.1	July 2020	Updates after TUV certification
1.2	July 2020	Further updates after TUV certification
1.3	July 2020	Additional commands added
1.4	August 2020	Formatting, adding updated product photos

Contents

About this Manual

Thank you for your purchase of this Quarch Module. This manual is intended as a technical reference, describing the features, operations and control API of the unit.

Additional documents available include Quick start guides, Datasheets and Application notes. These can be located on our website: www.quarch.com

This manual provides the following:

- Safety information
- An overview of the product and its features
- Details of the control interfaces
- Command specification
- Locations of additional help and examples.

Product Safety

This product is intended for experienced technical users in a test lab environment. It is essential that you are familiar with this manual before using the device.

Safety warnings

This product must be connected to the supplied 12V DC, 5A PSU. Do not use any other power supply. Stontronics Part Number: T5797ST

There are no user serviceable parts inside; do **NOT** open the enclosure.

The product contains a high-speed fan for cooling.

Risk Assessment

While these products have no significant hazard cases, they must be used carefully by competent personnel.

In new use cases, a risk assessment should be completed, to ensure the entire test setup is safe to operate and will not cause an unacceptable risk to the operators or other lab users.

This should include careful routing of power and control cables to avoid trip hazards.

Cleaning and Maintenance

The products may be a dusted with a dry cloth. Water and alcohol based (or any other liquid) cleaners should not be used.

All parts of the system should be inspected before and after use.

Damaged components should not be used and cannot be repaired by the user. Please contact Quarch to arrange for repair.

Manufacturer Details

The Torridon System is designed and manufactured by:

Quarch Technology Ltd (registered in Scotland no: SC307569)

Unit 7, Dalfaber Industrial Estate

Aviemore

UK

PH22 1ST

Web: www.quarch.com

Email: enquiries@quarch.com

Phone: +44 1343 508 140

Environmental Conditions

The products are designed for use in dry, dust free, indoor lab conditions, at altitudes below 2000m and ambient temperatures between 10C-40C and 0-80% non-condensing humidity.

Installation

The product is designed for use on a desk/bench location. The provided 12V DC, 5A PSU must be used.

Product Misuse

The product is designed lab testing of single miniSAS HD or External PCIe cables, complying with the relevant industry specifications. Other uses may cause incorrect operation, protection impairment or damage to the product and should be avoided.

In the Box

Each Cable Tester comes with the same basic kit of parts.

1x Cable Tester main unit

1x USB cable

1x 12V DC, 5A PSU with power cable (region specific)

Stontronics T5797ST Supply ONLY

Technical Specifications

Cable Tester products from Quarch are all based on the same design and can be used within the same testing system.

The QTL2250 is for testing SAS 4.0 and PCIe 4.0 cables following the external MiniSAS and PCIe specifications.

QTL2250 – single module layout

Front panel of QTL2250

Rear panel of QTL2250

QTL2250 – connections and indicators

Item	Location	Purpose
Status LED	Front panel	Test result status
Power/Action LED	Front panel	Unit power on and primary action state
Port 1	Front panel	Test port 1 for cable under test
Port 2	Front panel	Test port 2 for cable under test
CLK1	Front panel	Clock output port
CLK2	Front panel	Clock output port
REFCLK	Front panel	Reference clock output port
PSU	Rear panel	12V 5 Amp power supply input
USB	Rear panel	USB port for direct control from a local PC
LAN	Rear panel	100Base-T network port for remote control

Introduction

The Cable Tester drives test signals through a cable to verify its connection layout and high-speed performance.

Lane and sideband mapping verifies the basic electrical layout of the cable. Link training ensures the cable is capable of handling data transfer at the required rate.

With the basic training complete, we can now generate eye diagrams, run longer term BERT tests, and perform frequency sweeps.

The cable tester can also verify EEPROM access and retrieve its contents.

Main features

- Verify the electrical connectivity of a cable
- Test the performance with realistic data patterns at the full link speed
- Compare cables from different manufacturers

LEDs

The Cable Tester provides 2 LEDs on the front panel. For automatic testing, these provide basic details on pass/fail status. For more details, a control application must be run on an attached PC.

Power/Action LED

Color	Meaning
Off	No Power
Green	Power on standard state
Blue (Flashing)	Running cable test process
Blue (Solid)	Test process complete
Red	Unit fault

Status LED

Color	Meaning
Off	No cable/No result
Purple	Cable training in progress
Green	BERT running clean (BERT MODE) Test completed with success state
Red (Flashing)	Bert error seen
Orange (Solid)	Test completed with warnings/uncertain state
Red (Solid)	Test completed with failure state

Controlling the Module

The control interface for Cable Test Modules is via a LAN or USB connection.

All major features are available over both interfaces. Control is either via the Cable Tester application or our simple Python API.

Basic operation can be completed without any attached PC, using the LEDs to verify the pass/fail state for the tests.

The cable you wish to test should be connected in loopback configuration, between the two ports on the tester.

Communication protocol

Commands sent to the module are simple, text-based commands, roughly based on the SCPI specification. For instance, the following command resets the BERT counters for the TX_0 pair

```
BERT:TX_0:RESET
```

Download of recorded and eye data uses a compressed binary format for speed.

Quarch applications and examples handle the sending of commands and processing of recorded data, allowing very simple automation from Python scripts and similar.

LAN

A simple TCP based custom protocol is the default. Simplified HTTP/REST is also available.

USB

USB control is supported. The Quarch USB driver is required on Windows PCs. LibUSB can be used for Linux.

Control options

There are several applications and scripted control options for Cable Testers:

- Torridon Terminal

  This is a simple terminal application that can connect to a module and send manual commands. It is useful for testing, debugging and small amounts of manual control.

  This is a Windows-only application and can be run directly (no installer needed). It can connect to the module over USB or LAN.

- Cable Tester GUI

  This application is designed for fast control of the cable tester, along with access to advanced features and downloading of formal reports

- Python scripts

  The quarchpy python package provides simple control of the module

Default Startup State

On power up or reset, the module enters a default state.

Basic setting	Default value
Auto Test Mode	SAS GEN4
Transmit Pattern	PRBS7

If the unit is factory reset (**`CONFig:DEFault FACTory`**) then the following defaults are set

Basic setting	Default value
Auto Test Enable	ON
Auto Test EEPROM Decode	ON
Auto Test Link Training	ON
Auto Test Cable Mapping	ON
Auto Test Eyescan	ON
Auto Test Eye Height	ON
Auto Test BERT	ON
Auto Test Mode	SAS GEN4
Transmit Pattern	PRBS7

If Enables for individual parts of auto test are changed, they become non-volatile except in the case of a factory reset. In which case, the above defaults are set.

LAN setting	Default value
DHCP	ON
IP address	192.168.1.99
IP mask	255.255.255.0
DNS1,2	0.0.0.0
GATEWAY	192.168.1.1
NetBIOS	QTL2250-nn-nnn

nn-nnn = Serial number section, as stated on the product label

Command Set

These commands are based on the SCPI style control system that is used by many manufacturers of test instruments. The entire SCPI specification has NOT been implemented but the command structure will be very familiar to anyone who has used it before.

- Commands are NOT case sensitive.
- Commands are in a hierarchy separated by ':' (LEVel1:LEVel2:LEVel3).
- Most words have a short form (e.g. 'register' shortens to 'reg'). This will be documented as REGister, where the short form is shown in capitals.
- Many commands take parameters. These are separated by spaces after the main part of the command (e.g. "meas:volt:self 3v3?" obtains the 3v3 self test measurement).
- Query commands that return a value have a '?' on the end.
- Commands with a preceding '*' are special control/query commands, found on all devices.
- Commands that do not return a particular value will return "OK" or "FAIL". Unless disabled, the fail response will also append a text description for the failure if it can be determined.

Identify commands

HELLO?

Returns the descriptive name of the module

HELP

Returns basic help to the terminal, including a list of more detailed help commands.

CONFig:ALIAS [Name]
CONFig:ALIAS?

Sets or returns the module Alias. This is a custom text descriptor which can be used to identify between modules where multiple units are used.

Basic Setup commands

These are standard commands to setup the module

 # [comments]

Any command beginning with a # character is ignored as a comment.

***RST**

Triggers a reset; the module will behave as if it had just been powered on. A reset will cause USB/LAN connections to be lost and the module must be reconnected again.

CONFig:DEFault STATE

Sets the unit to its power on default state. This will leave the unit in the same state as if it was power cycled.

CONFig:DEFault FACTory

Sets the unit to the state it left the factory in. This will set any device stored parameters to factory defaults. It will not affect the calibration settings for the unit.

If this command changes the NetBIOS name, a power cycle or ***RST** command will be required for the change to take effect.

CONFig:FAULT:RESet

Resets fault flags and activates the unit if outputs have tripped due to over current/over power.

CONFig:FAULT?

Returns the state of the output fault flags. This will return OK unless an over current/over power fault has occurred.

CONFig:MESSages [SHORt|USER]
CONFig:MESSages?

Sets or returns the message returns mode. When ‘SHORT’, detailed error messages will be suppressed.

Ethernet commands

Ethernet commands are generally persistent through reset. Some network settings may not take effect until a power cycle.

CONFig:ETHernet:NAME[xxxx]
CONFig:ETHernet:NAME?

Sets the NetBIOS name of the unit. This is persistent through reset. You will have to reset/power cycle the unit for the changes to take effect.

CONFig:ETHernet:IP [x.x.x.x]
CONFig:ETHernet:IP?

Sets the static IP address of the unit. Setting a static IP address will disable DHCP if it is currently enabled. You will have to reset/power cycle the unit for the change to take effect.

CONFig:ETHernet:MASK [xxx.xxx.xxx.xxx]
CONFig:ETHernet:MASK?

Sets the IP mask of the unit.

CONFig:ETHernet:DNS[1|2] [x.x.x.x]
CONFig:ETHernet DNS[1|2]?

Sets the primary and secondary DNS addresses for the unit.

CONFig:ETHernet:GATE[x.x.x.x]
CONFig:ETHernet GATE?

Sets the Gateway of the unit. You will have to reset/power cycle the unit for the change to take effect.

CONFig:ETHernet:DHCP [ON|OFF]
CONFig:ETHernet:DHCP?

Enables/disables the DHCP client. When enabled, the controller will try to obtain an IP address from the local DHCP server.

CONFig:ETHernet:HTTP:PORT [#Number]
CONFig:ETHernet:HTTP:PORT?

Gets/sets the port number to use for HTTP control

CONFig:ETHernet:TCP:PORT [#Number]
CONFig:ETHernet:TCP:PORT?

Gets/sets the port number to use for TCP control

CONFig:ETHernet:REST [ON|OFF]
CONFig:ETHernet:REST?

Gets/sets the enable state for REST control of the device

CONFig:ETHernet:TCP [ON|OFF]
CONFig:ETHernet:TCP?

Gets/sets the enable state for TCP control of the device

CONFig:ETHernet:LOCATE [ON|OFF]
CONFig:ETHernet:LOCATE?

Gets/sets the enable state for the LAN location system. This allows devices to be located on the network via a broadcast packet. Disabling this will prevent applications such as TestMonkey and QIS from locating the module automatically (though manual connection can still be used).

Auto test commands

These commands are used to trigger tests to run on the module and to check the test status, using the main ‘auto test’ system.

RUN:TEST
RUN:TEST?

Begins a test process, based on the currently configured auto-tests. Return responses are [IDLE|BUSY|COMPLETE|FAIL]

RUN:STOP

Orders the current auto-test process to halt

RUN:STATus?

Returns the status of the test process that is running. This is a multi-line response containing user readable status points for each major test section and basic details on any failures if the testing is complete

RUN:INTerrupt?

Returns active interrupt flags. Flags are cleared automatically after they are read. This can be polled to trigger the download of results when they are complete or to detect that a cable has changed. Responses area comma list of: [NONE|TEST_RUN|PAT_CHANGE|CAB_CHANGE|FAULT|COMPLETE]

AUTO:ENAble [ON|OFF]
AUTO:ENAble?

Master enable flag for the auto-test system. When enabled, the currently selected tests will be run automatically when a cable is detected. This is the same sequence as can be run manually by the RUN:TEST command.

AUTO:[#module]ENAble [ON|OFF]
AUTO:[#module]ENAble?

Individual enable options for auto-test modules. When enabled, the test will run either on cable test (with master enable active) or on RUN:TEST command. Attempting to test higher level functions without mapping and training being complete is likely to cause failures.

Test modules available are:

Test module [#module]	Purpose
EEDECODE	Decode the free-side cable EEPROM to work out the cable type, protocol, and speed for later tests
MAP	Perform link mapping
TRAIN	Train the link to the target speed
EYEHEIGHT	Test the eye height
EYESCAN	Run a full eye scan
BERT	Run live BERT monitoring after testing

Link commands

These commands allow you to control the parameters used to test the high-speed links. Links are named with respect to where they are transmitted from:

Logical link name [#link_name]	Description
A0	Link transmitted from the 'A' port on lane number 0. This will be received on the 'B' port in a standard cable
A1	
A2	
A3	
B0	Link transmitted from the 'B' port on lane number 0. This will be received on the 'A' port in a standard cable
B1	
B2	
B3	

These link names are at a 'logical' level and can auto map to take account of cross-overs in the cable. After the mapping test is run, with a cable that has an unusual link cross-over, the link name (ie: A1) will now refer to the lane 1 transmitter on the A port and whichever receiver the link was detected on. The detected receiver may be on either port (allowing for loopbacks to be tested).

For debug purposes, [#link_name] parameters in commands can also take ‘physical’ link names. When these are used, mapping is ignored, and a 1-1 ‘standard’ cable is assumed. All names reference to the A port.

Physical link name [#link_name]	Description
TX_0	Link transmitted from the ‘A’ port on lane number 0 and received on the ‘B’ port on lane number 0
TX_1	
TX_2	
TX_3	
RX_0	Link received from the ‘A’ port on lane number 0 and transmitted from the ‘B’ port on lane number 0
RX_1	
RX_2	
RX_3	

For ‘standard’ 1-1 cables, the logical and physical names will work the same. For unusual routings, loopbacks and similar, the logical names are required

LINK:[#link_name]PATtern [PRBS7|PRBS9|PRBS15|PRBS23|PRBS31]
LINK:[#link_name]PATtern?

Sets/Gets the test pattern to run on a link

LINK:[#link_name]OFFset [#bits_integer]
LINK:[#link_name]OFFset?

Sets/Gets the bit-offset in the pattern to add on a link

LINK:[#link_name]STATus?

Returns the status of the link:
[ERROR|UNTESTED|MAPPING|MAPPED|TRAINING|TRAINED|
FAILED_MAP|FAILED_CONN|FAILED_SPEED|FAILED_LOCK]

LINK:[#link_name]MAPping?

Returns the mapping of the link, this returns the link_name that a given logical transmitter connects to.

LINK:SPEED [2.5G|3G|5G|6G|8G|12G|16G|22.5G|24G]
LINK:SPEED?

Sets/Gets the link speed to use for high-speed testing

LINK:XML?

Returns an XML data block with all link results data

Eye commands

These commands allow you to control the eye tests

EYE:[#link_name]TEST

Start an eye scan test

EYE:[#link_name]:HEIght:TEST

Start an eye height test

EYE:[#link_name]:HEIght?

Returns the height measurement of the eye in mV

EYE:[#link_name]XML?

Returns a block of XML describing the eye measurements

BERT commands

These commands allow you to track real time bit errors.

BERT:[#link_name]ENAble [ON|OFF]
BERT:[#link_name]ENAble?

Individual enable options for each BERT

BERT:[#link_name]STATus?

Returns the BERT status, to check it is running and results will be valid [DISABLED|RUNNING|SATURATED|ERROR]

BERT:[#link_name]HOLD [ON|OFF]

Holds the counters for a BERT link

BERT:[#link_name]RESet

Resets the counters for a BERT link

BERT:[#link_name]ERRors?

Integer count of errors detected

BERT:[#link_name]TIME?

Elapsed BERT time in seconds

BERT:[#link_name]RATE?

BERT error rate in exponential form as errors/bits sent

BERT:XML?

Returns an XML data block with all BERT data

Cable commands

These commands allow you access the cable EEPROM and similar functions

CABLE:[A|B]PRESent?

Checks if a cable is present on the selected port

CABLE:[A|B]:EEprom:PRESent?

Checks if an active EEPROM is responding on the selected port

CABLE:[A|B]:EEprom:XML?

Returns EEPROM contents and decode in an XML block

CABLE:IDentify?

Returns a basic decode of the attached cable, based on its EEPROM data

[PROTOCOL,SPEED]

Sideband commands

These commands allow testing of sideband signals

Sideband [#side_name]	Description
MODPRSL_A | MODPRSL_B	MODPRSL (present) pin on A or B port
INTL_A | INTL_B	INTL pin, normally connected to the EEPROM
SDA_A | SDA_B	SDA pin, normally connected to the EEPROM
SCL_A | SCL_B	SCL pin, normally connected to the EEPROM
CADDR_A | CADDR_B	CADDR pin, normally connected to the EEPROM

`SIDEband:[#side_name]STATus?`

Returns the status of a given sideband [ERROR|UNTESTED|MAPPING|INTERNAL_CONNECT|NO_CONNECT| DIRECT_CONNECT]

`SIDEband:[#side_name]MAPping?`

Returns the mapping for a given sideband (#side_name or INTERNAL)

Debug Commands

These commands are used to look at the details of the module and its operation. You may be requested to use these if reporting a fault to Quarch.

***IDN?**

Displays a standard set of information, identifying the device. An example return is shown below:

```
Family: Quarch Cable Tester      [The parent family of the device]
Name: PCIe 4.0 / SAS4.0 Cable Tester  [The name of the device]
Part#: QTL2250-01                [The part number of the hardware]
Processor: QTL2212-01,5.001      [Part# and version of firmware]
Bootloader: QTL2212-01,5.001     [Part# and version of bootloader]
FPGA 1: QTL2211,1.3              [Version of FPGA core]
Serial#: QTL2250-01-001          [Serial number of the module]
```

***TST?**

Runs a set of standard tests to confirm the device is operating correctly; these tests are also performed at start up. Returns ‘OK’ or ‘FAIL’ followed by a list of errors that occurred, each on a new line.

***SERIAL?**

Returns the serial number of the internal power module.

***MAC?**

Returns the MAC address of the module

MEASure:VOLT:SELF [channel]?

Returns a self test voltage measurement on the modules internal rails.

channel can be: 12V, 3V3, 1V8, 1V8_MGT, 1v5, 1v2, 0v9, 0v85

Customer support from Quarch

There are multiple ways to access the support you need. You can contact us directly or access an extensive range of valuable support materials from http://quarch.com/support.

Contact us direct

Get going quickly and easily, with help direct from the engineers:

- Call **+44 1343 508 140** or email support@quarch.com during UK office hours.
- Our international partners are well trained in the use of our products and can deal with many basic technical queries from within your time zone, if you prefer. Check http://quarch.com/resellers for the contact details of your regional supplier.

Access support from the Quarch website

You can download up-to-date software and drivers, technical manuals, datasheets and more from our website. To help you get started quickly we provide additional documents, such as examples in Perl, Python and C# and Telnet and Serial instructions.

Key places to visit on the Quarch website

- Register your Quarch product to confirm your international warranty: http://quarch.com/product-registration
- Download a wide range of documentation, free applications and drivers to help you make the best possible use of your Quarch tools: http://quarch.com/content/downloads
- Access the Quarch support forum (http://quarch.com/forum):
  - Find discussion topics, support information and testing ideas.
  - Browse existing topics or login to your user account to ask for information and advice.
- Sign up for Quarch Technical Updates to get the most out of your Quarch products. Updates are published approximately once a quarter and include news about the latest features, tools, application notes and software updates. See http://quarch.com/content/sign-quarch-technical-updates.