

## Quarch Technology Ltd

# Torridon Active SATA Multiplexer Module

## Technical Manual

For use with:

**QTL1390 - Torridon Active SATA Multiplexer Module**

Using Quarch firmware version 4.0 and above

Friday, 17 June 2011



## Change History

1.0	1st June 2011	Initial Release
1.1	17 June 2011	Added missing commands

## Contents

<b>Introduction .....</b>	<b>4</b>
<b>Technical Specifications .....</b>	<b>5</b>
Electrical Characteristics:.....	5
Mechanical Characteristics:.....	5
QTL1390 Active SATA Multiplexer Module .....	5
Front Panel LEDs.....	7
<b>Control Interfaces .....</b>	<b>9</b>
Voltage Measurements .....	10
Default Startup State .....	11
<b>Controlling the Module .....</b>	<b>12</b>
Serial Command Set.....	12
SCPI Style Commands .....	12
<b>Control Register Map .....</b>	<b>16</b>
Register Definitions .....	17
Other Registers.....	18

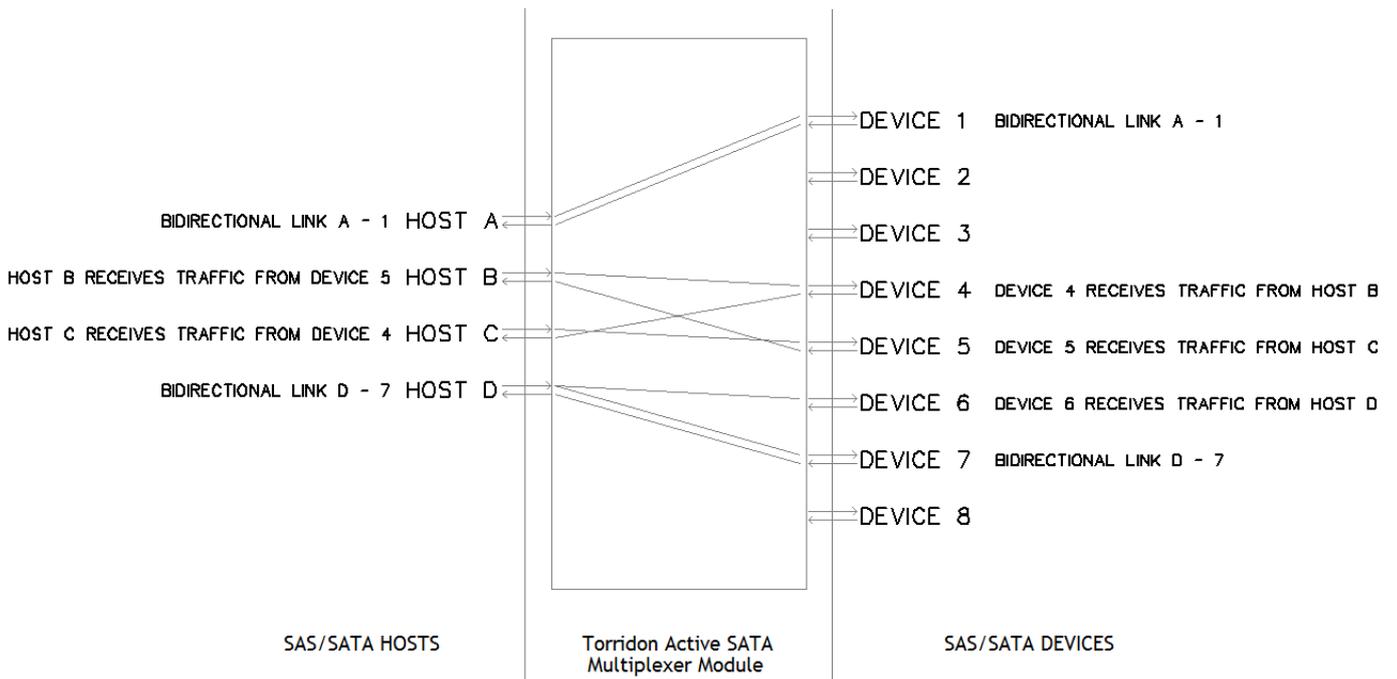
## Introduction

The **Torricon Active SATA Multiplexer Module** has 4 SATA Host ports and 8 SATA Device ports, all ports can transmit SAS or SATA data at speeds up to 6Gb/s.

The user can set up a bi-directional connection between any 2 ports as if they were cabled together. Data received at one port can also be forwarded to any of the other ports, sending a copy of the data received, or allowing for non standard cabling setups.

An example configuration is shown below where:

- HOST A is connected to DEVICE 1
- HOST D is connected to DEVICE 7, in addition the data received from HOST D is forwarded to DEVICE 6
- Data received at HOST B is sent to DEVICE 4, the data received at DEVICE 4 is sent to HOST C, data received at HOST C is sent to DEVICE 5, data received at DEVICE 5 is sent to HOST B



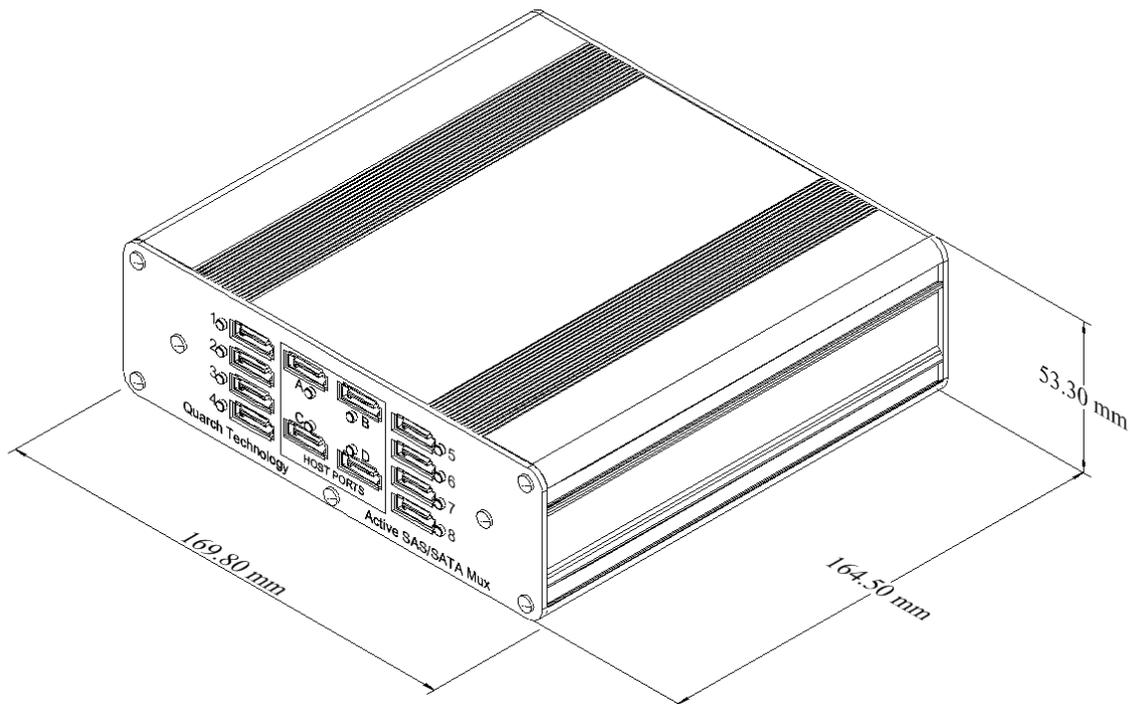
## Technical Specifications

### Electrical Characteristics:

Paramter	Min	Typical	Max
Input differential voltage swing	200mV	-	2000mV
Output differential voltage swing (Programmable)	350mV	-	1750mV
Output rise/fall time (20%-80%)		60 pS	
Deterministic output jitter	-	0.1UI	0.2UI
Random output jitter	-	6 mUI RMS	8 mUI RMS
Propagation delay	-	1 nS	-
Channel to channel skew	-	300 pS	-

### Mechanical Characteristics:

#### QTL1390 Active SATA Multiplexer Module



## Basic Concepts

Each port on the multiplexer has an associated transmitter that sends data out from the port, and a receiver that reads the data coming into the port. The internal pathways are configured by specifying for each transmitter, the receiver port that it should take data from.

In this manual the word CONNECTION is used to imply a bidirectional link between two ports, i.e port A takes its data from port 5, and port 5 takes its data from port A. Bidirectional CONNECTIONS can be made with the single command:

### **MUX:CONnect [PORT] [PORT]**

This command will connect the two ports specified (i.e transmitter A to receiver 5 and transmitter 5 to receiver A) and it will remove any other prior connections associated with those two ports

**In order to create only simple bi-directional connections, always use the MUX:CONnect command.**

We use the term FORWARD to describe a unidirectional link (i.e Transmitter 6 to Receiver B, but not Transmitter B to Receiver 6). Unidirectional FORWARDS can be made with command:

### **MUX:FORward [Rx Port] [Tx Port]**

This command will send the data received at [Rx Port] and transmit it to [Tx port]. This command does not change any other pre-existing paths on the switch.

Examples:

We may have a Host cables to port A and a device cabled to port 6, we can connect ports A and 6 with the command:

### **MUX:CONnect A 6**

We may then wish to forward the data to a protocol analyzer connected to host port B and device port 4:

### **MUX:FORward A 4**

### **MUX:FORward 6 B**

We are now sending the data received at port A to port 6 and port 4, and we are sending the data received at port 4 to port A and port B. We can analyze the link traffic without having the analyser present in the link.

### Front Panel LEDs

The unit uses multicolor LEDs to give a visual indication of the connections in place. Each port has an LED that indicates the source of the data it's transmitting.

Each Host port has an associated color and the color is always the same. A solid LED means that there is a bidirectional link in place with a device port, which will also have a solid LED of the same color. A flashing LED means that this port is forwarding data from a device port, but not transmitting data back to it, the number of flashes indicates which device port.

Host Port	LED Color	Behaviour		
		OFF	FLASHING	ON
HOST A	PURPLE	This port is off, or the port it's receiving data from is off	This host port is receiving data from a device port, but not transmitting its received data back to that port. The number of flashes indicates which device port i.e 5 flashes = device port 5 <b>(port is receiving forwarded data)</b>	This host port is receiving data from a device port, and transmitting its received data to that port <b>(bidirectional connection)</b>
HOST B	RED			
HOST C	CYAN			
HOST D	BLUE			

Each device port LED will change color depending on the Host port it is receiving data from. A solid LED means that there is a bidirectional link in place with a host port, which will also have a solid LED of the same color. A flashing LED means that this port is forwarding data from a host port, but not transmitting data back to it, the color of the LED indicates which host port.

Device Port	Behaviour		
	OFF	FLASHING	ON
Device 1	This port is off, or the port it's receiving data from is off	This device port is receiving data from a host port, but not transmitting its received data back to that port. The color of the LED indicates which host port i.e RED = HOST B <b>(port is receiving forwarded data)</b>	This device port is receiving data from a host port, and transmitting its received data to that port. The color of the LED indicates which host port. i.e. RED = HOST B <b>(bidirectional connection)</b>
Device 2			
Device 3			
HOST D			

## Control Interfaces

All Torridon Control Modules are designed to be used with a Torridon Array Controller (QTL1079) or a single Torridon Interface Card (QTL1144).

The control cable is an ultra-thin Flex cable.

Control Interface	Form Factor	Torridon Module Ports	Control Methods Available	Interfaces
Torridon Array Controller	1U 19" Rack Mounted unit	24 at the front, 4 at the rear	Serial Scripting Script Generation through <b>TestMonkey GUI</b>	Serial via DB9 or RJ45 Ethernet
Torridon Interface Card	102mm x 26mm PCB	1 port	Serial Scripting Script Generation through <b>TestMonkey GUI</b>  Real-time USB Control via <b>TestMonkey GUI</b>	Serial via DB9 or RJ45 USB

## Voltage Measurements

The modules are capable of measuring various voltages both for self test and to assist in the testing of a customer's system. The following measurement points are available:

Measurement Command	Description	Resolution / Accuracy
<b>MEASure:VOLTage:SELF 1v2?</b>	Returns the voltage of the modules internal 1.2v power rail	64mV / 5%
<b>MEASure:VOLTage:SELF 1v8?</b>	Returns the voltage of the modules internal 1.8v power rail	64mV / 5%
<b>MEASure:VOLTage:SELF 3v3?</b>	Returns the voltage of the modules internal 3.3v power rail	64mV / 5%
<b>MEASure:VOLTage:SELF 5v?</b>	Returns the voltage of the modules internal 5v power rail	64mV / 5%
<b>MEASure:VOLTage:SELF 12v?</b>	Returns the voltage of the modules internal 12v power rail	64mV / 5%

## Default Startup State

On power up or reset, the control modules enter a default state. Each host port is connected to a device port, and hosts and devices may be plugged into these ports and will be instantly connected

Host Port	Default Device Port
A	1
B	5
C	4
D	8

## Controlling the Module

The module can be controlled either by:

- Serial ASCII terminal (such as HyperTerminal)  
This is normally used with scripted commands to automate a series of tests. The commands are normally generated by a script or user code (PERL, TCL, C, C# or similar).
- Telnet Terminal (Only when connected to an Array Controller). This mode uses exactly the same commands as the serial ASCII terminal
- USB  
Quarch's TestMonkey application can control a single module via USB, this allows simple graphical control of the module.

## Serial Command Set

When connected via a serial terminal, the module has a simple command line interface

### SCPI Style Commands

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.

- SCPI commands are NOT case sensitive
- SCPI commands are in a hierarchy separated by ':' (LEVe11:LEVe12:LEVe13)
- 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.
- Some 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 all have a '?' on the end
- Commands with a preceding '\*' are basic control 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.

### # [comments]

Any line beginning with a # character is ignored as a comment. This allows commenting of scripts for use with the module.

### \*RST

Triggers a reset, the module will behave as if it had just been powered on

#### **\*CLR**

Clear the terminal window and displays the normal start screen. Also runs the internal self test. The same action can be performed by pressing return on a blank line.

#### **\*IDN?**

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

Family:	Torridon System	[The parent family of the device]
Name:	Ethernet Cable Pull Module	[The name of the device]
Part#:	QTL1271-01	[The part number of the hardware]
Processor:	QTL1159-01,3.50	[Part# and version of firmware]
Bootloader:	QTL1170-01,1.00	[Part# and version of bootloader]
FPGA 1:	1.0	[Version of FPGA core]

#### **\*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.

#### **CONFig:MODE BOOT**

Configures the card for boot loader mode (to update the firmware), requires an update utility on the PC.

#### **CONFig:MESSages [SHORT|USER]**

#### **CONFig:MESSages?**

Gets or sets the mode for messages that are returned to the user's terminal

**Short:** Only a "FAIL" or "OK" will be returned

**User:** Full error messages are returned to the user on failure

#### **CONFig:TERMinal USER**

Sets the terminal response mode to the default 'User' setting. This is intended for use with HyperTerminal or similar and manually typed commands

#### **CONFig:TERMinal SCRIPT**

Sets the terminal response mode for easier parsing. Especially useful from a UNIX/LINUX based system. Characters sent from the PC are not echoed by the device and a <CR><LF> is sent after the cursor to force a flush of the USART buffer.

**CONFig:TERMinal ?**

Returns the current terminal mode

**CONFig:DEFault:STATE**

Resets the state of the module. This will set all source/signal/glitch etc logic to its default power-on values. Terminal setting will not be affected. This command allows the module to be brought back to a known state without resetting it.

**MUX:CONnect [port 1] [port 2]**

Creates a bi-directional link between the two specified ports, any other ports that were connected to either port are turned off

**MUX:FORward [RX Port] [ TX port]**

Forwards data received at [RX port] to [Tx port] no other connections are altered.

**MUX:[PORT]:SOURce?**

Returns the port number that [PORT] is receiving data from

**MUX:[PORT]:STATE?**

Returns the port source of data for this port and any target port(s) that data from this port is sent to.

**MUX:OFF [Port|ALL|HOST|DEVICE]**

Turns off the receiver and transmitter for [Port], or if ALL is used, it turns off all ports. HOST and DEVICE turns off all host or all device ports, respectively.

**CONFig:MUX:[PORT]:PREEmphasis #**

**CONFig:MUX:[PORT]:PREEmphasis ?**

Set the level of transmit de-emphasis for [PORT] according to the following table:

0	No De-Emphasis
1	Approximately 2dB output de-emphasis

2	Approximately 4dB output de-emphasis
3	Approximately 6dB output de-emphasis

**CONFig:MUX:[PORT]:Equalisation #**

**CONFig:MUX:[PORT]:Equalisation ?**

Set the level of receiver equalization for [PORT] from 0 (minimum) to 15 (maximum)

**CONFig:MUX:[PORT]:AMPlitude #**

**CONFig:MUX:[PORT]:AMPlitude ?**

Set the transmitter differential amplitude for [PORT] from 0 (350mV) to 15 (1750mV)

## Control Register Map

This map is provided for debug and advanced setup purposes only. While you can use the 'Read' and 'Write' commands, we STRONGLY recommend you use the SCPI based commands instead of writing to the registers directly.

Access to the FPGA registers should not be required for the majority of operations and customers are encouraged to use the high level commands in order to maintain compatibility with future firmware versions. It is listed here for reference / debug purposes.

Address	Name	Description
0x00	Global Control	Control Register
0x01	Port 1 Control	Port Control Register
0x02	Port 2 Control	Port Control Register
0x03	Port 3 Control	Port Control Register
0x04	Port 4 Control	Port Control Register
0x05	Port 5 Control	Port Control Register
0x06	Port 6 Control	Port Control Register
0x07	Port 7 Control	Port Control Register
0x08	Port 8 Control	Port Control Register
0x09	Reserved	Reserved
0x0A	Port A Control	Port Control Register
0x0B	Port B Control	Port Control Register
0x0C	Port C Control	Port Control Register
0x0D	Port D Control	Port Control Register
0xFE	FPGA PART NUMBER	FPGA Part Number Register
0xFF	FPGA VERSION	FPGA Code Version Register

## Register Definitions

### 0x00 - Global Control

15	14	13	12	11	10	9	8
RESERVED (READ ONLY)							
7	6	5	4	3	2	1	0
RESERVED						BUSY	STROBE

Name

Description

STROBE

Toggle this bit to activate the new mux mappings

BUSY

This bit is set high when the FPGA is writing data across i2c, the strobe bit should not be set unless this bit is clear

**0x01 – 0x0D – Port Setup (Default Value)**

15	14	13	12	11	10	9	8
Reserved (11111)					AMPLITUDE (110)		
7	6	5	4	3	2	1	0
Reserved (1111)				INCOMING_PORT (0x0000)			

Name	Description
INCOMING_PORT	Set to 0 to power off output Set to 1-8, A-D to receive data from the specified Port 9, E and F are not valid values and will turn the port off, the same as 0
AMPLITUDE	Set the tx output swing for this port 0=500mVp-p 6 = 1500mVp-p typical

**0x0E – 0x1A – Signal Conditioning (Default Value)**

15	14	13	12	11	10	9	8
Reserved (11111)				EQUALIZATION (1001)			
7	6	5	4	3	2	1	0
Reserved (000000)						DE-EMPHASIS (00)	

Name	Description
EQUALIZATION	Set Equalization Level from 0 – No Equalization to 0xF – Maximum Equalization
DE-EMPHASIS	Set De-emphasis Level from 0 – No De-emphasis to 0x3 – 6dB de-emphasis

**Other Registers**

**0xFE – CPLD Part Number Register**

15..0
4 digit CPLD part number

	Name	Description
PART_NUMBER		Forms the QTLnnnn part number

**0xFF – CPLD Version Register**

15..8	7..0
MAJOR REVISION	MINOR REVISION

	Name	Description
MAJOR_REVISION		Should read as 1 or higher
MINOR_REVISION		Should read as 0 or higher