Quarch Technology Ltd

Torridon SBB 2.0 Canister Control Module

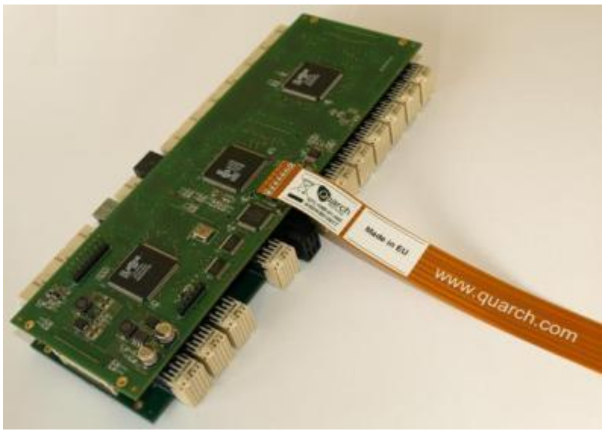
Technical Manual

For use with:

QTL1069 - Torridon SBB 2.0 Canister Control Module

Updated for firmware version 3.5x

Thursday, 08 January 2015



Change History

1.1	2nd November 2010	Added new commands for v3.5 firmware
1.2	25th November 2010	Updated with new format common sections
1.3	23rd February 2012	Minor corrections to command listings

Contents

Introduction

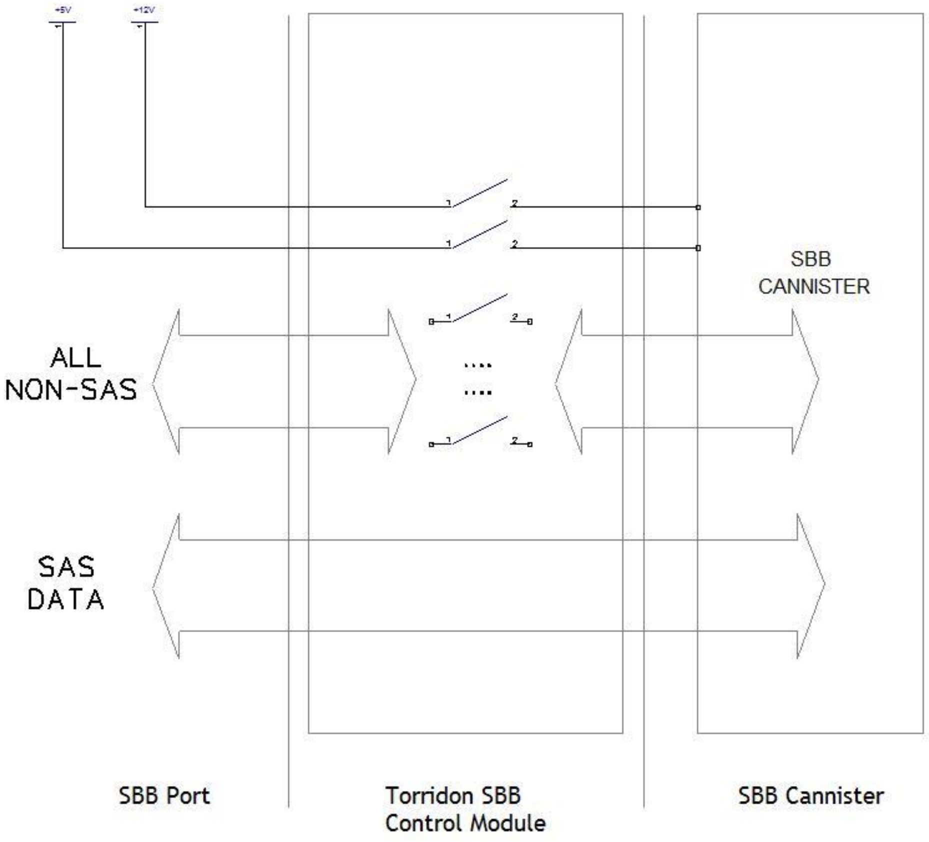
The **Torridon SBB 2.0 Canister Control Module** allows remote switching of the signals and power supply rails to an SBB 2.0 Canister Controller whilst plugged into a host system.

These features allow the user to perform tasks on the live storage system, such as:

- Hot insertion and removal of an SBB 2.0 canister card
- Precise timing control over each pin during a hot swap plug/pull
- Breaking signal/power lines to test system response to a failure

All of power and signal pins are individually switched, allowing complete control over the power up sequence of a canister, SAS Data pins are not switched. Precise timing control allows pins to be mated at a 10uS resolution, allowing high accuracy and repeatable testing.

Quarch modules may be customized to support other proprietary signals on request.



Technical Specifications

Switching Characteristics:

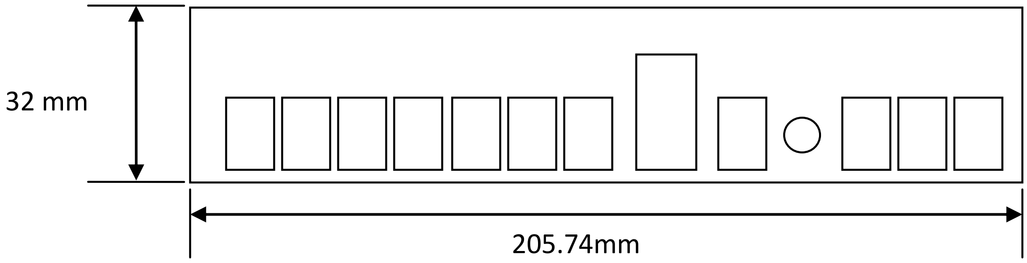
Connector Module	SBB Module Pins	Description	Switching Action
M1	HS1_AB .. HS8_AB HS1_BA .. HS8_BA	SBBMI High Speed Inter-Canister Communication Signals	Each signal pair connected between backplane and canister with 100ohm differential track
	$LS8_{AB}$, $LS8_{BA}$	SBBMI Low-Speed Inter-Canister Communication Signals	Each signal is individually switched by an analog switch
	AC_GOOD, STANDBY_PWR	SBBMI Low-Speed Drive Status Signals	Each signal is individually switched by an analog switch
M2	DRIVE_1..6_TX DRIVE_1..6_RX	SBBMI High-Speed Drive Communication Signals	Each signal pair connected between backplane and canister with 100ohm differential track
	DRIVE_1..6_FAULT_L DRIVE_1..6_GPO_L DRIVE_1..6_INPL_L	SBBMI Low-Speed Drive Status Signals	Each signal is individually switched by an analog switch
M3	MATED_L, ENCLOSURE_INTR_L, SGPIO_SLI, SGPIO_SCK, SGPIO_SDO, SGPIO_SDI, SCL0..2, SDA0..2, POWER_OFF_L, PS1..2_LED_L, PS1..2_PRES_L, PS1..2_ALERT_L, SLOT_ID, CARD_IO_TEST_L, TWI_BUS1_RST_L, TWI_BUS2_RST_L, SPECIAL_1*	SBBMI Low-Speed Canister Support Signals	Each signal is individually switched by an analog switch
	LS1..7_AB	SBBMI Low-Speed	Each signal is

	LS1..7_BA	Inter-Canister Communication Signals	individually switched by an analog switch
	ENC_DEF_LP1..7 ENC_DEF_HP1..7	SBBMI System Defined Inputs and Outputs	Each signal is individually switched by an analog switch
M5	HS9..17_AB HS9..17_BA	SBBMI High Speed Inter-Canister Communication Signals	Each signal pair connected between backplane and canister with 100ohm differential track
M6	12_A, 12V_B	12V Power Pins	Switched by 35A power FETs . These signals are individually switched to maintain compatibility with SBB 1.0 systems where 12V_B is a 5V power pin
M7	DRIVE_7..12_TX DRIVE_7..12_RX	SBBMI High-Speed Drive Communication Signals	Each signal pair connected between backplane and canister with 100ohm differential track
	DRIVE_7..12_FAULT_L DRIVE_7..12_GPO_L DRIVE_7..12_INPL_L	SBBMI Low-Speed Drive Status Signals	Each signal is individually switched by an analog switch
M8	DRIVE_13..18_TX DRIVE_13..18_RX	SBBMI High-Speed Drive Communication Signals	Each signal pair connected between backplane and canister with 100ohm differential track
	DRIVE_13..18_FAULT_L DRIVE_13..18_GPO_L DRIVE_13..18_INPL_L	SBBMI Low-Speed Drive Status Signals	Each signal is individually switched by an analog switch
M9	DRIVE_19..24_TX DRIVE_19..24_RX	SBBMI High-Speed Drive Communication Signals	Each signal pair connected between backplane and canister

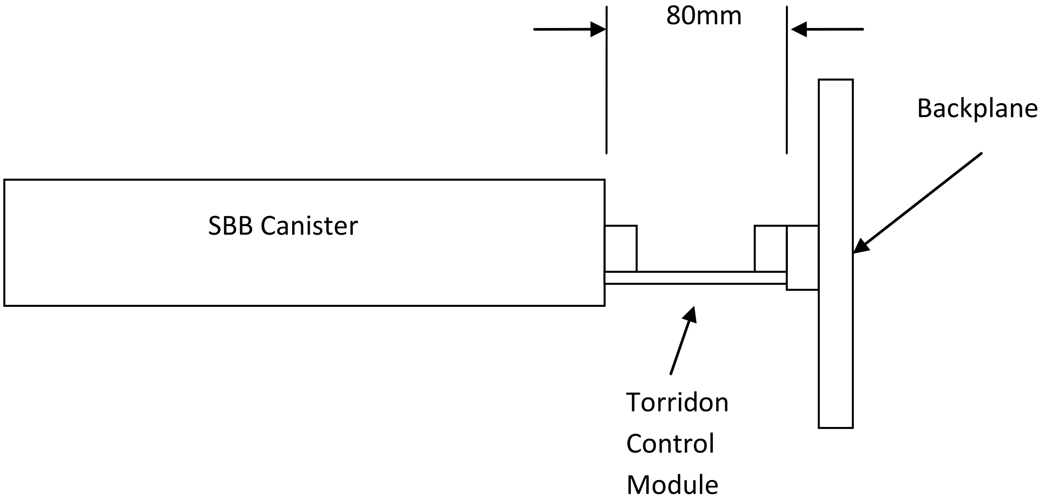
			with 100ohm differential track
	DRIVE_19..24_FAULT_L DRIVE_19..24_GPO_L DRIVE_19..24_INPL_L	SBBMI Low-Speed Drive Status Signals	Each signal is individually switched by an analog switch
M10	DRIVE_25..30_TX DRIVE_25..30_RX	SBBMI High-Speed Drive Communication Signals	Each signal pair connected between backplane and canister with 100ohm differential track
	DRIVE_25..30_FAULT_L DRIVE_25..30_GPO_L DRIVE_25..30_INPL_L	SBBMI Low-Speed Drive Status Signals	Each signal is individually switched by an analog switch
M11	DRIVE_31..36_TX DRIVE_31..36_RX	SBBMI High-Speed Drive Communication Signals	Each signal pair connected between backplane and canister with 100ohm differential track
	DRIVE_31..36_FAULT_L DRIVE_31..36_GPO_L DRIVE_31..36_INPL_L	SBBMI Low-Speed Drive Status Signals	Each signal is individually switched by an analog switch
M12	DRIVE_37..42_TX DRIVE_37..42_RX	SBBMI High-Speed Drive Communication Signals	Each signal pair connected between backplane and canister with 100ohm differential track
	DRIVE_37..42_FAULT_L DRIVE_37..42_GPO_L DRIVE_37..42_INPL_L	SBBMI Low-Speed Drive Status Signals	Each signal is individually switched by an analog switch
M13	DRIVE_43..48_TX DRIVE_43..48_RX	SBBMI High-Speed Drive Communication Signals	Each signal pair connected between backplane and canister with 100ohm differential track
	DRIVE_43..48_FAULT_L DRIVE_43..48_GPO_L DRIVE_43..48_INPL_L	SBBMI Low-Speed Drive Status Signals	Each signal is individually switched by an analog switch

- Ground Pin M3, G6 is used as a Mated signal by some vendors; this may also be independently switched to SBB canister presence. Subject to volume and lead time, the units may be customized to suit any proprietary mated circuitry.

Mechanical Characteristics:



- The Modules have the same cross section as an SBB canister, allowing them to fit into any SBB compliant storage enclosure. Connector modules may be populated as required.



- The modules displace an SBB Canister by approximately 80mm when installed in system.

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.

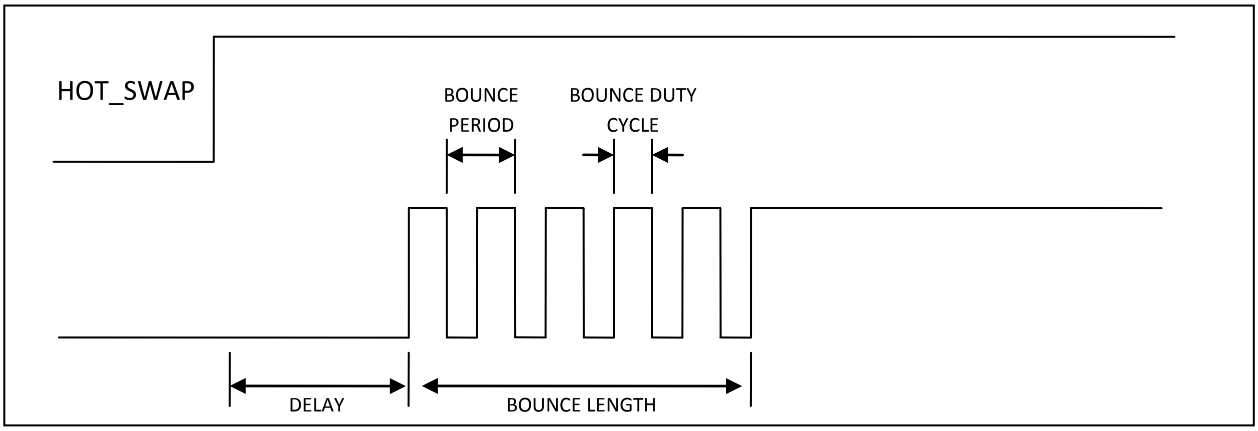
Control Interface	**Form Factor**	**Torridon Module Ports**	**Control Methods Available**	**Interfaces**
Torridon Array Controller	1U 19" Rack Mounted unit	24 at the front, 4 at the rear	Serial Scripting Script Generation through **TestMonkey GUI**	Serial via DB9 or RJ45 Ethernet
Torridon Interface Card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

Basic Concepts

Each switch on the module is called a 'Signal' and can be programmed to follow one of 6 programmable delay and bounce profiles (called 'Sources'). This allows the user to sequence the signal connections in the cable in up to six programmable steps.

Each of the programmable delay and bounce profiles is called a control source, S1 to S6. For each control source the user sets up a delay, and bounce parameters. Three special sources (S0, S7 and S8) are also provided as described in the table below.

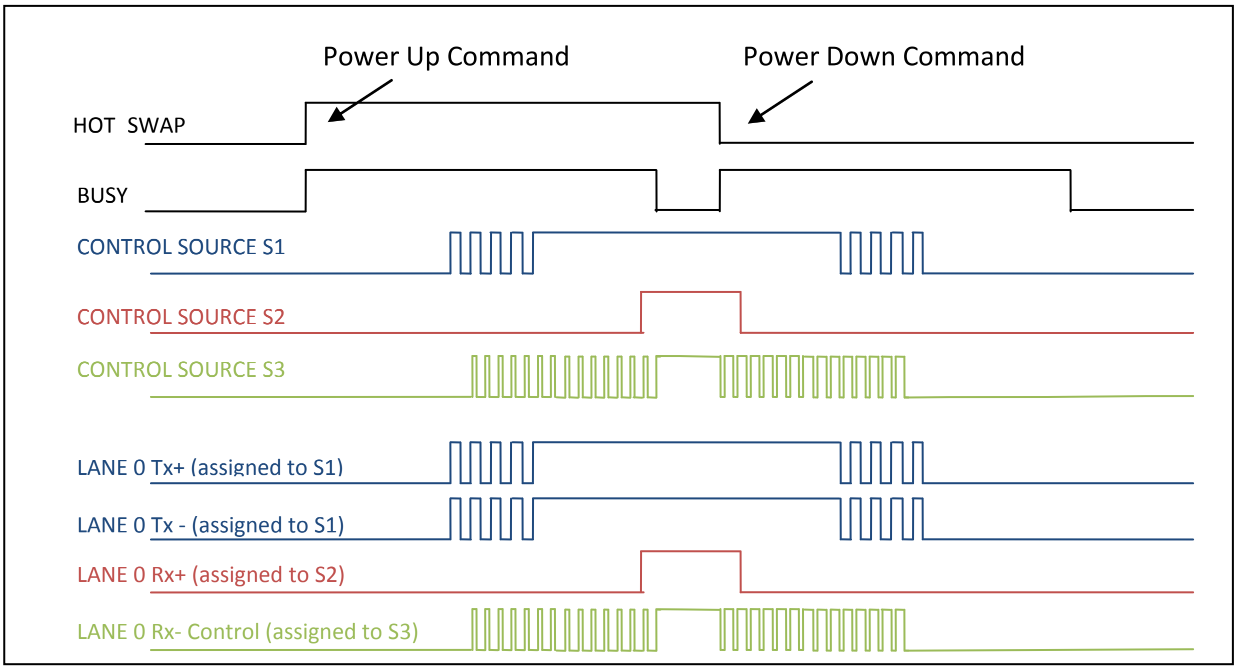
Control Source Parameters for a power up event (Basic Pin Bounce):



Once each delay period is set up, the user assigns each signal to follow the relevant control source, then uses the "run:power up" and "run:power down" commands to initiate the hot-swap.

The BUSY bit 1 in the control register is set during a power up, power down and short operation. This may be used to monitor for the completion of timed events.

Power up and Power down example:

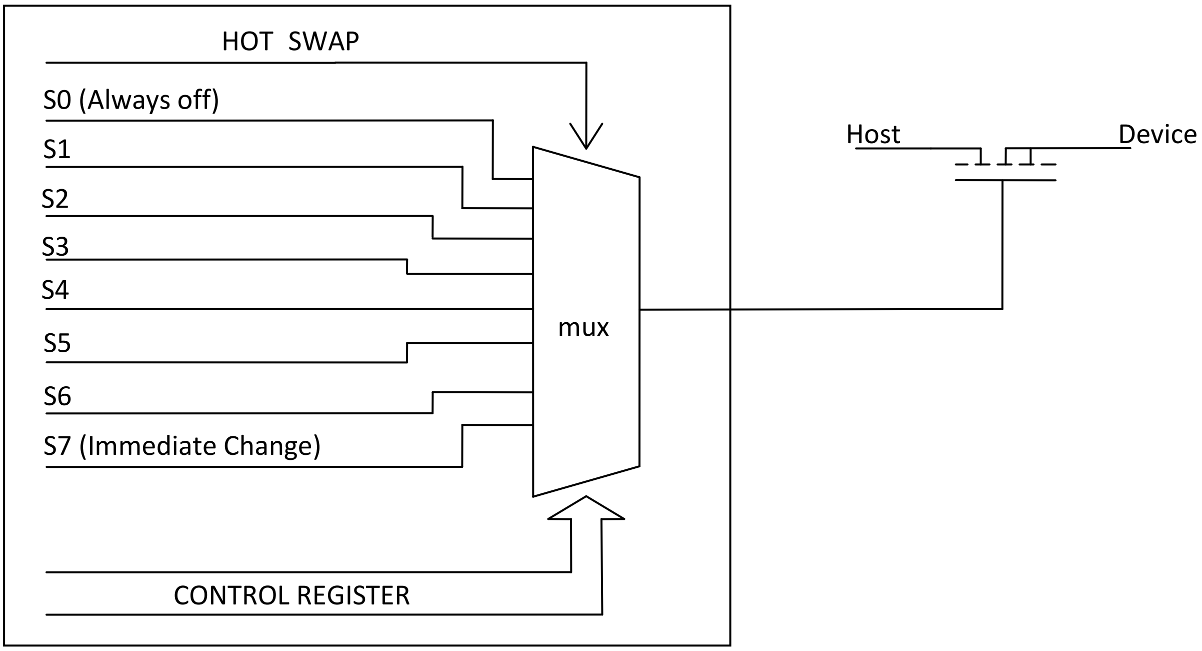

Power Up Command
Power Down Command
HOT SWAP
BUSY
CONTROL SOURCE S1
CONTROL SOURCE S2
CONTROL SOURCE S3
LANE 0 Tx+ (assigned to S1)
LANE 0 Tx - (assigned to S1)
LANE 0 Rx+ (assigned to S2)
LANE 0 Rx- Control (assigned to S3)

Signal Configuration

Each signal that is switched by the module is usually assigned to one of the 6 timed sources, S1 – S6. Each signal can also be assigned directly to 'always off' (source 0) or 'immediate change' (source 7).

To assign a signal to a control source, write to its CONTROL_REGISTER:

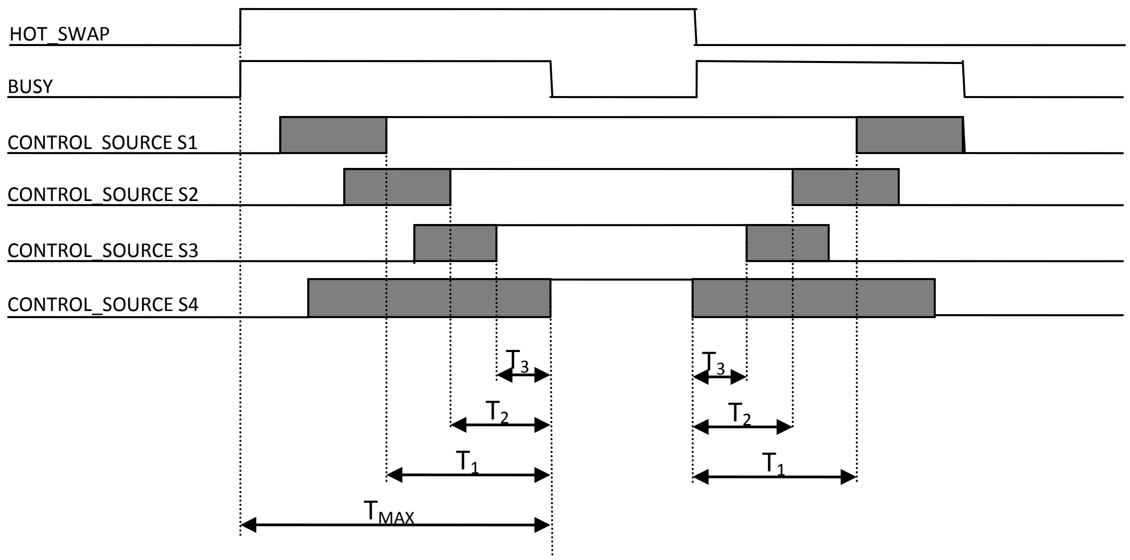
CONTROL_REGISTER Value	Description
0	Signal is always OFF
1	Signal assigned to control source 1
2	Signal assigned to control source 2
3	Signal assigned to control source 3
4	Signal assigned to control source 4
5	Signal assigned to control source 5
6	Signal assigned to control source 6
7	Signal changes with HOT_SWAP



This diagram shows the 8 possible source settings entering the control MUX for a switched signal. The value of the control register will determine which of the sources are used to control the signal. When enabled, the hot-swap line will cause the MUX to pass the control signal from that source through to the switch.

Power Up vs. Power Down Timing

Each control source is always configured with power-up parameters. The power-down profile is automatically generated by the module, and is the mirror image of the power up:



Power Down Times are automatically generated by the module but can be calculated as follows:

T_{MAX} = Largest Time Period
= the largest value from: (S1_DELAY + S1_BOUNCE_LENGTH),
(S2_DELAY + S2_BOUNCE_LENGTH),

up to

(S6_DELAY + S6_BOUNCE_LENGTH)

Power Down Delay T_X = T_{MAX} – (S_X_DELAY + S_X_BOUNCE_LENGTH)

If you require a different power down sequence then you can alter any of the source timing values, pin bounce or signal assignments while the module is in the plugged state. When you initiate the 'pull' action, the new settings will be used.

Pin Bounce Modes

Pin Bounce can be set in two ways:

1. Basic Pin-Bounce (Constant Oscillation Frequency):

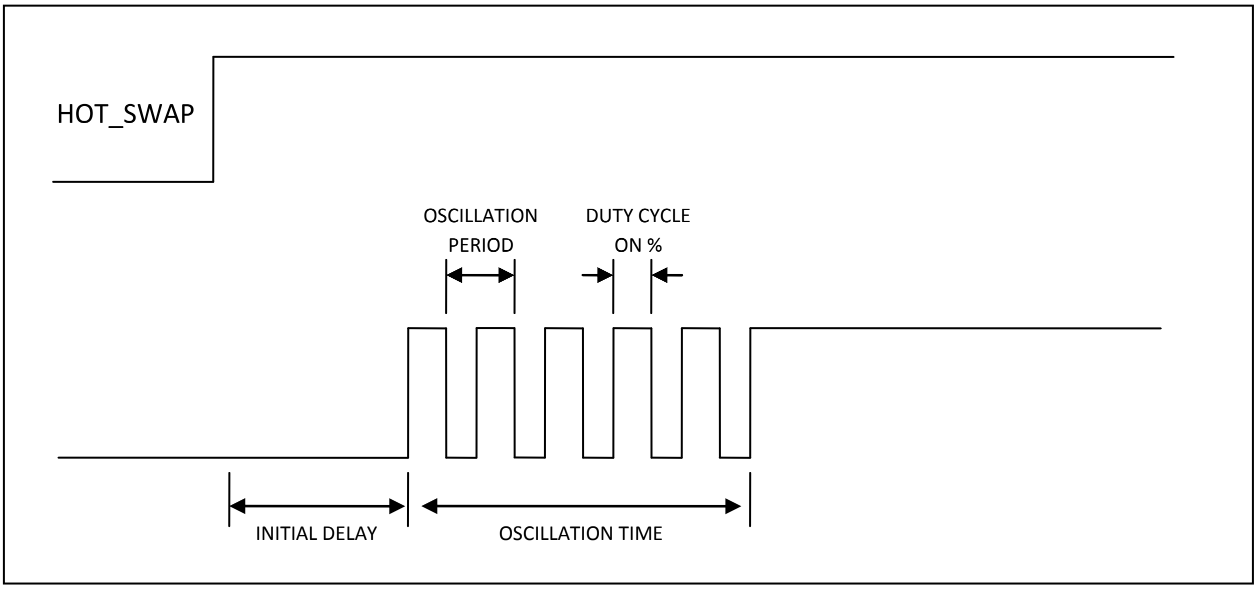
➢ The oscillation pattern length (Time) set in one of the two ranges:

- 0 - 127 milliseconds in steps of 1mS
- 0 – 1.27 seconds in steps of 10mS

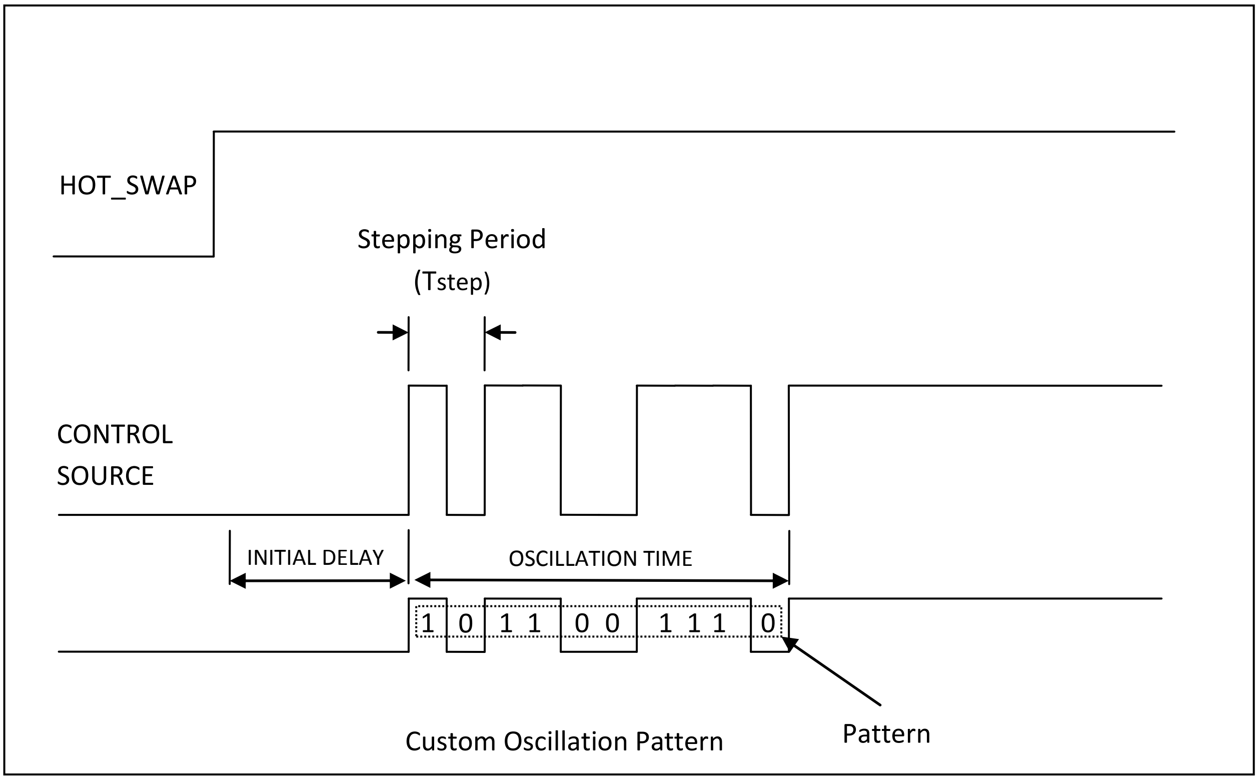
➢ The bounce period is for the pattern (T_{OSC}) is set on one of the two ranges:

- 0 - 1.27 milliseconds in steps of 10uS
- 0 – 127 milliseconds in steps of 1mS

➢ The Duty cycle (On %) is set as a percentage value in the range 1-99%



2. User Pin-Bounce (Custom Oscillation):
   - ➢ The oscillation pattern length (Time) set in one of the two ranges:
     - 0 - 127 milliseconds in steps of 1mS
     - 0 – 1.27 seconds in steps of 10mS
   - ➢ The stepping period (Tstep)is for the pattern is set on one of the two ranges:
     - 0 - 1.27 milliseconds in steps of 10uS
     - 0 – 127 milliseconds in steps of 1mS
   - ➢ The Custom pattern is described in 100 bits, where 2 bits are stepped through in each Tstep period. The 100 bit pattern will loop if the oscillation pattern time is longer than the available pattern.



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
MEASure:VOLTage:SELF 16v?	Returns the voltage of the modules internal 16v power rail	32mV / 3%
MEASURE:VOLTage:SELF12v?	Returns the voltage of the modules internal 12V power rail	16 mV / 3%
MEASure:VOLTage:SELF 5v?	Returns the voltage of the modules internal 5v power rail	8mV / 2%
MEASure:VOLTage:SELF 3v3?	Returns the voltage of the modules internal 5.3v power rail	4mV / 1%
MEASure:VOLTage 12vinA?	Returns the voltage of the 12V power pins on the backplane (unswitched) side of the module	16 mV / 3%
MEASure:VOLTage 12voutA?	Returns the voltage of the 12V power pins on the canister (switched) side of the module	16 mV / 3%
MEASure:VOLTage 12vinB?	Returns the voltage of the 12V power pins on the backplane (unswitched) side of the Module	16 mV / 3%
MEASure:VOLTage 12voutB?	Returns the voltage of the 12v power pins on the canister(switched) side of the	16 mV / 3%

	module	
MEASure:VOLTage Standbyin?	Returns the voltage of STANDBY_PWR on the backplane (unswitched) side of the module	8mV / 2%
MEASURE:VOLTage Standbyout?	Returns the voltage of STANDBY_PWR on the canister (switched) side of the module	8mV / 2%

Default Startup State

On power up or reset, the control modules enter a default state. To make the module as easy to use as possible, the default state is a 'standard' hot-swap scenario with preset source and signal settings such that the "run:power up" command will immediately power up the canister without needing any initial setup.

The default hot-swap scenario will connect pre-charge then power then pins, each step with a 25mS delay. No pin-bounce is used and all sources are enabled.

Source Number	**Source Enabled**	**Initial Delay**	**Pin Bounce Mode**	**Bounce Length**	**Bounce Period**	**Bounce Duty Cycle**
1	YES	0mS	Standard	0mS	0uS	50%
2	YES	25mS	Standard	0mS	0uS	50%
3	YES	50mS	Standard	0mS	0uS	50%
4	YES	0mS	Standard	0mS	0uS	50%
5	YES	0mS	Standard	0mS	0uS	50%
6	YES	0mS	Standard	0mS	0uS	50%

Signal	**Assigned Source**
12v Power rails	Source 1
Mated	Source 2
All other connections	Source 3

Hot-Swap State:
Canister is in the 'pulled' state, waiting for a power up command to attach it.

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 ':' (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.
- 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

DEPRECATED COMMANDS – Provided for backwards compatibility, we strongly suggest you use the 'Signal' and 'Source' commands instead.

REGister:READ [0xAA]

Returns the value of the register with address [0xAA]. [0xAA] should be in hex format and preceded by the suffix "0x". e.g. "0x6D". The value is returned in the same form as the address.

REGister:DUMP [0xA1] [0xA2]

Returns the value of each register in a range, starting at the first register address, up to the second. [0xA1] and [0xA2] should be in hex format and preceded by the suffix "0x". Each data value will be returned on a new line.

REGister:WRITe [0xAA] [0xDD]

Writes the byte [0xDD] to register [0xAA], both [0xDD] and [0xAA] should be in hex format and preceded by the suffix "0x". The command returns "OK" or "FAIL".

SOURce:[1-6|ALL]:SETup [#1] [#2] [#3] [#4]

Sets up the source in a single command. All parameters are positive integer numbers:

#1 = Initial delay (mS)
[Limits: 0 to 127ms in steps of 1ms, 0 to 1270ms in steps of 10ms]
#2 = Bounce length (mS)
[Limits: 0 to 127ms in steps of 1ms, 0 to 1270ms in steps of 10ms]
#3 = Bounce Period (uS)
[Limits: 10 to 1270us in steps of 10us, 1000 to 127000us in steps of 1000us]
#4 = Duty Cycle (%)
[Limits: 0 to 100% in steps of 1%]

SOURce:[1-6|ALL]:DELAY [#ms]
SOURce:[1-6|ALL]:DELAY?

Sets the initial delay of a source in mS. The delay is entered as a integer number with no units. E.g. “Source:1:delay 300”.

#1 = Initial delay (mS)
[Limits: 0 to 127ms in steps of 1ms, 0 to 1270ms in steps of 10ms]

SOURce:[1-6|ALL]:BOUNce:SETup [#1] [#2] [#3]

Sets up the bounce parameters in a single command. All parameters are positive integer numbers:

#1 = Bounce length (mS)
[Limits: 0 to 127ms in steps of 1ms, 0 to 1270ms in steps of 10ms]
#2 = Bounce Period (uS)
[Limits: 10 to 1270us in steps of 10us, 1000 to 127000us in steps of 1000us]
#3 = Duty Cycle (%)
[Limits: 0 to 100% in steps of 1%]

SOURce:[1-6|ALL]:BOUNce:LENgth [#ms]
SOURce:[1-6|ALL]:BOUNce:LENgth?

Sets the length of the pin bounce in mS. The delay is entered as a decimal number with no units. E.g. “Sour:2:boun:len 50”.

#1 = Bounce length (mS)
[Limits: 0 to 127ms in steps of 1ms, 0 to 1270ms in steps of 10ms]

SOURce:[1-6|ALL]:BOUNce:PERiod [#us]
SOURce:[1-6|ALL]:BOUNce:PERiod?

Sets the bounce period of the pin bounce in uS. The value is entered as a decimal number with no units. E.g. "Sour:6:boun:period 300".

#1 = Bounce Period (uS)

[Limits: 10 to 1270us in steps of 10us, 1000 to 127000us in steps of 1000us]

SOURce:[1-6|ALL]:BOUNce:DUTY [#%]
SOURce:[1-6|ALL]:BOUNce:DUTY?

Sets the duty cycle of the pin bounce as a %. The value is entered as a decimal number with no units. E.g. "source:3:bounce:duty 50".

#1 = Duty Cycle (%)

[Limits: 0 to 100% in steps of 1%]

SOURce:[1-6|ALL]:BOUNce:CLEAR

Removes any pin bounce from the source and sets all bounce settings to default values. See "Default Startup State" for details for the default settings.

SOURce:[1-6|ALL]:STATE [ON|OFF]
SOURce:[1-6|ALL]:STATE?

Sets or returns the enable state of the source. Any signals assigned to a disabled (off) source will immediately be disconnected and vice versa. If a source state is changed, all signals assigned to it will change at exactly the same time (if a change is required).

SIGnal:[SIG_NAME]:SETup [#num]
SIGnal:[SIG_NAME]:SOURce [#num]

Sets a given signal to a numbered timing source (0-7). SIGNAL_NAME is one of the items in the 'Signal Names' Appendix at the end of this manual.

SIGnal:[SIG_NAME]:SOURce?

Returns the source number that the signal is assigned to.

RUN:POWer [UP|DOWN]

Initiates a plug or pull operation (legacy name used to preserve compatibility between Torridon modules). This is done by changing the HOT_SWAP bit, register 0x00 bit 0. This is the master control for all switches on the card. The same action can be performed by writing this bit directly.

The command will fail if you order a power up when the module is already in the connected state and vice-versa as the action cannot be performed.

The “OK” response will be returned as soon as the hot-swap event has begun. If your timing sequence is very long you may have to poll the BUSY bit in register 0 to check when it has completed.

Legacy Commands

The following legacy commands are still supported from previous firmware versions to ensure backward compatibility:

voltage [12vinA|12voutA|12vinB|12voutB]

Returns the voltage on the specified rail in mV. Vin refers to the upstream or host side of the card, and Vout refers to the switched, canister side.

power [up|down]

Powers the canister up or down. This is done by changing the HOT_SWAP bit, register 0x00 bit 0. This is the master control for all switches on the card. The same action can be performed by writing this bit directly.

The command will fail if you order a power up when the canister is already in the plugged state and vice-versa as the action cannot be performed.

The “OK” response will be returned as soon as the hot-swap event has begun. If your timing sequence is very long you may have to poll the BUSY bit to check when it has completed.

write [0xAA] [0xDD]

Writes the byte [0xDD] to register [0xAA], both [0xDD] and [0xAA] should be in hex format and preceded by the suffix “0x”. The command returns “OK” or “FAIL”.

read [0xAA]

Returns the value of the register with address [0xAA]. [0xAA] should be in hex format and preceded by the suffix “0x”. e.g. “0x6D”. The value is returned in the same form as the address.

read [0xA1] to [0xA2]

Returns the value of each register in a range, starting at the first register address, up to the second. [0xA1] and [0xA2] should be in hex format and preceded by the suffix “0x”. Each data value will be returned on a new line.

Control Register Map

Address	Name	Description
0x00	Global Control	HOT_SWAP control
0x01	Glitch Control	Glitch Logic Control
0x02	RESERVED	Leave as 0x00
0x03	RESERVED	Leave as 0x00
0x04	S1 & S2 Control	Turn on / off / Pin Bounce
0x05	S3 & S4 Control	Turn on / off / Pin Bounce
0x06	S5 & S6 Control	Turn on / off / Pin Bounce
0x07	S1 Initial Delay	Delay 0 to 1.27s
0x08	S1 Oscillation Period	Period 0 to 127mS
0x09	S1 Oscillation Time	0 to 1.27s
0x0A	S1 Osc. Duty Cycle	1 to 99%
0x17..0x0B	S1 Custom Pattern	104 bit pattern
0x18	S2 Initial Delay	Delay 0 to 1.27s
0x19	S2 Oscillation Period	Period 0 to 127mS
0x1A	S2 Oscillation Time	0 to 1.27s
0x1B	S2 Osc. Duty Cycle	1 to 99%
0x28..0x1C	S2 Custom Pattern	104 bit pattern
0x29	S3 Initial Delay	Delay 0 to 1.27s
0x2A	S3 Oscillation Period	Period 0 to 127mS
0x2B	S3 Oscillation Time	0 to 1.27s
0x2C	S3 Osc. Duty Cycle	1 to 99%
0x39..0x2D	S3 Custom Pattern	104 bit pattern
0x3A	S4 Initial Delay	Delay 0 to 1.27s
0x3B	S4 Oscillation Period	Period 0 to 127mS
0x3C	S4 Oscillation Time	0 to 1.27s
0x3D	S4 Osc. Duty Cycle	0 to 100%
0x4A..0x3E	S4 Custom Pattern	104 bit pattern
0x4B	S5 Initial Delay	Delay 0 to 1.27s
0x4C	S5 Oscillation Period	Period 0 to 127mS
0x4D	S5 Oscillation Time	0 to 1.27s
0x4E	S5 Osc. Duty Cycle	1 to 99%
0x5B..0x4F	S5 Custom Pattern	104 bit pattern
0x5C	S6 Initial Delay	Delay 0 to 1.27s
0x5D	S6 Oscillation Period	Period 0 to 127mS
0x5E	S6 Oscillation Time	0 to 1.27s
0x5F	S6 Osc. Duty Cycle	1 to 99%
0x6C..0x60	S6 Custom Pattern	104 bit pattern
0xD0..0x6D	Signal Assignment Registers	Control nibble for each signal
0xF9	Glitch Length	Set the length of a Glitch
0xFA	Glitch Cycle	Set the cycle time for repeated glitches
0xFB	Glitch PRBS	Set the PRBS on percentage
0xFD	Version Register 3	FPGA 3 firmware version
0xFE	Version Register 2	FPGA 2 firmware version
0xFF	Version Register 1	FPGA 1 firmware version

Register Definitions

0x00 - Global Control

7	6	5	4
RESERVED	RESERVED	RESERVED	RESERVED

3	2	1	0
RESERVED	RESERVED	BUSY	HOT_SWAP

Name	Description
BUSY	This bit is cleared when the card is in a steady state, i.e. all delays/shorts/pin bounce activates are completed
HOT-SWAP	Changing this bit starts a hot-swap sequence based on the current timing parameters

0x01 - Glitch Control

7	6	5	4
RESERVED	RESERVED	RESERVED	RESERVED

3	2	1	0
RESERVED	GLITCH_PRBS	GLITCH_CYCLE	GLITCH_TRIGGER

GLITCH_PRBS	Selects PRBS glitch mode. When set, this bit overrides GLITCH_CYCLE
GLITCH_CYCLE	Selects Glitch cycle mode when set, glitch once mode when clear
GLITCH_TRIGGER	Set this signal to start a single glitch, or a glitch cycle, clear this signal to stop a glitch cycle

0x04 – S1 & S2 Control

7	6	5	4
0	0	S2_PIN_BOUNCE_MODE	S2_ENABLE

3	2	1	0

0	0	S1_PIN_BOUNCE_MODE	S1_ENABLE

Name	Description
S2_PIN_BOUNCE_MODE	When set, custom bounce pattern is enabled on S2
S2_ENABLE	When set, S2 is enabled, subject to HOT_SWAP state and rail bounce settings
S1_PIN_BOUNCE_MODE	When set, custom bounce pattern is enabled on S1
S1_ENABLE	When set, S1 is enabled, subject to HOT_SWAP state and rail bounce settings

0x05 – S3 & S4 Control

7	6	5	4
0	0	S4_PIN_BOUNCE_MODE	S4_ENABLE

3	2	1	0
0	0	S3_PIN_BOUNCE_MODE	S3_ENABLE

Name	Description
S4_PIN_BOUNCE_MODE	When set, custom bounce pattern is enabled on S4
S4_ENABLE	When set, S4 is enabled, subject to HOT_SWAP and rail bounce settings
S3_PIN_BOUNCE_MODE	When set, custom bounce pattern is enabled on S3
S3_ENABLE	When set, S3 is enabled, subject to HOT_SWAP state and rail bounce settings

0x06 – S5 & S6 Control

7	6	5	4
0	0	S6_PIN_BOUNCE_MODE	S6_ENABLE

3	2	1	0
0	0	S5_PIN_BOUNCE_MODE	S5_ENABLE

Name	Description
S6_PIN_BOUNCE_MODE	When set, custom bounce pattern is enabled on S6
S6_ENABLE	When set, S6 is enabled, subject to HOT_SWAP state and rail bounce settings
S5_PIN_BOUNCE_MODE	When set, custom bounce pattern is enabled on S5
S5_ENABLE	When set, S5 is enabled, subject to HOT_SWAP state and rail bounce settings

Source Registers

Each source is setup by a block of 17 byte wide registers. Below is the register map for a generic source. The list of registers in the title indicates the actual address of the byte in each of the 6 timed sources.

Source Delay [0x07, 0x18, 0x29, 0x3A, 0x4B, 0x5C]

7	6..0
DELAY_MULTIPLIER	DELAY

Name	Description
DELAY_MULTIPLIER	When 0, Delay Multiplier is 1mS When 1, Delay Multiplier is 10mS
DELAY	The Total delay between HOT_SWAP being set or the start of a power cycle event and the source going active T_{DELAY} = DELAY x DELAY_MULTIPLIER i.e. 00000010 = 2mS, 10001001 = 90mS

Source Bounce Period [0x08, 0x19, 0x2A, 0x3B, 0x4C, 0x5D]

7	6..0
BOUNCE_PERIOD_MULTIPLIER	BOUNCE_PERIOD

Name	Description
BOUNCE_PERIOD_MULTIPLIER	When 0, Delay Multiplier is 10uS When 1, Delay Multiplier is 1mS
BONUCE_PERIOD	The Period of the bounce frequency when pin-bounce is enabled Period = BOURCE_PERIOD x BOUNCE_PERIOD_MULTIPLIER i.e. 00000010 = 20uS, 10001001 = 9mS

Source Bounce Length [0x09, 0x1A, 0x2B, 0x3C, 0x4D, 0x5E]

7	6..0
BOUNCE_LENGTH_MULTIPLIER	BOUNCE_LENGTH

Name	Description
BOUNCE_LENGTH_MULTIPLIER	When 0, Delay Multiplier is 1mS When 1, Delay Multiplier is 10mS
BOUNCE_LENGTH	The duration of the pin bounce period when pin-bounce is enabled. T_{length} = BOUNCE_LENGTH x BOUNCE_LENGTH_MULTIPLIER i.e. 00000010 = 2mS, 10001001 = 90mS

Source Bounce Duty Cycle [0x0A, 0x1B, 0x2C, 0x3D, 0x4E, 0x5F]

7..0
BOUNCE_DUTY_CYCLE

Name	Description
BOUNCE_DUTY_CYCLE	The Duty Cycle of the bounce frequency, expressed as an (on time) percentage Values 0 – 100 are valid

Source Custom Pattern

[0x0B-0x17, 0x1C-0x28, 0x2D-0x39, 0x3E-0x4A, 0x4F-0x5B, 0x60-0x6C]

A 104 bit pattern that can be set by the user for a custom pattern to be used for pin bounce.

Offset	Bits
C	103..95
B	95..87
A	87..79
9	79..72
8	71..64
7	63..56
6	55..34
5	47..33
4	39..32
3	31..17
2	23..16
1	15..8
0	7..0

Name	Description
CUSTOM_PATTERN	The 104 bit custom pattern is held in 13 sequential registers. Bit 7, of the register with offset 0 is the first bit of the pattern.

Signal Registers

Two switched signals are controlled by each signal byte, one in each nibble. The 4 bits of the nibble stores a single number that describes which source the signal should be following.

Each nibble assigns the named signal to one of the six control sources, off, or on with HOT_SWAP:

Nibble Value	Assigned Control Source
0	OFF
1	Follow Source S1
2	Follow Source S2
3	Follow Source S3
4	Follow Source S4
5	Follow Source S5
6	Follow Source S6
7	Changes with HOT_SWAP state

Signal Assignment Registers

Each signal is assigned to a control source by setting a nibble (4 bytes) in the register map. The following table shows the location of each signal's control nibble.

Register	High Nibble	Low Nibble
0x6D	12V_A	12V_B
0x6E	STANDBY_PWR	POWER_OFF_L
0x6F	SLOT_ID	MATED_L
0x70	ENCLOSURE_INTR	AC_GOOD_L
0x71	PS1_ALERT_L	PS1_PRES_L
0x72	PS2_ALERT_L	PS2_PRES_L
0x73	PS2_LED_L	PS1_LED_L
0x74	TWI_BUS2_RST_L	TWI_BUS1_RST_L
0x75	SCL2	SDA2
0x76	SCL1	SDA1
0x77	SCL0	SDA0
0x78	ENC_DEF_HP_1	CARD_IO_TEST_L
0x79	ENC_DEF_HP_3	ENC_DEF_HP_2
0x7A	ENC_DEF_LP_1	ENC_DEF_HP_4
0x7B	ENC_DEF_LP_3	ENC_DEF_LP_2
0x7C	ENC_DEF_LP_5	ENC_DEF_LP_4
0x7D	ENC_DEF_LP_7	ENC_DEF_LP_6
0x7E	SGPIO_SCK	SGPIO_SLI
0x7F	SGPIO_SDI	SGPIO_SDO
0x80	LS1_BA	LS1_AB

Register	High Nibble	Low Nibble
0x81	LS2_BA	LS2_AB
0x82	LS3_BA	LS3_AB
0x83	LS4_BA	LS4_AB
0x84	LS5_BA	LS5_AB
0x85	LS6_BA	LS6_AB
0x86	LS7_BA	LS7_AB
0x87	LS8_BA	LS8_AB
0x88	DRIVE_2_FAULT_L	DRIVE_1_FAULT_L
0x89	DRIVE_4_FAULT_L	DRIVE_3_FAULT_L
0x8A	DRIVE_6_FAULT_L	DRIVE_5_FAULT_L
0x8B	DRIVE_8_FAULT_L	DRIVE_7_FAULT_L
0x8C	DRIVE_10_FAULT_L	DRIVE_9_FAULT_L
0x8D	DRIVE_12_FAULT_L	DRIVE_11_FAULT_L
0x8E	DRIVE_14_FAULT_L	DRIVE_13_FAULT_L
0x8F	DRIVE_16_FAULT_L	DRIVE_15_FAULT_L
0x90	DRIVE_18_FAULT_L	DRIVE_17_FAULT_L
0x91	DRIVE_20_FAULT_L	DRIVE_19_FAULT_L
0x92	DRIVE_22_FAULT_L	DRIVE_21_FAULT_L
0x93	DRIVE_24_FAULT_L	DRIVE_23_FAULT_L
0x94	DRIVE_26_FAULT_L	DRIVE_25_FAULT_L
0x95	DRIVE_28_FAULT_L	DRIVE_27_FAULT_L
0x96	DRIVE_30_FAULT_L	DRIVE_29_FAULT_L
0x97	DRIVE_32_FAULT_L	DRIVE_31_FAULT_L
0x98	DRIVE_34_FAULT_L	DRIVE_33_FAULT_L
0x99	DRIVE_36_FAULT_L	DRIVE_35_FAULT_L
0x9A	DRIVE_38_FAULT_L	DRIVE_37_FAULT_L
0x9B	DRIVE_40_FAULT_L	DRIVE_39_FAULT_L
0x9C	DRIVE_42_FAULT_L	DRIVE_41_FAULT_L
0x9D	DRIVE_44_FAULT_L	DRIVE_43_FAULT_L
0x9E	DRIVE_46_FAULT_L	DRIVE_45_FAULT_L
0x9F	DRIVE_48_FAULT_L	DRIVE_47_FAULT_L
0xA0	DRIVE_2_GPO_L	DRIVE_1_GPO_L
0xA1	DRIVE_4_GPO_L	DRIVE_3_GPO_L
0xA2	DRIVE_6_GPO_L	DRIVE_5_GPO_L
0xA3	DRIVE_8_GPO_L	DRIVE_7_GPO_L
0xA4	DRIVE_10_GPO_L	DRIVE_9_GPO_L
0xA5	DRIVE_12_GPO_L	DRIVE_11_GPO_L
0xA6	DRIVE_14_GPO_L	DRIVE_13_GPO_L
0xA7	DRIVE_16_GPO_L	DRIVE_15_GPO_L
0xA8	DRIVE_18_GPO_L	DRIVE_17_GPO_L
0xA9	DRIVE_20_GPO_L	DRIVE_19_GPO_L
0xAA	DRIVE_22_GPO_L	DRIVE_21_GPO_L
0xAB	DRIVE_24_GPO_L	DRIVE_23_GPO_L
0xAC	DRIVE_26_GPO_L	DRIVE_25_GPO_L
0xAD	DRIVE_28_GPO_L	DRIVE_27_GPO_L
0xAE	DRIVE_30_GPO_L	DRIVE_29_GPO_L
0xAF	DRIVE_32_GPO_L	DRIVE_31_GPO_L
0xB0	DRIVE_34_GPO_L	DRIVE_33_GPO_L
0xB1	DRIVE_36_GPO_L	DRIVE_35_GPO_L
0xB2	DRIVE_38_GPO_L	DRIVE_37_GPO_L

Register	High Nibble	Low Nibble
0xB3	DRIVE_40_GPO_L	DRIVE_39_GPO_L
0xB4	DRIVE_42_GPO_L	DRIVE_41_GPO_L
0xB5	DRIVE_44_GPO_L	DRIVE_43_GPO_L
0xB6	DRIVE_46_GPO_L	DRIVE_45_GPO_L
0xB7	DRIVE_48_GPO_L	DRIVE_47_GPO_L
0xB8	DRIVE_2_INPL_L	DRIVE_1_INPL_L
0xB9	DRIVE_4_INPL_L	DRIVE_3_INPL_L
0xBA	DRIVE_6_INPL_L	DRIVE_5_INPL_L
0xBB	DRIVE_8_INPL_L	DRIVE_7_INPL_L
0xBC	DRIVE_10_INPL_L	DRIVE_9_INPL_L
0xBD	DRIVE_12_INPL_L	DRIVE_11_INPL_L
0xBE	DRIVE_14_INPL_L	DRIVE_13_INPL_L
0xBF	DRIVE_16_INPL_L	DRIVE_15_INPL_L
0xC0	DRIVE_18_INPL_L	DRIVE_17_INPL_L
0xC1	DRIVE_20_INPL_L	DRIVE_19_INPL_L
0xC2	DRIVE_22_INPL_L	DRIVE_21_INPL_L
0xC3	DRIVE_24_INPL_L	DRIVE_23_INPL_L
0xC4	DRIVE_26_INPL_L	DRIVE_25_INPL_L
0xC5	DRIVE_28_INPL_L	DRIVE_27_INPL_L
0xC6	DRIVE_30_INPL_L	DRIVE_29_INPL_L
0xC7	DRIVE_32_INPL_L	DRIVE_31_INPL_L
0xC8	DRIVE_34_INPL_L	DRIVE_33_INPL_L
0xC9	DRIVE_36_INPL_L	DRIVE_35_INPL_L
0xCA	DRIVE_38_INPL_L	DRIVE_37_INPL_L
0xCB	DRIVE_40_INPL_L	DRIVE_39_INPL_L
0xCC	DRIVE_42_INPL_L	DRIVE_41_INPL_L
0xCD	DRIVE_44_INPL_L	DRIVE_43_INPL_L
0xCE	DRIVE_46_INPL_L	DRIVE_45_INPL_L
0xCF	DRIVE_48_INPL_L	DRIVE_47_INPL_L
0xD0	-	SPECIAL1

0xF9 - Glitch Control

7	6	5	4	3	2	1	0
GLITCH_MULTIPLIER			GLITCH_LENGTH				

Name	Description
GLITCH_LENGTH	Set the number of steps to glitch for
GLITCH_MULTIPLIER	Change step size, 0 = 50nS, 1 = 500ns, 2= 5us,3=50us,4=500us,5=5ms,6=50ms,7=500ms

0xFA - Glitch Control

7	6	5	4	3	2	1	0
GLITCH_CYCLE_MULTIPLIER	GLITCH_CYCLE						

GLITCH_CYCLE	Number of steps to stay off for
GLITCH_CYCLE_MULTIPLIER	Multiply GLITCH_CYCLE by 10 when set

0xFB – Glitch PRBS Control

7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	GLITCH_PRBS_DUTY		

Name	Description
GLITCH_PRBS_DUTY	Set the “on percentage” of the PRBS signal 0 = 1 in 256 steps is glitched 1 = 1 in 128 steps is glitched 2 = 1 in 64 steps is glitched 3 = 1 in 32 steps is glitched 4 = 1 in 16 steps is glitched 5 = 1 in 8 steps is glitched 6 = 1 in 4 steps is glitched 7 = 1 in 2 steps is glitched

Other Registers

0xFD – CPLD 3 Version Register

7..4	3..0
MAJOR REVISION	MINOR REVISION

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

0xFE – CPLD 2 Version Register

7..4	3..0
MAJOR REVISION	MINOR REVISION

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

0xFF – CPLD 1 Version Register

7..4	3..0
MAJOR REVISION	MINOR REVISION

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

Appendix 1 - Signal Names

The following signal names are used to specify a single. These may be used in commands that take a parameter “SIGNAL_NAME”.

Signals

12V_B,
12V_A,
POWER_OFF_L,
STANDBY_PWR,
MATED_L,
SLOT_ID,
AC_GOOD_L,
ENCLOSURE_INTR,
PS1_PRES_L,
PS1_ALERT_L,
PS2_PRES_L,
PS2_ALERT_L,
PS1_LED_L,
PS2_LED_L,
TWI_BUS1_RST_L,
TWI_BUS2_RST_L,
SDA2,
SCL2,
SDA1,
SCL1,
SDA0,
SCL0,
CARD_IO_TEST_L,
ENC_DEF_HP_**[1-4]**,
ENC_DEF_LP_**[1-7]**,
SGPIO_SLI,
SGPIO_SCK,
SGPIO_SDO,
SGPIO_SDI,
LS1_AB,
LS1_BA,
LS2_AB,
LS2_BA,
LS3_AB,
LS3_BA,
LS4_AB,
LS4_BA,
LS5_AB,
LS5_BA,
LS6_AB,

LS6_BA,
LS7_AB,
LS7_BA,
LS8_AB,
LS8_BA,
DRIVE_**[1-48]**_FAULT_L,
DRIVE_**[1-48]**_GPO_L,
DRIVE_**[1-48]**_INPL_L,
SPECIAL1,   (Proprietary mated signal for some vendors)