

DPhyGenCtl User's Manual

Table of Contents

1	<i>Overview</i>	1
1.1	Differences Between the P344 and P338	2
2	<i>Setup & Installation</i>	4
2.1	Quick-Start Summary	4
2.2	Installing DPhyGenCtl Software	4
2.3	Connecting the DPhy Generator	5
2.4	DPhy Generator Back-Panel Connector	5
3	<i>Common Command Data and Element Types</i>	7
3.1	Bytes	7
3.2	Integers	7
3.3	Byte or Integer Lists	7
3.4	DPhy LP States	7
4	<i>Operational Concepts</i>	9
4.1	Pattern Generator (PG)	9
4.2	DPhy Bus Behavior	9
4.3	Clock Lane Behavior	10
4.3.1	Non-Video-mode Clock Behavior	10
4.3.2	Video-mode Clock Behavior	11
4.3.3	File Command Clock Behavior	11
5	<i>Using DPhyGenCtl</i>	12
5.1	GUI Overview	12
5.2	Generic List Dialog	13
5.3	Connecting to the DPhy Generator	14
5.4	Instrument Configuration	15
5.4.1	Main Window Configuration Controls	15
5.4.2	Instrument Configuration Dialog	16
5.5	DPhy Timing Configuration	19
5.6	Commands	20
5.6.1	Defining a New Command	20
5.6.2	Naming a Command	21
5.6.3	Editing an Existing (Non-Macro) Command	22
5.6.4	Managing Commands (renaming, ordering, deleting)	22
5.6.5	Saving/Restoring Command Configurations	22
5.6.6	Special Command Types	22
5.6.6.1	Video-mode Commands	22

5.6.6.2	Display Stream Compression (DSC) Video	22
5.6.6.3	Macros	23
5.6.6.4	Phy Commands	23
5.6.6.5	File Command.....	23
5.6.6.6	PG File Command	23
5.6.6.7	Script Commands.....	24
5.6.6.8	Variable Argument Commands	24
5.6.6.9	Long Data File Format Commands	25
5.6.6.10	DCS Write Memory Commands.....	25
5.6.6.11	LP Delay	29
5.6.6.12	Delay To Pos.....	29
5.6.6.13	Mark Zero Pos	29
5.6.6.14	Wait Ext Event.....	29
5.6.6.15	Assert Trigger	30
5.6.6.16	Cmd Insertion Point	30
5.7	Video-mode Commands	30
5.7.1	Overview.....	30
5.7.2	Video Command Definition.....	31
5.7.2.1	File Name Syntax.....	31
5.7.2.2	Video Error Syntax	35
5.7.2.3	Image Viewing Dialog.....	35
5.7.3	Frame Timing Dialog.....	35
5.7.3.1	Frame Timing Configurations	37
5.7.3.2	Frame Parameter Controls	37
5.7.3.3	Frame Timing Controls.....	38
5.7.3.4	CSI Frame Construction Controls.....	38
5.7.3.5	DSI Frame Construction Controls	38
5.7.3.6	Common Frame Construction Controls.....	39
5.7.3.7	Stereoscopic Frame Controls	40
5.7.4	Video Frame Construction Details.....	40
5.7.4.1	Stereoscopic Frame Construction	41
5.7.4.2	Command Insertion.....	42
5.7.4.3	Video Memory Requirements.....	42
5.7.5	DSC Compression.....	42
5.7.5.1	DSC Configuration Dialog	43
5.7.5.2	Sending Compressed Frames in Video Mode.....	44
5.7.5.3	Sending Compressed Frames in Command Mode.....	45
5.7.5.4	Sending PPS Information.....	45
5.7.5.5	Compressing/Uncompressing Image Files	45
5.7.5.6	Viewing DSC Images	46
5.8	Macros.....	46
5.8.1	Defining a New Macro.....	46
5.8.2	Editing an Existing Macro	48
5.8.3	Copying a Macro.....	48
5.8.4	Inserting a Component Command	48
5.8.5	Replacing a Component Command	48

5.8.6	Editing a Component Command.....	49
5.8.7	Allowed Component Commands.....	49
5.8.8	Video-Mode Commands In Macros.....	50
5.8.8.1	“Enable Video Mode In Macros” Enabled	50
5.8.8.2	“Enable Video Mode In Macros” Not Enabled	50
5.8.9	Looping In Macros.....	51
5.8.10	HS Component Command Behavior	52
5.8.11	Additional Notes on Macro Behavior	52
5.9	Phy Commands	52
5.9.1	Clock On / Clock Off.....	53
5.9.2	ULPS.....	53
5.9.3	BTA.....	53
5.9.4	Escape Command.....	54
5.9.5	Skew Calibration.....	54
5.9.6	TGR Data Sequence.....	55
5.9.7	TGR Bit Sequence	55
5.9.8	PRBS9 Sequence	55
5.9.9	PRBS11 Sequence	56
5.9.10	PRBS18 Sequence	56
5.10	File Command	56
5.10.1	File Command File Syntax	58
5.10.2	Lane Group Arguments.....	60
5.10.3	LP_STATES Component Command.....	61
5.10.4	HS Burst Requirements.....	62
5.10.5	File Command Examples.....	62
5.11	PG File Command.....	63
5.12	Command Buttons	64
5.13	Operation.....	65
5.13.1	Sending a Command.....	65
5.13.1.1	Stop PG.....	66
5.13.1.2	Restart PG.....	66
5.13.2	Status.....	66
5.13.2.1	Contention Detection	66
5.13.2.2	DUT Response.....	67
5.13.3	Command Insertion.....	68
5.13.4	Controlling TrigOut	69
5.13.5	External Event Triggering.....	70
5.13.5.1	Controlling When Command Output Begins.....	70
5.13.5.2	Controlling Frame Advance During Video Mode	70
5.13.5.3	Controlling Macro Component Command Output	70
5.13.5.4	Triggering the External Event Signal Via Software	71
5.13.6	Wait Events.....	71
5.13.7	Causing Packet Errors.....	72
5.13.8	Script Recording	72

5.13.9	Updating Firmware	73
5.14	Customizing GUI Colors	74
5.15	GUI Options	75
5.16	Keyboard Shortcuts	76
5.17	Menus	76
6	<i>DPhyGenCtl RPC (Remote Procedure Calls)</i>	80

Contacting **The Moving Pixel Company**

Phone +1.503.626.9663 US Pacific Time Zone

Fax +1.503.626.9653 US Pacific Time Zone

Address **The Moving Pixel Company**
4905 SW Griffith Drive, Suite 106
Beaverton, Oregon 97005 USA

Email information@movingpixel.com

Web site <http://www.movingpixel.com>

1 Overview

The DPhyGenCtl application is the controlling software for the DPhy Generator (P344) made by **The *Moving Pixel Company*** (TMPC). Using this instrument, the user can generate CSI and DSI protocol and pattern stimulus on a MIPI DPhy bus for receiver testing, *notably supporting the DPhy 1.2 specification*. This document describes the use and operation of DPhyGenCtl and the corresponding behavior of the DPhy Generator.

For those familiar with earlier TMPC MIPI products, the DPhy Generator instrument is very similar in functionality to the P331/P332/P338 DPhy probe family. Unlike prior DPhy solutions that required use of a general-purpose pattern generator (PG3A), the DPhy Generator is a stand-alone instrument containing an internal pattern generator. Thus, it is the only custom hardware required for DPhy generation.

The DPhy Generator is connected via USB to a host computer that runs the DPhyGenCtl software. It has the following capabilities:

- Connects via either USB2 or USB3, allowing for fast program download times.
- Contains internal pattern generator with 2 GB of program memory
- Supports one to four DPhy lanes, supporting frequencies up to 4.5 Gb/s
- Lane outputs are via SMA connectors
- Provides up to ~4.5 ns of independent lane skew below 3.2 Gb/s. Provides ~320 ps of skew above 3.2 Gb/s.
- Provides real-time, per-lane, high and low HS voltage adjustments and common high and low LP voltage adjustments.
- Supports configuration and adjustment of DPhy bus timing.
- Provides 4 KB receive buffer for DUT response capture of LP data.
- Supports lane 0 LP contention detection.
- Implements arbitrary logical-to-physical lane output mapping.
- Provides a configurable Trig Out signal, which can be asserted via MIPI stream and/or software control.

The DPhyGenCtl application is a Windows application that embodies knowledge of MIPI CSI, DSI, and DPhy protocols to build programs for the DPhy Generator. Based on its predecessor software, PGRemoteForP338, it provides the following functions:

- Comprehensive video support:
 - Support for 3D stereoscopic frame construction (DSI 1.2).
 - Purchase option for encoding and sending of DSC video frames (DSI 1.2).
 - Common source input image file formats (jpg, png, tiff, gif), used for both video-mode and Write Memory commands.
 - Automatic resizing of input images (in software) to fit display or camera dimensions.

- Basic test pattern generation via naming syntax (syntax dialog provided).
- Convenient image preview function in GUI.
- Automated CSI/DSI video-mode frame generation based on user frame timing.
- Automatic partitioning of single Write Memory command into multiple Write Memory command sequence.
- Video-mode frames can be added to macros.
- Video-mode frames can be constructed with a single-bit error at a given line and bit position.
- Generic File command support:
 - Uses text file description to describe mixed low-level LP/HS transitions and packet definition.
 - Allows nearly-arbitrary data lane signal generation for conformance testing.
 - Provides higher-level embedded commands for easy command definition, including HS burst entry, HS burst exit, clock on, clock off sequences. Also, automatic ECC and CRC generation.
 - Supports nested files for reuse of common definitions.
- Support for low-level Phy testing:
 - Low-level test HS burst sequences using user-defined or PRBS data.
 - User configuration of bus timings, e.g. HSPrepare, HSZero, HSTrail, HSExit, etc.
 - New flexible bus timing specification in component units of ns, UI, and TLPX, allowing for frequency agile configurations.
- Powerful and easy-to-use GUI controls for command manipulation:
 - Simple definition, naming, and sending of commands, including video-mode commands.
 - Push-button interface for assigning and organizing commands so they are available for single-click sending.
 - Macro definition for building and complex command sequences.
 - Provides named frame timing configurations and DPhy timing configurations.
 - Provides dialogs to reorder/delete/sort named command, frame timing, and DPhy timing drop-down lists.
- Automation Support:
 - Script command allows text file usage for: configuration, command/macro/program definition and output.
 - Remote-control capability via .NET DLL.

1.1 Differences Between the P344 and P338

For convenience, this section summarizes the major differences and improvements between the P338 / PGRemoteForP338 and the P344 / DPhyGenCtl solutions.

In general, usage, structure, and behavior are very similar between the two solutions. Anyone familiar with using PGRemoteForP338 will have no trouble understanding how to use DPhyGenCtl. For the most part, the capabilities of the P344 solution is a superset of the capabilities of the P338 solution, although the two are not always directly compatible. In particular, configuration file formats and File command syntax are not

interchangeable. Also, while RPC command support and scripting syntax are nearly identical, there are exceptions.

Table 1 and Table 2 summarize the major differences in capabilities and characteristics of the two solutions.

Table 1 – P338 versus P344

<u>Capability</u>	<u>P338</u>	<u>P344</u>
Maximum Bit Rate	1.6 Gbps	2.6 Gbps
Maximum Data Lane Count	4 or 8 (option)	4
Additional HW Required	PG3A	None
Host Communication	USB2 @ 30 MB/s	USB3 @ 200 MB/s
Memory Capacity	256 MB (in PG3A)	2 GB
Memory Usage (maximum HS data per data lane)	32 MB	Max 1.5 GB (1 lane), Max 1.0 GB (2 lanes), Max 500 MB (3, 4 lanes)

Table 2 – PGRemoteForP338 versus DPhyGenCtl

<u>Characteristic</u>	<u>PGRemoteForP338</u>	<u>DPhyGenCtl</u>
DSI version support	1.1	1.2
Stereoscopic frame support?	No	Yes
DSC compressed video support?	No	Yes (purchase option)
Video-mode frames allowed in macros?	No	Yes
Video-mode frame bit error control?	No	Yes
TGR and PRBS sequences for low-level Phy support?	No	Yes
Named DPhy Timing Configs?	No	Yes
Named Frame Timing Configs?	No	Yes
Video Decimation?	Yes	No
Dual-interface support?	Yes	No
Clock On/Off Model	Cumbersome	Simplified
File command support	Lacking mixed LP/HS signaling	Comprehensive, including mixed LP/HS signaling
DPhy Timing Specification	Units of ns only	Frequency-aware with units of ns, UI, and TLPX.

2 Setup & Installation

2.1 Quick-Start Summary

The following summarizes the overall procedure for installing and configuring the TMPC software and hardware for DPhy testing:

- 1) Install DPhyGenCtl Software (section 2.2)
- 2) Set up, connect, and power on the DPhy Generator (section 2.3)
- 3) Launch DPhyGenCtl and connect to the DPhy Generator instrument (section 5.3)
- 4) Select the MIPI standard to use: CSI or DSI (Standard menu)
- 5) Configure instrument operational parameters (section 5.4).
- 6) Configure video timing parameters for your device (section 5.7.3)
- 7) Configure DPhy bus timing and protocol definitions.(section 5.5)
- 8) Define commands and assign them to buttons, if desired (section 5.6.1).

Once these steps are completed, the system is ready to start testing.

2.2 Installing DPhyGenCtl Software

To install DPhyGenCtl, simply execute the setup.exe file on the installation CD and step through the setup windows. As part of the installation, a USB driver is installed called CyUSB3.sys. This driver is required for the software to recognize the DPhy Generator instrument when connected to USB.

Periodically, new versions of DPhyGenCtl are available on the Moving Pixel Company web-site. These upgrade versions do NOT install the CyUSB3.sys driver and assume the install machine already has the driver installed. Thus, new machines always must first have the initial setup installation run before any upgrades. Note that previous versions are automatically uninstalled when running installation of an upgrade version of DPhyGenCtl.

The DPhyGenCtl application is installed by default in :

c:\Program Files\TMPC\DPhyGenCtl

The following shortcuts are provided in the Start Menu, under DPhyGenCtl

- **DPhyGenCtl.exe**: a link to the application executable
- DPhyGenCtlUsersManual_x_x.pdf: a link to this manual
- **Uninstall DPhyGenCtl**: a link to an uninstall script

The DPhyGenCtl application runs under the WinXP, Win7, and Win8 operating systems and uses the Microsoft .NET Framework 4.0 platform.

Once the software has been installed for the first time, the license file named “TMPCLicense.txt” (obtained from The Moving Pixel Company) should be copied to the application directory. The license file enables DPhy operation (and perhaps other options in the future) based on the serial number of your DPhy Generator. Note that the software may be installed on, and the license file copied to, multiple host machines.

If a license file (or the correct license string in the license file) is not present, the software can still run in offline mode. This mode allows the user to interact with the GUI, define and save commands, and develop RPC applications without connecting to real hardware.

2.3 Connecting the DPhy Generator

The DPhy Generator is straightforward to set up, significantly more so than its predecessor probes without the need of the PG3A pattern generator connections.

- Connect the 24V power supply adaptor to the instrument.
- Connect a USB cable from the host machine to the instrument. Note: a USB3 cable is required to connect to the DPhy Generator. However, either a USB3 or USB2 port on the host computer can be used for connection (although USB2 operates at reduced speed). A USB3 port is distinguished from USB2 by its blue color.
- Connect your DUT to DPhy lane outputs via SMA cables.
- Power on the instrument

After power-on, assuming the CyUSB3 driver is installed, the instrument appears in the device manager of the host machine as “P344 DPhy Generator”. At this point launching DPhyGenCtl should show the instrument serial number in its Connect dialog, ready for use.

2.4 DPhy Generator Back-Panel Connector

For an overall description of the P344 DPhy Generator hardware, please see the P344 datasheet. However, for reference in this document, the back-panel connector pin-out is shown in Figure 1.

Currently, only some of the pins are used on the back-panel connector and note that no back-panel connections are *required* for normal operation, unless external events are to be used for program control.

The pins currently used are:

- **Evt0** – indicates when command insertion occurs
- **Evt1** – indicates lane 0 contention (OR of all the four contention flags)
- **GPI0** – external event input for program flow control
- **GPO0** – TrigOut from host or stream

Remaining signals on the connector are unused.

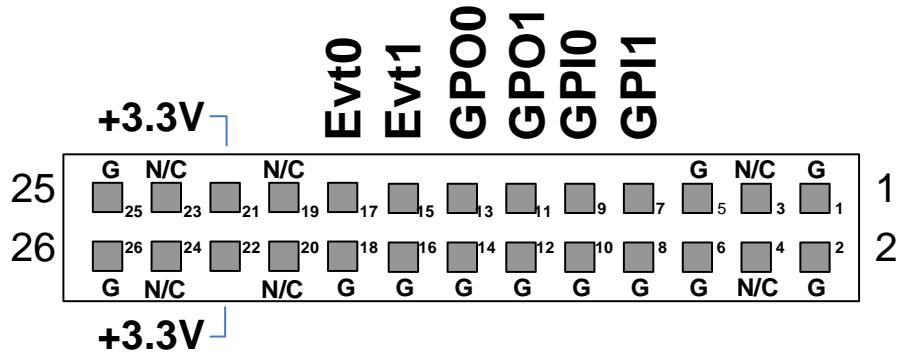


Figure 1— P344 Back-Panel Connector

3 Common Command Data and Element Types

Various data and element types are used in the DPhyGenCtl application, whether as parameter arguments to GUI commands, script commands, or component commands of a text file used in the File Command. This section describes common data and element types and how to specify them.

3.1 Bytes

Bytes are 8-bit integer values ranging from 0 to 255 and can be specified in decimal or hexadecimal. Unless otherwise specified for a particular command, the default radix for command byte arguments is decimal. If a command indicates it uses “hex byte” arguments, then the default radix is hexadecimal.

To explicitly define the radix of a data byte, append either a ‘d’ for decimal or ‘h’ for hexadecimal. For example, the following list describes a 4 byte sequence from 16-19 inclusive:

```
16 11h 18d 13h
```

3.2 Integers

Integers are 32-bit signed integer values and can be specified in decimal or hexadecimal (in which case the value is always non-negative). Unless otherwise specified, the default radix for integer arguments is decimal.

To explicitly define the radix of an integer, append either a ‘d’ for decimal or ‘h’ for hexadecimal. For example, the following list describes a 4 integer sequence from 256-259 inclusive:

```
256 101h 257d 103h
```

3.3 Byte or Integer Lists

Some commands allow parameters to be byte or integer lists. These lists are specified by one or more byte or integer values respectively, separated by spaces (see previous two sections for examples). The argument name for this value is usually Params[], which usually requires byte values unless otherwise indicated by the specific command.

3.4 DPhy LP States

DPhy LP states represent low-power voltage levels for the two wires of a lane and have integer values ranging from 0-3. The wire-order for LP states is (LP+, LP-): 0 (LP00), 1 (LP01), 2 (LP10) and 3 (LP11).

Some commands encode the LP state for all four data lanes plus the clock as a 10-bit hex value. The argument name for this value is LPValue[9:0]. For this argument, each pair of bits in LPValue contains the LP state for a corresponding lane as follows:

- LPValue[1:0] – LP State for lane 0

- LPValue[3:2] – LP State for lane 1
- LPValue[5:4] – LP State for lane 2
- LPValue[7:6] – LP State for lane 3
- LPValue[9:8] – LP State for clock lane

Note that if the clock is running when a command that sets the LP state is sent, the clock lane remains running and the LPValue bits for the clock are ignored.

4 Operational Concepts

4.1 Pattern Generator (PG)

As previously mentioned, prior MIPI solutions offered by The Moving Pixel Company have used a general-purpose pattern generator (PG3A) and a MIPI-specific probe (P331, P332, P338) for DPhy stream generation. The P344 DPhy Generator streamlines this architecture by incorporating the pattern generation function into the instrument. Thus, the PG3A is no longer required.

However, this document and the DPhyGenCtl software still refer to the PG as a distinct (internal) component of the DPhy Generator. For example, the status pane in the main window shows whether the PG is running or idle, and buttons labeled “Stop PG” and “Restart PG” are used to control the DPhy Generator's internal pattern generator engine for transmission.

4.2 DPhy Bus Behavior

Once the DPhy Generator has been powered on and DPhyGenCtl is connected, DPhy lanes including the clock lane are driven to LP11. This is the default state of the instrument whenever the PG is idle. Note that this behavior is different than previous MIPI probes (P331, P332, P338), which allowed the clock to continue to run when the PG was idle. See the next section for further discussion of clock lane behavior and control.

When commands are sent using DPhyGenCtl, one or more lanes become active, either transitioning to HS mode or signaling in LP mode, depending on the command and the current “DT Mode” setting. Generally, the “Lane Cnt” setting determines which lanes become active, though it is possible for the low-level commands to send LP data on any number of lanes, independent of “Lane Cnt”.

The Options->Loop Commands menu option determines whether commands are sent once or looped on the bus indefinitely. If this option is not checked, the command is sent and then the bus returns to the LP11 state, with the PG status indicating that it is idle. Otherwise, PG status indicates it is Running while the command is looping.

Note that some lower-level Phy commands inherently loop the PG, regardless of the “Loop Commands” setting. These include “Enter Ultra Low-Power Mode” and the TGR and PRBS test sequence commands.

Looping commands can be stopped by clicking the “Stop PG” button, returning the PG to idle state and the DPhy bus to LP11. Also, unless command insertion is enabled (Options->Enable Command Insertion), sending another command while a command is

looping causes the application to ask whether you want to stop the PG to allow the sending of the command.¹

4.3 Clock Lane Behavior

This section describes the behavior of the clock lane, which can be turned on/off according to the DPhy protocol. The only direct requirement of the clock lane is that it runs during HS bursts, but otherwise users have differing requirements on whether the clock should remain on during LP traffic.

One overarching behavior enforced by software is that the clock is always off (i.e. in the LP11 state) when the PG is idle. Thus, the clock is always off when sending a new command and it is always turned off (if necessary) when a command completes.

4.3.1 Non-Video-mode Clock Behavior

For non-video-mode commands, clock lane behavior is further governed by two options (Options menu):

- “Turn Clk On When Sending Non-Video Cmds”
- “Turn Clk Off After HS Burst (Non-Video)”

plus some other behavioral rules:

- The clock is automatically turned on (if necessary) whenever an HS command is sent
- The clock is turned on (if not already on) when a “Clock On” command is encountered in a macro.
- The clock is turned off (if not already off) when a “Clock Off” command is encountered in a macro.

These rules and options define the clock lane behavior for all non-video commands.²

The option “Turn Clk On When Sending Non-Video Cmds” causes the clock to turn on before a non-video command is sent. This option really only determines how the clock behaves when an LP command is sent (or a macro that starts with an LP command), as the clock automatically turns on if the command is an HS command anyway (or is a macro that starts with an HS command).

Similarly, the option “Turn Clk Off After HS Burst (Non-Video)” causes the clock to turn off after an HS command is sent. This option is only relevant for HS commands in macros (as the clock automatically turns off at the end of an HS command anyway).

¹ Unless the GUI option “Suppress confirmation message when stopping PG” is set, in which case the looping command is automatically stopped.

² Note that the File command is the one exception to this rule as it has its own clock on/off component commands.

Generally, the user desires one of two behaviors:

- 1) The clock turns on and remains on throughout all commands, including LP commands. In this case the user should check “Turn Clk On When Sending Non-Video Cmds” and uncheck “Turn Clk Off After HS Burst (Non-Video)”.
- 2) The clock is turned off for LP commands. In this case the user should uncheck “Turn Clk On When Sending Non-Video Cmds” and check “Turn Clk Off After HS Burst (Non-Video)”.

Other distinct behaviors can be controlled using the “Clock On” and “Clock Off” commands in macros.

4.3.2 Video-mode Clock Behavior

In video-mode, clock behavior is simply controlled through a single option in the Frame Timing dialog: “Turn Clk On/Off During LP11 Blanking”. When this option is checked, the clock turns off at the start of LP11 blanking and back on again at the end of LP11 blanking. If the option is unchecked, the clock remains on throughout video playback.

Note that video-mode frames added to macros keep this same behavior, even as non-video clock behavior rules and options are applied to other commands in the macro.

4.3.3 File Command Clock Behavior

The File command uses a user provided text file to control LP and HS signaling and send arbitrary packet data. During this time, the clock lane is also controlled by the user via component commands (CLK_ON and CLK_OFF) in the text file.

5 Using DPhyGenCtl

The DPhyGenCtl software is the configuration and control application for the DPhy Generator. Using its GUI interface, users can interact with the instrument, configuring commands to send on the bus, monitor status and review DUT response data. In addition, the application supports a remote control interface (called RPC) that supports controlling nearly all capabilities accessible through the GUI using a remote, user program.

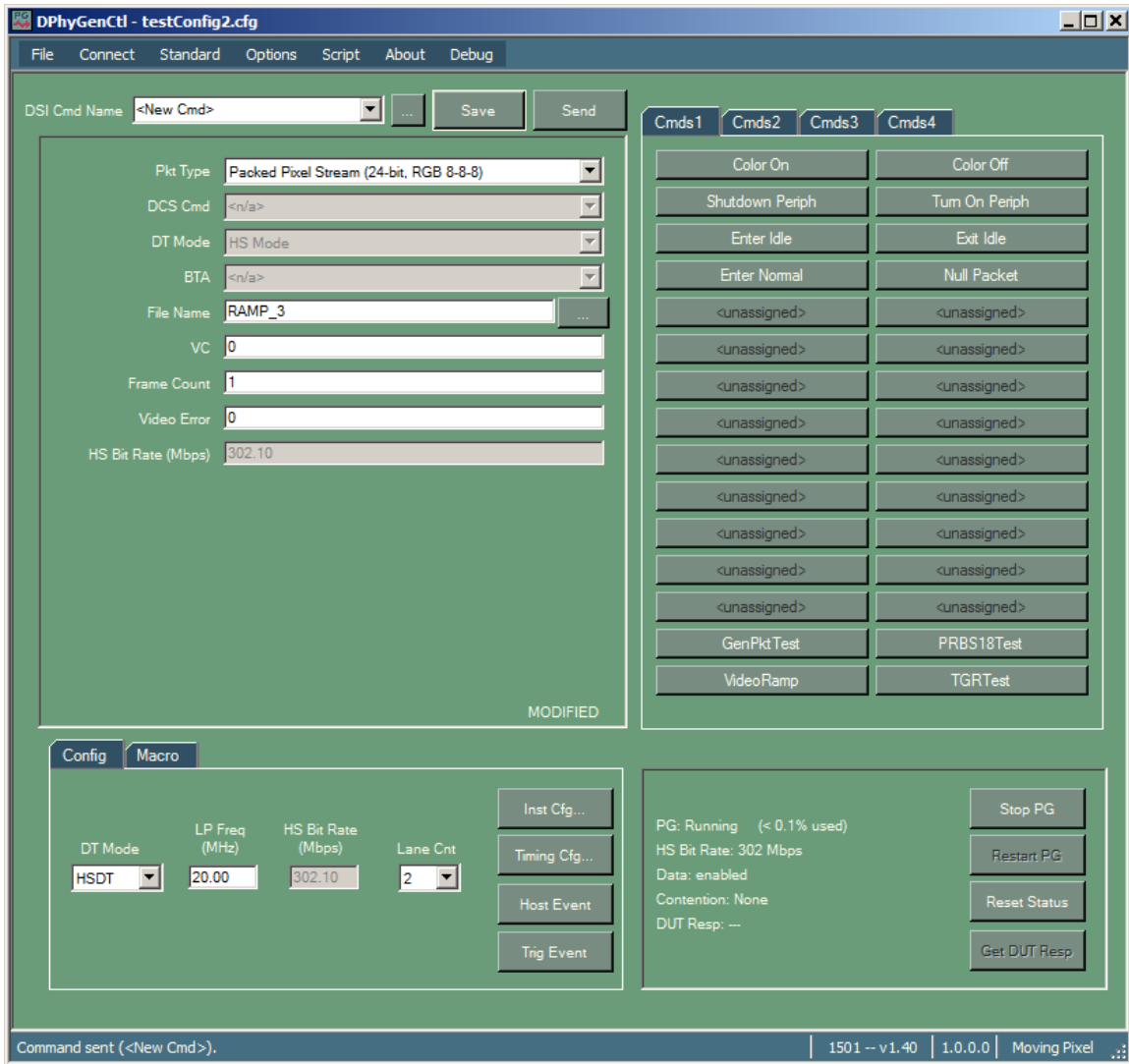


Figure 2 – DPhyGenCtl Main Window

5.1 GUI Overview

Most of the interaction between the user and DUT occurs in the DPhyGenCtl main window (see Figure 2). It has several distinct sections.

- The left pane of the main window allows the user to define commands and their arguments, naming them to allow for later recall and sending to the DPhy Generator, as well as assigning them to command buttons.
- The right pane of the main window consists of four tab pages with 30 command buttons each that can each be individually associated with a named command, allowing single-click sending of commands to the DPhy Generator.
- The bottom left pane of the main window provides controls for configuring parameters of instrument transmission, video timing definitions, physical characteristics of the DPhy bus, and trigger events.
- The bottom right pane of the main window shows instrument status and provides operational controls.
- The status bar at the bottom of the main window displays informational and error messages and indicates the current instrument connection and software/firmware versions.

The following sections describe more details of DPhyGenCtl operation.

5.2 *Generic List Dialog*

The generic List dialog is used to organize named item lists associated with a drop-down control. Named items are listed in a central list-box, with controls on the side to manipulate items in the list. The List dialog supports the follow functions:

Table 3 – List Dialog Functions

Function	Method
Clear the list	Click on the Clear button
Sort the list alphabetically	Check the Sort checkbox
Reset the list to its default	Click the Defaults button
Delete items in the list	Select one or more items in the list and click the Delete button
Move item(s) up in the list	Select one or more items in the list and click the Up button
Move item(s) down in the list	Select one or more items in the list and click to Down button

After viewing and/or modifying the list, clicking OK closes the dialog and records any changes. Clicking Cancel closes the dialog and discards any changes to the list.

A List dialog is associated with three controls in DPhyGenCtl:

- Cmd Name control in the main window
- Frame Timing Configuration control in the Frame Timing dialog
- DPhy Timing Configuration control in the DPhy Timing Configuration dialog

Immediately to the right of each of these controls is a button labeled with an ellipsis (“...”). Clicking this button brings up the List dialog, initialized with element names from the associated control. Figure 3 shows an example Frame Timing List dialog.

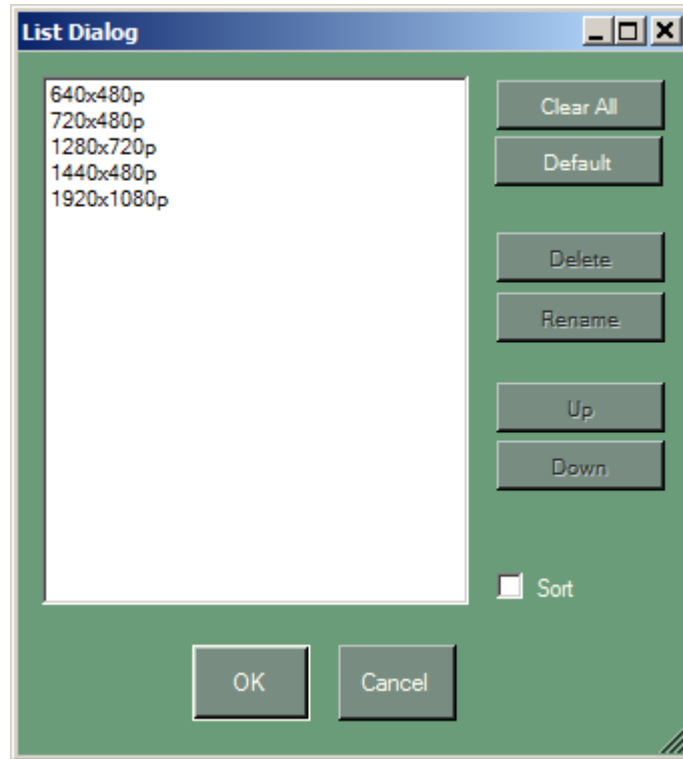


Figure 3 – Frame Timing List Dialog

5.3 Connecting to the DPhy Generator

Before any commands can be sent, the DPhyGenCtl application needs to connect to the DPhy Generator connected via USB to the host machine. The appropriate license file for the instrument must be located in the application directory for successful connection (see section 2.2).

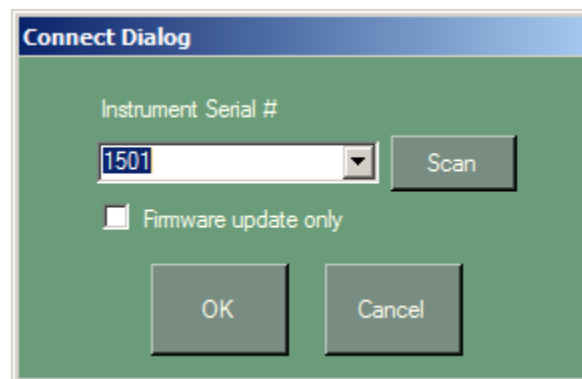


Figure 4 – Connection Dialog

When DPhyGenCtl is launched, the application automatically brings up a connection dialog (see Figure 4) with a drop-down control showing visible DPhy Generator instruments available for connection. An alternative way to bring up this dialog once the application is running is via the “Connect...” menu option in the Connect menu.

To connect to an instrument or change connection to a different instrument, select the serial number of the instrument you want to connect to and click OK. Select “Offline” to run in offline mode without an instrument connection. Clicking the Cancel button maintains the current connection state. Once the dialog is closed, the instrument serial number and firmware version is displayed in a bottom-right corner of the status pane to indicate the current connection. Otherwise, “Offline” is displayed.

If an instrument is not powered on and connected via USB before this dialog is launched, the Scan button can be used to show newly available instruments.

Also, if an instrument's firmware has been corrupted and does not boot properly, it is important to check “Firmware update only” when connecting to the instrument for the purpose of updating its firmware. This disables communication with the central FPGA, which may not respond properly if it is in a bad state. After connecting to the instrument in this way, please proceed directly to the Update Firmware dialog (Connect menu) without other main window control interactions.

When testing is complete, the "Disconnect" menu can be used to close the instrument and return to offline. The application does this automatically on closing so it is not required.

5.4 Instrument Configuration

There are two main sets of controls for instrument configuration. The main window contains the most important and commonly adjusted controls such as LP frequency, HS bit-rate, lane count, and DT Mode. Less used controls can be found in the Instrument Configuration Dialog.

5.4.1 Main Window Configuration Controls

Four controls in the “Config” tab at the bottom of the main window are used to describe principal DPhy configuration parameters. These controls are described below:

DT Mode: this combo box sets the default data transfer mode for commands whose DT Mode field is set to “Default”. Packets may be sent as either HS packets (HSDT mode) or LP packets (LPDT mode). Individual commands can override the default DT Mode setting by setting the DTMode field in their command arguments. Note: this mode only applies to non-video commands and does not affect video-mode operation or low-level DPhy commands such as escape commands as well as the PRBS/TGR test sequences.

HS Bit Rate: this control sets the lane bit rate of the DPhy bus for all commands *except* video-mode commands.³ The supported range for the HS Bit Rate setting of the DPhy Generator is 31.25 Mbps to 4500 Mbps.

When playing non-video-mode commands, changing this control setting does not immediately affect the bit rate. Instead it schedules a frequency reconfiguration for the next sent command. To see the current bit rate on the DPhy bus for a running program, look at the HS Bit Rate display in the status pane *not* the HS Bit Rate setting in the instrument configuration pane.

When playing video-mode commands, the HS Bit Rate control is made read-only and reflect the current video-mode bit rate. This is to eliminate possible confusion with the HS Bit Rate setting (which applies only to non-video commands).

LP Freq: this control sets the frequency of LP bits on the DPhy bus (i.e. determines T(LPX) in the DPhy standard). The supported range is 0.2 MHz to 30 MHz.

Lane Cnt: this control selects the number of DPhy lanes to use (1-4). HS packet data is demultiplexed onto DPhy lanes according to the DSI/CSI protocol specification. Unused lanes are held in the LP11 output state.

5.4.2 Instrument Configuration Dialog

Once connected, the DPhy Generator can be configured with parameters specific to the users testing needs via the Instrument Configuration dialog brought up from the main window via the "Inst Cfg..." button in the lower-left pane (Config tab) of the main window. Note that adjustments can be made even while a program is running

After entering desired settings on the dialog, click the OK button to accept them or Cancel to discard them. Since settings are updated as they are adjusted, Cancel in this case means setting control values back to their values before the dialog was opened. Clicking on the Defaults button sets all controls to their default values.

Figure 5 shows the Instrument Configuration dialog. Below is a description of its controls:

Force Test Pattern: checking this box causes a clock test HS bit pattern to be output on all DPhy lanes at the current HS Bit Rate setting.⁴ This mode is generally used while the instrument is connected to an oscilloscope to ensure the instrument is working properly and to visually adjust voltages and delays for calibration.

³ Video-mode commands follow the currently selected frame timing configuration, which sets the HS Bit Rate for the video command.

⁴ The HS Bit Rate used for the test pattern is the last configured bit rate, which may be different than the main window HS Bit Rate setting if it has been changed since the last command.

Common HS Voltage: this check box, when checked, disables the HS voltage controls for all lanes except for lane 0. Adjustments to data lane 0 HS voltages are automatically applied to all lanes. Otherwise, HS voltage settings can be independently adjusted.

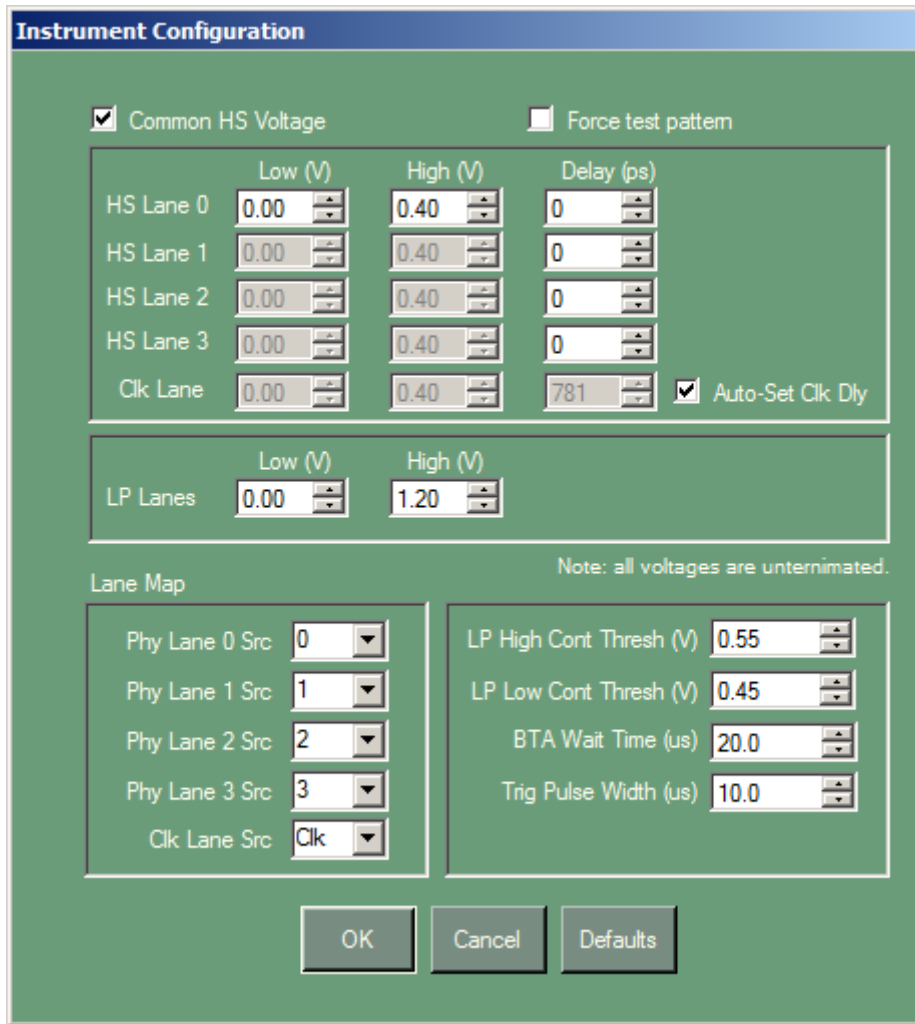


Figure 5 – Instrument Configuration Dialog

HS Low/High Voltage Controls: these controls set the *unterminated* HS voltage levels for DPhy lanes. When connecting to real hardware, HS voltages generally should be set to 0.0V (low) and 0.4V (high). Slave termination resistance reduces HS levels to the expected swing of 0.1 - 0.3V. HS voltage limits are -0.6V and 1.2V.

LP Low/High Voltage Controls: these controls set the LP voltage levels for all DPhy lanes. LP voltages generally should be set to 0.0V (low) and between 1.1V and 1.3V (high). LP voltage limits are -0.28V and 1.8V.

HS Delay Controls: these controls set the relative delays of each lane. Delay adjustments are in ps and have two limit maximums:

- Max delay is 4800 ps for bit rates less than 3 Gbps
- Max delay is 320 ps for bit rates greater than 3 Gbps

Be aware that when the bit rate is increased past 3 Gbps, existing delay values are automatically clipped to the new maximum delay (320 ps).

Auto-Set Clk Dly: this checkbox enables the automatic setting of clock delay to the ideal clock-to-data sampling delay for the current HS Bit Rate. When checked, the clock delay control is made read-only and reflects the automatic delay value. When unchecked, the clock delay control is user settable.

LP Contention Thresholds: these two controls determine the threshold voltage used for contention detection.

The LP Low contention threshold determines the voltage at which a LP Low fault is flagged (i.e. when the transmitter attempts to drive an LP0 and measures a voltage on the line greater than the threshold). This value is nominally 0.45V according to the DPhy specification.

The LP High contention threshold determines the voltage at which a LP High fault is flagged (i.e. when the transmitter attempts to drive an LP1 and measures a voltage on the line less than the threshold). This value is nominally 0.55V according to the DPhy specification.

BTA Wait Time: (not applicable for CSI) this control sets the time-out period after a BTA is sent to wait for a return BTA sequence from the DUT. Normally, if a return BTA is seen, the instrument acknowledges the BTA and continues its program. Similarly, if the time-out period expires, the instrument reasserts LP11 on lane 0 and continues its program.

Trig Pulse Width: set the trigger pulse width when a trigger pulse is sent on the TrigOut signal of the instrument, either through the “Assert Trigger” command in a program or the “Trig Event” button in the main window.

Phy Lane 0 Src: sets the logical source lane for physical lane 0. While normally, this is set to zero, this control allows logical lane 1, 2, 3, or Clk to be output on physical lane 0 as well. Note, however, that BTA handshaking, DUT response acquisition and contention detection is always associated with physical lane 0 regardless of logical mapping.

Phy Lane 1 Src: sets the logical source lane for physical lane 1. While normally, this is set to one, this control allows logical lane 0, 2, 3, or Clk to be output on physical lane 1 as well. One use of this control is to duplicate lane 0 traffic on lane 1 for separate scope monitoring during single-lane mode communication.

Phy Lane 2 Src: sets the logical source lane for physical lane 2. While normally, this is set to two, this control allows logical lane 0, 1, 3, or Clk to be output on physical lane 2 as well.

Phy Lane 3 Src: sets the logical source lane for physical lane 3. While normally, this is set to three, this control allows logical lane 0, 1, 2, or Clk to be output on physical lane 3 as well.

Clk Lane 3 Src: sets the logical source lane for the clock lane. While normally, this is set to Clk, this control allows logical lane 0, 1, 2, or 3 to be output on physical lane 3 as well.

Defaults: button to reset all controls to their defaults. The user is prompted via a message box to confirm the reset operation.

5.5 DPhy Timing Configuration

The DPhy Timing Configuration Dialog provides controls for the user to specify DPhy bus timings. Please see the DPhy specification for details of timing parameter definitions.

At the top of the dialog is a drop-down list of named timing configurations. A single configuration named “Default Timing” is pre-defined for the user, which sets reasonable, legal time constraints of each of the parameters.

Custom configurations can be defined by modifying any of the parameter fields. When a field is modified, the configuration name generally changes to “<custom>” and a text label “Modified” appears in the upper right corner of the dialog (though, if all fields match an existing named configuration, the matching configuration name is instead shown and the “Modified” label does not appear).

Click the Save button to save the custom configuration as a named configuration. The user is prompted for a configuration name, which is added to the drop down list control. Named timing configurations can be organized using the generic list dialog, brought up with the button labeled with an ellipsis (“...”) next to the drop-down configuration list control. See section 5.2 for more information about the generic list dialog.

Note that a configuration does not have to be named and saved to be used. Current parameter settings in the dialog are used to configure command timing regardless of whether part of a named configuration. In addition, current parameter settings are automatically saved and restored when the application is closed and reopened.

For each timing parameter, three columns of text boxes allow timing definition in units of ns, UI, TLPX. Some parameters have their TLPX or UI text boxes disabled for component settings that are not supported. A column of text boxes labeled “Actual” shows the resulting value that can be implemented, computed from the combination of the component values and quantized to a multiple of 2*UI.

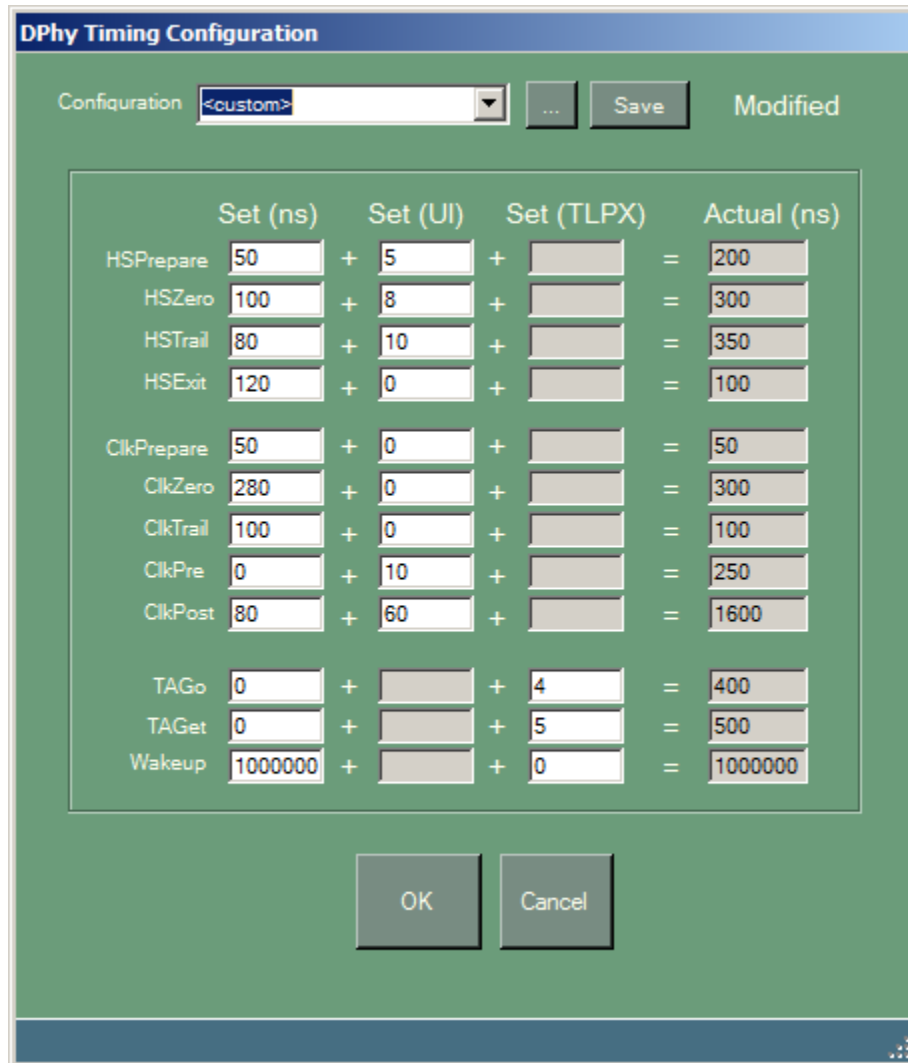


Figure 6 – DPhy Timing Configuration Dialog

5.6 Commands

5.6.1 Defining a New Command

Commands are defined using the left pane controls of the main window and are grouped based on the current standard according to category. Thus, before defining a command, the user should select the desired protocol via the Standard menu, currently either CSI or DSI. Then, to define a new command, perform the following steps:

- 1) Select the command type via the “Pkt Type” drop-down control. Note that once a command field is edited, the label in the “Cmd Name” control changes to "<New

- Cmd>" and "MODIFIED" appears below the command fields to prevent confusion with any existing command definition.
- 2) If the command type is a DCS command from the DSI command set (i.e. DCS Short Write, DCS Long Write, or DCS Read Request), the DCSCmdDesc drop-down control is enabled. Select the DCS command type from this control.
 - 3) Depending on the command selected, argument fields are labeled and enabled for the user to enter values. Certain arguments are pre-filled such as the virtual channel, DT Mode, and BTA. The user may change these pre-filled values if desired.
 - 4) Commands that support sending either as a LP or HS command initially have their DTMode argument set to "Default", which means the command is sent using the current default DT Mode setting (DT Mode control in the lower-left of the main window). Alternatively, the user can specify the DTMode for certain commands by changing the DTMode argument to either LPDT or HSDT.
 - 5) Setting BTA to "Yes" requests that a bus-turn-around sequence follow the sending of the command. This is usually used for read commands to allow the device under test to respond with data.
 - 6) Numeric fields are assumed to be decimal unless appended with 'h' (hexadecimal) or 'b' (binary). For example "67h" = 103 and "1101b" = 13.
 - 7) For most commands, derived fields of the command are displayed in read-only text boxes: DataID, WordCount, ECC, CRC. The ECC and CRC fields can be modified to cause packet errors by checking the checkbox associated with the field. The other read-only fields cannot be modified directly by the user but are computed according to the MIPI specification based on the command arguments.⁵ Note that special commands such as video commands, Phy commands, script commands, File commands, etc. do not fill in these fields because they do not represent a single MIPI protocol packet.

5.6.2 Naming a Command

Once command's argument fields are defined, the command may be named so that it can be assigned to a command button and saved in a configuration file. To name a command, click on the Save button to bring up a dialog to enter the command name. In the naming dialog, the command name field is initialized with the original command name present before editing. This allows the user the option to simply click OK to replace the original command. Otherwise, type in a new name to create a new command.

Named commands show up in the "Cmd Name" drop-down control in the main window. Selecting a named command in the list fills in the argument field controls with values associated with the command.

⁵ To modify read-only fields, the user can use the "Send Cmd To File" menu option (File menu). This outputs the command bytes to a text file that can subsequently be modified and resent using the File command.

5.6.3 Editing an Existing (Non-Macro) Command

An existing non-macro command definition can be edited as follows:

- Load the command field values into the edit controls either by right-clicking on a button having a command assignment and selecting "Edit Current Command" or selecting the command name from the "Cmd Name" drop-down control.
- Edit the command fields.
- Click the Save button.
- Click OK in the naming dialog.
- Click OK in the confirmation dialog to replace the existing command definition.⁶

Macro editing is described in Section 5.8.2.

5.6.4 Managing Commands (renaming, ordering, deleting)

The button labeled with an ellipsis ("...") immediately to the right of the "Cmd Name" drop-down control allows the user to manage the named command list, including renaming commands, reordering and sorting commands in the drop-down list, deleting commands, etc. Click this button to bring up a List dialog to perform these functions. See section 5.2 for more information about this dialog.

5.6.5 Saving/Restoring Command Configurations

A "default.cfg file" is provided with DPhyGenCtl installation that has a few example commands defined. Once you add new command definitions, you can save your command configuration, command set, and button assignments in a configuration file.

Note that command definitions and button assignments are *not automatically saved when the application is closed*. Use the "File->Save XXX.cfg As..." menu option and specify the name of the file to save to. Similarly, you can load a configuration file with the "File->Load..." menu option. The last configuration file saved before exit is restored when the application is restarted.

5.6.6 Special Command Types

5.6.6.1 Video-mode Commands

Video-mode commands are one of many possible video-mode command types that cause the application to build and send one or more video frames, which are sourced by user-provided image files (or special test pattern names). A video-mode frame consists of multiple packets and has specific structure defined by the current standard (CSI or DSI). Timing between frame structure elements is controlled by the current frame timing settings. Video-mode commands are described further in section 5.7.

5.6.6.2 Display Stream Compression (DSC) Video

DSI version 1.2 defines a new packet type: Compressed Pixel Stream. This packet type allows a compressed stream constructed according to the VESA Display Stream

⁶ The confirmation dialog can be disabled in the GUI Options dialog (Options->GUI Options).

Compression (DSC) standard to be sent. DPhyGenCtl provides a separate purchase option for extensive support for DSC-related functions, which includes the following⁷:

- Automated building of DSC streams from image files and test patterns for video-mode and Write Memory commands that use the compressed video format.
- Automated extraction of PPS data for DSC files, or automated construction of PPS data for image files when sending the Picture Parameter Set command.
- Support for the DPX and DSC file types in compression commands
- User decoding/encoding of DSC files to/from image files
- Image viewing of DSC files
- Write Memory partitioning and video-mode packetization option to send either one chunk per packet or possibly multiple chunks per packet.

Compressed video is described further in section 5.7.5.

5.6.6.3 *Macros*

A macro is a user-defined sequence of MIPI commands, sent as a single unit. Macros can be defined in the GUI, in a script file, or via a user program using RPC, then named and sent like any other command. Macro commands are described further in section 5.8

5.6.6.4 *Phy Commands*

Phy commands are special low-level or LP signaling commands. Some are defined by the DPhy specification such as BTA, ULPS, skew calibration sequence, clock on/off commands, etc. Other commands are not part of the DPhy specification but have been adopted from CPhy for bus testing such as TGR and PRBS sequences. Phy Commands are describe further in section 5.9.

5.6.6.5 *File Command*

The File Command is general-purpose, low-level command that uses a text file as input to define arbitrary LP and HS DPhy signaling sequences. Using special file command syntax, low-level LP and HS lane states and higher-level packet data bytes can be intermingled as desired to create user test sequences and error packets. Command syntax also supports useful features such as automatic ECC and CRC generation, HS burst signaling generation (e.g. HS entry/exit and sync), and packet data demultiplexing. The file command is described further in section 5.10.

5.6.6.6 *PG File Command*

The PG File command provides a method to send compiled programs that have been previously saved to a new binary file type called a PG file. This allows long compile-time programs to be built only once, saved to a PG file, then sent very quickly subsequently. This process is described further in section 5.11.

⁷ Without the DSC option, DPhyGenCtl does not provide these functions but still supports sending compressed video using binary files.

5.6.6.7 *Script Commands*

Script commands are RPC (remote procedure call) commands contained in a user-defined text file. RPC commands provide the function of almost every action provided in the GUI, such as command and macro definition, command button management, frame timing, DPhy timing, instrument configuration, sending of commands, etc.

Script files are sent using the “RPC Script” command, specifying or browsing for a file name in the “File Name” argument field. Please see section 6 and more generally the document “DPhyGenCtlRPC.pdf” for more information.

One new feature to DPhyGenCtl is the ability to embed script commands directly in the filename text. If the filename begins with a comment string “//” or script command symbol “#”, then the filename string is treated directly as a script. For multi-line scripts, editing can be achieved using the Script Edit Dialog, which is brought up instead of a file browsing dialog when the ellipsis button “...” is clicked next to the command filename (assuming the filename starts with a “//” or “#”).

The Script Edit Dialog is simply a dialog with a text box containing the script that the user can type in. A Clear button is provided to clear the text box. Otherwise, the user can edit the script and then click OK to accept the changes or Cancel to discard any changes.

5.6.6.8 *Variable Argument Commands*

Certain commands allow the user to specify a variable number of parameters through an argument text box named “Params[]”, specifically the following commands:

- 1) Generic Short Write Command (DSI)
- 2) Generic Read Request (DSI)
- 3) DCS Short Write (DSI)
- 4) DCS Read Request (DSI)
- 5) Custom DCS Command (DSI)
- 6) Custom Command
- 7) Custom Long Command

The differences between these commands are minimal. The Custom Command and Custom Long Command allow the user to specify an arbitrary DataID byte. The other commands implicitly fill in the Data Type and allow the user to specify a Virtual Channel argument. The DCS commands also allow the user to specify the DCS command byte. All allow a variable number of arbitrary arguments, computing the ECC, packet length (for long packets), and CRC (for long packets).

The actual packet constructed may have either long or short packet formatting, depending on the number of values parsed from the Params[] argument string. For non-DCS packet types, if Params[] contains 0, 1, or 2 bytes, then short packets are constructed. For DCS packet types, if Params[] contains 0 or 1 byte, then short packets are constructed (starting

with the DCSCmd byte). Otherwise, long packets are constructed and additional read-only fields for the packet length and CRC appear.

The Params[] argument string consists of byte values separated by white-space or commas. As with other argument fields, values are assumed to be decimal unless appended with 'h' (hexadecimal) or 'b' (binary). For example, "30 41h 10101b" is a legal parameter string.

5.6.6.9 Long Data File Format Commands

Certain non-video packet types (e.g. Long Pkt (Non-Image Data), Generic Long Write, DCS Long Write, etc) have a "File Name" argument. For these commands, with the exception of the DCS Write Memory commands, the file may be treated as a simple binary file or as an ASCII file with specific syntax. In either case, the file data is always sent as the payload of a single long packet.

A file is considered an ASCII file if it contains only legal ASCII characters and contains the line "# ASCII". ASCII files are imported using the following rules:

- Comment lines that start with "//" as well as blank lines can occur anywhere in the file.
- Data lines always must occur after the "# ASCII" marker line
- Data lines consist of one or more data bytes in hex format

For example, the following text is legal in an ASCII file:

```
// Example ASCII file syntax
# ASCII
20 5B 30 5D 20 3D 20 30 78 31
35 0D 0A 64 61 74 61 20 5B 31
5D 20 3D 20 30 78 31 31 0D 0A
```

For Write Memory commands, the file is treated as a binary or ASCII file only if it does not have a recognized image file extension (i.e. ".bmp", ".jpg"). Otherwise, the file is imported and processed as an image file. In addition, the Write Memory commands have special partitioning support in DPhyGenCtl, where a sequence of Write Memory commands is sent to implement the user request of a single Write Memory command. This additional support is offered because the Write Memory commands are often used to send video data to a frame buffer on the DUT. Please see the next section for more details about this support.

5.6.6.10 DCS Write Memory Commands

For Write Memory Start and Write Memory Continue commands, the "Write Len" argument specifies the number of bytes to send and the "File Offset" argument specifies the starting byte offset in the file or frame buffer⁸.

⁸ Note: "Write Len" can be set to 0 to send the contents of the entire file

Note that these parameters apply *after* the file has been imported, which for image files may include decoding, rescaling, and conversion to the output video format and for DSC files includes extracting the chunk data. Thus, the parameter name “File Offset” might be misleading when using image files (in this case, the parameter might be better called “Image Data Offset”. No processing occurs when using binary files.

For example, to send line 10 of an image to a 640 x 480 RGB121212 display:

- 1) Compute the line length as $3 * 12 * 640 / 8 = 2880$ bytes
- 2) Set Write Len = 2880
- 3) Set File Offset = $10 * 2880 = 28800$

DPhyGenCtl has additional features that enhance the behavior of sending Write Memory commands:

- **Image File Decoding** – If a file has extension “.bmp” or “.jpg” the image file is first imported before it is sent. When imported, an image file may first be rescaled if necessary to the output frame dimensions as described by the HActive and VActive fields of the current Frame Timing configuration. This only occurs if the option “Allow Image Rescaling” is enabled. Then, the image is converted to the format as specified in the “Image Decode Format” drop-down control.
- **Block Partitioning** – A single Write Memory command can be partitioned into a sequence of Write Memory commands. The user configures this behavior via “Configure Write Memory Commands...” in the Options menu.

Figure 7 shows the Write Memory Configuration dialog, which is brought up via the “Configure Write Memory Commands...” in the Options menu.

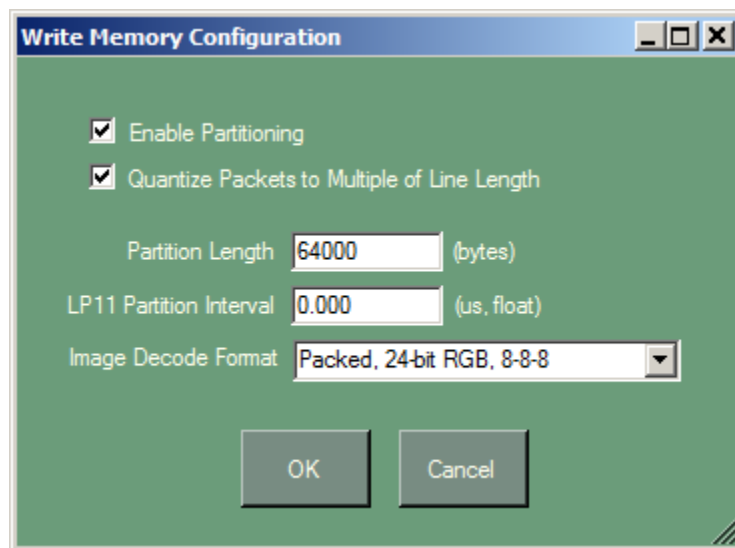


Figure 7 – Write Memory Configuration Dialog

The dialog has the following controls:

Enable Partitioning CheckBox – when checked, partitioning is enabled and Write Memory commands are decomposed into a sequence of Write Memory commands according to the parameters in the dialog. Otherwise, “Write Len” argument values greater than 64 KB (the maximum single packet payload length) return an error.

Quantize Packets to Multiple of Line Length – when checked, partition lengths are set to the greatest multiple of the image line length less than the “Partition Length” setting. The image line length is computed in one of two ways, depending on whether “Allow Image Rescaling” is enabled. If rescaling is enabled, then the image line length is computed using the HActive setting in the Frame Timing configuration and the pixel format for the selected “Image Decode Format”. Otherwise, the image line length is computed using the input line width times the bytes per pixel for the format specified in “Image Decode Format”.

Partition Length – indicates the maximum length to use for component Write Memory commands in the sequence.

LP11 Partition Interval – indicates the amount of time to spend in LP11 between each component Write Memory command in the sequence. Specify zero for no delay between component commands. In this case, if the “Send single packet per HS burst” option is not set and the command is sent in HS mode, the entire command sequence will be sent in a single HS burst (otherwise, each component Write Memory command will be sent in its own burst).

Image Decode Format – specifies the destination image format to decode files that have an extension of “.bmp” or “.jpg” (or also “.dpx” for compressed video).

5.6.6.10.1 Sending Compressed Video

If the “Image Decode Format” is set to “DSC Compressed” then data sent with the Write Memory command is assumed to be compressed video. Behavior is different depending on whether the DSC license option is installed:

With DSC License:

- Files with “.dsc” and “.dpx” extensions may be used
- Image files and test patterns can be used. If necessary (and enabled), the image is rescaled to the current output frame dimensions (as specified in the current Frame Timing configuration). Then, the image is compressed using the current DSC configuration settings.
- For image and DSC files, partitioning occurs (if enabled), ignoring the partition length. The partition length is set to the chunk size in the compressed stream or a multiple of the chunk size, based on the “Send One Chunk Per Packet” setting in the DSC configuration.

- For binary files, partitioning occurs (if enabled), according to the Partition Length setting. In this case, the user should set the Partition Length to the length of one or more chunks per the DSI standard.

Without DSC License:

- Only binary files may be used. Behavior is same as for binary files with license.

5.6.6.10.2 Partition Example 1

Consider a Write Memory Start command issued with the following arguments:

File Name = test.bmp
File Offset = 0
Write Len = 100000

and Write Memory configuration settings:

Enable Partitioning = true
Quantize Packets to Multiple of Line Length = false
Partition Length = 30000
LP11 Partition Interval = 1.5 us
Image Decode Format = Packed 24-bit RGB 8-8-8

Then, DPhyGenCtl sends the following packet sequence:

- 1) Write Memory Start with RGB888 image data bytes 0-29999
- 2) LP11 for 1.5 us
- 3) Write Memory Continue with RGB888 image data bytes 30000-59999
- 4) LP11 for 1.5 us
- 5) Write Memory Continue with RGB888 image data bytes 60000-89999
- 6) LP11 for 1.5 us
- 7) Write Memory Continue with RGB888 image data bytes 90000-99999
- 8) LP11 for 1.5 us

5.6.6.10.3 Partition Example 2

Consider a Write Memory Continue command issued with the following arguments:

File Name = test.bmp
File Offset = 0
Write Len = 60000

and Write Memory configuration settings:

Quantize Packets to Multiple of Line Length = true
Partition Length = 30000
LP11 Partition Interval = 0.0 us
Image Decode Format = Packed 24-bit RGB 8-8-8

HActive = 640 (frame timing dialog)

Then, DPhyGenCtl sends the following packet sequence:

- 1) Write Memory Continue with RGB888 image data bytes 0-28799
- 2) Write Memory Continue with RGB888 image data bytes 28800-57599
- 3) Write Memory Continue with RGB888 image data bytes 57600-59999

Note that the output image line length is 1920 bytes and that the first two partitions contain exactly 15 lines of image data because “Quantize packets to Multiple of Line Length” is set to true.

5.6.6.11 LP Delay

The “LP Delay” command is a low-level command that allows the user to assert a particular LP state on all data lanes of the DPhy bus for a specified period of time. This command is generally used in a macro sequence.

The “LP Delay” command has two arguments:

- **Delay (us)** – the length of time to hold the bus in the specified LP state. This value is floating point.
- **LPValue[9:0]** – value encoding the LP states for all four data lanes plus the clock lane as described in section 3.4.

Note that if the clock lane is in HS mode when the “LP Delay” command is sent, LPValue[9:8] controlling the clock lane LP state are ignored. Use the “Clock Off” command to first return the clock lane back to LP11 before sending the “LP Delay” if clock lane control is desired.

5.6.6.12 Delay To Pos

The “Delay To Pos” command is equivalent to “LP Delay” except that its Delay parameter is relative to a position in a macro called “Zero Pos”, defined by the user with the “Mark Zero Pos” command. This command is useful when defining video frames in macros, allowing the user to define relative times between packets (e.g. active video packets, line start, line end, etc). See the “LP Delay” command for a definition of command arguments.

5.6.6.13 Mark Zero Pos

The “Mark Zero Pos” command (no arguments) defines a “Zero Pos” location in a macro. Subsequent “Delay To Pos” commands use “Zero Pos” as a reference position/time in the macro to define the relative duration of a delay. Note that multiple “Mark Zero Pos” commands are allowed in the same macro.

5.6.6.14 Wait Ext Event

The “Wait Ext Event” command inserts an indefinitely looping “LP Delay” period in a macro stream, causing output to suspend in the specified LP state until an external event edge occurs.

5.6.6.15 Assert Trigger

The “Assert Trigger” command is equivalent to “LP Delay” except it has a third parameter (“Trig Cmd”) that specifies the specific trigger command to send, controlling the Trig Out signal of the DPhy Generator. The trigger command is implemented near the beginning of the “LP Delay” period, but exact timing depends on the output frequency and may vary relative to the period start with each trigger command.

Table 4 shows the trigger commands. Note that the trigger pulse width is defined in the Instrument Configuration dialog.

Table 4 – Assert Trigger Commands

Trigger Command Value	Description
1	Set Trig Out low
2	Set Trig Out high
3	Toggle Trig Out state
4	Pulse Trig Out

5.6.6.16 Cmd Insertion Point

The “Cmd Insertion Point” is equivalent to “LP Delay” except it additionally defines a location in the macro where another command can be inserted while the macro is looping. This feature is equivalent to command insertion during video-mode.

The first two parameters (Delay, LPValue[9:0]) are the same as for “LP Delay”. A third parameter, called Discard, is a flag that affects how the Delay period is affected by the length of the inserted command.

Table 5 – Cmd Insertion Point Discard Setting

Discard Value	Description
0	The inserted command is truly inserted into the program stream. Thus, if the Delay setting is 10 us and the inserted command requires 20 us, the ending time of the LP delay period would be 30 us.
1	The inserted command consumes the LP delay period as needed. If the inserted command does not complete before the end of the delay period, then subsequent stream data is held off until the inserted command is complete. Thus, in the prior example, the ending time of the LP delay period would be 20 us.

Please refer to section 5.13.3 for further details of command insertion.

5.7 Video-mode Commands

5.7.1 Overview

When a user sends a video-mode command (e.g. DSI Pixel Stream commands, CSI RGB, CSI Raw, etc) DPhyGenCtl automatically builds a packet stream representing one or

more video frames, having the appropriate timing and structure according to the DSI or CSI standard. The video sequence can be played once or looped continuously to provide ongoing real-time video for emulation of baseband DSI display streams or CSI camera streams.

All video-mode commands accept a “File Name” argument to specify an image reference containing the video frame data to send. To send a video sequence of more than one frame, “File Name” uses a special naming convention to allow DPhyGenCtl to derive the file names for subsequent frames in the sequence (see section 5.7.2).

Before sending a video-mode command, additional parameters of video-mode playback must be appropriately configured in the Frame Timing dialog (see section 5.7.3). These timing parameters, like other application settings, are saved and restored in an application configuration file when DPhyGenCtl is closed and restarted.

Once a video command has been defined and the timing configuration parameters have been set, sending the command initiates the sending of the video-mode stream. The internals of how video frames are constructed is described in section 5.7.4. Once video frame packets have been built and downloaded to the DPhy Generator instrument, video begins playing on the DPhy bus.

If the “Loop Commands” option (Options menu) is checked, video frames are played and looped indefinitely, stopping only when the “Stop PG” button at the bottom of the main window is pressed. When the DPhyGenCtl stops a looping video sequence, the current frame is completed before exiting and returning to the LP11 quiescent state.

5.7.2 Video Command Definition

All video-mode commands have the four arguments:

- 1) File Name – source image file or test pattern name
- 2) VC – virtual channel
- 3) Frame Count – number of frames to send
- 4) Video Error – control word encoding a data bit error position in the frame

The VC and “Frame Count” arguments are self-explanatory. However, the “File Name” and “Video Error” fields are described further in the next sections.

5.7.2.1 File Name Syntax

The “File Name” argument to video-mode (and Write Memory) commands may reference many source types and structures:

- **Test Pattern:** special naming conventions are provided to allow simple test patterns to be constructed and used as the input image source for video-mode commands.
- **Image File:** an image file with extension BMP or JPG

- **Video File:** common video files (AVI, MPEG2, MPEG4, WMV, FLV).⁹
- **DPX File:** an image file with extension DPX.¹⁰
- **DSC File:** a compressed image file with extension DSC.¹⁰
- **Binary File:** any file that doesn't fall in the above categories is treated as a binary file. In this case, rescaling and format conversion are not supported and the file data is expected to be in the proper format for the video-mode command being sent. In particular, the file should have at least the number of bytes required by the current timing configuration and video format. That is, $HActive * VActive * BytesPerPixel$.¹¹

When “Frame Count” is greater than one, thus requiring an image sequence as input, the file name may:

- 1) explicitly refer to an image sequence (i.e. video file)
- 2) implicitly refer to an image sequence (using special multi-file sequence naming)
- 3) reference only one image (in which case it replicated for the entire sequence)

5.7.2.1.1 Multi-File Sequence Naming

To play a sequence of multiple frames, the video-mode “File Name” argument may reference the first frame of the sequence and have the following format:

“<name><index>.<ext>”

where <name> is an arbitrary file name, <index> is an integer and <ext> is the file extension. DPhyGenCtl derives the file names of subsequent frames automatically by incrementing <index>.

For example: if “c:\frame1.bmp” is specified as the file name and the frame count is set to four, DPhyGenCtl builds a video sequence from the files:

- 1) c:\frame1.bmp
- 2) c:\frame2.bmp
- 3) c:\frame3.bmp
- 4) c:\frame4.bmp

Multi-file Sequence Notes:

- All single image reference types can be used in multi-file sequence naming.

⁹ If a video file is referenced in a Write Memory command, currently only the first frame is used.

¹⁰ DSC and DPX files are only supported when using the DSC compressed video format.

¹¹ The number of bytes per pixel for binary files when using the DSC compressed video format is derived from the configuration file in the current DSC configuration, which must have a BITS_PER_PIXEL assignment.

- If “File Name” is a video file, the first “Frame Count” frames are used from the file.
- If “File Name” does not have the multi-file naming syntax (or a test pattern name is used), the sequence is still built repeatedly using the single given “File Name” reference.

5.7.2.1.2 Stereoscopic File and Sequence Naming

Stereoscopic file and sequence naming can be accommodated with further conventions. If these conventions are not used, then standard naming conventions can be used to define a single-image stream from which left/right images are automatically created by extracting alternating pixels from each image in the sequence.

To provide stereoscopic images, the further convention is to append an ‘L’ and ‘R’ to the file name or file sequence and specify the left image file name in the command. Note this convention applies only to image files and binary files and not currently to video files.

For example, a stereoscopic video sequence can be specified with “File Name” set to “c:\frame1L.bmp” and uses the files:

- 1) c:\frame1L.bmp, c:\frame1R.bmp
- 2) c:\frame2L.bmp, c:\frame2R.bmp
- 3) c:\frame3L.bmp, c:\frame3R.bmp
- 4) c:\frame4L.bmp, c:\frame4R.bmp

5.7.2.1.3 Test Pattern Naming

Special file names can be used to generate test patterns rather than specifying an image file name. Simple solid color frames, basic ramps, and grids can be generated using special file name conventions.

In order to aid usage, press Ctl-t to bring up the Text Pattern Syntax dialog. Figure 8 shows this dialog which describes the naming convention used to generate different test patterns.

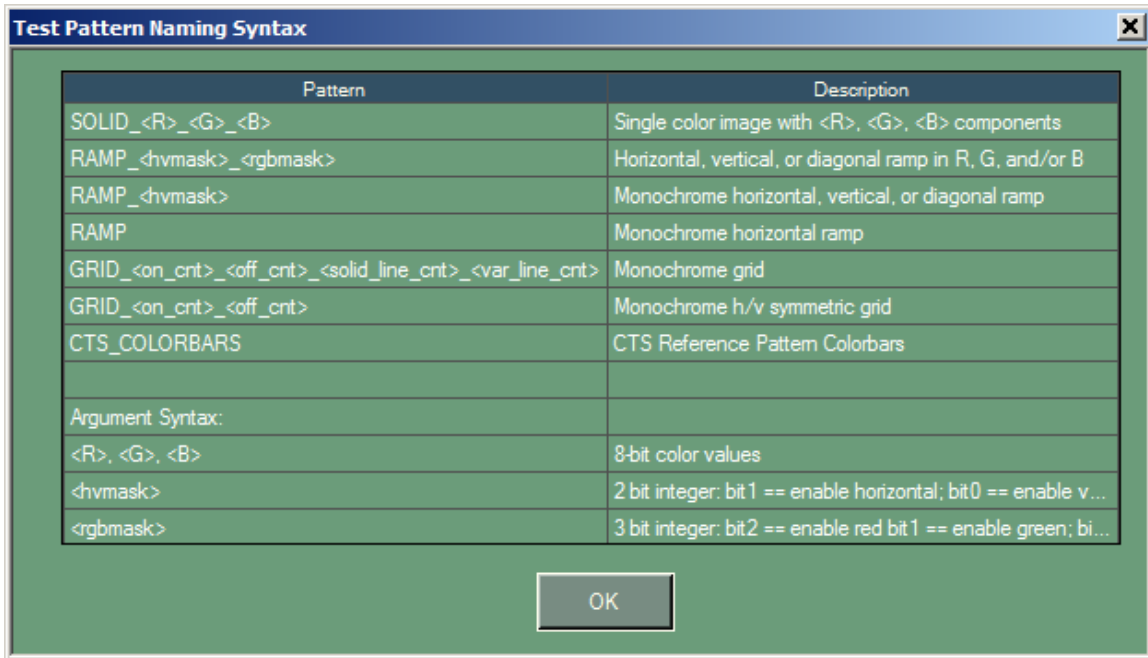


Figure 8 – Test Pattern Syntax Dialog

A single, solid color frame is generated with the syntax:

SOLID_<R>_<G>_.

For example, to generate a full-frame consisting of the single color with red = 100, green = 150, and blue = 200 (8-bit values), enter “solid_100_150_200” for the “File Name” argument of the video-mode or Write Memory command.

A horizontal, vertical, or diagonal ramp can be generated with the syntax:

RAMP_<hvmask>_<rgbmask>.

The <hvmask> value, if present, can be 1 (vertical), 2 (horizontal) or 3 (diagonal). The <rgbmask> value, if present, specifies which color planes are active. For example, RAMP_3_4 specifies a diagonal ramp in red-only and RAMP_2_3 specifies a horizontal ramp in both green and blue.

A monochrome grid pattern can be generated with the syntax:

GRID_<on_cnt>_<off_cnt>_<solid_line_cnt>_<var_line_cnt>

The grid pattern consists of two types of lines: solid lines and variable lines. Variable lines are defined by the user with <on_cnt> and <off_cnt>, basically defining the repeating pattern of “on” pixels followed by “off” pixels. Then the test pattern is formed

by a repeating pattern of solid lines for <solid_line_cnt> followed by variable lines for <var_line_cnt>.

For example, <on_cnt> = 1 and <off cnt> = 9 defines a variable line that has a single white pixel every 10 pixels. If <solid_line_cnt> = 1 and <var_line_cnt> = 9, then the naming syntax is GRID_1_9_1_9 and the pattern formed is a grid with single line thickness with 10 x 10 squares. GRID_5_15_5_15 defines a grid with 5-pixel line thickness with 20 x 20 squares.

Finally, the CTS Reference Pattern Colorbars can be generated using the test pattern name:

CTS_COLORBARS

This test pattern is described in the MIPI CTS for DPhy document in Appendix D.

5.7.2.2 Video Error Syntax

The “Video Error” argument allows the user to encode a bit position in the video frame(s) to force a bit error in the frame data during output. The argument is encoded as follows:

- Video Error bits [31:16] indicates the error line in the image (ones-based)
- Video Error bits [15:0] indicates the error bit position in the line (zero-based)
- It is set to 0 for no error insertion.

For example, to cause a bit error in the most-significant bit of pixel 10 of line 10 in a CSI Raw 10 image, this corresponds to bit position $10 * 10 + 9 = 109 = 0x6D$. Thus, set:

Video Error = A006Dh.

5.7.2.3 Image Viewing Dialog

An image viewer is built into the application to view image files used in video and DSI Write Memory commands. The viewer can be invoked by pressing Alt-i, causing the image file (if present) of the current command to be displayed in a separate image window. Note that if the image size is large, it is re-scaled to a reasonable viewing size. Press Return or click OK to close the image window.

Viewing also applies if “File Name” is a test pattern name and so is a convenient way to see that the test pattern name syntax is correct. Also, for video files, currently only the first frame is shown.

5.7.3 Frame Timing Dialog

The “Timing Cfg...” button in the main window brings up a dialog to set video-mode frame timing settings. These settings select video-mode protocol options and define active and blanking parameters of video-mode playback. Please note that this dialog does not configure DSI video frames sent via command-mode (i.e. using the Write Memory commands) *except* if an image file is specified and the “Allow Image Rescaling” option

is enabled. In this case, the input image is rescaled according to HActive by VActive before sending.

Figure 9 shows the DSI layout of the Frame Timing dialog:

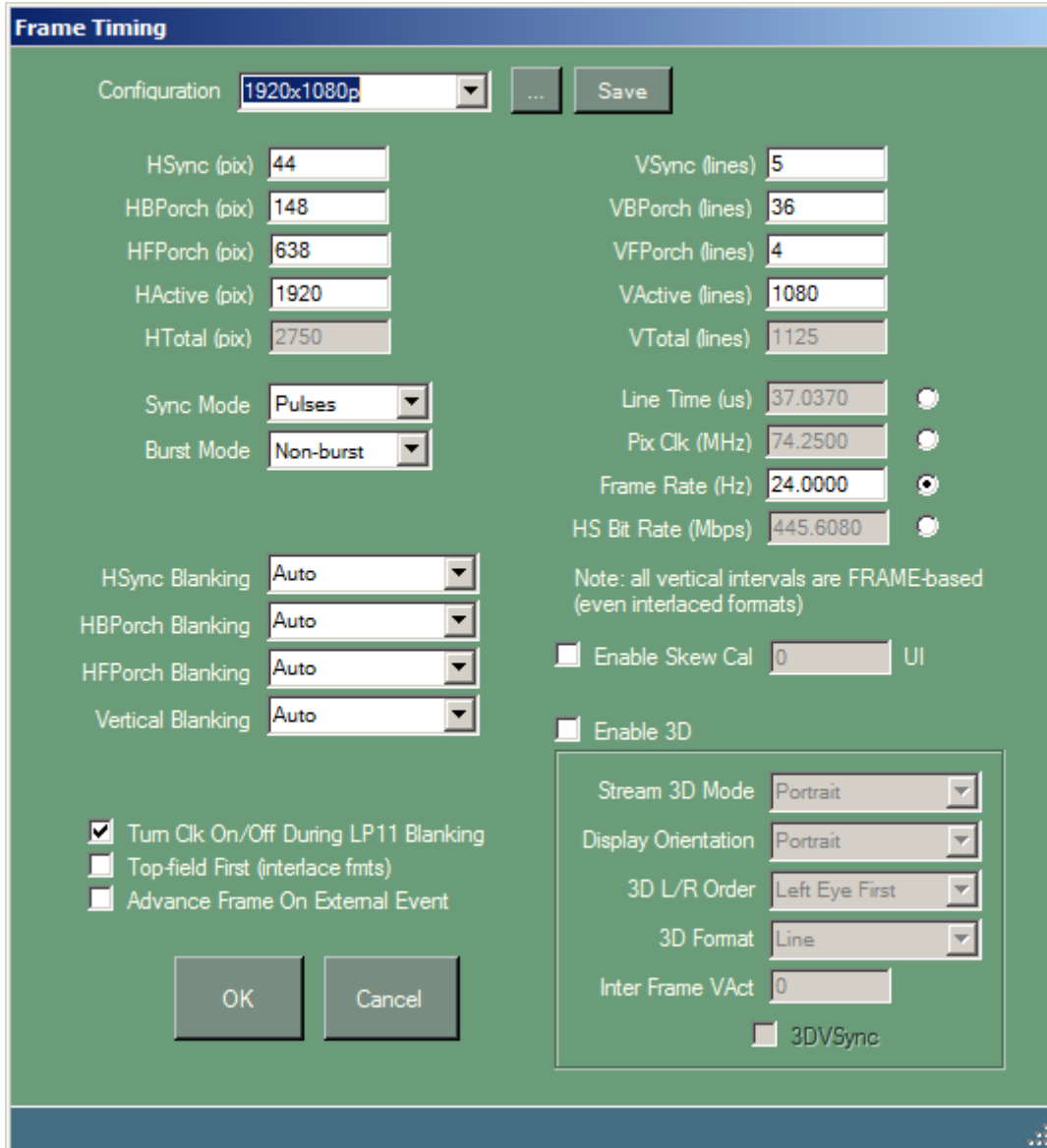


Figure 9 – Frame Timing Dialog (DSI)

Depending on the current MIPI protocol (DSI or CSI), certain fields have different names or are disabled. In particular, CSI:

- Has options “Use LS/LE”, “Frame Numbering”, and “Line Numbering”
- Disables front-porch and back-porch timings

- Disables 3D stream construction fields and “Top-field First” interlace option.

And, alternatively, DSI:

- Has options “Sync Mode”, and “Burst Mode”
- Enables front-porch and back-porch timings
- Enables 3D stream construction fields and “Top-field First” interlace option

5.7.3.1 *Frame Timing Configurations*

At the top-most part of the Frame Timing dialog is a configuration drop-down control along with two buttons, labeled ‘...’ and ‘Save’. These controls are used for named configuration management as follows:

Configuration: drop-down control showing pre-defined and user-defined configuration names. Selecting a configuration name fills in its definition in the remaining frame timing dialog controls.

Pre-defined configurations contain example settings for common frame dimensions. However, please be aware that these configurations may not work for all combinations video formats and lane counts, because of legal timings depend significantly on these parameters.

...: button to launch a List dialog to manage named frame timing configurations (i.e. rename, reset, reorder, delete, etc).

Save: button to save the current control settings as a named frame timing configuration.

5.7.3.2 *Frame Parameter Controls*

Controls near the top part of the window specify horizontal and vertical blanking frame component dimensions as well as the active dimensions of the frame. Read-only H and V totals are automatically computed as field values are entered.

For CSI, note that the HSync blanking period must be long enough to account for all packet overhead in the line (i.e. SOT/EOT and packet headers). For DSI, each blanking period must be long enough to account for individual blanking segment overhead, with front porch blanking additionally accounting for active packet overhead.

If the blanking setting(s) are insufficient, an error message is displayed when a video command is sent. The message also estimates the additional number of pixels needed for HSync to make the video frame implementable, e.g.

“Error: HSync is too short to implement LP11 blanking (need approx. 17 more pixels)”.

5.7.3.3 *Frame Timing Controls*

Controls in the center-right of the window let the user specify the pixel clock timing in four different ways, via: line time duration, pixel clock frequency, frame rate, or HS bit rate. Entering a value in one of these four controls automatically updates the other three.

The HS Bit Rate text box sets/indicates the lane bit rate for the current timing configuration. This value is dependent on the current timing settings, the current packet type selection in the main window (indicating the video format) and the lane count. If the current packet type is not a video command, HS Bit Rate displays “N/A” and the text box is disabled. Otherwise, the text box is enabled, allowing the user to enter a bit rate. Note that if the Lane Count is subsequently changed, the HS Bit Rate changes along with it (keeping the line time, pixel clock, and frame rate constant).

5.7.3.4 *CSI Frame Construction Controls*

There are three frame construction options for CSI:

Use LS/LE: drop-down control selecting Line Start/Line End packet usage:

- **Yes:** Line Start and Line End packets are inserted into the video stream and bracket each active video packet.
- **No:** Line Start and Line End packets are not inserted into the video stream.

Frame Numbering: drop-down control selecting Frame Numbering usage:

- **0:** the frame number parameter in the Frame Start and Frame End packets is set to zero.
- **1,2,3...:** the frame number parameter in the Frame Start and Frame End packets is set, starting with 1 and incrementing each frame.

Note that the frame number only increments for each unique frame in the sequence and repeats when looping. For example, if only a single-frame sequence is sent, the repeating frame always has frame number equal to ‘1’.

Line Numbering: drop-down control selecting Line Numbering usage:

- **0:** the line number parameter in the Line Start and Line End packets is set to zero.
- **1,2,3...:** the line number parameter in the Line Start and Line End packets is set, starting with 1 and incrementing each line.

5.7.3.5 *DSI Frame Construction Controls*

There are three frame construction options for DSI:

HSync Mode: drop-down control selecting the HSync Mode:

- **Events:** horizontal syncs in active lines of video frames are conveyed with only an HSync Start packet

- **Pulses:** horizontal syncs in active lines of video frames are conveyed with an HSync Start packet, followed by blanking time, and then an HSync End packet. Note this option is only available in Non-Burst mode.

Burst Mode: drop-down control selecting the burst mode:

- **Non-Burst:** non-burst mode assumes the DSI clock is just sufficient to transmit active lines with no extra bandwidth. When this option is selected, the HS Bit Rate control is made read-only in the frame dialog and automatically set according to frame timing settings.
- **Burst:** burst mode assumes the clock is set faster than necessary to transmit the video frame. Currently, the excess bandwidth associated with each line is treated as part of horizontal front-porch blanking.

Top-Field-First (interlace fmts): selecting this option causes the top-field (i.e. lines 1, 3, 5...) of the image to be sent before the bottom-field (i.e. lines 2,4,6...) when sending interlaced video. Otherwise, the bottom-field is sent before the top-field.

5.7.3.6 *Common Frame Construction Controls*

Common frame construction options for both CSI and DSI indicate the method to be used for blanking segment implementation:

HSync Blanking: this control specifies whether to implement the HSync blanking period via LP11 or an HS blanking packet. A third option is “Auto” which implements the blanking segments via LP11 unless it is too short in which case the segment is implemented with an HS blanking packet.

HBPorch Blanking: this control (disabled for CSI) specifies whether to implement the HBackPorch blanking period via LP11 or an HS blanking packet. A third option is “Auto” which implements the blanking segments via LP11 unless it is too short in which case the segment is implemented with an HS blanking packet.

HFPorch Blanking: this control (disabled for CSI) specifies whether to implement the HFrontPorch blanking period via LP11 or an HS blanking packet. A third option is “Auto” which implements the blanking segments via LP11 unless it is too short in which case the segment is implemented with an HS blanking packet.

Vertical Blanking: this control specifies whether to implement the blanking periods during Vertical blanking as LP11 or HS blanking packets. A third option is “Auto” which implements the blanking segments via LP11 unless it is too short in which case the segment is implemented with an HS blanking packet. Note that vertical blanking lines in DSI have one blanking period in “Event” mode and two blanking periods in “Pulse” mode (with the HSync End packet in between them).

Turn Clk On/Off During LP11 Blanking: this checkbox specifies if the clock should be turned off at the beginning of LP11 blanking and back on again at the end of LP11 blanking.

Advance Frame On External Event: this checkbox specifies if each frame in a video sequence should loop individually and only advance to the next frame in the sequence when a rising edge is detected on the external event line of the instrument (GPIO of the back-panel connector).

Enable Skew Cal: this checkbox indicates if a deskew burst sequence is to be sent before each frame in a video sequence. If checked, the corresponding textbox is enabled for the user to enter the deskew burst duration (in UI).

5.7.3.7 *Stereoscopic Frame Controls*

The DSI 1.2 specification introduces a new option for sending stereoscopic frames based on the MIPI SDF specification. Stereoscopic frame construction is supported via controls in lower-right corner of the Frame Timing dialog and, for the most part, correspond directly to parameters fields of the SDF specification.

Enable 3D: when checked, this checkbox enables 3D stereoscopic video mode, and enables the remaining stereoscopic configuration controls. As according to the SDF specification, when 3D mode is enabled, the VSync Start packet is sent with a parameter byte that indicates the format of the 3D frame.

Stream 3D Mode: this drop-down control selects whether video-mode frames should be considered to be in portrait or landscape mode.

Display Orientation: this drop-down control selects whether the display is operating in portrait or landscape mode.

3D L/R Order: this drop-down control selects whether data is sent left-eye first or right-eye first.

3D Format: this drop-down control determines how left and right images are to be interleaved (pixel, line, or frame).

3D VSync: this checkbox determines whether a second VSync is sent between left and right images when 3D Format is set to Frame mode.

Inter Frame VAct: the text box accepts an integer value that determines how many lines should be sent between left and right images when 3D Format is set to Frame mode and 3D VSync is not checked.

5.7.4 Video Frame Construction Details

In contrast to relatively simple non-video-mode commands, video-mode commands define a complicated program stream of LP signaling and HS packets.

In CSI, a frame consists of the following:

- (VSync – 1) blanking lines
- First active line with Frame Start packet
- (VActive – 1) remaining active lines
- Blanking line with Frame End packet

All lines are of equal length in time; thus the lines with Frame Start and Frame End have shorter real horizontal blanking to account for the timing difference.

In DSI, a frame consists of the following:

- Blanking line with VSync Start packet
- (VSync – 1) blanking lines
- **Pulse Mode:** Blanking line with VSync End packet
- **Event Mode:** Blanking line with normal HSync Start packet
- (VBPorch – 1) blanking lines
- VActive active lines
- VFPorch blanking lines

A blanking segment can be implemented by HS blanking packet or a round-trip transition to LP11 for the required duration. Blanking segments set to “Auto” that are long enough to incur the overhead of the round-trip transition to LP11 use this method. Otherwise, the interval is implemented with a blanking packet. Of course, blanking packets also have overhead, and so segments must be at least long enough to accommodate the minimum length blanking packet.

While the desired timing characteristics of a video line are defined in the frame timing dialog, actual timings may slightly differ. This is because of the complexity of constructing arbitrary packet timing given internal restrictions for storing and playing LP states and HS bits in the instrument. In particular, all LP and HS periods are required to have periods that are a multiple of 2*UI and minor quantization of blanking segments may result.

5.7.4.1 Stereoscopic Frame Construction

The 3D stereoscopic standard impacts only the payload content of video-mode frames. Thus, the timing, dimensions, and structure of 3D frames are identical to non-3D frames. Regardless of 3D parameter settings, output frames always have dimension HActive by VActive. Input images are always rescaled if necessary during output frame construction.

Video-mode commands sent when stereoscopic video is enabled (i.e. Enable 3D checked) cause 3D frames to be constructed and sent. If the “File Name” argument references a

single image or image sequence (see section 5.7.2.1), then each single image is used to build a pseudo-stereoscopic image pair by extracting alternating left/right pixels, rescaling if necessary, and then building the composite output image based on the SDF specification.

If the “File Name” argument references a true stereoscopic image or image sequence, then each image pair is imported, rescaled if necessary, and then used to build the composite output image.

Note that in the case where 3D Format is set to Frame and 3DVSyn is checked, 2 * “Frame Count” images are sent in the output.

5.7.4.2 Command Insertion

Note that video frames are automatically built with a command insertion point in the first vertical blanking line. Command insertion is a special mode that allows most commands to be inserted into the program stream *while the program is looping*. This function is also available for the user during macro looping using the “Cmd Insertion Point” command. Please see section 5.13.3 for more information.

5.7.4.3 Video Memory Requirements

Memory requirements for the storage of video sequences in the DPhy Generator cannot be exactly described, due to the complexity of encoding, packing, and outputting the program stream as well as numerous dependencies such as number of lanes, video format, frame timing, and DPhy bus timing. However, in general, a reasonable upper bound can be given, based on HS bit packing in the instrument's memory.

Basically, the packing efficiency of HS bits in memory is 75%, regardless of the number of lanes. Since the amount of memory in the DPhy Generator is 2 GB, this means a maximum of 1.5 GB of memory is available for video data. Thus:

$$\text{Max Frames} = 1.5 \text{ GB} / ((H_{\text{Active}} * V_{\text{Active}} * \text{BytesPerPixel}))$$

A more accurate way to determine how many frames fit into memory given specific frame timing settings is to send a small sequence consisting of, say, 10 or 20 frames. Once running, the status readout indicates how much memory of the PG is used, as a percent of the total available.

Note that, currently, the DPhyGenCtl application does not manage host memory with great efficiency and cannot generally achieve the maximum frame count supported by the DPhy Generator (an out-of-memory error occurs when building the sequence). If this is a concern, please contact The Moving Pixel Company for more information. To improve the maximum frame sequence count, please ensure your host computer has a 64-bit processor, with at least 8 GB of memory, and a large page-file configuration.

5.7.5 DSC Compression

Extensive Display Stream Compression (DSC) support is provided with the DSC License option (please contact the Moving Pixel Company for licensing information). Without

the licensing option, “Compressed Pixel Stream” and “Picture Parameter Set” commands can still be sent, but only accept binary file input for their “File Name” parameter. This section documents the additional features supported with the DSC License.

5.7.5.1 DSC Configuration Dialog

The DSC Configuration dialog sets all parameters related to DSC encoding and is brought up using the “Configure DSC” menu item (Options menu). As with other configurations (frame timing and DPhy timing), DSC configurations can be specified and saved, allow for easy selection in the future.

A summary of dialog controls follows (see Figure 10). Please note that dialog settings always override any equivalent settings in the configuration file.

- **Configuration:** drop-down control to select a named DSC configuration. When selected, the parameters of the configuration fill in the other dialog controls.
- **“...”:** button next to the Configuration drop-down control to bring up a List dialog to order, rename, and delete named configurations.
- **Save:** button to save the current control settings to a named configuration. A name dialog is provided for the user to enter in a configuration name. If the name exists, the user is asked whether it is okay to overwrite the existing configuration.
- **Config File:** file name of a DSC configuration text file. The syntax of this file is beyond the scope of this document (please see the DSC reference encoder documents). However, it should be noted the LINE_BUFFER_BPC is set to BITS_PER_PIXEL + 1 by default, but can be overridden if set in the configuration file. Standard configuration files are located in the application directory and are already pre-defined in named configurations for use in DPhyGenCtl.
- **Browse:** button to bring up a file dialog to select a configuration file.
- **Slice Width:** text box to set the slice width. Set to 0 for the entire picture width.
- **Slice Height:** text box to set the slice height. Set to 0 to use the entire picture height.
- **Enable 422:** checkbox to enable 422 color component sampling.
- **Use YUV Input:** checkbox to use the YUV color space.
- **Use Block Prediction:** checkbox to enable block prediction.
- **Send One Chunk Per Packet:** checkbox selecting how chunk data is to be sent, both for Write Memory commands and video-mode commands. If checked, one chunk will be sent per packet. Otherwise, one or more chunks representing an image line will be sent in a single packet.

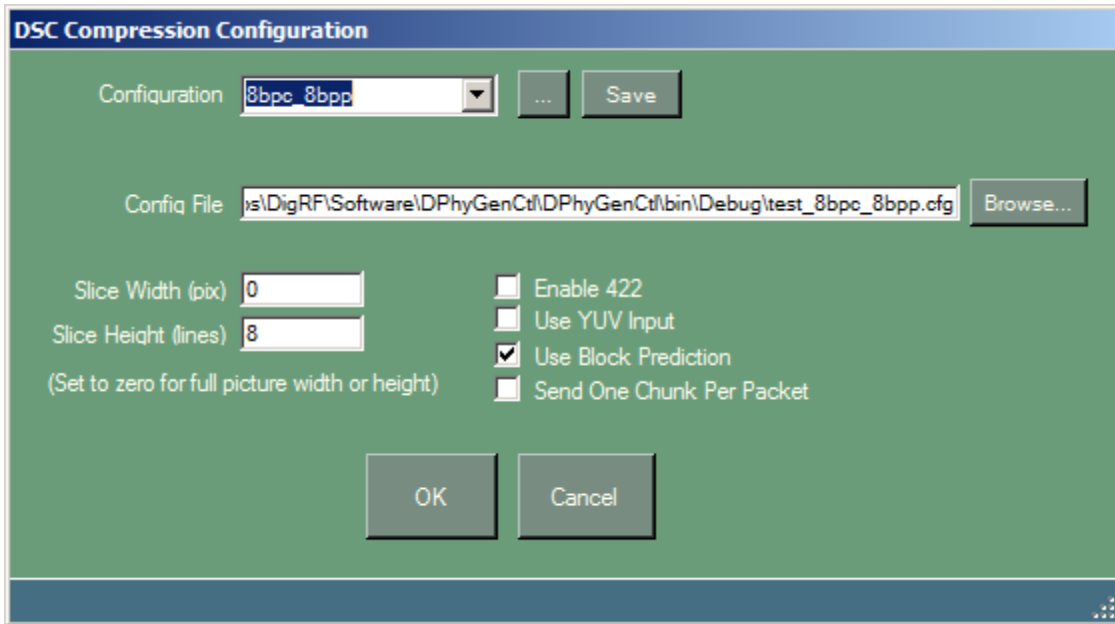


Figure 10 – DSC Configuration Dialog

5.7.5.2 *Sending Compressed Frames in Video Mode*

The “Compressed Pixel Stream” packet type is used to send one or more frames of compressed video. This command is very similar to other video-mode packet types, accepting the same arguments and implementing the frame timing as configured by user settings. The biggest difference is that image frames are first compressed (if not already compressed) using the current DSC configuration settings before the output video-mode frame is built.

Notes:

- In addition to the standard image, video, test pattern, and binary file types, the “File Name” argument has been extended to support DSC and DPX file types for this command. If a DSC file is provided, chunk data is sent directly as “Compressed Pixel Stream” packet payload. If a DPX file is provided, it is first compressed (like other non-binary file types) before used as packet payload.¹²
- Like other video-mode packet types, setting Frame Count greater than one causes a multi-frame sequence to be sent. Multi-file naming syntax can be used to specify the sequence of source image files to use.
- The “Video Error” parameter also is supported, with its value still encoding the line and error bit location. Note that the bit location indicates the error bit in the *compressed source data* and so can only affect payload data (not the packet

¹² Note that the DPX file type does not support automatic rescaling even if “Allow Image Rescaling” is enabled. Thus, the file must already have the correct output frame dimensions.

header or CRC). This applies regardless of how many packets are used to send a line (i.e. independent of the “Send One Chunk Per Packet” setting).

5.7.5.3 Sending Compressed Frames in Command Mode

As with uncompressed images, DSC compressed video frames can be sent using the DCS Write Memory commands. This is achieved by selecting the “DSC Compressed” option for “Image Decode Format” in the Write Memory Configuration dialog and sending a Write Memory command.

Command arguments and behavior are the same, with the following extensions and clarifications:

- In addition to the standard image, test pattern, and binary file types, the “File Name” argument has been extended to support DSC and DPX file types for this command. If a DSC file is provided, chunk data is sent directly as Write Memory packet payload. If a DPX file is provided, it is first compressed (like other non-binary file types) before used as packet payload.
- The “Partition Length” and “Quantize Packets to Multiple of Line Length” settings in the Write Memory configuration are ignored. The payload length is always either one chunk or one line's chunk data, based on the setting of the DSC configuration option “Send One Chunk Per Packet”.

Please see the more detailed description of operation in section 5.6.6.10.1.

5.7.5.4 Sending PPS Information

The Picture Parameter Set command conveys DSC compression information to the DUT in advance of sending compressed video packets. Without the DSC license option, only binary files can be sent with this command.

With the DSC license option, the user can also specify one of the following as the “File Name” argument:

- DSC compressed file name. In this case, the PPS structure in the file is sent as payload for the command.
- Any valid image or test pattern file name. In this case, a PPS structure is built using the current DSC configuration settings and active frame dimensions in the current frame timing configuration.

5.7.5.5 Compressing/Uncompressing Image Files

With the DSC license option, two keyboard functions are enabled in the main GUI window:

- **Control - C:** Compress the file or test pattern associated with the current “File Name” and output to a DSC file. The user is prompted for an output file name. If “Allow Image Rescaling” is enabled, the input image is first rescaled to the active frame dimensions in the current frame timing configuration.

- **Control - U:** Uncompress the DSC file in the current “File Name” field and output to an image file. The user is prompted for an output file name, which can have any common image extension (e.g. BMP, JPG, DPX).

5.7.5.6 Viewing DSC Images

With the DSC license option, the image viewing keyboard function is extended to view DSC and DPX files. Simply press **Control – I** to view the file when entered into the “File Name” field of the current command.

5.8 Macros

A macro is a sequence of MIPI commands, grouped and named as a single, new command. Once defined, they can be assigned to command buttons and sent to the DPhy Generator like any other command. When viewing and creating macros, controls in the Macro tab of the main window is used. These controls displays the component commands of the macro and allows basic macro editing (see Figure 11 below).

5.8.1 Defining a New Macro

To define a new macro, click on the “Start Macro” button at the bottom right of the Macro tab pane. This button changes its caption to “End Macro” and activates the Macro Cmds list-box with no component commands initially present.¹³ In this editing mode (called macro mode), commands that are sent either via a command button, via the Send button at the top of the main window, or via an RPC programmatic command are entered into the macro instead of immediately sent to the DPhy Generator.

Note that in macro mode, the Send button label changes to “Insert” to indicate commands are to be inserted into the macro definition. If the Shift key is held down, the button label changes to “Replace” to indicate commands are to *replace* the current command in the macro definition.

Figure 11 shows an example of a macro in edit mode. Commands are added to the macro at the location highlighted in the macro pane. When added, a dialog appears for the user to name the component command of the macro.¹⁴ The user can scroll and select a command (or the blank line at the end of the command list) to mark the next command insertion/replacement location. Selecting a command also displays its arguments for viewing and modification.

Six buttons are available in the macro pane to manipulate the component commands of a macro.

- **Clear All** removes all the component commands in the macro.
- **Delete** removes the selected component command from the macro.
- **Rename** brings up a dialog to rename the selected component command

¹³ Note: if the current Cmd Name is a macro then the list-box will be initialized with the macro's component commands. If these commands are unwanted, the Clear All button can be used to delete them.

¹⁴ Note: the component command name dialog can be disabled by checking the GUI Option “Don't ask for component command names during macro definition” (Options->Set GUI Options...). Component commands can still be renamed after insertion using the “Rename” command button.

- **Up** moves the selected component command up within the command sequence
- **Down** moves the selected command down within the command sequence.
- **LS/LE** adds a “Loop Start” and “Loop End” command pair to the macro (which can then be positioned as desired using Up/Down). Before adding, a loop count dialog appears for the user to enter a loop count (0 for infinite).

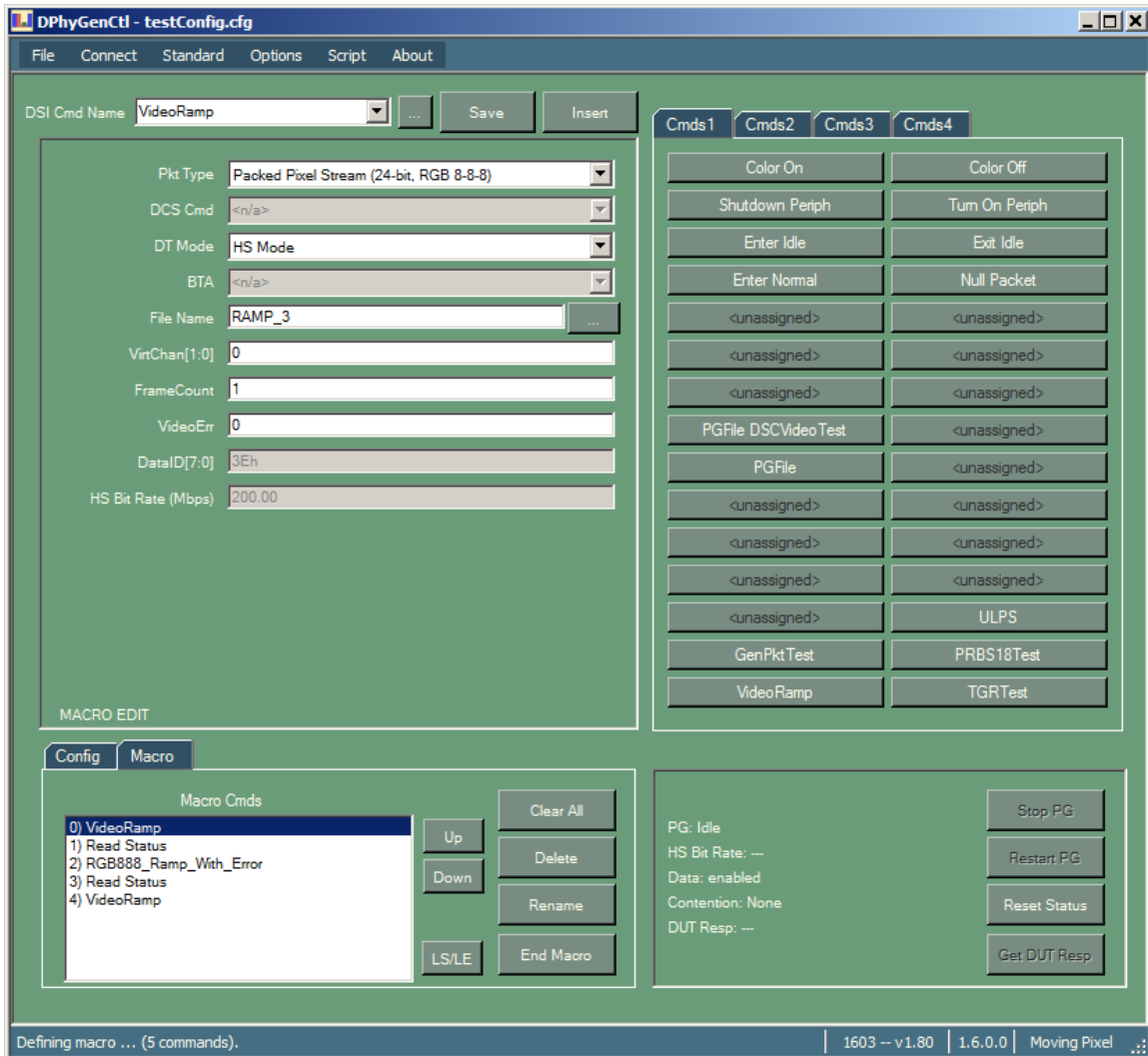


Figure 11 – Macro Editing

When macro definition is complete, the user clicks on the “End Macro” button, bringing up a dialog to name the macro, save it, and exit macro mode (see Figure 12). Alternatively, the user may click Cancel to return to macro editing or “Discard Edits” to exit macro made, discarding changes entirely.

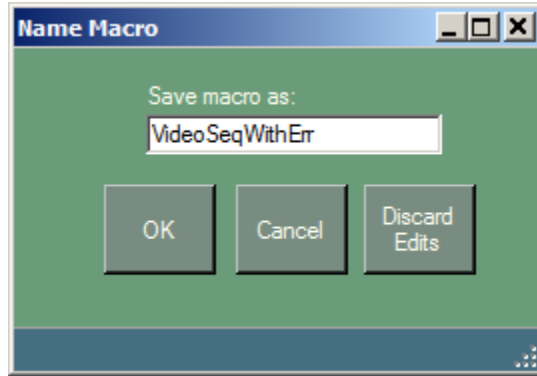


Figure 12 – Name Macro Dialog

As with command definition, if the user specifies the same name when saving a macro, the user is asked for permission to replace the existing macro definition.

5.8.2 Editing an Existing Macro

To edit an existing macro, the user has two options. If the macro is assigned to a command button, the user may right-click on it and select “Edit Current Command...”. Alternatively, the user can select it in the command combo-box at the top of the main window then click on the “Start Macro” button. The macro can then be edited as normal and saved by clicking on the “End Macro” button. The name field is filled in with the original macro name and the user can click OK and confirm overwriting the existing macro.

5.8.3 Copying a Macro

To copy a macro, simply load a macro for editing, click on the “End Macro” button, and change the original name to a new name when saving.

5.8.4 Inserting a Component Command

To insert a new component command into a macro, perform the following (once in macro editing mode):

- Click on a component command in the macro to set the insertion location
- Send a command to insert it into the macro using any normal method, i.e.
 - a. Select a Cmd Name and click the ‘Send’ button
 - b. Select a Pkt Type, fill in arguments, and click the ‘Send’ button
 - c. Click on a Cmd button

5.8.5 Replacing a Component Command

To replace an existing component command in a macro, perform the following (once in macro editing mode):

- Click on a component command in the macro to select the command to replace

- Send a command to replace it into the macro using any normal method *while holding down the Shift key*:
 - a. Select a Cmd Name and click the 'Send' button
 - b. Select a Pkt Type, fill in arguments, and click the 'Send' button
 - c. Click on a Cmd button

5.8.6 Editing a Component Command

To edit and replace a component command in a macro, perform the following (once in macro editing mode):

- Click on a component command in the macro to retrieve its argument settings.
- Change the argument settings as desired.
- Click the 'Send' button while holding down the Shift key to replace the modified command into the macro.

5.8.7 Allowed Component Commands

The intent of a macro is to output a MIPI command sequence in real-time on the bus. Allowed macro component commands include all standard or custom MIPI packet types (LPDT or HS) accessible via the PktType drop-down control, as well as all Phy commands and Miscellaneous commands except for the "RPC Script" command.

In addition, the following notes apply to macro behavior:

- Component commands with BTA enabled are allowed in macros and behave as expected. The PG waits for a return BTA before continuing with the remaining component commands in the macro.
- Write Memory Start and Write Memory Continue commands are fully supported in macros, including image file decoding and command partitioning to support large "Write Len" counts.
- Macros cannot contain other macros; that is, macro nesting is not allowed. However, macros can nonetheless be added to other macros, in which case the component commands of the macro being added are themselves duplicated and added to the macro being edited.
- Video-mode commands can now be added to macros. Please see the next section for more information.

When a macro is built using an RPC program or script, allowed macro component commands are any command that can be sent via the following RPC commands:

- SEND_MIPI_CMD
- ADD_MIPI_CMD
- SEND_IMPAIRED_MIPI_CMD

In addition, a new feature supported by macros is the ability to contain certain non-MIPI configuration commands, allowing on-the-fly application reconfiguration (but not hardware reconfiguration!) for subsequent commands in the macro. For example, certain frame timing or DPhy timing parameters can be changed between two frames defined in the macro.

Generally, the following classes of non-MIPI RPC commands are allowed in macros:

- Frame buffer load and deallocation
- WriteMemory configuration
- DPhy Configuration
- Frame Timing configuration (except for frequency-related settings)
- DSC configuration
- Option setting that applies to program construction
- Bayer encoding type

For more details, please refer to the document `DPhyGenCtlRPCCommands.pdf`, which indicates which configuration RPC commands are allowed in macros (Note 9).

5.8.8 Video-Mode Commands In Macros

Video-mode command behavior in macros is determined by the setting of the option “Enable Video Mode In Macros” (Options menu). This section describes the two behaviors.

5.8.8.1 “Enable Video Mode In Macros” Enabled

If the option “Enable Video Mode In Macros” is enabled, a video-mode command in a macro causes a full video frame to be sent using the current frame timing configuration. However, note that when video-mode frames are played in macros, the clock rate used is the HS Bit Rate setting for the macro, *not the configured HS Bit Rate in the frame timing configuration*. If accurate frame timing is required, the user must set the HS Bit Rate in the main window to the same as the HS Bit Rate in the frame timing dialog when sending the macro

5.8.8.2 “Enable Video Mode In Macros” Not Enabled

If the option “Enable Video Mode In Macros” is not enabled, only a single video data packet is played in the macro. In this case:

- The entire contents of the given file are put in the video packet (which, to be MIPI legal, must be less than 64 Kbytes)
- The “Frame Count” parameter is ignored
- The file argument is assumed to reference a binary file. That is, BMP file decoding is not supported.

Note that one additional variation is supported for video commands, though in general, it is accessible only through RPC (either via script or user program). This variation allows the video command to reference an internal frame buffer, previously loaded with a `LOAD_FRAME` RPC command. In this case, the filename used to reference the frame

buffer is USERFRAME<n> where n is 0 through 3 and the two video command arguments refer to a byte offset and byte length in the frame buffer data.

5.8.9 Looping In Macros

The “Start Loop” and “End Loop” commands are used in macros to provide an easy way to duplicate a group of commands. Simply bracket the command group with the loop commands, for example

- ...
- Start Loop
- Cmd0
- Cmd1
- Cmd2
- ...
- End Loop
- ...

With the exception of infinite looping, this construct *does not* implement true looping and thus does not save any memory or construction time. But it is a useful convenience, especially for video frame construction, where a command sequence that implements one video line can be looped multiple times.

The “Start Loop” command has one argument, “Loop Count”, which indicates the number of times to loop the command group between start and end. A value of 0 indicates infinite looping, which causes the macro program to be constructed and sent as described in the next paragraph.

Commands that precede the infinite Start Loop are output normally and can be considered like an initialization sequence. Commands contained in the infinite loop are treated like a normal looping macro, outputting indefinitely until the “Stop PG” button is pressed. Note that commands that follow the infinite loop are discarded and never sent.

Generally, nesting of loops is *not* allowed, though multiple loops can occur serially within a macro. One exception is that non-infinite loops can occur within an infinite loop.

Note that there are pitfalls with using looping in macros. In particular, be aware that large loop counts can easily cause out-of-memory errors and so should be used with care (infinite loops do not consume additional memory). Also, loop sequences must be constructed carefully such that discontinuities do not occur at loop boundaries. It is important to ensure that clock state and bus state have legal transitions at loop boundaries. For example, looping on a single CLOCK_ON command will result in an illegal MIPI sequence, since the clock is never turned off before looping.

5.8.10 HS Component Command Behavior

When macros are parsed by DPhyGenCtl, the application can group consecutive HS component commands (that don't have BTA enabled) into a single HS burst. Whether consecutive HS commands are sent in a single burst or sent individually in their own burst is determined by the setting of the "Send Single Pkt Per HS Burst" option (Options menu).

For mixed burst behavior, the user can disable this option, which will by default group consecutive HS commands into a single burst. Then, for those HS commands to be sent in their own burst, the user can put "LP Delay" commands between them.

5.8.11 Additional Notes on Macro Behavior

Below lists some final miscellaneous notes on DPhyGenCtl behavior when working with macros:

- As macros are commands, macro names must be distinct from command names.
- On the other hand, component command names in macros are simply descriptive names for the component command and have no relation to command names in the application. Component command names do not need to be unique to each other or to command names in the application.
- Once component commands are added to a macro, they are duplicated and no longer are associated with the original command. Specifically, if the original command is then modified and saved, the component command in the macro having the same name does NOT change.
- When building a macro via script or RPC, component commands are added to the macro when using the RPC commands SEND_MIPI_CMD or ADD_MIPI_CMD.
- Also, when building a macro via script or RPC, sending a configuration command that is explicitly allowed in a macro (see section 5.8.7) adds the command to the macro. On the other hand, sending a configuration command that is not explicitly allowed in a macro is instead immediately executed and NOT added to the macro.

5.9 *Phy Commands*

Commands in the "Phy Commands" category of the "Pkt Type" drop-down control invoke lower-level behaviors and modes of the DPhy Generator instrument.

These consist of the following command types, described subsequently:

- Clock On/Off
- Escape Command Sequences (e.g. BTA, ULPS, etc)
- Skew Calibration Sequence
- TGR Sequences
- PRBS Sequences

5.9.1 Clock On / Clock Off

There are two Phy commands that control clock behavior in a macro: “Clock On” and “Clock Off”.¹⁵ These commands operate as expected, with behavior that prevents illegal duplicate clock on/off sequences, that is:

- 1) If the clock lane is off (i.e. in LP11) in a macro when the “Clock On” command is sent, the clock lane outputs proper DPhy signaling to turn the clock on at the current “HS Freq” setting. If the clock is already on, the lane clock remains running and no additional signaling occurs as a result of the command.
- 2) If the clock lane is on in a macro when the “Clock Off” command is sent, the clock lane outputs proper DPhy signaling to turn the clock off (i.e. transition to LP11). If the clock is already off, the lane clock remains off and no additional signaling occurs as a result of the command.

For further discussion of clock lane behavior, please see section 4.3.

5.9.2 ULPS

The ULPS command causes the designated lanes to enter the ULPS state. The LaneMask[4:0] argument indicates which lanes are to transition to ULPS (with bit 0 representing lane 0, bit 1 representing lane 1, etc. and bit 4 representing the clock lane). This command implicitly embeds a Wait Host Event command following the ULPS entry sequence, which waits for an event signal that can be generated by the user using the “Host Event” button of the main window. Behavior is as follows:

When the command is sent, the PG outputs the ULPS entry sequence on selected lanes and continues to run as the instrument holds lanes indefinitely in ULPS. During this time, status indicates that the PG is “Waiting Host Event”. Until this event is seen (or the PG stopped), lanes remain in ULPS. When the host event occurs, the PG outputs the ULPS exit sequence on selected lanes and returns to LP11.

Note that this command can be looped like most commands. If the “Loop Commands” option is checked, then the ULPS command repeats after the Host Event occurs. Please note that, from the user's perspective, this can be confusing, as nothing seems to change since the PG status remains static: the PG continues to run and the “Waiting Host Event” message is still displayed.

5.9.3 BTA

BTA (bus turn-around) is a handshake protocol to transfer bus ownership to the DUT. (applicable only in DSI). Read commands by default have their BTA argument set to “Yes” and this argument can be set to “Yes” by the user for other commands as well. When BTA is set, the BTA signaling sequence is appended to the command. In addition, a BTA sequence can also be sent on its own, without accompanying data by selecting “Send BTA” in the “Pkt Type” command field.

¹⁵ The “Clock On” command can also be sent as a single command to turn the clock on briefly before it is turned off at the end of the command, but this is only be useful for low-level testing.

After a BTA sequence is sent, the DPhy Generator suspends program output and places its drivers in high-impedance until it receives a return BTA sequence from the slave DUT or times out. When a return BTA sequence is seen on the bus, the instrument drives a BTA acknowledgement sequence and then resumes program output. In the event of a BTA timeout, the instrument resumes program output without sending a BTA acknowledge.

This wait-for-BTA process can be initiated without first sending an initial BTA sequence by using the “Wait For BTA” command. This command does not transmit any data on the bus but simply drives high-impedance and waits for a BTA sequence from the slave DUT or eventually timing out.

Notes:

- These commands are most useful when embedded in a macro with other MIPI commands. Component commands in the macro following a “Send BTA” or “Wait For BTA” are not output until the return BTA is recognized or the BTA timeout has expired.
- The BTA timeout is set in the Instrument Configuration dialog with the control “BTA Wait Time”. To avoid any timeout, the user can set the “Disable Command Timeout” option (Options menu). If necessary, the “Stop PG” button can be used to terminate a command waiting for a return BTA.
- While the DPhy Generator is waiting for a return BTA from the DUT, the main window status bar displays “Waiting on BTA...”.

5.9.4 Escape Command

The Escape Command option lets the user send a Phy escape command byte. The only argument is the command byte to send. Note that this command is different than a packet sent in LPDT mode. LPDT mode uses an escape command byte of 0x87 to put the receiver in LPDT mode, and then sends the packet bytes.

5.9.5 Skew Calibration

The Skew Calibration command is a command that sends a deskew burst on all active lanes as defined in DPhy 1.2. This is used for receiver calibration at high bit rates (generally > 1.5 Gbps).

The deskew burst sequence is similar to a HS burst with the following exceptions:

- The sync pattern is 16 ones (i.e. 0xffff)
- Instead of packet data, a clock pattern is sent.

The skew calibration command has one argument which is the duration of the burst in UI (quantized to a multiple of 2).

The user can send the Skew Calibration command as a single command or embed the command in a macro where desired. In addition, the Frame Timing dialog allows the user to enable sending a deskew burst at the start of each frame in a single or looping sequence. This applies to both video-mode commands and video-mode sequences sent in a macro.

To enable sending a deskew burst at the start of each video frame, check the “Enable Skew Cal” checkbox in the Frame Timing dialog (Timing Cfg... button in the main window). This enables a textbox next to the checkbox to enter in the burst duration in UI.

5.9.6 TGR Data Sequence

The TGR function has been adopted from CPhy to provide low-level support for arbitrary HS data transmission, which can be useful for general testing.

The TGR Data Sequence command starts by sending an HS burst entry sequence on all active lanes using the current DPhy timing settings. Note that no Sync byte is sent. Following the HS burst entry sequence, all active lanes loop on a specific data pattern defined by the Params[] argument, which is a variable-length list of data bytes. The pattern loops for the number of iterations specified by the LoopCnt parameter, eventually exiting back to LP11 with the HS burst exit sequence.

If LoopCnt is set to zero, the burst data loops indefinitely in HS mode, never completing a single burst until the “Stop PG” button is pressed (at which point the HS burst exit sequence is output and the PG becomes idle). Otherwise, after one complete burst has been sent, the burst is repeated again and again, only stopping when the “Stop PG” button is pressed. Note that this outer looping occurs even if the “Loop Commands” option is unchecked.

Note that one constraint on the LoopCnt parameter is that the total number of looped bytes in the sequence (i.e. LoopCnt * sequence bytes) must be greater than or equal to 64. Otherwise, an error message is displayed.

5.9.7 TGR Bit Sequence

The TGR Bit Sequence command is nearly identical to the TGR Data Sequence command (see previous section) except that the user indicates sequence bits (0 or 1) rather than bytes. In this case, the total number of looped bits in the sequence (i.e. LoopCnt * sequence bits) must be greater than or equal to 256. Otherwise, an error message is displayed.

5.9.8 PRBS9 Sequence

The PRBS9 Sequence has been adopted from the CPhy specification to provide low-level support for randomized data transmission, which can be useful for general testing.

The PRBS9 Sequence command starts by sending an HS burst entry sequence on all active lanes using the current DPhy timing settings. Note that no Sync byte is sent.

Following the HS burst entry sequence, each active lane outputs a specific data pattern generated by the PRBS9 algorithm, which consists of a sequence of 16-bit randomized words having a length specified by the WordCnt parameter. After the data sequence is complete, lanes are returned to LP11 with the HS burst exit sequence.

If WordCnt is set to zero, the burst data loops indefinitely in HS mode, never completing a single burst until the “Stop PG” button is pressed (at which point the HS burst exit sequence is output and the PG becomes idle). Otherwise, after one full burst has been sent, the command has completed, but like other commands may optionally be looped either in a macro (using Loop Start and Loop End commands) or as an individual command using the Option->Loop commands menu option).

In addition to the WordCnt parameter, the PRBS9 Sequence command accepts four more arguments, initializing each lane's PRBS generator.¹⁶ Optionally, the user can specify identical seed values causing all active lanes to have identically synchronized data patterns:

- L0_Seed[15:0] – lane 0 starting PRBS seed (should be nonzero)
- L1_Seed[15:0] – lane 1 starting PRBS seed (should be nonzero)
- L2_Seed[15:0] – lane 2 starting PRBS seed (should be nonzero)
- L3_Seed[15:0] – lane 3 starting PRBS seed (should be nonzero)

5.9.9 PRBS11 Sequence

The PRBS11 Sequence is nearly identical to the PRBS9 Sequence (see previous section) except that the PRBS algorithm is generated from an 11-bit LFSR.

5.9.10 PRBS18 Sequence

The PRBS18 Sequence is nearly identical to the PRBS9 Sequence and PRBS11 Sequence (see previous sections) except that the PRBS algorithm is generated from an 18-bit LFSR. To this end, the initial seed parameters are 18-bit values.

5.10 File Command

The File Command is general-purpose, low-level command that uses a text file as input to define arbitrary LP and HS DPhy signaling sequences. Using special file command syntax, low-level LP and HS lane states and higher-level packet data bytes can be intermingled as desired to create user test sequences and error packets. Command syntax also supports useful features such as automatic ECC and CRC generation, HS burst signaling (e.g. HS entry/exit and sync), and packet data demultiplexing.

Note that the “Send Cmd To File” option (File menu), while not yet implemented, is intended to allow automatic file generation of any single command (non-macro, non DCS Write Memory, non-video command). The format of this file is ASCII text, suitable for editing and re-sending to hardware via the File command.

¹⁶ Note that inactive lanes remain in LP111 and their seed values have no effect on output.

The File Command can be found in the “Miscellaneous Commands” category of the “Pkt Type” drop-down control and has one argument (“File Name”) that specifies the source text file name to parse and use to construct the DPhy signal. The command is sent like any other normal command and can be part of a macro. The DT Mode is ignored when sending the File Command, but other GUI settings such as the HS Bit Rate, LP Freq, Lane Cnt, and DPhy bus timings may affect the output, depending on the component commands in the file.

The file used for the file command is a form of script file. Each line can be one of the following:

- 1) **Command line** – A command line always begins with the ‘#’ symbol, followed by a space, followed by the command and any arguments, separated by spaces. The command and arguments always use only one line (not including data)
- 2) **Data line** – A data line consists of one or more hex data values, separated by spaces.
 - Multiple data lines can be used to form a longer sequence of data values.
 - Concatenation of data lines occurs to form one data sequence until the next command line (or end of file).
 - A data sequence is always associated with the preceding command.
 - Depending on the associated command, data values may represent HS data bytes, single-lane LP states, or 4-lane LP states.
- 3) **Comment line (or blank line)**: Lines that start with “//” or that are blank (with only spaces or tabs) are considered comment lines and are ignored by the parser.

In constructing an output stream, the file associated with the File Command is parsed in sequence: one line at a time, one component command at a time. After parsing the entire file, the result is a single output stream that is then added to a macro or sent on to the DPhy bus.

If the File Command is sent stand-alone (i.e. not in a macro), it file should be constructed to assume the initial bus state is LP11 in all lanes. Similarly, after the command is sent, the bus is forced back to LP11. In a macro, the File Command begins at the last state occurring in the previous component command in the macro, which generally is LP11. It is possible, however, for a previous component command in a macro to finish in a bus state other than LP11, for example, from a second File Command, LP Delay command, etc.

Note that certain degenerate patterns may not be able to be implemented by the DPhy Generator as there are structural requirements on the program stream it can output. Thus, it is possible for the stream described in the File Command to generate an error during construction if it fails these requirements. The error message says

“Program cannot be sent by hardware at the current HS Bit Rate”.

While exact requirements are too complicated to document, failure is usually due to too few data bytes on average in constructed output records relative to the HS Bit Rate throughput requirements. Generally, a work-around is to add more bytes to the stream or decrease the HS Bit Rate.

Another possible error message is:

“DefineBlock: error PG block size is too small”.

This error occurs if the program contains too little data, failing the minimum block length requirements of the PG. This error is most commonly seen when the user forgets to bracket HS packet data with HS_BURST_ENTRY and HS_BURST_EXIT commands.

The next sections describe component commands and data.

5.10.1 File Command File Syntax

Table 6 – File Command File Syntax

Command	Description
HS_BYTES <lgwd>	Sends one or more HS packet bytes, allocated to lanes according to the argument.
HS_BYTES_PLUS_CRC <lgwd>	Sends one or more HS packet bytes, automatically appending a 2-byte CRC to the stream. Bytes are allocated to lanes according to the argument.
HS_BYTES_PLUS_ECC <lgwd>	Sends one or more HS packet bytes (generally 3), automatically appending an ECC byte to the stream. Bytes are allocated to lanes according to the argument.
HS_BITS <lg>	Sends a sequence of HS bits. Bits are assigned to one lane or replicated across active lanes according to the argument.
HS_ZERO <lg> <dur>	Sends a sequence of HS zero bits. Bits are assigned to one lane or replicated across active lanes according to the argument. The number of sequence bits is derived from <dur> and is always quantized to a multiple 2 UI.
HS_ONE <lg> <dur>	Sends a sequence of HS one bits. Bits are assigned to one lane or replicated across active lanes according to the argument. The number of sequence bits is derived from <dur> and is quantized to multiple of 2 UI.
LP_STATES [ACT] [<dur>]	Sends one or more LP states. If the optional ACT argument is present, then state values that follow are interpreted as 2-bit states that are to

	be replicated and sent on each active lane. Otherwise, state values are interpreted as 8-bit states, 2 bits per lane. If the <dur> argument is not present, the current LP frequency period is used.
LP_ESC_BYTES	Sends one or more data bytes in LP escape mode using spaced-one-hot encoding.
LP_ESC_BYTES_PLUS_CRC	Sends one or more data bytes in LP escape mode using spaced-one-hot encoding, automatically appending a 2-byte CRC to the stream.
LP_ESC_BYTES_PLUS_ECC	Sends one or more data bytes in LP escape mode using spaced-one-hot encoding, automatically appending an ECC byte to the stream.
HS_BURST_ENTRY	Sends the HS burst entry sequence on active lanes using the current DPhy timing settings. This includes LP signaling, HS zero, and the HS Sync byte. No element stream is associated with this command.
HS_BURST_EXIT	Sends the HS burst exit sequence on active lanes using the current DPhy timing settings. This includes HSTrail, and LP signaling. No element stream is associated with this command.
CLK_ON	If the clock is off, turns the clock on according to the DPhy specification and DPhy timing settings. Otherwise, this command has no effect.
CLK_ON	If the clock is on, turns the clock off according to the DPhy specification and DPhy timing settings. Otherwise, this command has no effect.
IF <flag>	In conjunction with ENDIF, allows blocks to be enabled or disabled for parsing. Nesting of the command is allowed.
ENDIF	In conjunction with IF, allows blocks to be enabled or disabled for parsing. Nesting is allowed.
FILE <filename>	Parses the commands in the given text file, inserting them into the current stream. Nesting of this command is allowed. Individual component commands and their associated data <i>may not</i> span multiple files.
<lg>	Name is shorthand for “lane group”. Consists

	of one of the following symbols: ACT 0 1 2 3
<lgwd>	Name is shorthand for “lane group with demux”. Consists of one of the following symbols: ACT 0 1 2 3 DEMUX
<dur>	Integer duration in ns. Value is quantized to a multiple of the HS Bit Rate and must be greater than or equal to 40 ns.
<flag>	Integer value used for boolean function: 0 == false, 1 == true
<filename>	String value containing a file path. If the file name is relative, its path is considered to start from the same directory as the text file containing the command using the argument.
[]	Arguments in brackets are optional.

5.10.2 Lane Group Arguments

This section describes the meaning of the lane group arguments.

ACT

This argument causes the element stream associated with the command to be duplicated and assigned to each active lane.

For example, if lane count is two, the following command assigns the same three-byte stream to lanes 0 and 1:

```
# HS_BYTES ACT
AAh BBh CCh
```

DEMUX

This argument causes the byte stream associated with the command to be demultiplexed one byte at a time across active lanes. For example, if lane count is three, the following command:

```
# HS_BYTES DEMUX
1 2 3 4 5 6 7
```

assigns bytes to lanes as follows:

```
lane 0: 1 4 7
lane 1: 2 5
lane 2: 3 6
```

To support this function, the parser maintains a current lane index, which is initialized to zero when transitioning from LP to HS. Each HS_BYTES command using the DEMUX

argument updates the index so after the example command, the current lane index would be set to 1. Thus, if the command was issued twice, the overall lane byte sequences would be:

```
lane 0: 1 4 7 3 6
lane 1: 2 5 1 4 7
lane 2: 3 6 2 5
```

and the current lane index would be set to 2. Be aware that the current lane index is not updated if the DEMUX argument is not used so use caution when mixing HS_BYTES commands with different <lgwd> arguments.

0, 1, 2, 3

These arguments designate a lane assignment for the element stream associated with the command. If the lane is not an active lane, the command is ignored. If this argument form is used, identical commands with the same sequence length must be present in active lane order.

For example, if lane count is four, a legal command sequence would be:

```
# HS_BYTES 0
1 2 3 4
# HS_BYTES 1
ah bh ch dh
# HS_BYTES 2
11 12 11 12
# HS_BYTES 3
a3h b2h c1h d0h
```

Note this command sequence would also be legal even if lane count < 4, though inactive lane commands would be ignored. For example this would allow a 4-lane sequence to be defined that can also be sent when lane count is less than 4.

5.10.3 LP_STATES Component Command

The LP_STATES command has two optional arguments. If the first argument is “ACT”, then data values are interpreted as singular LP states (0-3) which are duplicated and applied to all active lanes (inactive lanes remain in LP11).

If the first argument is not “ACT”, then data values are interpreted as 10-bit values, with each 2-bit pair representing the LP state for its associated lane (bits 1..0 => lane 0, bits 3..2 => lane 1, etc). The clock lane bits are set via bits 9..8, unless the clock is on, in which case these bits are ignored.

A second optional argument, <dur>, indicates an integer duration in nanoseconds for each LP state in the sequence. If this argument is not present, 1/LPFreq is used for the duration.

For example, the following represents the HS burst entry sequence on all active lanes at 10 MHz

```
# LP_STATES ACT 100
3 1 0
```

To output the HS burst entry on lane 0-2, regardless of the number of active lanes and using the current LPFreq setting, discard the ACT parameter and use 8-bit values that describe the LP state for each lane:

```
# LP_STATES
3ffh 3d5h 3c0h
```

5.10.4 HS Burst Requirements

One requirement for HS bursts implemented by the File command is that all active lanes must have the same number of HS data so that all lanes transition from HS to LP at the same time. This requirement must be explicitly ensured by the user or an error will occur while sending the File command.

Note if the HS_BURST_EXIT command is used, active lanes greater than or equal to the current lane index are allowed to have exactly one byte less than lesser active lanes. In this case, an additional byte of HS Trail is appended to these lanes to meet the alignment requirement.

A second requirement for HS bursts is that all active lanes must have an even number of HS bits (hardware requires all segments to be a multiple of 2 UI). Please keep this in mind when using the HS_BITS command. While odd bit counts are allowed in the command, the total number of bits per lane in the burst must be even.

5.10.5 File Command Examples

A common example is simply to send an arbitrary HSDT packet using the current lane count:

```
# HS_BURST_ENTRY

// Generic Long Write header with automatic ECC
# HS_BYTES_PLUS_ECC DEMUX
29h 06 00

// Payload with automatic CRC
# HS_BYTES_PLUS_CRC DEMUX
1 2 3 4 5 6

# HS_BURST_EXIT
```

Or if you want to specify the ECC and CRC yourself:

```
# HS_BURST_ENTRY

// Generic Long Write header with manual ECC
# HS_BYTES DEMUX
29h 06 00 23h

// Payload with manual CRC
# HS_BYTES DEMUX
1 2 3 4 5 6 f1h 47h

# HS_BURST_EXIT
```

To send a LPDT command:

```
// Escape entry (LP11, LP10, LP00, LP01, LP00)
# LP_STATES
3ffh 3feh 3fch 3fdh 3fch

// LPDT escape command
# LP_ESC_BYTES
87h

// LPDT packet header with automatic ECC
# LP_ESC_BYTES_PLUS_ECC
29h 06 00

// LPDT packet payload with automatic CRC
# LP_ESC_BYTES_PLUS_CRC
1 2 3 4 5 6

// Escape exit (LP10, LP11)
# LP_STATES
3feh 3ffh
```

5.11 PG File Command

A PG file is an encoded binary program file (with extension .pg) that can be constructed using the “Send Cmd To File” menu option (File menu). The “PG File” command is then used to directly download the compiled program contained in the PG file to the DPhy Generator for playback, bypassing potentially a long program encoding time. Thus, good candidate commands for saving to a PG file are long video sequences, DSC video sequences, and long macros with many commands.

Effectively, the “PG File” command is like the “Restart PG” button, except it is applicable to any prior saved compiled program not just to the last command sent.

Similarly, all configuration settings that affect program construction such as lane count, bit rates, frame timing, DPhy timing, options, etc. are embedded in the compiled program at the time the PG file is created. Thus, when the PG File command is sent, it is unaffected by current configuration settings that affect program construction¹⁷. And, with only two exceptions, the current configuration is unchanged when the PG File command is sent. The exceptions are that the “HS Bit Rate” and “Lane Cnt” settings *are* set to the values stored in the PG file when the PG File command is sent.

To create a PG file, first select the Cmd to send in the Cmd Name drop-down control or choose a Pkt Type and define a new command using the command definition controls. Then, select File -> “Send Cmd To File” and enter the PG file name in the dialog box that appears. The command program will be compiled and saved to the PG file.

Note that if the GUI option “Auto-define new “PG File Cmd” after “Send Cmd to File” operation” is checked (Options -> Set GUI Options...) a new PG File Cmd using the newly created PG file will be automatically created.

5.12 Command Buttons

The top-right pane of the main application window consists of four tab pages of 30 command buttons (labeled Cmds1, Cmds2, Cmds3, Cmds4). Clicking on the tab page header makes the 30-button command set available for use.

Command buttons can be associated with named commands by the user, allowing single-click sending of the named command on to the DPhy bus. When assigned with commands, buttons are labeled with their command name and enabled. Otherwise, they are disabled and show the label “<unassigned>”. Note that separate button pages are associated with DSI and CSI command sets, activated when the protocol is selected using the Standard menu.

Command buttons have an associated context menu, brought up by right-clicking on the button. The operations available from the button context menu are as follows:

- **Assign Current Cmd:** assigns the currently selected command in the Cmd Name drop-down control to the button
- **Assign Cmd:** assigns a named command selected from a sub-menu to the button
- **Remove Assignment:** removes the current command assignment from the button
- **Remove All Assignments on Current Cmd Page:** removes all command assignments from all 30 buttons on the button page.
- **Remove All Assignments on All Cmd Pages:** removes all command assignments from all button pages

¹⁷ Though, the PG File command is still affected by two options that affect program behavior -- “Disable Event Timeout” and “Enable Command Insertion” – as well as all run-time settings in the instrument configuration dialog such as voltages, delays, etc.

- **Set Button Tooltip:** sets a tooltip string to display when the mouse hovers over the command button. To clear the tooltip, set the tooltip to the empty string. Note that removing a button assignment also clears the tooltip string.
- **View Current Cmd:** fills in the argument field controls with values associated with the command. For a macro, argument fields for the first component command in the macro are shown. In addition, the Macro tab is selected and the macro command definition is shown as read-only.
- **Edit Current Cmd:** for non-macro commands, this command is identical to View Current Cmd. For macros, in addition, the macro command definition is brought up in edit mode.

Note that, unlike application settings such as option settings, DPhy timing definitions, frame timing, and instrument configuration settings, command definitions and button assignments are *not* automatically saved when the application closes. Instead, command definitions and button assignments must be saved explicitly to a configuration file using the Save Cfg... menu option (File menu).

5.13 Operation

5.13.1 Sending a Command

Once the DPhy Generator instrument, frame timing, and DPhy timing configurations are set and desired options are checked, commands may be sent to the DPhy Generator for output. Unnamed commands are sent after selecting the packet type and filling in the command arguments simply by clicking the Send button. Named commands, once defined, are sent by selecting the command name from the Cmd Name drop-down control and clicking the Send button. Alternatively, if a named command has been assigned to a button, left-clicking the button sends the command to the DPhy Generator for output.

After the user sends a command, messages in the status bar updates progress. During this time, DPhyGenCtl performs the following tasks:

- The command is parsed and any necessary external file data imported (and decoded in the case of BMP files)
- MIPI protocol packets are built along with any necessary signaling structures (i.e. SOT, EOT, BTA, etc).
- Packet data is packed into internal data records and a play sequence is constructed for the DPhy Generator to use during output.
- Data records and play sequence are downloaded to the instrument.
- The internal PG of the instrument is “Run”, initiating output, with control returning to the user at this point. Sending of the command is complete.
- However, if the command is a looping command, or contains event wait component commands, the DPhy Generator continues to run. Information in the status pane of the main window updates on-going operation.

5.13.1.1 Stop PG

As described above, the sending of commands goes through several stages of processing, some of which may take some time. Program download, in particular, can take tens of seconds to complete for video commands, when large blocks of video data are sent. DSC frame compression also can be time-consuming, depending on the frame size. During this time, the user may click the “Stop PG” button (located in the bottom-right pane of the main window) to abort command sending. In addition, once a command is running and the PG is actively looping (or waiting on an event), “Stop PG” may be clicked to terminate operation.

5.13.1.2 Restart PG

The “Restart PG” button in the bottom-right pane of the main window is used to resend the previous command, bypassing the potentially lengthy download process. This is useful, in particular, for video commands. When the PG is restarted, all timing configuration and parameters such as frequency, lane count, and clock start/stop mode from the original command are used (ignoring any changes since the command was sent).

5.13.2 Status

During operation, additional status is displayed in the lower-right pane of the DPhyGenCtl main window. This status is updated every second and includes the following:

- The PG run state (running or idle)
- The memory use of a running program (as a percent available in the instrument)
- The HS Bit Rate of a running program.
- The data lane status (enabled, ULPS, waiting on event, etc)
- Whether any LP contention has been detected since the last status reset
- Read response data received from the DUT after a BTA.¹⁸

5.13.2.1 Contention Detection

The DPhy Generator is equipped with contention and data receivers on lane 0, whose thresholds can be set by the user in the Instrument Configuration dialog (see section 5.4.2 for more details). When the instrument drives an LP high voltage but detects a voltage below the LP High contention threshold, an LP high fault is detected. When the instrument drives an LP low voltage but detects a voltage above the LP Low contention threshold, an LP low fault is detected.

As these faults can occur on either LP signal wire this results in four possible contention states. The mnemonics displayed in DPhyGenCtl for contention detection is as follows:

¹⁸ Note: read response data is only displayed after a running program has stopped or the user explicitly requests that read response data should be read from the instrument with the “Get DUT Resp” button.

- **LP-Tx0** – contention detected on the LP- wire of lane 0 when transmitting LP0
- **LP+Tx0** – contention detected on the LP+ wire of lane 0 when transmitting LP0
- **LP-Tx1** – contention detected on the LP- wire of lane 0 when transmitting LP1
- **LP+Tx1** – contention detected on the LP+ wire of lane 0 when transmitting LP1

Once contention is detected, flags are held until a subsequent command is sent or the Status Reset button is pressed to clear state. Note also that the aggregate contention state is reflected on the Evt1 signal of the instrument back-panel (see section 2.4), allowing for instrument trigger on the onset contention.

5.13.2.2 DUT Response

When a command or macro containing a BTA is sent, the DPhy Generator monitors the link for an LPDT response command from the DUT. The DUT response (if any) is displayed in the status pane of the main window as the string “DUT Resp: “ followed by the first number of hex data bytes. If the response contains more than 12 bytes, a button labeled with an ellipsis (“...”) appears so the user can click to view the entire response data in a dialog. Note that if the program is looping, DUT response data *does not appear* until the program is stopped or the user clicks the “Get DUT Resp” button.

To convey the LP signaling states transmitted by the DUT in generating its read response, an extra partial byte of data is included before the actual DUT response message. In particular, the first byte of each response conveys the initial two bits seen during escape entry (LP10, LP00, LP01, LP00), which is interpreted as binary ‘1’ followed by binary ‘0’ or equivalently a 0x1 data byte.

In addition, a special flag (0x100) is set in this value to indicate it represents a new escape-entry command. This allows for multiple responses to be queued and subsequently easily parsed by the user. Thus, to summarize, the first value displayed for each escape entry command should be 0x101.

Subsequent bytes displayed after escape-entry, represent the command data, starting with the escape command, which is expected to be 87h for an LPDT command. Thus, the first header byte of each DUT response packet will be conveyed in the second byte of the displayed DUT response data following each 0x101 value.

The DUT response buffer is reset every time a DPhyGenCtl command is sent, but this could consist of a macro with multiple read commands. The response buffer is relatively large (4 Kbytes) and thus can capture a significant amount of data from multiple reads, which are simply concatenated in the buffer.

Example:

During video-mode, say the user inserts four read commands over the course of a test and their responses are queued. When “Get DUT Resp”

button is pressed and the ellipsis button is pressed, the DUT response bytes might look like those in Figure 13:

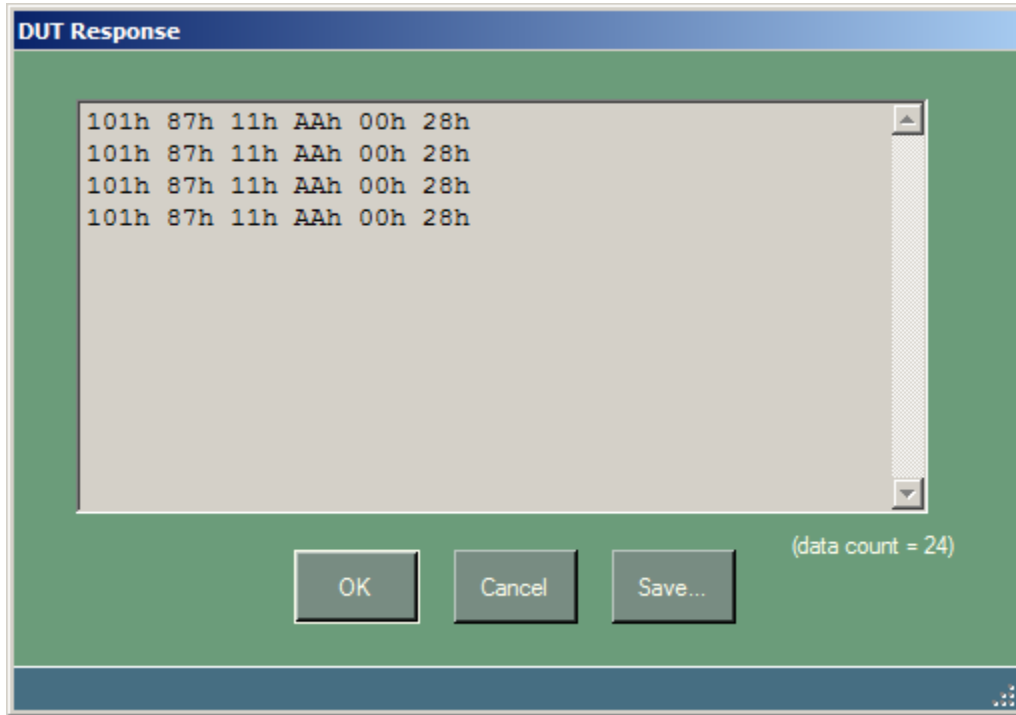


Figure 13 – DUT Response Dialog

Each response is delimited by the escape-entry value of 101h. This is followed by 87h for the LPDT command byte and then a DataID of 11h, indicating a generic short read response packet.

Note, there are two RPC commands that can be used to obtain the DUT response:

- GET_DUT_RESPONSE can be used programmatically to obtain the last response from the DUT.
- SAVE_DUT_RESPONSE can be used either in a script file or programmatically to save or append the last response from the DUT to a text file.

Please see the DPhyGenCtlRPC manual for more information.

5.13.3 Command Insertion

Command insertion is the ability to insert commands into an actively looping stream, whether in a video-mode command or macro. While video-mode commands automatically define an insertion point in the first vertical blanking line of a frame, the user may also explicitly define an insertion point in a macro, denoting the location where the inserted command should output. The “Cmd Insertion Point” command is used for this purpose.

The following describes command insertion behavior:

- Command insertion is enabled using the “Enable Command Insertion” option. Checking this option means that a command sent while a program is running is intended for insertion.
- When command insertion is enabled, to send a subsequent command but not insert it, either first click the “Stop PG” button or uncheck the “Enable Command Insertion” option.
- An inserted command may be sent in HSDT or LPDT mode and inherits the LPFreq, HS Bit Rate, and LaneCnt settings used in the looping program (regardless of possibly edited GUI control settings).
- If the inserted command requires more time than that reserved by the Cmd Insertion Point (via its Delay parameter), the looping program is suspended until the inserted command completes.
- When a command is inserted, DPhyGenCtl checks to see if it can detect the insertion command code in the output stream. If it is not detected within ½ second, an error is reported. To disable this check, select the “Disable command timeout” option. This option should be checked if the “Wait on external event to start” option is enabled or the macro contains a “Wait Ext Event” command.
- If a command is inserted when the PG is waiting on an external event (and the command timeout is disabled), the inserted command is queued to be sent when output resumes and the insertion point is reached.
- If a macro contains multiple command insertion points, an inserted command is output at the next encountered insertion point. Unless sequencing is controlled via external events, the user has no practical control over which command insertion point is used for the inserted command.

5.13.4 Controlling TrigOut

The DPhy Generator has a TrigOut front-panel signal that can be controlled both through the GUI and stream commands. It can be used as a scope or DUT-related triggering event.

The GUI main window has a “Trig Event” button that generates a pulse on the TrigOut signal when pressed. The event occurs whether or not the PG is running. However, note that the front-panel TrigOut *does not* occur if stream output is stalled waiting on an event. The duration of the pulse is configured in the Instrument Configuration dialog via the “Trig Pulse Width” control.

Also, the “Assert Trigger” command, which can be embedded in a macro, allows full control and timing of the TrigOut signal, allowing direct setting of the signal level (high or low), toggling state, or generating a pulse sequence (see section 5.6.6.15).

The TrigOut signal is duplicated (with some differences in behavior) on the back-panel GPO0 signal (see section 2.4 for back-panel description). Like the front-panel TrigOut,

the back-connector signal is controlled both from the “Assert Trigger” command and the “Trig Event” button of the DPhy Generator GUI. However, the back-panel TrigOut has two differences in behavior from the front-panel TrigOut:

- 1) It does not have even nearly synchronous timing with the command in the stream output when controlled via the “Assert Trigger” command, occurring well before command output. The time difference is somewhat variable and is on the order of microseconds.
- 2) Unlike the front-panel Trig-Out, it supports “Trig Event” output even when output is stalled waiting on an event.

5.13.5 External Event Triggering

The GPIO input on the back-connector of the DPhy Generator pin (see section 2.4) can be used as an external event signal for controlling output flow control. The triggering event on this signal is a rising edge (logic level ‘0’ transitioning to a ‘1’). When detected, the latency of the event signal is bit-rate dependent, but generally is on the order of 100 ns.¹⁹

Currently, there are three ways this signal can be used:

- 1) To control when command output begins
- 2) To control frame advance during video sequence playback
- 3) To control when component commands in a macro are output

These ways are described in the next sections.

5.13.5.1 Controlling When Command Output Begins

The option “Wait on external event to start” (Options menu) allows users to configure the PG to wait for an external trigger event before command output begins (including video-mode commands). When this option is enabled, commands that are sent do not immediately output on the DPhy bus. Instead, while the stream is prepared and downloaded to the instrument as usual, output is held off until the external event is detected.

5.13.5.2 Controlling Frame Advance During Video Mode

The option “Advance Frame On External Event” in the Frame Timing dialog (see section 5.7.3.6) allows video-mode to be configured such that each frame loops indefinitely until the external event is detected. When the event is detected, output moves to the next frame in the sequence.

5.13.5.3 Controlling Macro Component Command Output

The command “Wait Ext Event” (section 5.6.6.14) can be used to pause output in a macro until the external event is detected. When the generator is paused, the DPhy bus is held in a static LP state as specified by the user.

¹⁹ However, the *effective* latency for the external input event when used for advancing video frames is much longer (one or two frames) due to the video program and pipeline architecture.

Note that the “Disable Command Timeout” option (Options menu) applies in this case. If this option is checked, the instrument pauses and waits indefinitely for the external event. Otherwise, there is a static timeout period of one millisecond after which the program continues automatically.

5.13.5.4 Triggering the External Event Signal Via Software

Note that the HostEvent button in the main window can be used to cause the External Event signal (i.e. GPIO) to fire. This allows the user to test without external hardware connected or otherwise simply use a GUI button (or RPC command) for program flow control.

5.13.6 Wait Events

During program output, there are several events that cause output to suspend (in an LP state) until the event fires. They are:

- 1) **BTA Ack:** triggered by the DUT responding with the proper LP acknowledgement sequence immediately following a BTA. The default timeout associated with this event is 1 us and is not user-configurable.
- 2) **BTA Response:** triggered by the DUT eventually responding with a proper BTA response sequence after receiving the bus via a BTA. The default timeout associated with this event is 20 us but can be set by the user in the Instrument Configuration dialog.
- 3) **External Event:** triggered by a rising edge on the GPIO input pin on the P344 back connector. The default timeout associated with this event is 1 ms and is not user-configurable.
- 4) **Host Event:** triggered in software via the HostEvent button of the main GUI window. There is no timeout associated with this event.

Actions that are associated with one or more of these events are shown Table 7:

Table 7 – Action/Event Association

Action	Associated Event(s)
Sending a MIPI packet with it BTA parameter set to “Yes”	BTA Ack, BTA Response
Sending a “Send BTA” command	BTA Ack, BTA Response
Sending a “Wait For BTA” command	BTA Response
Sending a “Wait Ext Event” command	External Event
Setting the “Wait on external event to start” option	External Event
Sending a ULPS command	Host Event

Most events as noted in their description have timeouts. Once program output has been suspended waiting on an event longer than its timeout, program output automatically resumes as if the event occurred.

If timeouts are not desired, for example during low-level testing, they can be disabled via the option: “Disable Event Timeout” (Options menu). When event timeouts are disabled, another feature of the software may prove useful. In particular, the Host Event button can be used to force any event causing program suspension, allowing simulation of the event.

5.13.7 Causing Packet Errors

The DPhyGenCtl GUI supports the ability to modify a packet's ECC and CRC fields to cause receive errors in packets. Normally, when a packet type is selected for a command in the main window and its parameters set, read-only fields are displayed showing the DataID and ECC for short packets and, additionally, the WordCount and CRC for long packets. Checkboxes next to the ECC and CRC fields allow user modification of these fields to cause a receive error in the packet.

To change the ECC or CRC, simply check the associated checkbox, making the field value editable. When the control is unchecked, the correct ECC or CRC value is recomputed and the field is restored back to its original read-only state.

This capability applies to all CSI/DSI commands, including those within macros, except for video-mode commands. For video-mode commands, there is no ability for the user to set the ECC or CRC of any of the video frame packets. However, payload data errors can be introduced using the “Video Error” field (see section 5.7.2.2).

When applied to the Write Memory commands, if the command is partitioned into multiple Write Memory commands, the user-set ECC and/or CRC applies only to the first Write Memory command. Remaining Write Memory commands are always built with their correct ECC and CRC.

An additional mechanism for introducing packet errors is described in the description of RPC. In addition to allowing the ECC and CRC to be set, the SEND_IMPAIRED_MIPI_CMD RPC call allows any byte in a packet to be XORed with an error mask before it is sent. Please see the description of this command in the DPhyGenCtlRPC manual for more details.

5.13.8 Script Recording

Script recording is a mode in which MIPI commands sent to the DUT are also written as RPC commands to a designated script text file. All commands, including macros and video-mode commands are recorded. After recording, the script file can be sent as any other script file or used as a template for creating additional script files.

In addition, the current configuration state can be written to the script file. This allows the script file to initialize DPhyGenCtl to a particular configuration or be sent to the Moving Pixel Company to document your configuration when requesting assistance.

To enable script recording, select “Start Recording” from the Script menu. When this option is selected, a browse dialog is opened for you to designate the text script file name to use for recording.

Once script recording is enabled, the status bar displays the text “Recording” in a new pane. The script file is initialized with a header containing information about the current software and hardware configuration.

Any MIPI command sent (without error) during recording also causes an RPC command to be appended to the script file. In addition, selecting the menu option “Write Current State” from the Script menu causes the current state to be written to the script file. To end script recording, select “End Recording” from the Script menu.

5.13.9 Updating Firmware

Periodically, new firmware releases may be issued for the DPhy Generator. The latest firmware release can be downloaded from The Moving Pixel Company's website and (after unzipping) is labeled DPhyGenFirmware_VerX_Y.rbx, where the X and Y represent version and release digits respectively.

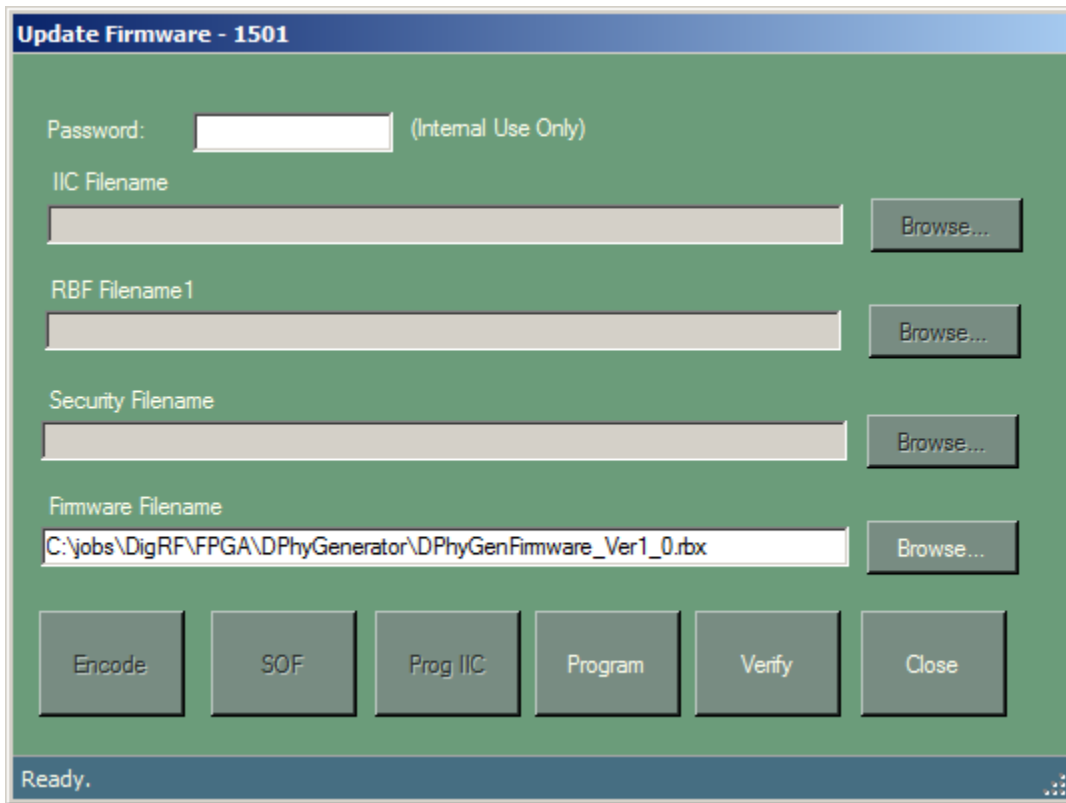


Figure 14 – Firmware Dialog

To update firmware in the DPhy Generator, perform the following steps:

- Connect to the instrument normally. If for some reason the instrument's firmware has been corrupted and is non-responsive, it is important in the Connect dialog to check the "Firmware update only" checkbox to prevent interaction with the FPGA.
- Select the Connect->Update Firmware.... menu option to bring up the firmware dialog.
- Click the "Browse" button next to the "Firmware Filename" to select the RBX firmware file on your machine, obtained from the Moving Pixel Company.
- Click the Program button.
- Programming takes a couple minutes, with status updating progress.
- When programming is complete, close the application and power-cycle the instrument. On power-on, you can watch the instrument LEDs for the appropriate sequence: i.e. the red and yellow LEDs alternate for a couple seconds ending with the green LED remaining solidly on.

5.14 Customizing GUI Colors

In the DPhyGenCtl, the user can configure the colors of various elements of the GUI (e.g. the background and foreground colors of buttons, forms, menus, etc). This is achieved using the Color Options dialog, brought up via the Options menu by selecting the "Set Colors" menu item.

The Options dialog (see Figure 15) consists of a column of buttons on the left showing the current color settings of GUI controls. In the center of the dialog is a color square showing all the colors associated with a given luminance. Clicking and dragging the mouse inside of the color square selects the color of the currently selected control.

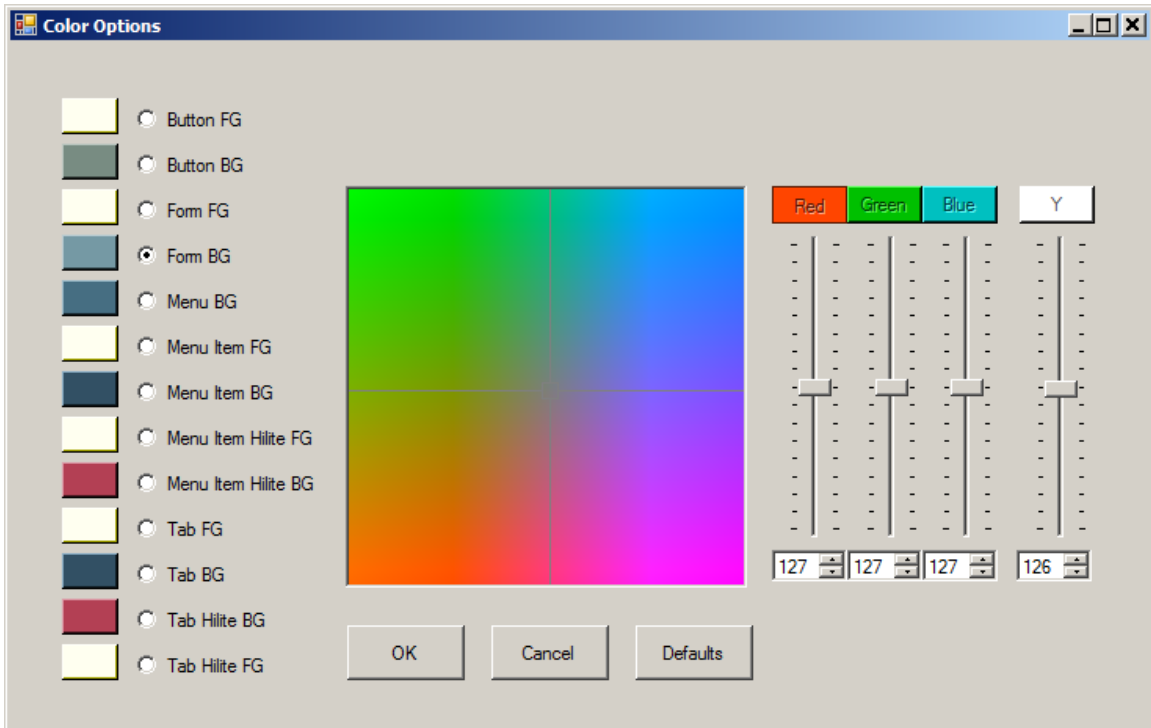


Figure 15 – Color Options Dialog

The luminance can be changed by adjusting the slider labeled ‘Y’. Alternately, you can change the R, G, B color components using the sliders on the right of the color square. Yet another way to change the color is to click on a control button color (far right column). This loads the selected color as the current color. For example, to set the Button foreground the same as the Form foreground, click the Button FG radio button and click the color button next to the Form FG button.

As the control color changes, the main window colors changes in real time, so you can see how colors interact. To restore the default colors, click on the Defaults button. When you are done changing colors, to accept the new colors, click the OK button. To discard your color changes, click the Cancel button. Color settings are saved on application exit and restored when the application is restarted.

5.15 GUI Options

The Options->Set GUI Options... menu item brings up a dialog to selectively enable/disable common warnings and configure behaviors in the application (see Figure 16). Options are self-explanatory.

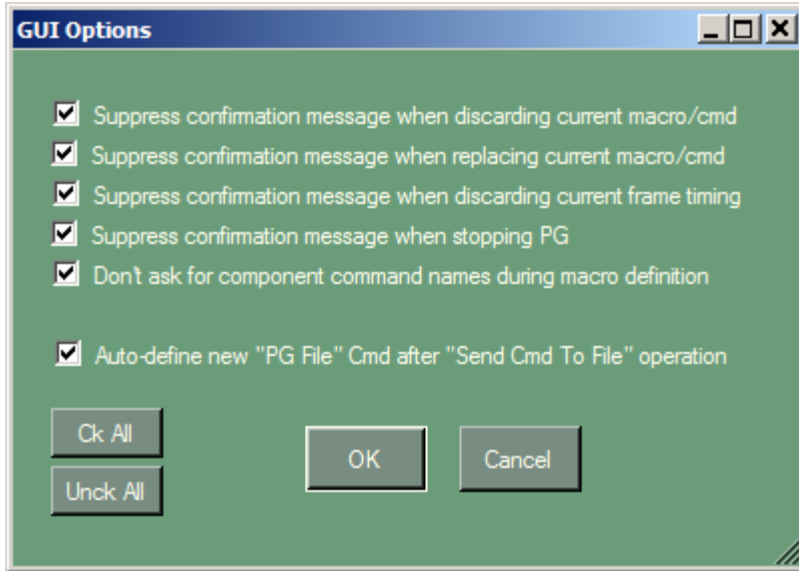


Figure 16 – GUI Options Dialog

5.16 Keyboard Shortcuts

A few keyboard shortcuts are available when the main window is active. A dialog listing them can be brought up using the Ctl-K key sequence (see figure):

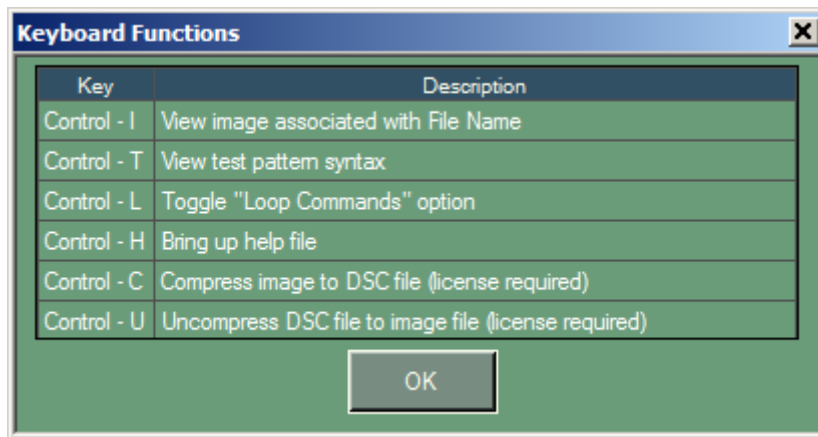


Figure 17 – Keyboard Function Dialog

Note that the view image and DSC compression functions apply to the current File Name for the current command.

5.17 Menus

This section outlines the menu commands available in DPhyGenCtl:

Menu	Description
FILE MENU	

Load...	Loads a previously saved command configuration file, overwriting any current commands and button assignments.
Save <fn>...	Saves the current command definitions and button assignments from both the CSI and DSI protocol sets to the current configuration file name. Note that the command configuration file does NOT store application settings, including options, DPhy configuration, PG configuration, and frame timing configuration settings.
Save <fn> As...	Same as the Save command above except a dialog appears to browse for and enter a new configuration file name.
Send Cmd To PG File...	Builds a compiled binary PG file containing the program associated with the current command. The PG file can then be subsequently sent using the PG File command (see section 5.11).
<recent files>	Lists the four most recent configuration files. Selecting one of these file names loads the file.
Clear Recent File List	Clears the most-recent-file-list.
Exit	Exits the application.
<u>CONNECT MENU</u>	
Connect...	Brings up the PG connection dialog (see section 5.3).
Disconnect	Disconnects from the current instrument, putting the application in offline mode.
Enable RPC...	When checked, enables the application to accept and process incoming RPC requests (see the DPhyGenCtlRPC manual for more information).
Update Firmware...	Updates DPhy Generator firmware (see section 5.13.9).
<u>STANDARD MENU</u>	
DSI	Selects the DSI command set and button assignments.
CSI	Selects the CSI command set and button assignments.
<u>OPTIONS MENU</u>	
Loop Commands	When checked, causes commands (including video-mode commands and macros) to be looped indefinitely. Looping can be stopped with the Stop PG button.
Wait on external event to start	When checked, configures the PG to wait for an external event on its Evt1 input before outputting a sent command. When the event input is a logic '0', the PG waits until it goes to '1' before continuing (see section 5.13.4 for more information). Note that the external event can also be used to control macro output using the "Wait Ext Event" packet type. See section 5.13.4.
Disable event timeout	When checked, disables the normal timeout for wait events during program output, such as for a BTA response, BTA acknowledgement, and external events. This option is

	particularly useful for macros that include SEND_BTA, BTA_WAIT or WAIT_EXT_EVENT commands, which can cause arbitrary delays based on the testing environment and the behavior of the device under test. See section 5.13.6.
Enable command insertion	When checked, sending a non-video command while a video-mode command is looping causes the packet to be inserted (once) into the vertical blanking interval without stopping video playback. If unchecked, sending any command while a video-mode command is looping causes DPhyGenCtl to confirm that the user wants to stop video playback to send the command. This function also works with a looping macro containing a command insertion point. See section 5.7.4.1.
Enable Video Mode In Macros	When checked, sending a video-mode command when defining a macro inserts the video frame into the macro. If unchecked, only a single video-mode packet is added to the macro.
Send single packet per HS burst	When checked, macros that contain consecutive HS packets will be sent in individual HS bursts (i.e. bracketed by the HS burst entry/exit sequence). Otherwise, consecutive HS packets will be sent in a single HS burst.
Allow Image Rescaling	When checked, this option causes images specified in a Write Memory or video-mode command to be rescaled to the current output frame dimensions before conversion to the appropriate pixel format. Output frame dimensions are specified by the HActive and VActive fields of the Frame Timing configuration. If this option is not checked, an error is thrown when a command is sent whose image argument does not have the current output frame dimensions.
Turn Clk On When Sending Non-Video Cmds	When checked, the clock is automatically turned on before a non-video command is sent. Otherwise, the clock remains off unless required by the command sent (i.e. Clock On, HS command, video-mode command in macro). See section 4.3.
Turn Clk Off After HS Burst (Non-Video)	When checked, the clock is automatically turned off after an HS command is sent in a macro. See section 4.3.
Enable EoT packets	When checked, an EoT packet is automatically appended to the end of an HS burst.
Encode RAW format as Bayer...	When checked, a dialog is shown to select one of four Bayer patterns to encode CSI Raw video-mode data: GRBG, RGGB, BGGR, GBRG. The first two letters indicate the alternating colors for the first and every other line. The second two letters indicate the alternating colors for the second and every other line. If unchecked, Raw video-mode data is encoded as gray-scale.
Configure DPhy timing...	When selected, the DPhy Timing Configuration dialog, is shown, allowing the user to enter specific DPhy timing parameters such as HSPrepare, HSZero, HSTrail, etc (see

	section 5.5).
Configure Write Memory Commands...	When selected, the Config Write Memory dialog is shown, allowing the user to configure how a DCS Write Memory command should be partitioned into multiple Write Memory commands (see section 5.6.6.10).
Configure DSC...	When selected, the DSC Configuration dialog is shown, allowing the user to specify the parameters of DSC compression (see section 5.7.5).
Set GUI Options...	Brings up a dialog for setting GUI options, allowing the user to selectively enable/disable command warning messages (see section 5.15).
Set Colors...	Brings up the Color Options dialog, allowing the user to change colors of various control elements in the GUI.
<u>SCRIPT MENU</u>	
Start Recording...	Brings up a file dialog to select an output text file to save recorded script commands. Then, the application is put in script recording mode, which logs and outputs MIPI commands to the script file as they are sent to the DUT. See section 5.13.8.
End Recording	Ends script recording.
Write Current State	Writes the application configuration state as RPC commands to the current script recording file (this option is only enabled during script recording).
<u>ABOUT MENU</u>	
Help...	Displays this manual as a help file.
Help RPC...	Displays the DPhyGenCtlRPC manual as a help file.
About...	Brings up a summary window displaying the current software version. This dialog also includes a summary of release notes of changes/fixes for each version

6 DPhyGenCtl RPC (Remote Procedure Calls)

To facilitate automated testing and remote control, DPhyGenCtl can respond to incoming RPC requests from other applications via a Microsoft .NET TCP server port. To support this function, a separate .NET DLL called DPhyGenCtlRPC.DLL is provided. Any programming language that can interface to a .NET DLL can use this automation capability.

Alternatively, RPC commands have a script form that can be used in a text script file and run from the DPhyGenCtl GUI. In this case, no external programming is required. Many customers use script files extensively as they are a very powerful and convenient tool to configure and control DPhyGenCtl and the DPhy Generator.

Two separate documents have been written to detail the RPC communication interface, supported commands, and example client code provided with installation of the DPhyGenCtl application. The documents are located in the application installation directory with shortcuts provided in the Start Menu. They are:

- DPhyGenCtlRPC_<w>_<x>.pdf – provides description of RPC: its usage, operation and calling structure.
- DPhyGenCtlRPCCommands_<y>_<z>.xls – summary and description of all commands in the DPhyGenCtlRPC.DLL.

where <w>, <x>, <y> and <z> are the document version and release numbers. Please refer to this document for more information.