

This file documents RPC calls supported by the DPhyGenCtl software version 2.4 (and above) and DPhyGenCtlRPC.dll version 1.7 (and above). To make an RPC call, a routine in DPhyGenCtlRPC.dll is used. This DLL is a managed .NET DLL and can be called easily from any .NET language. Other languages such as Labview, Python, Matlab, etc. can be used, but knowledge of how to interface with a .NET dll via COM is required. On the server side, the DPhyGenCtl software must have the RPC option enabled (Connect->Enable RPC menu option) and both the client and server must use the same IP port number (any unused port number on the host up to 65535 can be used).			
Before any RPC calls can be made, connection to the server is required using the Connect routine. After successful connection, any RPC command or query can be sent. MIPI commands are a special category of RPC command and are sent using special functions in the DLL: MIPICmd, AddMIPICmd, ImpairedMIPICmd, AddImpairedMIPICmd. Arguments to these calls are described in the DPhyGenCtl RPC Manual.			
Non-MIPI commands, used for configuration and control of the DPhyGenCtl application, use two generic routines in the DLL that have several variants supporting zero, one, two, three, or four arguments. The first routine is RPCCmd, used for commands or queries that return a single integer. The second is RPCQuery which is used for queries that return a value other than an integer. Which routine and which variant should be used is dependent on the parameters and return value of the specific command.			
Note that the RPCCmd and RPCQuery routines have generic arguments, a concept supported in .NET that allows parameters of any type to be passed in. For non-.NET users of the DLL, these functions cannot be used. Instead, non-polymorphic, non-generic forms of these routines are provided, with distinct names to indicate their parameter types. For example, RPCCmdI is a single-integer argument command that returns an integer. RPCCI is a dual-integer argument command that returns an integer. RPCQueryIRetS is a single-integer argument command that returns a string. And so on.			
Two classes are defined in the DLL: RPCCmds, which contains the command values for all RPC commands, and RPCDefs, which contains all enumerated values for parameters to RPC commands.			
An example Visual Studio project written in C# is installed with DPhyGenCtl in the application directory (usually Program Files (x86)/TMPC/DPhyGenCtl) called DPhyGenCtlRPCTest that includes examples of the usage of almost all RPC commands written in C#. The program is meant to be copied out of the Program Files directory and then run in the Visual Studio debugger, using breakpoints and stepping through the various commands, so as to see the effect commands have in the DPhyGenCtl GUI.			
DPhyGenCtl RPC Command Reference			
RPC Command	Description	DLL Call	Notes
<u>Command Management and Output</u>			
SEND_MIPI_CMD	Outputs a MIPI command on the DPhy bus. See User's Manual for argument description.	MIPICmd	Note 0
ADD_MIPI_CMD	Create a named command in DPhyGenCtl (will delete an existing command by the same name if it exists). See User's Manual for argument description.	AddMIPICmd	Note 0
SEND_IMPAIRED_MIPI_CMD	Outputs an impaired MIPI command (i.e. command with error) on the DPhy bus. See User's Manual for argument description.	ImpairedMIPICmd	Note 0
ADD_IMPAIRED_MIPI_CMD	Create a named impaired command in DPhyGenCtl (will delete an existing command by the same name if it exists). See User's Manual for argument description.	AddImpairedMIPICmd	Note 0
SEND_NAMED_CMD_FROM_NAME arg1 = command name (string)	Given a command name, output the command on the DPhy bus.	RPCCmd	Note 0
SEND_NAMED_CMD_FROM_INDEX arg1 = command index (int)	Given a command index, output the command on the DPhy bus. The command index is the position of the command in the command list.	RPCCmd	Note 0

SELECT_NAMED_CMD arg1 = command name (string)	Given a command name, select the command in the GUI Cmd drop-down box. Useful for indicating the video format before setting the HS_BIT_RATE in the Frame Timing dialog (i.e. via SET_TIMING_BIT_RATE).	RPCCmd	
DELETE_NAMED_CMD arg1 = command name (string)	Given a command name, delete the command from the command list.	RPCCmd	
DELETE_ALL_NAMED_CMDS no arguments	Clear the command list.	RPCCmd	
SET_SORT_CMDS arg1 = enableFlag (int, 0 = disable, 1 = enable)	Set the flag which determines whether the command list is sorted alphabetically when commands are added. In addition, if the flag is set true, the current command list is sorted.	RPCCmd	
GET_NAMED_CMD_COUNT	Gets the number of commands in the command list.	RPCCmd	
GET_NAMED_CMD_NAME arg1 = command index (int)	Given a command index, return the command name.	RPCQuery	
<u>Instrument Configuration</u>			
START_EDIT_CONFIG no arguments	In conjunction with END_EDIT_CONFIG, this command is required and used to bracket one or more instrument configuration commands. Opens the Instrument dialog in DPHYGenCtl.	RPCCmd	
END_EDIT_CONFIG no arguments	In conjunction with START_EDIT_CONFIG, the command is required and used to bracket one or more instrument configuration commands. Closes the Instrument dialog in DPHYGenCtl.	RPCCmd	
SET_DT_MODE arg1 = DTMode (enum, DT_x defines)	Sets the default DT_MODE setting for MIPI packets.	RPCCmd	Note 1
SET_LANE_CNT arg1 = lane count (int, 1-4)	Sets the number of active data lanes.	RPCCmd	Note 1
SET_LP_FREQ arg1 = LPFreq (float, Hz, 0.2-30 MHz)	Sets the LP frequency. Note that usage always quantizes LP periods to a multiple of 2*UI.	RPCCmd	Note 1
SET_HS_BIT_RATE arg1 = HSBitRate (float, bps, 32 Mbps - 6.0 Gbps)	Sets the HS bit rate.	RPCCmd	Note 1
SET_FORCE_TEST_PATTERN arg1 = enableFlag (int, 0 = disable, 1 = enable)	Forces an HS clock pattern on all clock/data lanes for low-level testing.	RPCCmd	Note 1
SET_AUTO_SET_CLOCK_DELAY arg1 = enableFlag (int, 0 = disable, 1 = enable)	Enables automatic adjustment of the clock delay to an optimal (quadrature) setting whenever the clock frequency is changed.	RPCCmd	Note 1
SET_ALL_HS_VOLT_COMMON arg1 = enable (int, 0 = disable, 1 = enable)	Links all HS voltages to the same value as lane 0.	RPCCmd	Note 1
SET_HS_LOW_VOLT arg1 = lane (int, 0-4, clk == 4) arg2 = voltage (float, V, -0.6 to 1.2)	Sets the HS low voltage for the given lane.	RPCCmd	Note 1
SET_HS_HIGH_VOLT arg1 = lane (int, 0-4, clk == 4) arg2 = voltage (float, V, -0.6 to 1.2)	Sets the HS high voltage for the given lane.	RPCCmd	Note 1
SET_LP_LOW_VOLT arg1 = voltage (float, V, 0.45 to 1.8)	Sets the LP low voltage for all lanes.	RPCCmd	Note 1
SET_LP_HIGH_VOLT arg1 = voltage (float, V, 0.45 to 1.8)	Sets the LP high voltage for all lanes.	RPCCmd	Note 1

SET_LANE_DELAY arg1 = lane (int, 0-4, clk == 4) arg2 = delay (sec) (float, 0 - 4800 or 320 ps, based on HS Bit Rate threshold of 3 Gbps),	Sets the lane delay skew for the given lane.	RPCCmd	Note 1
SET_LP_LOW_CONT_THRESH arg1 = threshold voltage (float, V, 0.0 - 1.0)	Sets the LP low contention threshold.	RPCCmd	Note 1
SET_LP_HIGH_CONT_THRESH arg1 = threshold voltage (float, V, 0.0 - 1.0)	Sets the LP high contention threshold.	RPCCmd	Note 1
SET_BTA_WAIT_TIME arg1 = wait time (float, sec, 0.1 us - 100 ms)	Sets the timeout after a BTA has been sent to wait for a response BTA. After the timeout, the probe will continue its program as if a BTA had been received. Note that this value is ignored and the timeout is infinite if the "Disable Cmd Timeout" option is set.	RPCCmd	Note 1
SET_TRIG_PULSE_WIDTH arg1 = pulse width (float, sec, 0.1 us - 100 ms)	Sets the pulse width of trigger pulses.	RPCCmd	Note 1
SET_LANE_MAP arg1 = lane map setting (int)	Sets the output lane map, which routes data from logical lanes to physical output lanes on the instrument. Each nybble in the lane map setting represents the source lane to use for a physical output lane. Bits 3..0 map a logical data stream to physical output lane 0. Bits 7..4 map a logical data stream to physical output lane 1. Etc. For example, the default lane map is 43210h. Note that output lane 4 is the clock lane.	RPCCmd	Note 1
SET_LINK_TO_SLAVE arg1 = link flag (0 == don't link to slave, 1 == link to slave)	Sets flag to link slave instrument configuration to master settings in dual-instrument mode. If set, master instrument configuration settings are forwarded to the slave automatically. Otherwise, slave instrument configuration settings must be explicitly set (first using SET_RPC_TARGET).	RPCCmd	Note 10,11
<u>GUI Configuration</u>			
SET_OPTION arg1 = option (int, OPT_x defines) arg2 = 0 (disable), 1 (enable)	Sets or clears a GUI option.	RPCCmd	Note 9
GET_OPTION arg1 = option (int, OPT_x defines) retval = option value: 0 (disabled) or 1 (enabled)	Returns the current GUI option settings	RPCCmd	Note 10
SET_MIPI_STANDARD arg1 = standard (int, STD_x defines)	Sets the current MIPI standard.	RPCCmd	
SET_BAYER_ENCODE_TYPE arg1 = Bayer encode format (string, "GRBG", "RGGB", "BGGR", "GBRG")	Sets the Bayer encode type for CSI Raw images (when the Bayer encoding option is enabled).	RPCCmd	Note 9
SET_LOOP_COUNT arg1 = loop count (int, 0 - 8M, 0 == infinite)	Sets the loop count for command output.	RPCCmd	Note 10

SET_RPC_TARGET arg1 = (int, 0 == local/master, 1 == slave)	Sets the final destination for RPC commands when operating in a dual-instrument configuration. If the RPC target is set to the slave, all subsequent RPC commands are sent to and executed by the slave GUI until the RPC target is set back to the master. Note: RPC commands sent to the slave should generally only be status reads and instrument configuration writes (e.g. to set real-time voltage, delays, etc). See User's Manual for more information.	RPCCmd	Note 10,11
SET_ACTIVE_LINKS arg1 = active links (int, 0 == master, 1 == slave, 2 == master/slave)	Sets which links are active in a dual-instrument configuration.	RPCCmd	Note 10,11
SYNC_WITH_SLAVE arg1 = RPC port number (int, use 0 to disconnect) arg2 = slave inst serial number	Requests connection with slave GUI to run in a dual-instrument configuration. The slave GUI is launched (and hidden) with RPC enabled on the given port before connection proceeds. An instrument serial number is also be provided for the slave to connect to a DPhy Generator instrument (0 can be used for offline).	RPCCmd	Note 10,11
SHOW_GUI arg1 = window state (int, 0 == hide, 1 == show)	Shows/hides the GUI window	RPCCmd	Note 10
SHOW_SLAVE_GUI arg1 = window state (int, 0 == hide, 1 == show)	Shows/hides the slave GUI window	RPCCmd	Note 10
USER_WAIT arg1 = user message (string)	When used in a script, displays a message box with the given user message. User clicks on OK button to continue running script.	RPCCmd	
INST_CONNECT arg1 = instrument serial number (int)	Connects to the DPhy Generator instrument having the given serial number.	RPCCmd	
INST_DISCONNECT no arguments	Disconnects from the current DPhy Generator instrument. Puts the software in offline mode.	RPCCmd	
STOP_PG no arguments	Stops the (internal) PG of the DPhy Generator, returning all lanes of the Dphy bus to LP11.	RPCCmd	
RESTART_PG no arguments	Restarts the (internal) PG of the DPhy Generator, resending the last command that was output. (Does not require a re-download of the program to the instrument.)	RPCCmd	
LOAD_CONFIG arg1 = file name to load configuration (string)	Loads button and command configuration from the given configuration file, overwriting the current configuration.	RPCCmd	
SAVE_CONFIG arg1 = file name to save configuration (string)	Saves the current button and command configuration to the given configuration file.	RPCCmd	
LOAD_APP_CONFIG arg1 = file name to load configuration (string) arg2 = includeCfgInfo flag (0 = don't include, 1 = include)	Loads settings from the given application configuration file, overwriting the current settings. If the includeCfgInfo flag is set and the file contains a CFG info section (button and command definitions), they are loaded as well.	RPCCmd	
SAVE_APP_CONFIG arg1 = file name to save configuration (string) arg2 = includeCfgInfo flag (0 = don't include, 1 = include)	Saves application settings and optionally CFG info settings (button and command definitions) to the given application configuration file.	RPCCmd	
ASSIGN_BUTTON arg1 = button page (int, 0-3) arg2 = button number in page (int, 0-29) arg3 = command name (string) arg4 = button tooltip (string)	Assigns an existing named command to a button in the GUI. Will replace an existing button assignment. Use "" to unassign. Use button page = -1 for all pages and button number = -1 for all buttons on page (e.g. for group unassignment).	RPCCmd	

WRITE_CURRENT_STATE arg1 = script file name (string)	Writes the current state as RPC commands to the given script filename (overwrites existing file).	RPCCmd	
FORCE_TRIG arg1 = trigger command (0 = low, 1 = high, 2 = toggle, 3 = pulse)	Asserts TrigOut on the CPhy Generator instrument. Note: this is for immediate trigger control (use ASSERT_TRIG to control TrigOut from PG program). Use SET_TRIG_PULSE_WIDTH to set the pulse width.	RPCCmd	
<u>Frame Timing Configuration</u>			
SET_TIMING_HSYNC arg1 = pixel count (int)	In DSI, sets the horizontal sync period for video-mode frames. In CSI sets the horizontal blanking period.	RPCCmd	Note 9
SET_TIMING_HBPORCH arg1 = pixel count (int)	In DSI, sets the horizontal back porch period for video-mode frames. Unused in CSI.	RPCCmd	Note 9
SET_TIMING_HFPORCH arg1 = pixel count (int)	In DSI, sets the horizontal front porch period for video-mode frames. Unused in CSI.	RPCCmd	Note 9
SET_TIMING_HACTIVE arg1 = pixel count (int)	Sets the horizontal active period for video-mode frames.	RPCCmd	Note 9
SET_TIMING_VSYNC arg1 = line count (int)	In DSI, sets the vertical sync period for video-mode frames. In CSI sets the vertical blanking period.	RPCCmd	Note 9
SET_TIMING_VBPORCH arg1 = line count (int)	In DSI, sets the vertical back porch period for video-mode frames. Unused in CSI.	RPCCmd	Note 9
SET_TIMING_VFPORCH arg1 = line count (int)	In DSI, sets the vertical front porch period for video-mode frames. Unused in CSI.	RPCCmd	Note 9
SET_TIMING_VACTIVE arg1 = line count (int)	Sets the vertical active period for video-mode frames.	RPCCmd	Note 9
SET_TIMING_LINE_TIME arg1 = line time (float, sec)	Sets the line time for video-mode frames. Also, sets the line time parameter as the master setting to use when timing is recomputed (i.e. when other parameters are changed).	RPCCmd	
SET_TIMING_PIX_CLK arg1 = pixel clock frequency (float, Hz)	Sets the pixel clock frequency for video-mode frames. Also, sets the pixel clock frequency parameter as the master setting to use when timing is recomputed (i.e. when other parameters are changed).	RPCCmd	
SET_TIMING_FRAME_RATE arg1 = frame rate (float, Hz)	Sets the frame rate for video-mode frames. Also, sets the frame rate parameter as the master setting to use when timing is recomputed (i.e. when other parameters are changed).	RPCCmd	
SET_TIMING_HS_BIT_RATE arg1 = HS bit rate (float, bps)	Sets the HS bit rate to use for video-mode frames. Also, except in DSI burst mode video, sets the HS bit rate parameter as the master setting to use when timing is recomputed (i.e. when other parameters are changed).	RPCCmd	
SET_TIMING_HSYNC_ENABLE_DSI_PULSE_MODE arg1 = 0 (disable), 1 (enable)	If enabled, sets DSI video mode frames to pulse mode. Otherwise, event mode is implemented.	RPCCmd	Note 9
SET_TIMING_ENABLE_DSI_BURST_MODE arg1 = 0 (disable), 1 (enable)	If enabled, sets DSI video mode frames to burst mode. Otherwise, non-burst mode is implemented.	RPCCmd	Note 9
SET_TIMING_ENABLE_CSI_LSLE_MODE arg1 = 0 (disable), 1 (enable)	If enabled, includes line start and line end packets in CSI video-mode frames. Otherwise, line start and line end packets are not included.	RPCCmd	Note 9
SET_TIMING_ENABLE_CSI_FRAME_NUMBERING arg1 = 0 (disable), 1 (enable)	If enabled, sets and increments the frame number argument in vsync start and vsync end packets in CSI video-mode frames. Otherwise, sets the frame number argument to zero.	RPCCmd	Note 9

SET_TIMING_ENABLE_CSI_LINE_NUMBERING arg1 = 0 (disable), 1 (enable)	If enabled, sets and increments the line number argument in line start and line end packets in CSI video-mode frames. Otherwise, sets the line number argument to zero.	RPCCmd	Note 9
SET_TIMING_HSYNC_BLANKING_MODE arg1 = blanking mode (int, x_BLANK_MODE defines)	Sets the blanking mode to use for horizontal sync (DSI) or horizontal blanking (CSI) in video mode frames.	RPCCmd	Note 9
SET_TIMING_HBPORCH_BLANKING_MODE arg1 = blanking mode (int, x_BLANK_MODE defines)	Sets the blanking mode to use for horizontal back porch in DSI video mode frames.	RPCCmd	Note 9
SET_TIMING_HFPORCH_BLANKING_MODE arg1 = blanking mode (int, x_BLANK_MODE defines)	Sets the blanking mode to use for horizontal front porch in DSI video mode frames.	RPCCmd	Note 9
SET_TIMING_VERTICAL_BLANKING_MODE arg1 = blanking mode (int, x_BLANK_MODE defines)	Sets the blanking mode to use for vertical blanking in video mode frames.	RPCCmd	Note 9
SET_TIMING_TOP_FIELD_FIRST arg1 = 0 (disable), 1 (enable)	Sets the field order in DSI video mode frames for interlaced formats.	RPCCmd	Note 9
SET_TIMING_CLK_ON_OFF_DURING_LP11_BLANKING arg1 = 0 (disable), 1 (enable)	Sets whether the clock should be turned off during LP11 blanking in video mode frames.	RPCCmd	Note 9
SET_3D_PARAMETER arg1 = 3D parameter number (int, PARAM_3D_x defines) arg2 = 3D parameter value (int)	Sets a 3D parameter for DSI stereoscopic display.	RPCCmd	Note 9
SET_TIMING_MASTER arg1 = timing master (int, MASTER_x defines)	Sets the timing parameter to use as the master setting for clock rate calculation.	RPCCmd	
SET_ADVANCE_FRAME_ON_EXTERNAL_EVENT arg1 = 0 (disable), 1 (enable)	Sets an option for video-mode sequence construction where each frame in the sequence loops individually until an external event occurs, causing playback to advance to the next frame in the sequence.	RPCCmd	
SET_TIMING_ENABLE_SKEW_CAL arg1 = 0 (disable), 1 (enable)	Enables or disables the sending of a skew calibration sequence between video-mode frames.	RPCCmd	Note 9
SET_TIMING_SKEW_CAL_UI arg1 = skew calibration burst duration (UI)	Sets the skew calibration duration in UI between video-mode frames. Set to 0 to use the skew calibration setting of the current Dphy Configuration.	RPCCmd	Note 9
SET_TIMING_ENABLE_ALT_CAL arg1 = 0 (disable), 1 (enable)	Enables or disables the sending of an alternate calibration sequence between video mode frames.	RPCCmd	
SET_TIMING_ALT_CAL_UI arg1 = alt calibration burst duration (UI)	Sets the alternate calibration duration in UI between video-mode frames. Set to 0 to use the skew calibration setting of the current Dphy Configuration.	RPCCmd	
SET_TIMING_ENABLE_CONTINUOUS_MODE arg1 = 0 (disable), 1 (enable)	Sets the looping mode for video mode commands.	RPCCmd	Note 9
SELECT_TIMING_CONFIG arg1 = timing configuration name (string)	Selects the named timing configuration to use as the current frame timing configuration.	RPCCmd	
SAVE_TIMING_CONFIG arg1 = timing configuration name (string)	Saves the current timing configuration definition as a named timing configuration having the given name.	RPCCmd	
DELETE_TIMING_CONFIG arg1 = timing configuration name (string)	Deletes the named timing configuration.	RPCCmd	
DELETE_ALL_TIMING_CONFIGS no arguments	Deletes all named timing configurations	RPCCmd	
SET_SORT_TIMING_CONFIGS arg1 = enableFlag (int, 0 = disable, 1 = enable)	Sets the flag that determines whether the timing configuration list is sorted alphabetically when configurations are added. In addition, if the flag is set true, the current timing configuration list is sorted.	RPCCmd	
SET_DUAL_INTERFACE_MODE arg1 = mode (int, DUAL_IF_x defines)	Sets the dual-interface mode that controls how video frames are sent across the master/slave interfaces when in dual-instrument configuration.	RPCCmd	Note 10, 11

<u>DPhy Configuration</u>			
SET_DHY_PARAMETER arg1 = DPhy parameter (DPHY_PARAM_x defines) arg2 = nsValue (int) arg3 = UIValue (int) arg4 = TLPXValue (int)	Sets the value for the given DPhy parameter. Each component has different units and the aggregate duration of all three components is used as the actual parameter setting. Note that only some components are visible and changeable in the DPHYGenCtl GUI.	RPCCmd	Note 9
SET_ENABLE_PREAMBLE arg1 = enableFlag (int, 0 = disable, 1 = enable)	Enables preamble sequence output before HS_SYNC for HS bursts when the HS frequency is above the PREAMBLE_FREQ threshold.	RPCCmd	
SET_ENABLE_PREAMBLE_FREQ arg1 = frequency threshold (int, Mbps)	Sets the PREAMBLE_FREQ threshold above which the preamble sequence is output for HS bursts.	RPCCmd	
SELECT_DPHY_CONFIG arg1 = DPhy configuration name (string)	Selects the named DPhy configuration to used as the current Dphy configuration.	RPCCmd	Note 9
SAVE_DPHY_CONFIG arg1 = DPhy configuration name (string)	Saves the current DPhy configuration as a named DPhy configuration having the given name.	RPCCmd	
DELETE_DPHY_CONFIG arg1 = DPhy configuratio name (string)	Deletes the named DPhy configuration.	RPCCmd	
DELETE_ALL_DPHY_CONFIGS no arguments	Deletes all named DPhy configurations.	RPCCmd	
SET_SORT_DPHY_CONFIGS arg1 = enableFlag (int, 0 = disable, 1 = enable)	Sets the flag that determines whether the DPhy configuration list is sorted alphabetically when configurations are added. In addition, if the flag is set true, the current DPhy configuration list is sorted.	RPCCmd	
<u>Macro Defintion</u>			
START_MACRO no arguments	Puts the application in macro definition mode (discarding previous macro, if open). Subsequent MIPI commands sent will append to current macro until END_MACRO is issued.	RPCCmd	
END_MACRO arg1 = macro name (string)	When in macro definition mode, ends the current macro definition, creating a named macro command in the application.	RPCCmd	
SEND_MACRO	When in macro definition mode, ends the current macro definition and sends the current macro definition, without naming. The macro definition is subsequently discarded.	RPCCmd	
MEASURE_MACRO no arguments retval = measured macro duration (int, UI)	When in macro definition mode, ends the current macro definition and measures the period required if it were played. The macro definition is subsequently discarded. Note: measurement includes implicit clock on / off effects due to option settings (i.e. turn off relevant options if clock on / off effects are unwanted in measurement).	RPCCmd	
RENAME_MACRO_COMPONENT_COMMAND arg1 = zero-based component command index (int) arg2 = component command name (string)	When in macro definition mode, names an existing component command in the macro, as specified by the component command index. For example, use 0 to name the first component command in the macro.	RPCCmd	
<u>Status Commands</u>			
IS_INST_CONNECTED no arguments retval = 0 (offline), 1 (connected)	Indicates if the application is connected to a DPhy Generator instrument.	RPCCmd	

IS_PROGRAM_RUNNING no arguments retval = 0 (idle), 1 (running)	Indicates if the internal PG of the DPHY Generator is running. If the instrument is idle, all lanes of the MIPI bus including the clock lane are in LP11.	RPCCmd	
GET_LP_CONTENTION no arguments retval = contention mask (int)	Returns a bus contention mask consisting of 4 bits relating to lane 0 contention (since last contention reset). The first two bits (0 and 1) indicate contention when transmitting a LP0 on LP- and LP+ respectively. The next two bits (2 and 3) indicate contention when transmitting LP1 on LP- and LP+ respectively.	RPCCmd	
RESET_LP_CONTENTION no arguments	Resets contention status.	RPCCmd	
GET_DUT_RESPONSE no arguments retval = DUT response (byte array)	Gets the queued DUT response(s). See Users Manual for more description.	RPCQuery	
SAVE_DUT_RESPONSE arg1 = file name to save DUT response data (string) arg2 = append flag (0 = new file, 1 = append to existing file) arg3 = comment to save with response (string)	Saves the queued DUT response to an ASCII text file. If the append flag is false, any existing file with the given file name is deleted. Set comment to "" for no comment.	RPCCmd	
GET_INST_GP_INPUT_STATE no arguments retval = 2-bit GP Input pin state (int)	Returns the current input state on the two GP Input pins on the instrument back panel connector.	RPCCmd	
SAVE_INST_GP_INPUT_STATE arg1 = file name to save GP input state (string) arg2 = append flag (0 = new file, 1 = append to existing file) arg3 = comment to save with response (string)	Saves the current input state on the two GP Input pins on the instrument back panel connector to an ASCII text file. If the append flag is false, any existing file with the given file name is deleted. Set comment to "" for no comment.	RPCCmd	
<u>Frame Buffer Commands</u>			
LOAD_FRAME arg1 = buffer number (int, 0-3) arg2 = conversion video format data type (int) arg3 = image file name (string)	Loads the frame from the given file into the given frame buffer, converting to the given video format on input. Frame data can then be used in subsequent commands by using the name "USERFRAME0" "USERFRAME1", "USERFRAME2" or "USERFRAME3" as the filename, primarily in frame construction macros. Use DEALLOC_FRAME to deallocate frame memory when done.	RPCCmd	Note 9
DEALLOC_FRAME arg1 = buffer number (int, 0-3)	Deallocates the frame stored in the given frame buffer.	RPCCmd	Note 9
<u>WriteMemory Configuration</u>			
SET_WM_PARTITION_LENGTH arg1 = maximum bytes per WriteMem partition (int)	Sets the maximum number of bytes to use when partitioning WriteMemory commands.	RPCCmd	Note 9
SET_WM_PARTITION_INTERVAL arg1 = LP11 time between WriteMem partitions (float, sec)	Sets the LP11 time to insert between WriteMemory partitions.	RPCCmd	Note 9
SET_WM_IMAGE_DECODE_FORMAT arg1 = WriteMem image decode format (FMT_x defines)	Sets the image file decode format to use for WriteMemory image files.	RPCCmd	Note 9

SET_WM_RGB_BIT_ORDER arg1 = RGB bit order (WM_BIT_ORDER_xxx defines)	Sets the RGB bit order to use when sending images via WriteMemory	RPCCmd	Note 9
<u>RPC Command Support</u>			
GET_RPC_CMD_STRING_FROM_NAME arg1 = Cmd name (string)	Given a Cmd name, returns an RPC script string for the MIPI command associated with a named command.	RPCQuery	
GET_RPC_CMD_STRING_FROM_INDEX arg1 = Cmd index (int)	Given a Cmd index, returns an RPC script string for the MIPI command associated with a named command.	RPCQuery	
EXECUTE_RPC_CMD_STRING arg1 = RPC command string (string)	Executes the given RPC command string. The string may have multiple commands separated by <CR><LF> or <CR>. All rules of RPC script files are followed.	RPCCmd	
<u>DSC Config. Commands (license required)</u>			
SELECT_DSC_CONFIG arg1 = DSC configuration name (string)	Selects the named DSC configuration to use as the current DSC configuration.	RPCCmd	Note 9
SAVE_DSC_CONFIG arg1 = DSC configuration name (string)	Saves the current DSC configuration definition as a named DSC configuration having the given name.	RPCCmd	
DELETE_DSC_CONFIG arg1 = DSC configuration name (string)	Deletes the named DSC configuration.	RPCCmd	
DELETE_ALL_DSC_CONFIGS no arguments	Deletes all named DSC configurations	RPCCmd	
SET_SORT_DSC_CONFIGS arg1 = enableFlag (int, 0 = disable, 1 = enable)	Sets the flag that determines whether the DSC configuration list is sorted alphabetically when configurations are added. In addition, if the flag is set true, the current DSC configuration list is sorted.	RPCCmd	
SET_DSC_SLICE_WIDTH arg1 = slice width (int, pixels)	Sets the slice width to use in DSC encoding (set to 0 for full image width)	RPCCmd	Note 9
SET_DSC_SLICE_HEIGHT arg1 = slice height (int, pixels)	Sets the slice height to use in DSC encoding (set to 0 for full image height)	RPCCmd	Note 9
SET_DSC_VERSION_12 arg1 = enableFlag (int, 0 = version 1.1, 1 = version 1.2)	Sets the DSC version to 1.2 (if enabled) or 1.1 (if not enabled)	RPCCmd	Note 9
SET_DSC_ENABLE_SIMPLE_422 arg1 = enableFlag (int, 0 = disable, 1 = enable)	Sets the SIMPLE_422 option for DSC encoding. This command has been renamed from the DSC 1.1 implementation (SET_DSC_ENABLE_422), which is still supported.	RPCCmd	Note 9
SET_DSC_ENABLE_NATIVE_422 arg1 = enableFlag (int, 0 = disable, 1 = enable)	Sets the NATIVE_422 option for DSC encoding. If this option is set, the USE_YUV_INPUT option must also be set.	RPCCmd	Note 9
SET_DSC_ENABLE_NATIVE_420 arg1 = enableFlag (int, 0 = disable, 1 = enable)	Sets the NATIVE_420 option for DSC encoding. If this option is set, the USE_YUV_INPUT option must also be set.	RPCCmd	Note 9
SET_DSC_USE_REC_709 arg1 = enableFlag (int, 0 = disable, 1 = enable)	Sets the USE_REC_709 option for DSC encoding. Uses the Rec. 709 RGB-to-YUV color space conversion (if enabled) or the Rec. 601 color-space conversion (if not enabled).	RPCCmd	Note 9
SET_DSC_USE_YUV_INPUT arg1 = enableFlag (int, 0 = disable, 1 = enable)	Sets the USE_YUV_INPUT option for DSC encoding	RPCCmd	Note 9
SET_DSC_USE_BLOCK_PREDICTION arg1 = enableFlag (int, 0 = disable, 1 = enable)	Sets the USE_BLOCK_PREDICTION option for DSC encoding	RPCCmd	Note 9

SET_DSC_ONE_CHUNK_PER_PACKET arg1 = enableFlag (int, 0 = disable, 1 = enable)	Sets a flag for use when sending DSC compressed video (in command mode or video-mode) If the flag is set, each chunk will be sent in its own packet. Otherwise, all chunks that compose a line will be sent in one packet.	RPCCmd	Note 9
SET_DSC_CONFIG_FILENAME arg1 = configuration filename (string)	Specifies the DSC configuration file for DSC encoding.	RPCCmd	Note 9
DSC_COMPRESS_FILE arg1 = source image filename (string) arg2 = DSC filename (string)	Compresses the given source image according to the current DSC configuration settings. The source image filename is any supported image format including a test pattern. If "Allow Image Rescaling" option is set, the source image (unless source is a DPX file) will be first resized to the current active frame dimensions before encoding. (Requires DSC Option)	RPCCmd	
DSC_UNCOMPRESS_FILE arg1 = DSC filename (string) arg2 = destination image filename (string)	Uncompresses the given source DSC file, writing the output image to the given destination filename. The output image format is determined by the file extension, i.e. DPX, BMP, JPG, etc. (Requires DSC Option)	RPCCmd	
<u>Miscellaneous Commands</u>			
COMPUTE_CRC arg1 = data bytes (byte[])	Computes the CRC of the given data bytes.	RPCCmd	
COMPUTE_ECC arg1 = data bytes (byte[3])	Computes the ECC of the given 3 data bytes.	RPCCmd	
DPhyGenCtl RPC MIPI Cmd Reference			
MIPI Cmd	Arguments	Notes	
<u>DSI Video Commands</u>			
PACKED_PIXEL_STREAM_565	arg1 = FrameCount; arg2 = VideoErr	Notes 2,3	
PACKED_PIXEL_STREAM_666	arg1 = FrameCount; arg2 = VideoErr	Notes 2,3	
LOSSE_PIXEL_STREAM_666	arg1 = FrameCount; arg2 = VideoErr	Notes 2,3	
PACKED_PIXEL_STREAM_888	arg1 = FrameCount; arg2 = VideoErr	Notes 2,3	
PACKED_PIXEL_STREAM_101010	arg1 = FrameCount; arg2 = VideoErr	Notes 2,3	
PACKED_PIXEL_STREAM_121212	arg1 = FrameCount; arg2 = VideoErr	Notes 2,3	
LOOSE_PIXEL_STREAM_20_YCBCR_422	arg1 = FrameCount; arg2 = Interlaced; arg3 = VideoErr	Notes 2,3	
PACKED_PIXEL_STREAM_24_YCBCR_422	arg1 = FrameCount; arg2 = Interlaced; arg3 = VideoErr	Notes 2,3	
PACKED_PIXEL_STREAM_16_YCBCR_422	arg1 = FrameCount; arg2 = Interlaced; arg3 = VideoErr	Notes 2,3	
COMPRESSED_PIXEL_STREAM	arg1, arg2, arg3 unused	Note 4	
<u>DSI Frame Packets</u>			
VSYNC_START	arg1, arg2, arg3 unused		
VSYNC_END	arg1, arg2, arg3 unused		
HSYNC_START	arg1, arg2, arg3 unused		
HSYNC_END	arg1, arg2, arg3 unused		
DSI_NULL_PKT	arg1 = WriteLen (payload byte count)		
DSI_BLANKING_PKT	arg1 = WriteLen (payload byte count)		

<u>DSI Non-video Packets</u>			
COLOR_MODE_OFF	arg1, arg2, arg3 unused		
COLOR_MODE_ON	arg1, arg2, arg3 unused		
SHUT_DOWN_PERIPHERAL	arg1, arg2, arg3 unused		
TURN_ON_PERIPHERAL	arg1, arg2, arg3 unused		
PICTURE_PARAMETER_SET	arg1, arg2, arg3 unused	Note 4	
GENERIC_LONG_WRITE	arg1, arg2, arg3 unused	Note 4	
GENERIC_SHORT_WRITE	arg1 = ParamCnt (0, 1, 2), arg2 = Param1[7:0], arg3 = Param2[7:0]		
GENERIC_READ	arg1 = ParamCnt (0, 1, 2), arg2 = Param1[7:0], arg3 = Param2[7:0]		
DCS_LONG_WRITE	arg1, arg2, arg3 (per DCS command code)	Note 4	
DCS_SHORT_WRITE	<u>Enable Skew Cal</u> : this checkbox indicates if a deskew burst sequence is to be sent before each frame in a video		
DCS_READ	arg1, arg2, arg3 (per DCS command code)		
SET_MAX_RETURN_PKT_SIZE	arg1 = MaxReturnPktSize[15:0]		
SCRAMBLING_MODE	arg1 = Data 0; arg2 = Data 1		
COMPRESSION_MODE	arg1 = Data 0; arg2 = Data 1		
<u>DSI Response Packets</u>			
ACK_AND_ERROR_REPORT	arg1 = ErrorBits[15:0]		
GENERIC_SHORT_READ_RESPONSE	arg1 = ParamCnt (1, 2), arg2 = Param1[7:0], arg3 = Param2[7:0]		
GENERIC_LONG_READ_RESPONSE	arg1, arg2, arg3 unused	Note 4	
DCS_SHORT_READ_RESPONSE	arg1 = ParamCnt (1, 2), arg2 = Param1[7:0], arg3 = Param2[7:0]		
DCS_LONG_READ_RESPONSE	arg1, arg2, arg3 unused	Note 4	
<u>CSI Video Commands</u>			
PIXEL_STREAM_RGB444	arg1 = Frame Count; arg2 = Video Error	Notes 2,3	
PIXEL_STREAM_RGB555	arg1 = Frame Count; arg2 = Video Error	Notes 2,3	
PIXEL_STREAM_RGB565	arg1 = Frame Count; arg2 = Video Error	Notes 2,3	
PIXEL_STREAM_RGB666	arg1 = Frame Count; arg2 = Video Error	Notes 2,3	
PIXEL_STREAM_RGB888	arg1 = Frame Count; arg2 = Video Error	Notes 2,3	
PIXEL_STREAM_RAW6	arg1 = Frame Count; arg2 = Video Error	Notes 2,3	
PIXEL_STREAM_RAW7	arg1 = Frame Count; arg2 = Video Error	Notes 2,3	
PIXEL_STREAM_RAW8	arg1 = Frame Count; arg2 = Video Error	Notes 2,3	
PIXEL_STREAM_RAW9	arg1 = Frame Count; arg2 = Video Error	Notes 2,3	
PIXEL_STREAM_RAW10	arg1 = Frame Count; arg2 = Video Error	Notes 2,3	
PIXEL_STREAM_RAW12	arg1 = Frame Count; arg2 = Video Error	Notes 2,3	
PIXEL_STREAM_RAW14	arg1 = Frame Count; arg2 = Video Error	Notes 2,3	
PIXEL_STREAM_RAW16	arg1 = Frame Count; arg2 = Video Error	Notes 2,3	
PIXEL_STREAM_RAW20	arg1 = Frame Count; arg2 = Video Error	Notes 2,3	
PIXEL_STREAM_YUV420L_8	arg1 = Frame Count; arg2 = Interlaced; arg3 = Video Error	Notes 2,3	
PIXEL_STREAM_YUV422_8	arg1 = Frame Count; arg2 = Interlaced; arg3 = Video Error	Notes 2,3	
PIXEL_STREAM_YUV422_10	arg1 = Frame Count; arg2 = Interlaced; arg3 = Video Error	Notes 2,3	
USER_8BIT_TYPE1	arg1, arg2, arg3 unused	Note 4	
USER_8BIT_TYPE2	arg1, arg2, arg3 unused	Note 4	
USER_8BIT_TYPE3	arg1, arg2, arg3 unused	Note 4	
USER_8BIT_TYPE4	arg1, arg2, arg3 unused	Note 4	

USER_8BIT_TYPE5	arg1, arg2, arg3 unused	Note 4	
USER_8BIT_TYPE6	arg1, arg2, arg3 unused	Note 4	
USER_8BIT_TYPE7	arg1, arg2, arg3 unused	Note 4	
USER_8BIT_TYPE8	arg1, arg2, arg3 unused	Note 4	
<u>CSI Compressed Video Packets</u>			
RAW_COMPRESSED_10_8_10	arg1 = FrameCount; arg2 = User DataType	Notes 3,5	
RAW_COMPRESSED_10_7_10	arg1 = FrameCount; arg2 = User DataType	Notes 3,5	
RAW_COMPRESSED_10_6_10	arg1 = FrameCount; arg2 = User DataType	Notes 3,5	
RAW_COMPRESSED_12_8_12	arg1 = FrameCount; arg2 = User DataType	Notes 3,5	
RAW_COMPRESSED_12_7_12	arg1 = FrameCount; arg2 = User DataType	Notes 3,5	
RAW_COMPRESSED_12_6_12	arg1 = FrameCount; arg2 = User DataType	Notes 3,5	
<u>CSI Frame Packets</u>			
FRAME_START	arg1 = FrameNum[15:0]		
FRAME_END	arg1 = FrameNum[15:0]		
LINE_START	arg1 = LineNum[15:0]		
LINE_END	arg1 = LineNum[15:0]		
CSI_NULL_PKT	arg1 = WriteLen (payload byte count)		
CSI_BLANKING_PKT	arg1 = WriteLen (payload byte count)		
<u>CSI Non-video Packets</u>			
GENERIC_SHORT_PKT1	arg1 = Value[15:0]		
GENERIC_SHORT_PKT2	arg1 = Value[15:0]		
GENERIC_SHORT_PKT3	arg1 = Value[15:0]		
GENERIC_SHORT_PKT4	arg1 = Value[15:0]		
GENERIC_SHORT_PKT5	arg1 = Value[15:0]		
GENERIC_SHORT_PKT6	arg1 = Value[15:0]		
GENERIC_SHORT_PKT7	arg1 = Value[15:0]		
GENERIC_SHORT_PKT8	arg1 = Value[15:0]		
CSI_LONG_PKT	arg1, arg2, arg3 unused	Note 4	
<u>Custom Packets</u>			
CUSTOM_COMMAND	arg1 = DataID, plus command should have non-null payload array		
CUSTOM_LONG_COMMAND	arg1 = DataID, plus command should have non-null payload array		
<u>Phy Commands</u>			
RESET_TRIGGER	arg1, arg2, arg3 unused	Note 12	
TRIGGER1	arg1, arg2, arg3 unused	Note 12	
TRIGGER2	arg1, arg2, arg3 unused	Note 12	
TRIGGER3	arg1, arg2, arg3 unused	Note 12	
ESCAPE	arg1 = CmdByte[7:0]	Note 12	
SEND_BTA	arg1, arg2, arg3 unused	Note 6, 12	
WAIT_FOR_BTA	arg1, arg2, arg3 unused	Note 6, 12	
ULPS	arg1 = LaneMask[4:0]	Note 6, 12	

EOT_PKT	arg1, arg2, arg3 unused	Note 12	
SKEW_CALIBRATION	arg1 = duration (UI), set to 0 to use setting in current Dphy Configuration)	Note 12	
ALT_CALIBRATION	arg1 = duration (UI), set to 0 to use setting in current Dphy Configuration)	Note 12	
TGR_DATA_SEQ	arg1 = LoopCnt, plus command should have non-null payload array	Note 6, 12	
TGR_BIT_SEQ	arg1 = LoopCnt, plus command should have non-null payload array	Note 6, 12	
PRBS9_SEQ	arg1 = seed (applies to all lanes), arg2 = WordCnt (16-bit words)	Note 6, 12	
PRBS11_SEQ	arg1 = seed (applies to all lanes), arg2 = WordCnt (16-bit words)	Note 6, 12	
PRBS18_SEQ	arg1 = seed (applies to all lanes), arg2 = WordCnt (16-bit words)	Note 6, 12	
CLOCK_ON	arg1, arg2, arg3 unused	Note 6, 12	
CLOCK_OFF	arg1, arg2, arg3 unused	Note 6, 12	
Miscellaneous Commands			
LP_DELAY	arg1 = delay (ns); arg2 = LPValue[9:0]	Notes 6, 7, 12	
LP_DELAY_UI	arg1 = delay (UI); arg2 = LPValue[9:0]	Notes 6, 7, 12	
LP_DELAY_TO_POS	arg1 = delay (ns); arg2 = LPValue[9:0]	Notes 6, 7, 12	
LP_DELAY_TO_POS_UI	arg1 = delay (UI); arg2 = LPValue[9:0]	Notes 6, 7, 12	
MARK_ZERO_POS	arg1, arg2, arg3 unused	Notes 6, 7, 12	
WAIT_EXT_EVENT	arg1 = LPValue[9:0]	Notes 6, 7, 12	
CMD_INSERTION_POINT	arg1 = delay (ns); arg2 = LPValue[9:0]; arg3 = discard flag (0 = suspend program while inserting, 1 = discard input while inserting)	Notes 6, 7, 12	
ASSERT_TRIG	arg1 = delay (ns); arg2 = LPValue[9:0]; arg3 = Trigger Cmd	Notes 6, 7, 12	
LOOP_START	arg1 = loop iterations (int, use 0 for infinite loop)	Note 8, 12	
LOOP_END	arg1, arg2, arg3 unused	Note 8, 12	
PG_FILE	Command should have non-null ".pg" filename created using the File->"Send Cmd to File" menu option	Note 12	
FILE_COMMAND	Command should have non-empty text filename containing component file command syntax	Note 12	
RPC_SCRIPT	Command should have non-empty text filename containing script commands	Note 12	
DPhyGenCtl RPC DCS Cmd Reference			
DCS Cmd	Arguments	Notes	Cmd Code
DCS Short Write Commands			
ENTER_IDLE_MODE	arg1 = 0		0x39
ENTER_INVERT_MODE	arg1 = 0		0x21
ENTER_NORMAL_MODE	arg1 = 0		0x13
ENTER_PARTIAL_MODE	arg1 = 0		0x12
ENTER_SLEEP_MODE	arg1 = 0		0x10
EXIT_IDLE_MODE	arg1 = 0		0x38
EXIT_INVERT_MODE	arg1 = 0		0x20
EXIT_SLEEP_MODE	arg1 = 0		0x11
NOP	arg1 = 0		0x0
SET_DISPLAY_OFF	arg1 = 0		0x28
SET_DISPLAY_ON	arg1 = 0		0x29

SET_TEAR_ON	arg1 = 1, arg2 = M[0]		0x35
SET_TEAR_OFF	arg1 = 0		0x34
SOFT_RESET	arg1 = 0		0x01
SET_ADDRESS_MODE	arg1 = 1, arg2 = Mode[7:0]		0x36
SET_3D_CONTROL	arg1 = 1, arg2 = Mode[7:0]		0x3d
SET_GAMMA_CURVE	arg1 = 1, arg2 = GammaCurve[7:0]		0x26
SET_PIXEL_FORMAT	arg1 = 1, arg2 = Format[7:0]		0x3a
SET_VSYNC_TIMING	arg1 = Timing[7:0]		0x40
<u>DCS Long Write Commands</u>			
SET_SCROLL_START	arg1 = VertScrollPtr[15:0]		0x37
SET_TEAR_SCANLINE	arg1 = Line[15:0]		0x44
SET_COLUMN_ADDRESS	arg1 = StartColumn[15:0]; arg2 = EndColumn[15:0]		0x2a
SET_PAGE_ADDRESS	arg1 = StartPage[15:0]; arg2 = EndPage[15:0]		0x2b
SET_PARTIAL_ROWS	arg1 = StartRow[15:0]; arg2 = EndRow[15:0]		0x30
SET_PARTIAL_COLUMNS	arg1 = StartColumn[15:0]; arg2 = EndColumn[15:0]		0x31
SET_SCROLL_AREA	arg1 = TopFixedArea[15:0]; arg2 = VertScrollArea[15:0] arg3 = BotFixedArea[15:0]		0x33
SET_CABC_MIN_BRIGHTNESS	arg1 = Brightness[15:0]		0x5e
SET_DISPLAY_BRIGHTNESS	arg1 = Brightness[15:0]		0x51
WRITE_LUT	arg1, arg2, arg3 unused	Note 4	0x2d
WRITE_MEMORY_START	arg1 = FileOffset (bytes) arg2 = WriteLen (bytes), can use 0 for entire file/buffer	Note 4	0x2c
WRITE_MEMORY_CONTINUE	arg1 = FileOffset (bytes) arg2 = WriteLen (bytes), can use 0 for entire file/buffer	Note 4	0x3c
<u>DCS Read Commands</u>			
GET_ADDRESS_MODE	arg1, arg2, arg3 unused		0x0b
GET_3D_CONTROL	arg1, arg2, arg3 unused		0x3f
GET_COMPRESSION_MODE	arg1, arg2, arg3 unused		0x03
GET_BLUE_CHANNEL	arg1, arg2, arg3 unused		0x08
GET_GREEN_CHANNEL	arg1, arg2, arg3 unused		0x07
GET_RED_CHANNEL	arg1, arg2, arg3 unused		0x06
GET_PIXEL_FORMAT	arg1, arg2, arg3 unused		0x0c
GET_DIAGNOSTIC_RESULT	arg1, arg2, arg3 unused		0x0f
GET_DISPLAY_MODE	arg1, arg2, arg3 unused		0x0d
GET_POWER_MODE	arg1, arg2, arg3 unused		0x0a
GET_SCANLINE	arg1, arg2, arg3 unused		0x45
GET_SIGNAL_MODE	arg1, arg2, arg3 unused		0x0e
GET_CABC_MIN_BRIGHTNESS	arg1, arg2, arg3 unused		0x5f
GET_CONTROL_DISPLAY	arg1, arg2, arg3 unused		0x54
GET_DISPLAY_BRIGHTNESS	arg1, arg2, arg3 unused		0x52
GET_ERROR_COUNT_ON_DSI	arg1, arg2, arg3 unused		0x05
GET_IMAGE_CHECKSUM_CT	arg1, arg2, arg3 unused		0x15
GET_IMAGE_CHECKSUM_RGB	arg1, arg2, arg3 unused		0x14
GET_POWER_SAVE	arg1, arg2, arg3 unused		0x56

READ_DDB_START	arg1, arg2, arg3 unused		0xa1
READ_DDB_CONTINUE	arg1, arg2, arg3 unused		0xa8
READ_MEMORY_START	arg1, arg2, arg3 unused		0x2e
READ_MEMORY_CONTINUE	arg1, arg2, arg3 unused		0x3e
READ_PPS_CONTINUE	arg1, arg2, arg3 unused		0xa9
READ_PPS_START	arg1, arg2, arg3 unused		0xa2
NOTES:			
Note 0	If command is called during macro definition, the command will not be sent on the DPHY bus (or defined in the case of ADD_MIPI_CMD or ADD_IMPAIRED_MIPI_CMD). Instead the command will be inserted as a component command into the macro.		
Note 1	Configuration command must be bracketed by START_CONFIG and END_CONFIG.		
Note 2	Argument description for all CSI and DSI formats except DSI YCBCR formats (video-mode frames) arg1 = FrameCount which is the number of frames to output in the sequence arg2 = VideoErr ([31:0] ones-based line index, [15:0] bit position) Argument description for DSI YCBCR formats (video-mode frames): arg1 = FrameCount which is the number of frames to output in the sequence arg2 = Interlaced (0 = not interlaced, 1 = interlaced; only for YCBCR formats) arg3 = VideoErr ([31:0] ones-based line index, [15:0] bit position) Argument description for single video packets in macros: arg1 = starting source byte offset in buffer or file arg2 = payload length See video-mode command description in DPHYGenCtlUsersManual for more details.		
Note 3	Usage option 1: provide non-zero length payload array. Allows image data to be sent in the command. Payload length should be FrameCount * HTotal * VTotal * bytesPerPix. Usage option 2: provide non-empty filename. Filename can be test pattern, frame buffer, video file, image file or image file sequence, binary file or binary file sequence. See video-mode command description in DPHYGenCtlUsersManual for more details. Note: an internal frame buffer previously loaded with LOAD_FRAME can be referenced using a filename of "USERFRAME<n>", where <n> is 0-3. The option is only available when "USERFRAME<n>" is used in a video-mode command in a macro with the Option "Enable Video Mode in Macros" unchecked. This usage facilitates video frame construction via macros.		

Note 4	Usage option 1: provide non-zero length payload array. Allows command data to be sent in the command. For WriteMemory commands, if this option is used, FileOffset and WriteLen are ignored. Usage option 2: provide non-empty filename. For non-video commands, filename is expected to be binary file. For WriteMemory commands and video-mode commands, filename can be test pattern, frame buffer, video file, image file or binary file.		
Note 5	CSI Compression commands require a purchased licence option from The Moving Pixel Company.		
Note 6	Additional Command Descriptions:		
SEND_BTA	Sends BTA LP sequence and waits for BTA Ack then response BTA from DUT		
WAIT_FOR_BTA	Waits for response BTA from DUT		
ULPS	Runs the PG, entering ULPS on lanes whose bit is set in LaneMask. The PG runs and lanes stay in ULPS until the HostEvent fires or the PG is stopped.		
TGR_DATA_SEQ, TGR_BIT_SEQ	Sends an HS burst (without Sync) having LoopCnt iterations of payload data bytes or bits (0 or 1) respectively.		
PRBS9_SEQ, PRBS11_SEQ, PRBS18_SEQ	Sends an HS burst (without Sync) having 2 * WordCnt bytes of a PRBS9, PRBS11, or PRBS18 sequence		
CLOCK_ON	Turns the clock on (in macro)		
CLOCK_OFF	Turns the clock off (in macro)		
LP_DELAY and LP_DELAY_UI	Holds the DPhy bus in the given LP state for delay ns or UI (in macro)		
LP_DELAY_TO_POS and LP_DELAY_TO_POS_UI	Holds the DPhy bus in the given LP state for delay ns or UI past the last defined ZERO_POS position (in macro)		
MARK_ZERO_POS	Defines a position during macro definition to use as a time reference for the LP_DELAY_TO_POS. Can be used multiple times in macro.		
WAIT_EXT_EVENT	Suspends output in the given LP state until the external event fires.		
CMD_INSERTION_POINT	Defines a command insertion point in a macro. Commands sent while macro is looping are inserted (if insertion is enabled).		
ASSERT_TRIG	Assert TrigOut signal. Note: this is for TrigOut control during program output (use FORCE_TRIG command to affect TrigOut immediately). Trigger Cmd can be 0 (assert low), 1 (assert high), 2 (toggle), 3 (pulse). Use SET_TRIG_DURATION to set pulse width.		
Note 7	LPValue[9:0] encodes the LP state of each data lane plus the clock lane. Bits 1..0 correspond to lane 0, bits 3..2 correspond to lane 1, etc, and bits 9..8 correspond to the clock lane. If the clock is running, LP bits are ignored.		

Note 8	START_LOOP and END_LOOP are used in macros to provide an easy way to duplicate a group of commands (bracket group with loop commands). With the exception of infinite looping (Loop Count = 0), it does not implement true looping and thus does not save any memory or construction time. But it is a useful shorthand, especially for video frame construction, where a command sequence that implements one video line can be looped multiple times. Note that 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.		
Note 9	Allowed non-MIPI command in macro for on-the-fly reconfiguration.		
Note 10	Requires DPhyGenCtl version 2.4 (or above).		
Note 11	Requires RevB DPhy Generator hardware.		
Note 12	In DPhyGenCtl version 2.4 (or above) all phy commands and miscellaneous commands can use a shortcut syntax when used in an RPC script. This means that instead of using the normal command syntax with SEND_MIPI_CMD, the user can specify just the command name and its arguments. For example: <pre># LOOP START 10 # SKEW_CALIBRATION 1024 # FILE_COMMAND "testFile.txt" # LOOP_END</pre> could be used instead of <pre># SEND_MIPI_CMD LOOP_START 0 0 DT_HS 0 10 0 0 "" null # SEND_MIPI_CMD SKEW_CALIBRATION 0 0 DT_HS 0 1024 0 0 "" null # SEND_MIPI_CMD FILE_COMMAND 0 0 DT_HS 0 0 0 0 "testFile.txt" null # SEND_MIPI_CMD LOOP_END 0 0 DT_HS 0 0 0 0 "" null</pre>		
MIPI Standards			
STD_DSI	0		
STD_CSI	1		
DT Modes			
DT_DEFAULT	0		
DT_LP	1		
DT_HS	2		
Timing Masters			

MASTER_LINE_TIME	0		
MASTER_PIX_CLK	1		
MASTER_FRAME_RATE	2		
MASTER_HS_BIT_RATE	3		
Options			
OPT_LOOP_COMMANDS	1		
OPT_WAIT_ON_EXT_EVENT_TO_START	2		
OPT_DISABLE_EVENT_TIMEOUT	3		
OPT_ENABLE_CMD_INSERTION	4		
OPT_ENABLE_VIDEO_MODE_IN_MACROS	5		
OPT_SEND_SINGLE_PKT_PER_HS_BURST	6	Note 9	
OPT_CLOCK_ON_WHEN_SENDING_CMDS	7	Note 9	
OPT_CLOCK_OFF_AFTER_HS_BURST	8	Note 9	
OPT_ENABLE_EOT_PKTS	9	Note 9	
OPT_ENCODE_RAW_FORMAT_AS_BAYER	10	Note 9	
OPT_ALWAYS_USE_PREDICTOR1_FOR_COMPRESSION	11	Note 9	
OPT_ENABLE_WM_PARTITIONING	12	Note 9	
OPT_ALLOW_IMAGE_RESCALING	13	Note 9	
OPT_WM_QUANTIZE_TO_LINE_LENGTH_MULTIPLE	14	Note 9	
OPT_ENABLE_HOLD_TRAIL_TEST_MODE	15		
OPT_SEND_PPS_WITH_COMPRESSED_VIDEO	16	Note 9	
OPT_ENABLE_DSI_SCRAMBLING	17		
OPT_ENABLE_CSI_SCRAMBLING	18		
OPT_SEND_BTA_ON_ALL_ACTIVE_LANES	19		
3D Parameters			
PARAM_3D_ENABLE	0		
PARAM_3D_STREAM_MOD	1		
PARAM_3D_DISPLAY_ORIENTATION	2		
PARAM_3D_LR_ORDER	3		
PARAM_3D_FORMAT	4		
PARAM_3D_VSYNC	5		
PARAM_3D_INTER_FRAME_VACT	6		
3D Stream Mode / 3D_Display Orientation			
PORTRAIT_MODE	0		
LANDSCAPE_MODE	1		
3D LR Order			
LEFT_FIRST	0		
RIGHT_FIRST	1		

3D Formats			
FORMAT_3D_LINE	0		
FORMAT_3D_FRAME	1		
FORMAT_3D_PIXEL	2		
DCS Write Memory Image Decode Formats			
FMT_RGB565_16	1		
FMT_RGB666_18	2		
FMT_RGB666_24	3		
FMT_RGB888_24	4		
FMT_YCBCR_420_12	11		
FMT_YCBCR_422_16	13		
FMT_RGB101010_30	23		
FMT_RGB121212_36	24		
FMT_YCBCR_422_LOOSE_20	25		
FMT_YCBCR_422_24	26		
FMT_DSC_COMPRESSED	33		
DCS Write Memory RGB Bit Order			
WM_BIT_ORDER_LEGACY	0		
WM_BIT_ORDER_DCS_ANNEX_A	1		
DPhy Timing Parameters			
DPHY_PARAM_HS_PREPARE	0		
DPHY_PARAM_HS_ZERO	1		
DPHY_PARAM_HS_TRAIL	2		
DPHY_PARAM_HS_EXIT	3		
DPHY_PARAM_CLK_PREPARE	4		
DPHY_PARAM_CLK_ZERO	5		
DPHY_PARAM_CLK_TRAIL	6		
DPHY_PARAM_CLK_PRE	7		
DPHY_PARAM_CLK_POST	8		
DPHY_PARAM_TA_GO	9		
DPHY_PARAM_TA_GET	10		
DPHY_PARAM_WAKEUP	11		
DPHY_PARAM_HS_IDLE_POST	12		
DPHY_PARAM_HS_IDLE_CLKHS0	13		
DPHY_PARAM_HS_IDLE_PRE	14		
DPHY_PARAM_PREAMBLE	15		
DPHY_PARAM_SKEW_CAL	16		
DPHY_PARAM_ALT_CAL	17		
Trigger Commands			

TRIG_CMD_LOW	1		
TRIG_CMD_HIGH	2		
TRIG_CMD_TOGGLE	3		
TRIG_CMD_PULSE	4		
Dual-Interface Modes			
DUAL_IF_MODE_CLONE_IMAGE	0		
DUAL_IF_MODE_DUAL_IMAGE	1		
DUAL_IF_MODE_LEFT_RIGHT_IMAGE	2		
DUAL_IF_MODE_EVEN_ODD_PIXEL	3		
Active Links Modes			
ACTIVE_MASTER	0		
ACTIVE_SLAVE	1		
ACTIVE_MASTER_SLAVE	2		