DPhyDecoderCtl User's Manual

Table of Contents

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	
Documentation		

1 Overview

The Moving Pixel Company DPhy Decoder is an instrument designed to monitor traffic on up to 4 MIPI DPhy lanes running up to 1.5 Gbps.[1] The instrument operates as stand-alone acquisition hardware for DPhy bus activity, and is used in conjunction with controlling software that also provides DSI, CSI-2, and DPhy protocol decode of acquired data.

The DPhy Decoder's main functions are to:

- Convert serial HS packet data to parallel bytes
- Convert encoded LPDT and escape mode data to parallel bytes
- Process DPhy data and package into records
- Provide protocol and state triggering for acquisition
- Provide DSI/CSI-2 packet filtering based on packet type
- Store unfiltered records into memory for acquisition
- Provide link activity indicators, status, and statistics

The DPhy Decoder is controlled from a Windows application called DPhyDecoderCtl, which communicates with the Decoder via a USB connection. Using this application, the user can monitor DPhy bus activity, select modes of operation, configure the triggering and filtering functions of the instrument, and initiate the acquisition and capture of DPhy traffic.

In addition, DPhyDecoderCtl provides post-processing and protocol disassembly of acquired data. For those familiar with Textronix bus support packages that run on the instrument, the disassembly window of DPhyDecoderCtl has a similar look-and-feel. However, extensive improvements have been made to the basic listing functionality, providing sophisticated functions for displaying, filtering, and searching captured data. This includes a video frame summary listing that provides statistics, navigation, viewing, and saving of video frames to files

[1] The DPhy Decoder is a successor instrument to the DPhy Preprocessor (no longer available), which provided front-end dedicated acquisition for a Tektronix Logic Analyzer. With the new DPhy Decoder, no instrument is required.

2 Setup & Installation

This section describes the steps for installing DPhyDecoderCtl on the host computer, which should be a Windows machine running Windows XP or Windows 7 (32-bit or 64-bit). As DPhyDecoderCtl is memory and processing intensive, the computer should have a fast processor and lots of memory (at least 4 GB are recommended).

To install DPhyDecoderCtl, simply execute the setup.exe file provided and step through the installation windows. The DPhyDecoderCtl application is installed by default in the **c:\Program Files\TMPC\DPhyDecoderCtl** directory.

One requirement for running the DPhyDecoderCtl application is the Microsoft .NET Framework 3.5 platform. You can check if .NET 3.5 is installed through **Start->Settings->Control Panel** and double-clicking "Add or Remove Programs". Scroll through the list of applications for "Microsoft .NET Framework 3.5".

If you are installing from an installation CD provided by The Moving Pixel Company, Microsoft .NET 3.5 will be installed automatically (if necessary). On the other hand, if you are running the application installation from a setup file downloaded from our website or received via email, the setup files for installing .NET 3.5 will not be present.

In this case, running the setup.exe on a machine without .NET 3.5 installed will appear to begin .NET installation (asking the user to accept supplemental license terms), but then will subsequently report an error installing system components. If this occurs, the user should close the error report and manually install .NET 3.5.

To install Microsoft .NET 3.5 (without an installation CD from The Moving Pixel Company) your computer must be connected to the internet. The following is a link to the installation executable (note: this file is over 200 MB):

http://download.microsoft.com/download/6/0/f/60fc5854-3cb8-4892-b6db-bd4f42510f28/dotnetfx35.exe

2.1 Installing License Key

DPhyDecodeCtl requires a license key file to be installed to operate. The license key file is a text file called TMPCLicense.txt and is provided by the Moving Pixel Company on purchase. This file contains license strings enabling operation based on the serial number of the DPhy Decoder instrument. To enable multiple instruments on the same machine, the license file will contain multiple strings. After software installation, please copy your TMPCLicense.txt file to the application directory, by default:

C:\Program Files\TMPC\DPhyDecoder (32-bit machines)
C:\Program Files (x86)\TMPC\DPhyDecoder (64-bit machines)

3 DPhy Decoder Concepts

3.1 Data Records

The main function of the DPhy Decoder hardware is to monitor up to four lanes of DPhy bus traffic and package data into records for acquisition and display. All facets of bus communication are captured from lane LP signaling to escape mode sequences to LPDT and HS burst packets.

Depending on the current DPhy bus state, one of two types of data records are generated. While the exact details of the two record types is not necessary for the user to understand, a quick overview may be helpful.

If one or more lanes are transmitting data in an HS burst, then an HS data record is output. Otherwise, an LP data record is output. Both record types have associated status that conveys auxiliary state information about the record including various header and CRC tags, trigger and filtering information, etc.

An HS data record contains information about each lane: either an HS data byte if the lane is currently transmitting an HS packet, or the LP state of the lane if it is not (normally this would be the LP11 state). An LP data record contains information about the LP state of all lanes (including the clock lane) as well as possibly a decoded LP data byte from an escape command. In addition, the LP data record contains what is called the lane zero "sequence state", which is an encoded value representing a state in the LP signaling state machine (e.g. BTA).

3.2 Hardware Filtering

Another function of the DPhy Decoder is to filter records based on certain user criteria, restricting the data acquired so that it contains more useful information to the user. The Decoder hardware implements two filtering mechanisms:

1) LP record filtering based on mode setting ("Delta" and/or "Escape")
2) DSI/CSI packet filtering based on user criteria

The next two sections describe these mechanisms.[2]

3.2.1 LP Record filtering

The LP record filtering modes are extremely useful -- the user is likely to always enable them except for very low-level debugging acquisitions.

[2] Note that the DPhyDecoderCtl software also has support for packet filtering. While both hardware and software filtering achieve a similar end in allowing selective viewing of DPhy traffic, hardware filtering has the additional benefits of saving acquisition memory and decoding time.

3.2.1.1 Delta Mode

One method for filtering LP records is called Delta Mode. When this mode is enabled, only records containing a lane that has changed state from the previous record is output. This significantly compresses the record set captured by the instrument and compacts the disassembly view. As timestamps can indicate duration in a particular LP state, duplicated records are superfluous.

For example, much of the time on the DPhy bus is spent with all lanes in LP11. In delta mode, all such periods will be represented by a single record. Another example is in viewing any LP signaling (HS burst, BTA, Escape, etc). Generally, signal states in these modes persist for one clock period of the LP frequency, which would result tens to hundreds of duplicated records in the instrument. In delta mode, these record sets would be reduced to one record.

An enhancement to delta mode is the option to specify a timeout to force output of periodic records even though no change has occurred. This allows the instrument to finish acquisition even if the bus is static for a long period of time.

3.2.1.2 Escape Mode

A second method for filtering LP records is called Escape Mode. This mode is used to filter records that represent bit transitions in escape mode, allowing only those that occur on packet byte boundaries to be saved to memory. Because of the Spaced-One-Hot coding method used for communication in escape mode, this will increases useful record density by 16x. *Note that if Escape Mode is not set, no decoding of escape bytes is performed.* An implication of this mode not being set is that no filtering or triggering of LPDT packets is performed.

3.2.2 Packet filtering

Another filtering mechanism of the instrument is the ability to filter MIPI packets based on packet type, simplifying the record trace, eliminating records of no interest to the user. This function can also be viewed in the opposite sense as that of *selecting* records of a particular packet type. Using this feature implements focused acquisition to view only specific records of interest.

Packet filtering is applied based on the DataType field in the packet header and applies to both HS and LPDT packets. DCS packets can also be filtered using the DCSCmd byte.

Implementation note: in the case of LP record filtering, filtered records are simply discarded. However, for packet filtering, HS records may contain up to 4 lanes and packets can start and end in the middle of a HS record. In this case, information is contained in the data record indicating which lanes should be filtered allowing the instrument to display data accordingly.

3.3 Triggering

Triggering is the process of starting acquisition based on trigger criteria specified by the user. Once the specified criteria is met, the instrument begins filling up memory with data records. Note that even after acquisition has started, trigger criteria continues to be evaluated and the result is associated with data records for subsequent search and display. Thus, there is no restriction on the number of records that may have their trigger flag set.

In the DPhy Decoder, various trigger modes are supported. Simple modes such as triggering "Immediately" or on "CRC Error" or "ECC Error" don't require any further configuration. Other modes require the user to define predicate terms that are then used in an arbitrary Boolean trigger equation. Often this equation is simply an ANDing or ORing of predicate terms.

The most common form of triggering is based on packet data, i.e. predicate terms are built from comparing masked fields in DSI/CSI packets and compared to constants. Other trigger modes use LP Signal State, lane-zero LP Sequence State (e.g. BTA), or the lane-zero Escape command to build their predicates.

4 Using DPhyDecoderCtl

4.1 Connecting to the Instrument

When the application is started, a connection dialog (see Figure 1) is shown for the user to select a DPhy Decoder instrument powered-on and reachable via USB. This dialog can also be brought up for the user to switch instruments using the option in the Control menu labeled "Connect to DPhy Decoder...".

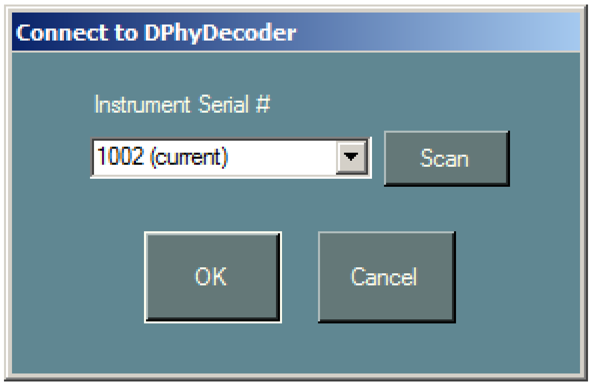


Figure 1 – Connection Dialog

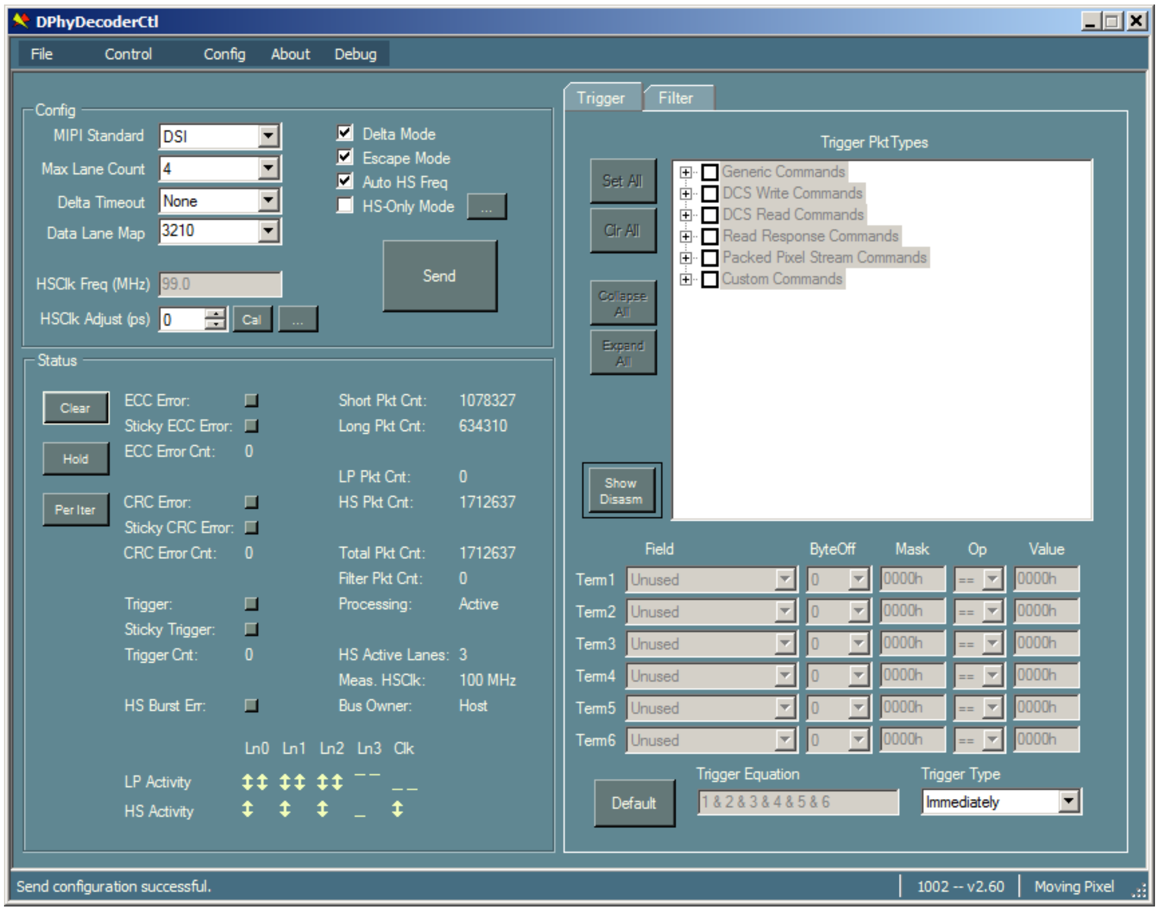
A drop-down box is provided with reachable instrument serial numbers as well as an option called "Offline". Offline mode allows using the application to be used without a connection to real hardware, useful for demonstration of the GUI and building and saving configurations. There is also a "Scan" button that the user can press to request that the application search for any new instruments that have recently been powered on.

Note that while multiple copies of the application can run at the same time, only one instance can be connected to a given instrument at a time. Serial numbers of instruments that are in-use by another copy of the application are marked "(in use)".

To connect to an instrument, simple select the appropriate serial number and click OK. Note that the serial number and firmware version of the currently connected instrument appears in the status bar at the bottom right of the main window.

4.2 Main Window

Once connected, the main window is shown (see Figure 2). The instrument serial number is shown in a bottom right pane on the status bar. The main window contains all the user-controls for configuring and operating the instrument.



Config
MIPI Standard: DSI
Max Lane Count: 4
Delta Timeout: None
Data Lane Map: 3210
HSClk Freq (MHz): 99.0
HSClk Adjust (ps): 0
☑ Delta Mode
☑ Escape Mode
☑ Auto HS Freq
☐ HS-Only Mode

Status

ECC Error:	■	Short Pkt Cnt:	1078327
Sticky ECC Error:	■	Long Pkt Cnt:	634310
ECC Error Cnt:	0		
		LP Pkt Cnt:	0
CRC Error:	■	HS Pkt Cnt:	1712637
Sticky CRC Error:	■		
CRC Error Cnt:	0	Total Pkt Cnt:	1712637
		Filter Pkt Cnt:	0
Trigger:	■	Processing:	Active
Sticky Trigger:	■		
Trigger Cnt:	0	HS Active Lanes:	3
		Meas. HSClk:	100 MHz
HS Burst Err:	■	Bus Owner:	Host

	Ln0	Ln1	Ln2	Ln3	Clk
LP Activity	⇕⇕	⇕⇕	⇕⇕	‾‾	__
HS Activity	⇕	⇕	⇕	_	⇕

Trigger PktTypes
☐ Generic Commands
☐ DCS Write Commands
☐ DCS Read Commands
☐ Read Response Commands
☐ Packed Pixel Stream Commands
☐ Custom Commands

	Field	ByteOff	Mask	Op	Value
Term1	Unused	0	0000h	==	0000h
Term2	Unused	0	0000h	==	0000h
Term3	Unused	0	0000h	==	0000h
Term4	Unused	0	0000h	==	0000h
Term5	Unused	0	0000h	==	0000h
Term6	Unused	0	0000h	==	0000h

Trigger Equation: 1 & 2 & 3 & 4 & 5 & 6
Trigger Type: Immediately

Send configuration successful. | 1002 -- v2.60 | Moving Pixel

Figure 2 – DPhyDecoderCtl GUI

The main window is divided into four areas of controls: configuration, status, trigger control, and filter control (the last two share a tabbed page). The following sections describe each of these areas.

4.2.1 Configuration Controls

The section of the main window provides controls that support overall mode configuration and are located in the upper-left section of the main window. They are:

MIPI Standard – drop-down control to select the DSI or CSI standard. This selection should be made first as it naturally resets the filtering and triggering options available to the user.

Max Lane Count – drop-down control to select the maximum number of lanes to monitor for activity. Unused lanes are forced to LP11 for the purposes of instrument monitoring and acquisition. Note that the instrument will monitor all DPhy bus activity less than or equal to the Max Lane Count setting. For example, with a Max Lane Count setting of 4, all HS bursts using 1, 2, 3, or 4 lanes will be detected automatically and the actual number of lanes participating in ongoing HS bursts is displayed in the status region.

Delta Mode – check box to enable delta mode filtering, which restricts record output to records that reflect a state change since the last record. See section 3.2.1.1 for more details.

Delta Timeout – drop-down control to set the delta mode time out duration. This setting is the maximum time to wait without outputting a record before forcing a record output. If timeout is set to "None", the instrument will wait indefinitely for a change in inputs.

Escape Mode – check box to enable escape mode filtering, which restricts LP record output during escape sequences to those that contain a decoded byte (as opposed to intermediate bits). See section 3.2.1.2 for more details.

Note that Escape Mode is a required setting if the user is to trigger/filter based on packets sent in LPDT mode. If Escape Mode is not set, records that contain the decoded LPDT byte will be interleaved with intermediate records and packet triggering/filtering will not apply to these records.

Data Lane Map – combo box to configure a logical-to-physical mapping of data lanes connected to the instrument. Each digit in the four-digit value selected represents a logical lane assignment. Each position in the four-digit value represents a physical lane connector (starting with lane 3 down to lane 0). Thus, the default setting of "3210" represents a unity mapping of logical lane to physical lane connector. Setting the map value to "0123", on the other hand, will decode the lane 3 connector as lane 0, lane 2 connector as lane 1, etc.

HSClk Freq – text box for the user to indicate the approximate lane clock frequency. This value is used to set the sampling phase of the data relative to the clock on input. Its accuracy becomes more important with increasing frequency, given smaller sampling windows. Generally, it is best to use auto-detection to set this value.

Auto HS Freq – check box to enable the instrument to automatically detect the HS clock frequency. When checked, the HSClk Freq text box is disabled and automatically filled in with the detected HSClk frequency.

This mode is very useful. However, HSClk frequency analysis can be inaccurate when the lane clock is turning on and off with minimal time in HS mode. In these cases, the instrument may perform better with auto HS frequency detection turned off and a manual frequency setting entered by the user.

Also, when this mode is enabled, and the clock frequency changes enough to require a new sampling phase in the hardware, there may be a brief period when errors are detected before the frequency and sampling phase change is completed. Thus, as a convenience, all statistics are reset when a significant clock change is detected.

HS-Only Mode – check box to allow synchronizing to a stream that has no transitions to LP11. This box should be unchecked for normal operation.

HS-Only operation causes the instrument to search for a known 32-bit header pattern in the incoming stream. When this pattern is found, the box locks to this packet alignment and falls back to normal operation.

To synchronize to an HS-only stream for the first time, click the button labeled "…" next to this checkbox and set the 32-bit header pattern to use for synchronization. It can be any header pattern that repeatedly occurs in the stream and doesn't occur in packet payload data (or at least occurs infrequently). Clicking the Default button in the pattern entry dialog causes the DSI HS Sync Start packet pattern to be entered (12000021h).

Once the pattern is set, checking the HS-Only Mode checkbox initiates synchronization. HS-Only synchronization also occurs whenever HS-Only Mode is checked and the Send button is clicked.

HSClk Adjust – up/down control to adjust fine delay timing on the incoming HSClk for very high-frequency acquisitions (i.e. 600-750 MHz). This timing control adjusts the clock *relative to the ideal quadrature setting* for acquisition and, normally, this value should thus be set to zero. However, modest adjustments are sometimes required depending on less-than-ideal acquisition conditions depending on input signal integrity, non-standard voltages, cabling, noise, or timing skew on the bus.

Cal – button to automate HSClk adjustment. To use, set up the system so valid HS traffic is continuously sent on the bus. Then click this button to initiate a sweep search within +/- 400 ps around zero in 20 ps increments, measuring the number of ECC and CRC errors received in a short time period. The sweep takes a few seconds and when complete, the optimum HSClk adjustment setting is displayed and assigned to the control.

Send – button to send the current configuration (including filter and trigger settings) to the Decoder. This is required operation to update changes made in the triggering or filtering configuration. The user is notified that Send should be clicked by the appearance of the text "Modified" to the lower-right of the Send button. Note that changes to settings described in this section – configuration controls – are sent immediately to the DPhyDecoder and do not require a Send button click.

4.2.2 Filter Controls

The filter control page is one of two tab pages occupying the right side of the main window. To show the page, click on the Filter tab at the top.

The central feature of the filter control section is a checked tree-view containing all packet types. Top-level nodes in the tree-view consist of categories of packet types and second-level nodes in the tree-view represent the packet types themselves. Category nodes are not able to be checked by the user, but they are automatically checked if at least one node within the category is checked.

To select a packet type to be filtered, the user simply opens the tree node category (if closed) and checks the packet type to be filtered by the Decoder (and then clicks the Send button). When a packet it filtered by the DPhyDecoder hardware, if it spans one or more full data records, these records are simply discarded before acquisition into memory. Otherwise, filter bits are set in the acquired record for software to hide it from view during disassembly.

For DSI, the top-level categories are:

- Generic Commands
- DCS Write Commands
- DCS Read Commands
- Read Response Commands
- Packed Pixel Stream Commands
- Custom Commands

For CSI, the top-level categories are:

- Generic Commands
- YUV Commands
- RGB Commands
- Raw Commands
- User Commands
- Custom Commands

These categories are mostly self-explanatory, based on the packet type definitions for each respective MIPI specification. The Generic Commands category includes all packet types defined in the specification that do not fall in any of the other categories. The Custom Commands category includes user-defined data types using the Custom Command dialog (Config -> Define Custom Commands… menu option).

Other controls in the Filter tab section are:

Set All – button that checks all the packet types in the tree-view

Clr All – button that clears all the checked packet types in the tree-view

Collapse All – button the collapses all the category nodes in the tree-view

Expand All – button that expands all the category nodes in the tree-view

Show Disasm – button that shows the disassembly window.

4.2.3 Trigger Controls

The trigger control page is one of two tab pages occupying the right side of the main window. To show the page, click on the Trigger tab at the top.

As outlined in section 3.3, the DPhy Decoder has several triggering modes. The triggering mode is selected via the Trigger Type drop-down control at the bottom-right corner of the Trigger Controls section. The trigger mode should be selected first before other fields as the control options change based on mode. The trigger modes available are:

- **Packet Fields –** This is the most common form of triggering. Comparisons of packet types and packet fields are used to create predicates evaluated in a Boolean equation to determine trigger state.
- **LP Signal State –** This mode uses comparisons of LP signal state for any or all lanes (including the clock lane) as predicates evaluated in a Boolean equation to determine trigger state.
- **LP Sequence State –** This mode uses comparisons of the lane zero sequence state (e.g. BTA) as predicates evaluated in a Boolean equation to determine trigger state.
- **Escape Command –** This mode uses comparisons the escape command byte as predicates evaluated in a Boolean equation to determine trigger state.
- **CRC Error –** This mode sets the trigger flag on CRC error.
- **ECC Error –** This mode sets the trigger flag on ECC error.
- **CRC or ECC Error –** This modes set the trigger flag on either a CRC or ECC error.
- **Immediately –** This mode causes immediate triggering once the Run button is clicked.

4.2.3.1 Packet Fields Triggering

When triggering based on packet fields, triggering is a two stage process. The first stage selects the set of packets based on packet type for further evaluation.[3] The second stage then evaluates these qualifying packets using the designated predicate terms and trigger equation to determine the trigger flag state.

4.2.3.1.1 Stage 1 Trigger Selection

The first stage selection process of packets for trigger evaluation is identical to packet filtering for output. As with the filter control section, the trigger control section has a checked tree-view containing all packet types. Top-level nodes in the tree-view consist of categories of packet types and second-level nodes in the tree-view represent the packet types themselves. Category nodes are not able to be checked by the user, but they are automatically checked if at least one node within the category is checked.

To select a packet type to be included in trigger evaluation, the user simply opens the tree node category (if closed) and checks the desired packet type. Note that in this mode, if no packets are selected, *no trigger will result*, regardless of the trigger equation settings.

For DSI, the top-level categories are:

[3] Note that both HS packets and LPDT packets can be considered for triggering in this mode, but only if Escape Mode is enabled. Otherwise only HS packets are considered.

- Generic Commands
- DCS Write Commands
- DCS Read Commands
- Read Response Commands
- Packed Pixel Stream Commands
- Custom Commands

For CSI, the top-level categories are:

- Generic Commands
- YUV Commands
- RGB Commands
- Raw Commands
- User Commands
- Custom Commands

These categories are mostly self-explanatory, based on the packet type definitions for each respective MIPI specification. The Generic Commands category includes all packet types defined in the specification that do not fall in any of the other categories. The Custom Commands category includes user-defined data types using the Custom Command dialog (Config -> Define Custom Commands… menu option).

Other controls in this section are:

Set All – button that checks all the packet types in the tree-view

Clr All – button that clears all the checked packet types in the tree-view

Collapse All – button the collapses all the category nodes in the tree-view

Expand All – button that expands all the category nodes in the tree-view

Show Disasm – button that shows the disassembly window.

4.2.3.1.2 Stage 2 Trigger Selection

Stage-two trigger selection defines the predicates and the trigger equation to be used for final trigger evaluation. Six predicate terms are supported where each term targets a 16-bit field (byte aligned) anywhere in packet bytes 1-11. This byte range allow all fields in short packets and long DCS packets with up to three 16-bit arguments to be fully supported. Predicates are defined by masking the field and comparing it to a value. After the predicate terms have been defined, the user specifies a logical trigger equation that uses the field comparison results.

The following list describes the controls used for term definitions:

- **Field –** drop-down box identifying the field name. This use of this control is not required, as it simply acts as a "preset" for other controls in the term. However, if this field is used, it should be selected first, as it will overwrite the other term controls with appropriate values.

The first option is always "Unused" which sets the term controls to an always true comparison. This allows it to be included in the default trigger equation (all terms ANDed) with no effect. This selection is merely a convenience for readability because the trigger equation actually specifies whether a term is used or not and, thus, it is not necessary to set unused terms to this setting.

The second option in the field drop-down control is always "Custom", which enables all the other term controls for user settings.

Next, follow common generic packet header fields (i.e. VC, DataType, DataID, WordCount, ECC) which initialize the byte offset and mask fields appropriately.

Finally, any additional options vary with the selected packet type. If a single packet type with known fields is selected in stage 1, then they are included as options here.

- ByteOff – drop-down box to set the byte offset of the field. For example, a byte offset of 4 would specify the field to be the first two payload bytes of a packet. (Note in this case that all of the selected packets from stage 1 should be long packets as byte 4 is not defined for short packets.).
- Mask – text box to set the mask value. This value is treated as a 16-bit value and ANDed with the packet field before comparison with the term value. Values are parsed as decimal unless appended with an 'h', in which case they are parsed as hexadecimal (i.e. 20h = 32 decimal). The LSB corresponds to the earlier byte in the packet. For example, if the mask were set to FF00h with ByteOff == 4, the resulting masked value would have packet byte 5 in the MSB and zero in the LSB.
- **Op** – drop-down box to select the predicate comparison operation. All logical comparisons are supported, i.e. ==, !=, >, >=, <, <=.
- **Value** – text-box to set the term comparison value. Values are parsed as 16-bit decimal numbers unless appended with an 'h', in which case they are parsed as 16-bit hexadecimal (i.e. 20h = 32 decimal). As with the mask value, the LSB represents the earlier packet byte.

The final controls used for stage two triggering are the trigger equation specification itself and the default button:

- **Trigger Equation** – text box to specify how term predicates should be combined in a Boolean equation to determine the final trigger state. Terms predicates are identified using the numbers '1' through '6'. The '&' and '|' symbols for AND and OR operations and parentheses can be used to set precedence (e.g. "1 & (2 | 3) & 4").

- **Default** – button to reset all term controls and the trigger equation to their default. The default setting is for all term controls to be "Unused" and the trigger equation to be "1 & 2 & 3 & 4 & 5 & 6".

4.2.3.2 LP Signal State Triggering

When triggering based on LP Signal State, the packet field controls are disabled and triggering is solely determined by predicate term and trigger equation definition.[4] Please refer to section 4.2.3.1.2 for an overview of these controls.

The only difference in this mode is that the field options for the predicate terms operate on a composite value derived from LP records. This composite value is 10 bits and indicates the current LP state of all lanes including the clock lane. The five 2-bit fields represent the LP state of each lane, where 0 = LP00, 1 == LP01, 2 == LP10, and 3 == LP11.

Bit definitions for the fields are as follows:

[1:0]	lane 0 LP state
[3:2]	lane 1 LP state
[5:4]	lane 2 LP state
[7:6]	lane 3 LP state
[9:8]	clock LP state

For example, a mask of 0x300 and a value of 0x100 define a predicate to test if the clock lane is in the LP01 state.

For convenience, the field drop-down box in this mode contains entries for all combinations of lane and state, i.e. "Lane0 in LP00", "Lane0 in LP01"... "Clock in LP10", "Clock in LP11". Also, note that the ByteOff field is unused in this mode and is disabled.

4.2.3.3 LP Sequence State Triggering

When triggering based on LP Sequence State, the packet field controls are disabled and triggering is solely determined by predicate term and trigger equation definition.[4] Please refer to section 4.2.3.1.2 for an overview of these controls.

The only difference in this mode is that the field options for the predicate terms operate solely on the lane zero sequence state. DPhy defines various LP signaling sequences such as BTA, ESCAPE, and HS and triggering is possible on the occurrence of any of them. While each state and intermediate states are represented by a particular field value, only a few are generally useful. These particular states – "HS Burst", "BTA", "BTA Exit" and "Escape Entry" – are available for convenience in the field drop-down box.

[4] Note that in this mode, triggering is disabled for the special case where all terms are set to "Unused, regardless of the trigger equation.

For completeness, Table 1 documents all the possible sequence states. Numbers in the state names indicate receipt of a particular LP signal value (i.e. 0 = LP00, 1 = LP01, 2 = LP10, 3 = LP11). For example, if the sequence state is LPSTATE_3, lane 0 is currently in the LP11 state. If in LPSTATE_32, lane 0 has seen LP11 followed by LP10.

Table 1 – Lane 0 Sequence States

State	**Value**
LPSTATE_UNKNOWN	0
LPSTATE_3	1
LPSTATE_32	2
LPSTATE_320	3
LPSTATE_3202	4
LPSTATE_BTA	5
LPSTATE_BTA_EXIT	6
LPSTATE_3201	7
LPSTATE_ESCAPE	10
LPSTATE_31	11
LPSTATE_HS	12

4.2.3.4 Escape Command Triggering

When triggering based on the Escape Command, the packet field controls are disabled and triggering is solely determined by predicate term and trigger equation definition.[4] Please refer to section 4.2.3.1.2 for an overview of these controls.

The only difference in this mode is that the field options for the predicate terms operate on the escape command value.

For convenience, the field drop-down box in this mode contains entries for predefined escape commands in the DPhy specification. These are: "LPDT", "ULPS", "Reset Trigger", "Trigger 1", "Trigger 2", and "Trigger 3". The user may also select "Custom Escape Cmd" to define other command values.
.

4.2.3.5 CRC Error Triggering

When triggering on CRC errors, all other trigger controls are disabled. The trigger flag is set for packets that have a CRC error.

4.2.3.6 ECC Error Triggering

When triggering on ECC errors, all other trigger controls are disabled. The trigger flag is set for packets that have an ECC error.

4.2.3.7 Trigger Immediately

When triggering immediately, acquisition begins immediately after the Run button is clicked. The trigger flag is not set for any packet.

4.2.4 Status Controls

When connected to a DPhyProcessor instrument, the status control section provides continuously updated statistics about bus activity as monitored by the Decoder. Status updates occur about once a second.

The status display generally falls into three categories: activity, statistics, and statistics control.

4.2.4.1 Activity

The activity display at the bottom of the status pane shows graphically the LP and HS activity occurring on each lane. The following notes describe the activity display:

- Activity for the LP+, LP-, and HS signals or each lane is represented by an indicator graphic.
- The top row for LP activity consists of 10 indicators, a pair for each lane, with the left indicator in each pair representing LP+ and the right indicator representing LP-.
- The bottom row for HS activity consists of 5 indicators, one for each lane.
- Indicators can be low (low horizontal bar), high (high horizontal bar), or transitioning (vertical bar with arrows on both ends).
- The indication spans the period since the last status update, so for example, if a toggling indicator appears, it means that the signal transitioned at least once since the last update.
- Indicators are valid and updated even if the acquisition pipeline is stalled waiting for an event (i.e. instrument Output is inactive -- for example, while in delta mode with no timeout and a quiescent bus).

4.2.4.2 Statistics

Many statistic about data traffic are collected and displayed in this section, the predominant part of the status control display. Each statistic is summarized below:

The ECC byte in packet headers check the integrity of the packet header. The ECC is tested for packets sent in both HS and LPDT mode if Escape Mode is enabled. Otherwise, the ECC is tested (and errors can be triggered on) only in HS packets. Controls and statistics relating to ECC status are:

- **ECC Error** – LED-control indicating whether an ECC error occurred since the last update. If red, an error occurred; if green, no error occurred.

- **Sticky ECC Error –** LED-control indicated whether an ECC error occurred since the last clear. If red, an error occurred; if green, no error occurred.
- **ECC Error Cnt –** counts the number of ECC errors that occurred since the last clear.[5]

The CRC bytes at the end of long packets check the integrity of the packet payload for long packets. The CRC is tested for packets sent in both HS and LPDT mode if Escape Mode is enabled. Otherwise, the CRC is tested (and errors can be triggered on) only in HS packets. Controls and statistics relating to CRC status are:

- **CRC Error –** LED-control indicating whether a CRC error occurred since the last update. If red, an error occurred; if green, no error occurred.
- **Sticky CRC Error –** LED-control indicated whether a CRC error occurred since the last clear. If red, an error occurred; if green, no error occurred.
- **CRC Error Cnt –** counts the number of CRC errors that occurred since the last clear.[5]

The trigger flag is set whenever the conditions defined by the trigger controls are met. Controls and statistics relating to trigger status are:

- **Trigger –** LED-control indicating whether a trigger event occurred since the last update. If light blue, a trigger occurred; if green, no trigger occurred.
- **Sticky Trigger –** LED-control indicated whether a trigger event occurred since the last clear . If light blue, a trigger event occurred; if green, no trigger occurred.
- **Trigger Cnt –** counts the number of trigger events that occurred since the last clear.[5]

Packet count statistics summarize the number of occurrences of each type of packet.[6] Statistics relating to packet counts are:

- **Short Pkt Cnt –** counts the number of short packets since the last clear.
- **Long Pkt Cnt –** counts the number of long packets since the last clear.
- **LP Pkt Cnt –** counts the number of LS packets since the last clear.
- **HS Pkt Cnt –** counts the number of HS packets since the last clear.
- **Total Pkt Cnt –** counts the total number of packets since the last clear.
- **Filter Pkt Cnt –** counts the number of packets filtered since the last clear.

Lastly, other miscellaneous statistics provided are:

[5] Note when any of the ECC, CRC, or Trigger counts reach 999 million, they are all reset to zero and continue counting.
[6] Note when any of these packet counts reach 999 million, all counts reset to zero and continue counting.

- **Processing**– displays "Active" if valid records are being processed and the internal pipeline of the instrument is moving. Two common cases when the processing will be inactive are when in Delta Timeout mode with no timeout and a quiescent bus, and when in Escape mode when the Escape entry sequence has been seen but no transitions are occurring, in particular, when the lane is in ULPS.
- **HS Active Lanes** – Indicates the maximum number of lanes that were part of an active HS burst since the last clear button reset. If zero, no HS packets were seen.
- **Meas. HSClk** – Indicates the measured HS clock over a short time period (milliseconds). Note that this value is correct only if the clock does not turn on and off during the measurement period. Otherwise, the value displayed represents the average clock frequency.

4.2.4.3 Statistics Control

Three buttons control how statistics are displayed:

- **Clear** – resets all statistics to zero. The clear function is also automatically applied when the Send button is pressed or when in Auto HS Freq and the detected frequency changes by more than 10 MHz.
- **Hold / Resume** – suspends or resumes statistics update. The button label alternates with each click, indicating the function to be performed when next pressed. For example, when "Resume" is displayed, statistics update is halted until the button is pressed.
- **Totals / Per Iter** – shows total counts or per update counts. The button label alternates with each click, indicating the function to be performed when next pressed. For example, when "Totals" is displayed, all statistics except for the sticky errors and trigger, summarize activity only since the last update (rather than since the last clear).

4.3 Disassembly Window

The disassembly window uploads, decodes, and displays captured records from the DPhy Decoder instrument. This window can be brought up from the main window by clicking on the "Show Disasm" button in the Trigger pane (center screen).

Figure 3 shows a disassembly listing showing the capture of a DSI video sequence. The control areas of the disassembly window can be broken down as follows:

- **Run/Stop Button** – This button at the top-right of the window controls the start of acquisition. It can force a trigger if desired and can also be used to abort long processing tasks. Depending on the current processing state, it's name may display "Run", "Stop", or "Abort".
- **Listing Grid** – The listing grid is the central feature of the disassembly window. It contains decoded records and their fields.

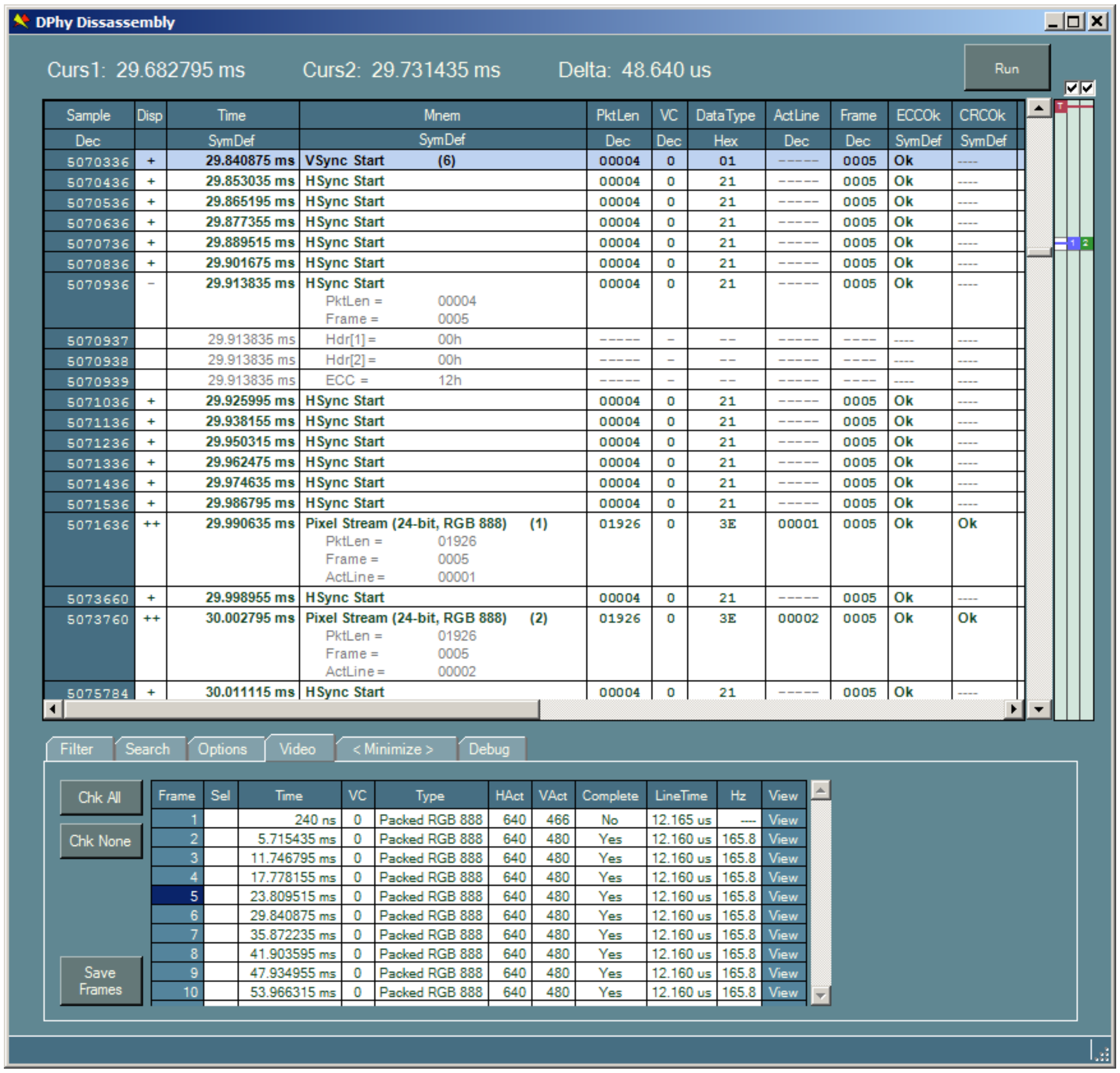
- **Cursor Status Display** – The top part of the window shows a text readout of the two cursor locations (time relative to start of trace). The time delta between the two cursors is also displayed.
- **Cursor Control Panel** – The cursor panel is a control next to the vertical scroll bar of the listing grid (far right) that spatially indicates relevant locations in the trace, specifically: trigger location, cursor locations, and current view location.
- **Disassembly Control Panel** – The tabbed control beneath the listing grid contains all the controls associated with disassembly. A separate tab is provided for the functions of filtering, searching, acquisition and disassembly options, and video frame summary.

4.3.1 Run/Stop Button

When the DPhy Decoder is connected and monitoring a DPhy bus that has activity, clicking the Run button initiates an acquisition. Acquisition is configured via the options tab of the disassembly control panel, in particular how much memory to use, where to position the trigger record and whether to include timestamp records (see section 4.3.4.3).

The acquisition process has several phases:

- **Prefill** – memory is "prefilled" with data records before the trigger criteria are evaluated. This allows placement of the initial trigger record position in the overall trace capture (e.g. at the start, in the middle, near the end, etc).



Curs1: 29.682795 ms Curs2: 29.731435 ms Delta: 48.640 us

Sample	Disp	Time	Mnem	PktLen	VC	DataType	ActLine	Frame	ECCOk	CRCOk
Dec		SymDef	SymDef	Dec	Dec	Hex	Dec	Dec	SymDef	SymDef
5070336	+	29.840875 ms	VSync Start (6)	00004	0	01	-----	0005	Ok	----
5070436	+	29.853035 ms	HSync Start	00004	0	21	-----	0005	Ok	----
5070536	+	29.865195 ms	HSync Start	00004	0	21	-----	0005	Ok	----
5070636	+	29.877355 ms	HSync Start	00004	0	21	-----	0005	Ok	----
5070736	+	29.889515 ms	HSync Start	00004	0	21	-----	0005	Ok	----
5070836	+	29.901675 ms	HSync Start	00004	0	21	-----	0005	Ok	----
5070936	–	29.913835 ms	HSync Start PktLen = 00004 Frame = 0005	00004	0	21	-----	0005	Ok	----
5070937		29.913835 ms	Hdr[1] = 00h	-----	–	--	-----	----	----	----
5070938		29.913835 ms	Hdr[2] = 00h	-----	–	--	-----	----	----	----
5070939		29.913835 ms	ECC = 12h	-----	–	--	-----	----	----	----
5071036	+	29.925995 ms	HSync Start	00004	0	21	-----	0005	Ok	----
5071136	+	29.938155 ms	HSync Start	00004	0	21	-----	0005	Ok	----
5071236	+	29.950315 ms	HSync Start	00004	0	21	-----	0005	Ok	----
5071336	+	29.962475 ms	HSync Start	00004	0	21	-----	0005	Ok	----
5071436	+	29.974635 ms	HSync Start	00004	0	21	-----	0005	Ok	----
5071536	+	29.986795 ms	HSync Start	00004	0	21	-----	0005	Ok	----
5071636	++	29.990635 ms	Pixel Stream (24-bit, RGB 888) (1) PktLen = 01926 Frame = 0005 ActLine = 00001	01926	0	3E	00001	0005	Ok	Ok
5073660	+	29.998955 ms	HSync Start	00004	0	21	-----	0005	Ok	----
5073760	++	30.002795 ms	Pixel Stream (24-bit, RGB 888) (2) PktLen = 01926 Frame = 0005 ActLine = 00002	01926	0	3E	00002	0005	Ok	Ok
5075784	+	30.011115 ms	HSync Start	00004	0	21	-----	0005	Ok	----

Frame	Sel	Time	VC	Type	HAct	VAct	Complete	LineTime	Hz	View
1		240 ns	0	Packed RGB 888	640	466	No	12.165 us	----	View
2		5.715435 ms	0	Packed RGB 888	640	480	Yes	12.160 us	165.8	View
3		11.746795 ms	0	Packed RGB 888	640	480	Yes	12.160 us	165.8	View
4		17.778155 ms	0	Packed RGB 888	640	480	Yes	12.160 us	165.8	View
5		23.809515 ms	0	Packed RGB 888	640	480	Yes	12.160 us	165.8	View
6		29.840875 ms	0	Packed RGB 888	640	480	Yes	12.160 us	165.8	View
7		35.872235 ms	0	Packed RGB 888	640	480	Yes	12.160 us	165.8	View
8		41.903595 ms	0	Packed RGB 888	640	480	Yes	12.160 us	165.8	View
9		47.934955 ms	0	Packed RGB 888	640	480	Yes	12.160 us	165.8	View
10		53.966315 ms	0	Packed RGB 888	640	480	Yes	12.160 us	165.8	View

Figure 3 – Disassembly Window

- **Trigger** – trigger criteria are evaluated. While waiting for the trigger condition to evaluate true, prefill records continue to be acquired, discarding the oldest captured records to make room for new records. During this time, the status displayed in the status bar will be "Waiting for trigger…". Also, the label of the Run button will change to "Stop". Clicking the Stop button will force a trigger condition, moving the process to the next phase (Capture).
- **Capture** – capture memory is filled with data records. Because filter settings can cause the rate of acquisition to be very slow (or non-existent) a timeout is used to end this phase. If capture was forced by clicking on the Stop button, a one second timeout is used. Otherwise, a five second timeout is used. If a timeout occurs, a partial capture buffer is uploaded. During this time, the status displayed in the status bar will be "Waiting for capture buffer to fill…"

- **<u>Upload</u>** – capture memory is uploaded from the DPhy Decoder instrument into application memory. Upload speeds are about 20 MB/s. Thus, for a 128 MB buffer, upload would take about 6 seconds. During this phase, the Run button label will be set to "Abort" . Clicking the Abort button halts upload and discards any partially uploaded data.
- **<u>Decode</u>** – captured records are decoded and displayed in the listing grid. Generally, this is a fairly quick process, though depending on the filter and disassembly settings, record content, and the number of captured records, this can take some time. Progress is indicated via the status bar, with "Building records…" and a percent complete. The Run button label remains set to "Abort" for the user to request a halt to decoding. If aborted, partial results are displayed.[7]

4.3.2 Listing Grid

The listing grid displays one record per line, arranged in order from earliest in time at the top to latest in time at the bottom. Generally, only a subset of records is shown in the listing at any time, based on settings in the filter tab of the disassembly control panel, the global mnemonic view state, and the individual mnemonic view state for packets.[8]

Please note a distinction between raw data records acquired by the hardware and disassembly records displayed in the listing grid. For LP data records, there is a one-to-one correspondence between data records and disassembly records. For HS data records, there are N disassembly records for every data record, where N is the number of active lanes. Another way of putting it is that disassembly records in HS mode decode one HS packet byte. For the remainder of this document, the term "record" will refer to a disassembly record.

Columns in the grid represent data fields associated with records, whether direct content acquired from the DPhy bus, derived data from that content, or other attributes created by the application. Not all fields apply to all record types, in which case a field will generally contains dashes.

The listing grid contains vertical and horizontal scroll bars for scrolling visible rows and columns in the grid. Other mechanisms are available for scrolling or selecting viewable rows in the listing including: key strokes, clicking in the cursor panel, selecting among several context menu options, and buttons in the search and video tabs of the disassembly control panel.

4.3.2.1 Column (Field) Manipulation

The user has control over what columns are displayed, how they are ordered, and how field data are formatted in the column grid. Columns consist of a header in the first row, a radix selection in the second row, followed by field data starting with the third row. The following sections outline field- or column-related functions.

[7] In the event that the user wants to obtain a full decoding after aborting, clicking the "Reanalyze" button in the Options tab reinitiates decoding.
[8] These concepts are described further later in this section.

4.3.2.1.1 Selecting Columns

Similarly, single columns can be selected by right-clicking on the column header (i.e. the field name of a column). A contiguous region of columns can be selected by first selecting a column then selecting a second column while holding the shift key down. In addition, a discontinuous set of columns can be selected by clicking while holding the control key down. When a set of columns is selected, certain options in the column context menu will apply to each column, e.g. "Delete Columns", "Restore Select Column Width(s)"

4.3.2.1.2 Selecting Rows

(While not a column-related function, it makes sense to document this here.) Single rows can be selected by right-clicking on the row header (i.e. the Sample field of a record). A contiguous region of rows can be selected by first selecting a row then selecting a second (later) row while holding the shift key down. Once selected, rows can by copied to the clipboard with Ctrl-C or the "Copy Selected Rows" menu option in the Edit menu.

4.3.2.1.3 Add Column(s)

To add a column field in the listing grid, right-click on a column header and select the "Add Column(s)" menu option to bring up the Add Column dialog. Check the desired column fields to display and click OK.

4.3.2.1.4 Remove Column(s)

To remove a column field in the listing grid, select one or more columns and press the Delete key. Alternatively, right-click in one of the selected column headers and choose "Delete Column(s)". If multiple columns are to be deleted, the user is asked for confirmation.

4.3.2.1.5 Reordering Columns

To change the display order of a visible column, right-click and drag the column header to a new position.

4.3.2.1.6 Setting Column Widths

Column widths are automatically set to accommodate the widest string or value to be displayed and are updated whenever the field radix is changed. However, the user may adjust column width by selecting the right edge of a column header and dragging it. A column's width may be restored to its default using the column context menu option "Restore Selected Column Width(s)" or simply double-clicking on the column header.

4.3.2.1.7 Setting the Field Radix

Most fields are numeric and have an underlying value associated with it (Boolean fields are also numeric and are associated with a 0 or 1 value). Numeric fields can be displayed as binary, decimal, or hexadecimal by right-clicking the radix context menu of a particular column sub-header (second row in grid). In addition, some fields have a default symbolic association -- basically a built-in number-to-string mapping -- that may be available in the menu. Finally, a user can define his own symbol file to use for number-to-string translation (see section 4.3.2.1.9).

4.3.2.1.8 Setting the Time Field Reference

The time field displays the time-stamp of a record, indicating when the data was acquired relative to the start of trace capture (see section 4.3.2.2.4). DPhyDecodeCtl supports viewing time-stamps relative to a global time reference selected by the user via the radix context menu associated with the time field. Records occurring before the global time reference have a negative time field value and records occurring afterwards have a positive time field value.

The following time field display options are provided in the Time radix context menu:

- Relative to Start – display time-stamps relative to the first record.
- Relative to Trigger – display time-stamps relative to the trigger record
- Relative to Cursor1 – display time-stamps relative to cursor1
- Relative to Cursor2 – display time-stamps relative to cursor2
- Relative to Previous – display time-stamps relative to the preceding *visible* record

4.3.2.1.9 Defining and Using a Symbol File

To associate a radix file to a numeric field, select the "<New Symbolic File>" option in the radix context menu. This brings up an open file dialog to browse for and select the new symbol file. Once the symbol file has been selected, the context menu will enable the "Symbolic (File)" option (and it will be checked).
Symbol files are text files containing number-to-string mappings, where each mapping consists of a boolean predicate equation and a string. During disassembly, the equation from each mapping is evaluated in order for each field value and the string from the first mapping whose equation evaluates true is displayed.

Predicate equations consist of one or more terms, one term per line. In multi-term equations, all but the last term in the equation are followed by "AND" or "OR" forming a boolean equation from the terms. Finally, the last term of a predicate equation is followed by a colon, and then the symbolic string to associate with the equation enclosed in quotes.

Each predicate term consists of a relational operator and a value. Relational operators are ">", "<", ">=", "<=", "==", and "!=". Values are interpreted as decimal unless an 'h' or a 'b' is appended to denote hexadecimal or binary respectively. In addition, for hex or binary values, the character 'x' may be used as a "don't care" digit (if the first character is an 'x' it represents don't care digits up to the width of the field)

For example, the following lines define a predicate equation to detect values between 0x10 and 0x40 or if a value is odd:

```
// predicate equation
>  10h  AND
<  40h  OR
== x1b  :   "Range between (10h, 40h) or odd"
```

For values in the disassembly where this equation evaluates true, the string "Range between (10h, 40h) or odd" will be displayed.

Some notes about symbol file usage:

- Note that blank lines and lines that begin with "//" (which can be used for comments) are ignored by the parser.
- Predicate terms are interpreted as a sum-of-products equation and so AND has higher precedence than OR. Thus, predicates that describe the equation "a AND b OR c AND d" is parsed "(a AND b) OR (c AND d)".
- White space is **required** between the relational operator, the value, the "AND", "OR" and ":" delimiters, and the string.
- Mappings are evaluated in order, so subsequent predicate equations do not need to exclude values that would have already satisfied prior mappings.
- If not mapping qualifies for a value, a string with "?" is displayed. To avoid this, a default mapping can be appended to the end of the symbol file with the catch-all predicate ">= 0".

4.3.2.2 Field Descriptions

4.3.2.2.1 Mnem Field

The mnemonic field is a symbolic-only field that conveys the record type. The string displayed in this field can be customized to tabulate and show certain other fields in the record and their values. This behavior is configured using the "Mnem Cfg…" button on the Options tab of the disassembly control panel. Using this feature is simply another way to view fields in a record, using the mnemonic column to display the field rather than using a separate column in the listing.

4.3.2.2.2 Sample Field

The Sample field represents a record's ordinal number in the listing starting with 0 and incrementing by one for each possible logical byte position based on the maximum lane count setting. This means records associated with HS data records will have their sample field increment by 1. On the other hand, records associated with LP data records will have their sample field increment by the maximum lane count.

4.3.2.2.3 Disp Field

The Disp field is a column used for showing the mnemonic view state of packet headers. This field is blank for non-packet header records but contains one of three indicators for a record containing a packet header: "+", "++", or "-". These indicators correspond to a mnemonic view state of "closed", "mnem-fields", "mnem- and packet-fields" respectively. When clicked, the mnemonic view state of the packet cycles to the next state (see section 4.3.2.5).

4.3.2.2.4 Time Field

The Time field represents displays a record's time-stamp, which the user may choose to display relative to a number of time references (e.g. start of trace, cursor or trigger position, etc.) Please see section 4.3.2.1.8 for more information.

By default, the time-stamp is displayed with a radix of Symbolic (Default), which displays the time as a string, e.g. "1.300 ns", but alternatively it may be displayed as a numeric value (hex or decimal), which converts time to an integer number of picoseconds. This allows the time field to be used in search and filter criteria, for example, in restricting record display or search to a limited time range.[9]

4.3.2.2.5 Param Fields

In the MIPI CSI-2 and DSI, many packet types are associated with one, two or three named parameter fields (as opposed to larger blocks of generic data such as pixel data, blanking bytes, LUT tables, etc.). The disassembly window organizes these packet parameters into the fields: "Param1", "Param2" and "Param3".

For example, the Generic Short Write packet can have 0, 1, or 2 parameters. For zero-parameter writes, all Param fields will have dashes displayed in them. For one-parameter writes, "Param1" will display the value of the single-byte parameter. And for two-parameter writes, "Param1" and "Param2" will display the single-byte parameters.

Another example is the DCS Set Column Address command that has two 16-bit parameters. In this case, "Param1" and "Param2" will display the 16-bit StartColumn and EndColumn parameters respectively.

Finally, note that when the Param fields are added as mnemonic fields using the "Mnem Cfg…" button, the actual parameter names are displayed in the mnemonic. For this reason, in general, the user may always want to have the Param fields selected as fields to display in the mnemonic.

4.3.2.2.6 LPSeq Field

The LPSeq field indicates the LP sequence state of the DPhy Decoder at the time the record was recorded. The LP sequence state represents a position in decoding DPhy LP signaling, for example, the state LP-320 records having seen the sequence LP11, LP10, LP00 (the first three signaling states of a BTA). The following LP sequence states are used:[10]

State	**LP Seq**
Stop	LP11
LP-Rqst	LP11, LP10
LP-320	LP11, LP10, LP00
LP-3202	LP11, LP10, LP00, LP10
BTA	LP11, LP10, LP00, LP10, LP00
BTA-Exit	LP11, LP10, LP00, LP10, LP00, LP10
LP-3201	LP11, LP10, LP00, LP01
LP-Esc	LP11, LP10, LP00, LP01, LP00
HS-Rqst	LP11, LP01
HS-Go	LP11, LP01, LP00

[9] Currently, searching or filtering on the time-stamp field when in symbolic mode does not behave as desired, as comparisons are made alpha-numerically in this radix.

[10] Note that the DPhy Decoder does not distinguish between the LP00 state called HSPrepare and the HS-0 state called HSZero. The sequence state for this period is called HS-Go.

SOT	LP11, LP01, LP00, HS Sync Byte
HS	In HS burst
EOT	EOT (inverted bit-sense) at the end of HS burst

4.3.2.2.7 PktLen Field
The PktLen field applies to records that contain the first header byte of a packet and indicates the total number of bytes associated with the packet (including header and CRC fields).

4.3.2.2.8 VC Field
The VC field applies to records that contain the first header byte of a packet and displays the 2-bit virtual channel field of the packet.

4.3.2.2.9 DataType Field
The DataType field applies to records that contain the first header byte of a packet and displays the 6-bit data type field of the packet.

4.3.2.2.10 DCSCmd Field
The DCSCmd field applies to records that contain the first header byte of a DCS read request short write, or long write and displays the 8-bit DCSCmd field.

4.3.2.2.11 LP Fields
There are five LP fields: “LP0”, “LP1”, LP2”, “LP3” and “LPClk”. These fields are present in all record types and reflect the LP lane state of lanes 0-3 and Clk respectively. Note that unused lanes (i.e. those exceeding than the maximum lane count setting - 1) always show the LP11 state, even if the lane input is disconnected. A default SymDef mapping is provided to map field values of 0-3 to LP00, LP01, LP10, and LP11 respectively. .

4.3.2.2.12 Data Fields
There are four Data fields: “Data0”, “Data1”, “Data2”, “Data3”. These fields show the HS or LP data byte value per lane associated with the record. HS records will display bytes in the Data fields associated with active lanes. They will also show data bytes associated with HSZero and HSTrail states (fractional bytes are not displayed). On the other hand, only Data0 may show LP data bytes, including LPDT packet bytes, as well as escape and trigger byte codes associated with the escape signaling protocol.

4.3.2.2.13 Trig Field
The Trig field applies to all record types and is a Boolean value (0 or 1) that represents the result of trigger criteria evaluation for the record. There is no limit to how many records may have their Trig field set. Moreover, non-zero prefill settings may mean the record associated with the hardware trigger event is not the first true value in the trace. If it is important, use the Cursor Panel or context menu to go to the record associated with the hardware trigger. A default SymDef mapping is provided to map field values of 0 to “---“ and 1 to “Trig”.

4.3.2.2.14 Frame Field
The Frame field applies to records that contain the first header byte of all video-related packet types, including CSI Frame Start, Frame End, Line Start, Line End; DSI VSync Start and VSync End; blanking and video packets. Starting at one, it reflects an ordinal frame number associated with the packet. It also corresponds with frames listed in the

video tab of the Disassembly Control panel. Generally, the frame number will increment with each Frame Start or VSync Start packet, but might otherwise increment if a video packet occurs with a new VC or DataType.

4.3.2.2.15 ActLine Field

The ActLine field applies to records that contain the first header byte of an active video packet, *where a preceding Frame Start or VSync Start packet has already been seen* (otherwise, the line number is unknown). Starting at one, it reflects the active line number of the packet in the frame.

4.3.2.2.16 Lane Field

The Lane field applies to all records and indicates the corresponding data lane of its data byte. For LP records, Lane field is always zero. For HS records, the Lane field can be 0 through the maximum lane count minus 1.

4.3.2.2.17 ECC Fields

There are three ECC fields: “ECC”, “ExpECC”, and “ECCOk”, displaying the ECC byte contained in a packet header, the expected (computed) ECC bytes associated with the header, and a comparison result of the two respectively. These fields apply to records that contain the first header byte of a packet. ECCOk has a default SymDef mapping for the boolean comparison of 0 == “Err” and 1 == “Ok”.

4.3.2.2.18 BusOwn Field

The BusOwn field applies to all DSI records and indicates which end of the bidirectional link is transmitting (i.e. owns the bus). A default SymDef mapping for the boolean value is 0 == “Dev” and 1 == “Host”.

4.3.2.2.19 Raw Fields

There are three fields that represent raw data records, status and timestamps stored in DPhy Decoder memory during capture. They are “RawData”, “RawStatus” and “RawTime”. The formats of these fields are currently undocumented and are not generally useful to the user as their content is reflected in other displayed fields in the disassembly.

4.3.2.3 Trigger and Cursors

The hardware trigger record and two cursor records can be displayed in the listing grid, changing the background shading of their associated records. The trigger record indicates the location of the hardware trigger for capture and is a static fixed position that cannot be changed but can be used as a reference for time measurement and Goto operations. Its Trig field will be set to 1, though it may not be the only record with Trig == 1, nor is it necessarily the first occurrence of a record with Trig == 1, depending on the Prefill percentage setting.

Cursors are enabled by checking the associated checkbox in the cursor panel to the right of the listing grid and can also be controlled from the cursor panel control (see Cursor Panel section for more details). The current cursor positions and the delta time between them are displayed above the listing grid, using the current global time reference set in the time field radix context menu.

Cursor related functions are also available in the column context menu available by right-clicking in any grid cell (except the radix sub-header row). In particular, the user can select "Goto Cursor1" or "Goto Cursor2" to scroll the listing grid to the record location of the indicated cursor. Also, if the context menu is brought up from a non-header cell, the "Move Cursor1 Here" and "Move Cursor2 Here" options are also available.

4.3.2.4 Views

A main feature of the disassembly window is to allow the user to set criteria for filtering records, only displaying a subset of acquired records from the disassembly view. This is done through several mechanisms: selecting a view setting, adjusting the global mnemonic view or the mnemonic view of individual packets, and setting packet and record filter criteria. Each mechanism successively restricts records shown in the disassembly.

There are two global view settings that control disassembly, set via the View combo-box in the Filter tab of the Disassembly Control Panel.

All	Shows all record types, including non-packet traffic such as LP state, LP Escape and HS burst signaling.
Packet	Shows only records that contain packet bytes, including HS, LPDT, and escape-mode commands.

When a view is selected, the listing is updated to display records only applicable to the new view. In the case of packet records, the mnemonic view state determines whether only the first header byte is shown or all packet bytes are shown (see next section).

4.3.2.5 Mnemonic View State

Mnemonic view state is an attribute of records that contain the first byte of a protocol packet. Its setting determines whether only the first byte of a packet is displayed, or all bytes of the packet are displayed, and whether the user-specified mnemonic fields associated with the packet are displayed.

There are three mnemonic view states:

Closed	Only the first header byte of the packet is displayed.
Mnem-Fields	Only the first header byte of the packet plus its mnemonic fields are displayed. Mnemonic fields are fields selected by the user to be displayed with the mnemonic via "Mnem Cfg…" button).
Mnem-and Packet-Fields	All packet bytes are displayed. Mnemonic fields are also shown for the first header byte of a packet.

The mnemonic view state of a packet is shown in the Disp field in the record using the indicators "+", "++", or "-" (corresponding to the states in order in the table above). To cycle an individual packet's mnemonic view state, click on the indicator in the Disp column. To cycle all packet mnemonic view states at once, click the "Cycle Mnem" button in the Search tab of the Disassembly Control Panel.

4.3.2.6 Column Context Menu

The column context menu is brought up by right-clicking in any cell except for the radix row (second row) of the listing grid. Depending on the cell location clicked to bring up the menu, certain options will be enabled or disabled. Here is a summary of all the functions in the column context menu:

Function	Description	Shortcut
Add Column(s)	Brings up dialog to add columns (i.e. make fields visible) to the listing grid.	
Delete Column(s)	Deletes selected columns.	
Restore Selected Column Width(s)	Restores selected columns to their default width.	
Goto Cursor1	Scrolls the listing grid to the cursor1 record.	Ctl-1
Goto Cursor2	Scrolls the listing grid to the cursor2 record.	Ctl-2
Goto Trigger	Scrolls the listing grid to the hardware trigger record.	Ctl-T
Goto…	Brings up a dialog to enter a sample number to scroll the listing grid to (the nearest visible record is used).	Ctl-G
Move Cursor1 Here	Moves the cursor1 position to the current record.	
Move Cursor2 Here	Moves the cursro2 position to the current record.	
Search Frame Start	Fills in the search criteria to search for Frame Start for CSI or VSync Start for DSI. Then, the Next Search function is performed. .	Ctl-Alt-F
Search Same Mnem	Fills in the search criteria to search for records that match the mnemonic value associated with the current record. Then, the Next Search function is performed. Note that the current cell can be any field type.	Ctl-Alt-M
Search Same Val	Fills in the search criteria to search for records that match the current cell's field value. Then, the Next Search function is performed.	Ctl-Alt-V
Search Same Mnem And Val	Fills in the search criteria to search for records that match the mnemonic value associated with the current record and the current cell's field value. Then, the Next Search function is performed.	Ctl-Alt-B
Next Search	Scrolls to the next record satisfying the search criteria	Ctl-S
Next Packet	Scrolls to the next packet start.	Ctl-P

Next Frame Start	Scrolls to the next CSI Frame Start or DSI VSync Start packet.	Ctl-F
Next Same Mnem	Scrolls to the next record that matches the mnemonic value associated with the current record.	Ctl-M
Next Same Val	Scrolls to the next record that matches the current cell's field value.	Ctl-V
Next Same Mnem And Val	Scrolls to the next record that matches the mnemonic value associated witht the current record and the current cell's field value.	Ctl-B
Prev Search	Scrolls to the previous record satisfying the search criteria	Alt-S
Prev Packet	Scrolls to the previous packet start.	Alt-P
Prev Frame Start	Scrolls to the previous CSI Frame Start or DSI VSync Start packet.	Alt-F
Prev SameMnem	Scrolls to the previous record that matches the mnemonic value associated with the current record.	Alt-M
Prev Same Val	Scrolls to the previous record that matches the current cell's field value.	Alt-V
Prev Same Mnem And Val	Scrolls to the previous record that matches the mnemonic value associated witht the current record and the current cell's field value.	Alt-B
Exclude All But Frame Start	Sets and applies the filter criteria to filter all records except for CSI Frame Start of DSI VSync Start packets.	Ctl-X, F
Exclude All But Same Mnem	Sets and applies the filter criteria to filter all records except those that match the mnemonic value associated with the current record.	Ctl-X, M
Exclude All But Same Val	Sets and applies the filter criteria to filter all records except those that match the current cell's field value.	Ctl-X, V
Exclude All But Same Mnem and Val	Sets and applies the filter criteria to filter all records except those that match the mnemonic value associated with the current record and the current cell's field value.	Ctl-X, B
Modify Filter From Current Cell	Contains a sub-menu with three items: ▪ Replace Filter Term ▪ Add OR Filter Term ▪ Add AND Filter Term Selecting one of these menu items will perform the requested function based on the current cell's field name and value. The operator used is "==".	
Modify Search From Current Cell	Contains a sub-menu with three items: ▪ Replace Search Term ▪ Add OR SearchTerm	

	▪ Add AND Search Term Selecting one of these menu items will perform the requested function based on the current cell's field name and value. The operator used is "==".	
Close Mnem View	Sets the mnemonic view of the current packet to "Closed". This option is only enabled when the current mnemonic view is "Mnem- And Packet-Fields".	

4.3.2.7 Radix Context Menu Summary

The radix context menu is brought up by right-clicking in the sub-header row (second row) of the listing grid and is used to set the display format of the current column. The functions in the menu have been described in earlier sections. Here is a summary of all the functions in the radix context menu (except for the context menu for the Time field):

- **Hexadecimal –** sets the radix to hexadecimal (enabled for numeric fields)
- **Decimal –** sets the radix to decimal (enabled for numeric fields)
- **Binary –** sets the radix to binary (enabled for numeric fields)
- **Symbolic (Default) –** sets the radix to symbolic using the default translation internal to the application.
- **Symbolic (File) –** sets the radix to symbolic using the translation defined by the symbolic file specified by the user using the "<New Symbolic File>" menu option.
- **<New Symbolic File> –** brings up an open file dialog to select a symbol file to associate with the current column.

In addition, the time field radix context menu has several more options to set the desired time field reference, which determines how the time field is displayed. One of these options is always checked:

- **Relative to Start –** sets the global time field reference to the first acquired record
- **Relative to Previous –** sets the time field reference for each record to the previous displayed record (i.e. displays delta time between records)
- **Relative to Trigger –** sets the global time field reference to the trigger record
- **Relative to Cursor1 –** sets the global time field reference to the cursor1 record
- **Relative to Cursor2 –** sets the global time field reference to the cursor2 record

4.3.3 Cursor Control Panel

The cursor panel is a custom control with three narrow columns on the right side of the listing grid. Each narrow column is similar to a vertical scroll bar, each associated with either a cursor or the trigger (for the purposes of this description, the trigger can be thought of as a special immovable cursor). Cursors represent a specific record position in the listing, similar to a scroll bar, where the top of the panel represents the start of the listing and the bottom of the panel represents the end of the listing.

The first column in the cursor panel represents the trigger and, if enabled, is drawn as a red square labeled with a 'T'. The trigger cursor is always enabled. The second and third columns represent cursor1 and cursor2 respectively and, if enabled, are drawn as a blue square labeled with a '1' or a blue square labeled with a '2'. A check box above each cursor column can be used to enable or disable each cursor. Disabled cursors are drawn with a gray background in the cursor panel and are not visible in the disassembly listing.

Like a scroll-bar, the cursor panel is meant to represent the entire vertical span of the listing windows. Cursor positions within the panel symbolically indicate where each lies within the listing, i.e. a cursor toward the top of the panel indicates a record position near the start of the listing and a cursor toward the bottom of the panel indicates a record position near the end of the listing. The times associated with cursor1 and cursor2 positions and the difference between them are displayed in readouts above the listing grid.

In addition to the cursor symbols, a horizontal white region is drawn across the cursor panel. This region represents the currently viewed region of the listing, having a position and thickness corresponding to the current viewed region relative to the total listing.

The cursor panel supports the following operations for setting cursor positions and scrolling the listing:

- Left-clicking and dragging a cursor square moves the cursor position. While dragging, the current cursor position (Sample #) is displayed in a pop-up box.
- If the Shift key is pressed and held while dragging, the listing will scroll with the cursor position, otherwise the listing will not scroll.
- If the Control key is pressed and held while dragging, the cursor will scroll more slowly, allowing for finer control.
- Double clicking a cursor in the panel scrolls the listing window to the cursor position.
- Double-clicking in the cursor panel (not on a cursor) scrolls the listing window to the position clicked.

4.3.4 Diassembly Control Panel

The Diassembly Control Panel is located at the bottom part of the disassembly window and has five tabs: Filter, Search, Options, Video, <Minimize>. The functions and controls provided by each tab are described in the following sections.

4.3.4.1 Filter Tab

The filter tab provides controls to filter or show records based on user criteria. Specifically, most controls in this tab help to define and apply a filter equation that is evaluated against each record to determine whether or not the record is displayed. Note that the application of the filter result is against the visible records already defined by the current global view and packet mnemonic views.

Some controls such as the Filter, Show, and Disable radio buttons and the View combo box update the disassembly listing as soon as they are used. On the other hand, changes in the Mnemonic Type list box and the Filter Term grid do not take effect until the Apply button is clicked.

Here is an overview of the controls contained in the filter tab:

- **View Combo Box** – selects the current disassembly view. Two views are provided: "All" or "Packet" (see section 4.3.2.4 for more details).
- **Mnemonic Type List Box** – contains a list of all mnemonic types defined for the current view. The user checks packet-types and record-types to use as criteria for the overall filter equation.
- **Filter Term Grid –** defines predicate terms of the filter equation based on packet fields.
- **Set All Button** – checks all mnemonic types in the list box.
- **Clear All Button –** clears all selected mnemonic types in the list box.
- **Filter Radio Button –** enables record filtering and uses the filter equation to determine which records should not be displayed (i.e. if the filter equation evaluates true for a record, it is not displayed).
- **Show Radio Button** – enables record filtering and uses the filter equation to determine which records should be displayed (i.e. if the filter equation evaluates true for a record, it is displayed).
- **Disable Radio Button –** disables record filtering.
- **Clear Button –** clears all terms in the filter term grid.
- **Apply Button** – Applies any changes to the filter criteria, updating the disassembly listing.

The Mnemonic Type list box and the Filter Term grid together define the overall filter equation. The list box defines a set of mnemonic types that qualify records for further evaluation by the filter terms. That is, records that satisfy the filter equation must have at least one lane with a mnemonic type than is checked in the list box. So, for example, at the extremes, if no types are checked, the filter equation will evaluate false for all records. And if all types are checked, the filter equation for each record will evaluate solely to the result from the Filter Term grid.

The Filter Term grid defines zero or more Boolean terms (one per row) which compare a record field to a constant value. Terms are logically connected via AND or OR operations to build a Sum-Of-Products equation used in the overall filter equation. The AND operation takes precedence over the OR operation so A AND B OR C AND D is evaluated as (A AND B) OR (C AND D).

The filter grid defines four columns as follows:

- **Op** – defines how the term is logically connected to its predecessor, either via AND or OR (the first term, which has no predecessor is fixed to show IF). In addition to showing the logical operation associated with the term, this field is a combo box that provides options for adding or deleting terms. For AND terms, two options are available: add another AND term to the product or delete the current AND term. For IF or OR terms, two additional options are available: add another OR term to the equation or delete the current OR term (which includes all AND terms of the product).

- **Field** – combo box to select a record field name
- **Comp** – combo box to select a comparison operator (==, !=, <, <=, >, >=)
- **Value** – text box to enter a comparison value

To define a filter equation, begin with filling in the Field, Comp, and Value fields of the first term in the Filter Term grid. Next, select the Op drop-down and select "<Insert OR term>" or "<Insert AND term>" to add a new term to the equation. Fill in the fields of that term and continue inserting terms until the equation is fully defined.

Another method is to right-click in the disassembly grid and select one of the menu options that begin with "Exclude All But…". Alternatively, each of these menu options has a keyboard shortcut sequence that begins with Ctl-X. For example, if you right-click on a cell in the PktLen field that has a value of 4 and select "Exclude All But Same Val", the filter list box will check all packet types and enter the term"IF PktLen == 00004" in the filter grid. Then, the Show radio button will be selected and then the disassembly will update.

Figure 4 shows the results of this operation for a video capture. In particular, filtering has eliminated all packets except for VSync Start and HSync Start. Note that many rows have a double-line dividing them. This is an optional highlighting method to indicate there are packets in the view that have been filtered. This highlight is enabled by checking "Highlight Filter Gaps" in the Options tab.

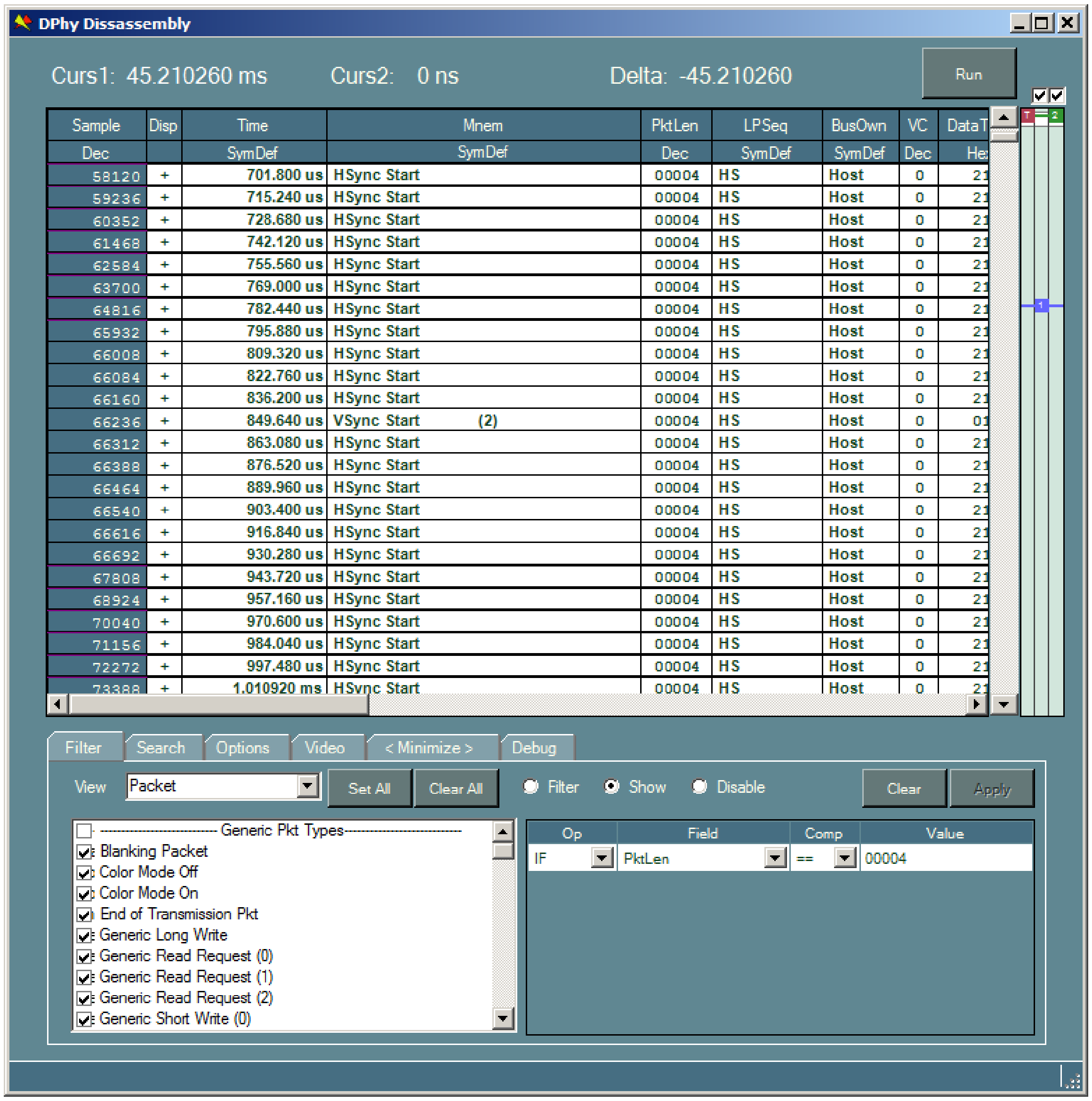
Numeric field values are assumed to be in decimal unless appended with an 'h' for a hexadecimal value or a 'b' for a binary value. When using hex or binary values, an 'x' may be used as a "don't care" placeholder. Thus, the value "1xh" will match values from 10h to 1fh.

Symbolic field comparison symbolic is supported, but the user should understand that values are compared using string-comparison and so results may not be as intended. For example, if the user constrains the filter equation to select a time range when the Time field is set to symbolic, he might set up the following equation:

IF Time > 100 us AND Time < 200 us

However, the result of the string comparisons against Time values such as "120.888 ns" and "190.666666666 ms" will not be as desired. In general, the "==" and "!=" comparison operations are most predicatble, though be aware that comparisons are case-sensitive.

Note that changing the radix of a field from numeric to symbolic or symbolic to numeric *after defining filter or search terms* can affect the term's validity. Comparisons are performed on field values in the radix they are displayed (not the radix at the time the predicates were built). An error will occur if filtering or searching using a non-numeric predicate value on a field that is currently being displayed as a numeric value (though hex, decimal, and binary conversions are automatic).



Curs1: 45.210260 ms   Curs2: 0 ns   Delta: -45.210260

Sample	Disp	Time	Mnem	PktLen	LPSeq	BusOwn	VC	DataT
Dec		SymDef	SymDef	Dec	SymDef	SymDef	Dec	He
58120	+	701.800 us	HSync Start	00004	HS	Host	0	21
59236	+	715.240 us	HSync Start	00004	HS	Host	0	21
60352	+	728.680 us	HSync Start	00004	HS	Host	0	21
61468	+	742.120 us	HSync Start	00004	HS	Host	0	21
62584	+	755.560 us	HSync Start	00004	HS	Host	0	21
63700	+	769.000 us	HSync Start	00004	HS	Host	0	21
64816	+	782.440 us	HSync Start	00004	HS	Host	0	21
65932	+	795.880 us	HSync Start	00004	HS	Host	0	21
66008	+	809.320 us	HSync Start	00004	HS	Host	0	21
66084	+	822.760 us	HSync Start	00004	HS	Host	0	21
66160	+	836.200 us	HSync Start	00004	HS	Host	0	21
66236	+	849.640 us	VSync Start (2)	00004	HS	Host	0	01
66312	+	863.080 us	HSync Start	00004	HS	Host	0	21
66388	+	876.520 us	HSync Start	00004	HS	Host	0	21
66464	+	889.960 us	HSync Start	00004	HS	Host	0	21
66540	+	903.400 us	HSync Start	00004	HS	Host	0	21
66616	+	916.840 us	HSync Start	00004	HS	Host	0	21
66692	+	930.280 us	HSync Start	00004	HS	Host	0	21
67808	+	943.720 us	HSync Start	00004	HS	Host	0	21
68924	+	957.160 us	HSync Start	00004	HS	Host	0	21
70040	+	970.600 us	HSync Start	00004	HS	Host	0	21
71156	+	984.040 us	HSync Start	00004	HS	Host	0	21
72272	+	997.480 us	HSync Start	00004	HS	Host	0	21
73388	+	1.010920 ms	HSync Start	00004	HS	Host	0	21

View: Packet   ○ Filter [● Show] ○ Disable

- [ ] ------------------------ Generic Pkt Types ------------------------
- [x] Blanking Packet
- [x] Color Mode Off
- [x] Color Mode On
- [x] End of Transmission Pkt
- [x] Generic Long Write
- [x] Generic Read Request (0)
- [x] Generic Read Request (1)
- [x] Generic Read Request (2)
- [x] Generic Short Write (0)

Op	Field	Comp	Value
IF	PktLen	==	00004

Figure 4 – Filter Example

4.3.4.2 Search Tab

The search tab contains controls for searching and optionally highlighting records that satisfy given criteria. Statistics about filtered and matched search records are also displayed. Here is a description of the controls contained in the filter tab:

- **Highlight Selected –** when checked, records satisfying the search criteria are highlighted in the listing.
- **Search Cnt Readout** – displays the number of records and percentage of all acquired records that satisfy the search criteria. Depending on the size of the trace, the current view, and the complexity of the filter and search equations, performing a count of all search records in the trace can be time consuming. An initial attempt to update the search count is made during disassembly update, but with a ½ second timeout. If the timeout occurs, the Search Cnt is displayed as "<update>", indicating the user needs to click on the Update Srch Cnt button if to recomputed.
- **Visible Cnt Readout –** displays the number of records and percentage of all acquired records) that are currently visible in the listing.
- **Hidden Cnt Readout –** displays the number of records and percentage of all acquired records that are currently hidden from view in the listing.
- **Total Cnt Readout–** displays the number of acquired records.
- **Cycle Mnem Button** – cycles the mnemonic view of all disassembly packets to the next view state.
- **Update Srch Cnt Button** – updates the Search Cnt readout if the initial attempt during disassembly update timed out (see Search Cnt Readout description).
- **Next Pkt Button** – scrolls to the next packet in the disassembly listing. Alternate ways of performing this function is to select "Next Packet" from the disassembly context menu or pressing Ctl-P.
- **Prev Pkt Button** – scrolls to the previous packet in the disassembly listing. Alternate ways of performing this function is to select "Prev Packet" from the disassembly context menu or pressing Alt-P.
- **Next Search Button**– scrolls the listing to the next record matching the search criteria. Alternate ways of performing this function is to select "Next Search" from the disassembly context menu or pressing Ctl-S.
- **Prev Search Button** – scrolls the listing to the previous record matching the search criteria. Alternate ways of invoking this function is the select "Prev Search" from the disassembly context menu or pressing Alt-S.
- **Search Term Grid** – allows the user to define predicate terms of the search equation. This grid operates identically to the Filter Term grid described in the previous section.
- **Clear Button –** clears and filter term grid.
- **Apply Button –** Applies changes to the search term grid, updating the disassembly listing and readouts.

To search for and highlight records that satisfy a particular predicate equation, search criteria can be entered in to the search term grid identically to the filter term grid. The search equation is built from terms defined and applied to each record in the listing when the Apply button is clicked. If the Highlight Selected checkbox is checked, records that satisfy the search equation are highlighted in the listing (note that the search equation is only evaluated against unfiltered records). The "Next Search" and "Prev Search" buttons can be used to scroll the listing to the next and previous qualifying records respectively.

4.3.4.3 Options Tab

The Options tab contains controls for configuring acquisition, decoding, and disassembly display.

4.3.4.3.1 Acquisition Options

There are three user-controls for acquisition. They are described below:

- **Prefill Pct –** this up/down control sets the trigger position relative to the acquisition. The value is an integer percentage of the acquisition memory buffer size.
- **Memory Depth –** this combo-box provides options for setting the size of the acquisition buffer.
- **Acquire Timestamps –** this check box determines whether timestamps will be recorded with each acquired data record.

Setting the Prefill Pct value to 0 will cause the trigger record to appear at the beginning of the trace. Set to 50 to cause the trigger record to appear in the middle of the trace. And so on.

The largest memory buffer size is currently 128 MB. The smaller the size, the faster acquisition, upload, decoding, and display will be. In addition, filter and search functions are also faster with smaller traces.

Currently, a data record representing one LP or HS data byte (per lane) requires 12 bytes of acquisition memory is timestamps are used. Otherwise, it requires 8 bytes of acquisition memory. Thus, the maximum number of data bytes that can be captured and displayed without timestamps is 42.6 MB (assuming a 4-lane system). With timestamps, the maximum number of data bytes is 64 MB.

4.3.4.3.2 Disassembly Options

Disassembly controls are provided for configuration decoding and display. More general options, unrelated to protocol are:

- **Grid Font Size –** this up/down control increases or decreases the font size of the grid display. Values can range from 6 to 12.
- **Highlight Filter Gaps –** this check box sets whether or not filtered records cause a double-thick horizontal grid-line to be displayed between visible rows.
- **Disable Disassembly –** this check box disables disassembly after acquisition and upload. It is generally only useful for debugging, allowing a trace that causes a catastrophic error in decoding to be saved and sent to the Moving Pixel Company for analysis.
- **Reanalyze –** this button restarts decoding and disassembly on the current trace data. This may be needed if initial decoding was aborted before decoding was complete (in which case, only a partial disassembly is displayed). Click on the Reanalyze button to restart and complete a full decoding.

The remaining controls in the Options window are protocol-related options. They are:

- **Mnem Cfg…** (CSI & DSI) – this button bring up the Mnemonic Configuration dialog, allowing the user to select fields to display with the mnemonic in the listing.
- **Interpret WriteMem As Video Data (DSI)** – this check box indicates that WriteMemoryStart and WriteMemoryContinue commands should be interpreted as containing video data. WriteMemoryStart packets always start a new frame. The data format and line length are set by "WriteMem Pixel Fmt" and "WriteMem Pixels Per Line" respectively. Note that if this option is checked and both WriteMemory and Packed Pixel packets are present in the trace, only WriteMemory frame data will be catalogued in the Video tab.
- **Show Video Payload As Pixels (CSI & DSI) –** this check box configures how payload data for video packets is displayed in the disassembly. Without this option checked, each payload byte is displayed one-byte-per-record with no pixel decoding (just as all other non-video packets). However, with this option checked, pixels are decoded and presented, one pixel per record, with actual color components values displayed.[11]
- **RAW Decode Fmt (CSI) –** this combo box indicates the format to use for decoding RAW video packets.

Bayer GRBG	Decodes lines 1,3,5… as alternating green, red pixels and lines 2,4,6… as alternating blue, green pixels.
Bayer RGGB	Decodes lines 1,3,5… as alternating red, green pixels and lines 2,4,6… as alternating green, blue pixels.
Bayer BGGR	Decodes lines 1,3,5… as alternating blue, green pixels and lines 2,4,6… as alternating green, red pixels.
Bayer GBRG	Decodes lines 1,3,5… as alternating green, blue pixels and lines 2,4,6… as alternating red, green pixels.
Monochrome	Decodes raw pixels as luminence values.

- **WriteMem Pixel Fmt (DSI) –** this combo box indicates the pixel format to use when decoding WriteMemory commands as video data. Options are:

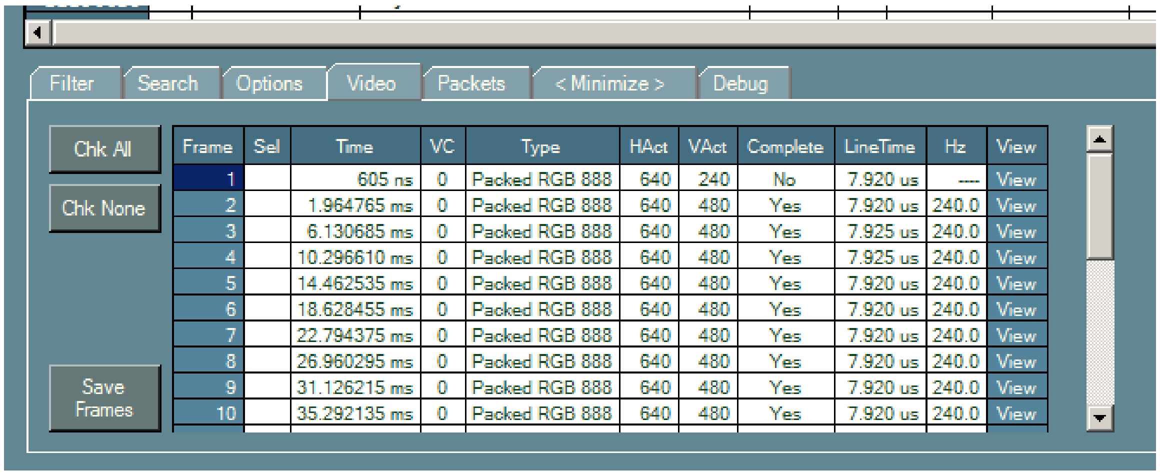
Packed RGB 565	Packed RGB 121212
Packed RGB 666	Loose 20-bit YCbCr 422
Loose RGB 666	Packed 16-bit YCbCr 422
Packed RGB 888	Packed 24-bit YCbCr 422
Packed RGB 101010	

- **WriteMem Pixels Per Line (DSI) –** this text box indicates the number of pixels per line when decoding WriteMemory commands as video data.

[11] This discussion assumes that the mnemonic view of the video packet is "Mnem and Packet Fields". Otherwise, payload bytes are not visible.

4.3.4.4 Video Tab

The Video tab of the Disassembly Control panel contains information about video frames decoded from the trace (see Figure 5)



Frame	Sel	Time	VC	Type	HAct	VAct	Complete	LineTime	Hz	View
1		605 ns	0	Packed RGB 888	640	240	No	7.920 us	----	View
2		1.964765 ms	0	Packed RGB 888	640	480	Yes	7.920 us	240.0	View
3		6.130685 ms	0	Packed RGB 888	640	480	Yes	7.925 us	240.0	View
4		10.296610 ms	0	Packed RGB 888	640	480	Yes	7.925 us	240.0	View
5		14.462535 ms	0	Packed RGB 888	640	480	Yes	7.920 us	240.0	View
6		18.628455 ms	0	Packed RGB 888	640	480	Yes	7.920 us	240.0	View
7		22.794375 ms	0	Packed RGB 888	640	480	Yes	7.920 us	240.0	View
8		26.960295 ms	0	Packed RGB 888	640	480	Yes	7.920 us	240.0	View
9		31.126215 ms	0	Packed RGB 888	640	480	Yes	7.920 us	240.0	View
10		35.292135 ms	0	Packed RGB 888	640	480	Yes	7.920 us	240.0	View

Figure 5 – Video Tab

In the video tab, a grid control displays one-line-per-decoded-frame.with the following fields:

Frame	A frame number assigned to the frame
Sel	Click-area to select the frame for saving. An 'X' is alternately displayed and cleared with each click.
Time	Time stamp of first frame packet (generally Frame Start or VSync Start packet)
VC	Virtual channel
Type	Video format
HAct	Number of active pixels in a packet. If variable packet lengths were found, indicates the longest packet found.
VAct	Number of active lines
Complete	'Yes' is displayed if Frame Start and Frame End (CSI) or VSync Start and a second VSync Start (starting the next frame for DSI) was seen.
LineTime	The longest time period between active packets (CSI) or HSync/VSync Start packets (DSI).
Hz	The Frequency between Frame Start (CSI) or VSync Start (DSI) and the next Frame Start or VSync Start.
View	Click-area labeled "View" to bring up a window displaying the video frame.

4.3.4.4.1 Scrolling Disassembly to Frame Start

To scroll disassembly to the location of the first packet in the video frame, click on the frame number in the video grid. The first packet of a video frame is usually the Frame Start or VSync Start packet, but if these packets are missing, they could be other video packet types.

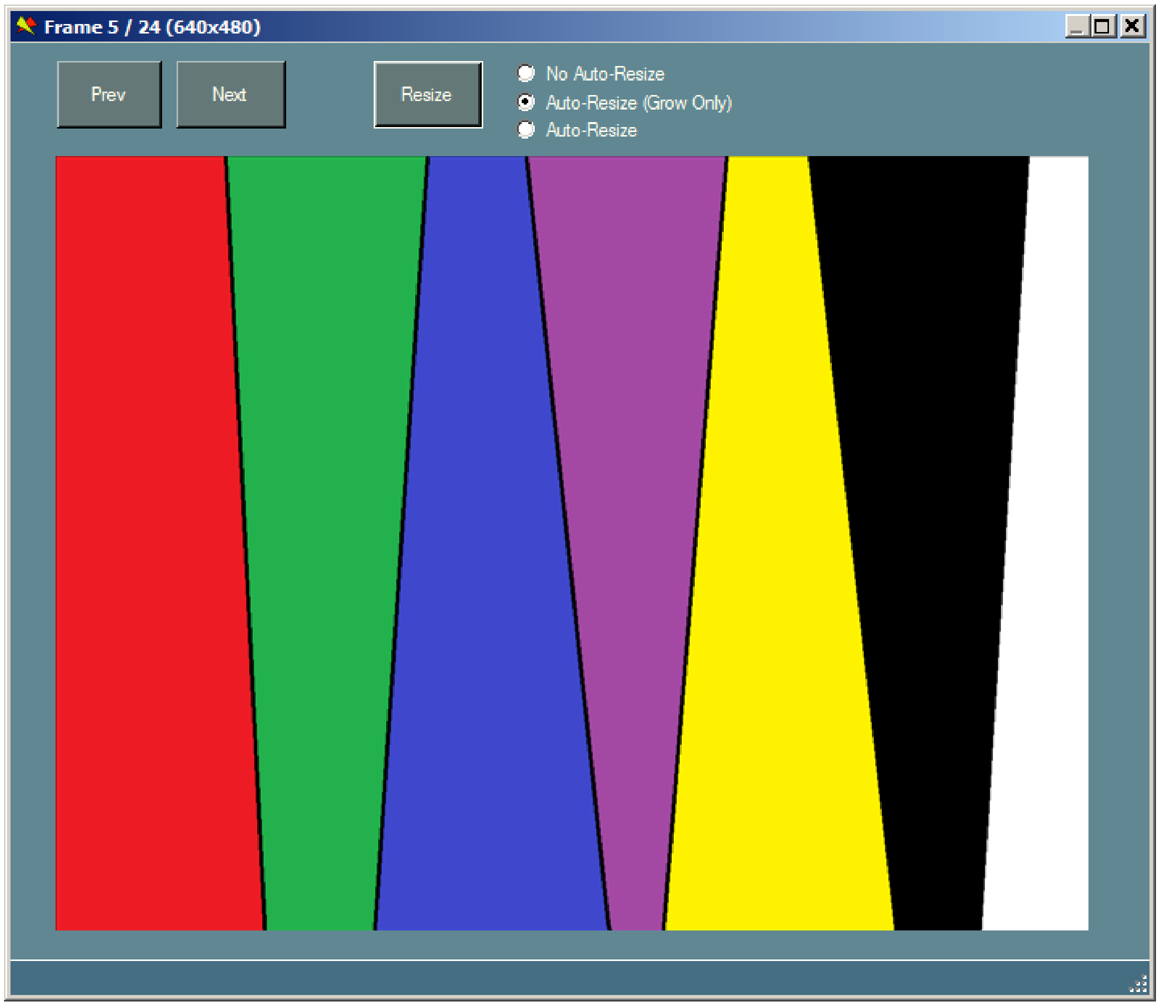


Figure 6 – Frame Display Dialog

4.3.4.4.2 Viewing Video Frames

To view a video frame, click on the "View" cell associated with the frame in the video grid (see Figure 6). This brings up the Frame Display dialog showing the captured frame. The Frame Display dialog has buttons labeled "Next" and "Prev" to step forward and backward in the frame sequence, allowing each frame to be viewed in turn. The title of the dialog indicates which frame number is being viewed and its dimensions.

The user can resize the dialog manually using the mouse by grabbing a dialog edge and dragging. If the image becomes too large for the dialog size, scroll bars appear to pan the image into view.

In addition, the dialog has a button labeled Resize and option buttons to select on of the following:

- No Auto-Resize
- Auto-Resize (Grow Only)
- Auto-Resize

These controls determine when the dialog is resized to fit the image. Note that it does not resize the image itself, just the dialog. Selecting “No Auto-Resize” means that the dialog is never automatically resized to fit the image. Whereas, “Auto-Resize (Grow Only)” causes the dialog to auto-resize if it is too small to display the image (but not too large) and “Auto-Resize” allows the dialog to always adjust to the image it displays.

4.3.4.4.3 Saving Video Frames

To save video frames to files, click on the cell labeled “View” for each frame you want to save, causing an ‘X’ to be displayed indicating that the frame is selected to be saved. Clicking on an ‘X’ unselects the frame. Alternatively, clicking on the “Chk All” button selects all frames and clicking on the “Chk None” button clears all frame selections.

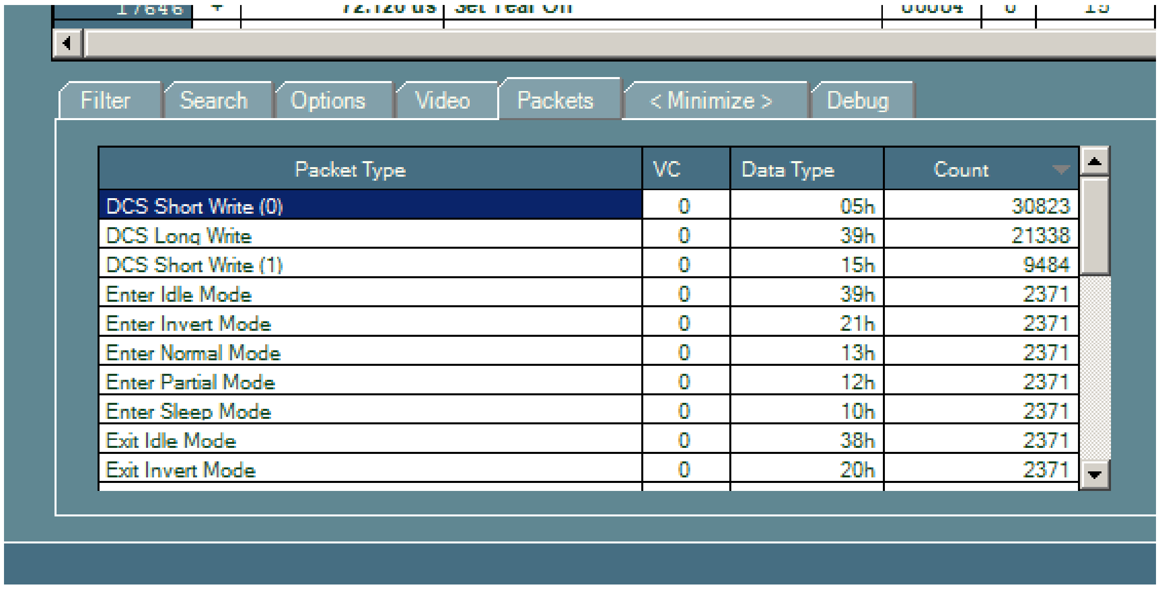
Once frames have been selected, clicking the “Save Frames” button brings up a save dialog to browse and enter a file name.

The extension used for the file name indicates the format to use for saving the frame data. If the extension is “BMP”, “JPG”, “GIF”, “.PNG”, “.TIFF” (case is irrelevant), the file is saved in the requested format. Otherwise, the file is saved in binary format, exactly as received from the MIPI bus.

In saving a single frame, the file name is used unchanged. In saving multiple frames, the file name, minus extension, should end in a number. This number will be incremented for each frame saved. For example, if the save file name is “Frame123.bmp” and three frames are selected, they will be saved as “Frame123.bmp”, “Frame124.bmp”, and “Frame125.bmp”.

4.3.4.5 Packets Tab

This tab, shows a summary of all packet types acquired in the trace (see Figure 7).



Packet Type	VC	Data Type	Count
DCS Short Write (0)	0	05h	30823
DCS Long Write	0	39h	21338
DCS Short Write (1)	0	15h	9484
Enter Idle Mode	0	39h	2371
Enter Invert Mode	0	21h	2371
Enter Normal Mode	0	13h	2371
Enter Partial Mode	0	12h	2371
Enter Sleep Mode	0	10h	2371
Exit Idle Mode	0	38h	2371
Exit Invert Mode	0	20h	2371

Figure 7 – Packets Tab

In this display the packet counts of each packet type, separated by Virtual Channel, are shown. Note that totals include packet types that may be hidden from display via filter criteria. Clicking on a column header alternately sorts the display in ascending or descending order based on the column content. As seen in the figure, DCS command categories are summarized as well as individual DCS packet types.

4.3.4.6 <Minimize> Tab

The <Minimze> tab is used to collapse the Disassembly Control panel, so that only the tab labels remain. This allows the disassembly grid can be expanded using up almost all the area of the disassembly window. The Disassembly Control panel re-expands when any other tab is clicked.

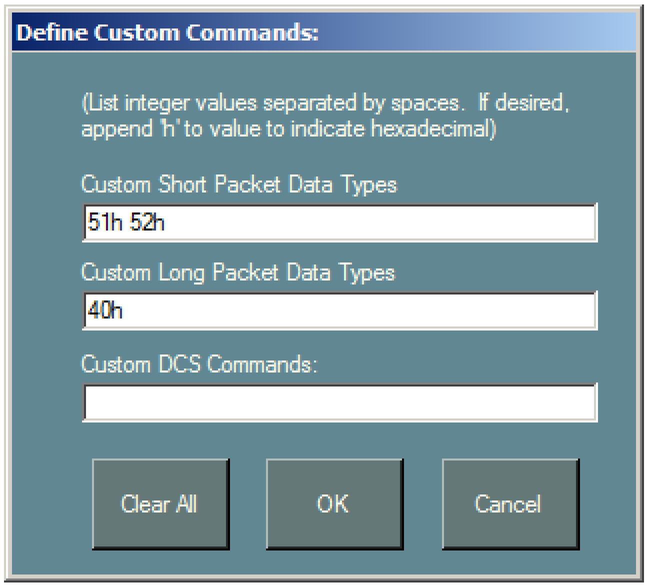
4.4 Defining Custom Commands

Custom commands are DataTypes or DCS command codes that are not defined in the relevant MIPI specification. If present in the DPhy bus traffic, custom DataTypes *for long packets* are important to enumerate to ensure the DPhyDecoder does not get confused during parsing. This is because the default behavior when the instrument encounters an unknown DataType is to assume it is a short packet. In addition, if the user wants to filter or trigger on fields particular to a custom packet type, these DataTypes and/or DCS command codes need to be enumerated as well.

To define custom commands, a dialog is provided when "Define Custom Commands…" is selected from the Config menu (see Figure 8). This dialog lets you specify custom short packet data types, long packet data types, and DCS commands. Simply enter lists of values (separated by spaces) in the appropriate text boxes and click OK. Note that DataTypes do not include the VC field and are masked to 6-bits. Values are parsed as decimal unless appended with an 'h', in which case they are parsed as hexadecimal (i.e. 20h = 32 decimal).

Clicking the Cancel button exits the dialog without registering any changes. Clicking the Clear All button clears all the text boxes.

Once custom DataType and DCS commands are been defined, options for selecting these packets will appear in the Custom Commands tree-view node for both triggering and filtering.


Define Custom Commands:

(List integer values separated by spaces. If desired, append 'h' to value to indicate hexadecimal)

Custom Short Packet Data Types: 51h 52h

Custom Long Packet Data Types: 40h

Custom DCS Commands:

Figure 8 – Custom Command Dialog

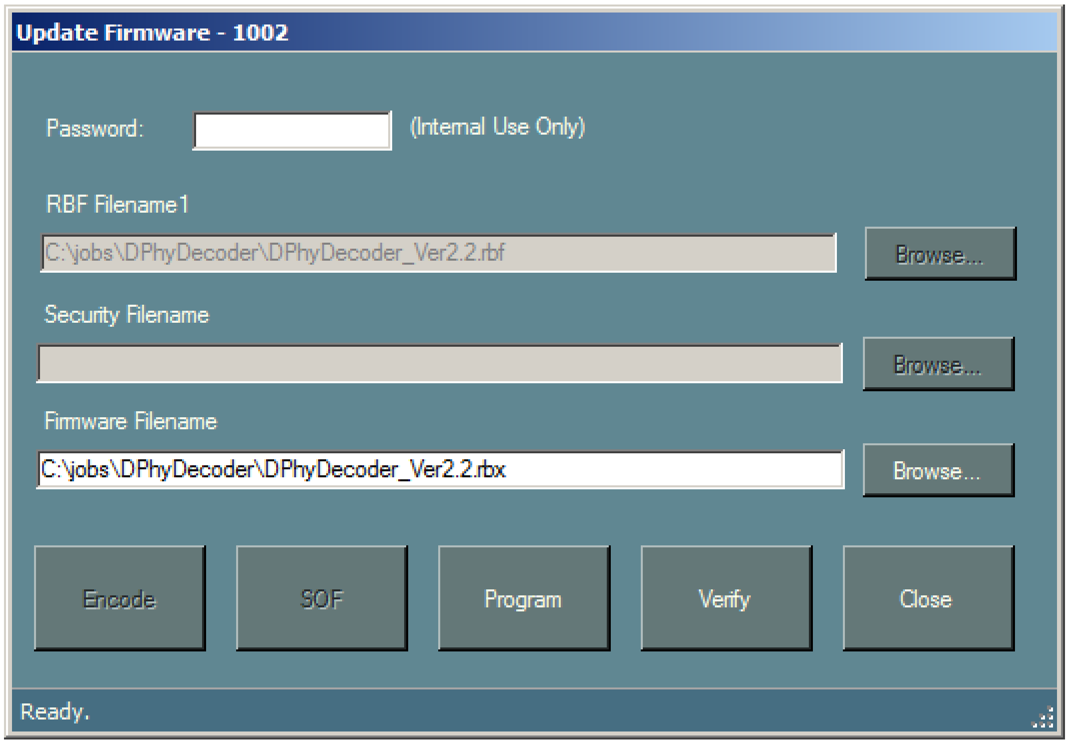
4.5 Firmware Update

Periodically, new releases of firmware may be available for the DPhyDecoder instrument. Firmware is easily updated through the use of the Firmware dialog, accessible by selecting “Update Firmware” from the Control menu (see Figure 9).

Many of the features of this dialog are for internal use only. For users, the only options are to program new firmware or to verify existing firmware. Simply browse for the firmware filename (extension .rbx) and click either Program or Verify respectively. Once the task has begun, the relevant button changes its lable to Stop, allowing the user to halt the task before completion. This is fine for verification but is not generally recommended when programming, as this will render the instrument inoperative until firmware has been fully programmed.

Either task takes about 5 minutes to complete, with status messages providing updates on progress. When programming is complete, shut down the application and power cycle the instrument to reload the new firmware. You should check that the first two LEDs on the instrument “twinkle” for a few seconds at power-on, indicating a successful boot.

Otherwise, try power cycling again. If the problem persists, you will need to reprogram firmware again.


Update Firmware - 1002

Password: (Internal Use Only)

RBF Filename1: C:\jobs\DPhyDecoder\DPhyDecoder_Ver2.2.rbf

Security Filename:

Firmware Filename: C:\jobs\DPhyDecoder\DPhyDecoder_Ver2.2.rbx

Ready.

Figure 9 – Firmware Update Dialog

4.6 Configuration Files

DPhyDecoderCtl has three file types that contain application configuration, user settings and captured data. They are all binary files with undocumented formatting. The first file type (.bin) is only used by the application to record global application state and cannot be explicitly generated by the user. Another file type (.cfg) contains almost all user settings and can be saved/loaded via File menu options. Finally, the last file type (.trc) contains captured trace data plus a few user settings important for disassembly. A trace file can also be saved/loaded via File menu options.

User configuration settings and application state are automatically saved in two application files when the application is closed and restored when the application is launched. These files are:

C:\ProgramData\Moving Pixel Company\DPhyDecoderCtl\AppConfig.bin

- last used firmware files
- last used instrument serial number, disassembly window
- last used disassembly window position and size
- user color scheme

C:\ProgramData\Moving Pixel Company\DPhyDecoderCtl\DefaultSettings.cfg

- all other configuration settings except for captured data

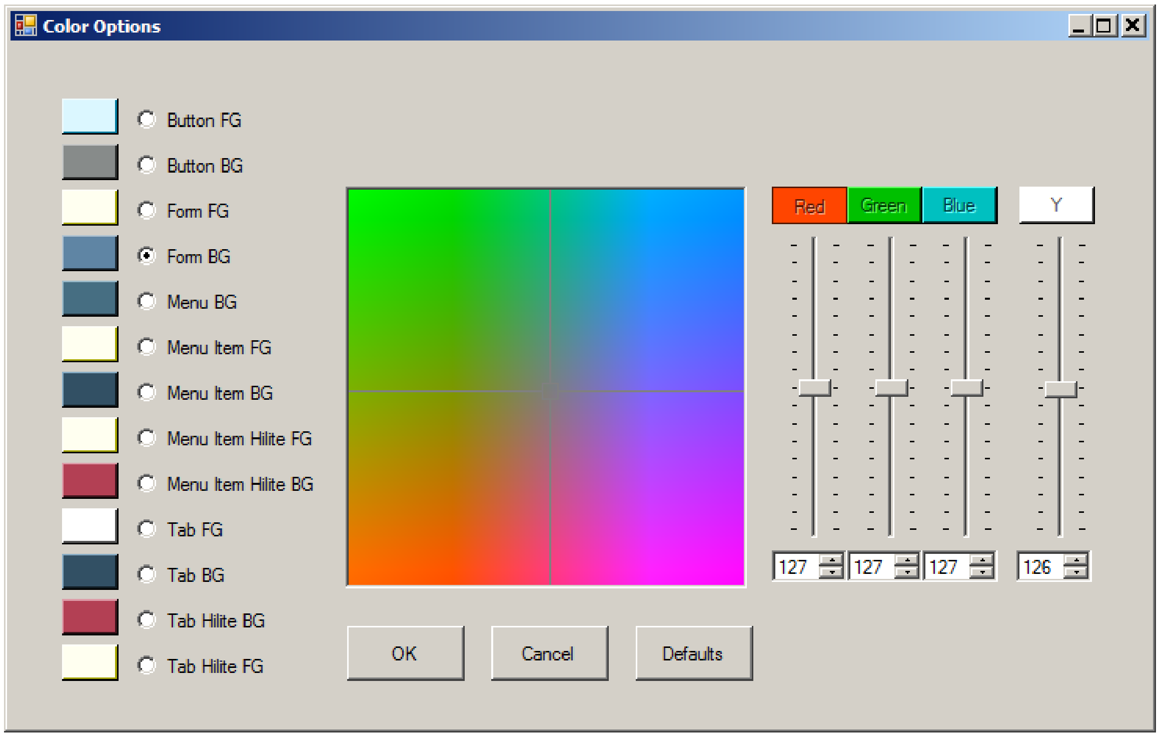
Note that when a trace file is loaded, a few critical user settings are also loaded and will overwrite their current values. These settings are: DPhy standard, lane count, HS frequency, and custom packet definitions.

4.7 Color Dialog

The color options dialog allows the user to adjust colors for use in the application. This dialog is experimental and only a few colors have the greatest impact on the aesthetics of the application, in particular, the Button background and the Form background.

To bring up the Color Options Dialog, select "Set Colors…" in the Options menu. To change a color, first select the color type option, for example in Figure 10, the "Form BG" (form background) is selected. Then left-click and drag in the color square, adjust the color sliders, or type in color values to set the color. The color rectangles to the left of the option buttons show the currently assigned color. Clicking on a rectangle makes the selected color the current color, i.e. fills in the color settings. This way, control colors can easily be copied to other controls.

As colors are selected, the main window colors change to reflect the new settings. When finished, click OK to keep the new colors, Cancel to discard the new colors, or Defaults to restore the colors to the application defaults.



Red 127, Green 127, Blue 127, Y 126; [✓ Form BG]

Figure 10 – Color Dialog

4.8 Menus

This section outlines the menu commands available in DPhyDecoderCtl:

- **File**:
  - **Load –** loads a previously saved configuration file, overwriting the current configuration.
  - **Save Cfg --** saves the current configuration to a file.
  - **Save Cfg As… --** same as the **Save Cfg** command above except a dialog appears to browse and enter a new configuration file name.
  - **Load Trace** – loads a previously saved trace file, overwriting the current captured data.
  - **Save Trace As**… **--** saves the current captured data to a trace file specified through a file dialog.
  - **<recent files>** a list of 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
- **Connect**:
  - **Connect to DPhy Decoder…** – brings up the connection dialog (see section 4.1).
  - **Disconnect From DPhy Decoder** – disconnects from the instrument, putting the application in offline mode.
  - **Update Firmware**... – brings up a the firmware update dialog to reprogram the DPhy Decoder firmware (see section 4.5).
- **Config:**
  - **Define Custom Commands…** – brings up the custom command dialog (see section 4.3)
  - **Set Color**s **–** brings up the Color Options dialog to customize the colors of forms, buttons, and other controls.
- **About**
  - **Help –** displays this 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.

5 Quick Start

This section gives an overview of how to setup and configure the application to make an initial acquisition and disassembly.

1. Run the Setup program that installs the DPhyDecoderCtl application and the USB driver used to communicate with the DPhy Decoder.

2. Connect the DPhy Decoder instrument to your DUT. Please refer to the DPhy Decoder Hardware Datasheet and User Manual for how to connect your hardware to the instrument.
3. Connect the USB cable from the DPhy Decoder to the host.
4. Power-on the DPhy Decoder.
5. Launch DPhyDecoderCtl.
6. The Connection dialog appears asking for you to select the serial number of the instrument for connection (see section 4.1). If your instrument's serial number does not appear in the combo box, check that the instrument is powered-on and connected via USB to the host computer.
7. Configure the main window controls as follows:
    - Set the MIPI standard to CSI or DSI as appropriate
    - Set the Max Lane Count to the number of lanes connected to your DUT
    - Set the Data Lane Map to "3210"
    - Set the Delta Timeout to, say, "42 ms".
    - Check "Delta Mode", "Escape Mode", and Auto HS Freq"
    - Set the Trigger Type to "Immediately"
8. Click the Send button to make sure all settings have been sent to the hardware.
9. For initial testing, if possible, set up your DUT to generate continuous traffic. It is best to initially start out at a relatively low frequency (100-300 MHz).
10. At this point, you should see LP and HS activity at the bottom of the status pane in the main window and packet statistics begin to increment.
11. Click on the "Show Disasm" button to bring up the Disassembly Window.
12. The default acquisition options (Prefill Pct == 0, Memory Depth = 1 MB, and Acquire Timestamps checked) are reasonable setting to use for a first acquisition. Thus, simply click on the "Run" button at the top, right corner of the disassembly window to capture your first trace data.

6 Remote Control

To facilitate automated control and testing, DPhyDecoderCtl supports receiving requests from other applications via a TCP/IP server port. This section describes details of the communication interface, supported commands, and example client code provided with installation of the DPhyDecoderCtl application. It is assumed that the user is conversant with programming in the .NET environment or otherwise knowledgeable about interfacing between his preferred programming environment (e.g. LabView, MATLAB, Python) and .NET.

6.1 Using DPhyDecoderCtl as an RPC Server

To enable DPhyDecoderCtl as an RPC server, simply select the "Enable RPC" menu option in the Control menu. If this option is not already checked, a simple dialog will appear requesting the port number to use for incoming RPC requests. You will need to use a port number that is not already in use by your system and other applications. This number should be between 1024 and 65525, though higher numbers in the range may be preferred as there are fewer port numbers at the high-end that are used by well-known applications. For more information, go to the following link:

http://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers

To disable DPhyDecoderCtl as an RPC server, uncheck the "Enable RPC" menu option.

When DPhyDecoderCtl is enabled as an RPC server, the user can interact with the GUI as normal. In addition, RPC calls from other applications can interact with the application as well, able to perform many of the functions programmatically that the user can perform using the GUI.

Notes:

- Currently there is no access control to prevent multiple applications from connecting and making RPC calls simultaneously to the DPhyDecoderCtl server even as the user is using the application, which, of course, may cause confusing behavior.
- Where a user initiated action would normally elicit a message box on DPhyDecoderCtl, usually a warning or a confirmation, the equivalent RPC call will not do so. Instead, the operation proceeds as if the user had responded affirmative to the message.
- RPC calls block until DPhyDecoderCtl has completed processing them. In some cases, this can take a few seconds, for example when saving and restoring large trace files. For client applications that require a responsive interface, these RPC calls should be made from a separate thread.

6.2 DPhyDecoderCtlRPC Library

To facilitate the user in building RPC client programs to interact with DPhyDecoderCtl, a DLL is provided (DPhyDecoderCtlRPC.dll). This library is written in C# and is managed by .NET and it provides classes to encapsulate error codes, command definitions, command field definitions, and a few simple calls to establish a connection to the DPhyDecoderCtl server, send commands, and read status. These classes are contained in the namespace "DPhyDecoderCtlRPC" and are described below.

Note: for those interfacing to the DPhyDecoderCltRPC DLL from a language other than C# or an environment different than .NET and you have suggestions how to make the DLL more usable for your needs, please contact the Moving Pixel Company. This may be necessary especially with respect to function signatures, which currently take advantage of C# generic types and may not be easily portable to other languages or environments.

6.2.1 Class Overview

The DPhyDecoderCtlRPC namespace contains four classes:

- DPhyDecoderCtlRPCClient – main client instance to connect and send commands to the DPhyDecoderCtl server
- RPCErrs – error code definitions
- RPCCmds – RPC command code definitions

- RPCDefs – RPC command parameter definitions

These classes are further described in the following sections.

6.2.1.1 DPhyDecoderCtlRPCClient Class

The DPhyDecoderCtlRPCClient Class represents the main class of the library, providing an interface to connect to the DPhyDecoderCtl server, send RPC commands, and obtain results.

The client constructor takes no arguments and is simply created with the call:

```
DPhyDecoderCtlRPCClient client = new DPhyDecoderCtlRPCClient();
```

Next, should be a call to connect to a DPhyDecoderCtl server. The server should be running and enabled for RPC. As an example, the following call connects to a DPhyDecoderCtl server located on the local host machine and enabled for RPC at port 2799:

```
int rc = client.Connect("", 2799);
```

When finished sending RPC commands, the client can disconnect from the server with

```
int rc = client.Disconnect();
```

6.2.1.2 RPCErrs Class

The RPCErrs class contains definitions for the possible error codes that can occur in the DPhyDecoderCtl server. However, note that only a small subset of the errors in this class will be returned by an RPC call. No comprehensive documentation of these errors is provided currently – the names are meant to provide a more descriptive indication of the error that occurred. In many cases, the error code returned is also supplemented with an error message that is returned as well by the RPC call.

A few error codes deserve further comment:

- **FAIL**: used as a generic catch-all failure code.
- **LOCAL_FAILURE:** indicates that an error occurred on the client side (before reaching the DPhyDecoderCtl server). Often this error is associated with an error marshaling unexpected or incorrect command arguments.
- **ARGTYPE_MISMATCH**: this error is returned if the expected argument type for a command parameter does not match the argument type it received.
- **INVALID_PARAM:** this is a common error returned indicating that one or more arguments are out of range .

6.2.1.3 RPCCmds Class

The RPCCmds class contains definitions for all the possible RPC commands that can be sent to the DPhyDecoderCtl server. They are documented in Appendix A. These command codes are used with the RPCCmd and RPCQuery calls (and their variants) in the DLL.

6.2.1.4 RPCDefs Class

The RPCDefs class contains constants to be used for RPC command parameters. These command codes are documented in Appendix A

6.2.2 Sending RPC Commands

Sending an RPC command requires the use of a generic RPC call RPCCmd, which has four function signatures, accommodating 0, 1, 2, and 3 command arguments respectively:

```
public int RPCCmd(int cmdCode, ref string errMsg,
                        ref string statusMsg)

public int RPCCmd<T1>(int cmdCode, T1 arg1,
                        ref string errMsg, ref string statusMsg)

public int RPCCmd<T1, T2>(int cmdCode, T1 arg1, T2 arg2,
                        ref string errMsg, ref string statusMsg)

public int RPCCmd<T1, T2>(int cmdCode, T1 arg1, T2 arg2, T3 arg3,
                        ref string errMsg, ref string statusMsg)
```

Commands that execute successfully return 0. Those that result in an error return a negative error code from the RPCErrs class. In addition, commands that result in a negative error code may provide a description of the error in the returned string errMsg parameter.

The RPCCmd functions are also used in the case of queries, when the return type is an integer. In this case, the return code can be negative to indicate and error or otherwise represents the returned value of the query. For example, the GET_TRIGGER_COUNT command, if successful, returns the number of occurrences of the trigger condition since the last reset.

For queries that don't return a single integer result, the RPCQuery functions are used.

In addition, one other call may be used, specifically for calls that return other than an integer return code:

```
public int RPCQuery<T1>(int cmdCode,
         ref T1 respVal, ref string errMsg, ref string statusMsg)

public int RPCQuery<T1, T2>(int cmdCode, T1 arg1,
         ref T1 respVal, ref string errMsg, ref string statusMsg)
```

```
   public int RPCQuery<T1, T2, T3>(int cmdCode, T1 arg1, T2 arg2,
            ref T1 respVal, ref string errMsg, ref string statusMsg)
```

The calls that require RPCQuery are tagged in their description in Appendix A with a "(RPCQuery)" designation.

Which call should be used depends on the RPC command. For example to set the HS frequency, the single-argument signature call is used, e.g.

client.RPCCmd(RPCCmds.SET_HS_CLOCK_FREQUENCY,
300.0f, ref eMsg, ref sMsg)

On the other hand, to set the delta mode option, the dual-argument signature call is used, e.g.

client.RPCCmd(RPCCmds.SET_OPTION, RPCDefs.OPT_DELTA_MODE,
true, ref eMsg, ref sMsg)

6.2.2.1 Alternate Command Interface For Non-.NET Environments

Some languages or programming environments cannot support the generic style argument passing (which is a Microsoft C# construct). Accordingly, the DphyDecoderCtlRPCClient DLL supports the following alternate procedure calls:

```
public int RPCCmdF(int cmdCode, float arg1,
                  ref string errMsg, ref string statusMsg);
public int RPCCmdI(int cmdCode, int arg1,
                  ref string errMsg, ref string statusMsg);
public int RPCCmdIF(int cmdCode, int arg1, float arg2,
                  ref string errMsg, ref string statusMsg);
public int RPCCmdII(int cmdCode, int arg1, int arg2,
                  ref string errMsg, ref string statusMsg);
public int RPCCmdS(int cmdCode, string arg1,
                  ref string errMsg, ref string statusMsg);
public int RPCCmdIRetS(int cmdCode, int arg1, ref string rVal,
                  ref string errMsg, ref string statusMsg);
public int RPCCmdSRetS(int cmdCode, string arg1, ref string rVal,
                  ref string errMsg, ref string statusMsg);
```

6.3 TestDPhyDecoderRPC Project

When DPhyDecoderCtl is installed, a Visual Studio 2008 project written in C# is included that demonstrates use of the RPC interface. The project is called TestDPhyDecodereRPC and is located in the DPhyDecoderCtl top-level application directory:

c:\Program Files\TMPC\DPhyDecoderCtl

The Main procedure in the project is located in Program.cs and simply connects to the DPhyDecoderCtl server, which is assumed to be running on the local host and enabled for RPC at port 3333. In addition, DPhyDecoderCtl should be connected to a DPhy Decoder instrument. Once connected, the routine simply progresses through almost all of the RPC commands available. This code doesn't perform anything particularly useful and is intended to be run in the debugger, stepping or running with breakpoints set at locations of interest.

Note in the References section the reference to the DPhyDecoderCtlRPC DLL, which is located in the bin/Debug directory. In addition, a type library of the DLL is also provided, which may be useful to non-.NET programmers as it contains all of the DLL routine calling signatures.

7 Appendix A: RPC Command Reference

Table 1: RPC Commands (RPCCmd class)

RPC Command	Code	Description
File I/O Commands		
LOAD_CONFIG_FILE arg1 = configuration filename (string)	0x01	Loads a configuration file from the given file.
SAVE_CONFIG_FILE arg1 = configuration filename (string)	0x02	Saves the current configuration from the given file.
LOAD_TRACE_FILE arg1 = trace filename (string)	0x03	Loads a trace file from the given file.
SAVE_TRACE_FILE arg1 = trace filename (string)	0x04	Saves the current acquisition to the given file.
Configuration Write Commands		
SET_CUSTOM_SHORT_PACKETS arg1 = list of short packet dataIDs (int[], range 0-63)	0x20	Sets the custom short dataID list for decoding.
SET_CUSTOM_LONG_PACKETS arg1 = list of long packet dataIDs (int[], range 0-63)	0x21	Sets the custom long dataID list for decoding.
SET_CUSTOM_DCS_PACKETS arg1 = list of DCS commands (int[], range 0-255)	0x22	Sets the custom DCS command list for decoding.
SET_MIPI_STANDARD arg1 = standard (int, use STANDARD_x defines)	0x23	Sets the current MIPI standard.
SET_MAX_LANE_COUNT arg1 = maximum lane count(int, range 1-4)	0x24	Sets the Max Lane Count setting
SET_DELTA_TIMEOUT arg1 = delta timeout value (int, use TIMEOUT_x defines)	0x25	Sets the Delta Timeout setting
SET_DATA_LANE_MAP arg1 = data lane mapping (int, 16-bit value)	0x26	Sets the Data Lane map setting.
SET_HS_CLOCK_FREQUENCY arg1 = HS clock frequency (float, MHz, range 30.0-750.0)	0x27	Sets the HS clock frequency
SET_OPTION arg1 = option number (int, use OPT_x defines) arg2 = option value (bool, or int with value of 0 or 1)	0x28	Sets a checkbox option
SET_CLOCK_DELAY arg1 = clock delay (int, ps, range -1400 to 7000)	0x29	Sets the HSClk Adjust settings. Note: this value is applied relative to the ideal (or best) clock setting (i.e. 0 sets the clock offset to ideal).
SET_HS_ONLY_PATTERN Arg1 = HS-Only header pattern (int, LSB is DataID)	0x2A	Sets the 32-bit HS-Only pattern to used in HS-Only mode.
CAL_HS_CLOCK returns error of number of error-free ps found in window	0x2B	Sweeps the HSClk Adjust setting to automatically determine the optimum sampling location. (Equivalent to clicking the Cal button)

Configuration Read Commands

GET_CUSTOM_SHORT_PACKETS respVal = list of short packet dataIDs (int[])	0x40	Gets the custom short dataID list (RPCQuery)
GET_CUSTOM_LONG_PACKETS respVal = list of long packet dataIDs (int[])	0x41	Gets the custom long dataID list (RPCQuery)
GET_CUSTOM_DCS_COMMANDS respVal = list of DCS commands (int[])	0x42	Gets the custom DCS command list (RPCQuery)
GET_MIPI_STANDARD returns error or MIPI standard (STANDARD_x define)	0x43	Gets the current MIPI standard setting
GET_MAX_LANE_COUNT returns error or Max Lane Count	0x44	Gets the current Max Lane Count setting
GET_DELTA_TIMEOUT returns error or Delta Timeout (TIMEOUT_x define)	0x45	Gets the current Delta Timeout setting
GET_DATA_LANE_MAP returns error or Data Lane Map value	0x46	Gets the current Data Lane Map setting
GET_HS_CLOCK_FREQUENCY respVal = HS clock frequency (float, MHz)	0x47	Gets the current HS Clock Frequency setting (RPCQuery)
GET_OPTION arg1 = option number (int, use OPT_x defines) returns error or option value	0x48	Gets a checkbox option setting
GET_CLOCK_DELAY returns error or clock delay setting (ps)	0x49	Gets the current HSClk Adjust setting

Status Read Commands

GET_ECC_ERROR_COUNT returns error or ECC error count	0x60	Gets the ECC error count.
GET_CRC_ERROR_COUNT returns error or CRC error count	0x61	Gets the CRC error count.
GET_TRIGGER_COUNT returns error or trigger count	0x62	Gets the trigger count.
GET_SHORT_PACKET_COUNT returns error or short packet count	0x63	Gets the short packet count.
GET_LONG_PACKET_COUNT returns error or long packet count	0x64	Gets the long packet count.
GET_LP_PACKET_COUNT returns error or LP packet count	0x65	Gets the LP packet count.
GET_HS_PACKET_COUNT returns error or HS packet count	0x66	Gets the HS packet count.
GET_TOTAL_PACKET_COUNT returns error or total packet count	0x67	Gets the total packet count.
GET_FILTER_PACKET_COUNT returns error or filter packet count	0x68	Gets the filter packet count.
GET_HS_ACTIVE_LANES returns error or number of HS active lanes	0x69	Gets the number of HS active lanes detected.
GET_MEASURED_HS_CLOCK_FREQUENCY respVal = measured HS clock frequency (float, MHz)	0x6a	Gets the measured HS clock frequency.
GET_LANE_ACTIVITY returns error or lane activity	0x6b	Gets the lane activity.

Miscellaneous Control Commands

SEND_CONFIG	0x70	Sends the current configuration to the hardware, resetting if "No Reset On Send" option is unset.
CLEAR_STATS	0x71	Resets the status counters.

Trigger/Filter Configuration Commands

SET_TRIGGER_DEFAULTS	0x80	Resets trigger controls to their defaults (No packet types selected, all terms Unused, Trigger Immediately)
SET_TRIGGER_EQUATION arg1 = trigger equation (string)	0x81	Sets the Trigger Equation string
SET_TRIGGER_TYPE arg1 = trigger type (int, use TRIGGER_x defines)	0x82	Sets the Trigger Type
SET_TRIGGER_TERM arg1 = term array (int[5)) where: 0: term index (1-6) 1: packet byte offset (0-9) 2: mask (16-bit int) 3: comparison opcode (use OP_x defines) 4: value (16-bit int)	0x83	Sets a trigger term
CLEAR_TRIGGER_PACKET_TYPES	0x84	Clears all trigger packet type selections
ADD_TRIGGER_PACKET_TYPES arg1 = list of dataIDs and DCSCmds (int[]) (OR values with CATEGORY_x defines)	0x85	Adds trigger pacekt type selections to the current selection set.
CLEAR_FILTER_PACKET_TYPES	0x86	Clears all filter packet type selectiosn
ADD_FILTER_PACKET_TYPES arg1 = list of dataIDs and DCSCmds (int[]) (OR values with CATEGORY_x defines)	0x87	Adds filter packet type selections to the current selection set.

Acquisition Control Commands

SET_PREFILL_PCT arg1 = prefill percent (int, 0-99)	0xa0	Sets the prefill buffer percentage (i.e. trigger position in acquisition buffer)
SET_MEMORY_DEPTH arg1 = memory depth (int, (1<<13) to (1 << 27))	0xa1	Sets the acquisition buffer size in bytes.
RUN	0xa8	Starts an acquisition
STOP	0xa9	Forces a trigger during an acquisition.
ABORT	0xaa	Stops an acquisition, upload, or disassembly.
GET_RUN_STATUS returns error or run status (STATUS_x defines)	0xab	Gets the current acquisition.
GET_PREFILL_PCT returns error or prefill percent (int)	0xac	Gets the current prefill buffer percentage
GET_MEMORY_DEPTH returns error or memory depth (int)	0xad	Gets the current acquisition buffer size.

Disassembly Window Commands

GET_FRAME_COUNT returns the number of frames (int)	0xf0	Gets the number of frames detected in the listing.
GET_SAMPLE_COUNT returns the number of samples (int)	0xf1	Gets the total number of samples in the acquisition.
GET_DISASM_FIELD_STRING returns the disassembly field string (string)	0xf2	Gets the string displayed in the disassembly window, given the sample number and column name. (RPCQuery)
GET_VIDEO_SUMMARY_FIELD_STRING returns the video summary field string (string)	0xf3	Gets the string displayed in the video summary tab, given the ones-based frame number and column name. (RPCQuery)
GET_FRAME_START_SAMPLE returns the sample number of the frame start (int)	0xf4	Gets the sample number of the start of a frame given its ones-based frame number.

SAVE_FRAME
arg1 = ones-based frame index (int)
arg2 = file name (string)
	0xf5	Saves the referenced frame to a file. The file type is determined by the extension (BMP, JPG, TIF, GIF) or otherwise is saved in binary format.

SET_CURRENT_SAMPLE arg1 = sample number (int)	0xf6	Sets the current sample record in the disassembly window (scrolls to view).

SET_MNEM_VIEW_STATE
arg1 = start sample index (int)
arg2 = end sample index (int, inclusive)
arg3 = mnem view state (int, VIEW_STATE_x define)
	0xf7	Sets the mnemonic view state for the given sample range.

SET_SEARCH_STRING arg1 = search string (string) Define terms as "<IF | AND | OR> <field> <op> <value>" Use '\n' to separate terms Example: "IF CRCOk == Err\n OR ECCOk == Err"	0xf5	Sets the search field equation controls in the search tab from the given string.
NEXT_PREV Arg1 = search function opcode (int) Use (NEXT_x and PREV_x defines)	0xf7	Call a search function. Same as context menu search functions.

EXPORT_CSV
arg1 = start sample index (int)
arg2 = end sample index (int, inclusive)
arg3 = CSV file name (relative to DPhyDecoderCtl app)
	0xf8	Exports the given sample range to a CSV file.

Table 2: MIPI Standards (RPCDefs class)

STANDARD_DSI	0
STANDARD_CSI	1

Table 3: Delta Timeouts (RPCDefs class)

TIMEOUT_NONE	0
TIMEOUT_1_US	1
TIMEOUT_10_US	2
TIMEOUT_82_US	3
TIMEOUT_650_US	4
TIMEOUT_5_MS	5

TIMEOUT_42_MS	6	

Table 4: Options (RPCDefs class)

OPT_DELTA_MODE	0	
OPT_ESCAPE_MODE	1	
OPT_AUTO_HS_FREQ	2	
OPT_NO_RESET_ON_SEND	3	
OPT_STATS_PER_ITER	4	
OPT_ACQUIRE_TIMESTAMPS	5	
OPT_HOLD_STATS	6	
OPT_SUSPEND_SEND_FOR_RPC	7	Enables/disables auto-send after each RPC command. Use before long sequence of RPC configuration commands.
OPT_HS_ONLY_MODE	8	

Table 5: Trigger Types (RPCDefs class)

TRIGGER_PACKET_FIELDS	0
TRIGGER_LP_SIGNAL_STATE	1
TRIGGER_LP_SEQUENCE_STATE	2
TRIGGER_ESCAPE_COMMAND	3
TRIGGER_CRC_ERROR	4
TRIGGER_ECC_ERROR	5
TRIGGER_IMMEDIATELY	6
TRIGGER_CRC_OR_ECC_ERROR	7

Table 6: Trigger Term Opcodes (RPCDefs class)

OP_EQ	0
OP_NEQ	1
OP_LT	2
OP_LTE	3
OP_GT	4
OP_GTE	5

Table 7: Run Status (RPCDefs class)

STATUS_IDLE	0
STATUS_CAPTURING	1
STATUS_TRIGGERED	2
STATUS_UPLOADING	3
STATUS_DISASSEMBLING	4

Table 8: Trigger/Filter Packet Type Flags (RPCDefs class)

CATEGORY_STANDARD	0x000
CATEGORY_DCS	0x100
CATEGORY_PERIPHERAL	0x200

Table 9: Mnemonic View States (RPCDefs class)

VIEW_STATE_MNEM_ONLY	0

VIEW_STATE_MNEM_PLUS_MNEM_FIELDS	1
VIEW_STATE_MNEM_PLUS_PKT_BYTES	2

Table 10: NEXT_PREV Opcodes (RPCDefs class)

SEARCH_FRAME_START	13
SEARCH_SAME_MNEM	14
SEARCH_SAME_VAL	15
SEARCH_SAME_MNEM_AND_VAL	16
NEXT_SEARCH	18
NEXT_PACKET	19
NEXT_FRAME_START	20
NEXT_SAME_MNEM	21
NEXT_SAME_VAL	22
NEXT_SAME_MNEM_AND_VAL	23
PREV_SEARCH	25
PREV_PACKET	26
PREV_FRAME_START	27
PREV_SAME_MNEM	28
PREV_SAME_VAL	29
PREV_SAME_MNEM_AND_VAL	30
NEXT_NONNULL_FIELD_VAL	100
PREV_NONNULL_FIELD_VAL	101