Programmer's Guide

DLPC7540

ABSTRACT

This guide provides details of the software interface requirements for a DLPC7540 controller-based system. This description includes the communication protocol, initialization, default settings, common use cases, and command descriptions.

Table of Contents

List of Figures

List of Tables

Trademarks

All trademarks are the property of their respective owners.

1 Scope

This guide provides details of the software interface requirements for a DLPC7540 controller-based system as part of the software revision v5.0. This description includes the communication protocol, initialization, default settings, common use cases and command descriptions.

Figure 1-1 shows a typical projector system using DLPC7540 controller that includes external power management IC and the .47 4K HSSI DMD.

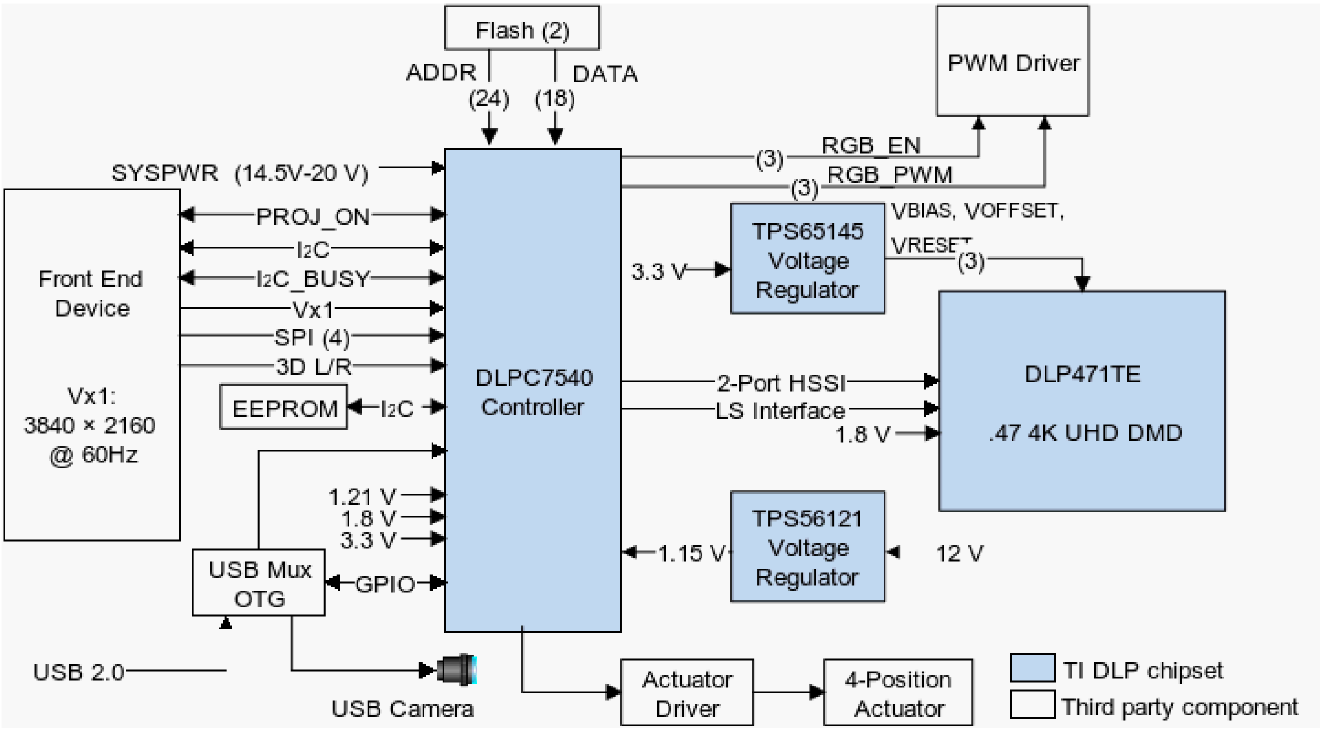


Figure 1-1. Typical Projector System Block Diagram

2 References

1. DLPC7540 data sheet
2. I^{2}C bus specification
3. DLP Composer tool user's guide

3 Acronyms

CCA - Color coordinate adjustment

CFI - Common flash interface

CSC - Color space conversion

DB – Dynamic black

DLPC – DLP controller

DMD - Digital Micromirror Device

HSG - Hue saturation gain

SFG – Solid field generator

GPIO – General purpose input/output

LUT - Lookup table

PWM – Pulse width modulation

SSI - Solid state illumination

TPG - Test pattern generator

4 System Boot

The DLP controller boots from the parallel flash connected at the PM_CSZ_0 line. There is no ROM code built into the controller. It is mandatory to have parallel flash connected on the CS0 chip select line.

4.1 Data In flash

These are the major sections of data present in the flash memory.

- Bootloader application
- Main application
- Configuration data
- Display sequences
- Splash images
- Auto-init batch file

4.2 Bootloader Application

The bootloader is the first application that runs from the flash memory when the system is turned on or reset. The bootloader application copies itself from flash to internal RAM for execution. This application performs flash update (erase, program). It also identifies a valid main application in the flash and only then begins to run the main application. This application reads the GPIO_64 (HOLD_BOOTZ) signal at the startup and if the signal reads 0, the application remains in the boot application mode. This useful option forces the firmware to update in case the main application on the flash is corrupted. Use the DLP Control Program to update the flash firmware.

Table 4-1 lists all commands supported by the bootloader application for flash update. Figure 4-1 shows example use of bootloader commands for updating the flash contents.

Table 4-1. Supported Flash Update Commands

Command	Description
Boot Hold Reason	Reason for controller to be in boot application mode.
Get Flash ID	Returns the flash ID
Get Flash Sector Info	Retrieve flash number of sectors and sector size information
Flash Lock/Unlock	User must send this command to unlock the flash for erase/program access. This is to prevent accidental erase/programming of flash data.
Erase Sector	Command to erase sector – user to provide sector address as input
Initialize Flash Read/Write	Command to specify start address on the flash along with number of bytes to be written or readback
Get Checksum	Command to compute checksum and return it. Command takes flash address and number of bytes to compute the checksum.

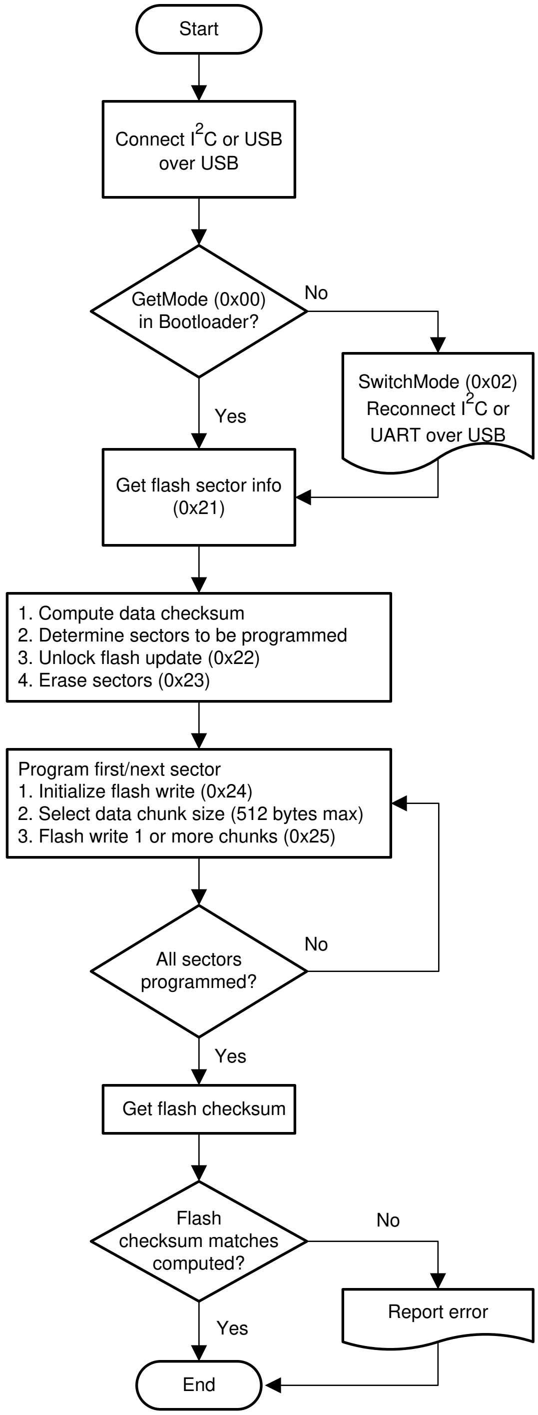


Figure 4-1. Flash Update Flow Diagram

4.3 Main Application

This application runs during the normal projector operation and performs full system initialization including DMD, illumination sub-system and peripherals. controller responds to all the control commands from host controller and takes appropriate actions and sends responses.

4.4 Commands Supported by Bootloader and Main Applications

The bootloader application and main application support these commands:

Command Name	Description
Get Mode	Returns the current mode – bootloader or main Application
Get Version	Returns software version information
Switch Mode	Switch between bootloader mode and main application mode

4.5 Debug Terminal

The application prints several status messages and debug information onto its UART Port0. User has the option to configure that to a different port using DLP Composer tool. The default settings for this UART port is set to 115200 baud rate, 8-bit data, no parity, 1-stop bit and no flow control. These parameters may be changed as per user preference using DLP Composer tool.

The level of messages on the debug terminal is configurable via Set Debug Message Mask command. User may choose to route the debug messages to USB port instead of UART port using Enable USB Debug Log command.

4.6 HOST_IRQ/SYSTEM_BUSY

GPIO_58 acts as HOST_IRQ/SYSTEM_BUSY signal, configured as open drain GPIO from the controller side. The user can assign a different pin for this purpose using DLP Composer tool. The GPIO indicates when the controller is free or busy. During power-on-reset, the front-end communication device must wait until the signal goes to LOW state when host processor is available to receive commands. When the signal remains continuously HIGH, this indicates problem with controller boot-sequence. In this case it is important that the issue resolved before proceeding

4.7 Heartbeat

After a successful boot-up, the controller begins toggling the GPIO_28. Typical operation reflects 1Hz, 50% duty-cycle. If the device detects and error, the signal changes to 5Hz, 50% duty-cycle. The front-end component can read through the System Status command. Use the DLP Composer tool to assign a different pin and/or polarity for this purpose.

4.8 Low-Level Fault

When the controller startup sequence encounters an error condition, the controller transitions to bootloader mode and sets GPIO_23 (connected to an LED) HIGH to indicate a low-level fault condition. See the debug message printed in the UART debug terminal or with the Get Boot Hold Reason command to get more details about the cause of error. When the Arm processor detects one of the data abort, pre-fetch abort or undefined instruction exceptions, it starts blinking GPIO_23 in specific hex code format. During this error condition, communication with the controller is not possible. The only way to communicate with the controller is to analyze the hex error code from the pattern and then debug the hardware to fix the issue.

5 System Status

The front-end controller can poll the System Status Read command to get information related to system status and error conditions.

6 Version

The front-end can query the version information of application software and the underlying API library using the Version Read command.

7 Power Modes

The DLP Controller can operate in Normal or Standby power modes. Use Set Power command to toggle between the two modes. In the standby state the controller will be consuming the minimal power. When the system is not in use, user can set system in Standby state. Note the state transition may take several hundred milliseconds to complete. User shall ensure power mode transition is complete using Get Power Mode command.

8 Display Modes

When the system is operating in normal power mode, the user can change the display mode of the system using Set Display command. Display of contents from different sources such as Test Pattern (internal preprogrammed), Solid Field, Splash (logo image), Curtain, and External Source are supported. Controller firmware is designed to hide mode transition artifacts. However, where the transition artifacts are not completely hidden, user has the option to freeze the display (using Enable Freeze command), display a blank curtain or turn off illuminator (using Set Illumination Enable command) to hide those artifacts.

9 Source Detection and Configuration

Follow specific steps to configure the controller to display a source properly. When operating in External display mode, the controller automatically scans the connectors for activity and runs the automatic source detection and locking algorithm.

The front-end controller gets information about scan status using Get Datapath Scan Status command. After a source is detected, all information about the source can be queried using Get Source Configuration command. Anytime after source detection, the user may override any of the auto-detected parameters via Set Source Configuration command.

If the source detection is incorrect or if any source parameters have changed, initiate a re-sync using Autolock Setup command.

10 Internal Sources

This section describes multiple internal source options to display by using Display command.

10.1 Test Patterns (TPG)

The controller has several pre-defined patterns that the user can select using TPG Pre-defined Pattern command. Use the DLP Control program tool to configure these pre-defined patterns the controller TPG block generates the pattern data. This option is useful for:

- Testing the DLP hardware without an external source
- Isolating whether an issue is arising from front-end source or related to image processing by the controller

These are additional configuration commands specific to test patterns.

Command Name	Description
Set TPG Border	Set border around selected TPG, border width can be 0-20; and border color programmable (R,G,B) value in the range of 0-1023.
Set TPG Resolution	Set the TPG pattern resolution. In case the resolution of the pattern is smaller than the display resolution of the DMD, controller will fill the area with pleasing color.
Set TPG FrameRate	Configure TPG frame-rate between 30Hz-120Hz. There is dependency of the TPG resolution with regard to frame-rate; for example, in case of 4K resolution, the maximum frame-rate is limited to 60Hz, and in case of 1080p resolution, the maximum frame rate is limited to 240Hz.

10.2 Solid Field (SFG) Color

When operating in SFG mode, the controller fills the entire display image area with a solid color. Use the Set SFG Color command to choose the color.

10.3 Curtain

Curtain color is similar to SFG color, but generated at the last block of the controller datapath processing unit. This is for hiding any or all artifacts up the line. The command can generate fixed colors as defined in the Set Curtain Color command.

11 Display Formatting

The controller provides several functions related to formatting the displayed image that are summarized in the table below.

Command Name	Description
Image Flip	Flip image in horizontal and/or vertical direction.
Keystone Angles	Controller adjust the display image automatically as per the 3D keystone settings where it takes three co-ordinates Pitch, Yaw, Roll; for 1D keystone user can set the Yaw and Roll to 0. This feature is useful when projecting on surfaces/screens non-orthogonal to projector.
Keystone Corners	Configures the 2D keystone correction when the corners of the corrected image are known.
Display Image Size	Define a custom displayed image size.
Manual Warp Table	Send warp points to be used for image warping.
Manual Warp Table	Define the width and height of two-dimensional point array sent using Manual Warp Table command. This command also enables/disables manual warping feature.

12 Image Processing

The controller has multiple digital image processing options summarized in the table below:

Command Name	Description
Brightness Adjustment	Provides ability to add or subtract fixed bias from each of the input R,G, B channels.
Contrast Adjustment	Provides option apply gain to the pixel data.
Hue and Color Control	Provides option to apply hue adjustment in degrees and gain in percent for each input channels.
Image Sharpness	Provides option to apply both horizontal and vertical sharpness filters.
RGB Offset	Offset the levels of the RGB channels in the datapath after brightness, contrast, hue, color, gain, CSC (color space conversion).
RGB Gain	Adjusts individual R, G & B gains of the source image. This function adjusts R, G, and B gains by altering the color space conversion coefficients.
Color co-ordinate adjustment (CCA)	CCA takes both desired and measured individual color xyY information for the color adjustment.
Hue Saturation Gain (HSG)	Same as CCA but the colors are expressed in HSG color space.

13 Warping

DLPC7540 has a powerful warping engine that enables performing arbitrary geometric modulation of the input video frame. This section describes the APIs that are provided for the customer to use this warp engine and implement a custom warp application.

The manual warping can be used in applications like projection surface non-uniformity correction, optical distortion correction and non-planar surface projection and so on.

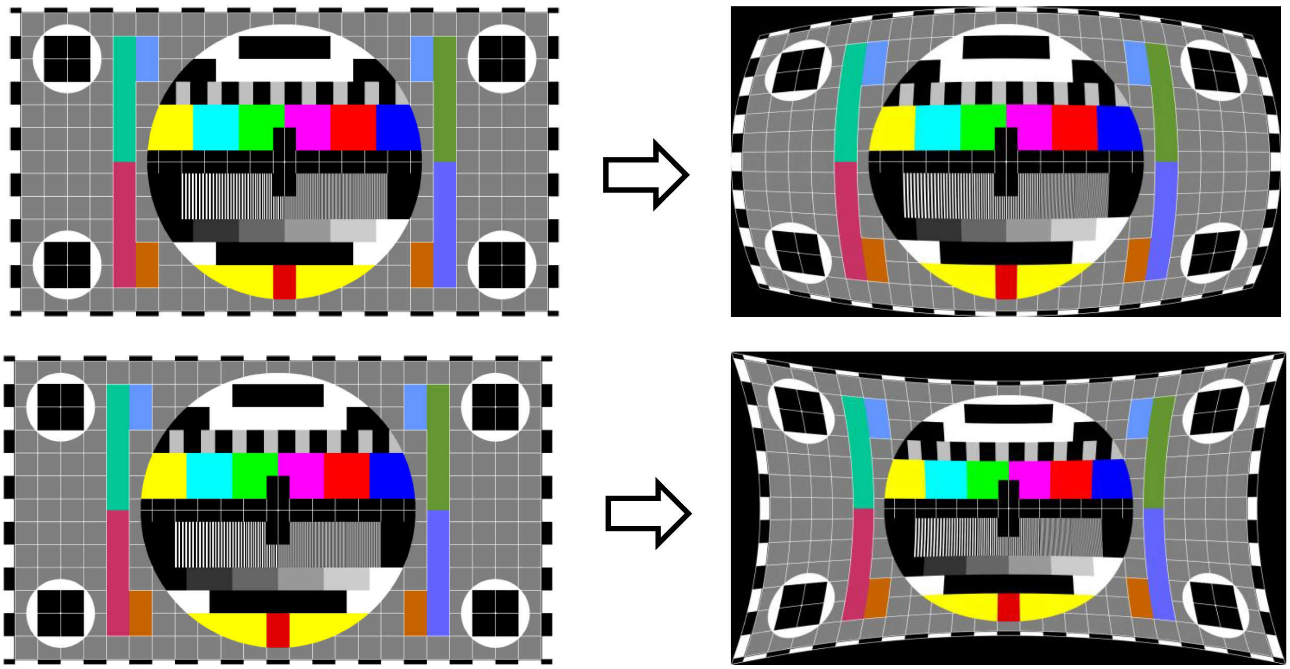


Figure 13-1. Example of Warping

13.1 Control Point Table

Control point table is used to configure position of source points. These source points are moved by the warping engine to their respective destination Manual Warp Table. Control point table also determine the number of rows and columns in Manual Warp Table. The number of destination points is implied to be same number as the source points.

13.2 Manual Warp Table

A manual warp table is a way to configure the warping engine using a 2D array of points called a warping map. These points represent the destination position of the input pixel on the DMD.

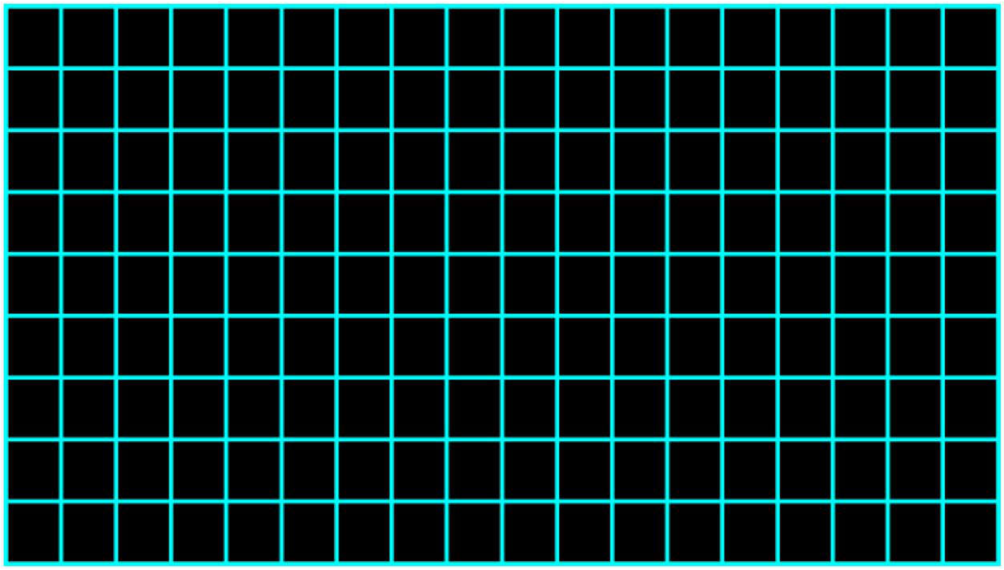


Figure 13-2. Source Control Points

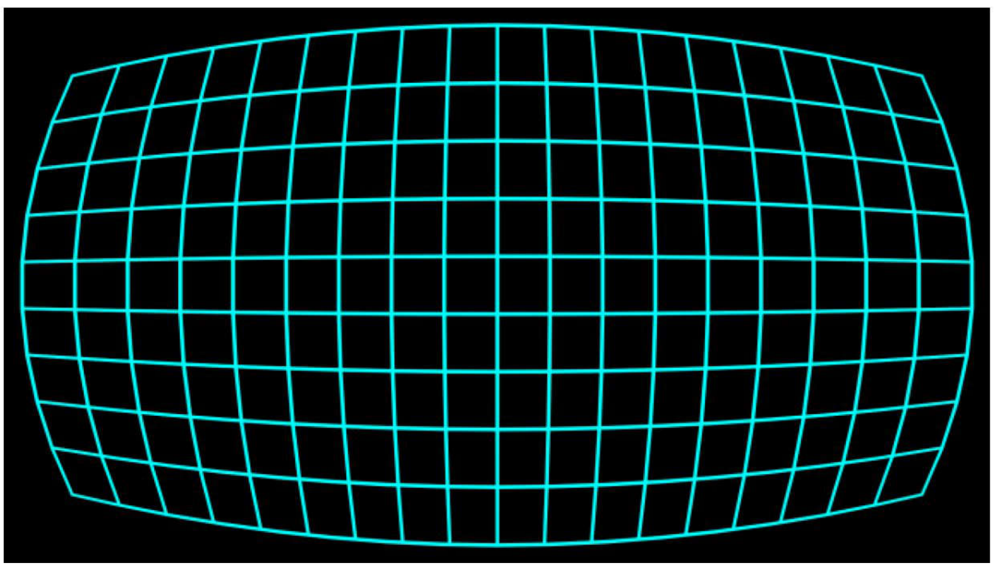


Figure 13-3. Destination Control Points

The intersection points in the previous image represent the 2D array of positions.Figure 13-2 represents the original pixel positions to move in the input frame (Control Points). Figure 13-3 represents the corresponding pixel destination position on the DMD (Warping Map).

This warping map is loaded to the system as a linear array called the warping table. The warping engine moves the grid point pixels according to this warping table. All the intermediate pixels are moved in a bilinear fashion.

13.3 Table Constraints

There are 62 horizontal and 32 vertical control points used to define a 62 × 32 map on the input image. Users can choose to define all 62 horizontal points and 32 vertical points for a custom map of source points. Users can also choose to define a number of horizontal points and vertical points in the ranges 2 to 62 and 2 to 32, respectively. In the latter case, all points remain equidistant. For unequal distribution, the former mode must be used.

The total number of points in the user-defined table cannot exceed the number of points in the control point map. A 62 × 32 control point map is most optimal and provides 1984 points. Because the warp table represents the whole output rectangle, the table must have at least 2 rows and 2 columns. The points in the warp table must monotonically increase in the left-to-right and top-to-bottom directions (that is, points cannot crisscross each other). If any of these conditions is violated, the command will return failure.

13.4 Example Warp Table

Figure 13-4 shows a butterfly like warping use case with 4 × 3 control points on a 3840 × 2160 product resolution. For simplicity, this example uses symmetric warping and integer positions. Customers can define an asymmetric map and real number positions in fixed point 13.3 format.

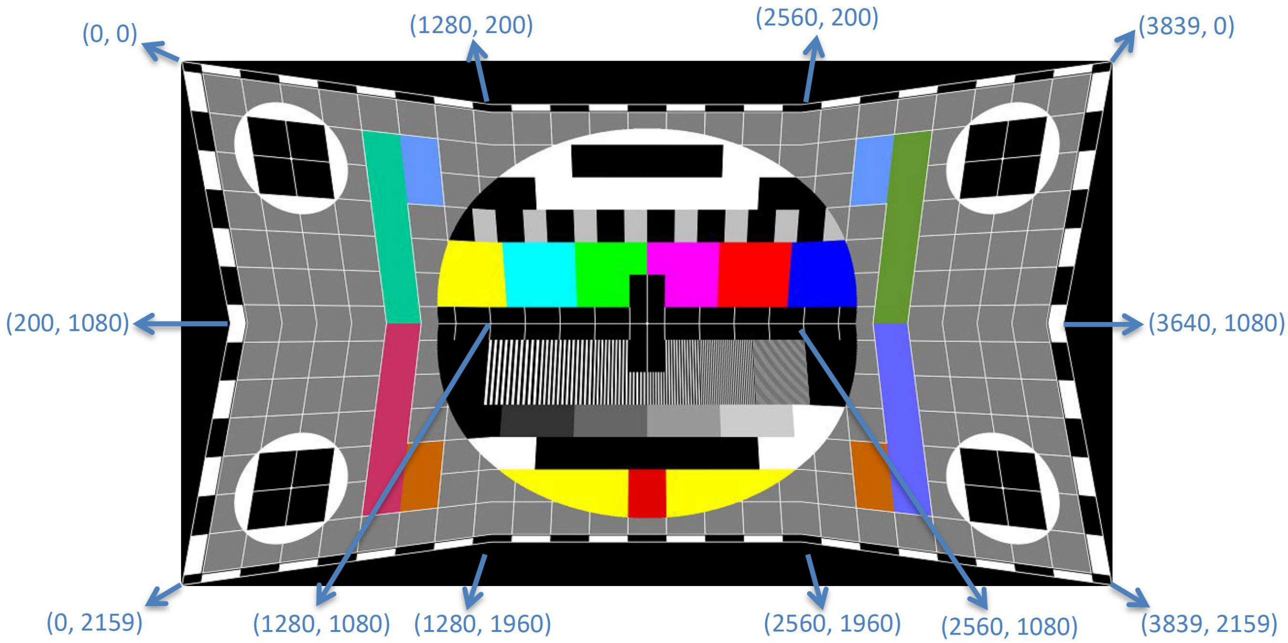


Figure 13-4. Example of Warping

Table Dimension

Number of Rows : 3

Number of Columns : 4

Control Points / Source Points

Number of Horizontal Control Points : 4 (= number of columns)

Number of Vertical Control Points : 3 (= number of rows)

The following map is generated:

[0.0, 0.0] [1279.4, 0.0] [2559.6, 0.0] [3839.0, 0.0]

[0.0, 1079.0] [1279.4, 1079.0] [2559.6, 1079.0] [3839.0, 1079.0]

[0.0, 2159.0] [1279.4, 2159.0] [2559.6, 2159.0] [3839.0, 2159.0]

Following equation calculates the control points for a given DMD resolution and warp map dimension:

Xi = (DMDWidth - 1) / (NumColumns - 1) × ColumnNumber

Yi = (DMDHeight - 1) / NumRows - 1) × RowNumber

Warping Map Points

[0.0, 0.0] [1280.0, 200.0] [2560.0, 200.0] [3839.0, 0.0]

[200.0, 1080.0] [1280.0, 1080.0] [2560.0, 1080.0] [3640.0, 1080.0]

[0.0, 2159.0] [1280.0, 1960.0] [2560.0, 1960.0] [3839.0, 2159.0]

Table 13-1. Manual Warp Table

Table Index	Floating Point Format	13.3 Fixed Point Format
0	0.0	0
1	0.0	0
2	1280.0	10240
3	200.0	1600
4	2560.0	20480
5	200.0	1600
6	3839.0	30712
7	0.0	0
8	200.0	1600
9	1080.0	8640
10	1280.0	10240
11	1080.0	8649
12	2560.0	20480
13	1080.0	8649
14	3640.0	29120
15	1080.0	8649
16	0.0	0
17	2159.0	17272
18	1280.0	10240
19	1960.0	15680
20	2560.0	20480
21	1960.0	15680
22	3839.0	30712
23	2159.0	17272

13.5 Manual Warping Commands

The following sections describe the manual warping commands that can configure the warping engine.

13.5.1 CMD_SetManualWarpControlPoints [Command ID: 0x35, Destination: 4]

This command configures the warping engine control points. According to these parameters the warp table is interpreted as a 2D warp map. The manual warp table should be configured using WriteManualWarpTable command after this operation. Warp Table is always assumed have the number of warp points same as the control points set-up using this command.

Note 1: If ‘Control Points Defined By Array’ is set to True, provide all 62 horizontal and 32 vertical points. If set to false, the number of control points is per number of columns and rows specified and the points are distributed evenly (equidistant) from each other.

Note 2: For use cases where it is desired to avoid any interpolation errors (that is, where user-defined control points are used in the controller hardware without any modification), TI recommends to use the exact control point values as shown in the example in the table below. This is because those are the control points that the API uses internally for scaling the warp map that is eventually merged with user-provided manual warp map. Using any other control point values can result in interpolation errors while merging the user-provided map with the API internal scaling map.

Example Command Packet

Table 13-2. Control Points NOT Defined By Array

ByteIndex	Value	Description
0	0x44	Command packet header (destination 4 , response required)

Table 13-2. Control Points NOT Defined By Array (continued)

ByteIndex	Value	Description
1	0x35	Command opcode
2	0x0	Control points defined by array = 0
3	0x0004	Number of equidistant horizontal control points
5	0x0003	Number of equidistant vertical control points

Table 13-3. Control Points Defined By Array

ByteIndex	Value	Description
0	0x44	Command packet header (destination 4, response required)
1	0x35	Command opcode
2	0x1	Control points defined by array = 1
Horizontal control points		
3	0x0000	0
5	0x0040	64
7	0x0080	128
9	0x00BF	191
11	0x00FF	255
13	0x013E	318
15	0x017E	382
17	0x01BD	445
19	0x01FC	508
21	0x023C	572
23	0x027B	635
25	0x02BB	699
27	0x02FA	762
29	0x033A	826
31	0x0379	889
33	0x03B9	953
35	0x03F8	1016
37	0x0437	1079
39	0x0477	1143
41	0x04B6	1206
43	0x04F6	1270
45	0x0535	1333
47	0x0575	1397
49	0x05B4	1460
51	0x05F3	1523
53	0x0633	1587
55	0x0672	1650
57	0x06B2	1714
59	0x06F1	1777
61	0x0731	1841
63	0x0771	1905
65	0x0790	1936
67	0x07D0	2000
69	0x0810	2064
71	0x084F	2127
73	0x088F	2191

Table 13-3. Control Points Defined By Array (continued)

ByteIndex	Value	Description
75	0x08CE	2254
77	0x090E	2318
79	0x094D	2381
81	0x098C	2444
83	0x09CC	2508
85	0x0AOB	2571
87	0x0A4B	2635
89	0x0A8A	2698
91	0x0ACA	2762
93	0x0B09	2825
95	0x0B48	2888
97	0x0B88	2952
99	0x0BC7	3015
101	0x0C07	3079
103	0x0C46	3142
105	0x0C86	3206
107	0x0CC5	3269
109	0x0D05	3333
111	0x0D44	3396
113	0x0D83	3459
115	0x0DC3	3523
117	0x0E02	3586
119	0x0E42	3650
121	0x0E81	3713
123	0x0EC1	3777
125	0x0EFF	3839
127	0x0EC1	3777
129	0xOEFF	3839
Vertical control points		
131	0x0000	0
133	0x0046	70
135	0x008B	139
137	0x00D1	209
139	0x0117	279
141	0x015C	348
143	0x01A2	418
145	0x01E8	488
147	0x022D	557
149	0x0273	627
151	Ox02B8	696
153	0x02FE	766
155	0x0344	836
157	0x0389	905
159	0x03CF	975
161	0x0415	1045
163	0x045A	1114
165	0x04A0	1184

Table 13-3. Control Points Defined By Array (continued)

ByteIndex	Value	Description
167	0x04E6	1254
169	0x052B	1323
171	0x0571	1393
173	0x05B7	1463
175	0x05FC	1532
177	0x0642	1602
179	0x0687	1671
181	0x06CD	1741
183	0x0713	1811
185	0x0758	1880
187	0x079E	1950
189	0x07E4	2020
191	0x0829	2089
193	0x086F	2159

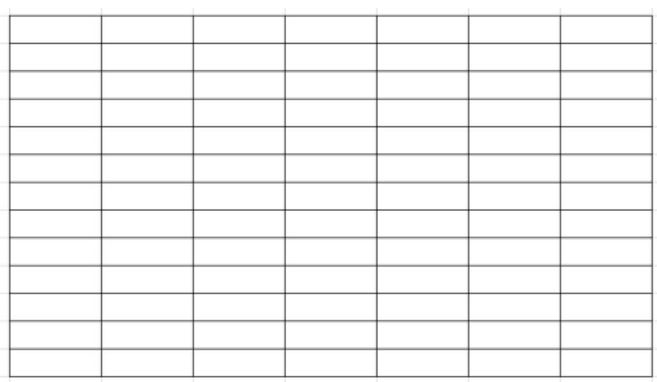
13.5.2 CMD_GetManualWarpControlPoints [Command ID: 0x35, Destination: 4]

This command reads the warping control points.

When to use control points defined by array

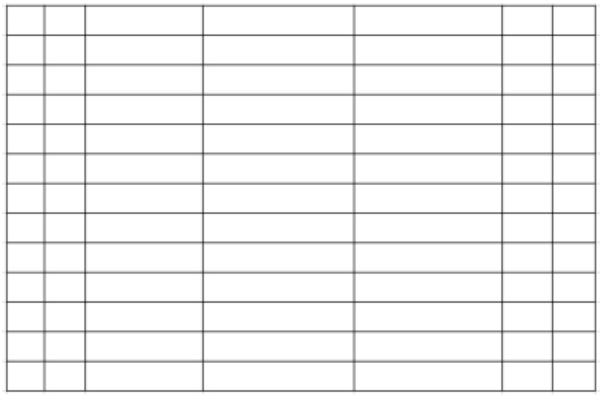
When control points defined by array = 0, the whole input rectangle to warp is divided into M×N equal parts where M×N are specified as number of rows and columns in the SetManualWarpControlPoints() command.

The number of warp table points must match with this M×N.



Instead of equally divided control points, to make some areas sparse and some areas dense, use SetManualWarpControlPoints() with control points defined by array = 1. Because the row and columns are straight lines at the input of warp, specify the maximum 64 horizontal points and maximum 32 vertical points. All other intersection points can be derived from those.

By using the array, input points to warping can be changed to something like the following example.



13.5.3 CMD_WriteManualWarpTable [Opcode: 0x34, Destination: 4]

This command loads the manual warp table into the system. It has the start index and an arbitrary number of points as parameters. The start index is a 16-bit number that represents the linear index into the 2D warp table. The warp table is always assumed have the same number of warp points as the control points set up using SetManualWarpControlPoints command.

Each point is passed as two 13.3 fixed point numbers that represents X and Y coordinates. Because the total command packet size cannot exceed 512 bytes, the table is loaded by invoking the command multiple times with different start indexes. If the total number of points passed in the manual warp table is less than 62×32 then software generates the warp map for all 62×32 control points from the warp map of input even spaced control points. For each 62×32 control points it finds the four closest control points in even spaced warp map and then using bilinear interpolation calculate the output position.

Note

Manual warp map is applied after any scaling and keystone correction performed by the warp engine.

Manual warp control points just be set before using this command.

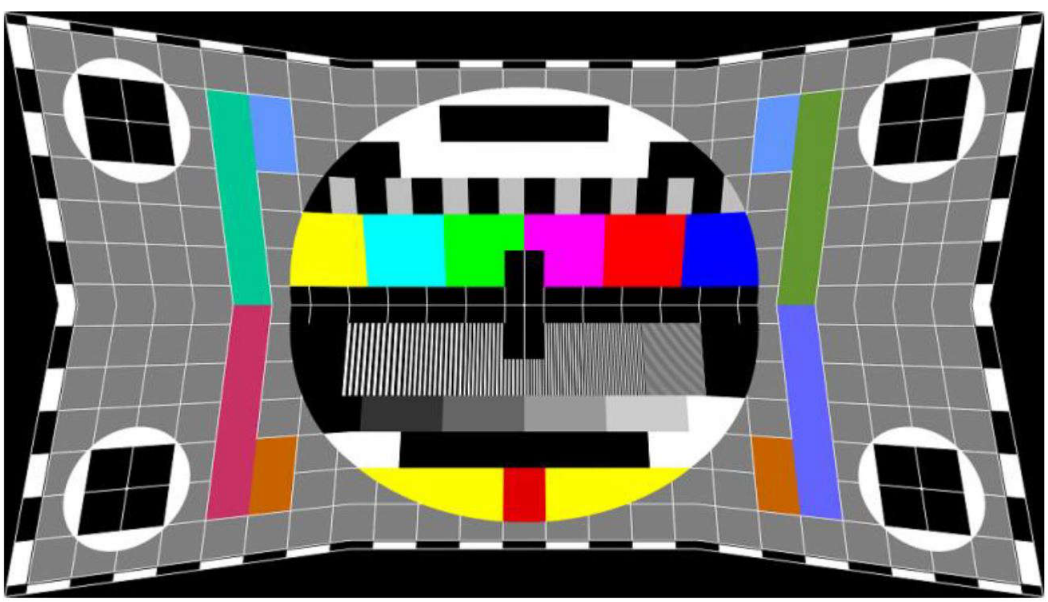


Figure 13-5. Example Output

Table 13-4. Example Command Packet

Byte Index	Value	Description
0	0x54	Command packet header (destination 4, length present, response required)
1	0x34	Command opcode

Table 13-4. Example Command Packet (continued)

Byte Index	Value	Description
2-3	0x0031	Number of bytes in payload (49)
4-5	0x0000	Start index in the table (0)
6-7	0x0000	X1 = 0.0
8-9	0x0000	Y1 = 0.0
10-11	0x2800	X2 = 1280.0
12-13	0x0640	Y2 = 200.0
14-15	0x5000	X3 = 2560.0
16-17	0x0640	Y3 = 200.0
18-19	0x77F8	X4 = 3839.0
20-21	0x0640	Y4 = 200.0
22-23	0x0000	X5 = 0.0
24-25	0x21C0	Y5 = 1080.0
26-27	0x2800	X6 = 1280.0
28-29	0x21C0	Y6 = 1080.0
30-31	0x5000	X7 = 2560.0
32-33	0x21C0	Y7 = 1080.0
34-35	0x71C0	X8 = 3640.0
36-37	0x21C0	Y8 = 1080.0
38-39	0x0000	X9 = 0.0
40-41	0x4378	Y9 = 2159.0
42-43	0x2800	X10 = 1280.0
44-45	0x3D40	Y10 = 1960.0
46-47	0x5000	X11 = 2560.0
48-49	0x3D40	X11 = 1960.0
50-51	0x77F8	X12 = 3839.0
51-52	0x4378	Y12 = 2159.0

Method for sending large number of warp points

As previously mentioned, the number of bytes in one command cannot exceed 512. To send a large number of warp points, send multiple packets. An example method for sending 62×32 warp points is shown here:

Command 1: index = 0, followed by 124 pairs of X, Y (124×2×2 = 496 bytes)

Command 2: index = 124, followed by the next 124 pairs of X, Y

Command 3: index = 248, followed by the next 124 pairs of X, Y

and so on through:

Command 16: index = 930, followed by the last 124 pairs of X, Y

Optimization to reduce the data sent by the command interface

Consider the case where out of 62×32 warp points, only some have changed (say the 10th, 11th, 12th and 45th point), then the following method can be used.

Command 1: index= 0, followed by (X10, Y10, X11, Y11, X12, Y12)

Command 2: index = 45, followed by (X45, Y45)

Command 3: ApplyManualWarping command with Enable = 0

Command 4" ApplyManualWarping command with Enable = 1

This optimization can be used instead of sending all of the 62×32 warp points when any point changes.

13.5.4 CMD_ReadManualWarpTable [Opcode: 0x34, Destination: 4]

This command reads the warp points.

13.5.5 CMD_ConfigureSmoothWarp [Command ID: 0x38, Destination: 4]

This command loads a manual warp table into the system. The edges connecting two warp points in this case are not straight lines but are ‘smoother’; that is, at the warp point, the edge is continuous and does not form a vertex (except for corners). This is done by fitting a 2nd degree polynomial curve (of the form of ax2 + bx + c) to warp points, contrary to the Write Manual Warp Table command which fits straight line to warp points.

Only a warp table with up to 5 rows and 5 columns (25 points) can be defined with this command.

Note

Manual warp map is applied on top of any scaling and keystone correction.

Manual warp control points command is NOT required before using this command since it always assumes uniform 3x3 map.

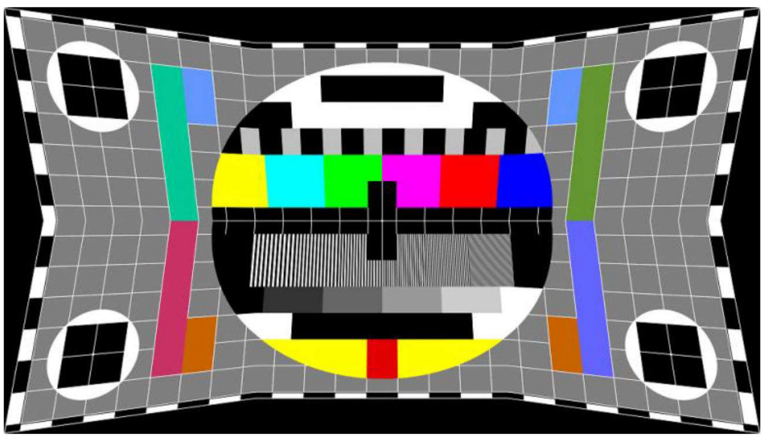


Figure 13-6. CMD_ConfigureSmoothWarp Example Output 1

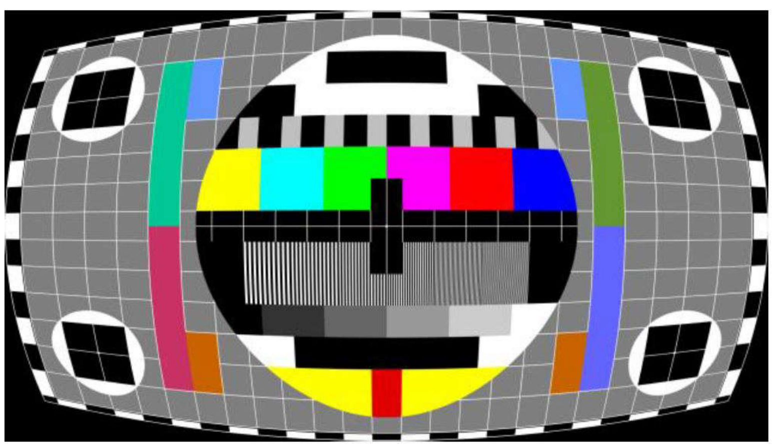


Figure 13-7. CMD_ConfigureSmoothWarp Example Output 2

Table 13-5. Example Command Packet

Byte Index	Value	Description
0	0x44	Command packet header (destination 4, response required)
1	0x38	Command opcode
3-4	0x01	X1 = 0.0
5-6	0x0000	Y1 = 0.0

Table 13-5. Example Command Packet (continued)

Byte Index	Value	Description
7-8	0x3C00	X2 = 1920.0
9-10	0x0640	Y2 = 200.0
11-12	0x77F8	X3 = 3839.0
13-24	0x0000	Y3 = 0.0
15-16	0x0640	X4 = 200.0
17-18	0x21C0	Y4 = 1080.0
19-20	0x3C00	X5 = 1920.0
21-22	0x21C0	Y5 = 1080.0
23-24	0x71C0	X6 = 3640.0
25-26	0x21C0	Y6 = 1080.0
27-28	0x0000	X7 = 0.0
29-30	0x4378	Y7 = 2159.0
31-32	0x3C00	X8 = 1920.0
33-34	0x3D40	Y8 = 1960.0
35-36	0x77F8	X9 = 3839.0
37-38	0x4378	Y9 = 2159.0

13.5.6 CMD_ApplyManualWarping [Command ID: 0x36, Destination: 4]

This command is used to enable or disable manual warping feature in the system.

Note

A manual warp map must exist in the system before using this command to enable the warping.

Table 13-6. Example CMD_ApplyManualWarping Command Packet

Byte Index	Value	Description
0	0x44	Command packet header (destination 4, response required)
1	0x36	Command opcode

13.6 Optical (Lens) Distortion Correction

The manual warp table access commands described in the previous sections are typically used for surface correction that needs to be adjusted depending on the projection surface conditions.

In addition to that, some of the lens related distortions like geometric distortion and lateral chromatic distortion can also be corrected by using the warp map. The warp map for such correction can be determined one time at the factory based on calibration. That map may then be stored in an EEPROM or flash in the system and then merged with the surface correction warp table using the “merge” option of Set Manual Warp Table Update Mode command.

In the TI reference application, TI has made provision to store a lens correction map in the EEPROM (see Figure 13-8).

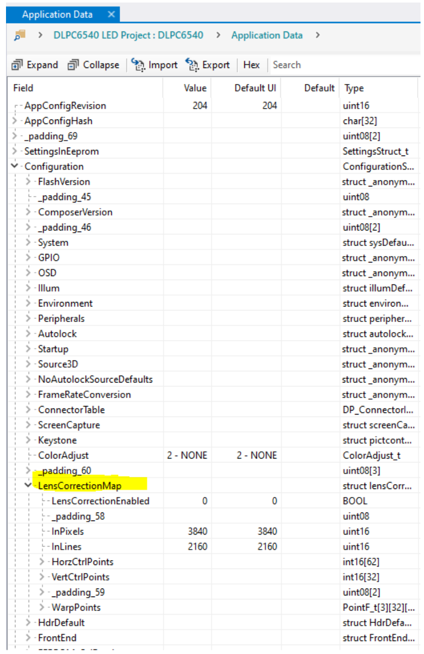


Figure 13-8. Storage of Lens Correction Map

14 Introduction to Blending

The Edge Blending Function (EBF) block supports spatially variable gain and offset for each color channel in linear light space. This section describes the APIs and commands that are provided for the customer to use this EBF block and implement a custom blend map application.

Manual blending can be used to achieve seamless overlap of two or more projectors, to increase the projected image area, and improve on-screen resolution and brightness. The EBF block compensates for color differences across different regions on the projected image due to light fall-off or surface material color, or to provide uniform color and brightness when overlapping two or more projectors.

Figure 14-1 shows that the gain value can be modified on the right edge of left projector and left edge of right projector to achieve a seamless final image.

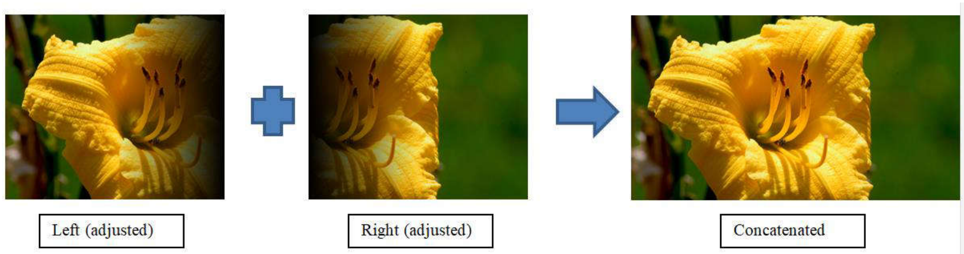


Figure 14-1. Example of Blending

Consider a monochromatic pixel *pin*, the EBF block can modify the intensity of this pixel by applying a gain and an offset:

pout = *pin* + GAIN + OFFSET

14.1 Blend Map Control Points

The application must provide 63 horizontal and 32 vertical locations of control points. The control points must lie within the resolution of the image (that is, 0 to Hres–1 for horizontal control points and 0 to Vres–1 for vertical control points, where Hres and Vres are the horizontal and vertical resolution of image, respectively). This results in 2016 control points on the image, and the gain and offset values must be declared for these for each of the color channels R, G, and B.

14.2 Blend Map Gain Values

This gain is multiplied with the pixel intensity at that sampling pixel. Gain must be applied for each of the RGB channel of each sampling pixel. Gain value lies between 0 and 1.999 (precision: u1.12). The gain value must be in the fixed point format of u1.12.

14.3 Blend Map Offset Value

The offset is added to each sampled pixel. This is a signed quantity. Permitted range for offset values is -255 to +255. Convert the value to internal floating point of s1m8e4 before passing it to the command. A matlab script to convert standard floating-point value to the format required by the command can be downloaded from http://software-dl.ti.com/secure/software/dlp_display/General/manual_blending_helpers.zip.

14.4 Constraints

63 horizontal and 32 vertical control points define a 63×32 map on the input image. Half the control points must be in the left half of image and half in the right half of image. From a total of 2016 control points, 1008 control points need to be in each half of the image when the image is divided vertically in two halves from the middle of the horizontal resolution.

While declaring the horizontal control point locations, make sure to have the 31st control point at the center of the image horizontally.

14.5 Manual Blending Commands

The following manual blending commands can be used to configure the EBF block of the DLPCx54x controller.

14.5.1 CMD_EnableEdgeBlending [Command ID: 0x2F]

This command enables or disables the EBF block. This can also be used to get the current state of EBF block.

Table 14-1. Example CMD_EnableEdgeBlending Command Packet

Byte Index	Value	Description
0	0x44	Command packet header (destination 4, response required)
1	0x2F	Command opcode
2	0x01	Enable edge blending

14.5.2 CMD_SetBlendMapControlPoints [Opcode: 0x2E]

This command is used to pass the Horizontal (Hres) and Vertical (Vres) resolution of image and also the Horizontal and vertical control point's location.

The values of the control point locations must in the range of 0 to (Hres – 1) for the horizontal control points and 0 to (Vres – 1) for the vertical control points.

The following example shows how this command can be used for an image of resolution 1280×1080 and the control points are evenly spaced.

Ideally, more control points are needed in the region where gain or offset change is required, but keep in mind the constraints mentioned before.

Note

The 31st horizontal control point location is at the center of image horizontally.

Half of control point locations are in left half of image and other half in right half image.

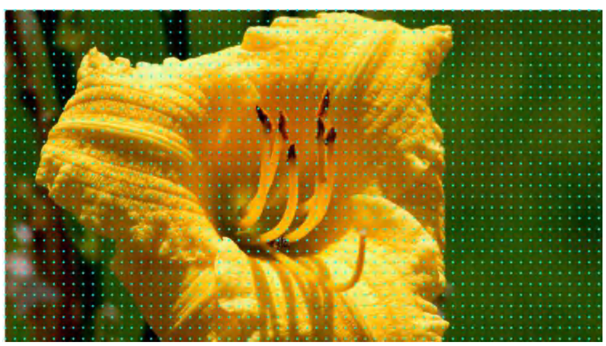


Table 14-2. Example CMD_SetBlendMapControlPoints Command Packet

Byte Index	Value	Description
0	0x54	Command packet header (destination 4, response required)
1	OX2E	Command opcode
2-3	1280	Horizontal image resolution
4-5	1080	Vertical image resolution
Horizontal Control Points		
6-7	0x0000	Horizontal control points location start: H0 = 0
8-9	0x0016	H1 = 22
10-11	0x002A	H2 = 42
12-13	0x003F	H3 = 63
14-15	0x0054	H4 = 84
16-17	0x0068	H5 = 104

Table 14-2. Example CMD_SetBlendMapControlPoints Command Packet (continued)

Byte Index	Value	Description
18-19	0x007D	H6 = 125
20-21	0x0091	H7 = 145
22-23	0x00A6	H8 = 166
24-25	OX00BB	H9 = 187
26-27	0xO0CF	H10 = 207
28-29	OX00E4	H11 = 228
30-31	0x00F9	H12 = 248
32-33	0x010D	H13 = 269
34-35	0x0122	H14 = 290
36-37	0x0136	H15 = 310
38-39	0x014B	H16 = 331
40-41	0x0160	H17 = 352
42-43	0x0174	H18 = 372
44-45	0x0189	H19 = 393
46-47	0x019E	H20 = 414
48-49	0x01B2	H21 = 434
50-51	0x01C7	H22 = 455
51-52	0x01DB	H23 = 475
53-54	0x01F0	H24 = 496
55-56	0x0205	H25 = 517
57-58	0x0219	H26 = 537
59-60	0x022E	H27 = 558
61-62	0x0243	H28 = 579
63-64	0x0257	H29 = 599
65-66	0x0280	H30 = 640
67-68	0x0281	H31 = 641
69-70	0x0295	H32 = 661
71-72	0x02AA	H33 = 681
73-74	0x02BE	H34 = 682
75-76	0x02D3	H35 = 702
77-78	0x02E8	H36 = 723
79-80	0x02FC	H37 = 744
81-82	0x0311	H38 = 764
83-84	0x0326	H39 = 785
85-86	0x033A	H40 = 806
87-88	0x034F	H41 = 826
89-90	0x0363	H42 = 847
91-92	0x0378	H43 = 867
93-94	0x038D	H44 = 888
95-96	0x03A1	H45 = 909
97-98	0x03B6	H46 = 929
99-100	0x03CB	H47 = 950
101-102	0x03DF	H48 = 971
103-104	0x03FF	H49 = 991
105-106	0x0401	H50 = 1023
107-108	0x041D	H51 = 1053
109110	0x0432	H52 = 1074

Table 14-2. Example CMD_SetBlendMapControlPoints Command Packet (continued)

Byte Index	Value	Description
111-112	0x0446	H53 = 1094
113-114	0x045B	H54 = 1115
115-116	0x0470	H55 = 1136
117118	0x0484	H56 = 1156
119120	0x0499	H57 = 1177
121-122	0x04AD	H58 = 1197
123-124	0x04C2	H59 = 1218
125-126	0x04D7	H60 = 1239
127-128	0x04EB	H61 = 1259
129-130	0x04FF	H62 = 1279
Vertical Control Points		
131-132	0x0000	Vertical control points location start: V0 = 0
133-134	0x0024	V2 = 36
135-136	0x0047	V3 = 71
137-138	0x0069	V4 = 105
139-140	0x008C	V5 = 140
141-142	0x00AF	V6 = 175
143-144	0xO0D2	V7 = 210
145-146	0x00F5	V8 = 245
147-148	0x0117	V9 = 279
149-150	0x013A	V10 = 314
151-152	0x015D	V11 = 349
153-154	0x0180	V12 = 384
155-156	0x01A3	V13 = 419
157-158	0x01C5	V14 = 453
159-160	0x01E8	V15 = 488
161-162	0x020B	V16 = 523
163-164	0x022E	V17 = 588
165-166	0x0251	V18 = 593
167-168	0x0274	V19 = 628
169-170	0x0296	V20 = 662
171-172	0x02B9	V21 = 697
173-174	0x02DC	V22 = 732
175-176	0x02FF	V23 = 767
177-178	0x0322	V24 = 802
179-180	0x0344	V25 = 836
181-182	0x0367	V26 = 871
183-184	0x038A	V27 = 906
185-185	0x03AD	V28 = 941
187-186	0x03D0	V29 = 976
189-188	0x03F2	V30 = 1010
191-192	0x0415	V31 = 1045
193-194	0x0437	V31 = 1079

14.5.3 CMD_ GetBlendMapControlPoints [Command ID: 0x2E]

This command can be used to 'get' the value of control points location set by the CMD_SetBlendMapControlPoints command.

Example:

Byte Index	Value	Description
0	0x44	Command packet header (destination 4, response required)
1	OX2E	Command opcode

14.5.4 CMD_SetBlendMapGainValues [Command ID: 0x2B]

This command sets the gain values for the 2016 (63×32) control points.

The gain values must be specified for each of the color channels.

Gain values must be between 0 and 1.999 (precision: u1.12). The the gain value must be in the fixed point format of u1.12.

Example: Suppose that a user wants to set the gain for a particular control point as 1.25. To convert to u1.12 format, the required value needs to be multiplied by 4096 to convert it to required format before passing it to this command. In u1.12, 1.25 is 5120.

There are two features here for using the command. User can select the broadcast values option from the Color Channel Select. What broadcast does is that user needs to pass values only once and it will broadcasted to all the three color channels R,G,B.

Otherwise user needs to pass value for R, G, B color channels separately. So the command will then be required to use 3 times one time for each color channel.

Another feature supported is passing compressed values. The compression used is RLE2 compression. For passing compressed values user needs to select the checkbox of Compression Enabled.

Example: Suppose that a user passes gain values as uncompressed, then user would need to use this command multiple times because of the restrictions (see the note below).

User has the following options for passing gain values using the command,

1. Pass Compressed values (RLE2) and broadcast it to all color channels
2. Pass Uncompressed values and broadcast it to all color channels
3. Pass separate Compressed values for each of color channel
4. Pass separate Uncompressed values for each of color channel

---

Note

The maximum number of bytes that can be passed at a time is 502 bytes and user needs to make sure not to exceed this. Suppose user is using the option of compress disabled then user will have to pass 2016 values to command. Because all values cannot be passed at the same time user must use the same command multiple times, and in each time update the Index value from which user wants to pass value.

The following table is an example of the values to pass to this command. The compression enabled and broadcast enabled options are used. After compression, the 2016 values of gain for a single color channel reduces to 64 in this example.

Table 14-3. Example CMD_ SetBlendMapGainValues Command Packet

Byte Index	Value	Description
0	0x54	Command packet header (destination 4, length present, response required)
1	0x2B	Command opcode
2	0x01	Compression enabled: the command can accept compressed values
		now

Table 14-3. Example CMD_ SetBlendMapGainValues Command Packet (continued)

Byte Index	Value	Description
3	0x00	Broadcast selected in color channel select
4-5	0x0040	Total number of compressed values = 64 (for this example)
6-7	0x0000	Start Index = 0
Compressed gain value		
8-9	0x003F	G0 = 63
10-11	0x1000	G1 = 4096
12-13	0x0000	G2 = 0
14-15	0x003F	G3 = 63
16-17	0x0000	G4 = 0
18-19	0x003F	G5 = 63
20-21	0x0000	G6 = 0
22-23	0x003F	G7 = 63
24-25	0x0000	G8 = 0
26-27	0x003F	G9 = 63
28-29	0x0000	G10 = 0
30-31	0x003F	G11 = 63
32-33	0x0000	G12 = 0
34-35	0x003F	G13 = 63
36-37	0x0000	G14 = 0
38-39	0x003F	G15 = 63
40-41	0x0000	G16 = 0
42-43	0x003F	G17 = 63
44-45	0x0000	G18 = 0
46-47	0x003F	G19 = 63
48-49	0x0000	G20 = 0
50-51	0x003F	G21 = 63
51-52	0x0000	G22 = = 0
53-54	0x003F	G23 = 63
55-56	0x0000	G24 = 0
57-58	0x003F	G25 = 63
59-60	0x0000	G26 = 0
61-62	0x003F	G27 = 63
63-64	0x0000	G28 = 0
65-66	0x003F	G29 = 63
67-68	0x0000	G30 = 0
69-70	0x003F	G31 = 63
71-72	0x003F	G32 = 63
73-74	0xOF7F	G33 = 3967
75-76	0x003F	G34 = 63
77-78	0xOE75	G35 -3701
79-80	0x003F	G36 = 63
81-82	0x0D6C	G37 = 3436
83-84	0x003F	G38 = 63
85-86	0x0C6A	G39 = 3178
87-88	0x003F	G40 = 63
89-90	0x0B60	G41 = 2912
91-92	0x003F	G42 = 63

Table 14-3. Example CMD_ SetBlendMapGainValues Command Packet (continued)

Byte Index	Value	Description
93-94	0x0A57	G43 = 2647
95-96	0x003F	G44 = 63
97-98	0x094D	G43 = 2381
99-100	0x003F	G4663
101-102	0x0844	G47 = 2116
103-104	0x003F	G48 = 63
105-106	0x0742	G49 = 1858
107-108	0x003F	G50 = 63
109-110	0x0638	G51 = 1592
111-112	0x003F	G52 = 63
113-114	0x052F	G53 = 1327
115116	0x003F	G54 = 63
117118	0x0425	G55 = 1061
119120	0x003F	G56 = 63
121-122	0x031C	G57 = 796
123-124	0x003F	G58 = 63
125-126	0x021A	G59 = 538
127-128	0x003F	G60 = 63
129130	0x0111	G61 = 273
131-132	0x003F	G62 = 63
133-134	0x0007	G63 = 7

14.5.5 CMD_ GetBlendMapGainValues [Command ID: 0x2B]

This command is used to get the gain values set by CMD_ SetBlendMapGainValues. Select the same color channel select option that was used in CMD_ SetBlendMapGainValues and pass the index and number of values to get.

If compressed values were set, then this command returns compressed values. If uncompressed values were set, this command returns uncompressed values.

Table 14-4. Example CMD_ GetBlendMapGainValues Command Packet

Byte Index	Value	Description
0	0x54	Command packet header (destination 4 ,length present, response required)
1	0x2B	Command opcode
2	0x00	Broadcast selected in color channel select
3-4	0x0000	Start index = 0
5-6	0x0040	Number of values to be read. This example reads 64 values.

14.5.6 CMD_ SetBlendMapOffsetValues [Command ID: 0x2D]

This command sets the offset values for the 2016 (63×32) control points.

The offset values must be specified for each color channel, R, G, and B.

Note

The maximum number of bytes that can be passed in one command is 502 bytes.

The range of offset values in the standard floating pt format is -255 to +255. Provide the offset values to the command in the internal floating point format of s1m8e4. A matlab script to convert standard floating-point value to the format required by the command can be downloaded from https://software-dl.ti.com/secure/software/dlp_display/General/manual_blending_helpers.zip

This code takes the input from user in the standard floating point format in the range of -255 to +255 and converts it to the internal floating point value required by this command.

Two options exists in this command:

Broadcast values: Broadcast sets the same offset values for all three color channels (R,G,B).

Compressed values: The compression used is RLE2 compression. To pass compressed values, select Compression Enabled. A sample RLE2 compression algorithm can be downloaded from https://software-dl.ti.com/secure/software/dlp_display/General/manual_blending_helpers.zip.

The following options for passing gain values are used by this command:

- Pass compressed values (RLE2) and broadcast to all color channels
- Pass uncompressed values and broadcast to all color channels
- Pass separate compressed values for each of color channel
- Pass separate uncompressed values for each of color channel

This command needs to be used in a similar way as to CMD_ SetBlendMapGainValues.

14.5.7 CMD_ GetBlendMapOffsetValues [Command ID: 0x2D]

This command gets the offset values. Select the same color channel selection used in CMD_SetBlendMapOffsetValues, the offset values, and pass the index and number of values to get.

This command returns compressed or uncompressed values depending on what was set with CMD_ SetBlendMapOffsetValues.

This command is similar to CMD_GetBlendMapGainValues.

14.5.8 CMD_ApplyBlendMap [Command ID: 0x2C]

This command applies the blend map to the EBF block. This command enables passing the values of the blend map to the API.

Table 14-5. Example CMD_ApplyBlendMap Command Packet

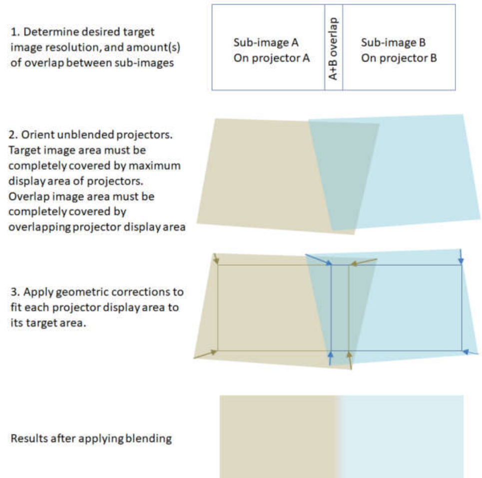
Byte Index	Value	Description
0	0x44	Command packet header (destination 4, response required)
1	0x2C	Command opcode
2	0x01	Enable edge blending

14.6 Manual Blending Application Commands

There are two modes of blending available through manual blending application commands: 2D Luminance, and Simple 1D RGB.

Simple 1D RGB mode is designed to support simple blending use cases where no warping is required. This implies that the blended projector display areas must be overlapped orthogonally. Any warping causes visual artifacts. Gains and offsets are manually configured visually by the user for each projector to match black and white levels between projectors. This mode supports controlling the RGB gain for matching white color and luminance between projectors. The gain automatically is linearly ramped from the full value at the beginning of the overlapping region to zero at the end to create a smooth transition between projectors. Also supported are RGB offsets for the non-overlapping region, overlapping regions, and the regions when the POM from the adjacent projectors overlap. The overlap amount is specified by the user, and POM width is automatically detected through the hardware configuration.

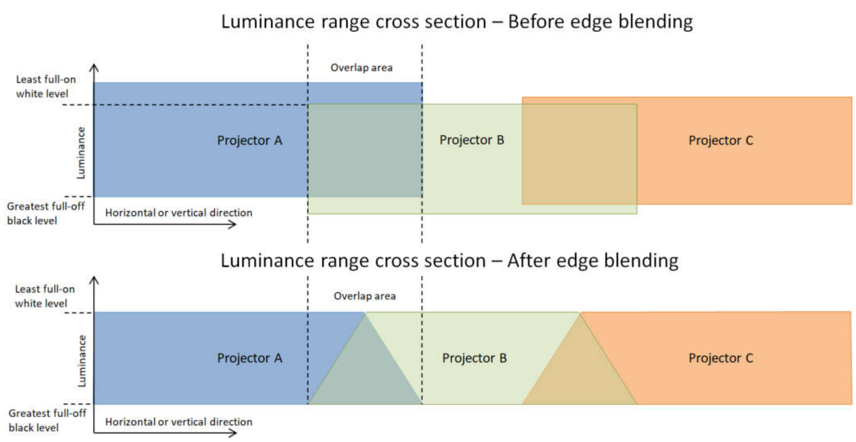
2D Luminance blending supports the blending of a rectangular grid of N×M projectors up to 27 in total. This mode allows for warping to be used with blending as follows (illustration of two projector case):



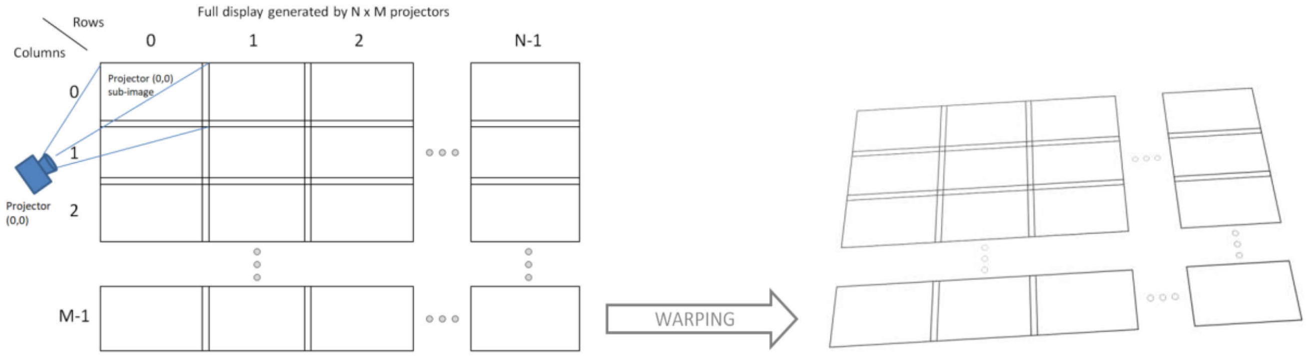
For Simple 1D RGB mode, warping is not allowed. Therefore, the projectors must be overlapped and orthogonally aligned. The overlaps between all projectors in the horizontal (X) direction and vertical direction (Y) are configurable. The gain and offset values for the blending map for any given projector within the blending system is directly determined by the user for Simple 1D RGB mode. For 2D Luminance mode, gain and offset are calculated from:

- The full-on white levels of all projectors in the blending system
- The full-off black levels of all projectors in the blending system
- The number of pixels of vertical and horizontal overlaps between adjacent projectors
- Geometric correction information

The gain and offset values in the blending map makes sure that there is a single luminance range across all projectors, and that there is a ramped transition from one projector to another within the overlapping area.



Geometric correction is available via keystone corners only, which has the implication that the warping can only accommodate planar surfaces.



An example Python script for 2D Luminance mode is given illustrating a 3×3 blended projector system. The script is usable within DLP control program:

```
from dlpc754x.commands import *
from System.Threading import *
#Enable edge blending
WriteEnableEdgeBlending(1)
#Enable manual warping
WriteApplyManualWarping(1)
#Assign number of projectors in system
NumProjectorsColumns = 3
NumProjectorsRows = 3
#Assign index of self projector
SelfProjectorColumn = 1
SelfProjectorRow = 1
#Assign white and black luminance levels for all projectors in blending system
#These are specified in white_level/black_level pairs, in order of the projectors from upper left
to bottom right
#This means size of the array must be 2*NumProjectorsColumns*NumProjectorsRows
WhiteBlackLevels = [
999,1, 1000,1, 999,1,
1000,1, 999,1, 1000,1,
999,1, 1000,1, 1000,1
#Write blending system parameters
print WriteEdgeBlendingSystemParams(NumProjectorsColumns, NumProjectorsRows, SelfProjectorColumn,
SelfProjectorRow, WhiteBlackLevels)
#Assign horizontal and vertical overlaps
OverlapHorizontal = 500
OverlapVertical = 200
#Assign geometric adjustment type to keystone corners
GeometericAdjustmentType = DispEdgeBlendGeometryModeT.KeystoneCorners
#Assign coordinates to keystone corners (unwarped default for 3840x2160)
KeystoneTopLeftX = 0
KeystoneTopLeftY = 0
KeystoneTopRightX = 3839
KeystoneTopRightY = 0
KeystoneBottomLeftX = 0
KeystoneBottomLeftY = 2159
KeystoneBottomRightX = 3839
KeystoneBottomRightY = 2159
GeometryParams = [KeystoneTopLeftX, KeystoneTopLeftY, KeystoneTopRightX, KeystoneTopRightY,
KeystoneBottomLeftX, KeystoneBottomLeftY, KeystoneBottomRightX, KeystoneBottomRightY]
#Write edge blending configuration
print WriteEdgeBlendingConfiguration(OverlapHorizontal, OverlapVertical, GeometericAdjustmentType,
GeometryParams)
```

14.6.1 CMD_SetEdgeBlendingSystemParams [Command ID: 0x3D]

This command sets the blending system parameters for manual edge blending. Information required includes the total number of rows and columns of projectors that are being blended together into create a single display, and the position of the self projector within the display. Only rectangular arrays of projectors are supported. The acceptable range of values for number of rows or columns of projectors is from 1-32.The acceptable range of indexes for row or column of the self projector is from 0-31. For Simple 1D RGB mode, the gain parameters specified for each projector determine the gain of the non-overlapped area, and the scale of the linear ramp from the gain value to 0. Offsets are specified for active overlapping areas (1 or 2), overlapping POM areas(1 or 2), and the non-overlapped area. This makes the total number of unique regions to be adjusted either 3 or 5, depending on whether the projector being configured has one adjacent projector or two. So for any size 1D system, projector having 2 adjacent projectors ("interior" projector) requires 5 sets of offset values. Projectors having 1 adjacent projector ("exterior" projector) requires 3 sets of offset value. The gains and offsets specified have red, green and blue (RGB) components, which means 3 different values are required to specify each. This makes the total number of photometric parameters required always either 12 or 18 (3 Gain + 9 offset or 3 Gain +15 offset). The order of the offset values (0…n) in the parameter set are specified from left to right, or top to bottom (Starting from side of screen with pixel 0,0). Gains and offsets are values from [0 to 1], and are in 10:22 fixed point format.

Exterior projector parameters: {Gain_R, Gain_G, Gain_B, Offset_0_R, Offset_0_G, Offset_0_B, Offset_1_R, Offset_1_G, Offset_1_B, Offset_2_R, Offset_2_G, Offset_2_B}

Interior projector parameters: {Gain_R, Gain_G, Gain_B, Offset_0_R, Offset_0_G, Offset_0_B, Offset_1_R, Offset_1_G, Offset_1_B, Offset_2_R, Offset_2_G, Offset_2_B, Offset_3_R, Offset_3_G, Offset_3_B, Offset_4_R, Offset_4_G, Offset_4_B}

Gains should begin at 1 by default, and be lowered by the user to balance white levels. Offsets should begin at 0 by default, and be raised to balance black levels. Because the total brightness offset of the overlapped areas is the combination of the overlapping projectors, each projector can contribute more or less offset to this area and achieve the same result. For 2D luminance mode, the luminance values (in nits) for white and black levels for all projectors must be provided to ensure a uniform illumination across the display. This command does not change the state of the warping map or the blending map, but does make changes to the parameters which are being used to perform the blending and warping map creation by CMD_SetEdgeBlendingConfiguration. This command does not write to non-volatile storage. To write the configuration to non-volatile storage, a call must be made to CMD_SetEdgeBlendingConfiguration after running this command.

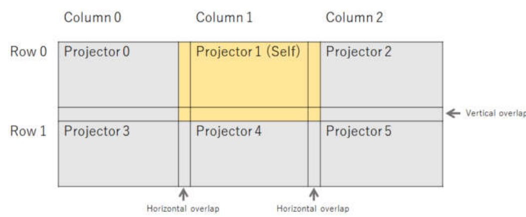


Figure 14-2. Example Blending System Diagram

Table 14-6. Example Command Packet for 2D

Byte Index	Value	Description
0	0x54	Command packet header (destination 4, length present, response required)
1	0x3D	Command opcode
2-3	0x0003	Number of columns of projectors in blending system = 3
4-5	0x0002	Number of rows of projectors in blending system = 2
6-7	0x0001	Column index of self in blending system = 1
8-9	0x0000	Row index of self in blending system = 0
10	0x00	2D Luminance mode
11-14	0x03EF	White luminance level of projector 0 in blending system
15-18	0x0003	Black luminance level of projector 0 in blending system
19-22	0xO3EO	White luminance level of projector 1 in blending system
23-26	0x0009	Black luminance level of projector 1 in blending system
27-30	0x03E3	White luminance level of projector 2 in blending system
31-34	0x0004	Black luminance level of projector 2 in blending system
35-38	0x03EF	White luminance level of projector 3 in blending system
39-42	0x0003	Black luminance level of projector 3 in blending system
43-46	0xO3EO	White luminance level of projector 4 in blending system
47-50	0x0009	Black luminance level of projector 4 in blending system

Table 14-6. Example Command Packet for 2D (continued)

Byte Index	Value	Description
51-54	0x03E3	White luminance level of projector 5 in blending system
55-58	0x0004	Black luminance level of projector 5 in blending system

14.6.2 CMD_GetEdgeBlendingSystemParams [Command ID: 0x3D]

This command gets the blending system parameters for semi-manual edge blending. See CMD_SetEdgeBlendingSystemParams for description of the parameters. This command does not read from non-volatile storage. To read the configuration from non-volatile storage, a call must be made to CMD_GetEdgeBlendingConfiguration before running this command.

14.6.3 CMD_SetEdgeBlendingConfiguration [Command ID: 0x3E]

This command sets projector overlap and geometric correction parameters for manual edge blending. It creates and applies blending and warping maps for given blending inputs. It is necessary to separately call commands to enable manual warping and enable edge blending for the results to take effect.

Overlaps with adjacent projectors must be specified in pixels. There is a single value for horizontal overlap, and a single value for vertical overlap. This means overlaps between all projectors for a given direction (horizontal or vertical) must be identical. Overlaps set to zero cause no blending to occur. Temporarily keeping the overlaps to zero can be useful for initial geometric adjustments.

Storage Options provides the capability to store the configuration to nonvolatile storage, and to enable application of the stored blending configuration at startup. The stored information includes that provided through both CMD_SetEdgeBlendingConfiguration and CMD_SetEdgeBlendinSystemParams. Changes to Storage Options can be made using any valid blending configuration, including the trivial case of no blending: 1x1 system, zero overlaps, no geometric correction.

Geometry parameters are dependent on geometric adjustment type. The options for geometric adjustment type are keystone corners or no geometric correction. For no geometric correction, no additional parameters are required. When using Simple 1D RGB mode, no geometric correction should be used.

Byte Index	Value	Description
0	0x54	Command packet header (destination 4, length present, response required)
1	Ox3E	Command opcode
2-3	0x0050	Horizontal overlap with other projectors =80 pixels
4-5	0x0040	Vertical overlap with other projectors = 64 pixels
6	0x01	Geometric adjustment type: 1 = Keystone corners
7	0x02	Storage Options: 2 = Store in nonvolatile memory and apply at startup
8-9	0x0064	Projector 0: Top left corner X coordinate = 100
10-11	0x006F	Projector 0: Top left corner Y coordinate = 111
12-13	0x0EE2	Projector 0: Top right corner X coordinate = 3810
14-15	0x0001	Projector 0: Top right corner Y coordinate = 1
16-17	0x000A	Projector 0: Bottom left corner X coordinate = 10
18-19	0x0870	Projector 0: Bottom left corner Y coordinate = 2160
20-21	0x0F00	Projector 0: Bottom right corner X coordinate = 3840
22-23	0x0834	Projector 0: Bottom right corner Y coordinate = 2100
24-25	0x0064	Projector 1: Top left corner X coordinate = 100
26-27	0x006F	Projector 1: Top left corner Y coordinate = 111
28-29	0x0EE2	Projector 1: Top right corner X coordinate = 3810
30-31	0x0001	Projector 1: Top right corner Y coordinate = 1
32-33	0x000A	Projector 1: Bottom left corner X coordinate = 10
4-35	0x0870	Projector 1: Bottom left corner Y coordinate = 2160
36-37	0x0F00	Projector 1: Bottom right corner X coordinate = 3840
38-39	0x0834	Projector 1: Bottom right corner Y coordinate = 2100

Byte Index	Value	Description
40-41	0x0064	Projector 2: Top left corner X coordinate = 100
42-43	0x006F	Projector 2: Top left corner Y coordinate = 111
44-45	0x0EE2	Projector 2: Top right corner X coordinate = 3810
46-47	0x0001	Projector 2: Top right corner Y coordinate = 1
48-49	0x000A	Projector 2: Bottom left corner X coordinate = 10
50-51	0x0870	Projector 2: Bottom left corner Y coordinate = 2160
52-53	0x0F00	Projector 2: Bottom right corner X coordinate = 3840
54-55	0x0834	Projector 2: Bottom right corner Y coordinate = 2100
56-57	0x0064	Projector 3: Top left corner X coordinate = 100
58-59	0x006F	Projector 3: Top left corner Y coordinate = 111
60-61	0x0EE2	Projector 3: Top right corner X coordinate = 3810
62-63	0x0001	Projector 3: Top right corner Y coordinate = 1
64-65	0x000A	Projector 3: Bottom left corner X coordinate = 10
66-67	0x0870	Projector 3: Bottom left corner Y coordinate = 2160
68-69	0x0F00	Projector 3: Bottom right corner X coordinate = 3840
70-71	0x0834	Projector 3: Bottom right corner Y coordinate = 2100
72-73	0x0064	Projector 4: Top left corner X coordinate = 100
74-75	0x006F	Projector 4: Top left corner Y coordinate = 111
76-77	0x0EE2	Projector 4: Top right corner X coordinate = 3810
78-79	0x0001	Projector 4: Top right corner Y coordinate = 1
80-81	0x000A	Projector 4: Bottom left corner X coordinate = 10
82-83	0x0870	Projector 4: Bottom left corner Y coordinate = 2160
84-85	0x0F00	Projector 4: Bottom right corner X coordinate = 3840
86-87	0x0834	Projector 4: Bottom right corner Y coordinate = 2100
88-89	0x0064	Projector 5: Top left corner X coordinate = 100
90-91	0x006F	Projector 5: Top left corner Y coordinate = 111
92-93	0x0EE2	Projector 5: Top right corner X coordinate = 3810
94-95	0x0001	Projector 5: Top right corner Y coordinate = 1
96-97	0x000A	Projector 5: Bottom left corner X coordinate = 10
98-99	0x0870	Projector 5: Bottom left corner Y coordinate = 2160
100-101	0x0F00	Projector 5: Bottom right corner X coordinate = 3840
102-103	0x0834	Projector 5: Bottom right corner Y coordinate = 2100

14.6.4 CMD_GetEdgeBlendingConfiguration [Command ID: 0x3E]

This command gets overlap and geometry parameters for semi-manual edge blending. See CMD_SetEdgeBlendingConfiguration for description of the parameters.

14.7 Cropping of Input Image for Blending Setup

Based on the minimum and maximum overlap (blending area), display resolution, and layout size, calculate the required input image sizes that must be cropped before sending to each projector:

Projector display resolution (W × H)

- 3840 × 2160
- 1920 × 1080

Overlap: minimum 10% and maximum 45% for display width and height

Projector layout (M × N)

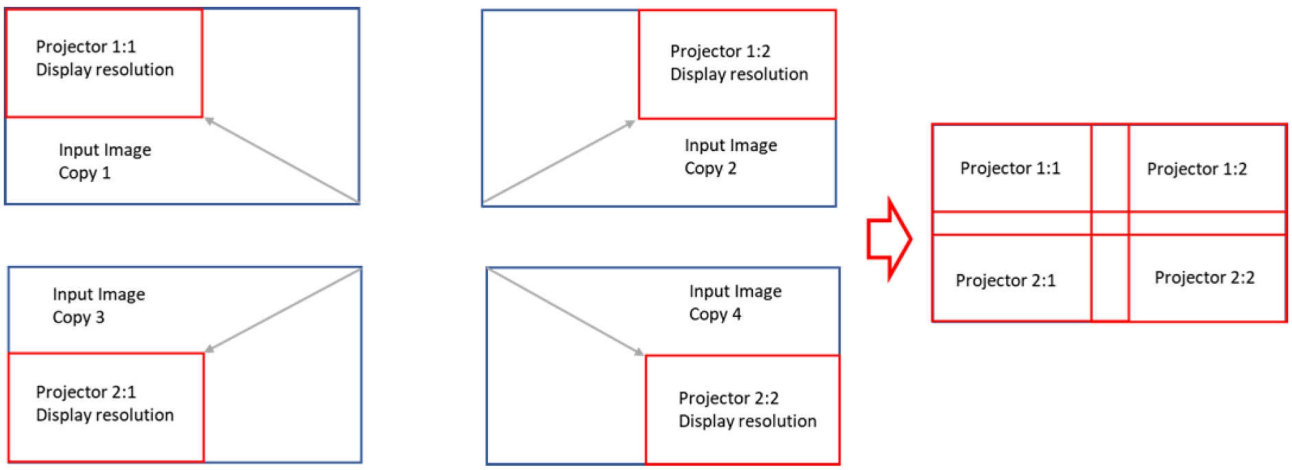
- Up to 5 × 5

Input image size (desired image to be displayed by all blended projectors blended):

- Calculate input image width:
  – Max width: (W × M) – (10% of W) × (M – 1)
  – Min width: (W × M) – (45% of W) × (M – 1)
- Calculate input image height
  – Max height: (H × N) – (10% of H) × (N – 1)
  – Min height: (H × N) – (45% of H) × (N – 1)

Crop the input image for each projector in the blending layout. The following example is for a 2 × 2 layout:

- Make 4 copies of the input (original) image
- Crop each copy to the projector display
- Input each cropped copy into the appropriate projector in the blending layout size



Example:

W × H = 3840 × 2160 and M × N = 2 × 2 layout and 10% overlap (blending area)

– Overlap calculation:
  - Width overlap: 10% (384)
  - Height overlap: 10% (216)

– Input image size calculation
  - Max width (3840 × 2) – (10% of 3840) × (2 – 1) = 7680 – 384 × 1 = 7296
  - Max height (2160 × 2) – (10% of 3840) × (2 – 1) = 4320 – 216 × 1 = 4104

– Crop the input image for each projector in the layout

The resulting layout is as follows:

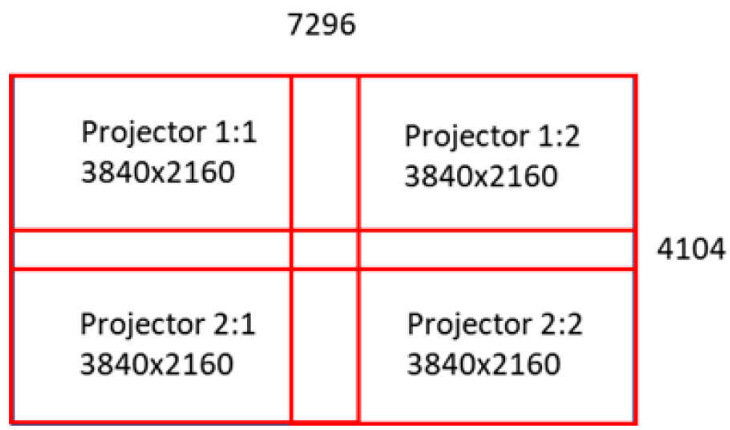


Figure 14-3. Example Blending Layout Results

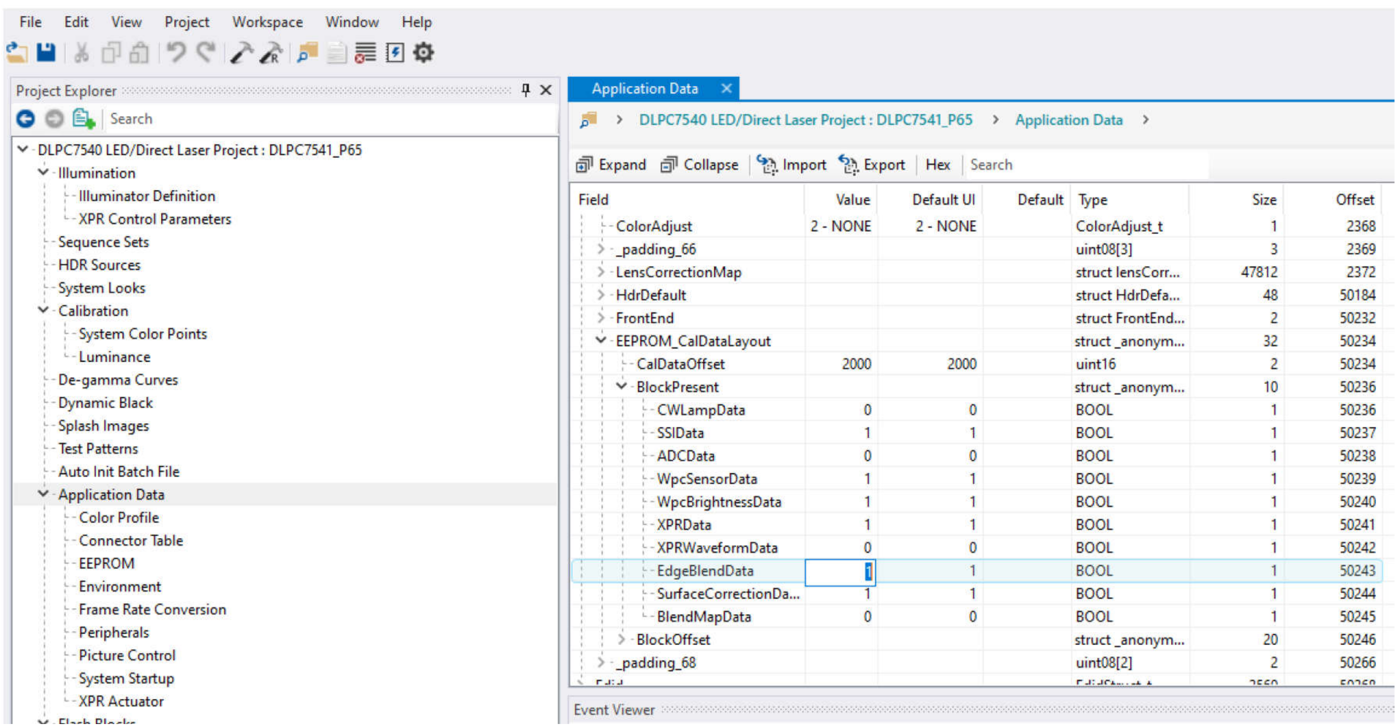
14.8 Storing Edge Blend Configuration in EEPROM

There is an option of storing the edge blending configuration in EEPROM. Below is shown how to set storage to EEPROM.

Storing in EEPROM

1. Open DLP Composer software and select the required DLP composer project file and open it. In the project explorer option on right click on the Application Data option.
2. In the Application Data field tree, expand the Configuration option.
3. In this expand the EEPROM_CalDataLayout option.
4. In this expand the Block Present option.
5. For the EdgeBlendData set the value as 1 to enable storage in EEPROM.

   The following figure shows the EdgeBlendData whose value needs to be assigned as 1 here.



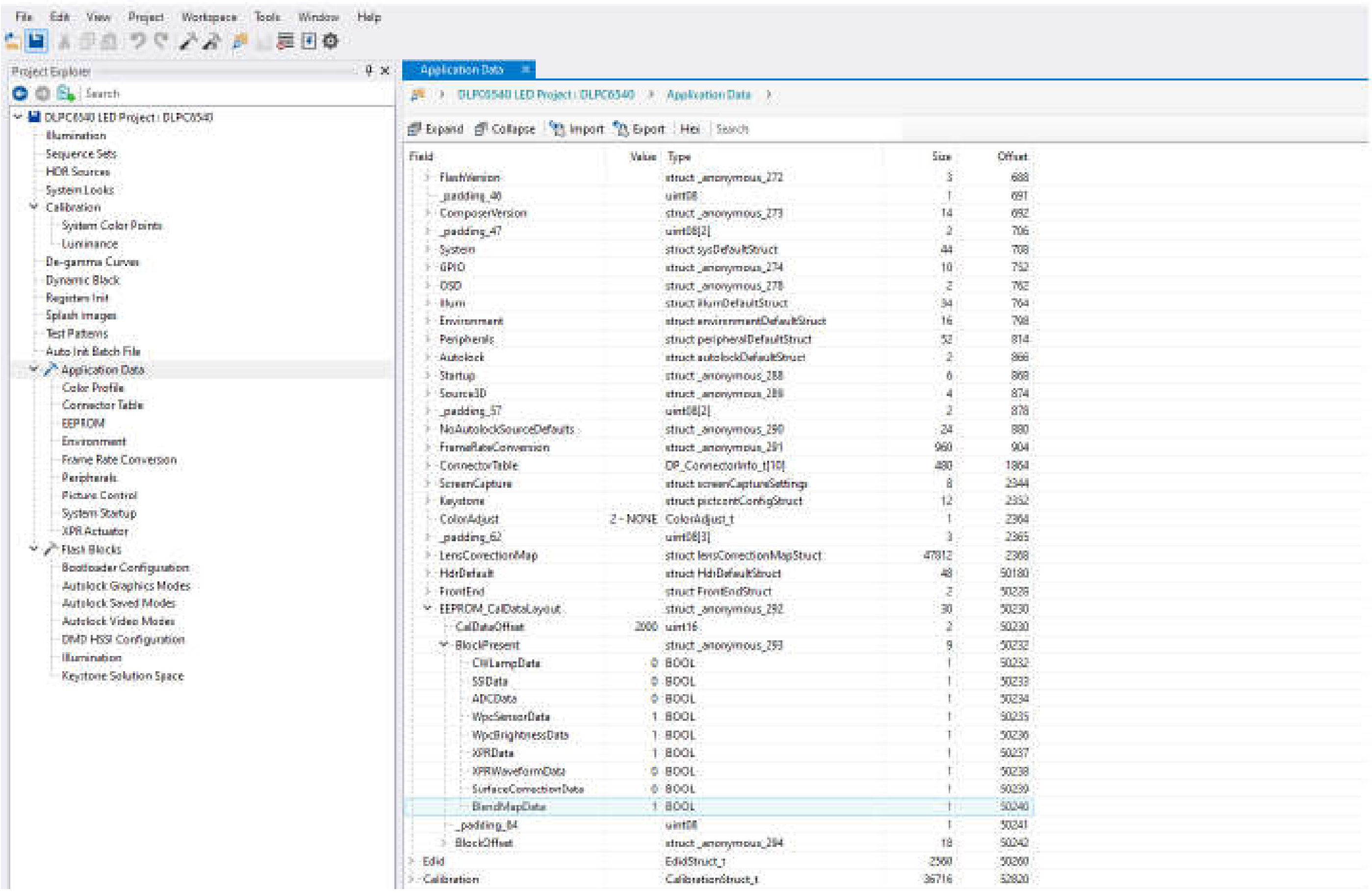
14.9 Storing in EEPROM or Secondary Flash

Edge blending configuration data can be stored in EEPROM or secondary flash.

Storing in EEPROM

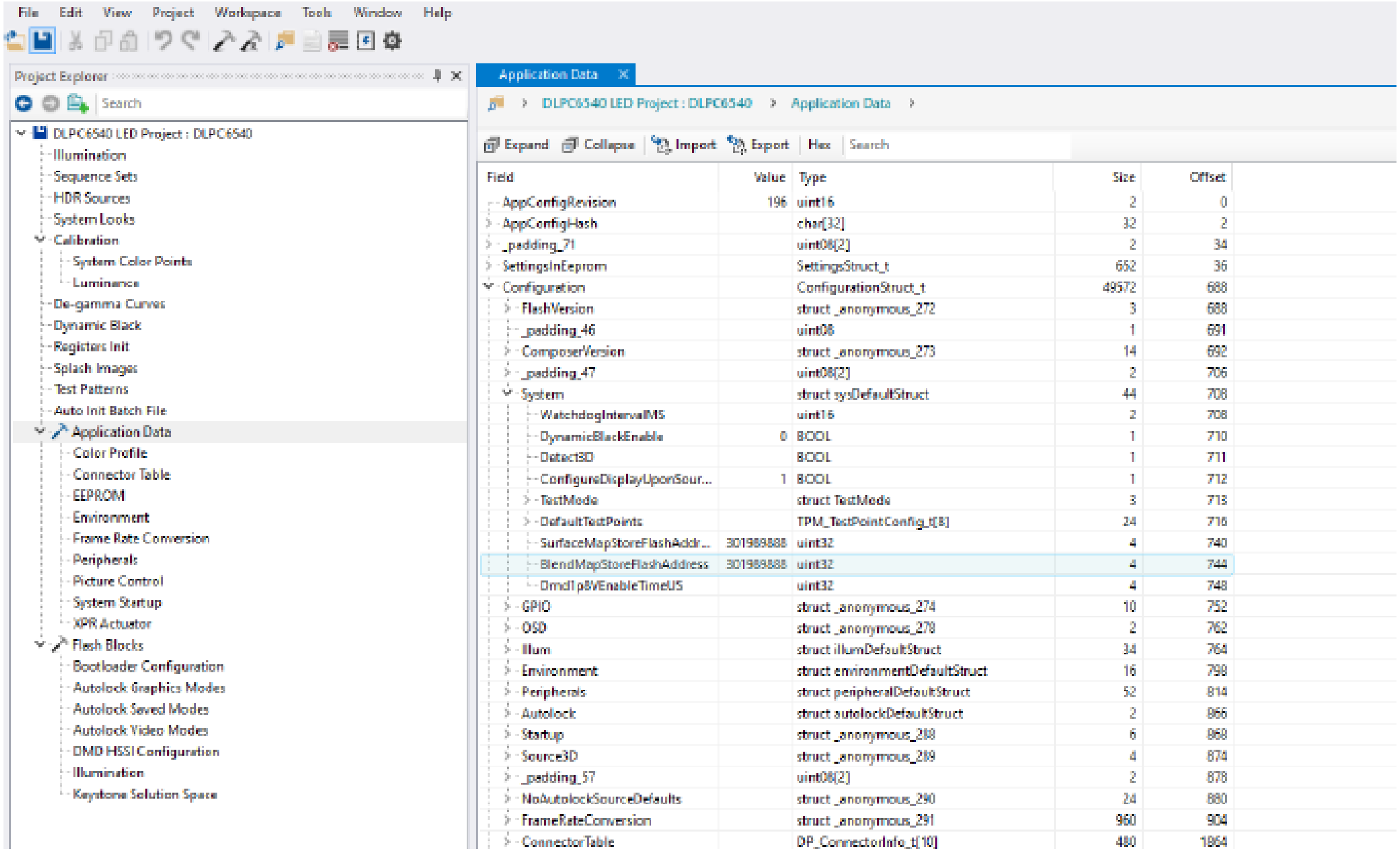
1. Open the DLP Composer software, select the required DLP composer project file, and open the project.
2. In the project explorer option on right, click on the Application Data option.
3. Expand the EEPROM_CalDataLayout option.
4. Expand the Block Present option.
5. For the BlendMapData set the value as 1 to set the storage as EEPROM.

   The following figure shows the BlendMapData whose value needs to assigned as 1 here.



Storing in Secondary Flash

1. Open the DLP Composer software, select the required DLP composer project file, and open the project.
2. In the project explorer option on right click, on the Application Data option.
3. Expand the EEPROM_CalDataLayout option.
4. Expand the Block Present option.
5. For the BlendMapData make sure it is set to 0 value.
6. In the Application Data option expand the Configuration and then system.
7. Set the address in the BlendMapStoreFlashAddress where the Blend Map data is to be stored.



14.10 Manual Blending GUI in Control Program

The first two steps in the GUI focuses on getting the prerequisite information from the user and the next two steps focuses on the Edge Blending.

The Manual Blending GUI page enables user to do both *photometric corrections* (brightness adjustments in the overlap region) as well as *geometric corrections* (angle adjustments of the keystone corners in the projected image).

Note

The GUI supports the edge blending for a **maximum of 5×5 projectors** layout as per the algorithm restriction.

Step 0 : Prerequisites

Before using the Manual Blending GUI page, please make sure that the following prerequisites are done.

1. The required number of projectors need to be physically set up with the desired overlap region.
2. Make the physical adjustments to the projectors as far as possible to make sure that the projected images are properly overlapped and geometrically aligned.
3. Connect one of the projectors (to be blended) to the Control Program and establish the Command Interface Connection.

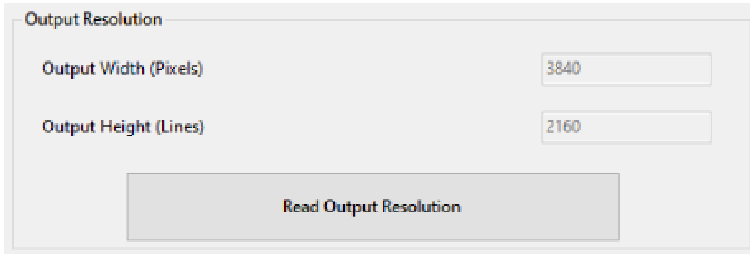
When the connection is ready, use the Manual Blending page starting from Step 1 and go in the order of steps.

Step 1 : Read Parameters

(A) Read Output Resolution

This is the first and foremost step in the process of Manual Blending to read the output resolution information from the projector system. The default value of Output Width is 0 and Output Height is 0. Clicking on the "Read

Output Resolution" button populates the valid (>0) output resolution information if and only if the connection to the projector system has been established.

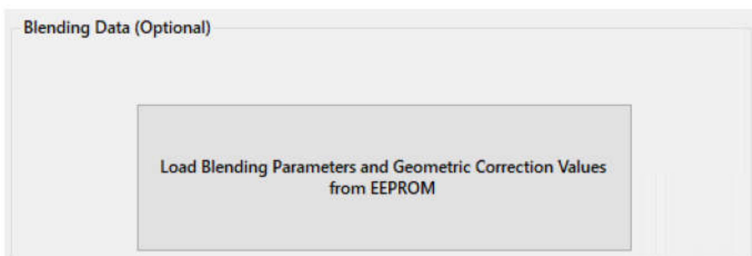


When the output resolution information has been read from the projector system, all other GUI elements that are relevant for the next step of action are enabled.

Note: Unless the output resolution information is populated, none of the following functionalities in the GUI would work as expected.

(B) Read Blending Data (Optional)

The user is given two options to either to read the manual blending parameter values stored in the EEPROM and populate into the fields in the page using the button "Load Blending Parameters and Geometric Correction Values from EEPROM" or manually enter data in the GUI in Step 2. When the blending parameter values from the EEPROM are loaded in the GUI, then the user can apply the same on the screen by clicking on the "Initialize and Apply" button in Step 2.



Step 2 : Initialization of Blending Parameters

This step focuses on getting multiple system information from the user to initialize the parameters that are required for edge blending in the following steps. The user is supposed to enter the following information:

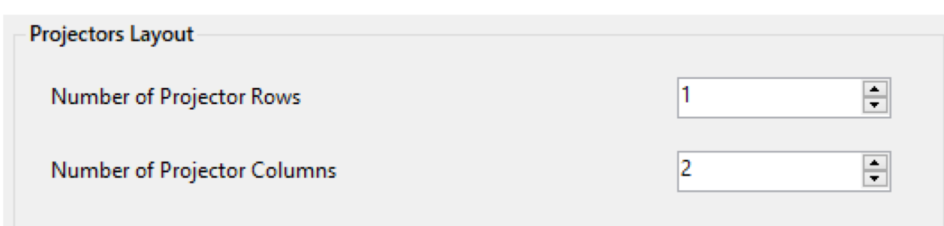
Projectors Layout

Layout in which the projectors are arranged, that is, horizontally stacked (columns) or vertically stacked (rows) to combine the projected image.

Based on the values provided by the user, the table of rows and columns depicting the projector layout in the sections "Current Projector in the Layout", "Black Luminance Values (Nits)" and "White Luminance Values (Nits)" is changed to give a visual prepresentation of the projectors layout.

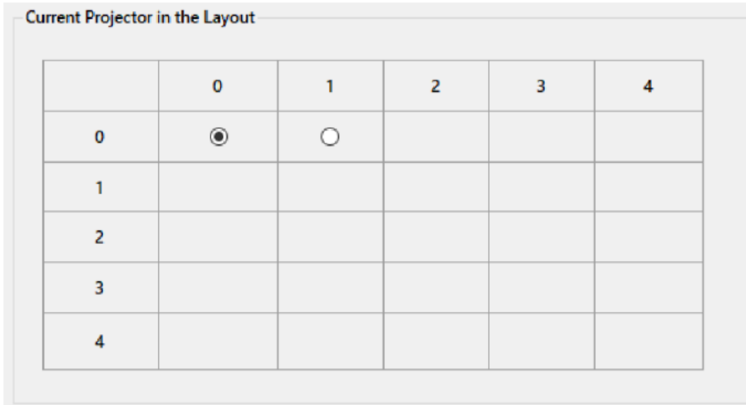
Note

A 1x1 layout, that is, one row and one column which means only one projector, is not supported.



Current Projector in the Layout

In the table of rows and columns depicting the projectors layout, identify the projector system that is currently being blended, that is, the projector system to which the current instance of control program is connected to.

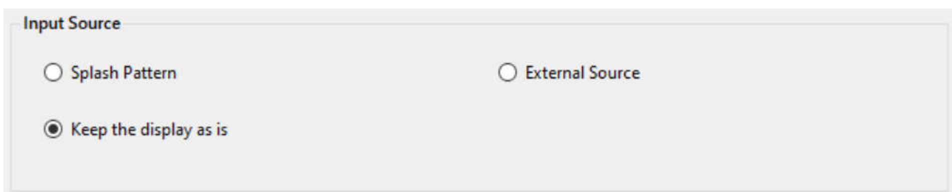


Input Source

Select the input source used to display the pattern or image that should be used for calibration. **This calibration patternor image must match the output resolution of the system identified in Step 1**.

By default, "**Splash Pattern**" is the input source that displays a **built-in default calibration pattern** used for the Manual Blending process.

If "**External Source**" is the input source selected, then the user is expected to display a **custom calibration pattern through External Video Input**. The option to "Keep the display as is" makes it so no updates are made to the display. This is the most useful option for quickly tuning the RGB values for Simple 1D RGB mode.



Overlap Information

The user is allowed to give the overlap and fusion zone information in either exact number of pixels or lines or as a percentage in both horizontal and vertical directions.

When the "Projectors Layout" has

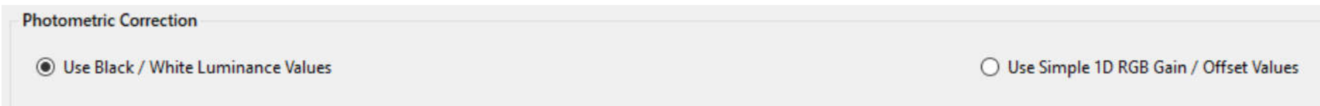
- More than one row, then the user must enter the "Vertical Overlap Height".
- More than one column, then the user must enter the "Horizontal Overlap Width".

The GUI controls are disabled when those values are not needed or used.

1. Note: The minimum overlap supported is 1%.
2. "Corner" systems can support a maximum overlap of 60%.
3. "Non-Corner" systems (edge or middle systems) can support a maximum overlap of 49%.

Photometric Correction Mode

For 2D cases and cases that include warping, "Use Black / White Luminance Values" must be selected. For 1D cases without warping, "Use Simple 1D RGB Gain / Offset Values" can be chosen.

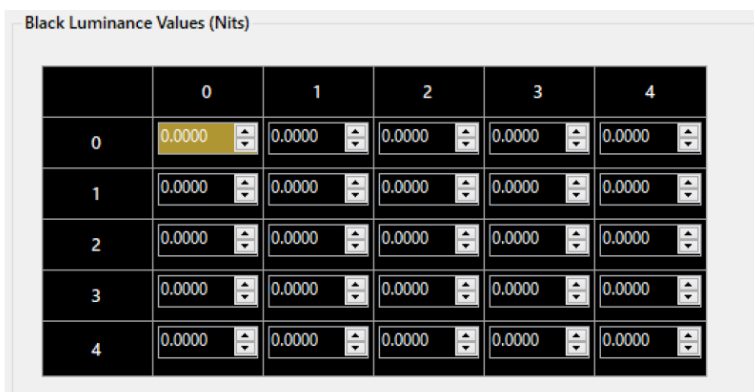


Black Luminance Values (Nits)

- Required for 2D Luminance mode.
- The black luminance values of all the projectors in the layout are entered in the table here.
- The black luminance value for the current projector that is being calibrated is highlighted.
- The minimum value and default value is 0.0000.
- Floating point values are supported up to 4 decimal places.
- The maximum value cannot be greater than or equal to the white luminance value of the corresponding projector, that is, 0 ≤ Black Luminance Value < White Luminance Value of the corresponding projector.

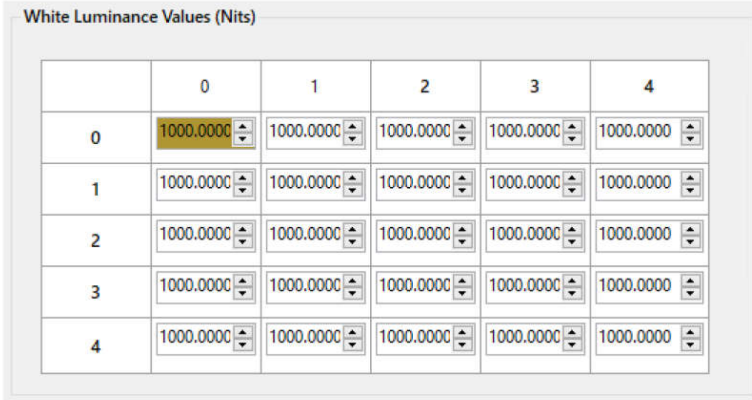
Note

Display the black curtain in the projector system and measure the brightness using a light meter to get the value in nits.



White Luminance Values (Nits)

- Required for 2D Luminance mode
- The white luminance values of all the projectors in the layout are entered in the table here.
- The white luminance value for the current projector that is being calibrated is highlighted.
- The default value is 1000.0000.
- Floating point values are supported up to 4 decimal places.
- The minimum value cannot be less than or equal to the black luminance value of the corresponding projector, that is, 0 ≤ Black Luminance Value < White Luminance Value of the corresponding projector.

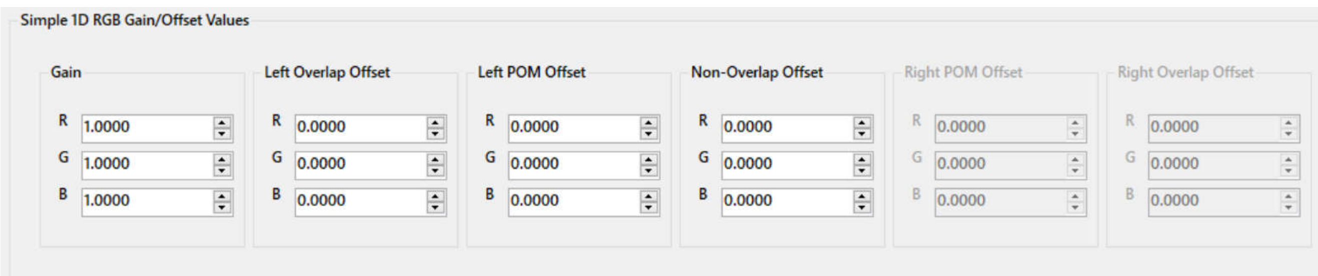


White Luminance Values (Nits)

	0	1	2	3	4
0	1000.0000	1000.0000	1000.0000	1000.0000	1000.0000
1	1000.0000	1000.0000	1000.0000	1000.0000	1000.0000
2	1000.0000	1000.0000	1000.0000	1000.0000	1000.0000
3	1000.0000	1000.0000	1000.0000	1000.0000	1000.0000
4	1000.0000	1000.0000	1000.0000	1000.0000	1000.0000

Simple 1D RGB Values

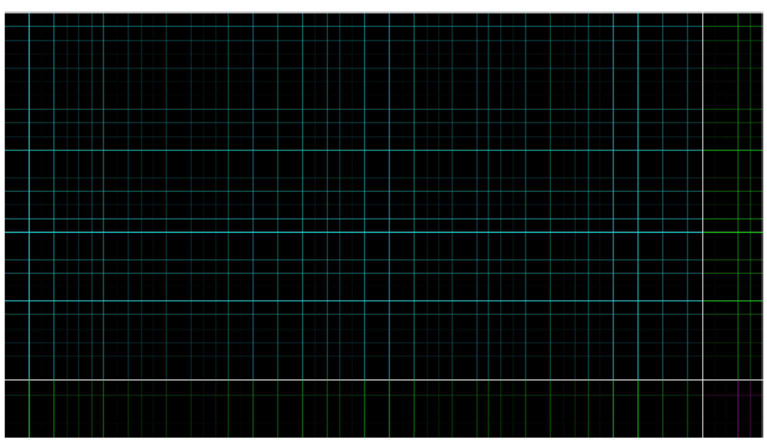
The RGB gain and offset values for the projector are entered in the below fields. The acceptable range of values are between 0 and 1 for all fields. TI recommends starting with the gain fields as all 1, and the offset fields as all 0. The gain values should be lowered so that a white field appears to be the same brightness as the least bright projector in the system. Raise the offset values to where the black field brightness of all projectors is the same.



After all of the above values are entered, click the "Initialize and Apply" button to initialize all the blending parameters and apply the values on the screen.

Note

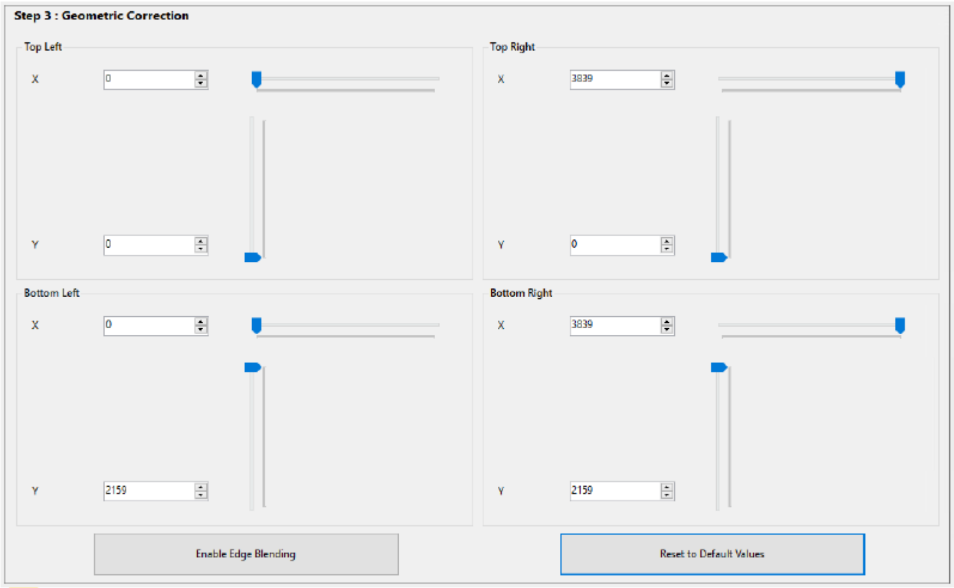
When the initialization is done, the default "Splash Pattern" input source looks something like this on the screen. The green lines depict where only 2 projectors overlap and the magenta lines depict where more than 2 projectors overlap.



Step 3 : Geometric Correction

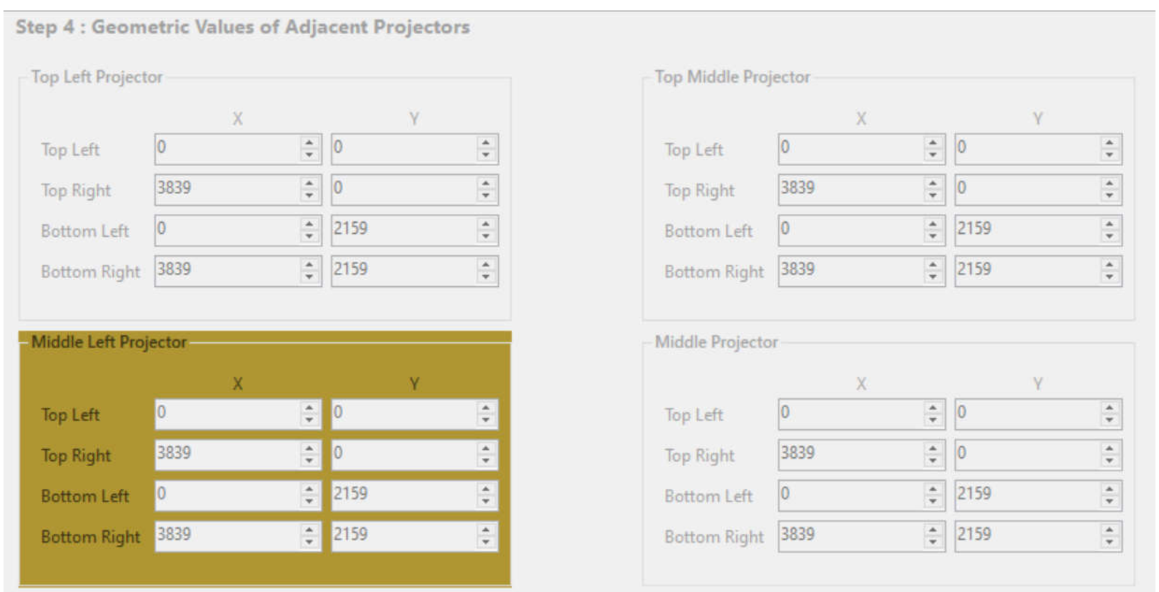
After the blending parameters are initialized, the user is allowed to perform the geometric correction only for the keystone corners.

- When the user changes any of the keystone corner values, then the updated values are sent through the ESW commands to the projector system and the immediate visual change can be seen on the projected pattern/image on the screen.
- Clicking on the "Reset to Default Values" button would revert all the value changes done so far to their corresponding default values.
- Clicking on the "Enable Edge Blending" button enables the user to enable the edge blending in the overlap region of the pattern displayed, that is, the lines in the pattern on the overlap region will not look sharper.
  – By default, Edge Blending will be disabled to ensure that the user first makes the geometric correction with the sharper looking lines in the pattern on the overlap region.
  – When the geometric correction is done, then the user can click on this button to enable the edge blending in the overlap region.
  – When the edge blending is enabled, this button will be toggled to "Disable Edge Blending".



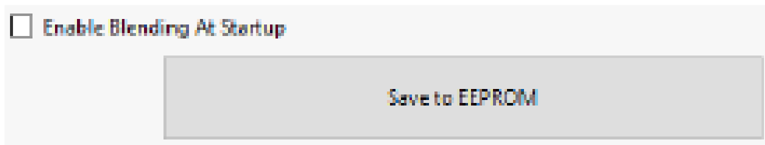
Step 4: Enter Adjacent projector Geometric Parameters

For the 2D Luminance mode algorithm to correctly calculate the shapes of the overlapping regions, the keystone values of the adjacent projectors must be entered. The values can remain at default (unwarped) until all projectors in the system have completed the warping phase



Step 5:Apply Manual Blending Corrections to EEPROM

1. When satisfied with the manual blending corrections done so far and ready to store it in the EEPROM, choose if the blending data to be stored in EEPROM needs to be applied at startup or not. This can be done by using the "Enable Blending At Startup" checkbox. By default, this check box is unchecked.
   - If the checkbox is checked, then the blending data stored in EEPROM is applied at system startup.
   - If the checkbox is unchecked, then the blending data stored in EEPROM are hot applied at system startup.
2. Click the "Save to EEPROM" button to write the manual blending parameter values in the page to the EEPROM and apply it at startup or not based on the "Enable Blending at Startup" check box state.



15 Illumination Control

The controller has built in driver functions to control different types of solid-state illumination (SSI) systems. Two types of signals go into the illumination system:

- Enable strobes: Enable strobes are meant for turning ON/OFF of the Red, Green and Blue illumination modules independently. These signals are generated by the PWM sequences.
- Current control signals: Current control signals meant for driving specified illumination module with specific level of current. The current levels can be specified by the user using DLPA3005 set current command. In case algorithms like LED WPC (White Point Control) and/or Dynamic Black are enabled, then the illumination current values are determined by these algorithms.

Commands are provided for current control of three types of systems:

- General PWM based
- DLPA3005 PMIC based

Use the Set Illumination Enable command to turn the illuminators ON or OFF.

16 Peripherals

Commands listed in this section are provided to control and configure peripherals such as GPIO, PWM and UART.

16.1 GPIO

There are 88 GPIO pins in the system. Some of these pins are dedicated for system specific operations. Refer to DLPC7540 controller data sheet for the freely available GPIOs. The functions listed below can be used to setup the available GPIOs.

Command Name	Description
GPIO Configuration	Configure GPIO as input or output; in case of output, configure as Standard or OpenDrain type, with default value as HIGH or LOW.
Set/Get GPIO	Change the state of the output GPIO to HIGH or LOW and to Read the state of input GPIO pin.

Users are advised to have proper (default) pullup resistance and pulldown resistance on the GPIO pins to avoid problems especially when the controller is in reset or boot-up state. By default, all the freely available GPIOs are configured as INPUT and tri-stated. The system reconfigures them only upon receiving these commands.

16.2 PWM

Controller has 3 general purpose PWM output pins and 2 general purpose input pins. User has option to configure PWM pins as mentioned below.

Command Name	Description
PWM Output Configuration	Port number, frequency, duty cycle, enable/disable
PWM Input Configuration	Port number, sample rate, enable/disable

17 Interface Protocol

17.1 Supported Interfaces

The communication interfaces supported for DLP controller include a serial data bus conforming to the I^2C specification up to 400 kHz, USB 2.0 and UART interface. In addition to control commands, parallel flash programming is also supported over these interfaces.

17.2 I^2C Target

While writing to the DLPC operating in the I^2C target configuration, the first byte following the start condition should be the DLPC device write address (34h). It is possible to change the device address to any other desired value using DLP Composer tool. The remaining bytes are sent as specified in the command protocol section Section 18.

While reading from the DLPC in I^2C target configuration, the first byte following the start condition should be DLPC device write address +1 (35h default) followed by header and opcode bytes as explained later in the document. All reads from DLPC via I^2C interface starts with a write as explained above specifying the opcode for read. The host should then continue the I^2C transaction with a Restart-Read followed by the number of bytes associated with the command and finally the Stop.

17.3 USB

The DLPC7540 controller has USB OTG 2.0 compliant hardware. When connected to a USB host, the controller configures as USB device (target) mode operating at high speed (480 Mbit/s).The controller enumerates one of the interfaces as a generic WinUSB device with two bulk endpoints. A USB bulk transfer sends the command and response packets through these endpoints. The OUT endpoint is used for command packet and the IN endpoint is used for response packet. The USB transfer size can vary from 1 byte to 512 bytes. When the host sends the USB IN request, the controller responds with NAK until there is a response packet ready from the software.

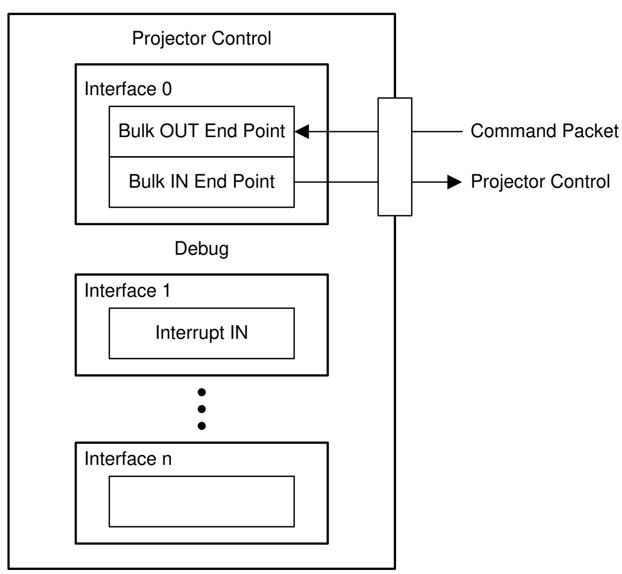


Figure 17-1. USB Core

18 Command Protocol

This section describes the command protocol implemented in DLPC7540. This is the protocol to be used by any external controller to control the DLPC7540 controller using any of the supported commands. The same protocol is applicable across all supported peripheral interfaces (USB, I2C, UART) and application types (bootloader, reference application).

This protocol specifies a flexible length header. The minimum header length is one byte. The first header byte indicates how to interpret the remaining bytes such as opcode, data and checksum (for error detection). There is also a destination parameter in the header that directs the command to different entities within the projector application.

Use this flexible header length method for application that require a minimum of overhead bytes can opt for the one byte header. For a more robust application, configure a larger header that includes data length and/or checksum.

18.1 Command Packet

The command packet defines the packet format to follow when commands are sent to the DLP Controller. Fields that are always present are indicated in **bold**, and optional fields are indicated in normal font.

The definition of which fields are present is based on the 1-byte header field. The length field is mandatory if a command is defined as having variable data size.

Table 18-1. Command Packet format

Field	Size (bytes)	Description
Header	1	See Table 18-2.
Opcode	1 or 2 based on opcode length field in the header	Command opcode. Command opcode number greater than 0xFF should be sent using 2 bytes. Other opcodes can be sent with 1 byte or 2 bytes. In case of 2-byte opcode, first byte is the LSB.
Length	2 or 0 based on data length present field in the header	Length of the command data in bytes following this byte. Checksum is not included in length. For example, length = 10 means that there are 10 bytes of data after this length field. The LSB of length is sent first followed by the MSB.
Data	0-511 (total of maximum 512 bytes in the whole message including header and checksum)	Parameters and data

Checksum	1 or 0
	(optional as checksum present field of header byte)
		Checksum of all bytes in the message including header bytes. Fletcher's checksum is implemented as shown here:
		`uint32  SimpleChecksum = 0;`
		`uint32 SumofSumChecksum = 0;`
		`uint08 *Addr = (uint08 *) StartAddress;`
		`while (NumBytes--)`
		`{`
		`    SimpleChecksum += *Addr++;`
		`    SumofSumChecksum += SimpleChecksum;`
		`}`

Table 18-2. Command Header Byte

Bits	Field name	Values
0:2	Destination	See Section 18.3.
3	Opcode Length	1 = Two byte opcode 0 = One byte opcode
4	Data length Present	1 = Length field present in the extended header 0 = No length field
5	Checksum Present	1 = Checksum present after data bytes 0 = Checksum not present
6	Reply Requested	1 = Device sends a response packet to every write command. This field is applicable only for write commands. 0 = Response packet not sent for write commands.
7	Read Command	1 = Read Command 0 = Write Command

18.2 Response Packet

The Response packet is the format in which the DLP Controller replies to the host. The Response packet format is followed for both Write Response and Read Responses. For write commands, the Response packet is sent only if the "reply requested" bit is set in the command header.

The DLP Controller matches the response header to the same format as the incoming Command packet header. There is however an exception - if the Response packet is for a command that expects variable number of data bytes, the Response packet includes the length field (irrespective of whether the command packet had length mentioned or not). Also see Section 18.6 related to variable sized commands.

Similar to the definition of Command packet, fields in **bold** represents fields that are always present.

Table 18-3. Response Packet Format

Field	Size (bytes)	Description
Header	1	See Table 18-4
Length	2 or 0 (Optional as per Datalength Present field in the header)	Length of the command data in bytes following this byte. Checksum is not included in length. For example, length=10 means there are 10 bytes of data after this length field. The LSB of length is sent first followed by the MSB.
Data	0-511 (total of max 512 bytes in the whole message including header and checksum)	Response data bytes depends on the command code. If error bit in the header is set, there is one data byte. This byte indicates the error code that caused the command to nack. The error code definitions are listed in Table 18-5.
Checksum	1 or 0 (optional as per Checksum present field of header byte)	Checksum of all bytes in the message including header bytes. Fletcher's checksum.

Table 18-4. Response Header Byte

Bits	Field Name	Values
0:2	Destination	See Section 18.3
3	Reserved	NA
4	Datalength Present	1 = Length field present in the extended header 0 = No length field
5	Checksum Present	1 = Checksum present after data bytes 0 = Checksum not present
6	Error	1 = Error. First data byte will have the error code that gives more information about the failure 0 = No error
7	Busy	1=system busy/response not ready; 0=response ready. Applicable only for I2C based communication

Table 18-5. Error Code Definitions

Error Code	Description
1	Invalid destination
2	Invalid or unknown command
3	Invalid length
4	Allocated buffer is not enough to store a command
5	Length information missing for a variable sized command
6	Checksum mismatch
7	Controller not compatible to run the application
8	Read not supported
9	Write not supported
10	Execution failed
11	Invalid response length

Table 18-5. Error Code Definitions (continued)

Error Code	Description
12	Buffer full

Write responses are optional as described in the command header description above. If response is requested, it is imperative to read the response (both Write Response and Read Response) immediately following the respective Command Packet. The response of a command is lost as soon as the DLP Controller receives another set of bytes from the host.

18.3 Destination Details

The following table lists the mapping of destination number to application. Destination value 0 is reserved and shall not be used by any application. Both bootloader and application software implement Destination 1. This implementation allows for sharing common commands between applications. The first 32 command IDs (0 to 31) are reserved for this purpose. Bootloader can use the command IDs outside of the reserved command range to provide specific commands that it supports. See Section 20 for details.

Table 18-6. Destination Numbers

Destination Number	Destination
0	Reserved
1	Commands common to bootloader and reference application.
2	Extended commands or projector control
3	Remote function call (for internal testing purposes only)
4	System commands
5-7	Reserved

18.4 Error Handling and Recovery

As all physical interfaces support the same protocol, it is difficult to support the start conditions that each interface supports. Also, depending on payload size, a command packet may be sent over multiple packets.

It is also important for the DLP Controller to know the start of a command to be able to parse and execute the command successfully. This means the host and DLP Controller should always be in sync. This will be the case if both host and DLP controller gets reset and powers on together. However, if an error occurs in either side, or if one of host/DLP Controller asynchronously resets the sync is lost. Since the start conditions specific to physical interfaces are not monitored, we need another mechanism to recover when such an error occurs.

To support this use case, the DLP Controller monitors the time of arrival of each group of bytes. If any group of bytes come outside of a defined timeout (750ms) compared to the last group, it is treated as the start of a new command.

The timeout is always measured from the last received group of bytes and not from the group of bytes where it encountered an error. This means, if the host keeps sending command one after the other without a timeout, all of it will be discarded.

It is valid to include multiple commands in a single group, or send commands back to back without waiting for the defined timeout. Both of these cases are controlled by the command handler which execute all such chained commands in the order they are received.

18.5 System Busy - I^{2}C scenarios

When using I^{2}C protocol, the target component pulls the clock line low when it needs to indicate it is busy processing and cannot accept any more data from host. Be aware that when there are multiple target devices on the same bus, the entire communication on the bus gets halted until the busy target component releases the bus. To prevent this undesirable effect, the controller supports following options for the host:

18.5.1 GPIO Implementation

A separate GPIO line (GPIO 58 by default) reports to the host component that the controller is busy or not busy. Upon power-on-reset, the front-end communication device must wait until the signal goes to LOW state. A signal that remains HIGH continuously indicates a problem with controller boot-sequence. The source of the problem must be resolve before proceeding.

When a command is sent, the HOST_IRQ/SYSTEM_BUSY signal is pulled HIGH until the command completes execution. If the device attempts to send another command while execution of the first command is ongoing, system confirms whether the HOST_IRQ/SYSTEM_BUSY signal is HIGH or LOW and then takes the decision to send the command. This process ensures that there is no clock stretching, and other devices on the I^2C bus are not affected, but it ensures that the command handler is occupied and no other command can be sent at this point. Use the DLP Composer tool to assign a GPIO for this purpose.

18.5.2 Short Status Response

When the I^2C master requests a data read, the Busy flag (7 the bit in the header byte) indicates the short status. If it is set, it means that the DLPC is busy and does not have a response to send back yet. The host can use the System busy pin as a check for the controller's availability to receive. When this bit is set, the rest of the bits of the response header shall be treated as don't care and no further bytes to be read. The expectation is that the host will keep reading from the Controller for response until this bit is cleared. When that occurs the response header is valid, and remaining data is as per the command.

If the host abandons a read command midway or sends another command immediately after sending read command, the response bytes in the controller buffer gets discarded and the new command gets processed.

For USB communication layer, the controller indicates the busy status by NAK response to the read request.

18.6 Support for Variable Data Size

For large data handling commands such as flash download and flash read, the user can allow some commands to support variable number of data bytes. To support this use case, the commands that require variable data size, is mandated to include length as part of the Command packet header. The Command Handler uses the given length to decode the received Command packet and execute correctly. Similar to the Command packet, the data in the response packet may also be variable. The Command Handler includes length in the Response packet header for such commands.

The command protocol is designed to support commands up to a length of 65535 bytes (2-byte length field). However, due to memory limitations, the command handler implementation limits it to a maximum size of 512 bytes in a Command packet (this includes all bytes in the command like header, checksum etc.).

19 Auto-Initialization Batch File

The DLPC7540 systems provide the option of an auto-initialization batch file that can be included in the flash image using the DLPDLC-GUI. The auto-initialization batch file allows the user to specify a set of commands (as described in this document) to be executed at system startup (such as booting up in Splash display mode or fixed video input for fast boot up time etc). The firmware executes the command specified in this batch-file in the specified order after it completes its own initialization procedure.

This feature provides allows the user to pre-configure the system to consistent set of powerup conditions.

20 Command Descriptions

Consider these guidelines applicable to all the command descriptions that follow in this document.

- Byte order. Wherever a parameter is specified as more than 1 byte in length, the order in which it must be sent/read is LSB first and MSB last.
- Parameter for read commands: All read commands where a read parameter is not explicitly mentioned in the command description, means those commands don't accept a read parameter. A read parameter is defined only for certain read commands to specify the details of what is being sought to be read.
- When the input parameters to a command are in fixed point format, it is specified such as format = s8.2 or format = u12.4 etc. where s stands for signed and u stands for unsigned.

  Fixed-Point Representation:

  This representation has fixed number of bits for integer part and for fractional part. Negative numbers are represented in two's complement format.

Fixed Point representation - [Integer][Fraction]

Example: Assuming the format is signed and using 32-bit format, with16 bits for the integer part and 16 bits for the fractional part. This will be referred to as s15.16 format.

In this case, -43.625 and 43.625 are represented as follows:

[1111111111010101][1010000000000000] = 0xFFD5A000 = -43.625

[0000000000101011][1010000000000000] = 0x002BA000 = +43.625

21 System Commands

The commands described in this guide are compatible with Software version v5.1.0.

21.1 3D

3D

Table 21-1. Enable Three D [Opcode: B1h | Destination: 4]

Set Enable Three D	
Write Parameters	
Byte	**Description**
Byte 0	Enable bit0: TRUE - Enable Processing, FALSE - Disable Processing.
Enables 3D functionality.	

Get Enable Three D
Data returned is in the same format as the Write Parameters.
Returns whether 3D is enabled or not.

Table 21-2. Three D Source Configuration [Opcode: B2h | Destination: 4]

Set Three D Source Configuration	
Write Parameters	
Byte	**Description**
Byte 0	Format 0 = VSync separated (field sequential) format. 1 = VSync separated (frame sequential progressive) format. 2 = Over Under (vertically packed) half resolution format. 3 = Over Under (vertically packed) full resolution format. 4 = Side by Side (horizontally packed) half resolution format. 5 = Side by Side (horizontally packed) full resolution format. 6 = Undefined format.
Byte 1	LR Reference 0 = 3D LR from frame determines L/R (High=Left). 1 = GPIO determines L/R (High=Left). 2 = Vsync/Hsync alignment determines L/R. 3 = LR 1st Frame 4 = LR reference is embedded in video data. 5 = Undefined LR reference.
Byte 2	Frame Dominance 0 = VSync separated sources only Left Eye is 1st frame in 3D image pair. 1 = VSync separated sources only Right Eye is 1st frame in 3D image pair. 2 = Undefined Frame Dominance.
Byte 3	LR Encoding 0 = Single colored lines encoding. 1 = No Encoding 2 = LR 75 25 Encoding 3 = Undefined LR encoding.
Byte 4	TB Reference 0 = Top is Left Eye. 1 = Top is Right Eye. 2 = No Top/Bottom reference is available. 3 = Undefined Top/Bottom reference.
Byte 5	OE Reference 0 = Odd field is Left Eye. 1 = Odd field is Right Eye. 2 = No Odd/Even reference is available. 3 = Undefined Odd/Even reference.
Byte 6	Num Active Blank Lines
Byte 7	Number Of Encoded Lines
Bytes 8-9	Left Encoded Line Location
Bytes 10-11	Right Encoded Line Location

Table 21-2. Three D Source Configuration [Opcode: B2h | Destination: 4] (continued)

Set Three D Source Configuration	
Byte 12	Blanking Color 0 = ChannelA=0 ChannelB=1023 ChannelC=0 for RGB sources. YUV sources will be converted. 1 = ChannelA=1023 ChannelB=0 ChannelC=0 for RGB sources. YUV sources will be converted. 2 = ChannelA=0 ChannelB=0 ChannelC=1023 for RGB sources. YUV sources will be converted. 3 = ChannelA=1023 ChannelB=0 ChannelC=1023 for RGB sources. YUV sources will be converted. 4 = ChannelA=0 ChannelB=1023 ChannelC=1023 for RGB sources YUV sources will be converted. 5 = ChannelA=1023 ChannelB=1023 ChannelC=0 for RGB sources. YUV sources will be converted. 6 = ChannelA=1023 ChannelB=1023 ChannelC=1023 for RGB sources. YUV sources will be converted. 7 = ChannelA=0 ChannelB=0 ChannelC=0 for RGB sources. YUV sources will be converted. 8 = 75% of the line is Blue 25% is black 9 = 25% of the line is Blue 75% is black 10 = Undefined color.

Get Three D Source Configuration
Data returned is in the same format as the Write Parameters.

Table 21-3. Left Right Signal Polarity [Opcode: B3h | Destination: 4]

Set Left Right Signal Polarity	
Write Parameters	
Byte	**Description**
Byte 0	Left Right Polarity Is Inverted bit0: TRUE - Left / Right Frame are swapped. FALSE - Left / Right Frame are normal.
This command inverts the L/R signal polarity.	

Get Left Right Signal Polarity
Data returned is in the same format as the Write Parameters.
This command tells whether L/R signal polarity is inverted or not.

21.2 Administrative

Administrative

Table 21-4. Mode [Opcode: 00h | Destination: 1]

Get Mode	
Return Parameters	
Byte	**Description**
Byte 0	Mode Info bit0: Application Mode 0 = Bootloader 1 = Main Application 2 = Main Application True Global bit1: Controller Configuration 0 = Single 1 = Multiple
This command returns whether we are in Bootloader or in Main Application.	

Table 21-5. Controller Info [Opcode: 00h | Destination: 4]

Get Controller Info	
Return Parameters	
Byte	**Description**
Bytes 0-3	Controller ID
Bytes 4-12	Controller Name
Returns DLP Controller Information.	

Table 21-6. Version [Opcode: 01h | Destination: 1]

Get Version	
Return Parameters	
Byte	**Description**
Byte 0	App Major
Byte 1	App Minor
Bytes 2-3	App Patch
Byte 4	0-Production; A-Alpha; B-Beta
Byte 5	(0-Production; 1-255-Alpha/Beta)
Byte 6	(0-Not a test build; 1-255-Test-build-number)
Byte 7	API Major
Byte 8	API Minor
Bytes 9-10	API Patch
Byte 11	0-Production; A-Alpha; B-Beta
Byte 12	(0-Production; 1-255-Alpha/Beta)
Byte 13	(0-Not a test build; 1-255-Test-build-number)
This command returns the version of the currently active Application and the version of the underlying API library. The currently active application can be queried using Get Mode command.	

Table 21-7. DMD Info [Opcode: 01h | Destination: 4]

Get DMD Info	
Return Parameters	
Byte	**Description**
Bytes 0-3	DMD device ID
Bytes 4-7	DMD Fuse ID
Bytes 8-25	Reserved
Bytes 26-33	DMD Name
Returns the DMD information.	

Table 21-8. Switch Mode [Opcode: 02h | Destination: 1]

Set Switch Mode	
Write Parameters	
Byte	**Description**
Byte 0	Application to switch to 0 = Switch to bootloader 1 = Via reset 2 = Switch to application regardless of the BOOT_HOLD GPIO State. This option is provided for debug purposes only 3 = Switch to application with DMD True Global Enabled regardless of the BOOT_HOLD GPIO State.
This command is used to switch between bootloader and application mode.	

Table 21-9. DMD Resolution [Opcode: 02h | Destination: 4]

Get DMD Resolution	
Return Parameters	
Byte	**Description**
Bytes 0-1	Effective width of DMD in pixels.
Bytes 2-3	Effective height of DMD in lines.
Returns the DMD width and height in number of pixels and lines respectively.	

Table 21-10. Flash Version [Opcode: 03h | Destination: 4]

Get Flash Version	
Return Parameters	
Byte	**Description**
Byte 0	Flash Version Major
Byte 1	Flash Version Minor
Byte 2	Flash Version Subminor
Returns version number that uniquely identifies the flash image.	

Table 21-11. Flash Layout Version [Opcode: 04h | Destination: 4]

Get Flash Layout Version	
Return Parameters	
Byte	**Description**
Bytes 0-1	Flash Config Layout Version
Bytes 2-33	Flash Config Layout Hash
Bytes 34-35	Application Config Layout Version
Bytes 36-67	Application Config Layout Hash
Returns supported Layout revision numbers and hash for flash config and app config layout.	

Table 21-12. Product Configuration Failure Cause [Opcode: 05h | Destination: 4]

Get Product Configuration Failure Cause	
Return Parameters	
Byte	**Description**
Byte 0	Cause of product configuration failure. 0 = Invalid Controller for the product configuration 1 = Invalid DMD for the product configuration 2 = DMD project data does not match the actual DMD 3 = PAD cannot be used to drive SSI or DMD in ECD system 4 = Invalid Pad Configuration
Use this command to get the cause of product configuration failure if Product Configuration Failed is set in system status command.	

Table 21-13. System Status [Opcode: 06h | Destination: 4]

Get System Status	
Return Parameters	
Byte	**Description**
Bytes 0-3	System Status Word 0 bit0: Color Wheel spinning bit1: Color Wheel phase lock bit2: Color Wheel frequency lock bit3: Lamp is lit bit4: Memory test of internal DRAM passed bit10: Frame Rate Conversion Enable bit11: Sequence Phase Lock bit12: Sequence Frequency Lock bit13: Sequence search bit29: System Color Point Calibration Enable bit30: Variable Illumination Calibration Enable bit31: Brilliant color Calibration Enable
Bytes 4-7	System Status Word 1 bit0: Sequence Error bit1: Pixel clock out of range bit2: Vsync valid bit6: UART port 0 communication error (If Port Enabled) bit7: UART port 1 communication error (If Port Enabled) bit8: UART port 2 communication error (If Port Enabled) bit9: SSP port 0 communication error (If Port Enabled) bit10: SSP port 1 communication error (If Port Enabled) bit11: SSP port 1 communication error (If Port Enabled) bit12: I2C port 0 communication error (If Port Enabled) bit13: I2C port 1 communication error (If Port Enabled) bit14: I2C port 2 communication error (If Port Enabled) bit15: DLPC Initialization Error bit16: Lamp Hardware error occurred bit17: Lamp PPPRF Time Out bit19: No frequency bin found for the selected mode bit20: DLPA3005 Communication error (If DLPA3005 present) bit21: UMC refresh bandwidth underflow bit22: DMD initialization error bit23: DMD power down error bit24: Source definition not present bit25: Sequence binary not present bit26: Product configuration failed bit27: Dither mask not loaded
Bytes 8-11	System Status Word 2 bit0: EEPROM initialization failure
Command to read status information from DLP Controller. If status interrupt is enabled (configurable via default UI tool in DLP Composer), reading back this command will acknowledge/deactivate the interrupt pin until the next change in status.	

Table 21-14. EEPROM Data Present [Opcode: 07h | Destination: 4]

Get EEPROM Data Present	
Return Parameters	
Byte	**Description**
Bytes 0-1	Calibration Data Blocks bit0: Colorwheel Lamp Data Present bit1: SSI Calibration Data Present bit2: ADC Calibration Data Present bit3: WPC Sensor Calibration Data Present bit4: WPC Brightness Table Data Present bit5: XPR Calibration Data Present bit6: XPR Waveform Calibration Data Present bit7: Edge Blend Data Present bit8: Surface Correction Data Present
Reports which of the calibration data blocks are present in EEPROM. Use this command before sending EEPROM Invalidate command (0x0A).	

Table 21-15. General Delay Command [Opcode: 08h | Destination: 4]

Set General Delay Command	
Write Parameters	
Byte	**Description**
Bytes 0-3	Delay In Milliseconds
On receipt of this command controller wait for specified period before executing the next command. This command to be used in Auto Initialization batchfile configuration. Use this command to insert delay between execution of two commands.	

Table 21-16. EEPROM Invalidate [Opcode: 0Ah | Destination: 4]

Set EEPROM Invalidate	
Write Parameters	
Byte	**Description**
Byte 0	Invalidate Settings Data bit0: Invalidate Settings
Bytes 1-2	Invalidate Calibration Data bit0: Invalidate Colorwheel Lamp Data bit1: Invalidate SSI Calibration Data bit2: Invalidate ADC Calibration Data bit3: Invalidate WPC Sensor Calibration Data bit4: Invalidate WPC Brightness Table Data bit5: Invalidate XPR Calibration Data bit6: Invalidate XPR Waveform Calibration Data bit7: Invalidate Edge Blend Data bit8: Invalidate Surface Correction Data
Invalidates the user settings portion of EEPROM data or calibration portion of EEPROM data or both as per input arguments and restarts the system. If none of the settings or calibration data is selected, then the command does nothing. Note : Chose valid flags as returned in Get EEPROM Data Present command.	

Table 21-17. Splash Capture [Opcode: 0Bh | Destination: 4]

Set Splash Capture
Write Parameters
Captures the current external image displayed on the screen and stores it into the Flash memory as a Splash image.

Table 21-18. Splash Capture Status [Opcode: 0Ch | Destination: 4]

Get Splash Capture Status	
Return Parameters	
Byte	**Description**
Byte 0	Capture State 0 = Image Capture Terminated because of error or Timeout 1 = External Image is being written into the internal DRAM Splash buffer 2 = Image is successfully captured into internal DRAM Splash buffer 3 = Image is being programmed into the Flash memory 4 = Image is successfully programmed into Flash memory
Byte 1	Completion Status (percentage)
Returns the current status of splash capture.	

Table 21-19. Terminate Splash Capture [Opcode: 0Dh | Destination: 4]

Set Terminate Splash Capture
Write Parameters
Terminates any ongoing Splash Capture

21.3 Autolock

Autolock

Table 21-20. Autolock Control [Opcode: 24h | Destination: 4]

Set Autolock Control	
Write Parameters	
Byte	**Description**
Byte 0	Autolock Control 0 = Resync 1 = Start 2 = Stop
This command provides user control to relock to a source or to start/stop autolock algorithm.	

21.4 Bootloader

Bootloader

Table 21-21. Boot Hold Reason [Opcode: 12h | Destination: 1]

Get Boot Hold Reason	
Return Parameters	
Byte	**Description**
Byte 0	Reason code 0x00 BOOT_HOLD jumper in HOLD position 0x01 Switched to programming mode initiated by main app 0x02 Reading flash info failed 0x03 Flash layout mismatch 0x04 Can't initialize ARM peripherals 0x05 Can't allocate memory pool 0x06 Failure in initialization task 0x07 Controller is invalid to run the application 0x08 Error in USB initialization 0x09 Error in i2c initialization 0x0A Error getting app configuration 0x0B App configuration layout mismatch
Returns the code that specifies the reason for being in bootloader mode.	

Table 21-22. Flash Info [Opcode: 20h | Destination: 1]

Get Flash Info	
Read Parameters	
Byte	**Description**
Byte 0	Chip Select of Flash Device 0 = Flash chip select 0 memory domain 1 = Flash chip select 1 memory domain 2 = Flash chip select 2 memory domain

Return Parameters	
Byte	**Description**
Bytes 0-1	Manufacturer ID
Bytes 2-9	Device ID
Bytes 10-13	Device size in bytes
Bytes 14 - *	Sector Information bits 0-31: Sector Size bits 32-47: Number Sectors
Byte 14	Availability bit0: 0 = Flash can be used for programming; 1 = Flash cannot be used for programming.
This command returns the flash device and manufacturer IDs. Only CFI compliant flash devices are supported. The system can have multiple flash devices. The command returns the info for the flash present at the given chip select. Note: Chip Select 0 Flash is required for system operation. Other Flash Chip Selects are technically optional but required for Splash Capture and Warp Operations.	

Table 21-23. Programmable Flash Sector Information [Opcode: 21h | Destination: 1]

Get Programmable Flash Sector Information	
Return Parameters	
Byte	**Description**
Bytes 0 - *	Sector Information bits 0-31: Sector Size bits 32-47: Number Sectors
This command returns the flash sector information read from CFI compliant flash devices. If the flash is not CFI compliant, this command fails. The sectors returned by this command are the only ones available for programming a flash image. The system is designed such that the flash image is in a contiguous memory space. If a system has multiple flash parts, then the software checks the size of the flash at ChipSelect 0. If this is equal to the maximum supported size (32MB), then a flash device at ChipSelect 1 (if present) will also be supported for flash programming. Similarly, if the size of flash devices at both ChipSelect 0 and 1 are 32MB, then a flash device at ChipSelect 2 (if present) will be supported for flash programming as well. The command appends the sector information for each part, which is supported for flash programming, and provides them as output.	

Table 21-24. Unlock Flash For Update [Opcode: 22h | Destination: 1]

Set Unlock Flash For Update	
Write Parameters	
Byte	**Description**
Bytes 0-3	Flash Update lock/unlock 0 = Lock 4154802215 = Unlock
This command unlocks the flash update operation (Download, Erase). By default the flash update operations are locked. This is to prevent accidental modification of flash contents. To unlock, the pre-defined key shall be send as the unlock code. Calling this command with any other parameter will lock the flash update commands.	

Get Unlock Flash For Update	
Return Parameters	
Byte	**Description**
Byte 0	0 = Locked 1 = Unlocked
This command returns whether the flash is in unlocked state.	

Table 21-25. Erase Sector [Opcode: 23h | Destination: 1]

Set Erase Sector	
Write Parameters	
Byte	**Description**
Bytes 0-3	Sector Address
This command erases the sector of the flash where the given address falls in. This command is a flash update command, and requires flash operations to be unlocked using Unlock Flash for Update command. The sector address shall be specified as an offset from flash start address. For example in a flash device where all sectors are 64KB of size, sector addresses shall be specified as follows : Sector 0 = 0 Sector 1 = 0x10000 Sector 2 = 0x20000 and so on...	

Table 21-26. Initialize Flash Read Write Settings [Opcode: 24h | Destination: 1]

Set Initialize Flash Read Write Settings	
Write Parameters	
Byte	**Description**
Bytes 0-3	Start Address offset to program data to where Offset 0 refers to first byte in the flash, 1 refers to second byte and so on. This offset must be an even number.
Bytes 4-7	This specifies the number of bytes Flash Write command should expect or the number of bytes Flash Read command should return. This must be an even number.
This command initializes flash read/write operation. This command shall be called before Flash Write command is sent. Note : For Flash Write, the Address and download size set up shall both be even.	

Table 21-27. Flash Write [Opcode: 25h | Destination: 1]

Set Flash Write	
Write Parameters	
Byte	**Description**
Bytes 0 - *	Data to write to flash memory
This command is used to program data to flash. This command shall be called only after setting the start address and size using the Initialize Flash Read/Write Settings command. This command is a flash update command, and requires flash operations to be unlocked using Unlock Flash for Update command. Flash write commands can be chained till the initialized number of bytes are programmed. The bootloader will auto-increment the address and size for each command. Only the initialized number of bytes will be programmed even if more data is provided. It is important to send only even number of bytes per flash write command to ensure all bytes are written. This is done so that all flash writes are optimized as per the multi-word write supported by the flash device. This command supports variable sized payload.	

Get Flash Write	
Read Parameters	
Byte	**Description**
Bytes 0-1	Number bytes to read in this command

Return Parameters	
Byte	**Description**
Bytes 0 - *	The bytes read from the flash
This command is used to read data from flash. This command shall be called only after setting the start address and size using the Initialize Flash Read/Write Settings command. Flash read commands can be chained until the initialized number of bytes are returned. The bootloader will auto-increment the address and size for each command. Only the initialized number of bytes will be returned. Calling the function after returning requested data will return in command failure. This command supports variable sized response.	

Table 21-28. Checksum [Opcode: 26h | Destination: 1]

Get Checksum	
Read Parameters	
Byte	**Description**
Bytes 0-3	Start Address offset for checksum computation where Offset 0 refers to first byte in the flash, 1 refers to second byte and so on.
Bytes 4-7	Number of bytes to compute checksum

Return Parameters	
Byte	**Description**
Bytes 0-3	Simple additive checksum
Bytes 4-7	Sum of simple additive checksum calculated at each address

This command computes and returns the checksum starting at the given address for the specified number of bytes. Checksum is calculated as below :

```
uint32 SimpleChecksum = 0;
uint32 SumofSumChecksum = 0;
uint08 *Addr = (uint08 *) StartAddress;

while (NumBytes--)
{
SimpleChecksum += *Addr++;
SumofSumChecksum += SimpleChecksum;
}
```

Table 21-29. Reset Flash [Opcode: 27h | Destination: 1]

Set Reset Flash	
Write Parameters	
Byte	**Description**
Byte 0	Chip Select 0 = Flash chip select 0 memory domain 1 = Flash chip select 1 memory domain 2 = Flash chip select 2 memory domain
This command resets the Flash device connected to the given chip select. Any partial commands given gets reset and the flash is put in read mode.	

21.5 Calibration

Calibration

Table 21-30. Initialize On The Fly Load Splash Image [Opcode: A9h | Destination: 4]

Set Initialize On The Fly Load Splash Image	
Write Parameters	
Byte	**Description**
Bytes 0-19	Header of the pattern Image (same format as the splash header)
Bytes 20-21	Width of the splash image, in pixels, to which given pattern is to be replicated
Bytes 22-23	Height of the splash image, in pixels, to which given pattern is to be replicated
This Command Initializes the on the fly splash load operation. It is used to give the information about the image (dimensions, pixel format, compression used and so on) and sets the data path for the splash load operation. It also resets prior operations. It should be called once before loading an image on the fly.	

Table 21-31. Load Splash Image On The Fly [Opcode: AAh | Destination: 4]

Set Load Splash Image On The Fly	
Write Parameters	
Byte	**Description**
Bytes 0 - *	Image Data
This Command receives the image data and writes it into the VPS DRAM Buffer. It performs decompression if required. Due to the limitation of USB, the entire image data may not be sent in one go. Hence, this command has to be called multiple times to load the entire image. It also stores any bytes that are unprocessed by the API (if the API does not have all the bytes required to perform an operation). These unprocessed bytes are sent over to callback function the next time data is received. The API displays the loaded splash once the entire image is loaded into the memory. The remaining pixels to load the image is also returned by the callback function.	

Table 21-32. XPR Calibration Pattern Display [Opcode: ABh | Destination: 4]

Set XPR Calibration Pattern Display
Write Parameters
This command loads a pre-defined XPR Calibration pattern as a splash image and displays it on the screen. A 64x64 pattern is repeated over a 3840x2160 display area.

Table 21-33. XPR 4Way Orientation [Opcode: B4h | Destination: 4]

Set XPR 4Way Orientation	
Write Parameters	
Byte	**Description**
Byte 0	Orientation number. Range 0 - 23.
This command sets the orientation number of the actuator position (which gets stored in EEPROM) There are 24 possible options 0 - 23; use this command while performing XPR calibration using TI provided XPR calibration splash image. Note : Use Display Image Size command to make sure the display area is 3840x2160 If the reported display resolution is less than or equal to 1080p this command will not have any influence on the displayed image.	

Get XPR 4Way Orientation
Data returned is in the same format as the Write Parameters.
This command retrieves the last set orientation number or the subframe order

Table 21-34. XPR Actuator Waveform Control Parameter [Opcode: B5h | Destination: 4]

Set XPR Actuator Waveform Control Parameter	
Write Parameters	
Byte	**Description**
Byte 0	XPR Command 0 = Fixed Output Enable 1 = DAC Gain 2 = Subframe delay 3 = Actuator Type (READ ONLY) 4 = Output Enable/Disable 5 = Clock Width 6 = DAC Offset 7 = Number of Segments 8 = Segment Length 9 = Invert PWM A 10 = Invert PWM B 11 = Subframe Filter Value 12 = Subframe Watch Dog 13 = Fixed Output Value
Byte 1	Channel number (0 or 1) of Actuator waveform control for which the command parameter has to be applied
Bytes 2-5	Data that needs to be passed to the command

Table 21-34. XPR Actuator Waveform Control Parameter [Opcode: B5h | Destination: 4] (continued)

Set XPR Actuator Waveform Control Parameter
This command configures/sets up the Actuator Waveform Control(AWC) block. Here, AWCx can be AWC 0 or 1. Bytes 2-5 contains the XPR command data as mentioned in Byte 0. Byte 1 contains AWC channel number, possible values are 0 or 1. **Fixed Output Enable** : Configures Actuator in fixed output mode. Byte 2 : 0x00 - Disable 0x01 - Enable Bytes 3-5 : Reserved must be set to 0x000000 **Gain** : Set Waveform Generator DAC/PWM Gain. Byte 2 : Range 0 - 255 format u1.7 (0 to 1.9921875) Bytes 3-5 : Reserved must be set to 0x000000 **Subframe delay** : Subframe delay Bytes 2-5; Range 0 - 262143 and lsb = 133.333ns **Actuator Type (READ ONLY)** : Actuator type Byte 2 : 0x00 - NONE 0x01 - Optotune (XPR-25 Model) 0x80 - TI Actuator Interface (EEPROM) 0x81 - TI Actuator Interface (MCU) Bytes 3-5 : Reserved must be set to 0x000000 **Output Enable/Disable** : Actuator output enable/disable Byte 2 : 0x00 - Disable 0x01 - Enable Bytes 3-5 : Reserved must be set to 0x000000 Note : Both AWC0 and AWC1 disabled/enabled together **Clock Width** : Defines the high and low width for the output clock (the clock period will be 2*(ClkWidth+1)) 0 = 1 (Clock period is two clocks); lsb = 8.33ns Bytes 2-5 : ClkWidth Example : ClkWidth = 0; will generate clock of 2*(0+1)*8.33 = 16.66ns **Offset** : DAC/PWM Output Offset Byte 2 : Range -128 - +127 format S7 (-128 to +127) Bytes 3-5 : Reserved must be set to 0x000000 **Number of Segments** : Defines number of segments Byte 2 : Range 2 - 255 Bytes 3-5 : Reserved must be set to 0x000000 **Segments Length** : Defines size of the segments Bytes 2-3 : Range 19 - 4095 Bytes 4-5 : Reserved must be set to 0x0000 **Invert PWM A** : Applicable when AWC is configured to PWM type instead of DAC Byte 2 : 0x00 - No inversion 0x01 - Inverted Bytes 3-5 : Reserved must be set to 0x000000 **Invert PWM B** : Applicable when AWC is configured to PWM type instead of DAC Byte 2 : 0x00 - No inversion 0x01 - Inverted Bytes 3-5 : Reserved must be set to 0x000000 **Subframe Filter Value** : Sets Subframe Filter Value - defines the minimum time between Subframe edges. Edges closer than the set value will be filtered out Byte 2 : 0 = Filter disabled, 0 = Filter time will be Val x 60us, Range : 0 - 255 Bytes 3-5 : Reserved must be set to 0x000000 **Subframe Watch Dog** : Defines the maximum time between Subframe edges; if timer expires, then the WG will automatically output the Fixed Output value, and the normal output will resume on the next subframe edge. Bytes 2-3 : 0 = Subframe watchdog disabled, 0 = Watchdog time will be Time x 60us, Range : Range : 0 - 1023 Bytes 4-5 : Reserved must be set to 0x0000 **Fixed Output Value** : Defines the value to be output on DAC/PWM when fixed output mode is selected. Byte 2 : Value to be output on DAC/PWM, Range -128 to 127 Bytes 3-5 : Reserved must be set to 0x000000 **Note** : To use **Subframe Filter Value** and **Subframe Watch Dog** care must be taken to set a value which approximately 10% more than 2x of the operating frequency. For example - 4K @ 60Hz, the value can be set as (1/(60*2))*1.10*10^6 = 9166us.

Get XPR Actuator Waveform Control Parameter	
Read Parameters	
Byte	**Description**
Byte 0	XPR Command 0 = Fixed Output Enable 1 = DAC Gain 2 = Subframe delay 3 = Actuator Type (READ ONLY) 4 = Output Enable/Disable 5 = Clock Width 6 = DAC Offset 7 = Number of Segments 8 = Segment Length 9 = Invert PWM A 10 = Invert PWM B 11 = Subframe Filter Value 12 = Subframe Watch Dog 13 = Fixed Output Value
Byte 1	Channel number of Actuator waveform control block for which the command parameter to be readback

Return Parameters	
Byte	**Description**
Bytes 0-3	Parameter value obtained for the command passed
This command gets the parameter set to the AWC waveform generator. Note : This command is supposed to be used only during the normal operating mode and not during the standby state.	

Table 21-35. DB Border Configuration [Opcode: BBh | Destination: 4]

Set DB Border Configuration	
Write Parameters	
Byte	**Description**
Bytes 0-1	number of lines top of border. Range 0 - 4095
Bytes 2-3	number of lines bottom of border. Range 0 - 4095
Bytes 4-5	number of pixels of left border. Range 0 - 4095
Bytes 6-7	number of pixels of right border. Range 0 - 4095
This command configures area of the DynamicBlack border region for the border exclusion function. The border exclusion function allows the user to reduce the letterbox (black border) effect on a primarily bright image where letterbox area reduces the overall scene brightness for the algorithm. It also helps the algorithm better handle images with bright subtitles where the subtitles increase the overall scene brightness. This command will also be used in a multi-controller configuration to exclude any image overlap required for other image processing algorithms.	

Get DB Border Configuration
Data returned is in the same format as the Write Parameters.
This Command returns the border region area for the DynamicBlack border exclusion function.

Table 21-36. DB Border Weight [Opcode: BCh | Destination: 4]

Set DB Border Weight	
Write Parameters	
Byte	**Description**
Byte 0	Weight value of border pixels 0 = 0% weighted; 1 = 25% weighted; 2 = 50% weighted; 3 = 75% weighted 0 = Weighted 0% 1 = Weighted 25% 2 = Weighted 50% 3 = Weighted 75%

Get DB Border Weight
Data returned is in the same format as the Write Parameters.
Sets weight value of the DynamicBlack border region for the border exclusion function

Table 21-37. DB Clip Pixels [Opcode: BDh | Destination: 4]

Set DB Clip Pixels	
Write Parameters	
Byte	**Description**
Bytes 0-1	Number of pixels that can be clipped. Range = 0 to 65535.
This command returns currently configured number of steps to allow the DynamicBlack aperture to move.	

Get DB Clip Pixels
Data returned is in the same format as the Write Parameters.
This command returns the currently selected number of pixels that can be clipped.

Table 21-38. DB Gain [Opcode: BEh | Destination: 4]

Set DB Gain	
Write Parameters	
Byte	**Description**
Bytes 0-1	Gain value. Typical value range is 1.0 to 8.0. Format = u4.12
This command controls the DynamicBlack gain value. Typical value range is 1.0 to 8.0. Manual Mode needs to be enabled to set the gain as it will override the gain value that is calculated every frame.	

Get DB Gain
Data returned is in the same format as the Write Parameters.
This command gets the DynamicBlack gain value. Typical value range is 1.0 to 8.0

Table 21-39. DB Histogram [Opcode: C2h | Destination: 4]

Get DB Histogram	
Return Parameters	
Byte	**Description**
Bytes 0-135	Start address of the DB histogram array. Array size is 34. The LSB of each bin represents 32 pixels. Each bin saturates at 0x0003FFFF.
This command returns the start address of the DynamicBlack(DB) histogram data. The histogram contains scene brightness data from the previous frame. The DB histogram contains 34 bins measuring non-overlapping intensity ranges in the displayed image. The value of each bin equals the number of pixels within the bin's intensity range. Each pixel's intensity is calculated as the maximum of its red, green, and blue values. In other words, pixel intensity = MAX(R, G, B). Each pixel has a format of unsigned 8.8, making 16 bit values. Bins 32 and 33 are special bins that represent pixels that have values of exactly zero and only fractional values respectively. This function can be used independently of aperture control for image improvement in dark scenes.	

Table 21-40. Current Led Color Point [Opcode: C4h | Destination: 4]

Get Current Led Color Point	
Return Parameters	
Byte	**Description**
Bytes 0-1	Chromatic x coordinate in (Transmitted in u1.15 format) Format = u1.15
Bytes 2-3	Chromatic y coordinate in (Transmitted in u1.15 format) Format = u1.15
Bytes 4-7	Luminance Y coordinate
Gets x,y coordinates of system's current white point. WPC should be initialized and calibration data should be set before calling this command.	

Table 21-41. WPC Optimal Duty Cycle [Opcode: C5h | Destination: 4]

Set WPC Optimal Duty Cycle
Write Parameters
Searches available duty cycles and sets the optimal one for correct LED white point. Sensor calibration Data should be set before using this command.

Get WPC Optimal Duty Cycle	
Return Parameters	
Byte	**Description**
Bytes 0-1	Red Ideal Duty Cycle (Transmitted in u8.8 format) Format = u8.8
Bytes 2-3	Green Ideal Duty Cycle (Transmitted in u8.8 format) Format = u8.8
Bytes 4-5	Blue Ideal Duty Cycle in (Transmitted in u8.8 format) Format = u8.8
Bytes 6-7	Red Optimal Duty Cycle (Transmitted in u8.8 format) Format = u8.8
Bytes 8-9	Green Optimal Duty Cycle (Transmitted in u8.8 format) Format = u8.8
Bytes 10-11	Blue Optimal Duty Cycle in (Transmitted in u8.8 format) Format = u8.8
Gets Ideal Duty Cycle for Current Target Color Point and the closest Duty Cycle Available. Sensor calibration Data should be set before using this command.	

Table 21-42. WPC Calibration Data [Opcode: C6h | Destination: 4]

Set WPC Calibration Data	
Write Parameters	
Byte	**Description**
Byte 0	LED Color 0 = Red 1 = Green 2 = Blue
Bytes 1-2	Chromatic x coordinate in u1.15 format
Bytes 3-4	Chromatic y coordinate in u1.15 format
Bytes 5-8	Luminance Y coordinate
Set WPC sensor calibration data through this command. WPC_Init() should complete successfully before invoking this command.	

Get WPC Calibration Data	
Read Parameters	
Byte	**Description**
Byte 0	LED Color 0 = Red 1 = Green 2 = Blue

Return Parameters	
Byte	**Description**
Bytes 0-1	Chromatic x coordinate in u1.15 format
Bytes 2-3	Chromatic y coordinate in u1.15 format
Bytes 4-7	Luminance Y coordinate
Bytes 8-11	Red Sensor Output
Bytes 12-15	Green Sensor Output
Bytes 16-19	Blue Sensor Output
Bytes 20-21	Duty Cycle Format = u8.8
Gets WPC sensor calibration data through this command	

Table 21-43. WPC Sensor Output [Opcode: CDh | Destination: 4]

Get WPC Sensor Output	
Return Parameters	
Byte	**Description**
Bytes 0-3	Red
Bytes 4-7	Green
Bytes 8-11	Blue
Returns Output of Integrating Sensor for Red, Blue and Green	

Table 21-44. Maximum SSI Drive Level [Opcode: CEh | Destination: 4]

Set Maximum SSI Drive Level	
Write Parameters	
Byte	**Description**
Byte 0	Color 0 = SSI SPI/PWM designator for RED 1 = SSI SPI/PWM designator for GREEN 2 = SSI SPI/PWM designator for BLUE 3 = Sample1 for SPI systems. 4 = Sample2 C2 5 = SSI SPI Yellow/ IR for PWM systems. 6 = SSI CYAN for SPI SENSE for PWM systems 7 = SSI MAGENTA for SPI / Frequency for PWM systems 8 = SSI SPI designator for WHITE 9 = SSI SPI designator for Black 10 = SSI SPI designator for IR
Bytes 1-2	Max Drive Level
Set Maximum Drive Level	

Get Maximum SSI Drive Level	
Read Parameters	
Byte	**Description**
Byte 0	Color 0 = SSI SPI/PWM designator for RED 1 = SSI SPI/PWM designator for GREEN 2 = SSI SPI/PWM designator for BLUE 3 = Sample1 for SPI systems. 4 = Sample2 C2 5 = SSI SPI Yellow/ IR for PWM systems. 6 = SSI CYAN for SPI SENSE for PWM systems 7 = SSI MAGENTA for SPI / Frequency for PWM systems 8 = SSI SPI designator for WHITE 9 = SSI SPI designator for Black 10 = SSI SPI designator for IR

Return Parameters	
Byte	**Description**
Bytes 0-1	Max Drive Level
Get Maximum Drive Level	

Table 21-45. SSI Duty Cycle Index [Opcode: CFh | Destination: 4]

Set SSI Duty Cycle Index	
Write Parameters	
Byte	**Description**
Bytes 0-1	Index
Select Duty Cycle index for SSI	

Get SSI Duty Cycle Index	
Return Parameters	
Byte	**Description**
Bytes 0-1	Number Of Duty Cycles
Bytes 2-3	Current Duty Cycle Index
Bytes 4-5	Current Red Duty Cycle Format = u8.8
Bytes 6-7	Current Green Duty Cycle Format = u8.8
Bytes 8-9	Current Blue Duty Cycle Format = u8.8
Bytes 10-11	Current Yellow Duty Cycle Format = u8.8
Bytes 12-13	Current Cyan Duty Cycle Format = u8.8
Bytes 14-15	Current Magenta Duty Cycle Format = u8.8
Bytes 16-17	Current White Duty Cycle Format = u8.8
Get Details of selected Duty Cycles and number of duty cycles available in current system look	

Table 21-46. Enable XPR Calibration Mode [Opcode: D1h | Destination: 4]

Set Enable XPR Calibration Mode	
Write Parameters	
Byte	**Description**
Byte 0	1 - Calibration mode enabled
This command sets the system in bypass mode. Setting the system in bypass mode disables any image processing to establish one to one correspondence between pixels on input source image and display image. Desirable for seeing clear splits of XPR subframes. There is no exit from calibration mode. Please restart the system.	

Get Enable XPR Calibration Mode
Data returned is in the same format as the Write Parameters.
This command gets the state of XPR calibration mode. Whether enabled or not.

Table 21-47. WPC Calibration Structure Override [Opcode: D2h | Destination: 4]

Set WPC Calibration Structure Override	
Write Parameters	
Byte	**Description**
Byte 0	LED Color 0 = Red 1 = Green 2 = Blue
Bytes 1-2	Chromatic x coordinate in u1.15 format
Bytes 3-4	Chromatic y coordinate in u1.15 format
Bytes 5-8	Luminance Y coordinate
Bytes 9-12	Red Sensor Output
Bytes 13-16	Green Sensor Output
Bytes 17-20	Blue Sensor Output
Bytes 21-22	Duty Cycle Format = u8.8
Set the entire WPC sensor calibration data structure through this command. WPC_Init() must complete successfully before invoking this command.	

21.6 Blending

Blending

Table 21-48. Blend Map Gain Values [Opcode: 2Bh | Destination: 4]

Set Blend Map Gain Values	
Write Parameters	
Byte	**Description**
Byte 0	Compressed Values Passed bit 0: 1 = Compressed data is passed below for Gain values. 0 = Uncompressed data is passed below for Gain values.
Byte 1	Color Channel Select 0 = Broadcast 1 = Green 2 = Red 3 = Blue
Bytes 2-3	Here the total number of compressed values needs to be passed only when CompressionEnable is Enabled otherwise pass 0 here
Bytes 4-5	Starting Index for filling the Gain values in array of Gain
Bytes 6 - *	Gain of control points .The format of input should be such that required gain which is a value between 0 to 1.99 be multiplied by 4096 before passing it in command.
This command takes input from the user gain values of control points as part of the Blend Map. Using this command user passes the Gain values for the 2016 (63×32) control points. The gain values needs to be specified for each of the color channel. There are two features here for using the command. User can select the broadcast values option from the Color Channel Select. Broadcast sends the same values to all the three color channels R,G,B. Otherwise pass the value for R, G, and B color channels separately. So the command will then be required to use 3 times one time for each color channel. Another feature supported is passing compressed values. The compression used is RLE2 compression. For passing compressed values user needs to enable the Compression Enabled bit.	

Get Blend Map Gain Values	
Read Parameters	
Byte	**Description**
Byte 0	Color Channel Select 0 = Broadcast 1 = Green 2 = Red 3 = Blue
Bytes 1-2	Start index in the Blend map channel gain values from which the data is to be read
Bytes 3-4	Number of entries to be read

Return Parameters	
Byte	**Description**
Byte 0	This tells whether the gain values obtained are compressed data or not.
Bytes 1 - *	Selected Color Channel gain values
This command reads from the blend map table already loaded using Set Blend Map Gain Values command. N Blend map gain values (that does not exceed the command packet size) can be read at a time from anywhere within the table.	

Table 21-49. Apply Blend Map [Opcode: 2Ch | Destination: 4]

Set Apply Blend Map	
Write Parameters	
Byte	**Description**
Byte 0	Blend Map Upload bit 0: 1 = Blend Map values is enabled and applied. 0 = Blend Map values is disabled.
This command passes the entire blend map to lower layer - API	

Table 21-50. Blend Map Offset Values [Opcode: 2Dh | Destination: 4]

Set Blend Map Offset Values	
Write Parameters	
Byte	**Description**
Byte 0	Compressed Values Passed bit 0: 1 = Compressed data is passed below for Offset values. 0 = Uncompressed data is passed below for Offset values.
Byte 1	Color Channel Select 0 = Broadcast 1 = Green 2 = Red 3 = Blue
Bytes 2-3	Here the total number of compressed values needs to be passed only when the first parameter is 1 otherwise pass 0 here
Bytes 4-5	Starting Index for filling the Offset values in array
Bytes 6 - *	Offset of control points. The format of input should be such that the offset values are in the internal floating point format of s1m8e4 .
Using this command user passes the Offset values for the 2016 (63×32) control points. The Offset values needs to be specified for each of the color channel R,G,B. The range of offset values in the standard floating point format is -255 to +255. There are two features here for using the command. User can select the broadcast values option from the Color Channel Select. Broadcast sends the same values to all three color channels R,G,B. Otherwise pass the value for R, G, and B color channels separately. So the command will then be required to use 3 times one time for each color channel. Another feature supported is passing compressed values. The compression used is RLE2 compression. For passing compressed values user needs to enable the Compression Enabled bit	

Get Blend Map Offset Values	
Read Parameters	
Byte	**Description**
Byte 0	Color Channel Select 0 = Broadcast 1 = Green 2 = Red 3 = Blue
Bytes 1-2	Start index in the Blend map channel Offset values from which the data is to be read
Bytes 3-4	Number of entries to be read

Return Parameters	
Byte	**Description**
Byte 0	This tells whether the offset values obtained are compressed data or not.
Bytes 1 - *	Selected Color Channel Offset
This command reads from the blend map compressed offset values already loaded using Set Blend Map Offset Values command. N Blend map offset values (that does not exceed the command packet size) can be read at a time from anywhere within the table.	

Table 21-51. Blend Map Control Points [Opcode: 2Eh | Destination: 4]

Set Blend Map Control Points	
Write Parameters	
Byte	**Description**
Bytes 0-1	Horizontal Display Resolution of Projector
Bytes 2-3	Vertical Display Resolution of Projector
Bytes 4 - *	Blend Map Horizontal control points position array Number of points in this array equal to 63.These control points are 0 based.
Bytes 4 - *	Blend Map Vertical control points position array Number of points in this array equal to 32.These control points are 0 based.
This command takes input of the user defined control points location in horizontal and vertical direction as part of the Blend Map	

Get Blend Map Control Points	
Return Parameters	
Byte	**Description**
Bytes 0-125	Blend map Horizontal control points position array. Number of points in this array equal to Blend Map Control Points X.
Bytes 126-189	Blend map Vertical control points position array. Number of points in this array equal to Blend Map Control Points Y.
This command gets the user defined blend map control points location stored in EEPROM.	

Table 21-52. Enable Edge Blending [Opcode: 2Fh | Destination: 4]

Set Enable Edge Blending	
Write Parameters	
Byte	**Description**
Byte 0	EBF State bit 0: 1 = Edge blending is enabled 0 = Edge blending is disabled
This command enables or disables the Edge blending function	

Get Enable Edge Blending
Data returned is in the same format as Write Parameters above.
Returns whether the Edge blending function is enabled or not.

Table 21-53. Edge Blending System Params [Opcode: 3Dh | Destination: 4]

Set Edge Blending System Params	
Write Parameters	
Byte	**Description**
Bytes 0-1	Number of columns of projectors in blending system Range = 1 to 32 with step size 1
Bytes 2-3	Number of rows of projectors in blending system Range = 1 to 32 with step size 1
Bytes 4-5	Column index of self in blending system Range = 0 to 31 with step size 1
Bytes 6-7	Row index of self in blending system Range = 0 to 31 with step size 1
Bytes 8-9	What type of photometric parameters to use when calculating gains and offsets
Bytes 10 - *	White and black luminance levels of projectors in blending system, raster scan order. Format is fixed point 16.16 (nits)
This command sets the blending system parameters for semi-manual edge blending. This command does not change the state of the warping map or the blending map	

Get Edge Blending System Params
Data returned is in the same format as Write Parameters above.
This command gets the blending system parameters for semi-manual edge blending

Table 21-54. Edge Blending Configuration [Opcode: 3Eh | Destination: 4]

Set Edge Blending Configuration	
Write Parameters	
Byte	**Description**
Bytes 0-1	Horizontal Overlap with other projectors (pixels)
Bytes 2-3	Vertical Overlap with other projectors (pixels)
Byte 4	Geometric Adjustment Type 0 = No Geometric Correction 1 = Keystone Corners
Byte 5	Storage Options 0 = Do Not Store 1 = Write To Storage 2 = Write To Storage Apply At Startup
Bytes 6 - *	Geometric adjustment parameters. Dependent on adjustment type used
This command sets overlap and geometry parameters for semi-manual edge blending, creates and applies blending and warping maps for given blending inputs. It is necessary to call commands to enable manual warping and enable edge blending separately, for the results to take effect. Geometry parameters are dependent on geometric adjustment type. For no geometric correction, no parameters are used. For correction by keystone corners, the 8 parameters are the (x,y) coordinates (zero based) of the keystone corners in raster scan order : top-left, top-right, bottom-left, bottom-right.	

Get Edge Blending Configuration
Data returned is in the same format as Write Parameters above.
This command gets geometry and overlap parameters for semi-manual edge blending. Geometry parameters are dependent on geometric adjustment type. For no geometric correction, no parameters are used. For correction by keystone corners, the 8 parameters are the (x,y) coordinates (zero based) of the keystone corners in raster scan order : top-left, top-right, bottom-left, bottom-right.

21.7 Debug Internal

Debug Internal

Table 21-55. Vx1 Hw Status [Opcode: 3Fh | Destination: 4]

Get Vx1 Hw Status	
Return Parameters	
Byte	**Description**
Byte 0	Is Source Locked bit 0: Source Locked
Byte 1	Is Bit Locked bit 0: Bit Locked
Byte 2	Is Byte Locked bit 0: Byte Locked
Byte 3	Is Data Locked bit 0: Data Locked
Byte 4	Is V Sync Stable bit 0: V Sync Stable
Byte 5	Is H Sync Stable bit 0: H Sync Stable
Bytes 6-7	Acitve Pixels Per Line (APPL) (Pixels)
Bytes 8-9	Active Lines Per Frame (ALPF) (Lines)
Bytes 10-11	Total Pixels Per Line (TPPL) Largest (Pixels)
Bytes 12-13	Total Lines Per Frame (TLPF) (Lines)
Bytes 14-15	TPPL Smallest (Pixels)
Bytes 16-17	Vertical Front Porch (VFP) (Lines)
Bytes 18-19	Vertical Back Porch (VBP) (Lines)
Bytes 20-21	Vsync Pulse Width (VSW) (Lines)
Bytes 22-23	Horizontal Front Porch (HFP) (Pixels)
Bytes 24-25	Horizontal Back Porch (HBP) (Pixels)
Bytes 26-27	Hsync Pulse Width (HSW) (Pixels)
Bytes 28-29	HSync To VSync Pixel clock count (Hs2Vs)
Bytes 30-31	VSync To HSync Pixel clock count (Vs2Hs)
Byte 32	H Sync Polarity bit 0: H Sync Polarity Is Positive
Byte 33	V Sync Polarity bit 0: V Sync Polarity Is Positive
Bytes 34-37	Freq Captured (kHz)
Reports Vx1 source HW interface status.	

21.8 Debug

Debug

Table 21-56. Memory [Opcode: 10h | Destination: 1]

Set Memory	
Write Parameters	
Byte	**Description**
Bytes 0-3	Memory Address, must be a multiple of 4.
Bytes 4-7	Value to write
This command attempts a direct write of the given 32-bit value to the given 32-bit memory address. The memory address is not verified whether it is a valid location.	

Get Memory	
Read Parameters	
Byte	**Description**
Bytes 0-3	Memory Address, must be a multiple of 4.

Return Parameters	
Byte	**Description**
Bytes 0-3	Value read from address
This command returns the 32-bit value stored at the given 32-bit memory address.	

Table 21-57. Memory Array [Opcode: 11h | Destination: 1]

Set Memory Array	
Write Parameters	
Byte	**Description**
Bytes 0-3	Start Address from which data is to be written
Byte 4	Access Info bits 0-5: Address increment steps. 0 - No increment bits 6-7: Write access width 0 = Uint32 1 = Uint16 2 = Uint08
Bytes 5-6	Number of words to be written
Byte 7	The number of bytes per word Range = 1 to 2 with step size 4
Bytes 8 - *	Data to be written
Writes a stream of words into the RAM memory (DRAM or IRAM) starting from the address specified. Performs no checks whether the specified memory address given is valid.	

Get Memory Array	
Read Parameters	
Byte	**Description**
Bytes 0-3	Start Address from which data is to be read
Byte 4	Access Info bits 0-5: Address increment steps. 0 - No increment bits 6-7: Read access width 0 = Uint32 1 = Uint16 2 = Uint08
Bytes 5-6	Number of words to be read
Byte 7	The number of bytes per word Range = 1 to 4 with step size 1

Return Parameters	
Byte	**Description**
Bytes 0 - *	Data
Reads a stream of words from memory starting from the address specified. Performs no checks whether the specified memory address given is valid.	

Table 21-58. Debug Message Mask [Opcode: E0h | Destination: 4]

Set Debug Message Mask	
Write Parameters	
Byte	**Description**
Bytes 0-3	Debug Mask bits 0-10: Reserved bit11: Communication related bit13: 3D bit14: RFC messaging bit15: I2C Handler bit17: Closed Captioning bit18: Reserved bit19: GUI bit20: Environment bit21: Illumination bit22: System functions bit23: EEPROM bit24: Datapath bit25: Autolock bit26: Projector Control bit27: Peripheral bit28: IR bit29: USB bit30: Mailbox
Set enable mask for debug messages. The mask identifies the sources of debug messages which are to be enabled for printing at the UART debug port. The mask bit corresponding to the source has to be set to enable it.	

Get Debug Message Mask	
Return Parameters	
Byte	**Description**
Bytes 0-3	Debug Mask bit11: Communication related bit13: 3D bit14: RFC messaging bit15: I2C Handler bit17: Closed Captioning bit18: DDC CI bit19: GUI bit20: Environment bit21: Illumination : DMD, wheel, lamp bit22: System functions bit23: EEPROM bit24: Datapath bit25: Autolock bit26: Projector Control bit27: Peripheral bit28: IR bit29: USB bit30: Mailbox
Retrieves the current debug message mask. The mask decides which sources of debug messages are enabled. A value of 1 in the mask bit corresponding to a source means that the source is enabled.	

Table 21-59. Enable USB Debug Log [Opcode: E1h | Destination: 4]

Set Enable USB Debug Log	
Write Parameters	
Byte	**Description**
Byte 0	1 = Enable debug log on USB port 0 = Disable debug log on USB port
Enables or disables the USB logging of messages. When USB logging is enabled, UART logging is stopped.	

Table 21-60. EEPROM Memory [Opcode: E2h | Destination: 4]

Set EEPROM Memory	
Write Parameters	
Byte	**Description**
Bytes 0-1	Index of the memory to write (0 means first byte in EEPROM, 1 means second byte and so on)
Bytes 2-3	Number of bytes to be written
Bytes 4-7	EEPROM password if the user needs to write to the TI specified memory space
Bytes 8- Number of bytes passed	Data
Writes data to EEPROM connected to the controller. The EEPROM holds settings and calibration data. The primary purpose of this function is for the user to write to areas of EEPROM outside of the settings and calibration data. If user wants to overwrite settings/calibration data, the password sent with the command should match the expected password. This is a protection mechanism to prevent accidental overwrite of settings/calibration data.	

Get EEPROM Memory	
Read Parameters	
Byte	**Description**
Bytes 0-1	Index of the memory to read from
Bytes 2-3	Number of bytes to read

Return Parameters	
Byte	**Description**
Bytes 0- Number of bytes passed	Data
This function reads data from EEPROM connected to the Controller which has settings and calibration data. Note : EEPROM data can be read without unlocking.	

Table 21-61. TI Actuator Interface Debug [Opcode: E4h | Destination: 4]

Set TI Actuator Interface Debug	
Write Parameters	
Byte	**Description**
Byte 0	Query type 0 = Query N number of bytes from offset address provided in next two bytes, Bytes 1-2 1 = Query Actuator information also print on UART debug port 2 = Query AWG Data Set for index number provided in next two bytes, Bytes 1-2 3 = Query AWG Edge table header for index number in next two bytes, Bytes 1-2
Bytes 1-2	Query type provided in Byte 0; not applicable when Query type = 1
Bytes 3-4	Number of bytes to be read when Query type = 0. (Note : Maximum 32 bytes can be read at a time.)
Command used to query actuator related information for debugging purpose. Use this command to retrieve information when actuator not running or system is in standby state.	

Get TI Actuator Interface Debug	
Return Parameters	
Byte	**Description**
Bytes 0-31	Actuator Data
Command returns queried data as per the settings made in the set command	

Table 21-62. Vsync Period [Opcode: E5h | Destination: 4]

Get Vsync Period	
Return Parameters	
Byte	**Description**
Bytes 0-3	Frame period in micro seconds
Returns the latest VSync period measurement	

Table 21-63. DMD Power [Opcode: E8h | Destination: 4]

Get DMD Power	
Return Parameters	
Byte	**Description**
Byte 0	Enable State bit0: 0 = Disable; 1 = Enable
Returns DMD power enable state	

Table 21-64. DMD Park [Opcode: E9h | Destination: 4]

Set DMD Park	
Write Parameters	
Byte	**Description**
Byte 0	Park State bit0: 0 = Unpark; 1 = Park
Parks/Unparks DMD	

Get DMD Park
Data returned is in the same format as the Write Parameters.
Returns 1 if DMD is Parked, else returns 0

Table 21-65. DMD True Global Reset [Opcode: EBh | Destination: 4]

Set DMD True Global Reset	
Write Parameters	
Byte	**Description**
Byte 0	True Global Mode bit0: 0 = True Global Reset Mode Disabled; 1 = True Global Reset Mode Enabled.
The TrueGlobalMode should be set to TRUE only during factory/assembly operation.	

Get DMD True Global Reset
Data returned is in the same format as the Write Parameters.

Table 21-66. Int Stack [Opcode: F0h | Destination: 4]

Get Int Stack	
Return Parameters	
Byte	**Description**
Bytes 0-3	Stack Size
Bytes 4-7	Stack Used
Bytes 8-11	Stack Free
Gives the current stack usage information	

Table 21-67. Print All Task Information [Opcode: F1h | Destination: 4]

Set Print All Task Information
Write Parameters
Prints(on UART) information of all tasks defined/created with RTOS.

Table 21-68. Resource [Opcode: F2h | Destination: 4]

Get Resource	
Return Parameters	
Byte	**Description**
Byte 0	Tasks High Count
Byte 1	Events High Count
Byte 2	Group Events High Count
Byte 3	Mailbox High Count
Byte 4	Memory Pools High Count
Byte 5	Semaphore High Count
Byte 6	Tasks Current Count
Byte 7	Events Current Count
Byte 8	Group Events Current Count
Byte 9	Mailbox Current Count
Byte 10	Memory Pools Current Count
Byte 11	Semaphore Current Count
Gives the maximum RTOS resource usage by the application.	

Table 21-69. EEPROM Free Area Offset [Opcode: FFh | Destination: 4]

Get EEPROM Free Area Offset	
Return Parameters	
Byte	**Description**
Bytes 0-1	Free Area Offset
This function idicates the EEPROM address offset which corresponds to the start of free area.	

21.9 General Operation

General Operation

Table 21-70. Power [Opcode: 10h | Destination: 4]

Set Power	
Write Parameters	
This commands toggles current power mode from standby to active or from active to power down. The Standby state corresponds to Low Power Mode.	

Get Power	
Return Parameters	
Byte	**Description**
Byte 0	Power State 0 = Reset 1 = Standby 2 = Active 3 = Cooling 4 = Warming
Returns current system power state.	

Table 21-71. Display [Opcode: 11h | Destination: 4]

Set Display	
Write Parameters	
Byte	**Description**
Byte 0	Source 0 = Display External 1 = Test Pattern 2 = Solid Field 3 = Splash 4 = Curtain
Displays the specified source. Note : If Display External projection mode is selected and if there is no source present it will show Splash or Solid Field depending on the default settings in the system.	

Get Display
Data returned is in the same format as the Write Parameters.
Returns the source which is currently being displayed.

Table 21-72. Enable Low Latency Mode [Opcode: 12h | Destination: 4]

Set Enable Low Latency Mode	
Write Parameters	
Byte	**Description**
Byte 0	Enable State bit0: 1 = Low latency mode is enabled, 0 = Low latency mode is disabled
Enables or disables the Low latency mode of operation in which processing delay (from the input source to the frame sent to DMD) by the Controller is limited to a maximum of one and a half frame delays.	

Get Enable Low Latency Mode
Data returned is in the same format as the Write Parameters.
Returns whether low latency mode is enabled or not.

Table 21-73. System Look [Opcode: 13h | Destination: 4]

Set System Look	
Write Parameters	
Byte	**Description**
Bytes 0-1	Look Index
This command sets the current system look. System looks shall be designed and configured via DLP Composer tool. System look determines the current group of sequences and color points to be loaded. This command also initiates the source definition change that corresponds to new look index.	

Get System Look
Data returned is in the same format as the Write Parameters.
This command gets the current system look.

Table 21-74. TPG Predefined Pattern [Opcode: 14h | Destination: 4]

Set TPG Predefined Pattern	
Write Parameters	
Byte	**Description**
Byte 0	Predefined test pattern number to be displayed
This command will set one of the pre-defined test patterns stored in flash (configured using DLP Composer tool). The function selects a pattern to load from flash into the test pattern generator hardware. The information retrieved from the flash includes pattern definition, color definition, and the resolution. The pre-defined patterns are included in the flash configuration data. Set Display command must be called to switch the display mode from other modes to TPG prior to or after this command .	

Get TPG Predefined Pattern
Data returned is in the same format as the Write Parameters.
Returns the current selection for pre-defined test patterns.

Table 21-75. TPG Border [Opcode: 15h | Destination: 4]

Set TPG Border	
Write Parameters	
Byte	**Description**
Byte 0	Width of the Border Range = 0 to 20 with step size 1
Bytes 1-2	Border Color Red Value Range = 0 to 1023 with step size 1
Bytes 3-4	Border Color Green Value Range = 0 to 1023 with step size 1
Bytes 5-6	Border Color Blue Value Range = 0 to 1023 with step size 1
Draws a border around the test pattern of given width and color. Note : To be used only when the Display is set as Test Pattern.	

Get TPG Border
Data returned is in the same format as the Write Parameters.
Returns Width in number of pixels and Color of Border for a test Pattern.

Table 21-76. TPG Resolution [Opcode: 16h | Destination: 4]

Set TPG Resolution	
Write Parameters	
Byte	**Description**
Bytes 0-1	Horizontal resolution of test pattern (Pixels) Range = 640 to 4096 with step size 1
Bytes 2-3	Vertical resolution of test pattern (Lines) Range = 480 to 2400 with step size 1
Sets horizontal and vertical resolution in number of pixels for current test pattern.	

Get TPG Resolution
Data returned is in the same format as the Write Parameters.
Returns horizontal and vertical resolution in number of pixels for current test pattern.

Table 21-77. TPG Frame Rate [Opcode: 17h | Destination: 4]

Set TPG Frame Rate	
Write Parameters	
Byte	**Description**
Byte 0	Frame rate of test pattern(Hz) Range = 30 to 240 with step size 1
Sets frame rate in Hz for current test pattern.	

Get TPG Frame Rate
Data returned is in the same format as the Write Parameters.
Returns frame rate in Hz for current test pattern.

Table 21-78. SFG Color [Opcode: 18h | Destination: 4]

Set SFG Color	
Write Parameters	
Byte	**Description**
Bytes 0-1	Red color level. Range = 0 to 1023 with step size 1
Bytes 2-3	Green color level. Range = 0 to 1023 with step size 1
Bytes 4-5	Blue color level. Range = 0 to 1023 with step size 1
Configures the solid color to be displayed when display is set to solid field generator (SFG). This command only sets the SFG color and does NOT display it. In order to display the SFG, Display needs to be set with SFG as source(Use Set Display command for this.)	

Get SFG Color
Data returned is in the same format as the Write Parameters.
Returns the solid color which is programmed to be displayed when display is set to SFG.

Table 21-79. SFG Resolution [Opcode: 19h | Destination: 4]

Get SFG Resolution	
Return Parameters	
Byte	**Description**
Bytes 0-1	Horizontal resolution of SFG(Pixels) Range = 0 to 4096 with step size 1
Bytes 2-3	Vertical resolution of SFG(Lines) Range = 0 to 2160 with step size 1
Gets the resolution of the displayed SFG image.	

Table 21-80. Curtain Color [Opcode: 1Ah | Destination: 4]

Set Curtain Color	
Write Parameters	
Byte	**Description**
Byte 0	The background color to be set as curtain. 0 = Black color 1 = Reserved 2 = White color 3 = Green color 4 = Red color 5 = Blue color 6 = Yellow color 7 = Cyan color 8 = Magenta color 9 = Reserved 10 = Reserved
Command to set the color to be used in curtain mode. Use Set Display command to switch to curtain mode.	

Get Curtain Color
Data returned is in the same format as the Write Parameters.
Command that returns the color used in curtain mode.

Table 21-81. Splash Load Image [Opcode: 1Bh | Destination: 4]

Set Splash Load Image	
Write Parameters	
Byte	**Description**
Byte 0	The 0-based index of Splash Image (0xff for captured splash). Range = 0 to 255 with step size 1
Sets the index of the splash image to be loaded and displayed. If already in Splash mode the requested splash image is displayed.	

Get Splash Load Image
Data returned is in the same format as the Write Parameters.
Gets the index of the splash image to be loaded and displayed.

Table 21-82. Enable Image Flip [Opcode: 1Ch | Destination: 4]

Set Enable Image Flip	
Write Parameters	
Byte	**Description**
Byte 0	Flip bit0: 0 = Vertical Flip of the image is disabled; 1 = Vertical Flip of the image is enabled. bit1: 0 = Horizontal Flip of the image is disabled; 1 = Horizontal Flip of the image is enabled.
Flips the data output to the display vertically or horizontally. This feature is provided to support use cases like ceiling mount, rear projection etc.	

Get Enable Image Flip
Data returned is in the same format as the Write Parameters.
Returns whether image flipping is enabled.

Table 21-83. Enable Freeze [Opcode: 1Dh | Destination: 4]

Set Enable Freeze	
Write Parameters	
Byte	**Description**
Byte 0	Freeze State bit0: 0 = Display freeze is disabled; 1 = Display freeze is enabled.
It enables or disables display freeze which freezes the current frame being displayed on the screen. Caution : Set Curtain or any operation that requires curtain will override Freeze and frozen image on the wall will be lost. The following operations require curtain (and will override Freeze) : Source Type Switch (Standard - XPR - 3D) Source Type Switch (interlaced - non-interlaced) Switch to Splash Display Splash Capture Low Latency Mode Switch Source Relocking Switch to Stand-By/Low-Power mode	

Get Enable Freeze
Data returned is in the same format as the Write Parameters.
Returns whether the current display is frozen.

Table 21-84. Keystone Angles [Opcode: 1Eh | Destination: 4]

Set Keystone Angles	
Write Parameters	
Byte	**Description**
Bytes 0-1	Pitch angle in degrees Range = -128 to 127.9960375 with step size 0.00390625 Format = s8.8
Bytes 2-3	Yaw angle in degrees Set to 0 for 1D correction Range = -128 to 127.9960375 with step size 0.00390625 Format = s8.8
Bytes 4-5	Roll angle in degrees Set to 0 for 1D/2D correction Range = -128 to 127.9960375 with step size 0.00390625 Format = s8.8
Configures the Keystone correction when the pitch, yaw, roll, throw ratio and vertical offset of corrected image are known. Keystone correction is used to remove the distortion caused when the projector is not orthogonal to the projection surface (screen). Keystone feature will be automatically enabled when this command is executed. Note : The actual range of these parameters depends on the light engine (projection optics); the range of Pitch, Yaw and Roll is derived from optical engine Vertical offset and Throw Ratio.(Maximum range : -40 to +40 degrees)	

Get Keystone Angles
Data returned is in the same format as the Write Parameters.
Returns the keystone configuration parameters currently set.

Table 21-85. Keystone Config Override [Opcode: 1Fh | Destination: 4]

Set Keystone Config Override	
Write Parameters	
Byte	**Description**
Bytes 0-1	Throw Ratio Format = u8.8
Bytes 2-3	Vertical Offset Format = s8.8
Bytes 4-5	Horizontal Offset Format = s8.8

Get Keystone Config Override
Data returned is in the same format as the Write Parameters.

Table 21-86. Enable Anamorphic Scaling [Opcode: 20h | Destination: 4]

Set Enable Anamorphic Scaling	
Write Parameters	
Byte	**Description**
Byte 0	Enable State bit0: 0 = Anamorphic Scaling is disabled; 1 = Anamorphic Scaling is enabled.
Enables or disables the anamorphic scaling	

Get Enable Anamorphic Scaling
Data returned is in the same format as the Write Parameters.
Returns whether anamorphic scaling is enabled or not.

Table 21-87. Display Image Size [Opcode: 21h | Destination: 4]

Set Display Image Size	
Write Parameters	
Byte	**Description**
Byte 0	Image Size Type 0 = Fill (uses DMD image size) 1 = Native (same as source size) 2 = Manual 3 = Image size maintains the aspect ratio of source and fills the DMD in at least one direction 4 = Image size maintains the aspect ratio of 5 = Image size maintains the aspect ratio of
Bytes 1-2	Cropped Area First Pixel
Bytes 3-4	Cropped Area First Line
Bytes 5-6	Cropped Area Pixels Per Line
Bytes 7-8	Cropped Area Lines Per Frame
Bytes 9-10	Display Area First Pixel
Bytes 11-12	Display Area First Line
Bytes 13-14	Display Area Pixels Per Line
Bytes 15-16	Display Area Lines Per Frame
Configures the cropping of input image and resizing of image that is displayed. Cropped area can be equal to or less than the input image size. The display area has to be within DMD effective number of pixels and lines. Note : 1. Cropped Area and Display Area parameters are valid only when image size type is set to Manual. 2. For TPG, SFG and Splash, Cropped Area parameter is ignored. For those sources, cropped area is automatically set as explained below : a. For TPG, cropped area is set to TPG resolution. b. For Splash, cropped area is set to Splash image size. c. For SFG, cropped area is set to SFG resolution which is equal to source area of last stable external source or TPG.	

Get Display Image Size
Data returned is in the same format as the Write Parameters.
Returns current image size, cropping and display settings.

Table 21-88. Source Configuration [Opcode: 22h | Destination: 4]

Set Source Configuration	
Write Parameters	
Byte	**Description**
Byte 0	0 = Input port sync is not modified(passed through). 1 = Input port sync is inverted. 2 = ALF Sync is selected as the Port sync source. Use when Autolock is used for source detection. 3 = Applicable for Topfield only. The TopField is decoded from HSync and VSync.
Byte 1	H Sync Configuration 0 = Input port sync is not modified(passed through). 1 = Input port sync is inverted. 2 = ALF Sync is selected as the Port sync source. Use when Autolock is used for source detection. 3 = Applicable for Topfield only. The TopField is decoded from HSync and VSync.
Byte 2	Top Field Configuration 0 = Input port sync is not modified(passed through). 1 = Input port sync is inverted. 2 = ALF Sync is selected as the Port sync source. Use when Autolock is used for source detection. 3 = Applicable for Topfield only. The TopField is decoded from HSync and VSync.
Byte 3	Down Sample Configuration - Configures the HW for downsampling. When downsampling is enabled, the PixelClockFreqInkHz is halved by the ASIC. This affects other parameters in the SRC structure such as TotalArea.PixelsPerLine, ActiveArea.PixelsPerLine and ActiveArea.FirstPixel. 0 = Down Sample Operation disabled (data pass through unmodified). 1 = Down Sample Operation enabled. Select First Data Sample Positions from Sample Position Reference. 2 = Down Sample Operation enabled. Select Second Data Sample Positions from Sample Position Reference.
Byte 4	3D Enable bit0: 0 = 3D Disabled 1 = 3D Enabled
Byte 5	Clock Polarity bit0: 0 = Data is clocked in on falling edge of the port clock 1 = Data is clocked in on rising edge of the port clock
Byte 6	Pixel Format 0 = Rgb 1 = Yuv444 2 = Yuv422 3 = Yuv420
Byte 7	External Data Enable bit0: 0 = External Data Enable is not used 1 = External Data Enable is used (typical for digital sources)
Byte 8	Interlaced bit0: 0 = Non Interlaced 1 = Interlaced
Byte 9	Offset Binary bit0: 0 = Incoming data is signed 2s complement; typical for RGB sources 1 = offset binary; typical for YUV sources
Byte 10	Top Field Inverted - Applicable only for interlaced sources that use field dependent scaling. Set to 0 for analog interlaced Graphics. Set to 1 for DVI Sources. bit0: 0 = Top field not inverted at scaler 1 = Top field inverted at scaler
Bytes 11-12	Total Area Pixels Per Line
Bytes 13-14	Total Area Lines Per Frame

Table 21-88. Source Configuration [Opcode: 22h | Destination: 4] (continued)

Set Source Configuration	
Bytes 15-16	Active Area First Pixel
Bytes 17-18	Active Area First Line
Bytes 19-20	Active Area Pixels Per Line
Bytes 21-22	Active Area Lines Per Frame
Bytes 23-24	Bottom Field First Line - Applicable for Interlaced Sources only. This term specifies the first(starting) active line in the Bottom Field. Valid range is TopFieldFirstLine to Active Number of Lines per Frame. For Field dependent framing, BottomFieldFirstLine >= TopFieldFirstLine (= ActiveArea.FirstLine)
Bytes 25-28	Pixel Clock Freq In kilohertz
Bytes 29-30	Color Space Conversion coefficient 0 - Coefficients used for converting YUV sources to RGB. For RGB Sources, this should be an Identity Matrix. All coefficients are defined to be signed, 2's complement values with 2 significant bits and 10 fractional bits (s2.10). For example, 1.0 = 0x0400.
Bytes 31-32	Color Space Conversion Coefficients 1
Bytes 33-34	Color Space Conversion Coefficients 2
Bytes 35-36	Color Space Conversion Coefficients 3
Bytes 37-38	Color Space Conversion Coefficients 4
Bytes 39-40	Color Space Conversion Coefficients 5
Bytes 41-42	Color Space Conversion Coefficients 6
Bytes 43-44	Color Space Conversion Coefficients 7
Bytes 45-46	Color Space Conversion Coefficients 8
Bytes 47-48	Offset Red - Also referred to as Black Level Adjustment. Range : -256 to 255.75 in signed 8.2 format(sign + 8 integer and 2 fractional bits). Adjusts the black level for the removal of HW induced bias and/or a pedestal embedded in the Source. For changing only the offset, call the SetImageOffset command.
Bytes 49-50	Offset Green
Bytes 51-52	Offset Blue
Byte 53	Is Video
Byte 54	Is High Definition Video
Bytes 55-58	Frame Rate Range = 0 to 65536 with step size 0.00390625 Format = u16.16
Configures the characteristics of the source on the Current active port. Note : 1. After sending CMD_SetSourceConfiguration command, CMD_SetDisplayImageSize command must be sent for the changes to take effect. 2. CSC will take effect only after sending the CMD_SetDisplayImageSize command. 3. CMD_SetSourceConfiguration command should not be used when the Display is set as TPG.	

Get Source Configuration	
Return Parameters	
Byte	**Description**
Byte 0	0 = Input port sync is not modified(passed through). 1 = Input port sync is inverted. 2 = ALF Sync is selected as the Port sync source. Use when Autolock is used for source detection. 3 = Applicable for Topfield only. The TopField is decoded from HSync and VSync.
Byte 1	H Sync Configuration 0 = Input port sync is not modified(passed through). 1 = Input port sync is inverted. 2 = ALF Sync is selected as the Port sync source. Use when Autolock is used for source detection. 3 = Applicable for Topfield only. The TopField is decoded from HSync and VSync.

Get Source Configuration	
Byte 2	Top Field Configuration 0 = Input port sync is not modified(passed through). 1 = Input port sync is inverted. 2 = ALF Sync is selected as the Port sync source. Use when Autolock is used for source detection. 3 = Applicable for Topfield only. The TopField is decoded from HSync and VSync.
Byte 3	Down Sample Configuration 0 = Down Sample Operation disabled (data pass through unmodified). 1 = Down Sample Operation enabled. Select First Data Sample Positions from Sample Position Reference. 2 = Down Sample Operation enabled. Select Second Data Sample Positions from Sample Position Reference.
Byte 4	3D Enable bit0: 0 = 3D Disabled 1 = 3D Enabled
Byte 5	Clock Polarity bit0: 0 = Clock Polarity Negative 1 = Clock Polarity Positive
Byte 6	Pixel Format 0 = Rgb 1 = Yuv444 2 = Yuv422 3 = Yuv420
Byte 7	External Data Enable bit0: 0 = External Data Disabled 1 = External Data Enabled
Byte 8	Interlaced bit0: 0 = Non Interlaced 1 = Interlaced
Byte 9	Offset Binary bit0: 0 = Incoming data is signed 2s complement; typical for RGB sources 1 = offset binary; typical for YUV sources
Byte 10	Top Field Inverted - Applicable only for interlaced sources that use field dependent scaling. Set to 0 for analog interlaced Graphics. Set to 1 for DVI Sources. bit0: 0 = Top field not inverted at scaler 1 = Top field inverted at scaler
Bytes 11-12	Total Area Pixels Per Line
Bytes 13-14	Total Area Lines Per Frame
Bytes 15-16	Active Area First Pixel
Bytes 17-18	Active Area First Line
Bytes 19-20	Active Area Pixels Per Line
Bytes 21-22	Active Area Lines Per Frame
Bytes 23-24	Bottom Field First Line - Applicable for Interlaced Sources only. This term specifies the first(starting) active line in the Bottom Field. Valid range is TopFieldFirstLine to Active Number of Lines per Frame. For Field dependent framing, BottomFieldFirstLine >= TopFieldFirstLine (= ActiveArea.FirstLine)
Bytes 25-28	Pixel Clock Freq In Kilo Hertz
Bytes 29-30	Color Space Conversion Coefficients 0
Bytes 31-32	Color Space Conversion Coefficients 1
Bytes 33-34	Color Space Conversion Coefficients 2
Bytes 35-36	Color Space Conversion Coefficients 3

Get Source Configuration	
Bytes 37-38	Color Space Conversion Coefficients 4
Bytes 39-40	Color Space Conversion Coefficients 5
Bytes 41-42	Color Space Conversion Coefficients 6
Bytes 43-44	Color Space Conversion Coefficients 7
Bytes 45-46	Color Space Conversion Coefficients 8
Bytes 47-48	Offset Red - Also referred to as Black Level Adjustment. Range : -256 to 255.75 in signed 8.2 format(sign + 8 integer and 2 fractional bits). Adjusts the black level for the removal of HW induced bias and/or a pedestal embedded in the Source. For changing only the offset, call the SetImageOffset command.
Bytes 49-50	Offset Green
Bytes 51-52	Offset Blue
Byte 53	Is Video
Byte 54	Is High Definition Video
Bytes 55-58	Frame Rate Range = 0 to 65536 with step size 0.00390625 Format = u16.16
Retrieves the source characteristics for the current active port.	

Table 21-89. Datapath Scan Status [Opcode: 25h | Destination: 4]

Get Datapath Scan Status	
Return Parameters	
Byte	**Description**
Byte 0	Scan Status 0 = Detect Stable Video 1 = Searching 2 = Sync detected 3 = Locked 4 = Suspended
Byte 1	Datapath State 0 = Standby 1 = Initializing 2 = Splash At Startup 3 = Idling 4 = Scanning 5 = Autolock 6 = Monitoring
Returns Current status of source detection.	

Table 21-90. Frame Rate Parms [Opcode: 26h | Destination: 4]

Get Frame Rate Parms	
Return Parameters	
Byte	**Description**
Bytes 0-3	Input Frame Rate Format = u16.16
Bytes 4-7	Output Frame Rate Format = u16.16
Byte 8	FRC Mode 0 = Fixed output frame rate range of 47-63Hz. 1 = FRC in sync with the incoming frame rate. 2 = FRC doubles the incoming frame rate. 3 = FRC triples the incoming frame rate. 4 = FRC 4 X incoming frame rate. 5 = FRC 6 X incoming frame rate. 6 = FRC 8 X incoming frame rate. 7 = FRC 10 X incoming frame rate.
Returns current Input Frame Rate, Output Frame rate and FRC mode	

Table 21-91. XPR Enable Mode Command [Opcode: 29h | Destination: 4]

Set XPR Enable Mode Command	
Write Parameters	
Byte	**Description**
Byte 0	Mode 0 = Decision to enable XPR is based on source resolution 1 = XPR should always be turned on 2 = XPR should always be turned off

Get XPR Enable Mode Command
Data returned is in the same format as the Write Parameters.

Table 21-92. VBO Configuration [Opcode: 30h | Destination: 4]

Set VBO Configuration	
Write Parameters	
Byte	**Description**
Byte 0	Data Map Mode 0 = 36bpp/30bpp RGB/YCbCr444 1 = 27bpp RGB/YCbCr444 2 = 24bpp RGB/YCbCr444 3 = 32bpp/24bpp/20bpp YCbCr422 4 = 18bpp YCbCr422 5 = 16bpp YCbCr422 6 = 12bpp/10bpp YCbCr420 Config 1 7 = 8bpp YCbCr420 Config 1 8 = 10bpp YCbCr420 Config 2 9 = 8bpp YCbCr420 Config 2 10 = Not a valid V-by-one data mode or mode is not used
Byte 1	Byte Mode 1 = 8bit mode (=3Byte mode) 2 = 10bit mode (=4Byte mode) 3 = 12bit mode (=5Byte mode)(12bit mode is reduced internally for 10bit processing)
Byte 2	Number of lanes can be 1 or 2 or 4 or 8
Byte 3	Enable Pixel Repeat bit0: Enable Pixel Repeat
Configures the characteristics of the Vx1 source.	

Get VBO Configuration
Data returned is in the same format as the Write Parameters.
Returns the characteristics of the Vx1 source.

Table 21-93. Fpd Configuration [Opcode: 31h | Destination: 4]

Set Fpd Configuration	
Write Parameters	
Byte	**Description**
Byte 0	Fpd Mode 0 = 30-bit Mode 0 1 = 30-bit Mode 1 2 = 30-bit Mode 2 3 = 24-bit Mode 0 4 = 24-bit Mode 1 5 = Not a valid FPD-Link data mode or mode is not used
Byte 1	Data Interface Mode 0 = Single Port (A) FPD-link interface 1 = Single Port (B) FPD-link interface 2 = Dual-Port FPD (AB) if PortA carries Even and PortB carries Odd data 3 = Dual-Port FPD (AB) if PortA carries Odd and PortB carries Even data 4 = Invalid input type
Byte 2	3D L/R Reference (Enable FPD Port data bit or force 0) bit0: Enable 3D Ref
Byte 3	Field 3D Enable (Enable FPD Port data bit or force 0) bit0: Enable Field
Byte 4	Pixel Repeat bit0: 1 = Enable pixel repeat mode 0 = Disable Pixel Repeat mode
Configures the characteristics of the FPD source.	

Get Fpd Configuration
Data returned is in the same format as the Write Parameters.
Returns the characteristics of the FPD source.

Table 21-94. Keystone Corners [Opcode: 3Ah | Destination: 4]

Set Keystone Corners	
Write Parameters	
Byte	**Description**
Bytes 0-1	X position of the top left corner
Bytes 2-3	Y position of the top left corner
Bytes 4-5	X position of the top right corner
Bytes 6-7	Y position of the top right corner
Bytes 8-9	X position of the bottom left corner
Bytes 10-11	Y position of the bottom left corner
Bytes 12-13	X position of the bottom right corner
Bytes 14-15	Y position of the bottom right corner
Configures the 2D Keystone correction when the corners of the corrected image are known. Keystone correction is used to remove the distortion caused when the projector is not orthogonal to the projection surface (screen). For the effects to take place, the Keystone feature has to be enabled.	

Get Keystone Corners
Data returned is in the same format as the Write Parameters.
Returns the keystone configuration parameters currently set. This command should be used when the keystone correction has been configured using the four corners of the corrected image. The keystone correction is observed only if the keystone feature is enabled, even if the parameters are configured correctly.

Table 21-95. Warp Timing Validation Enable Adjust Wrp [Opcode: 3Bh | Destination: 4]

Set Warp Timing Validation Enable Adjust Wrp	
Write Parameters	
Byte	**Description**
Byte 0	Enable state bit0: 1 = Automatic warp geometry adjustment is enabled 0 = Automatic warp geometry adjustment is disabled
This commands sets whether automatic warp geometry adjustment should be allowed or not.	

Get Warp Timing Validation Enable Adjust Wrp
Data returned is in the same format as the Write Parameters.
Returns whether Automatic Warp Adjustment is enabled or not.

Table 21-96. Is Warp Geometry Modified [Opcode: 3Ch | Destination: 4]

Get Is Warp Geometry Modified	
Return Parameters	
Byte	**Description**
Byte 0	Anonymous 1 bit0: 1 = True 0 = False
Returns whether the Warp geometry was modified or not.	

21.10 Illumination

Illumination

Table 21-97. Illumination Enable [Opcode: 80h | Destination: 4]

Set Illumination Enable	
Write Parameters	
Byte	**Description**
Byte 0	0 - Disabled 1 - Only Red LED Enabled 2 - Only Green LED Enabled 3 - Red and Green LEDs Enabled 4 - Only Blue LED Enabled 5 - Red and Blue LEDs Enabled 6 - Green and Blue LEDs Enabled 7 - All LEDs Enabled
Enables or Disables the illumination system. Additionally, For Systems with color wheel : - Illumination will be turned on only if the system has indicated that the color wheel is spinning This interlock is necessary to avoid burning the coatings off the glass For Lamp Based Systems with Lamp Strike Reset Enabled : - Only Illumination Disable is supported during run time To Enable illumination again, system should be powered down and then powered up again This is necessary to protect DMD from EMI generated during lamp strike event	

Get Illumination Enable
Data returned is in the same format as the Write Parameters.
Gets the enable state of lamp/SSI illumination.

Table 21-98. SSI Drive Levels [Opcode: 8Ch | Destination: 4]

Set SSI Drive Levels	
Write Parameters	
Byte	**Description**
Byte 0	Applicable only for PWM based LED driver
Bytes 1-2	Drive Level Red
Bytes 3-4	Drive Level Green
Bytes 5-6	Drive Level Blue
Bytes 7-8	Drive Level C1
Bytes 9-10	Drive Level C2
Bytes 11-12	Drive Level Sense
Sets the drive current of all the channels of the SSI based illumination. Applicable for DLPA3005 or PWM based LED drivers Command should not be used if Dynamic Black or White Point Correction is enabled	

Get SSI Drive Levels	
Read Parameters	
Byte	**Description**
Byte 0	Applicable only for PWM based LED driver

Return Parameters	
Byte	**Description**
Bytes 0-1	Drive Level Red
Bytes 2-3	Drive Level Green
Bytes 4-5	Drive Level Blue
Bytes 6-7	Drive Level C1
Bytes 8-9	Drive Level C2
Bytes 10-11	Drive Level Sense
Gets the drive current of all the channels of the SSI based illumination. Applicable for DLPA3005 or PWM based LED drivers	

21.11 Image Processing

Image Processing

Table 21-99. Image Algorithm Enable [Opcode: 40h | Destination: 4]

Set Image Algorithm Enable	
Write Parameters	
Byte	**Description**
Byte 0	Chroma Transient Improvement Enable bit 0: Chroma Transient Improvement Enable Bit
Byte 1	Gamma Correction Enable bit 0: Gamma Correction Enable Bit
Byte 2	Color Coordinate Adjustment Enable bit 0: Color Coordinate Adjustment Enable Bit
Byte 3	Brilliant Color Enable bit 0: Brilliant Color Enable Bit
Byte 4	White Point Correction Enable bit 0: White Point Correction Enable Bit
Byte 5	Dynamic Black Enable bit 0: Dynamic Black Enable Bit
Byte 6	HDR Enable bit 0: HDR Enable Bit

Table 21-99. Image Algorithm Enable [Opcode: 40h | Destination: 4] (continued)

Set Image Algorithm Enable
Sets enable flag for all Image Algorithms. 0 = Disable 1 = Enable **Chroma Transient Improvement** : This function enables/disables the Chroma Transient Improvement (CTI) function which filters the 4 : 4 : 4 sampled, chrominance (Cr and Cb) data on the B and C data channels. The chroma transient functions performs band pass filtering (supports two center frequencies) and median filtering for ringing minimization. It performs limiting and coring functions for the filtered output. **Gamma Correction** : This function enables/disables the Gamma Correction function which implements the removal of gamma transfer function applied at the source, via table lookup process called de-gamma. When enabled, perform de-gamma translation of the 10-bit RGB input to the common 12-bit floating point (S0M8E4) RGB output. When disabled, the full 10 bits of each data input to the Gamma Correction function are zero padded and MSB-aligned to 12-bits and passed through unmodified. **Color Coordinate Adjustment** : This function enables/disables the Spatially Adaptive Seven Primaries Color Correction Function Enable. When Disable forces 3x3 CSC (Color Space Conversion) with identity. **Brilliant Color** : This function enables/disables the BrilliantColor technology, Brilliant Color uses up to five colors, instead of just the three primary colors, red, green and blue, to improve color accuracy and brightens of secondary colors. This results in a new level of color performance that increases the brightness of the colors. **White Point Correction** : This function enables/disables the White Point Correction, typically used on LED type illumination systems. Sometimes due to increase in LED operating temperature or LED aging the LEDs output wavelength drifts, therefore white point of the system shifts. This algorithm using active light sensor feedback and factory calibrated values help maintain the white point of the system. **Dynamic Black** : Dynamic Black (DB) is an algorithm that reduces the amount of light reaching the projection path by means of LED output power through current control and compensates for reduced light by gaining up the RGB signals. **HDR** : High Dynamic Range (HDR) is an algorithm that maps wider brightness and color range of HDR source to the projector display range. HDR is affected by several factors such as illumination characteristics, duty cycle distribution and current running sequence. A valid HDR source should be set by HDR_SetHdrSourceConfiguration() before enabling HDR processing. Note: **Chroma Transient Improvement** is applicable to Analog SDTV sources only. DLPC6540 controller doesn't support Analog sources. Even if enabled on DLPC6540 controller, there is no changes in the displayed image when enabled.

Get Image Algorithm Enable
Data returned is in the same format as the Write Parameters.
Returns enable flag for all Image Algorithms '0' - Disabled or algorithm feature not available. '1' - Enabled

Table 21-100. Image Brightness [Opcode: 41h | Destination: 4]

Set Image Brightness	
Write Parameters	
Byte	**Description**
Bytes 0-1	Brightness Adjustment Range = -256.00 to 255.75 with step size .25 Format = s14.2
The brightness control provides the ability to add or subtract a fixed bias from each of the input channels. This may be used to remove any inherent offsets and/or adjust the brightness level. The brightness coefficients are signed, 11-bit (s8.2), 2's complement values between -256 and 255.75, inclusive. Brightness Control is used after Color Space Conversion.	

Get Image Brightness
Data returned is in the same format as the Write Parameters.
Returns Image Brightness Level.

Table 21-101. Image Contrast [Opcode: 42h | Destination: 4]

Set Image Contrast	
Write Parameters	
Byte	**Description**
Bytes 0-1	Contrast (%) Range = 0 to 200 with step size 1
Sets Image Contrast in percentage. Each contrast byte controls the gain applied to the input image data for a given data channel. The contrast gain has a range from 0 to 200 (0% to 200%) with 100 (100%) being nominal (default).	

Get Image Contrast
Data returned is in the same format as the Write Parameters.
Returns Image Contrast in percentage.

Table 21-102. Image Hue And Color Control [Opcode: 43h | Destination: 4]

Set Image Hue And Color Control	
Write Parameters	
Byte	**Description**
Byte 0	Hue Adjustment Angle (degrees) Range = -45 to 45 with step size 1
Bytes 1-2	Color Control Gain (%) Range = 0 to 200 with step size 1
Sets Image Hue Adjustment angle in degrees and Color Control Gain in percentage.	

Get Image Hue And Color Control
Data returned is in the same format as the Write Parameters.
Returns Image Hue Adjustment angle in degrees and Color Control Gain in percentage.

Table 21-103. Image Sharpness [Opcode: 44h | Destination: 4]

Set Image Sharpness	
Write Parameters	
Byte	**Description**
Byte 0	Sharpness value to apply. Range = 0 to 31 with step size 1
Configures the sharpness filter. A value of 0 is the least sharp (smoothest), while a value of 31 is the sharpest. This filter is in the back end of the data path, so both video and graphics are affected. TI recommends that the sharpness filters be disabled (sharpness=16) for graphics sources.	

Get Image Sharpness
Data returned is in the same format as the Write Parameters.
Returns the current sharpness value

Table 21-104. Image RGB Offset [Opcode: 45h | Destination: 4]

Set Image RGB Offset	
Write Parameters	
Byte	**Description**
Bytes 0-1	Red channel offset setting. Range = -256.00 to 255.75 with step size .25 Format = s14.2
Bytes 2-3	Green channel offset setting. Range = -256.00 to 255.75 with step size .25 Format = s14.2
Bytes 4-5	Blue channel offset setting. Range = -256.00 to 255.75 with step size .25 Format = s14.2
Offsets the levels of the RGB channels at a point in the data path after the following image processing functions have been applied - source offset, contrast, RGB Gain, Brightness and Color Space Conversion (including Hue and Color Adjustment).	

Get Image RGB Offset
Data returned is in the same format as the Write Parameters.
Returns Red, Green and Blue channel offset settings.

Table 21-105. Image RGB Gain [Opcode: 46h | Destination: 4]

Set Image RGB Gain	
Write Parameters	
Byte	**Description**
Bytes 0-1	Red channel gain setting. Range = 0 to 200 with step size 1
Bytes 2-3	Green channel gain setting. Range = 0 to 200 with step size 1
Bytes 4-5	Blue channel gain setting. Range = 0 to 200 with step size 1
Adjusts individual R, G and B gains of the source image. Gain is specified as a percentage from 0% - 200%, with 100% being nominal (no gain change). 0% will zero out the channel. This function adjusts R, G and B gains by altering the Color Space Conversion (CSC) coefficients. This function is only applicable to RGB sources.	

Get Image RGB Gain
Data returned is in the same format as the Write Parameters.
Returns gain setting for Red, Green and Blue color channels in percentage.

Table 21-106. CSC Table [Opcode: 47h | Destination: 4]

Set CSC Table	
Write Parameters	
Byte	**Description**
Byte 0	Index of the pre-defined CSC table in flash. Range = 0 to 7 with step size 1 0 = Table Full range Rgb 1 = Table Bt601 Yuv Video decoder 2 = Table Full range Yuv1 3 = Table Offset Rgb 4 = Table Bt601 Offset Yuv 5 = Table Full range Yuv 6 = Table Bt709 Offset Yuv 7 = Table Smpte 240m 8 = Table Bt2020 9 = Maxtable
Sets the Color Space Conversion Matrix with one of the CSC tables stored in flash.	

Get CSC Table
Data returned is in the same format as the Write Parameters.
Gets the index of the Color Space Conversion Matrix that is currently configured for use.

Table 21-107. Image CCA Coordinates [Opcode: 48h | Destination: 4]

Set Image CCA Coordinates	
Write Parameters	
Byte	**Description**
Bytes 0-1	Original Coordinate Red x Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 2-3	Original Coordinate Red y Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 4-5	Original Coordinate Red Luminance Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 6-7	Original Coordinate Green x Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 8-9	Original Coordinate Green y Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 10-11	Original Coordinate Green Luminance Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 12-13	Original Coordinate Blue x Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 14-15	Original Coordinate Blue y Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 16-17	Original Coordinate Blue Luminance Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 18-19	Original Coordinate White x Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 20-21	Original Coordinate White y Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 22-23	Original Coordinate White Luminance Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 24-25	Original Coordinate C1 x Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 26-27	Original Coordinate C1 y Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 28-29	Original Coordinate C1 Luminance Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15

Table 21-107. Image CCA Coordinates [Opcode: 48h | Destination: 4] (continued)

Set Image CCA Coordinates	
Bytes 30-31	Original Coordinate C2 x Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 32-33	Original Coordinate C2 y Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 34-35	Original Coordinate C2 Luminance Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 36-37	Original Coordinate DRA A x Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 38-39	Original Coordinate DRA A y Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 40-41	Original Coordinate DRA A Luminance Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 42-43	Original Coordinate DRA B x Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 44-45	Original Coordinate DRA B y Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 46-47	Original Coordinate DRA B Luminance Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 48-49	Original Coordinate DRA C x Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 50-51	Original Coordinate DRA C y Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 52-53	Original Coordinate DRA C Luminance Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 54-55	Target Coordinate Red x Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 56-57	Target Coordinate Red y Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 58-59	Target Coordinate Red Gain Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 60-61	Target Coordinate Green x Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15

Table 21-107. Image CCA Coordinates [Opcode: 48h | Destination: 4] (continued)

Set Image CCA Coordinates	
Bytes 62-63	Target Coordinate Green y Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 64-65	Target Coordinate Green Gain Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 66-67	Target Coordinate Blue x Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 68-69	Target Coordinate Blue y Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 70-71	Target Coordinate Blue Gain Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 72-73	Target Coordinate Cyan x Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 74-75	Target Coordinate Cyan y Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 76-77	Target Coordinate Cyan Gain Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 78-79	Target Coordinate Magenta x Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 80-81	Target Coordinate Magenta y Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 82-83	Target Coordinate Magenta Gain Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 84-85	Target Coordinate Yellow x Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 86-87	Target Coordinate Yellow y Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 88-89	Target Coordinate Yellow Gain Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 90-91	Target Coordinate White x Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
Bytes 92-93	Target Coordinate White y Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15

Table 21-107. Image CCA Coordinates [Opcode: 48h | Destination: 4] (continued)

Set Image CCA Coordinates	
Bytes 94-95	Target Coordinate White Gain Range = 0.0 to 1.99996948242 with step size 0.00003051757 Format = u1.15
This Command allows independent adjustment of the primary, secondary and white coordinates. Note: This call will override any CCA settings performed by prior calls.	

Get Image CCA Coordinates
Data returned is in the same format as the Write Parameters.
Returns the current color coordinate configuration.

Table 21-108. Image HSG [Opcode: 49h | Destination: 4]

Set Image HSG	
Write Parameters	
Byte	**Description**
Bytes 0-1	HSG Red Gain Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14
Bytes 2-3	HSG Red Saturation Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14
Bytes 4-5	HSG Red Hue Range = -1.0 to 1.0 with step size 0.00006103515 Format = s2.14
Bytes 6-7	HSG Green Gain Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14
Bytes 8-9	HSG Green Saturation Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14
Bytes 10-11	HSG Green Hue Range = -1.0 to 1.0 with step size 0.00006103515 Format = s2.14
Bytes 12-13	HSG Blue Gain Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14
Bytes 14-15	HSG Blue Saturation Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14
Bytes 16-17	HSG Blue Hue Range = -1.0 to 1.0 with step size 0.00006103515 Format = s2.14
Bytes 18-19	HSG Cyan Gain Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14
Bytes 20-21	HSG Cyan Saturation Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14
Bytes 22-23	HSG Cyan Hue Range = -1.0 to 1.0 with step size 0.00006103515 Format = s2.14
Bytes 24-25	HSG Magenta Gain Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14
Bytes 26-27	HSG Magenta Saturation Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14
Bytes 28-29	HSG Magenta Hue Range = -1.0 to 1.0 with step size 0.00006103515 Format = s2.14

Table 21-108. Image HSG [Opcode: 49h | Destination: 4] (continued)

Set Image HSG	
Bytes 30-31	HSG Yellow Gain Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14
Bytes 32-33	HSG Yellow Saturation Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14
Bytes 34-35	HSG Yellow Hue Range = -1.0 to 1.0 with step size 0.00006103515 Format = s2.14
Bytes 36-37	HSG White Red Gain Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14
Bytes 38-39	HSG White Green Gain Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14
Bytes 40-41	HSG White Blue Gain Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14
This command applies the given Hue, Saturation and Gain values for all colors. It does not affect colors having a gain of zero. Note: This call will override any CCA settings performed by prior calls.	

Get Image HSG
Data returned is in the same format as the Write Parameters.
This command returns the currently applied Hue, Saturation and Gain values for all the colors. If Gain for a color is zero then the HSG is not applied on the color.

Table 21-109. Image Gamma LUT [Opcode: 4Ah | Destination: 4]

Set Image Gamma LUT	
Write Parameters	
Byte	**Description**
Byte 0	Gamma look-up table to load.
This command loads the specified Gamma look-up table into memory from flash. A single load is accomplished by loading data for red, green and blue look-up tables. Use DLP Composer(tm) to create new gamma tables or modify existing gamma tables	

Get Image Gamma LUT
Data returned is in the same format as the Write Parameters.
Returns the table number of the Gamma look-up table currently loaded in memory.

Table 21-110. Image Gamma Curve Shift [Opcode: 4Bh | Destination: 4]

Set Image Gamma Curve Shift	
Write Parameters	
Byte	**Description**
Byte 0	Red Gamma curve shift. Range = -128 to 127 with step size 1
Byte 1	Green Gamma curve shift. Range = -128 to 127 with step size 1
Byte 2	Blue Gamma curve shift. Range = -128 to 127 with step size 1
Byte 3	Broadcasted shift to Gamma curves of all color. Range = -128 to 127 with step size 1
Used to specify the shifts in the Gamma curve of Red, Green and Blue. A left shift is a positive offset and a right shift is a negative offset. The effective brightness is increased with a left shift and decreased with a right shift.	

Get Image Gamma Curve Shift
Data returned is in the same format as the Write Parameters.
Returns Image Gamma Shift for red, green and blue as well as shift to be broadcasted to all colors

Table 21-111. Img White Peaking Factor [Opcode: 4Ch | Destination: 4]

Set Img White Peaking Factor	
Write Parameters	
Byte	**Description**
Byte 0	Amount of white processing. Range 0 to MaxValue (set by IMG_SetWhitePeakingRange()). The default value of MaxValue is 10.

Get Img White Peaking Factor
Data returned is in the same format as the Write Parameters.

Table 21-112. XPR Filter Strength Command [Opcode: 4Dh | Destination: 4]

Set XPR Filter Strength Command	
Write Parameters	
Byte	**Description**
Byte 0	Filter Strength setting determines how much of high frequency content is filtered out. Valid range 0-7Setting of 0 means least filtering of high frequency content (sharpest image; more flicker) Setting of 7 means most filtering of high frequency content (smoothest image; least flicker)

Get XPR Filter Strength Command
Data returned is in the same format as the Write Parameters.

Table 21-113. HDR Source Configuration [Opcode: 4Eh | Destination: 4]

Set HDR Source Configuration	
Write Parameters	
Byte	**Description**
Byte 0	Transfer Function 0 = Trad Gam Sdr 1 =Trad Gam Hdr 2 = PQ 3 = HLG
Bytes 1-4	Master Display Black Level (nits) Range = 0.0000 to 10000.0 Format = u16.16
Bytes 5-8	Master Display White Level (nits) Range = 0.0000 to 10000.0 Format = u16.16
Bytes 9-10	Master Display Color Gamut Red x Range = 0.0000 to 1.0000 Format = u1.15
Bytes 11-12	Master Display Color Gamut Red y Range = 0.0000 to 1.0000 Format = u1.15
Bytes 13-14	Master Display Color Gamut Green x Range = 0.0000 to 1.0000 Format = u1.15
Bytes 15-16	Master Display Color Gamut Green y Range = 0.0000 to 1.0000 Format = u1.15
Bytes 17-18	Master Display Color Gamut Blue x Range = 0.0000 to 1.0000 Format = u1.15
Bytes 19-20	Master Display Color Gamut Blue y Range = 0.0000 to 1.0000 Format = u1.15
Bytes 21-22	Master Display Color Gamut White x Range = 0.0000 to 1.0000 Format = u1.15
Bytes 23-24	Master Display Color Gamut White y Range = 0.0000 to 1.0000 Format = u1.15
HDR maps wider brightness and color range of HDR sources to projector brightness and color range. The mapping requires multiple source groups and system groups to define the HDR source and projection device properties respectively. This command sets the source properties and based on this information selects nearest source group for mapping.	

Get HDR Source Configuration
Data returned is in the same format as the Write Parameters.
Includes the metadata information.

Table 21-114. HDR Strength Setting [Opcode: 4Fh | Destination: 4]

Set HDR Strength Setting	
Write Parameters	
Byte	**Description**
Byte 0	HDR Strength Range = 0 to 10
Sets HDR strength which adjusts the electro-optical transfer function that is applied on the input HDR video signal. HDR strength can vary with the ambient brightness level. HDR strength is not applicable for HLG transfer function set by HDR source configuration.	

Get HDR Strength Setting
Data returned is in the same format as the Write Parameters.

Table 21-115. System Brightness Range Setting [Opcode: 50h | Destination: 4]

Set System Brightness Range Setting	
Write Parameters	
Byte	**Description**
Bytes 0-3	Min Brightness (nits) Range = 0.0000 to 10000.0 Format = u16.16
Bytes 4-7	Max Brightness (nits) Range = 0.0000 to 10000.0 Format = u16.16
Sets the system brightness range in nits. These are used in determining the appropriate EOTF and OOTF function to be applied on the HDR source. This need to set only for HDR functionality.	

Get System Brightness Range Setting
Data returned is in the same format as the Write Parameters.

Table 21-116. Image Color Profile [Opcode: 51h | Destination: 4]

Set Image Color Profile	
Write Parameters	
Byte	**Description**
Byte 0	Color Profile
Sets pre-configured Gamma table index and HSG settings as stored in the flash image.	

Table 21-117. Image Point HSG [Opcode: 52h | Destination: 4]

Set Image Point HSG	
Write Parameters	
Byte	**Description**
Byte 0	Point 0 = Row0 Col0 1 = Row0 Col1 2 = Row0 Col2 3 = Row0 Col3 4 = Row0 Col4 5 = Row1 Col0 6 = Row1 Col1 7 = Row1 Col2 8 = Row1 Col3 9 = Row1 Col4 10 = Row2 Col0 11 = Row2 Col1 12 = Row2 Col2 13 = Row2 Col3 14 = Row2 Col4
Bytes 1-2	HSG Red Gain Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14
Bytes 3-4	HSG Red Saturation Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14
Bytes 5-6	HSG Red Hue Range = -1.0 to 1.0 with step size 0.00006103515 Format = s2.14
Bytes 7-8	HSG Green Gain Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14
Bytes 9-10	HSG Green Saturation Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14
Bytes 11-12	HSG Green Hue Range = -1.0 to 1.0 with step size 0.00006103515 Format = s2.14
Bytes 13-14	HSG Blue Gain Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14
Bytes 15-16	HSG Blue Saturation Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14
Bytes 17-18	HSG Blue Hue Range = -1.0 to 1.0 with step size 0.00006103515 Format = s2.14
Bytes 19-20	HSG Cyan Gain Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14

Table 21-117. Image Point HSG [Opcode: 52h | Destination: 4] (continued)

Set Image Point HSG	
Bytes 21-22	HSG Cyan Saturation Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14
Bytes 23-24	HSG Cyan Hue Range = -1.0 to 1.0 with step size 0.00006103515 Format = s2.14
Bytes 25-26	HSG Magenta Gain Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14
Bytes 27-28	HSG Magenta Saturation Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14
Bytes 29-30	HSG Magenta Hue Range = -1.0 to 1.0 with step size 0.00006103515 Format = s2.14
Bytes 31-32	HSG Yellow Gain Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14
Bytes 33-34	HSG Yellow Saturation Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14
Bytes 35-36	HSG Yellow Hue Range = -1.0 to 1.0 with step size 0.00006103515 Format = s2.14
Bytes 37-38	HSG White Red Gain Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14
Bytes 39-40	HSG White Green Gain Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14
Bytes 41-42	HSG White Blue Gain Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14
This command applies the given hue, saturation and gain values for all colors, for a specified sample point. Point is a number 0-15 corresponding to one of the 5 x 3 PCC sample points in raster scan order. It does not affect colors having a gain of zero. Note : This call will override any CCA settings performed by prior calls.	

Get Image Point HSG	
Read Parameters	
Byte	**Description**

Get Image Point HSG	
Byte 0	Point 0 = Row0 Col0 1 = Row0 Col1 2 = Row0 Col2 3 = Row0 Col3 4 = Row0 Col4 5 = Row1 Col0 6 = Row1 Col1 7 = Row1 Col2 8 = Row1 Col3 9 = Row1 Col4 10 = Row2 Col0 11 = Row2 Col1 12 = Row2 Col2 13 = Row2 Col3 14 = Row2 Col4

Return Parameters	
Byte	**Description**
Bytes 0-1	HSG Red Gain Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14
Bytes 2-3	HSG Red Saturation Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14
Bytes 4-5	HSG Red Hue Range = -1.0 to 1.0 with step size 0.00006103515 Format = s2.14
Bytes 6-7	HSG Green Gain Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14
Bytes 8-9	HSG Green Saturation Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14
Bytes 10-11	HSG Green Hue Range = -1.0 to 1.0 with step size 0.00006103515 Format = s2.14
Bytes 12-13	HSG Blue Gain Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14
Bytes 14-15	HSG Blue Saturation Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14
Bytes 16-17	HSG Blue Hue Range = -1.0 to 1.0 with step size 0.00006103515 Format = s2.14
Bytes 18-19	HSG Cyan Gain Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14

Bytes 20-21	HSG Cyan Saturation Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14
Bytes 22-23	HSG Cyan Hue Range = -1.0 to 1.0 with step size 0.00006103515 Format = s2.14
Bytes 24-25	HSG Magenta Gain Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14
Bytes 26-27	HSG Magenta Saturation Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14
Bytes 28-29	HSG Magenta Hue Range = -1.0 to 1.0 with step size 0.00006103515 Format = s2.14
Bytes 30-31	HSG Yellow Gain Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14
Bytes 32-33	HSG Yellow Saturation Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14
Bytes 34-35	HSG Yellow Hue Range = -1.0 to 1.0 with step size 0.00006103515 Format = s2.14
Bytes 36-37	HSG White Red Gain Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14
Bytes 38-39	HSG White Green Gain Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14
Bytes 40-41	HSG White Blue Gain Range = 0.0 to 1.99993896485 with step size 0.00006103515 Format = s2.14
This command returns the currently applied hue, saturation and gain values for all the colors, for a specified sample point. Point is a number 0-15 corresponding to one of the 5 x 3 PCC sample points in raster scan order. If gain for a color is zero then the HSG is not applied on the color.	

Table 21-118. Spcc Control Points [Opcode: 53h | Destination: 4]

Set Spcc Control Points	
Write Parameters	
Byte	**Description**
Bytes 0-1	Sets vertical position for (row, col) sample points (1,0), (1,1), (1,2), (1,3), (1,4) (pixels)
Bytes 2-3	Sets horizontal position for (row, col) sample points (0,1), (1,1), (2,1) (pixels)
Bytes 4-5	Sets horizontal position for (row, col) sample points (0,3), (1,3), (2,3) (pixels)
Sets positions of control points for Multipoint sPCC.	

Get Spcc Control Points
Data returned is in the same format as the Write Parameters.
Returns positions of control points for Multipoint sPCC

Table 21-119. Pcc Coefficients Direct [Opcode: 54h | Destination: 4]

Set Pcc Coefficients Direct	
Write Parameters	
Byte	**Description**
Byte 0	Point 0 = Row0 Col0 1 = Row0 Col1 2 = Row0 Col2 3 = Row0 Col3 4 = Row0 Col4 5 = Row1 Col0 6 = Row1 Col1 7 = Row1 Col2 8 = Row1 Col3 9 = Row1 Col4 10 = Row2 Col0 11 = Row2 Col1 12 = Row2 Col2 13 = Row2 Col3 14 = Row2 Col4
Bytes 1-2	Pcc Red R Range = 0.0 to 1.99951171875 with step size 0.00006103515 Format = s2.14
Bytes 3-4	Pcc Red G Range = 0.0 to 1.99951171875 with step size 0.00006103515 Format = s2.14
Bytes 5-6	Pcc Red B Range = 0.0 to 1.99951171875 with step size 0.00006103515 Format = s2.14
Bytes 7-8	Pcc Green R Range = 0.0 to 1.99951171875 with step size 0.00006103515 Format = s2.14
Bytes 9-10	Pcc Green G Range = 0.0 to 1.99951171875 with step size 0.00006103515 Format = s2.14
Bytes 11-12	Pcc Green B Range = 0.0 to 1.99951171875 with step size 0.00006103515 Format = s2.14
Bytes 13-14	Pcc Blue R Range = 0.0 to 1.99951171875 with step size 0.00006103515 Format = s2.14
Bytes 15-16	Pcc Blue G Range = 0.0 to 1.99951171875 with step size 0.00006103515 Format = s2.14
Bytes 17-18	Pcc Blue B Range = 0.0 to 1.99951171875 with step size 0.00006103515 Format = s2.14
Bytes 19-20	Pcc Cyan R Range = 0.0 to 1.99951171875 with step size 0.00006103515 Format = s2.14

Table 21-119. Pcc Coefficients Direct [Opcode: 54h | Destination: 4] (continued)

Set Pcc Coefficients Direct	
Bytes 21-22	Pcc Cyan G Range = 0.0 to 1.99951171875 with step size 0.00006103515 Format = s2.14
Bytes 23-24	Pcc Cyan B Range = 0.0 to 1.99951171875 with step size 0.00006103515 Format = s2.14
Bytes 25-26	Pcc Magenta R Range = 0.0 to 1.99951171875 with step size 0.00006103515 Format = s2.14
Bytes 27-28	Pcc Magenta G Range = 0.0 to 1.99951171875 with step size 0.00006103515 Format = s2.14
Bytes 29-30	Pcc Magenta B Range = 0.0 to 1.99951171875 with step size 0.00006103515 Format = s2.14
Bytes 31-32	Pcc Yellow R Range = 0.0 to 1.99951171875 with step size 0.00006103515 Format = s2.14
Bytes 33-34	Pcc Yellow G Range = 0.0 to 1.99951171875 with step size 0.00006103515 Format = s2.14
Bytes 35-36	Pcc Yellow B Range = 0.0 to 1.99951171875 with step size 0.00006103515 Format = s2.14
Bytes 37-38	Pcc White R Range = 0.0 to 1.99951171875 with step size 0.00006103515 Format = s2.14
Bytes 39-40	Pcc White G Range = 0.0 to 1.99951171875 with step size 0.00006103515 Format = s2.14
Bytes 41-42	Pcc White B Range = 0.0 to 1.99951171875 with step size 0.00006103515 Format = s2.14
This command applies raw PCC Coefficients for all colors through direct access, for a specified sample point. Point is a number 0-15 corresponding to one of the the 5 x 3 PCC sample points in raster scan order. Note : This call will override any CCA settings performed by prior calls.	

Get Pcc Coefficients Direct	
Read Parameters	
Byte	**Description**

Get Pcc Coefficients Direct	
Byte 0	Point 0 = Row0 Col0 1 = Row0 Col1 2 = Row0 Col2 3 = Row0 Col3 4 = Row0 Col4 5 = Row1 Col0 6 = Row1 Col1 7 = Row1 Col2 8 = Row1 Col3 9 = Row1 Col4 10 = Row2 Col0 11 = Row2 Col1 12 = Row2 Col2 13 = Row2 Col3 14 = Row2 Col4

Return Parameters	
Byte	**Description**
Bytes 0-1	Pcc Red R Range = 0.0 to 1.99951171875 with step size 0.00006103515 Format = s2.14
Bytes 2-3	Pcc Red G Range = 0.0 to 1.99951171875 with step size 0.00006103515 Format = s2.14
Bytes 4-5	Pcc Red B Range = 0.0 to 1.99951171875 with step size 0.00006103515 Format = s2.14
Bytes 6-7	Pcc Green R Range = 0.0 to 1.99951171875 with step size 0.00006103515 Format = s2.14
Bytes 8-9	Pcc Green G Range = 0.0 to 1.99951171875 with step size 0.00006103515 Format = s2.14
Bytes 10-11	Pcc Green B Range = 0.0 to 1.99951171875 with step size 0.00006103515 Format = s2.14
Bytes 12-13	Pcc Blue R Range = 0.0 to 1.99951171875 with step size 0.00006103515 Format = s2.14
Bytes 14-15	Pcc Blue G Range = 0.0 to 1.99951171875 with step size 0.00006103515 Format = s2.14
Bytes 16-17	Pcc Blue B Range = 0.0 to 1.99951171875 with step size 0.00006103515 Format = s2.14
Bytes 18-19	Pcc Cyan R Range = 0.0 to 1.99951171875 with step size 0.00006103515 Format = s2.14

Bytes 20-21	Pcc Cyan G Range = 0.0 to 1.99951171875 with step size 0.00006103515 Format = s2.14
Bytes 22-23	Pcc Cyan B Range = 0.0 to 1.99951171875 with step size 0.00006103515 Format = s2.14
Bytes 24-25	Pcc Magenta R Range = 0.0 to 1.99951171875 with step size 0.00006103515 Format = s2.14
Bytes 26-27	Pcc Magenta G Range = 0.0 to 1.99951171875 with step size 0.00006103515 Format = s2.14
Bytes 28-29	Pcc Magenta B Range = 0.0 to 1.99951171875 with step size 0.00006103515 Format = s2.14
Bytes 30-31	Pcc Yellow R Range = 0.0 to 1.99951171875 with step size 0.00006103515 Format = s2.14
Bytes 32-33	Pcc Yellow G Range = 0.0 to 1.99951171875 with step size 0.00006103515 Format = s2.14
Bytes 34-35	Pcc Yellow B Range = 0.0 to 1.99951171875 with step size 0.00006103515 Format = s2.14
Bytes 36-37	Pcc White R Range = 0.0 to 1.99951171875 with step size 0.00006103515 Format = s2.14
Bytes 38-39	Pcc White G Range = 0.0 to 1.99951171875 with step size 0.00006103515 Format = s2.14
Bytes 40-41	Pcc White B Range = 0.0 to 1.99951171875 with step size 0.00006103515 Format = s2.14
This command gets raw PCC Coefficients for all colors through direct access, for a specified sample point. Point is a number 0-15 corresponding to one of the the 5 x 3 PCC sample points in raster scan order. Note : This call will override any CCA settings performed by prior calls.	

21.12 Manual WPC

Manual WPC

Table 21-120. WPC Target Manual Mode [Opcode: D4h | Destination: 4]

Set WPC Target Manual Mode	
Write Parameters	
Byte	**Description**
Byte 0	0 = Manual Mode Disable 1 = Manual Mode Enable
Sets/Resets the manual mode for specifying WPC target color point at run-time. When manual mode is set, all target color points specified in the project will be ignored. Software will set only the user specified target color point until the manual mode is reset using this same command.	

Get WPC Target Manual Mode
Data returned is in the same format as the Write Parameters.
Gets whether the manual mode for specifying WPC target color point at run-time is active. When manual mode is set, all target color points specified in the project will be ignored. Software will set only the user specified target color point until the manual mode is reset.

Table 21-121. WPC Target Color Point [Opcode: D5h | Destination: 4]

Set WPC Target Color Point	
Write Parameters	
Byte	**Description**
Bytes 0-1	CIE X Range = 0.0000 to 1.0000 Format = u0.16
Bytes 2-3	CIE Y Range = 0.0000 to 1.0000 Format = u0.16
Sets the target color point while in WPC Target Manual Mode.	

Get WPC Target Color Point
Data returned is in the same format as the Write Parameters.
Gets the currently active target color point for WPC.

21.13 Peripherals

Peripherals

Table 21-122. GPIO Pin Config [Opcode: 60h | Destination: 4]

Set GPIO Pin Config	
Write Parameters	
Byte	**Description**
Byte 0	Range = 0 to 87.
Byte 1	Input Output bit0: 1 = Output(Output buffer enabled) 0 = Input(Output buffer High Z)
Byte 2	Logic Value bit0: 1 = LogicVal 1 0 = LogicVal 0
Byte 3	Open Drain Configuration bit0: 1 = Open Drain output 0 = Standard output
Programs the direction, logic value and open drain characteristics of a single general purpose I/O pin.	

Get GPIO Pin Config	
Read Parameters	
Byte	**Description**
Byte 0	Range = 0 to 87.

Return Parameters	
Byte	**Description**
Byte 0	Input Output bit0: 1 = Output(Output buffer enabled) 0 = Input(Output buffer High Z)
Byte 1	Logic Value bit0: 1 = LogicVal 1 0 = LogicVal 0
Byte 2	Open Drain Configuration bit0: 1 = Open Drain output 0 = Standard output
Returns the direction, logic value and open drain configuration for a single general purpose I/O pin.	

Table 21-123. GPIO Pin [Opcode: 61h | Destination: 4]

Set GPIO Pin	
Write Parameters	
Byte	**Description**
Byte 0	Range = 0 to 87.
Byte 1	Logic Value bit0: 1 = LogicVal 1 0 = LogicVal 0
Sets the output logic value for the specified GPIO Pin.	

Get GPIO Pin	
Read Parameters	
Byte	**Description**
Byte 0	Range = 0 to 87.

Return Parameters	
Byte	**Description**
Byte 0	Logic Value bit0: 1 = LogicVal 1 0 = LogicVal 0
Returns the logic value for the specified GPIO pin.	

Table 21-124. Gen Purpose Clock Enable [Opcode: 63h | Destination: 4]

Set Gen Purpose Clock Enable	
Write Parameters	
Byte	**Description**
Byte 0	Clock to Configure
Byte 1	TRUE = Enable clock FALSE = Disable clock.
Bytes 2-5	Amount to divide the selected clock. This parameter is ignored if the clock is to be disabled. Range 2-127.

Get Gen Purpose Clock Enable	
Read Parameters	
Byte	**Description**
Byte 0	DDP Clock Output.

Return Parameters	
Byte	**Description**
Byte 0	Is Enabled

Table 21-125. Gen Purpose Clock Frequency [Opcode: 64h | Destination: 4]

Get Gen Purpose Clock Frequency	
Read Parameters	
Byte	**Description**
Byte 0	Clock for which the frequency configuration needs to be returned.

Return Parameters	
Byte	**Description**
Bytes 0-3	Clock frequency in kHz. Range = 787 to 50,000 kHz.

Table 21-126. PWM Output Configuration [Opcode: 65h | Destination: 4]

Set PWM Output Configuration	
Write Parameters	
Byte	**Description**
Byte 0	Port 0 = Output PWM 0 1 = Output PWM 1 2 = Output PWM 2 3 = Output PWM 3 4 = Output PWM 4 5 = Output PWM 5 6 = Output PWM 6 7 = Output PWM 7 8 = Color Wheel 0 PWM 9 = Color Wheel 1 PWM 10 = Color Wheel 2 PWM 11 = Input PWM 0 12 = Input PWM 1 13 = Used for setting the DynamicBlack PWM port when DB PWM is not used 14 = Invalid PWM Port
Bytes 1-4	Frequency Range = 20 to 10390000 with step size 1
Byte 5	Duty Cycle Range = 0 to 100 with step size 1
Byte 6	Output Enabled Byte bit0: 1 = Enabled 0 = Disabled
Sets the Duty cycle and frequency for the specified PWM port. It also enables or disables the port. NOTE : The frequency of LED PWM Ports is fixed and cannot be changed.	

Get PWM Output Configuration	
Read Parameters	
Byte	**Description**
Byte 0	Port 0 = Output PWM 0 1 = Output PWM 1 2 = Output PWM 2 3 = Output PWM 3 4 = Output PWM 4 5 = Output PWM 5 6 = Output PWM 6 7 = Output PWM 7 8 = Color Wheel 0 PWM 9 = Color Wheel 1 PWM 10 = Color Wheel 2 PWM 11 = Input PWM 0 12 = Input PWM 1 13 = Used for setting the DynamicBlack PWM port when DB PWM is not used 14 = Invalid PWM Port

Return Parameters	
Byte	**Description**

Bytes 0-3	Frequency Range = 20 to 10390000 with step size 1
Byte 4	Duty Cycle Range = 0 to 100 with step size 1
Byte 5	Output Enabled bit0: 1 = Enabled 0 = Disabled
Gets the Duty cycle and frequency for the specified PWM port. It also returns whether the port is currently enabled or disabled.	

Table 21-127. PWM Input Configuration [Opcode: 66h | Destination: 4]

Set PWM Input Configuration	
Write Parameters	
Byte	**Description**
Byte 0	Port 0 = PWM Input Counter 0 1 = PWM Input Counter 1
Bytes 1-4	Sample Rate
Byte 5	In Counter Enabled
Sets the sample rate, duty cycle, high pulse width and low pulse width of the specified PWM incounter port. It also enables or disables the port.	

Get PWM Input Configuration	
Read Parameters	
Byte	**Description**
Byte 0	Port 0 = PWM Input Counter 0 1 = PWM Input Counter 1

Return Parameters	
Byte	**Description**
Bytes 0-3	Sample Rate Range = 317 to 20780000 with step size 1
Byte 4	In Counter Enabled
Bytes 5-6	High Pulse Width
Bytes 7-8	Low Pulse Width
Byte 9	Duty Cycle Range = 0 to 100 with step size 1
Gets the sample rate, duty cycle, high pulse width and low pulse width of the specified PWM incounter port. It also returns whether the port is currently enabled or disabled.	

Table 21-128. I2C Passthrough [Opcode: 67h | Destination: 4]

Set I2C Passthrough	
Write Parameters	
Byte	**Description**
Byte 0	Port 0 = I2C Port 0 1 = I2C Port 1 2 = I2C Port 2 3 = Only three Ports are supported
Byte 1	7-bit Address - 0 = 10-bit Address; 1 = 7-bit Address
Byte 2	SubAddress Present - 0 = No sub-addr present; 1 = sub-addr present
Bytes 3-6	Clock Rate - 100Khz or 400Khz supported
Bytes 7-8	Device Address
Bytes 9- Number of bytes passed	Sub-address (if present)
Bytes 9 - *	Data Bytes
Writes data to specified I2C device address.	

Get I2C Passthrough	
Read Parameters	
Byte	**Description**
Byte 0	Port 0 = I2C Port 0 1 = I2C Port 1 2 = I2C Port 2 3 = Only three Ports are supported
Byte 1	7-bit Address 0 = 10-bit Address 1 = 7-bit Address
Byte 2	SubAddress Present - 0 = No sub-addr present; 1 = sub-addr present
Bytes 3-6	Clock Rate
Bytes 7-8	Device Address
Bytes 9-10	Byte Count
Bytes 11- Number of bytes passed	sub-address (if present)

Return Parameters	
Byte	**Description**
Bytes 0- Number of bytes passed	Data Bytes
Reads data from specified I2C device address.	

Table 21-129. DMD Temperature [Opcode: 69h | Destination: 4]

Get DMD Temperature	
Return Parameters	
Byte	**Description**
Bytes 0-1	Value in degree Celsius. Note : As a default condition, the firmware is configured to read TMP411A outputs using I2C port 2. This can be changed as desired. Using latest DLP Composer SW, the firmware can be rebuilt to receive serial data via I2C at ports 1 and 2 to which the hardware would need to be configured. Range = -256 to 255 with step size 1
This command applicable only if TMP411A temperature sensor is installed in the system.	

Table 21-130. EEPROM Lock State [Opcode: 6Ch | Destination: 4]

Set EEPROM Lock State	
Write Parameters	
Byte	**Description**
Byte 0	'0' - Unlocked '1' - Locked
Sets the lock state of EEPROM. When lock is set, all writes to EEPROM settings and/or calibration data from application software will not be actually written to the EEPROM. The locked mode is to be used only in factory where user wants to play around with various settings without actually recording them in the EEPROM. In normal use mode, the lock is not supposed to be set.	

Get EEPROM Lock State
Data returned is in the same format as the Write Parameters.
Gets the lock state of EEPROM.

Table 21-131. UART Configuration [Opcode: 6Dh | Destination: 4]

Set UART Configuration	
Write Parameters	
Byte	**Description**
Byte 0	UART Port 0 = Port0 1 = Port1 2 = Port2
Byte 1	Enable State bit0: 0 = Disable 1 = Enable
Byte 2	Baud Rate 0 = 1200 1 = 2400 2 = 4800 3 = 9600 4 = 14400 5 = 19200 6 = 38400 7 = 57600 8 = 115200 9 = 230400 10 = 460800 11 = 921600
Byte 3	Data Bits 0 = 5 1 = 6 2 = 7 3 = 8
Byte 4	Stop Bits 0 = 1 1 = 2
Byte 5	Parity 0 = Parity bit is neither transmitted or checked 1 = Even parity is transmitted and checked 2 = Odd parity is transmitted and checked
Byte 6	Flow Control 0 = Off 1 = Hardware flow control
Byte 7	Rx Trig Level 0 = One Eighth Full 1 = One Fourth Full 2 = One Half Full 3 = Three Fourths Full 4 = Seven Eighths Full
Byte 8	Tx Trig Level 0 = One Eighth Full 1 = One Fourth Full 2 = One Half Full 3 = Three Fourths Full 4 = Seven Eighths Full

Table 21-131. UART Configuration [Opcode: 6Dh | Destination: 4] (continued)

Set UART Configuration	
Byte 9	Rx Data Polarity 0 = Supply non-inverted version of UART_RXD input 1 = Supply inverted version of UART_RXD input
Byte 10	Rx Data Source 0 = UART_x.RXD is sourced by UART_x_RXD pin 1 = UART_x.RXD is sourced by LAMPSTAT pin
Initializes all programmable parameters for the specified UART port.	

Get UART Configuration	
Read Parameters	
Byte	**Description**
Byte 0	UART Port 0 = Port0 1 = Port1 2 = Port2

Return Parameters	
Byte	**Description**
Byte 0	Enable State bit0: 0 = Disable 1 = Enable
Byte 1	Baud Rate 0 = 1200 1 = 2400 2 = 4800 3 = 9600 4 = 14400 5 = 19200 6 = 38400 7 = 57600 8 = 115200 9 = 230400 10 = 460800 11 = 921600
Byte 2	Data Bits 0 = 5 1 = 6 2 = 7 3 = 8
Byte 3	Stop Bits 0 = 1 1 = 2
Byte 4	Parity 0 = Parity bit is neither transmitted or checked 1 = Even parity is transmitted and checked 2 = Odd parity is transmitted and checked
Byte 5	Flow Control 0 = Off 1 = Hardware flow control

Byte 6	Rx Trig Level 0 = One Eighth Full 1 = One Fourth Full 2 = One Half Full 3 = Three Fourths Full 4 = Seven Eighths Full
Byte 7	Tx Trig Level 0 = One Eighth Full 1 = One Fourth Full 2 = One Half Full 3 = Three Fourths Full 4 = Seven Eighths Full
Byte 8	Rx Data Polarity 0 = Supply non-inverted version of UART_RXD input 1 = Supply inverted version of UART_RXD input
Byte 9	Rx Data Source 0 = UART_x.RXD is sourced by UART_x_RXD pin 1 = UART_x.RXD is sourced by LAMPSTAT pin
Gets current configuration for the specified UART port.	

Table 21-132. Actuator EEPROM Free Memory Access [Opcode: 6Eh | Destination: 4]

Set Actuator EEPROM Free Memory Access	
Write Parameters	
Byte	**Description**
Bytes 0-1	Offset
Bytes 2-3	Size
Bytes 4 - *	Data Bytes
Writes data to Actuator EEPROM available memory.	

Get Actuator EEPROM Free Memory Access	
Read Parameters	
Byte	**Description**
Bytes 0-1	Offset
Bytes 2-3	Size

Return Parameters	
Byte	**Description**
Bytes 0- Number of bytes passed	Data Bytes
This Command reads data from Actuator EEPROM Free memory.	

Table 21-133. Actuator EEPROM Free Memory Info [Opcode: 6Fh | Destination: 4]

Get Actuator EEPROM Free Memory Info	
Return Parameters	
Byte	**Description**
Bytes 0-1	Offset
Bytes 2-3	Size
This command returns the XPR EEPROM address offset which corresponds to the start of free memory area and size available.	

21.14 Vision

Vision

Table 21-134. Auto Screen Fit Corners [Opcode: 81h | Destination: 4]

Get Auto Screen Fit Corners	
Return Parameters	
Byte	**Description**
Bytes 0-3	Top Left X Format = s16.16
Bytes 4-7	Top Left Y Format = s16.16
Bytes 8-11	Top Right X Format = s16.16
Bytes 12-15	Top Right Y Format = s16.16
Bytes 16-19	Bottom Right X Format = s16.16
Bytes 20-23	Bottom Right Y Format = s16.16
Bytes 24-27	Bottom Left X Format = s16.16
Bytes 28-31	Bottom Left Y Format = s16.16
Get auto screen fit corners	

Table 21-135. Cam Proj Calibration Params [Opcode: D6h | Destination: 4]

Get Cam Proj Calibration Params	
Return Parameters	
Byte	**Description**
Byte 0	Camera distortion model, FALSE = Use Brown's distortion model, TRUE = Use the fisheye distortion model
Byte 1	RGB or monochrome camera
Bytes 2-3	Camera horizontal resolution
Bytes 4-5	Camera vertical resolution
Bytes 6-9	Camera focal length in x direction Format = s16.16
Bytes 10-13	Camera focal length in y direction Format = s16.16
Bytes 14-17	Camera principal point in x direction Format = s16.16
Bytes 18-21	Camera principal point in y direction Format = s16.16
Bytes 22-25	Camera distortion coefficient K1 Format = s16.16
Bytes 26-29	Camera distortion coefficient K2 Format = s16.16
Bytes 30-33	Camera distortion coefficient P1 Format = s16.16
Bytes 34-37	Camera distortion coefficient P2 Format = s16.16
Bytes 38-41	Camera distortion coefficient K3 Format = s16.16
Bytes 42-45	Camera distortion coefficient K4 Format = s16.16
Bytes 46-49	Camera distortion coefficient K5 Format = s16.16
Bytes 50-53	Camera distortion coefficient K6 Format = s16.16
Bytes 54-55	Projector horizontal resolution
Bytes 56-57	Projector vertical resolution
Bytes 58-61	Projector focal length in x direction Format = s16.16
Bytes 62-65	Projector focal length in y direction Format = s16.16
Bytes 66-69	Projector principal point in x direction Format = s16.16
Bytes 70-73	Projector principal point in y direction Format = s16.16
Bytes 74-77	Projector distortion coefficient K1 Format = s16.16
Bytes 78-81	Projector distortion coefficient K2 Format = s16.16

Table 21-135. Cam Proj Calibration Params [Opcode: D6h | Destination: 4] (continued)

Get Cam Proj Calibration Params	
Bytes 82-85	Projector distortion coefficient K3 Format = s16.16
Bytes 86-89	Projector distortion coefficient K4 Format = s16.16
Bytes 90-93	Projector distortion coefficient K5 Format = s16.16
Bytes 94-97	Projector distortion coefficient K6 Format = s16.16
Bytes 98-101	Projector distortion coefficient P1 Format = s16.16
Bytes 102-105	Projector distortion coefficient P2 Format = s16.16
Bytes 106-109	Camera rotation matrix [0][0] Format = s16.16
Bytes 110-113	Camera rotation matrix [0][1] Format = s16.16
Bytes 114-117	Camera rotation matrix [0][2] Format = s16.16
Bytes 118-121	Camera rotation matrix [1][0] Format = s16.16
Bytes 122-125	Camera rotation matrix [1][1] Format = s16.16
Bytes 126-129	Camera rotation matrix [1][2] Format = s16.16
Bytes 130-133	Camera rotation matrix [2][0] Format = s16.16
Bytes 134-137	Camera rotation matrix [2][1] Format = s16.16
Bytes 138-141	Camera rotation matrix [2][2] Format = s16.16
Bytes 142-145	Camera translation vector [0] Format = s16.16
Bytes 146-149	Camera translation vector [1] Format = s16.16
Bytes 150-153	Camera translation vector [2] Format = s16.16
Gets calibration parameters for camera and projector	

Table 21-136. Projector Variation Index [Opcode: D7h | Destination: 4]

Set Projector Variation Index	
Write Parameters	
Byte	**Description**
Bytes 0-1	Cam Proj Variation Index
Set current variation index for projector	

Get Projector Variation Index
Data returned is in the same format as the Write Parameters.
Get current variation index for projector.

Table 21-137. Cam Params [Opcode: D8h | Destination: 4]

Set Cam Params	
Write Parameters	
Byte	**Description**
Bytes 0-1	Horizontal capture resolution
Bytes 2-3	Vertical capture resolution
Bytes 4-5	Bytes per pixel
Byte 6	camera Camera test, for debug purpose only bit0: start or stop camera test, for debug purpose only
Set Camera Resolution and BPP	

Get Cam Params	
Return Parameters	
Byte	**Description**
Bytes 0-1	Horizontal capture resolution
Bytes 2-3	Vertical capture resolution
Bytes 4-5	Bytes per pixel
Get Camera resolution and BPP	

Table 21-138. Enable Auto Warp [Opcode: D9h | Destination: 4]

Set Enable Auto Warp	
Write Parameters	
Byte	**Description**
Bytes 0-1	Horizontal resolution Range = 0 to 7680 with step size 10
Bytes 2-3	Vertical resolution Range = 0 to 4320 with step size 10
Bytes 4-5	Num Samples X Range = 2 to 62 with step size 1
Bytes 6-7	Num Samples Y Range = 2 to 32 with step size 1
Byte 8	Filter Flags
Bytes 9-12	CCA Detect Thresh Range = 0.0000 to 1.0000 with step size 0.0001 Format = u16.16
Bytes 13-14	Radius of the gaussian
Bytes 15-16	Gaussian centroid offset in pixels (from DMD edge)
Bytes 17-20	Pitch bias in degrees Range = -10 to 10 with step size 0.00390625 Format = s16.16
Bytes 21-24	Yaw bias in degrees Range = -10 to 10 with step size 0.00390625 Format = s16.16
Byte 25	Level of digital decimation for camera captures Range = 0 to 2 with step size 1
Byte 26	Enable or disable ROI processing for camera captures bit0: Pixel Sampling Enable
Set configuration and enable auto warp	

Table 21-139. Enable Auto Screen Fit [Opcode: DAh | Destination: 4]

Set Enable Auto Screen Fit	
Write Parameters	
Byte	**Description**
Bytes 0-1	Aspect Ratio Width
Bytes 2-3	Aspect Ratio Height
Bytes 4-7	CCA feature detection threshold Range = 0.0000 to 1.0000 with step size 0.0001 Format = u16.16
Bytes 8-9	Vertical offset in pixels of the 2x2 gaussian pattern Range = -300 to 300 with step size 1
Bytes 10-13	Percentage of detected screen frame to be filled Range = 80.0 to 120.0 with step size 0.1 Format = u16.16
Bytes 14-17	Intensity of gray solid field, used for pattern generation * Range = 0.0 to 100.0 with step size 0.1 Format = u16.16
Byte 18	Pixel sampling difference Range = 1 to 2 with step size 1
Byte 19	Variable Pixel Sampling bit0: Pixel Sampling Enable
Set configuration and enable auto screen fit	

Table 21-140. Resize Auto Screen Fit [Opcode: DBh | Destination: 4]

Set Resize Auto Screen Fit	
Write Parameters	
Byte	**Description**
Bytes 0-3	Percentage of detected screen frame to be filled Range = 80.0 to 120.0 with step size 0.1 Format = u16.16
Resize auto screen fit	

Table 21-141. Refine Auto Screen Fit [Opcode: DCh | Destination: 4]

Set Refine Auto Screen Fit	
Write Parameters	
Byte	**Description**
Bytes 0-1	Horizontal offset in pixels Range = -300 to 300 with step size 1
Bytes 2-3	Vertical offset in pixels Range = -300 to 300 with step size 1
Bytes 4-5	Horizontal offset in pixels Range = -300 to 300 with step size 1
Bytes 6-7	Vertical offset in pixels Range = -300 to 300 with step size 1
Bytes 8-9	Horizontal offset in pixels Range = -300 to 300 with step size 1
Bytes 10-11	Vertical offset in pixels Range = -300 to 300 with step size 1
Bytes 12-13	Horizontal offset in pixels Range = -300 to 300 with step size 1
Bytes 14-15	Vertical offset in pixels Range = -300 to 300 with step size 1
Refine auto screen fit	

Table 21-142. Asf Pattern [Opcode: DDh | Destination: 4]

Set Asf Pattern	
Write Parameters	
Byte	**Description**
Byte 0	Pattern Num
Bytes 1-2	Vertical offset in pixels of the 2x2 gaussian pattern Range = -300 to 300 with step size 1
Bytes 3-6	Intensity of gray solid field, used for pattern generation Range = 0.0 to 100.0 with step size 0.1 Format = u16.16
Project gaussian pattern defined by Index	

Table 21-143. Enable Auto Exposure [Opcode: DEh | Destination: 4]

Set Enable Auto Exposure	
Write Parameters	
Byte	**Description**
Bytes 0-1	Horizontal resolution Range = 0 to 7680 with step size 10
Bytes 2-3	Vertical resolution Range = 0 to 4320 with step size 10
Byte 4	Number of bytes per pixel supported by camera
Byte 5	1 : Enable 0 : Disable
Byte 6	Dim Projection until Image Saturation less than threshold Range = 0 to 100 with step size 1
Byte 7	Changes Exposure Time until Image Saturation less than threshold Range = 0 to 100 with step size 1
Enable Auto Exposure	

Table 21-144. Exposure Time [Opcode: DFh | Destination: 4]

Set Exposure Time	
Write Parameters	
Byte	**Description**
Bytes 0-1	Time Range = 0 to 10000 with step size 1
Set Camera's Absolute Exposure Time . UVC compliant Exposure Time Standard . Value 166 implies 16.6 ms	

Get Exposure Time
Data returned is in the same format as the Write Parameters.
Get Camera's Absolute Exposure Time. This value is multiple of 0.1 ms . For example, 166 is 16.6 ms.

Table 21-145. Enable Auto 2D Keystone [Opcode: F7h | Destination: 4]

Set Enable Auto 2D Keystone	
Write Parameters	
Byte	**Description**
Bytes 0-1	Horizontal resolution Range = 0 to 7680 with step size 10
Bytes 2-3	Vertical resolution Range = 0 to 4320 with step size 10
Bytes 4-7	CCA Detect Thresh Range = 0.0000 to 1.0000 with step size 0.0001 Format = u16.16
Bytes 8-9	Radius of the gaussian
Bytes 10-11	Gaussian centroid offset in pixels (from DMD edge)
Byte 12	Level of digital decimation for camera captures Range = 0 to 2 with step size 1
Byte 13	Enable or disable ROI processing for camera captures bit0: Pixel Sampling Enable
Set configuration and enable auto 2d keystone	

Table 21-146. Manual V Bus [Opcode: F8h | Destination: 4]

Set Manual V Bus	
Write Parameters	
Byte	**Description**
Byte 0	V Bus bit0: Manual V Bus Enable

Get Manual V Bus
Data returned is in the same format as the Write Parameters.

Table 21-147. Vision Status [Opcode: F9h | Destination: 4]

Set Vision Status	
Write Parameters	
Byte	**Description**
Bytes 0-3	Vision Status Word 0 bit0: Frame Buffer for Image capture not set or Buffer size is not sufficient. bit1: Attached camera does not support Uncompressed frame format. bit2: Selected frame resolution not supported by camera. bit3: Camera parameters set failed. bit4: Camera Probe Commit negotiation failed. bit5: Memory Pool failed for vision mode. bit6: Image Parameters failed. bit7: Memory request failed. bit8: Camera not found. bit9: Image captured failed. bit10: Camera's FOV not at center of Image bit11: Image Saturation Threshold Criteria not met bit12: Memory Request Fail for Auto Warp bit13: Memory Release Fail for Auto Warp bit14: Plan Fit Error bit15: Hole fit correction error bit16: Hole fit uncorrected error bit17: TODO bit18: TODO bit19: TODO bit20: TODO bit21: TODO bit22: TODO bit23: VIS AUTO SCREEN FIT ERR MEM REQUEST FAIL bit24: VIS AUTO SCREEN FIT ERR MEM RELEASE FAIL bit25: VIS AUTO SCREEN FIT ERR CAM2PROJ HOMOGRAPHY FAIL bit26: VIS AUTO SCREEN FIT ERR GAUSSIAN PAT NOT DETECTED bit27: VIS AUTO SCREEN FIT ERR GRAY SOLID PAT NOT DETECTED bit28: VIS AUTO SCREEN FIT ERR NOT ENOUGH CENTROIDS FOUND bit29: VIS AUTO SCREEN FIT ERR WARP MAP CREATION FAIL bit30: VIS AUTO SCREEN FIT ERR REFINE HOMOGRAPHY FAIL bit31: VIS AUTO SCREEN FIT ERR RESIZE HOMOGRAPHY FAIL
Bytes 4-7	Vision Status Word 1 bit0: VIS AUTO SCREEN FIT ERR FILL FACTOR PROC FAIL bit1: VIS AUTO 2DK ERR MEM REQUEST FAIL bit2: VIS AUTO 2DK ERR MEM RELEASE FAIL bit3: VIS AUTO 2DK ERR INCORRECT PAT INDEX bit4: VIS AUTO 2DK ERR GAUSSIAN PAT NOT DETECTED BLK bit5: VIS AUTO 2DK ERR GAUSSIAN PAT NOT DETECTED FLT bit6: VIS AUTO 2DK ERR NOT ENOUGH CENTROIDS FOUND bit7: VIS AUTO 2DK ERR CORRESPONDENCES NOT FOUND bit8: VIS AUTO 2DK ERR IVALID OPTICAL RAY bit9: VIS AUTO 2DK ERR INVALID CLOUD ELEMENT bit10: VIS AUTO 2DK ERR PLANE FIT FAIL bit11: VIS AUTO 2DK ERR MAP HOMOGRAPHY CALC FAIL bit12: VIS AUTO 2DK ERR SUBSAMPLING PTR NULL bit13: VIS AUTO 2DK ERR WARP MAP CREATION FAIL bit14: Unknown error in vision mode.
Command to read status information from Vision module.	

Get Vision Status
Data returned is in the same format as the Write Parameters.
Command to read status information from Vision module.

21.15 Warping

Warping

Table 21-148. Manual Warp Table [Opcode: 34h | Destination: 4]

Set Manual Warp Table	
Write Parameters	
Byte	**Description**
Bytes 0-1	Start index in the table for the data to be written
Bytes 2 - *	Warp map points in X, Y pairs where X, Y are in 13.3 fixed point format
This command writes to the warp map table that can be enabled using the Apply Manual Warping command. N warp map points can be loaded at a time to anywhere within the table. Maximum number of points that can be set using this command is 62 in the horizontal direction and 32 in the vertical direction. Overall max 1984 points. The number of points set by this command should match the number of control points specified using Set Manual Warp Control Points command. Each point is passed as two 13.3 fixed point numbers that represents X and Y coordinates. Since the total command packet size cannot exceed 512 bytes, the table shall be loaded by invoking the command multiple times with different start index.	

Get Manual Warp Table	
Read Parameters	
Byte	**Description**
Bytes 0-1	Start index in the table from which the data is to be read
Bytes 2-3	Number of entries to be read

Return Parameters	
Byte	**Description**
Bytes 0 - *	Warp map points in X, Y pairs where X, Y are in 13.3 fixed point format
This command reads from the warp map table already loaded using Set Manual Warp table. N warp map points (that does not exceed the command packet size) can be read at a time from anywhere within the table. Maximum table size is 1952.	

Table 21-149. Manual Warp Control Points [Opcode: 35h | Destination: 4]

Set Manual Warp Control Points	
Write Parameters	
Byte	**Description**
Byte 0	Indicates if the warp control points are explicitly defined by array of horizontal and vertical control points. 0 = Input image is uniformly divided to create equally spaced warp control points of dimension ((Number of Horizontal Control Points) x (Number of Vertical Control Points)). Warping map table loaded by the Set Manual Warp Table command is used as a two dimensional array with dimension (Warp Columns x Warp Rows). 1 = Warp control points are defined by Horizontal Control Points and Vertical Control Points parameters of this command. Warping map table loaded by the Set Manual Warp Table command is used as a two dimensional array with dimension (62 x 32).
Bytes 1 - *	If Control Points Defined By Array = 0, Send Number of Horizontal Control Points here. If Control Points Defined By Array = 1, Send 62 horizontal control points here in uint16 format
Bytes 1 - *	If Control Points Defined By Array = 0, Send Number of Vertical Control Points here. If Control Points Defined By Array = 1, Send 32 vertical control points here in uint16 format.
This command sets up the user defined control points of the warp map that shall be applied on top of the keystone correction, anamorphic scaling and other warp dependent feature settings if they are enabled. The warping map table loaded by the manual warp table write command is used as a two dimensional array with dimension which is defined based on the first argument of this command : TRUE = (Number of Horizontal Control Points) x (Number of Vertical Control Points) FALSE = (62 x 32) The points in the map should lie within the display area defined by display image size command. Any points lying outside the display area shall get cropped.	

Get Manual Warp Control Points	
Return Parameters	
Byte	**Description**
Byte 0	Indicates if the warp control points are explicitly defined by array of horizontal and vertical control points.
Bytes 1 - *	if Control Points Defined By Array = 0, Number of Horizontal control points followed by Number of Vertical control points are returned here if Control Points Defined By Array = 1, Actual(62) Horizontal control points followed by Actual(32) Vertical control points are returned here
This command gets up the user defined warping map control points.	

Table 21-150. Apply Manual Warping [Opcode: 36h | Destination: 4]

Set Apply Manual Warping	
Write Parameters	
Byte	**Description**
Byte 0	Enable bit 0: Warp Enabled
This command applies the manual warping control points and map table to the Warp HW defined by Set Manual Warp Control Points and Set Manual Warp Table respectively.	

Get Apply Manual Warping	
Return Parameters	
Byte	**Description**
Byte 0	Enable bit0: Manual Warp Enabled bit1: Surface Correction Warp Enabled bit2: Lens Correction Warp Enabled
This command returns whether warping feature is enabled or disabled for various use cases.	

Table 21-151. Smooth Warp Table [Opcode: 38h | Destination: 4]

Set Smooth Warp Table	
Write Parameters	
Byte	**Description**
Byte 0	Number of columns in the smooth warp matrix specified below (Range 3-5)
Byte 1	Number of rows in the smooth warp matrix specified below (Range 3-5)
Bytes 2 - *	Warp map points in X, Y pairs where X, Y are in 13.3 fixed point format
This command sets up the user defined MxN warping map that creates a parametric smooth curve. The edges connecting two warp points in this case are not straight lines but are 'smoother' – at the Warp Point, the edge is continuous and does not form a vertex (except for corners). This is done by fitting a 2nd degree polynomial curve to warp points, contrary to Write Manual Warp Table command which fits straight line to warp points.	

Get Smooth Warp Table
Data returned is in the same format as Write Parameters above.
This command returns the user defined MxN warping map points

Table 21-152. Manual Warp Table Update Mode [Opcode: 39h | Destination: 4]

Set Manual Warp Table Update Mode	
Write Parameters	
Byte	**Description**
Byte 0	0 = Overwrite Existing 1 = Merge with Existing
This command configures the warping engine warp points update mode. Only one manual warp can be applied to the warping engine. Hence, if multiple manual warp maps are required, they should be merged before applying. This Command enables or disables 'Manual Warp Merge Mode' In Merge mode, any new map written will be merged with existing manual warp map. Maps can be merged one after the other. For each map, control points need to be set-up using Set Manual warp control points command and warp points should be written using Write Manual Warp Command.	

Get Manual Warp Table Update Mode
Data returned is in the same format as Write Parameters above.
This command returns the set warp table write mode

Table 21-153. Uncorrected Point Cloud [Opcode: ACh | Destination: 4]

Get Uncorrected Point Cloud	
Read Parameters	
Byte	**Description**
Bytes 0-1	Starting index of data retrieval
Bytes 2-3	Number of entries to be returned

Return Parameters	
Byte	**Description**
Bytes 0 - *	Uncorrected Point Cloud data Range = -32768 to 32767 with step size 1
Get pre-processed 3D point cloud coordinates beginning at Index and ending after NumEntries have been returned	

Table 21-154. Corrected Point Cloud [Opcode: ADh | Destination: 4]

Get Corrected Point Cloud	
Read Parameters	
Byte	**Description**
Bytes 0-1	Starting index of data retrieval
Bytes 2-3	Number of entries to be returned

Return Parameters	
Byte	**Description**
Bytes 0 - *	Corrected Point Cloud data Range = -32768 to 32767 with step size 1
Get post-processed 3D point cloud coordinates beginning at Index and ending after NumEntries have been returned	

Table 21-155. Centroids [Opcode: AEh | Destination: 4]

Get Centroids	
Read Parameters	
Byte	**Description**
Bytes 0-1	Starting index of data retrieval
Bytes 2-3	Number of entries to be returned

Return Parameters	
Byte	**Description**
Bytes 0 - *	Uncorrected camera points data Range = -32768 to 32767 with step size 1
Get the 2D centroids array beginning at Index and ending after NumEntries have been retrieved	

Table 21-156. Surface Warp Map [Opcode: AFh | Destination: 4]

Get Surface Warp Map	
Read Parameters	
Byte	**Description**
Bytes 0-1	Starting index of data retrieval
Bytes 2-3	Number of entries to be returned

Return Parameters	
Byte	**Description**
Bytes 0 - *	Surface warp map data Range = -32768 to 32767 with step size 1
Get redistributed 2D surface warp map data	

Table 21-157. Point Cloud Dimensions [Opcode: B0h | Destination: 4]

Get Point Cloud Dimensions	
Return Parameters	
Byte	**Description**
Bytes 0-1	Horizontal resolution of point cloud
Bytes 2-3	Vertical resolution of point cloud
Retrieve dimensions of point cloud, minimum 8x8, maximum 62x32	

Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

- Spcc Control Points(53h) command added....156
- PCC Coefficients Direct(54h) command added....156
- All the commands in this category have been newly added....188
- EEPROM Lock State(6Ch)added....190
- All Lamp and CW related commands removed....190

IMPORTANT NOTICE AND DISCLAIMER

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265