DLPCx54x Application Note

Manual Warping and Blending Programmer's Guide

Revision History

Document Version	Date	Author(s)	Notes
0.1	06 September 2019	Sureshkumar Manimuthu	Initial Release
0.2	07 April 2020	Aayush Mittal	Control Points Update New Commands Added Flash Version Updated
0.3	10 August 2020	Sanjeev Kumar	Details added about how 62x332 warp points generated for manual warp table (0x34). Updated Smooth Warp command section applicable to SW version 0.11.x onwards.
0.4	18 September 2020	Sreeram Subramanian	Section 6.1 updated to explain ControlPointsSetByArray parameter and 6.2 updated to specify max points permissible.
0.5	25 January 2021	Sreeram Subramanian	Added blending map related sections to the document.
0.6	26 February 2021	Sreeram Subramanian	Updated the link to download the floating point conversion and RLE2 compression script.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to TI's Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

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

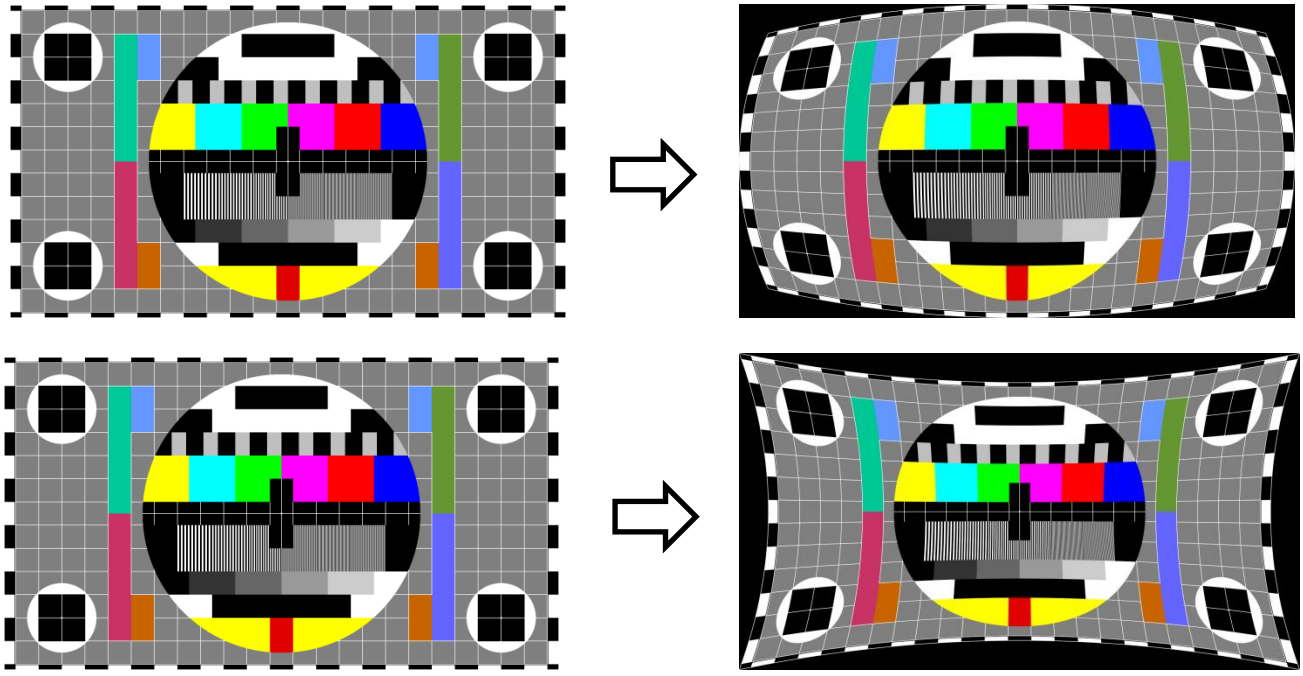
Table of Contents

1. Introduction to warping 6
2. Control Point Table 6
3. Manual Warp Table 6
4. Table Constraints 7
5. Example Warp Table 8
6. Manual Warping Commands 10
7. Introduction to blending 20
8. Blend Map Control Points 20
9. Blend Map Gain values 21
10. Blend Map Offset value 21
11. Constraints 21
12. Manual blending Commands 22
13. Storing in EEPROM or Secondary Flash 29

1. Introduction to warping

DLPCx54x has a powerful warping engine that enables performing arbitrary geometric modulation of the input video frame. This document describes the APIs and commands 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.

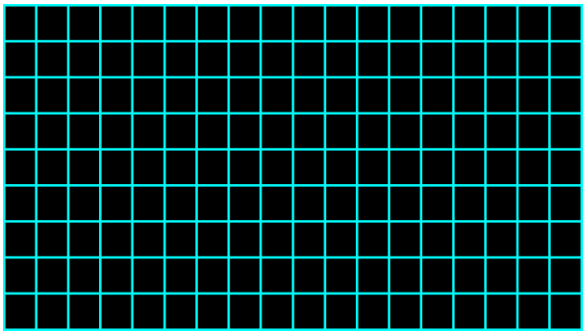


2. Control Point Table

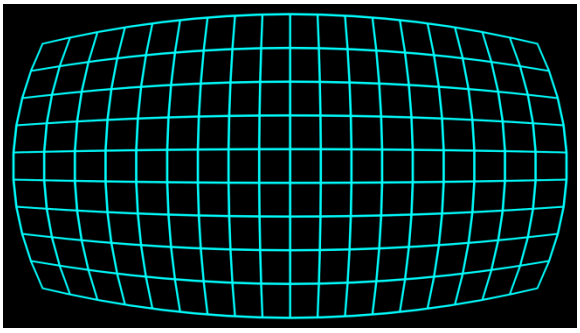
Control point table used to configure position of source points. These source points will be moved by the warping engine to their respective destination Manual Warp Table. This will also determine the number of rows and columns in Manual Warp Table. The number of destination points is implied to be same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

3. Manual Warp Table

Manual warp table is a way to configure the warping engine using a 2D array of points called as Warping Map. These points represent the destination position of the moved input pixel on the DMD.



Source Position (Control Points)



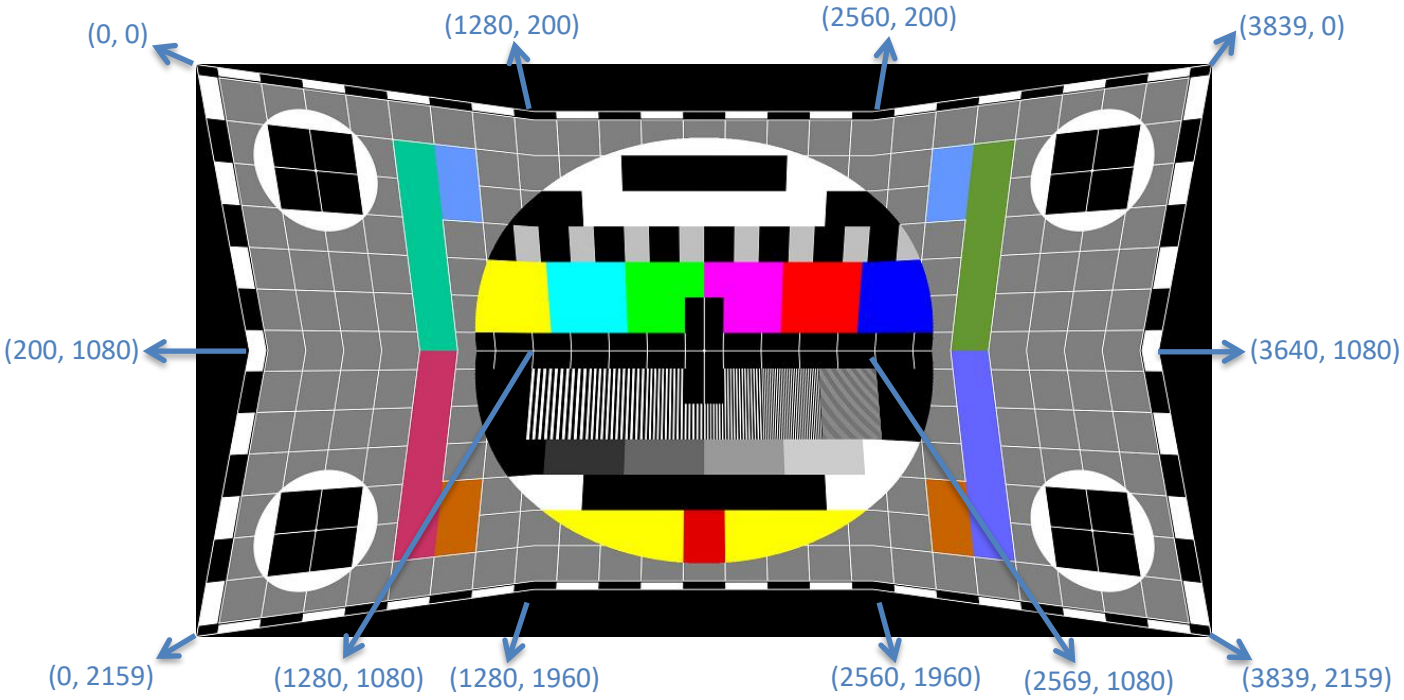
Destination Position (Warping Map Points)

The intersection points in the above image represent the 2D array of points / positions. The left side image represents the original pixel positions that will be moved in the input frame (Control Points). The right side image represents the corresponding pixel's 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.

4. Table Constraints

There are 62 horizontal and 32 vertical control points used to define a 62 X 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 number of horizontal points and vertical points in range 2-62 and 2-32 respectively. In later case, all points will remain equidistant. For unequal distribution, former mode needs to be used.
The total number of points in the user defined Table should not exceed the number of points in control point map above. A 62 X 32 control point map is most optimal and provides 1984 points. Since the warp table represents the whole output rectangle, it is mandatory to have at least 2 rows and 2 columns. **The points in the warp table should be monotonically increasing in the left to right and top to bottom direction (points cannot crisscross each other)**. If any of these conditions is violated, the command / API would return failure.

5. Example Warp Table



Below example depicts a butterfly like warping use case with 4x3 control points on a 3840x2160 Product Resolution. For simplicity, it uses a symmetric warping and integer positions. Customers can choose to use asymmetric map and real number positions (Fixed point 13.3 format).

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 Followinf Map is generated:

[0.0, 0.0]	[1279.4, 0.0]	[1586.6, 0.0]	[3839.0, 0.0]
[0.0, 1079.0]	[1279.4, 1079.0]	[1586.6, 1079.0]	[3839.0, 1079.0]
[0.0, 2159.0]	[1279.4, 2159.0]	[1586.6, 2159.0]	[3839.0, 2159.0]

Following equation shall be used for arriving at contrl 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]

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	800
6	3839.0	30712
7	0.0	0
8	200.0	1600
9	1080.0	8649
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

6. Manual Warping Commands

Following are the manual warping commands provided for configuring the warping engine in closed system architecture (DLPCx54x Controller).
Note : Conforms with Flash Version 0.7.2 or later.

6.1. 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: If 'Control Points Defined By Array' is set to True, all 62 Horizontal and 32 Vertical Points should be provided. If set to false, the number of control points will be as per number of columns and rows specified and will be distributed evenly/equidistant from each other.
Example Command Packet:

1. Control Points NOT Defined By Array

Byte Index	Value(s)	Description
0	0x44	Command Packet Header (Destination 4 , Response Required)
1	0x35	Command Opcode
2	0x00	Control Points Defined By Array (Byte 11 and ahead, set to False)
3-4	0x0F00	Input image width (3840)
5-6	0x0870	Input image height (2160)
7-8	0x0004	Number of columns
9-10	0x0003	Number of rows
11-134	0xXXXX	Horizontal Control Points (Don't care)
135-198	0xXXXX	Vertical Control Points (Don't care)

2. Control Points Defined By Array

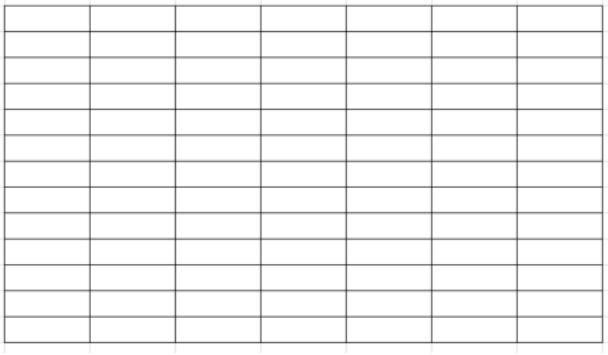
Byte Index	Value(s)	Description
0	0x44	Command Packet Header (Destination 4 , Response Required)
1	0x35	Command Opcode
2	0x01	Control Points Defined By Array (Byte 11 and ahead, set to True)
3-4	0x0F00	Input image width (3840)
5-6	0x0870	Input image height (2160)
7-8	0xXXXX	Number of columns (Don't care)
9-10	0xXXXX	Number of rows (Don't care)
		Horizontal Control Points
11-12	0x0000	X1 = 0
13-14	0x003F	X2 = 63
15-16	0x007E	X3 = 126
17-18	0x00BD	X4 = 189
19-20	0x00FC	X5 = 252
21-22	0x013B	X6 = 315
23-24	0x017A	X7 = 378
25-26	0x01B9	X8 = 441

27-28	0x01F8	X9 = 504
29-30	0x022D	X10 = 557
31-32	0x0276	X11 = 630
33-34	0x02B4	X12 = 692
35-36	0x02F3	X13 = 755
37-38	0x0332	X14 = 818
39-40	0x0371	X15 = 881
41-42	0x03B0	X16 = 944
43-44	0x03EF	X17 = 1007
45-46	0x042E	X18 = 1070
47-48	0x046D	X19 = 1133
49-50	0x04AC	X20 = 1196
51-52	0x04EB	X21 = 1259
53-54	0x052A	X22 = 1322
55-56	0x0569	X23 = 1385
57-58	0x05A8	X24 = 1448
59-60	0x05E7	X25 = 1511
61-62	0x0626	X26 = 1574
63-64	0x0666	X27 = 1638
65-66	0x06A4	X28 = 1700
67-68	0x06E3	X29 = 1763
69-70	0x0722	X30 = 1826
71-72	0x0761	X31 = 1889
73-74	0x079F	X32 = 1951
75-76	0x07DE	X33 = 2014
77-78	0x081D	X34 = 2077
79-80	0x085C	X35 = 2140
81-82	0x089B	X36 = 2203
83-84	0x08DA	X37 = 2266
85-86	0x0919	X38 = 2329
87-88	0x0958	X39 = 2392
89-90	0x0997	X40 = 2455
91-92	0x09D6	X41 = 2518
93-94	0x0A15	X42 = 2581
95-96	0x0A54	X43 = 2644
97-98	0x0A93	X44 = 2707
99-100	0x0AD2	X45 = 2770
101-102	0x0B11	X46 = 2833
103-104	0x0B50	X47 = 2896
105-106	0x0B8F	X48 = 2959
107-108	0x0BCE	X49 = 3022
109-110	0x0C0D	X50 = 3085
111-112	0x0C4C	X51 = 3148
113-114	0x0C8A	X52 = 3210
115-116	0x0CC9	X53 = 3273
117-118	0x0D08	X54 = 3336
119-120	0x0D47	X55 = 3399

121-122	0x0D86	X56 = 3462
123-124	0x0DC5	X57 = 3525
125-126	0x0E04	X58 = 3588
127-128	0x0E43	X59 = 3651
129-130	0x0E82	X60 = 3714
131-132	0x0EC1	X61 = 3777
133-134	0x0EFF	X62 = 3839
		Vertical Control Points
135-136	0x0000	Y1 = 0
137-138	0x0046	Y2 = 70
139-140	0x008B	Y3 = 139
141-142	0x00D1	Y4 = 209
143-144	0x0117	Y5 = 279
145-146	0x015C	Y6 = 348
147-148	0x01A2	Y7 = 418
149-150	0x01E8	Y8 = 488
151-152	0x022D	Y9 = 557
153-154	0x0273	Y10 = 627
155-156	0x02B9	Y11 = 697
157-158	0x02FE	Y12 = 766
159-160	0x0344	Y13 = 836
161-162	0x038A	Y14 = 906
163-164	0x03CF	Y15 = 975
165-166	0x0415	Y16 = 1045
167-168	0x045B	Y17 = 1115
169-170	0x04A1	Y18 = 1185
171-172	0x04E6	Y19 = 1254
173-174	0x052C	Y20 = 1324
175-176	0x0572	Y21 = 1394
177-178	0x05B7	Y22 = 1463
179-180	0x05FD	Y23 = 1533
181-182	0x0643	Y24 = 1603
183-184	0x0688	Y25 = 1672
185-186	0x06CE	Y26 = 1742
187-188	0x0714	Y27 = 1812
189-190	0x0759	Y28 = 1881
191-192	0x079F	Y29 = 1951
192-194	0x07E5	Y30 = 2021
195-196	0x082A	Y31 = 2090
197-198	0x086F	Y32 = 2159

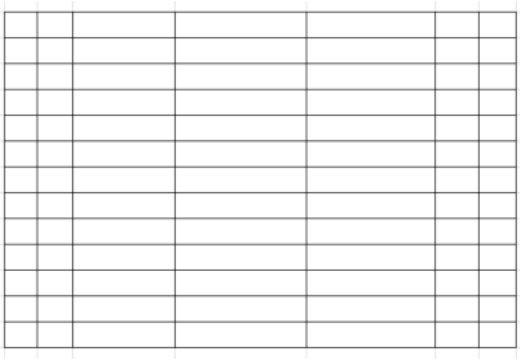
When to use Control Points Defined By Array:-

When Control points defined by Array = 0, the whole input rectangle to warp is divided into MxN equal parts where MxN are specified as number of rows and columns in the SetManualWarpControlPoints() command. Number of Warp table points must match with this MxN.



Instead of equally divided control points, if you want to make some areas sparse and some areas dense, you want to use SetManualWarpControlPoints() with Control points defined by Array = 1. Since the row and columns are straight lines at the input of warp, all you need to specify is the max 64 horizontal points and max 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 below for example.



6.2. CMD_WriteManualWarpTable [Opcode: 0x34, Destination: 4]

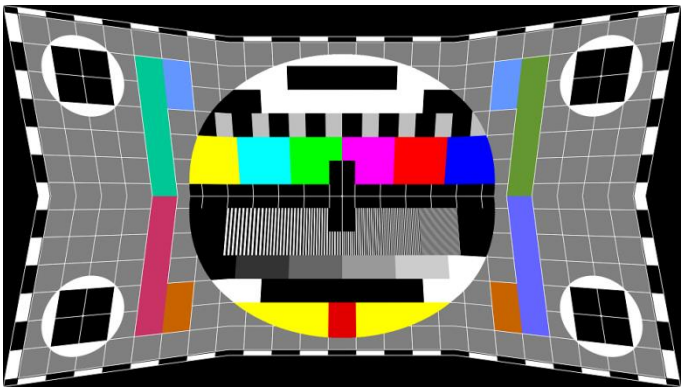
This command loads the manual warp table into the system. It has start index and arbitrary number of points as parameters. The start index is a 16 bit number that represents the linear index into the 2D warp table.

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. If total number of points passed in Manual Warp Table is less than 62x32 then software generates the warp map for all 62x32 control points from the warp map of input even spaced control points. For each 62x32 control points it finds the four closest control points in even spaced warp map and then using bilinear interpolation calculate the output position.

Note:

1. Manual warp map is applied on top of any scaling and keystone correction performed by the warp engine
2. Manual warp control points should be set before using this command

Example Output:



Example Command Packet:

Byte Index	Value(s)	Description
0	0x54	Command Packet Header (Destination 4, Length Present, Response Required)
1	0x34	Command Opcode
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 pointed out above the number of bytes in one command cannot exceed 512. So when you have to send a large number of warp points, it will have to be sent as multiple packets.

An example method for sending 62x32 warp points is shown below..

Command #1 – index=0, followed by 124 pairs of X, Y (124*2*2 = 496 bytes)
Command #2 – index =124, followed by next 124 pairs of X, Y
Command #3 – index =248, followed by next 124 pairs of X, Y

And so on

Command #16 – index = 930, followed by last 124 pairs of X, Y

6.3. CMD_ConfigureSmoothWarp[Command ID : 0x38, Destination: 4]

This command also loads Manual Warp Table into the system. The edges connecting two warp points in this case are not straight lines but are 'smoother' ie, At the Warp Point, 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 $aX^2 + bX + c$) to warp points, contrary to Write Manual Warp Table command which fits straight line to warp points.

Only a Warp Table with 3 rows and 3 columns (9-points) can be defined with this command.

Note:

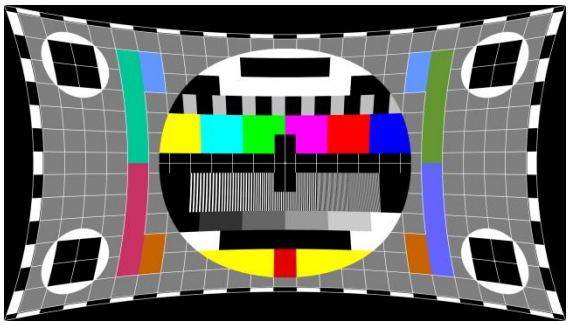
1. Manual warp map is applied on top of any scaling and keystone correction performed by the warp engine
2. Manual warp control points command is NOT required before using this command since it always assumes uniform 3X3 map

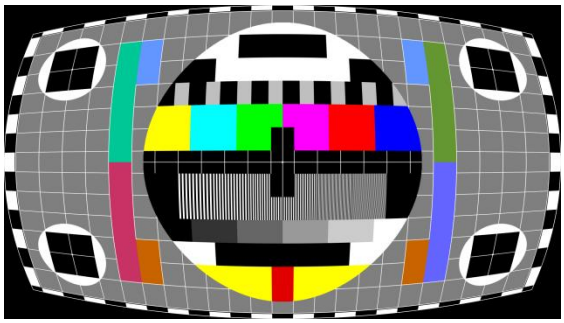
Two kinds of Smooth Warp possible in the command

1. Only parametric smooth curve defined by points on the top, bottom, left and right sides of the image.
2. Integration four corners warp map and smooth warp map. Four corners warp defined by TOP_LEFT, TOP_RIGHT, BOTTOM_LEFT and BOTTOM_RIGHT warp points. Smooth warp defined by TOP_CENTER, LEFT_CENTER, RIGHT_CENTER and BOTTOM_CENTER points.

Input warping map points are in the following order
- TOP_LEFT
- TOP_CENTER
- TOP_RIGHT
- LEFT_CENTER
- MID_CENTER
- RIGHT_CENTER
- BOTTOM_LEFT
- BOTTOM_CENTER
- BOTTOM_RIGHT

Example Outputs:





Example Command Packet:
SmoothWarp – Parametric smooth curve

Byte Index	Value(s)	Description
0	0x44	Command Packet Header (Destination 4 , Response Required)
1	0x38	Command Opcode
2	0x00	Parametric Smooth Curve
3-4	0x01	X1 = 0.0
5-6	0x0000	Y1 = 0.0
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

SmoothWarp – Integration four corners warp map and smooth warp map

Byte Index	Value(s)	Description
0	0x44	Command Packet Header (Destination 4 , Response Required)
1	0x38	Command Opcode
2	0x01	Integration four corners warp map and smooth warp map
3-4	0x01	X1 = 0.0
5-6	0x0000	Y1 = 0.0
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

6.4. CMD_SetManualWarpTableUpdateMode [Command ID: 0x39, Destination: 4]

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 map should be written using Write Manual Warp Command.

Note: Merge mode will also work with Smooth Warp Command, Control Points do not need to be set-up for this.

In Overwrite mode, any manual warp table written using command replaces any existing manual warp map, i.e, any new warp map passed is not merged. System boots in this mode by default.

Example Command Packet:

Byte Index	Value(s)	Description
0	0x44	Command Packet Header (Destination 4 , Response Required)
1	0x39	Command Opcode
2	0x01	Enable Merge Mode

6.5. CMD_ApplyManualWarping[Command ID: 0x36, Destination: 4]

This Command is used to Enable/Disable manual warping feature in the system.

Note: A manual warp map should exist in the system prior to enabling using this command

Example Command Packet:

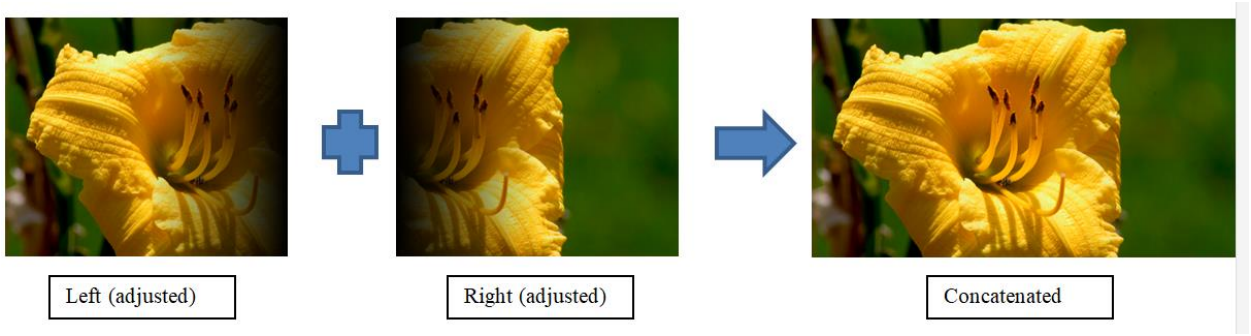
Byte Index	Value(s)	Description
0	0x44	Command Packet Header (Destination 4 , Response Required)
1	0x36	Command Opcode
2	0x01	Enable Manual Warping

7. Introduction to blending

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

The manual blending can be used in applications like achieving seamless overlap of two or more projectors, to increase the projected image area and improve on-screen resolution and brightness. The EBF block compensate 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.

In the below image we can see 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.



Consider a monochromatic pixel p_{in}, the EBF block can modify the intensity of this pixel by applying a gain and an offset:

$$\boldsymbol{p_{out} = p_{in} * GAIN + OFFSET}$$

8. Blend Map Control Points

User needs to provide 63 horizontal and 32 vertical locations of control points. The control points must lie within the resolution of image ie : 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. So user will have 2016 control points on the image and the gain and offset values will have to be declared for these for each of the colour channel R,G,B.

9. Blend Map Gain values

This gain will be multiplied with the pixel intensity at that sampling pixel. Gain needs to be applied for each of the RGB channel of each sampling pixel. Gain value lies between 0 - 1.999 (Precision: u1.12). The user needs to provide the gain value in the fixed point format of u1.12.

10. 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. User needs to 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 here.

11. Constraints

There are 63 horizontal and 32 vertical control points used to define a 63 X 32 map on the input image. User needs to give half the control points in left half of image and half in right half of image. What this means is that from a total of 2016 control points 1008 control points need to be in each half of image when image is divided vertically in two halves from the middle of horizontal resolution.
While declaring the horizontal control point's locations user needs to make sure to have the 31st control point at the centre of image horizontally.

12. Manual blending Commands

Following are the manual blending commands provided for configuring the EBF Block (DLPCx54x Controller).

12.1. CMD_EnableEdgeBlending [Command ID: 0x23]

This command Enables or Disables the EBF Block. This can also be used to 'get' the current state of EBF block.
Example Command Packet:

Byte Index	Value(s)	Description
0	0x44	Command Packet Header (Destination 4 , Response Required)
1	0x23	Command Opcode
2	0x01	Enable Edge Blending

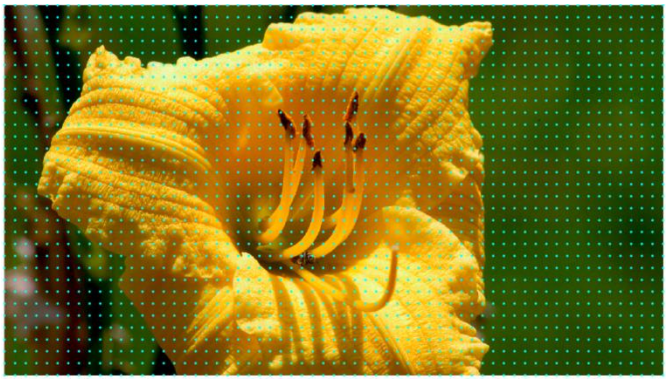
12.2. CMD_SetBlendMapControlPoints [Opcode: 0x29]

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 control point's location must in range of 0 – (Hres-1) for the horizontal control points and 0 – (Vres-1) for the vertical control points.

Below there is an example of how this command will be used for an image of resolution 1280x1080 and the control points are evenly spaced here.
Ideally user would want to have more number of control points in the region where gain or offset change is required. But user needs to keep in mind of constraints mentioned before.

Note:

1. The 31st Horizontal control point location is at the center of image horizontally.
2. Half of control point's locations are in left half of image and other half in right half image.



Example :

Byte Index	Value(s)	Description
0	0x54	Command Packet Header (Destination 4 , Response Required)

1	0x29	Command Opcode
2 – 3	1280	Horizontal image resolution
	1080	Vertical image resolution
4-5		**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
18-19	0x007D	H6 = 125
20-21	0x0091	H7 = 145
22-23	0x00A6	H8 = 166
24-25	0x00BB	H9 = 187
26-27	0x00CF	H10 = 207
28-29	0x00E4	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
109 - 110	0x0432	H52 = 1074
111 – 112	0x0446	H53 = 1094
113 – 114	0x045B	H54 = 1115
115 – 116	0x0470	H55 = 1136
117 - 118	0x0484	H56 = 1156
119 - 120	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	0x00D2	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

12.3. CMD_ GetBlendMapControlPoints [Command ID : 0x29]

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

Example:

Byte Index	Value(s)	Description
0	0x44	Command Packet Header (Destination 4 , Response Required)
1	0x29	Command Opcode

12.4. CMD_ SetBlendMapGainValues [Command ID: 0x2B]

Using this command user passes the Gain values for the 2016 (63x32) control points.
The gain values needs to be specified for each of the color channel.

Gain value lies between 0 - 1.999 (Precision: u1.12). The user needs to provide the gain value in the fixed pt format of u1.12.
Example - Suppose user wants to set the gain for a particular control point as 1.25 then 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 command. In u1.12 1.25 will be 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.

Suppose user passes gain values as uncompressed then user would need to use this command multiple times because of the restriction due to below Note(See Note below).

User has the following 4 options for passing gain values using the command,

1. Pass Compressed values (RLE2) and broadcast it to all colour channels
2. Pass Uncompressed values and broadcast it to all colour channels

3. Pass separate Compressed values for each of colour channel
4. Pass separate Uncompressed values for each of colour channel

Note: The maximum number of bytes that can be passed at a time is 510 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. Since all values cannot be passed at the same time user will have to use the same command multiple times and in each time updating the Index value from which user wants to pass value.

Below is an example of how values need to be passed to command. The option chosen is of Compress Enable and Broadcast enable.
After compression the 2016 values of gain for a single color channel reduces to 64 in this example.

Example:

Byte Index	Value(s)	Description
0	0x54	Command Packet Header (Destination 4, Length Present, Response Required)
1	0x2B	Command Opcode
2	0x01	Compression Enabled – This means command will accept compressed values now
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	0x0F7F	G33= 3967
75 – 76	0x003F	G34 = 63
77 – 78	0x0E75	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
93 – 94	0x0A57	G43 = 2647
95 – 96	0x003F	G44 = 63
97 – 98	0x094D	G43 = 2381
99 – 100	0x003F	G46 = 63
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
115 – 116	0x003F	G54 = 63
117 - 118	0x0425	G55 = 1061
119 - 120	0x003F	G56 = 63
121-122	0x031C	G57 = 796
123-124	0x003F	G58 = 63
125-126	0x021A	G59 = 538
127-128	0x003F	G60 = 63
129 - 130	0x0111	G61 = 273
131-132	0x003F	G62 = 63
133-134	0x0007	G63 = 7

12.5. CMD_ GetBlendMapGainValues[Command ID : 0x2B]

This command is used to get the Gain values as passed by user. User needs to select the Color Channel select as to which option was used in CMD_ SetBlendMapGainValues and also pass the index and number of values that user wants to get.

If compressed values were passed then values get will be of compressed only. If uncompressed values were passed then user will see uncompressed values.

Example of getting values. Here we are getting the values that we passed in the CMD_ SetBlendMapGainValues example.

Byte Index	Value(s)	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	No of Values to be read. Here we are reading 64 values

12.6. CMD_ SetBlendMapOffsetValues[Command ID : 0x2D]

Using this command user passes the Offset values for the 2016 (63x32) 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 pt format is -255 to +255. The user needs to provide the offset value in the internal floating pt format of s1m8e4 to the command.

A matlab script to convert standard floating-point value to the format required by the command can be downloaded from here.

What this code does is that it takes the input from user in the standard floating pt format in the range of -255 to +255 and converts it to internal floating pt value which actually needs to be passed to command.

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.

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. A sample RLE2 compression algorithm can be downloaded here.

User has the following 4 options for passing gain values using the command,

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

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

12.7. CMD_ GetBlendMapOffsetValues[Command ID : 0x2D]

This command is used to get the Offset values as passed by user. User needs to select the Color Channel select as to which option was used in CMD_SetBlendMap Offset Values and also pass the index and number of values that user wants to get.

If compressed values were passed then values get will be of compressed only. If uncompressed values were passed then user will see uncompressed values.

This command is similar to CMD_ GetBlendMapGainValues.

12.8. CMD_ApplyBlendMap[Command ID : 0x2D]

This command is used to apply the Blend map to EBF Block. This command needs to be used to pass the values passed by user of Blend map to API.

Example Command Packet:

Byte Index	Value(s)	Description
0	0x44	Command Packet Header (Destination 4 , Response Required)
1	0x2D	Command Opcode
2	0x01	Enable Edge Blending

13. Storing in EEPROM or Secondary Flash

There is an option of storing the Blend map control points, Gain and offset values either in the Secondary flash memory or in EEPROM.
Whether the values need to be stored in EEPROM or in Secondary flash needs to be decided before generating the flash file in DLP Composer software.

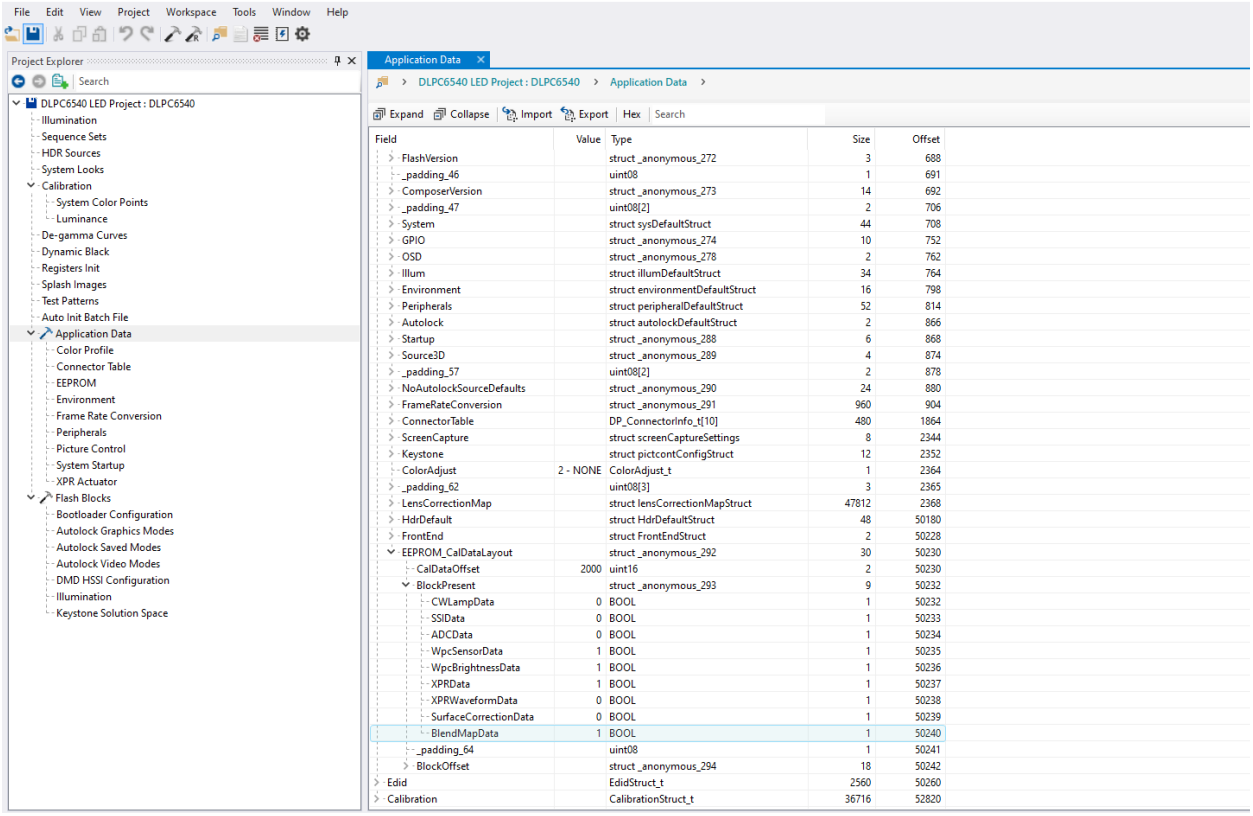
Below is shown how to set storage to EEPROM or Secondary flash.

13.1. Storing in EEPROM

1. Open DLP Composer software and select the required DLP composer project file and open it.
2. In the project explorer option on right click on the Application Data option.

3. Now expand the EEPROM_CalDataLayout option.
4. In this expand the Block Present option.
5. For the **BlendMapData** set the value as 1 to set the storage as EEPROM.

   In the below picture we can see the **BlendMapData** whose value needs to assigned as 1 here.



13.2. Storing in Secondary Flash

1. Open DLP Composer software and select the required DLP composer project file and open it.
2. In the project explorer option on right click on the Application Data option.
3. Now expand the EEPROM_CalDataLayout option.
4. In this expand the Block Present option.
5. For the **BlendMapData** make sure it it set to 0 value.
6. In the Application Data option expand the Configuration and then system.
7. Here we need to set the address in the **BlendMapStoreFlashAddress.** We need to pass the address here at which we want to store the Blend Map data.