DLPC6401 Formatter-Only Software

Programmer's Guide

Literature Number: DLPU012A
December 2013–Revised February 2019

Contents

Scope

This programmer's guide specifies the command and control interface to the DLPC6401 (formatter only) DLP® Pico™ projector software. It defines all applicable communication protocols: initialization, default settings, and timing and control register bit definitions. Figure 1 shows an example of a typical formatter-only system .

Trademarks

Pico, BrilliantColor, DynamicBlack are trademarks of Texas Instruments.
DLP is a registered trademark of Texas Instruments.

Related Documentation

This user's guide references the DLPC6401 device data sheet.

Applicable Documents

The following non-TI documents are for reference only: I²C Bus Specification and Philips Semiconductor 1994 *Desktop Video Data Handbook*.

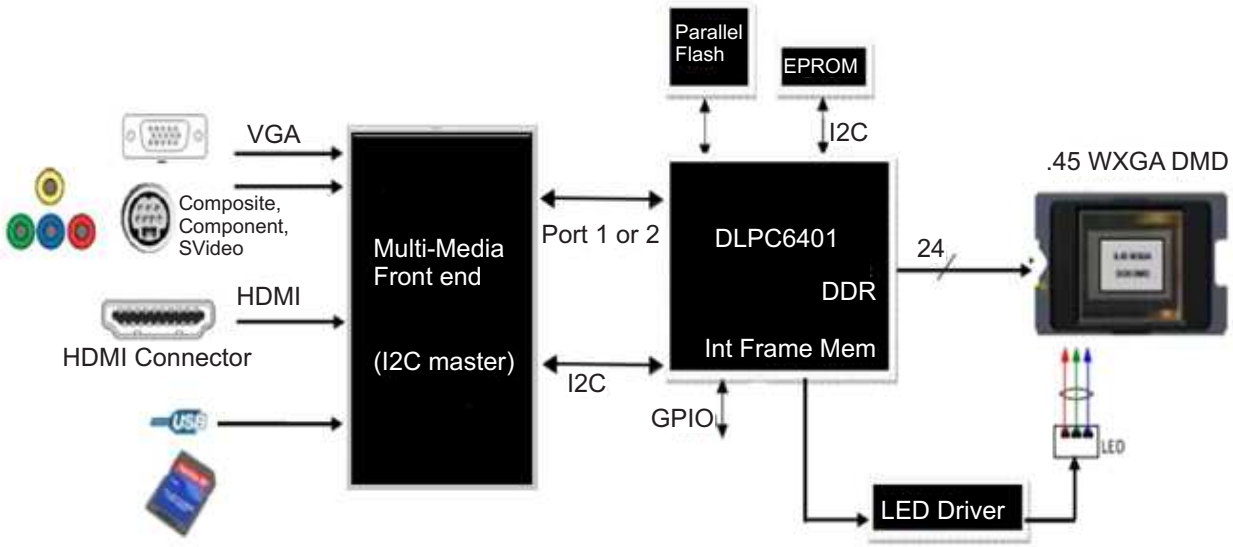


Figure 1. Typical Formatter-only System

Command Interface Protocol

1.1 Interface Standard

DLPC6401 commands can be executed using I²C.

NOTE: Port1 is 30-bit parallel port, and Port2 is LVDS that can interface front end.

1.1.1 I²C Interface

The protocol used in communicating information to DLPC6401 consists of a serial data bus conforming to the Philips I²C specification, up to 400 KHz.

1.1.2 I²C Ports

DLPC6401 supports two I²C Ports, Port-0 and Port-1.

Port-0 is used for interfacing to EEPROM. While using this port, DLPC6401 acts as I²C master. The I²C write address to access EEPROM from DLPC6401 is 0xA8 and read address is 0xA9.

Port-1 is primarily used for command and control interface. While using this port, DLPC6401 acts as an I²C slave.

1.2 I²C Command Protocol

1.2.1 WRITE Command

The I²C Address for the command interface is 8-bit, followed by a 8-bit sub-address. The sub-address is followed by variable length bytes of data. The length of bytes written depends upon the sub-address.

Table 1-1. WRITE Command Structure(1)(2)

Address	Sub-Address	Data			
(8-bit)	(8-bit)	(n-bytes...)			
0x34	CMD	0xdd	0xdd	0xdd	0xdd ...

(1) In the WRITE command, Bit 7 of CMD is always set to 1.
(2) DLPC6401 device will know how many bytes to expect from the host depending upon the command. Please refer to the command description for the command structure.

1.2.2 READ Command

Table 1-2. READ Command Structure

Address	Sub-Address	Data			
(8-bit)	(8-bit)	(n-bytes...)			
0x34	CMD				
(8-bit)	(8-bit)	(n-bytes...)			
0x34	STAT	0xdd	0xdd	0xdd	0xdd ...

1.2.2.1 STATUS (STAT) Byte

Table 1-3. (1)(2)(3)

Bits	Description	Reset	Type
7:2	Reserved	d0	wr
1	**Command or Parameter Error** 0: false 1: true		
0	**System Ready** 0: the system is currently busy (initializing, executing a command, and so forth) 1: the system is ready to execute a command		

(1) STAT byte indicates system status with reference to the requested READ command.
(2) While performing the read operation, the host has to repeat reading the status until the ready byte is set.
(3) If the system ready state is 1 only, then the next bytes following the STAT byte are valid read data bytes.

1.2.3 Extended WRITE Command

The typical I²C command address is 8-bit. This supports 256 READ or WRITE commands.

This command provides a backup command, if the commands set need to be extended beyond 256 commands.

Address	Sub-Address(1)	Extended-Address	Data		
(8-bit)	(8-bit)	(8-bit)	(n-bytes…)		
0x34	0x01	EADDR(2)	0xdd	0xdd	0xdd …

(1) Sub-Address = 1 is reserved for indicating extended address.
(2) EADDR is treated as extended sub-address.

1.2.4 Extended READ Command

The typical I²C command address is 8-bit. This supports 256 READ or WRITE commands.

In case, if the commands set needs to be extended, then the following approach is used.

Table 1-4. (1)(2)(3)

Address	Sub-Address	Extended-Address	Data		
(8-bit)	(8-bit)				
0x34	0x01	CMD			
(8-bit)	(8-bit)	(n-bytes…)			
0x35	STAT	0xdd	0xdd	0xdd	0xdd ...

(1) STAT byte indicates system status with reference to the requested read command.
(2) While performing the read operation, the host repeats reading the status, until the ready byte is set.
(3) If system ready state is 1 only, then the next bytes following the STAT byte are valid read data bytes.

1.3 DLPC6401 Command Interface Examples

1.3.1 READ Command Example

Figure 1-1 demonstrates an example of a reading system status command.

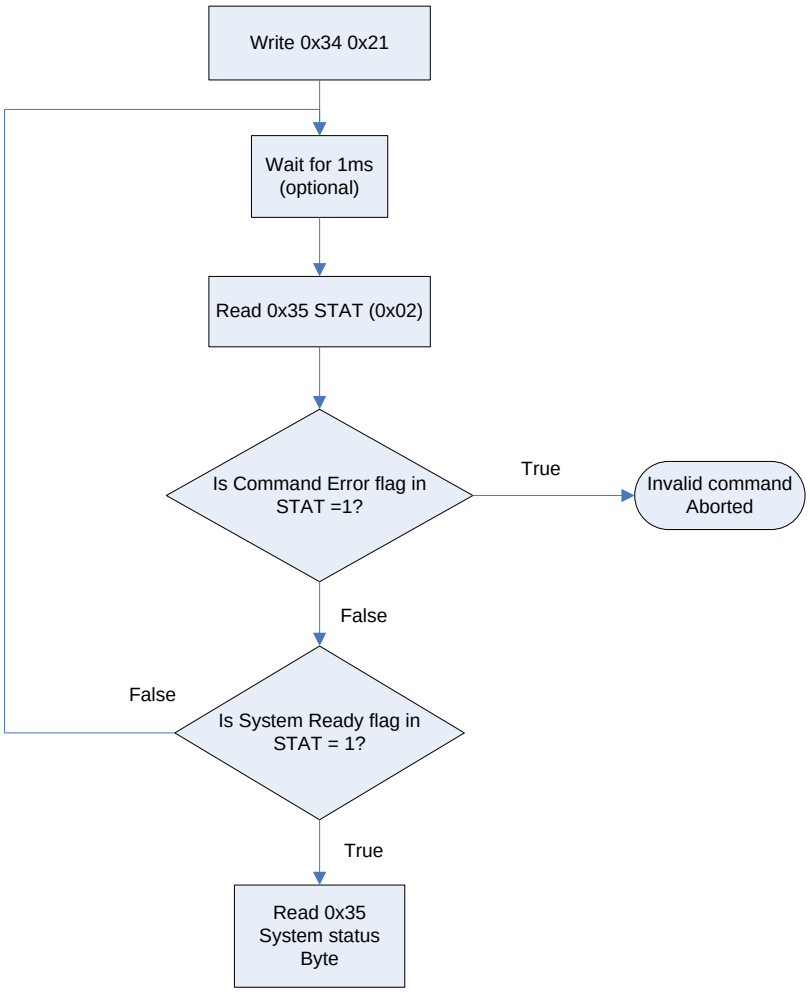


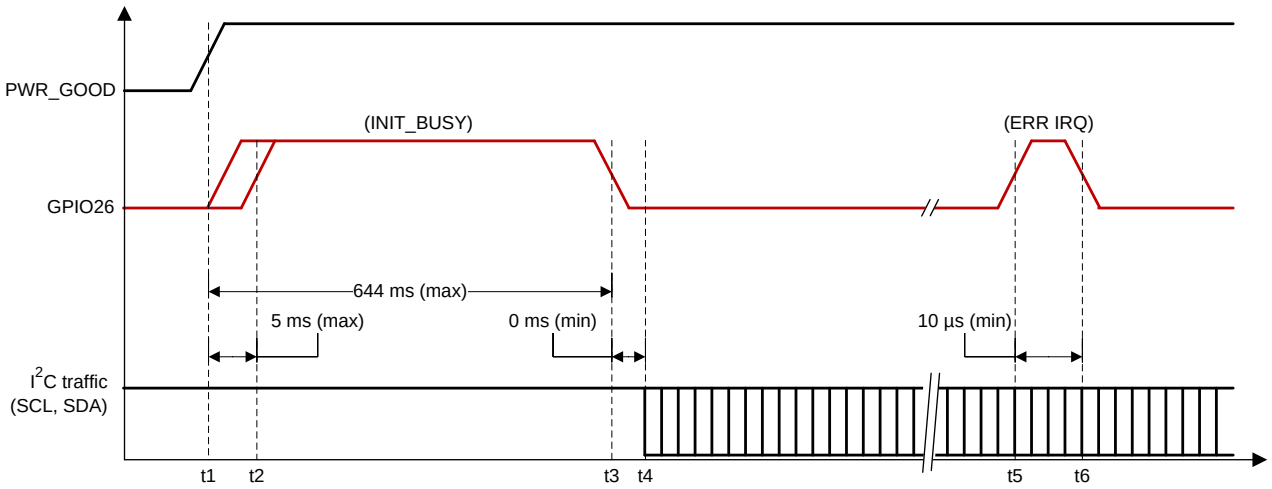
Figure 1-1. Reading System Status Command Example

Initialization

2.1 Introduction

The initialization method used in the DLPC6401 application is discussed in the following sub-sections.

2.2 Initialize the DLPC6401 Formatter Electronics



t2: GPIO26 is driven high within 5-ms after reset is released to indicate auto-initialization is busy

t3: I²C access to DLPC6401 should not start until GPIO26 (INIT_BUSY flag) goes low (this should not occur within 100 ms from the release of RESETZ

t5: An active high pulse on GPIO26 following the initialization period indicates an error condition has been detected. The source of the error is reported in the system status.

Figure 2-1. Initialization Timing

After assertion of the POWER signal, the DLPC6401 device application software asserts GPIO26 (within 5 ms) as the first step of software execution, indicating that the initialization process is in progress. DLPC6401 software continues to initialize the formatter hardware in incremental phases consisting of individual blocks, as discussed in the subsequent sections.

2.3 Internal Memory Test

Prior to transferring part of the code from parallel flash content to internal memory, the internal memory is initialized and a memory test is performed. The result of this test (pass or fail) is recorded in the system status. If the memory test fails, the initialization process is halted and GPIO26 (INIT_DONE) is asserted twice (as shown in Figure 2-1) to indicate an error situation. Memory test OK status bit is set to false in the system status command (x16h) which can be read using I²C command.

2.4 I²C Service Delay

The I²C service delay period begins on the transition of the hardware PWR_GOOD signal. I²C commands sent during the initialization in progress are not acknowledged nor processed. Commands sent during the initialization time are ignored. Only after de-asserting the GPIO26 (INIT_DONE) does the software respond to I²C commands.

2.5 Un-Park the Mirrors

After the formatter electronics are initialized (GPIO26 is de-asserted), the DMD mirrors are un-parked and the display of video is activated.

2.6 Power Down Sequence

The DC power supplies must be turned off, and PWR_GOOD must be set low, according to the timing in the DLPC6401 data sheet.

NOTE: Power Standby command must be sent before turning off the projector. See Section 3.1.7

2.7 Error Handling

- Transmitted bytes are less than expected for a command:
  - In this scenario, DLPC6401 times out while waiting for the remaining bytes. The current I^2C command is ignored.
- Transmitted bytes are more than expected for a command:
  - Only the valid number of bytes are accepted as part of the command remaining all bytes are ignored.
- Overflowing command queue (throttling):
  - In some rare scenarios, it is possible for the DLPC6401 command queue to overflow, if the incoming commands are faster than the rate at which the DLPC6401 device can service. In such a scenario, commands should only be sent when the status byte shows that the system is not busy.

2.7.1 Command Quick Reference

I^2C Address	Command Descriptions	Description
00h	Input source select	wr
02h	Pixel data format select	wr
03h	Port clock select	wr
04h	Input data channel swap	wr
05h	Flat panel display (FPD) mode and field select	wr
06h	Display curtain control	wr
07h	Power control	wr
08h	Long-axis image flip	wr
09h	Short-axis image flip	wr
0Ah	Test pattern (TPG) select	wr
0Bh	PWM invert	wr
0Ch	Splash screen select	wr
0Dh	Get current duty cycles	r
0Eh	Set current duty cycles	w
0Fh	Display frame rate	wr
10h	LED enable	wr
11h	Get version	r
12h	Debug message enable	wr
13h	Software reset	wr
14h	DMD park	wr
15h	Display buffer freeze	wr
16h	Hsync and Vsync polarity control	wr
17h	LVDS clock frequency select	wr
18h	Register command batch file	wr

I²C Address	Command Descriptions	Description
19h	LVDS polarity setting	wr
1Ah	Autolock enable	wr
20h	Hardware status	r
21h	System status	r
22h	Main status	r
23h	Brightness control	wr
24h	Contrast	wr
25h	Offset	wr
26h	Color space converter	wr
27h	CSC mailbox	wr
28h	Chroma interpolation	wr
29h	Chroma transient improvement	wr
2Ah	Color coordinate adjustment	wr
2Bh	Color coordinate adjustment mailbox	wr
2Ch	Color coordinate adjustment display color	wr
30h	Programming mode	wr
31h	Gamma correction	wr
32h	DLP® BrilliantColor™ feature control	wr
35h	DynamicBlack™ feature enable	wr
36h	DynamicBlack featureset aperture	wr
37h	DynamicBlack feature minimum	wr
38h	DynamicBlack feature maximum	wr
39h	DynamicBlack feature border configuration mailbox	w
3Ah	DynamicBlack feature LED off	wr
40h	PWM enable	wr
41h	PWM setup	wr
42h	Fan control	wr
43h	PWM capture configuration	wr
44h	GPIO interface	wr
45h	EEPROM get data interface	r
46h	EEPROM set data interface	w
47h	Update PWM frequency	wr
48h	Update LED usage time	wr
49h	Get LED usage time	r
4Ah	Reset LED usage time	w
4Bh	LED current control	wr
4Ch	Get LED sensor value	r
4Dh	Build LED sensor table	w
50h	Preset input and output resolution	wr
51h	Manual input and output horizontal resolution	wr
52h	Manual input and output vertical resolution	wr
53h	Crop first active line in frame	wr
54h	Crop last active line in frame	wr
55h	Crop first active pixel in frame	wr
56h	Crop last active pixel in frame	wr
57h	CCI control	wr
58h	LED max current	wr
59h	Set calibration data	wr

I²C Address	Command Descriptions	Description
5Ah	Current versus sensor table mailbox	wr
60h	Keystone command	wr
61h	Auto keystone control	wr
62h	3D frame synchronization	wr
63h	Temperature control	wr
64h	Read temperature	r
65h	RGB level check	wr
66h	Display position in frame	wr
68h	Display position user defined	wr
70h	Position OSD display	wr
71h	Slider bar display start OSD	wr
72h	Update slider progress OSD	wr
73h	Pull-down menu display start OSD	wr
74h	Pull-down navigation OSD	wr
75h	Display exit OSD	wr
76h	Information display start OSD	wr
77h	Get current displayed menu ID	r
78h	Set or get language ID	wr
79h	Set list index as dynamic field text	w

Command Description

3.1 General Commands

3.1.1 Input Source Select (I²C: 00h)

Byte	Bits	Description	Reset	Type
0	7:6	Reserved	×2	wr
	5:3	**Port Width** 0 – 30 bits 1 – 24 bits 2 – 20 bits 3 – 16 bits 4 – 10 bits 5 – 8 bits		
	2:0	**Selects Input Source** 0 – 30-bit Parallel Port 1 – Internal Test Pattern 2 – Splash Screen 3 – LVDS Port Source Other: Reserved		

3.1.2 Pixel Data Format Select (I²C: 02h)

Byte	Bits	Description						Reset	Type
0	7:4	Reserved						d1	wr
	3:0	Select the pixel format	**Pixel Formats Supported versus Source Type**						
			Parallel	**Test Pattern**	**Splash**	**LVDS**	**BT.656**		
		0 – RGB 4:4:4 (30-bit)	Y	Y	Y	Y	N		
		1 – YCrCb 4:4:4 (30-bit)	Y	N	N	N	N		
		2 – YCrCb 4:2:2	Y	N	Y	N	Y		

3.1.3 Port Clock Select (I²C: 03h)

Bytes	Bits	Description	Reset	Type
0	7:3	Reserved	d0	wr
	2:0	**Selects Port Input Clock**(1) 0: P1_CLKA: for parallel port 1: P1_CLKB: for parallel port 2: P1_CLKC: for parallel port		

(1) This command only selects clock for the parallel interface. For LVDS, the clock is automatically selected on Port-2 (LVDS).

3.1.4 Input Data Channel Swap (I²C: 04h)

This command configures the specified input data port along with mapping its data sub-channels to the corresponding ASIC input channels.

Byte	Bits	Description	Reset	Type
0	7	Port Number 0 – Port-1 (1) 1 – Port-2 (2)	x0	wr
	6:3	Reserved		
	2:0	Swap Data Channel(Default: 0) 0 – ABC = ABC = Straight through 1 – ABC = CAB = Rotate right 2 – ABC = BCA = Rotate left 3 – ABC = ACB = SWAP BC 4 – ABC = BAC = SWAP AB 5 – ABC = CBA = SWAP AC 6 – Invalid 7 – Invalid		

(1) Port-1 is a 30-bit parallel interface port.
(2) Port-2 is LVDS.

3.1.5 Flat Panel Display (FPD) Mode and Field Select (I²C: 05h)

Table 3-1. (1)(2)

Byte	Bits	Description	Reset	Type
0	7:6	**Pixel Mapping Mode** 0: Mode-1 1: Mode-2 2: Mode-3 3: Mode-4	x0	wr
	6:3	Reserved		
	2:0	**Field Signal Select** 0 : Map LVDS output from CONT1 onto field signal for ISD Port-2 1 : Map LVDS output from CONT2 onto field signal for ISD Port-2 2 : Force 0 onto field signal for ISD Port-2 (DEFAULT) 3 : Invalid		

(1) This command is used for LVDS source and is selected in the input source command.
(2) Pixel Mapping mode defines how LVDS output pixels are mapped into DLPC6401 ASIC Port-2. This table shows the mapping LVDS parallel data output buses RDA(6:0), RDB(6:0), RDC(6:0), RDD(6:0), and RDE(6:0) mapping into the 30-bit LVDS input port (Port2 of ASIC).

Table 3-2. Pixel Mapping Modes

	MODE1	MODE2	MODE3	MODE4
GRN[9]	RDB4	RDD3	RDE1	RDB4
GRN[8]	RDB3	RDD2	RDE2	RDB3
GRN[7]	RDB2	RDB4	RDD1	RDB2
GRN[6]	RDB1	RDB3	RDD2	RDB1
GRN[5]	RDB0	RDB2	RDB4	RDB0
GRN[4]	RDA6	RDB1	RDB3	RDA6
GRN[3]	RDD3	RDB0	RDB2	0
GRN[2]	RDD2	RDA6	RDB1	0
GRN[1]	RDE3	RDE3	RDB0	0
GRN[0]	RDE2	RDE2	RDA6	0
RED[9]	RDA5	RDD1	RDE1	RDA5
RED[8]	RDA4	RDD0	RDE0	RDA4

Table 3-2. Pixel Mapping Modes (continued)

	MODE1	MODE2	MODE3	MODE4
RED[7]	RDA3	RDA5	RDD1	RDA3
RED[6]	RDA2	RDA4	RDD0	RDA2
RED[5]	RDA1	RDA3	RDA5	RDA1
RED[4]	RDA0	RDA2	RDA4	RDA0
RED[3]	RDD1	RDA1	RDA3	0
RED[2]	RDD0	RDA0	RDA2	0
RED[1]	RDE1	RDE1	RDA1	0
RED[0]	RDE0	RDE0	RDA0	0
BLU[9]	RDC3	RDD5	RDE5	RDC3
BLU[8]	RDC2	RDD4	RDE4	RDC2
BLU[7]	RDC1	RDC3	RDD5	RDC1
BLU[6]	RDC0	RDC2	RDD4	RDC0
BLU[5]	RDB6	RDC1	RDC3	RDB6
BLU[4]	RDB5	RDC0	RDC2	RDB5
BLU[3]	RDD5	RDB6	RDC1	0
BLU[2]	RDD4	RDB5	RDC0	0
BLU[1]	RDE5	RDE5	RDB6	0
BLU[0]	RDE4	RDE4	RDB5	0
DATA_EN RDC6	RDC6	RDC6	RDC6	RDC6
VSYNC	RDC5	RDC5	RDC5	RDC5
HSYNC	RDC4	RDC4	RDC4	RDC4
CONT1	RDD6	RDD6	RDD6	RDD6
CONT2	RDE6	RDE6	RDE6	RDE6

3.1.6 Display Curtain Control (I²C: 06h)

This command enables solid color.

Byte	Bits	Description	Reset	Type
0	3:0	**Display Curtain Enable** 0 = Curtain disabled 1 = Curtain enabled	x0	wr
	7:4	**Display Curtain Color Select** (All undefined values are reserved) 0 = Black 1 = Red 2 = Green 3 = Yellow (Red + Green) 4 = Blue x7 = White 5 = Magenta (Red + Blue) 6 = Cyan (Green + Blue)		

3.1.7 Power Control (I²C: 07h)

Byte	Bits	Description	Reset	Type
0	7:2	Reserved	x0	wr
	1:0	This command controls the power modes 0 – Normal – The selected external source is displayed 1 – Standby – DMD is powered-down		

3.1.8 Long Axis Image Flip (I²C: 08h)

Byte	Bits	Description	Reset	Type
0	7:1	Reserved	x0	wr
	0:0	Flips the image along the long side of the DMD 0 – Disable flip 1 – Enable flip		

3.1.9 Short Axis Image Flip (I²C: 09h)

Byte	Bits	Description	Reset	Type
0	7:1	Reserved	x0	wr
	0:0	Flips the image along the short side of the DMD 0 – Disable flip 1 – Enable flip		

3.1.10 Test Pattern (TPG) Select (I²C: 0Ah)

Table 3-3. (1)(2)

Bits	Description	Reset	Type
7:4	Reserved	x8	wr
3:0	Internal test pattern select: 0000: Solid field 0001: Horizontal ramp 0010: Vertical ramp 0011: Horizontal lines 0100: Diagonal lines 0101: Vertical lines 0110: Grid 0111: Checkerboard 1000: ANSI checkerboard		

(1) TPG is always generated in RGB888 format.

(2) The resolution of the TPG should be configured by using the horizontal input resolution and vertical input resolution commands. Frame rate should be configured using frame rate commands.

3.1.11 PWM Invert (I²C: 0Bh)

Table 3-4. (1)

Bits	Description	Reset	Type
7:2	Reserved	d0	wr
1:0	0 – Pass through (default) 1 – Invert PWM values		

(1) This command compensates for some LED design problem when PWM and current are swapped. PWM 0 corresponds to max current and PWM 255 corresponds to 0 current. The default behavior “pass through” means PWM and current are in the correct order, PWM 0 corresponds to current 0. This command needs to be sent before powering up the system, or before setting LED maximum DC or LED current controls.

3.1.12 Splash Screen Select (I²C: 0Ch)

Table 3-5. (1)

Bits	Description	Reset	Type
7:5	Reserved	d0	wr
4:0	Splash screen image number stored in the flash memory Valid splash screen index as 0 to 0xF		

(1) If the source is already set to SPLASH, selecting the SPLASH number immediately displays the selected SPLASH image.

3.1.13 Get Current Duty Cycles (I²C: 0Dh)

Bytes	Bits	Description	Reset	Type
0	7:0	Red duty cycle (MSB)(1)		r
1	7:0	Red duty cycle (LSB)(1)		
2	7:0	Green duty cycle (MSB)(1)		
3	7:0	Green duty cycle (LSB)(1)		
4	7:0	Blue duty cycle (MSB)(1)		
5	7:0	Blue duty cycle (LSB)(1)		
6	7:0	Cyan duty cycle (MSB)(1)		
7	7:0	Cyan duty cycle (LSB)(1)		
8	7:0	Magenta duty cycle (MSB)(1)		
9	7:0	Magenta duty cycle (LSB)(1)		
10	7:0	Yellow duty cycle (MSB)(1)		
11	7:0	Yellow duty cycle (LSB)(1)		
12	7:0	White duty cycle (MSB)(1)		
13	7:0	White duty cycle (LSB)(1)		
14	7:0	Sequence group		
15	7:0	Sequence data group		
16	7:0	Sequence index (MSB)		
17	7:0	Sequence index (LSB)		

(1) Each duty cycle data is a 16-bit number in an unsigned 8.8 format

3.1.14 Set Current Duty Cycles (I²C: 0Eh)

This command updates a particular sequence on existing duty cycles within an existing sequence data group in the sequence group. We can store a maximum of 64 sequences in the flash.

Sequence index range is 0 to 0xffff.

Bytes	Bits	Description	Reset	Type
0	7:0	Sequence group		w
1	7:0	Sequence data group		
2	7:0	Sequence index (MSB)		
3	7:0	Sequence index (LSB)		

3.1.15 Display Frame Rate (I²C: 0Fh)

Table 3-6. (1)(2)(3)

Bytes	Bits	Description	Reset	Type
0	7:0	Display frame rate (15:8)	x0	wr
1	7:0	Display frame rate (7:0)		

(1) Frame Rate: Range (45 to 120 Hz in increments of 0.5 Hz)
(2) Input format is a 16-bit unsigned integer where Bit 0 represents the fractional bit. (i.e. if Bit 0 is set then the value is x.5 Hz)
(3) Examples: 120 Hz → 120 / 0.5 = 240 and 119.5 → 119.5 / 0.5 = 239

3.1.16 LED Enable (I²C: 10h)

Bits	Description	Reset	Type
0	Enable red LED: 1 – Disable LED 0 – Enable LED	b0	wr
1	Enable green LED: 1 – Disable red LED 0 – Enable red LED		
2	Enable blue LED: 1 – Disable red LED 0 – Enable red LED		
7:3	Reserved		

3.1.17 Get Version (I²C: 11h)

Byte	Bits	Description	Reset	Type
0	7:0	ARM application major software version	x0	r
1	7:0	ARM application minor software version		
2	7:0	ARM application sub-minor (patch) software revision (MSB)		
3	7:0	ARM application sub-minor (patch) software revision (LSB)		
4	7:0	ARM API major software version		
5	7:0	ARM API minor software version		
6	7:0	ARM API sub-minor (patch) software revision (MSB)		
7	7:0	ARM API sub-minor (patch) software revision (LSB)		
8	7:0	DMD revision		
9	7:0	DAD revision		
10	7:0	ASIC revision		

3.1.18 Debug Message Enable (I²C: 12h)

There are byte 0 and byte 1 masks to print various debug messages. To enable all debug messages, set two bytes to 0xFFFF. To disable all debug messages, set two bytes to 0. The default is 0 with no debug message output.

Byte	Bits	Description	Reset	Type
0	7:0	0xFF – Enable debug messages 0x00 – Disable debug messages	x0	wr
1	7:0	0xFF – Enable debug messages 0x00 – Disable debug messages		

3.1.19 Software Reset (I²C: 13h)

When the projector receives the software reset command, the data defines if a software reset occurs for the DLPC6401. When written with any value a software reset occurs. This command has no specific intended application. It is provided simply as a back-up recovery mechanism.

Byte	Bits	Description	Reset	Type
0	0	Software reset: 0 – Software reset is recognized (bit value is don't care) 1 – Software reset is recognized (bit value is don't care)	x0	wr

3.1.20 DMD Park (I²C: 14h)

The DMD PARK command provides a software programmable alternative to using the PWRGOOD signal. If PWRGOOD is tied HIGH, the DMD PARK command can warn the DLPC6401 of an impending DMD power loss, such that it can PARK the DMD mirrors and avoid reliability degradation.

Table 3-7.

Byte	Bits	Description	Reset	Type
0	0	DMD Park Control 0 = Un-park the DMD (default) 1 = Park the DMD	x0	wr

3.1.21 Display Buffer Freeze (I²C: 15h)

Table 3-8.

Byte	Bits	Description	Reset	Type
0	0	Display buffer swap freeze (also known as memory buffer swap disable) 0 = Enable buffer swapping (default) 1 = Disable buffer swapping (freeze) TI recommends to use freeze when changing source or operating modes to block temporary corruption caused by reconfiguration from reaching the display. When frozen, the last display image continues to be displayed.	x0	wr

3.1.22 HSync and VSync Polarity Control (I²C: 16h)

Byte	Bits	Description	Reset	Type
0	7:6	Reserved	x0	wr
	5:3	**VSync Polarity:** 0 – No Invert (default) 1 – Invert		
	2:0	**HSync Polarity:** 0 – No Invert (default) 1 – Invert		

3.1.23 LVDS Clock Frequency Range Select (I²C: 17h)

Bits	Description	Reset	Type
7:2	Reserved	d0	wr
1:0	0 - Source LVDS clock frequency range 25 to 65 MHz 1 - Source LVDS clock frequency range 65 to 90 MHz		

3.1.24 Register Command Batch File (I²C: 18h)

The register command batch file is a text file which only allows write commands and doesn't support a nested batch file. All register batch files are converted to binary and stored in flash. The syntax of the command batch file is:

W – Start with write

Device write address – 0x34

Cmd – Write command itself

Cmd Length – Number of bytes of data, not including write command and device address byte

Cmd Byte Parameter or Parameters

For example, this command is to write cmd 06h curtain control to display green color:

Write curtain control command 0x86 (MSB needs to be set for write command). The length of this command is 1 byte. The byte for 0x21 is to display green curtain.

w 0x34 0x86 0x1 0x21

Table 3-9. (1)

Bits	Description	Reset	Type
7:3	Reserved	d0	wr
2:0	**Register Command Batch File Selection Values** 0: Register batch file 0 (Auto Init Command Batch File) 1: Register batch file 1 2: Register batch file 2 3: Register batch file 3 4: Register batch file 4 5: Register batch file 5 6: Register batch file 6 7: Register batch file 7		

(1) The purpose of this command is to overwrite some system power-up settings or command defaults. Register batch file 0 is a special batch file which the system automatically executes when powered-up. Other register batch files can be executed via this I^2C command.

3.1.25 LVDS Polarity Setting (I^2C: 19h)

Byte	Bits	Description	Reset	Type
0	0	0 – No swap polarity (default) 1 – Swap polarity	x0	wr

3.1.26 Autolock Enable (I^2C: 1Ah)

This command allows the user to enable or disable the autolock. By default, autolock is disabled once the source is locked so that glitch does not affect the sync monitor.

Byte	Bits	Description	Reset	Type
0	0	0 – Enable autolock 1 – Disable (default) shut down autolock once the source is locked	x1	wr

3.2 Status Commands

3.2.1 Hardware Status (I^2C: 20h)

Byte	Bits	Description	Reset	Type
0	0	Internal initialization complete	x0	r
	1	Detection of illegal memory access		
	2	DMD reset waveform controller error		
	3	Forced swap error		
	4	FRB overflow		
	5	FRB underflow		
	6	**Sequencer Abort Status Flag** 0 = No error 1 = Sequencer detected an error condition that caused an abort		
	7	**Sequencer Error** 0 = No error 1 = Indicates that the sequencer detected an error		

3.2.2 System Status (I²C: 21h)

Byte	Bits	Description	Reset	Type
0	0	**Internal Memory Test OK** 0: false 1: true	x0	r
	1	**Mailbox Download Complete** 0: Incomplete – incorrect number of mailbox words received 1: Complete – expected number of mailbox words received		
	2	**Flash Memory Write Unlock** 0: Flash memory is write locked 1: Flash memory is write unlocked		
	3	**System Good Status** 0: Something is wrong 1: All hardware is functional		
	4	**User GPIO Status** (For instance fan status) 0: HW functional 1: HW failure		
	5	**Overtemperature Warning** 0: No warning 1: Warning		

3.2.3 Main Status (I²C: 22h)

Byte	Bits	Description	Reset	Type
0	0	**DMD Park Status** 0 = DMD mirrors are not parked 1 = DMD mirrors are parked (in a flat state)	x0	r
	1	**Sequence Run Flag** 0 = Sequencer is stopped 1 = Sequencer is running normally		
	2	**Frame Buffer Swap Flag** 0 – Not freeze 1 – Freeze		
	3	**Gamma Correction Function Enable** 0: Disable 1: Enable		
	4	**Extended Primary Color Correction Function Enable** 0: Disable 1: Enable		
	7:5	Reserved		

3.3 Image Control Commands

3.3.1 Brightness Control (I²C: 23h)

Table 3-10. (1)

Byte	Bits	Description	Reset	Type
0	2:0	Channel A MSB bit(10:8)	x0	wr
1	7:0	Channel A LSB bit(7:0)		
2	2:0	Channel B MSB bit(10:8)		
3	7:0	Channel B LSB bit(7:0)		
4	2:0	Channel C MSB bit(10:8)		
5	7:0	Channel C LSB bit(7:0)		

(1) This command adjusts the brightness of the source image. The adjustment is applied in the datapath after source offset, contrast gain, and RGB gain, but before hue, color, color space conversion, and RGB offset. The brightness is the signed 2's complement value with 2 fractional bits and a range of –256.00 to 255.75, inclusive. For YUV inputs, the brightness is applied to the luma channel only. For RGB inputs, the brightness is equally applied to all channels.

3.3.2 Contrast (I²C: 24h)

Table 3-11. (1)

Byte	Bits	Description	Reset	Type
0	7:0	RGB percentage from 0% to 200%, with 100% as nominal	x0	wr

(1) This command adjusts the contrast or white level of the source image. Contrast is specified as a percentage from 0% to 200% with 100% being nominal (no contrast change) and 0% being a black image. This command adjusts contrast by altering the Color Space Converter (CSC) coefficients. For video (YCrCb), this gain is applied to Y data only. For graphics (RGB), this is applied to all three channels.

3.3.3 Offset (I²C: 25h)

Table 3-12. (1)

Byte	Bits	Description	Reset	Type
0	2:0	Red MSB bit (10:8)	x0	wr
1	7:0	Red LSB bit (7:0)		
2	2:0	Green MSB bit (10:8)		
3	7:0	Green LSB bit (7:0)		
4	2:0	Blue MSB bit (10:8)		
5	7:0	Blue LSB bit (7:0)		

(1) This command offsets the levels of the RGB channels at a point in the datapath that is after source offset, contrast, brightness, and color space conversion. RGB channel offset adjustment range: –256 to 255.75 inclusive (signed 2's complement value with 2 fractional bits). This command is usually applicable to RGB sources. For YUV sources, results of this command might vary depending on the type of YUV source (4:2:2 or 4:4:4).

3.3.4 Color Space Converter (I²C: 26h)

A color space converter (CSC) is used to specify the color matrix that should be used to translate input data to RGB data or to color correct RGB input data.

The CSC accepts only 4:4:4 sample data format. The design contains one color space matrix with nine elements. As the interface shows, there are four matrixes to choose from.

Byte	Bits	Description	Reset	Type
0	7	1 – CSC enable 0 – CSC disable	x0	wr
	6:4	Reserved		
	3	**CSC Table Type** 0 – Predefined CSC Table in the Flash 1 – User defined CSC Table (uses CSC Mailbox command 27h)		
	2:0	**Predefined CSC Table Index** 000: CSC 0 = Full-range RGB 001: CSC 1 = ITU-BT.601 offset YUV (video decoder sources) 010: CSC 2 = Full-range YUV 2 (ADC sources) 011: CSC 3 = Offset RGB 100: CSC 4 = ITU-BT.601 offset YUV 101: CSC 5 = Full-range YUV 110: CSC 6 = ITU-BT.709 offset YUV 111: CSC 7 = SMTPE 240M		

3.3.5 CSC Mailbox (I^2C: 27h)

Table 3-13. (1)

Byte	Bits	Description	Reset	Type
0	12:8	CSC MSB	x0	wr
	15:13	Reserved		
1	7:0	CSC LSB		

(1) All 9 words (18 bytes) must be sent as one contiguous block to ensure the values are updated. Sending tables with bad data values will likely result in unacceptable screen images.

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} C1 & C2 & C3 \\ C4 & C5 & C6 \\ C7 & C8 & C9 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \end{bmatrix}$$

Figure 3-1.

All programmable CSC coefficient values represent numbers less than 4, but greater than or equal to –4. They are 13-bit signed 2's complement numbers with the binary point between bits 9 and 10.

Figure 3-2. Coefficient MSBYTE Bit Values

B12	B11	B10	B9	B8
SIGN	2^{1}	2^{0}	2^{-1}	2^{-2}

Figure 3-3. Coefficient LSBYTE Bit Values

B7	B6	B5	B4	B3	B2	B1	B0
2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}	2^{-9}	2^{-10}

Table 3-14. CSC Mailbox Sub-Address Map(1)

WORD	Coefficient
0	C1
1	C2
2	C3
3	C4

(1) Writing to the CSC mailbox register overwrites the CSC values established by the color space converter command. The contrast command data is applied to any data downloaded through this interface.

Table 3-14. CSC Mailbox Sub-Address Map[1] (continued)

WORD	Coefficient
4	C5
5	C6
6	C7
7	C8
8	C9

3.3.6 Chroma Interpolation (I^2C: 28h)

Chroma interpolation provides two methods of 4:2:2 to 4:4:4 conversions. The primary interpolates between sub-sampled values to produce the missing values. The secondary method is a chroma copy method that simply uses the chroma signal for 2 consecutive pixels. The chroma interpolation function can be disabled to support 4:4:4 data pass-thru for non-sub sampled (4:4:4) video and RGB applications.

Bytes	Bits	Description	Reset	Type
0	7	**Chroma Interpolation Enable** 0 - Chroma interpolation is disabled 1 - Chroma interpolation is enabled	x0	wr
	6:1	Reserved		
	0	**Chroma Interpolation Mode** 0 - Interpolation 1 - Chroma copy		

3.3.7 Chroma Transient Improvement (I^2C: 29h)

Table 3-15.

Byte	Bits	Description	Reset	Type
0	7	**Chroma Transient Improvement (CTI) Enable** 0: CTI function is disabled 1: CTI function is enabled	x0	wr
	6:4	Reserved		
	3:2	**Gain Select** 00:1 01:2 10:4 11:8		
	1	**Filter Type** 0: For low-bandwidth source 1: For high-bandwidth source1 – chroma copy		
	0	**Horizontal Band Pass Filter Frequency** 0: For low-bandwidth source 1: For high-bandwidth source		

3.4 Color Coordinate Adjustment (CCA)

3.4.1 Color Coordinate Adjustment (I²C: 2Ah)

This register, in conjunction with color coordinate adjustment mailbox, define how the projected image appears. See Section 3.5.2 for a description of this function.

Table 3-16.

Byte	Bits	Description	Reset	Type
0	0	**CCA Enable or Disable** 0: Disabled 1: Enabled	x0	wr
	7:1	Reserved		

3.4.2 Color Coordinate Adjustment Mailbox (I²C: 2Bh)

Byte	Bits	Description	Reset	Type
0	7:0	Coordinates to describe color correction (MSB) CCA (15:8)	x0	wr
1	7:0	Coordinates to describe color correction (MSB) CCA (7:0)		

CCA is a software that interfaces to the P7 color correction matrix, or PCC. CCA controls the gains for all colors (R, G, B, Y, C, and M), to account for secondary color changes impacted by providing gains only to R, G, and B (white point changes).

The data are 16-bit numbers in unsigned 1.15 format. The actual value is 1/32768 of the value written to this mailbox.

Figure 3-4. Coefficient MSBYTE Bit Values

B15	B14	B13	B12	B11	B10	B9	B8
2^{0}	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}

Figure 3-5. Coefficient LSBYTE Bit Values

B7	B6	B5	B4	B3	B2	B1	B0
2^{-8}	2^{-9}	2^{-10}	2^{-11}	2^{-12}	2^{-13}	2^{-14}	2^{-15}

Table 3-17. CCA Mailbox Sub-Address Map

Word	Description	Range
0	Measured x-coordinate for red	0 – 0.9999
1	Measured y-coordinate for red	0 – 0.9999
2	Measured luminance for red	0 – 1.0
3	Measured x-coordinate for green	0 – 0.9999
4	Measured y-coordinate for green	0 – 0.9999
5	Measured luminance for green	0 – 1.0
6	Measured x-coordinate for blue	0 – 0.9999
7	Measured y-coordinate for blue	0 – 0.9999
8	Measured luminance for blue	0 – 1.0
9	Measured x-coordinate for white RGB	0 – 0.9999
10	Measured y-coordinate for white RGB	0 – 0.9999
11	Measured luminance for white RGB(1)	0 – 1.0
12	Desired x-coordinate for red	0 – 0.9999
13	Desired y-coordinate for red	0 – 0.9999
14	Desired gain for red	0 – 1.0
15	Desired x-coordinate for green	0 – 0.9999
16	Desired y-coordinate for green	0 – 0.9999
17	Desired gain for green	0 – 1.0
18	Desired x-coordinate for blue	0 – 0.9999
19	Desired y-coordinate for blue	0 – 0.9999
20	Desired gain for blue	0 – 1.0
21	Desired x-coordinate for cyan	0 – 0.9999
22	Desired y-coordinate for cyan	0 – 0.9999
23	Desired gain for cyan	0 – 1.0
24	Desired x-coordinate for magenta	0 – 0.9999
25	Desired y-coordinate for magenta	0 – 0.9999
26	Desired gain for magenta	0 – 1.0
27	Desired x-coordinate for yellow	0 – 0.9999
28	Desired y-coordinate for yellow	0 – 0.9999
29	Desired gain for yellow	0 – 1.0
30	Desired x-coordinate for white	0 – 0.9999
31	Desired y-coordinate for white	0 – 0.9999
32	Desired gain for white	0 – 1.0

(1) The default is 1.

3.4.3 Color Coordinate Adjustment Display Color (I²C: 2Ch)

This command is used to display each color listed previously to measure the color points of the projector with a colorimeter.

Table 3-18.

Byte	Bits	Description	Reset	Type
0	3:0	**Display Pleasing Color** 1: Red 2: Green 3: Yellow 4: Blue 5: Magenta 6: Cyan 7: White	x1	wr
	6:4	Reserved		
	7:7	1: Enable pleasing color 0: Disable pleasing color		

3.5 Image Formatting Commands

The Digital Micro-Mirror Device is inherently linear in response. Degamma processing is used to remove the gamma curve, which was applied to video and graphics data at the source. The Degamma function consists of three 1024 entries in a 12-bit table.

Table 3-19. Gamma Correction (I2C: 31h)

Byte	Bits	Description	Reset	Type
0	5:0	Reserved	x0	wr
	7:6	**Gamma Enable or Disable** 00 – Disable 01 – Enable		
1	3:0	**Degamma Selection** 0: TI enhanced 1: Max brightness 2 to 15: OEM-defined tables		
	7:4	Reserved		

3.5.1 Degamma File Structure

Figure 3-6. Coefficient MSBYTE Bit Values

B15	B14	B13	B12	B11	B10	B9	B8
0	0	0	0	Ex[4]	Ex[3]	Ex[2]	Ex[1]

Figure 3-7. Coefficient LSBYTE Bit Values

B7	B6	B5	B4	B3	B2	B1	B0
Ex[0]	$Man[0]2^{-1}$	$Man[0]2^{-2}$	$Man[0]2^{-3}$	$Man[0]2^{-4}$	$Man[0]2^{-5}$	$Man[0]2^{-6}$	$Man[0]2^{-7}$

↑The binary point is between B7 and B6.

Ex[4:0] – Exponent value for gamma table entry

Man[6:0] – Mantissa value for gamma table entry

Gamma tables values use a 12-bit floating point format as follows:

$$Value = [1 + Mantissa\ ] \times 2\ (Exponent - 20) \quad (1)$$

Gamma tables are downloaded in low-code to high-code order and have 1024 entries each for red, green, and blue for a total download of 3072 12-bit floating point numbers, using 2 bytes each.

Table download order should be:

Table 3-20.

WORD	MSB	LSB
0	Red[0] (MSB)	Red[0] (LSB)
1	Red[1] (MSB)	Red[1] (LSB)
2	..	..
	..	..
	..	..
	Red[1023] (MSB)	Red[1023] (LSB)
	Green[0] (MSB)	Green[0] (LSB)
	…	…
	…	…
	Green[1023] (MSB)	Green[1023] (LSB)
	Blue[0] (MSB)	Blue[0] (LSB)
	…	…
	…	…
3071	Blue[1023] (MSB)	Blue[1023] (LSB)

When the last gamma entry has been received, the gamma table is loaded and takes effect. Gamma entries may be downloaded while displaying data.

3.5.2 BrilliantColor™ Technology Control (I^2C: 32h)

The BrilliantColor™ feature mode bits select between predefined display modes. If the BrilliantColor™ feature is disabled, there is no overlapping color and only R, G, and B duty cycles are displayed.

Table 3-21.

Byte	Bits	Description	Reset	Type
0	5:0	**BrilliantColor™ feature Mode Selection** 0 – BrilliantColor Look 1 1 – BrilliantColor Look 2 2 – BrilliantColor Look 3 3 – BrilliantColor Look 4 … 63 – BrilliantColor Look 64	x0	wr
	6	Reserved		
	7	**Enable BrilliantColor™ feature Processing** 0 – Disable 1 – Enable		

3.6 DynamicBlack Technology

The purpose of the DynamicBlack feature is to increase contrast and reduce dithering noise in darker scenes by gaining the signal and reducing the current level based on scene content. The DynamicBlack feature works only when the white point correction (WPC) algorithm is enabled.

3.6.1 DynamicBlack Feature Enable (I^2C: 35h)

The border pixel weight value sets the weighting of pixels within the border region relative to the interior of the image. The border region is setup via DynamicBlack border configuration mailbox command. When the DynamicBlack feature is enabled, the DynamicBlack LED off disable functionality controls whether the LED driver levels are set to 0, turning off the LED light output, for very-dark scenes. If the DynamicBlack LED off function is disabled, typical DynamicBlack functionality continues to operate.

The strength of DynamicBlack technology is the aggressiveness of the DynamicBlack algorithm. A higher strength value causes the DynamicBlack feature to be more aggressive in closing the aperture for a given scene brightness.

Table 3-22.

Bytes	Bits	Description	Reset	Type
0	7	**DynamicBlack Enable**[(1)] 0: Disable DynamicBlack processing 1: Enable DynamicBlack processing	x0	wr
	6	**DynamicBlack Border Control Enable**[(1)] 0: Disable DynamicBlack Border Exclusion 1: Enable DynamicBlack Border Exclusion		
	5:4	**DynamicBlack Border Pixel Weight**[(1)] 00: 0% 01: 25% 10: 50% 11: 75%		
	3:0	Reserved		
1	7:0	**DynamicBlack Strength** Valid Range: 0 to 3		

(1) This command is only valid if DynamicBlack processing is enabled in the flash image.

3.6.2 DynamicBlack Set Aperture (I^2C: 36h)

This command is used primarily to calibrate the DynamicBlack feature for a projection system. To use this command, the DynamicBlack feature must be enabled and then disabled. This command controls how far the aperture opens or closes over the light source.

Table 3-23.

Bytes	Bits	Description	Reset	Type
0	7:0	Reserved	x0	wr
1	7:0	**DynamicBlack Aperture Position** Valid Range: 0 to 254		

3.6.3 DynamicBlack Min (I²C: 37h)

Table 3-24.

Bytes	Bits	Description	Reset	Type
0	7:0	DynamicBlack min aperture setting (MSB) (15:8)(1)	x0	wr
1	7:0	DynamicBlack min aperture setting (LSB) (7:0)(1)		

(1) The minimum command value typically corresponds to the fully closed aperture position. The range is between 0 to 254.

3.6.4 DynamicBlack Max (I²C: 38h)

Bytes	Bits	Description	Reset	Type
0	7:0	DynamicBlack max aperture setting (MSB) (15:8)(1)	x0	wr
1	7:0	DynamicBlack max aperture setting (LSB) (7:0)(1)		

(1) The maximum command typically corresponds to the fully opened aperture position. The range is between 0 to 254.

3.6.5 DynamicBlack Border Configuration Mailbox (I²C: 39h)

DynamicBlack border configuration mailbox command consists of four words as detailed in Section 3.6.5.1 representing left pixel, right pixel, top line, and bottom line, in the same order.

Each word is 16-bits. The Table 3-25 shows how the individual word is represented. It can be noted that byte 0 represents MSB and byte 1 represents LSB in the word.

Table 3-25.

Bytes	Bits	Description	Reset	Type
0	7:0	MSB (15:8)	x0	w
1	7:0	LSB (7:0)		

3.6.5.1 DynamicBlack Border Configuration Setup Mailbox Sub-Address Map

This command is write only.

This command is used to configure the DynamicBlack border exclusion region. A value greater than or equal to the number of display pixels means no right edge border. A value greater than or equal to the number of display lines means no bottom border.

All four words (8 bytes) must be sent as one contiguous block to ensure the values are updated. Sending tables with bad data values will likely result in unacceptable screen images.

Table 3-26.

Word	Description	Range	Default
0	Left pixel (left edge of border region)	0 – 0x7ff	0 (no left edge border)
1	Right pixel (right edge of border region)	0 – 0x7ff	0x7ff (no right edge)
2	Top line (top line of border region)	0 – 0x7ff	0 (no top border)
3	Bottom line (bottom line of border region)	0 – 0x7ff	0x7ff (no bottom border)

3.6.6 DynamicBlack LED Off (I²C: 3Ah)

Table 3-27. (1)

Byte	Bits	Description	Reset	Type
0	5:0	DynamicBlack ratio limit (0.5 fixed point format)	x0	wr

(1) DynamicBlack ratio limit defines the ratio below which the algorithm switches off the LEDs. Using this command, the user can set the value of the limit itself. It can be set to 0, if LEDs are turned off.

3.7 Peripheral Interface

3.7.1 PWM Enable (I²C: 40h)

Table 3-28. (1)(2)

Byte	Bits	Description	Reset	Type
0	7	**Enable PWM Channel** 0: Disable 1: Enable	x0	wr
	6:3	Reserved		
	2:0	**PWM Channel Select** 000: PWM channel 0 001: PWM channel 1 010: PWM channel 2 011: PWM channel 3 100: PWM channel 4 101: PWM channel 5		

(1) Five programmable general purpose pulse-width-modulated (PWM) signals are provided. OEM can use these signals for a variety of control applications. PWM channels can also be used as GPIO pins. Enabling these as PWM outputs overrides any GPIO settings for respective pins.

(2) Before enabling the PWM signals using this command, pulse-width modulation must be set up using this command.

3.7.2 PWM Setup (I²C: 41h)

The PWM outputs of the ASIC can be used to control, for example, the speed of a fan when used with supporting circuitry. By varying the duty cycle, the supply voltage to the fan is varied, and thus the speed can be reduced.

An example circuit for controlling the fan speed can be seen in the reference design schematic.

A 18.67-MHz quantizing clock defines the PWM frequency and duty cycle. To calculate the PWM period to be set from the frequency by dividing 18666667 by the frequency value. So for 2 KHz, the period is 18666667 / 2000 = 9333 or 0x2475, so BYTE2 is 0x24 while BYTE3 is 0x75. The period can be set in increments of 53.5 ns.

API converts it back to frequency by adding 1 to period and then dividing 18666667 by (period + 1). Here integer division truncates the value, hence the return value is slightly different.

The PWM control function provides five PWM signals to devices outside the component set. The provided setup allows the parameters of these PWM signals to change.

The clock period of the output wave form in base clock resolution (53.57 ns) and the range is 1 to 0xfffff.

Table 3-29.

Bytes	Bits	Description	Reset	Type
0	7:5	**PWM Channel Select**[1][2] 000: PWM channel 0 001: PWM channel 1 010: PWM channel 2 011: PWM channel 3 100: PWM channel 4 101: PWM channel 5	x0	wr
	2:0	Reserved		
1	7:0	Bits(23:16) – Clock period in increments of 53.57 ns		
2	7:0	Bits(15:8) – Clock period in increments of 53.57 ns		
3	7:0	Bits(7:0) – Clock period in increments of 53.57 ns		
4	7:7	Reserved		
	6:0	Duty cycle (range = (1% to 99%))		

(1) If an invalid duty cycle is selected, the next-closest-possible duty cycle to the invalid duty cycle value is used.
(2) PWM channels can also be used as GPIO pins. Enabling these as PWM outputs overrides any GPIO settings for respective pins.

3.7.3 Fan Control (I^2C: 42h)

Table 3-30.

Bytes	Bits	Description	Reset	Type
0	7	**Enable Fan Control** 0 – Disable 1 – Enable	x0	wr
	6:4	Reserved		
	3	**Frequency** 0 to 100 KHz 1 to 24 KHz		
	2:0	**Fan Number** 0 = Fan 1 1 = Fan 2 2 = Fan 3		
1	7	Reserved		
	6:0	**Duty Cycle** Valid range: 0% to 100% 0% disables the fan		

3.7.4 PWM Capture Configuration (I^2C: 43h)

Table 3-31. (1)

Bytes	Bits	Description	Reset	Type
0	7	Reserved	x0	wr
	6:4	**Sample Rate in Hz** 000 – 1000 001 – 10000 010 – 100000 011 – 1000000 100 – 2000000 101 – 4000000 110 – 8000000 111 – 16000000		
	3:1	Reserved		
	0	**PWM Capture Port** 0 – Port-1 = PWM_IN_0 = GPIO5 1 – Port-2 = PWM_IN1 = GPIO6 2 – Port 3 = PWM_IN_2 = GPIO7		

(1) When reading PWM period, read 4 bytes, the definition of returned bytes.

Table 3-32.

Bytes	Bits	Description	Reset	Type
1	7:0	High period (15:8)	x0	wr
2	7:0	High period (7:0)		
3	7:0	Low period (15:8)		
4	7:0	Low period (7:0)		

3.7.5 GPIO Interface (I^2C: 44h)

Bytes	Bits	Description	Reset	Type
1	7	**Alternative Function Enable** 0: Disable Alternative function and use it as a general purpose GPIO 1: Configure the alternative mode. (When this mode is used select the alternative mode)	x0	wr
	6	**Alternative Function Mode** (Needed only when the alternative function enable (Bit7) is set to 1) 0: Use alternative function mode = 0 1: Use alternative function mode = 1		
	5	**Direction** 0: Input 1: Output		
	4	**Output Type** 0: Normal 1: Open drain		
	3	**Output state** (Default: 0) 0: Pin low 1: Pin high		
	2:0	Reserved		
0	7:6	Reserved	x0	wr
	5:0	**Select GPIO Pin** Table 3-33		

Table 3-33 shows the GPIO select value and their alternative mode function.

Table 3-33.

Select GPIO Pin	GPIO Name	Alternative Mode 1	Alternative Mode 2
000000	GPIO0	PWM_STD_0 (output)	Not supported
000001	GPIO1	PWM_STD_1 (output)	Not supported
000010	GPIO2	PWM_STD_2 (output)	Not supported
000011	GPIO3	Reserved (used by USB task)	Reserved
000100	GPIO4	PWM_STD_4 (output)	LED1_GRN
000101	GPIO5	PWM_IN_0 (input)	Not supported
000110	GPIO6	PWM_IN_1 (input)	Not supported
000111	GPIO7	PWM_IN_2 (input)	Not supported
001000	GPIO8(1)	Reserved	Reserved
001001	GPIO9(1)	Reserved	Reserved
001010	GPIO10	GPCLK2 (Output)	PROG_AUX_0 (output)
001011	GPIO11	GPCLK3 (Output)	
001100	GPIO12	GPCLK3 (Output)	LED sense frequency (input)
001101	GPIO13(2)	Reserved	Reserved
001110	GPIO14	Not supported	Not supported
001111	GPIO15	IR0 (input)	IR0
010000	GPIO16	PM_ADDR_21 (output)	PM_ADDR_22
010001	GPIO17	PM_ADDR_22 (output)	PM_ADDR_22
010010	GPIO18	Not supported	USB_ENZ
010011	GPIO19	Not supported	HOLD_BOOT_LOADER
010100	GPIO20	Option 2 = CSP_CLK (output)	ALF_COAST (output)
010101	GPIO21	Option 2 = CSP_DATA (output)	ALF_CLAMP (output)
010110	GPIO22	Not supported	Not supported
010111	GPIO23	Not supported	Not supported
011000	GPIO24	Not supported	Not supported
011001	GPIO25	Not supported	Not supported
011010	GPIO26(3)	Reserved	Reserved
011011	GPIO27	Not supported	Not supported
011100	GPIO28	Not supported	Not supported
011101	GPIO29	Not supported	Not supported
011110	GPIO30	Not supported	Not supported
011111	GPIO31	Option 2 = CW_INDEX (input)	LED2_RED
100000	GPIO 32	CW_PWM (output)	LED2_GRN
100001	GPIO 33	UART1_RTSZ (output)	Not Supported
100010	GPIO 34	UART1_CTSZ (input)	Not Supported
100011	GPIO 35	UART1_TXD (output)	Not Supported
100100	GPIO 36	UART1_RXD (input)	Not Supported
100101	GPIO 37	PMD_INTZ (input)	PROG_AUX 2 (output)

(1) GPIO8 and GPIO9 are used for I^2C only. GPIO8 is for SCL and GPIO9 is for SDA.
(2) GPIO13 is used for LED sensor only.
(3) GPIO26 is used to INIT_DONE status. This GPIO cannot be used as a general purpose GPIO (during power up).

3.7.6 EEPROM Get Data Interface (I²C: 45h)

With EEPROM access, the first 1056 bytes are reserved for internal data access and the next 512 bytes for auto init register batch file. The user can access all EEPROM space, except the reserved address space. For example, if the user wants to read the first 16 user-defined bytes, the user can set offset to 1567 (zero-based offset) and the number of bytes to read 16.

Offset 1567 to the end of EEPROM is accessible space to the user.

Assume the minimum size of EEPROM is at least 2K.

Bytes	Bits	Description	Reset	Type
0	7:0	Offset to the first available address of EEPROM (MSB)		w
1	7:0	Offset to the first available address of EEPROM (LSB)		
2	7:0	Number of bytes to read (MSB)		
3	7:0	Number of bytes to read (LSB)		
4 to number of bytes read + 4	7:0	Number of bytes intended to read		wr

3.7.7 EEPROM Set Data Interface (I²C: 46h)

With EEPROM access, the first 1056 bytes are reserved for internal data access and the next 512 bytes for auto init register batch file access. The user can access all EEPROM space, except the reserved address space. For example, if the user wants to write the first 32 user-defined bytes, the user can set offset to 1567 (zero-based offset) and specify the number of bytes to write 32.

Offset 1567 to the end of EEPROM is accessible space to the user.

Assume minimum size of EEPROM is at least 2K.

Bytes	Bits	Description	Reset	Type
0	7:0	Offset to the first available address of EEPROM (MSB)		w
1	7:0	Offset to the first available address of EEPROM (LSB)		
2	7:0	Number of bytes to write (MSB)		
3	7:0	Number of bytes to write (LSB)		
4 to number of bytes write + 4	7:0	Number of bytes intended to write		

3.7.8 Update PWM Frequency (I²C: 47h)

Byte	Bits	Description	Reset	Type
0	7:0	Valid PWM frequency input range: 1 to 187 KHz	x0	wr

3.7.9 Update LED Usage Time (I²C: 48h)

Byte	Bits	Description	Reset	Type
0	0	Enable updating LED usage time in EEPROM every 0.1 hour when LED is lit 0 – Disable LED usage time updating 1 – Enable LED usage time updating	x0	wr

3.7.10 Get LED Usage Time (I²C: 49h)

Return the number of hours LED was used from EEPROM.

Bytes	Bits	Description	Reset	Type
0	7:0	LED hours used (31:24)	x0	r
1	7:0	LED hours used (23:16)		
2	7:0	LED hours used (15:8)		
3	7:0	LED hours used (7:0)		

3.7.11 Reset LED Usage Time (I²C: 4Ah)

Byte	Bits	Description	Reset	Type
0	0	1 - Reset LED usage time in EEPROM	x0	w

3.7.12 LED Current Control (I²C: 4Bh)

The PWM for RGB LED is in range 0 to 255.

Table 3-34. (1)

Bytes	Bits	Description	Reset	Type
0	7:0	Red LED PWM (MSB)		wr
1	7:0	Red LED PWM (LSB)		
2	7:0	Green LED PWM (MSB)		
3	7:0	Green LED PWM (LSB)		
4	7:0	Blue LED PWM (MSB)		
5	7:0	Blue LED PWM (LSB)		

(1) Since DLPC6401 automatically adjusts the currents for maintaining the White Point, this command is primarily intended to be read only. However, if you want to explicitly control the LED PWMs, then you can do so by disabling the White Point Correction algorithm.

3.7.13 Get LED Sensor Value (I²C: 4Ch)

Bytes	Bits	Description	Reset	Type
0	7:0	Red LED PWM (MSB)		r
1	7:0	Red LED PWM (LSB)		
2	7:0	Green LED PWM (MSB)		
3	7:0	Green LED PWM (LSB)		
4	7:0	Blue LED PWM (MSB)		
5	7:0	Blue LED PWM (LSB)		

3.7.14 Build LED Sensor Table (I²C: 4Dh)

Byte	Bits	Description	Reset	Type
0	0	**Build LED Sensor Table** 1 – Rebuild LED sensor table (bit value is don't care)	x0	w

3.8 Framing Commands

3.8.1 Preset Input or Output Resolution (I²C: 50h)

Table 3-35 shows the supported input and output resolutions for the 0.45 WXGA DMD.

Table 3-35.

Bytes	Bits	Description					Reset	Type
		Input Resolution Options (Decimal)[1]	Source Resolution		Display Resolution			
			H	V	H	V		
0	4:0	0	320	240	1280	800	d0	wr
		1	480	320	1280	800		
		2	640	360	1280	800		
		3	640	480	1280	800		
		4	800	600	1280	800		
		5	1024	768	1280	800		
		6	1280	720	1280	800		
		7	1280	800	1280	800		
		8	1366	768	1280	800		
		9	1400	1050	1280	800		
		10	1600	1200	1280	800		
		11	1280	1024	1280	800		
		12	640	480	640	480		
		13	720	480	720	480		
		14	800	600	800	600		
		15	1024	768	1024	768		
		16	1280	720	1280	720		
		17 – Manual scale mode						

(1) Preset resolutions are for a commonly used resolution. If other resolutions are needed, use Manual Input or Output display resolution commands to achieve any desired resolution.

3.8.2 Manual Input Display Resolution (I²C: 51h)

Active input resolution for the number of pixels as horizontal resolution and number of lines as vertical resolution:

Bytes	Bits	Description	Reset	Type
0	7:0	Input horizontal resolution (LSB) b(7:0)	d0	wr
1	7:3	Reserved		
	2:0	Input horizontal resolution (MSB) b(10:8)		
2	7:0	Input vertical resolution (LSB) b(7:0)		wr
3	7:3	Reserved		
	2:0	Input vertical resolution (MSB) b(10:8)		

3.8.3 Manual Output Display Resolution (I²C: 52h)

Output resolution for the number of pixels as horizontal resolution and number of lines as vertical resolution in the DMD:

Bytes	Bits	Description	Reset	Type
0	7:0	Output horizontal resolution (LSB) b(7:0)	d0	wr
1	7:3	Reserved		
	2:0	Output horizontal resolution (MSB) b(10:8)		
2	7:0	Output vertical resolution (LSB) b(7:0)		wr
3	7:3	Reserved		
	2:0	Output vertical resolution (MSB) b(10:8)		

3.8.4 Crop First Active Line in Frame (I²C: 53h)

Table 3-36. (1)

Bytes	Bits	Description	Reset	Type
0	7:0	First active line in frame (relative to active lines) Image crop control MSB (7:0)	d0	wr
1	2:0	First active line in frame (relative to active lines) Image crop control MSB (10:8)		

(1) Crop number of lines from top of active input area, default = 0 (no top line cropping)

3.8.5 Crop Last Active Line in Frame (I²C: 54h)

Table 3-37. (1)

Bytes	Bits	Description	Reset	Type
0	7:0	Last active line in frame (relative to active lines) Image crop control MSB (7:0)	d0	wr
1	2:0	Last active line in frame (relative to active lines) Image crop control MSB (10:8)		

(1) Crop number of lines from bottom of active input area. default = 0 (no bottom line cropping)

3.8.6 Crop First Active Pixel in Frame (I²C: 55h)

Table 3-38. (1)

Bytes	Bits	Description	Reset	Type
0	7:0	First active pixel in line (relative to active pixels) Image crop control MSB (7:0)	d0	wr
1	2:0	First active pixel in line (relative to active pixels) Image crop control MSB (10:8)		

(1) Crop number of pixels from left of active input area, default = 0 (No Left Pixel Cropping)

3.8.7 Crop Last Active Pixel in Frame (I²C: 56h)

Table 3-39. (1)

Bytes	Bits	Description	Reset	Type
0	7:0	Last active pixel in line (relative to active pixels) Image crop control MSB (7:0)	d0	wr
1	2:0	Last active pixel in line (relative to active pixels) Image crop control MSB (10:8)		

(1) Crop number of pixel from right of active input area, default = 0 (no right pixel cropping)

3.8.8 Display Position in Frame (I²C: 66h)

This command places the image at a certain position within the DMD dimension.

Bytes	Bits	Description	Reset	Type
0	7:0	Position – 0 (Top left) 1 (Top right) 2 (Center) 3 (Bottom left) 4 (Bottom right) 5 (User Defined)	d0	wr

3.8.9 Display Position User Defined(I²C:68h)

This command allows the image to be positioned anywhere within the DMD dimension. This command automatically changes the Display Position in Frame (66h) command to USER DEFINED mode.

Bytes	Bits	Description	Reset	Type
0	7:0	Display First Pixel (LSB)	d0	wr
1	7:0	Display First Pixel (MSB)		
2	7:0	Display First Line (LSB)		
3	7:0	Display First Line (MSB)		

3.9 Color Correction Interface (CCI) Commands

The white point correction algorithm is responsible for maintaining desired white point by mitigating LED output variations.

There are two types of LED variations to consider:

1. Some LED variation differers with manufacturing, causing some LED samples to have different brightness or color points. The desired white point information is available from defined BrilliantColor feature Looks. Each BrilliantColor feature look usually has more than one duty cycle associated with it to account for these LED variations (called duty cycle cloud). The WPC algorithm selects the closest matching duty cycle from the duty cycle cloud to achieve the desired white point on a particular projector sample
2. Variations in brightness or color points with temperature or time LED brightness and even color points (in some cases) vary with temperature and over a certain period of time. These variations could shift the white point also. To handle brightness variations, the WPC algorithm varies the LED current to maintain the same white point. The color variation method for WPC is not supported by the DLPC6401.

The WPC algorithm operates in two modes:

1. **CCA adjust only mode**: In this mode, the algorithm runs at a fixed current level set by the user. The sensor feedback mechanism is used for refreshing measured CCA values and providing input to CCA algorithm to achieve the desired color points.
2. **Current adjust mode**: In this mode, applied current to the LEDs is varied continuously based on continuous sensor feedback. The measured CCA values are sent to CCA algorithm only when the

BrilliantColor looks are changed.

3.9.1 CCI Control (I²C: 57h)

CCA adjustment mode: In this mode, the algorithm runs at a fixed current level set by the user. The sensor feedback mechanism is used for refreshing measured CCA values and providing input to the CCA algorithm to achieve the desired color points.

CCA adjustment mode: In this mode, the algorithm runs at a fixed current level set by the user. The sensor feedback mechanism is used for refreshing measured CCA values and providing input to the CCA algorithm to achieve the desired color points

Bytes	Bits	Description	Reset	Type
0	7:2	Reserved	d0	wr
	1	**Color Correction Mode** 0 – CCA adjust mode (color correction performed via periodic CCA while LEDs driven at full current) 1 – Current adjust mode (color correction performed via continuous LED drive current adjustment)		
	0	0 – CCI disable 1 – CCI enable		

3.9.2 LED Max Current (I²C: 58h)

This command gets or sets maximum LED current value in terms of the PWM value needed for the color correction algorithm.

Bytes	Bits	Description	Reset	Type
0	7:0	Max current for red LED(15:8)	d0	wr
1	7:0	Max current for red LED(7:0)		
2	7:0	Max current for green LED(15:8)		
3	7:0	Max current for green LED(7:0)		
4	7:0	Max current for blue LED(15:8)		
5	7:0	Max current for blue LED(7:0)		

3.9.3 Set Calibration Data (I²C: 59h)

Table 3-40 contains LED calibration values – x, y, L, and sensor output for red, green, and blue LEDs.

Table 3-40.

Bytes	Bits	Description	Reset	Type
0	7:0	Rx(15:8) – External sensor x value represented in fixed point 1.5 for RED (MSB)	d0	wr
1	7:0	Rx(7:0) – External sensor x value represented in fixed point 1.5 for RED (LSB)	d0	wr
2	7:0	Ry(15:8) – External sensor y value represented in fixed point 1.5 for RED (MSB)	d0	wr
3	7:0	Ry(7:0) – External sensor y value represented in fixed point 1.5 for RED (LSB)	d0	wr
4	7:0	RL(15:8) – External sensor L value for red (MSB)	d0	wr
5	7:0	RL(7:0) – External sensor L value for red (LSB)	d0	wr
6	7:0	RS(31:24) – Internal sensor PWM value for red	d0	wr
7	7:0	RS(23:16) – Internal sensor PWM value for red	d0	wr
8	7:0	RS(15:8) – Internal sensor PWM value for red	d0	wr
9	7:0	RS(7:0) – Internal sensor PWM value for red	d0	wr
10	7:0	Gx(15:8) – External sensor x value represented in fixed point 1.5 for green (MSB)	d0	wr
11	7:0	Gx(7:0) – External sensor x value represented in fixed point 1.5 for green (LSB)	d0	wr
12	7:0	Gy(15:8) – External sensor y value represented in fixed point 1.5 for green (MSB)	d0	wr

Table 3-40. (continued)

Bytes	Bits	Description	Reset	Type
13	7:0	Gy(7:0) – External sensor y value represented in fixed point 1.5 for green (LSB)	d0	wr
14	7:0	GL(15:8) – External sensor L value for green (MSB)	d0	wr
15	7:0	GL(7:0) – External sensor L value for green (LSB)	d0	wr
16	7:0	GS(31:24) – Internal sensor PWM value for green	d0	wr
17	7:0	GS(23:16) – Internal sensor PWM value for green	d0	wr
18	7:0	GS(15:8) – Internal sensor PWM value for green	d0	wr
19	7:0	GS(7:0) – Internal sensor PWM value for green	d0	wr
20	7:0	Bx(15:8) – External sensor x value represented in fixed point 1.5 for blue (MSB)	d0	wr
21	7:0	Bx(7:0) – External sensor x value represented in fixed point 1.5 for blue (LSB)	d0	wr
22	7:0	By(15:8) – External sensor y value represented in fixed point 1.5 for blue (MSB)	d0	wr
23	7:0	By(7:0) – External sensor y value represented in fixed point 1.5 for blue (LSB)	d0	wr
24	7:0	BL(15:8) – External sensor L value for blue (MSB)	d0	wr
25	7:0	BL(7:0) – External sensor L value for blue (LSB)	d0	wr
26	7:0	BS(31:24) – Internal sensor PWM value for blue	d0	wr
27	7:0	BS(23:16) – Internal sensor PWM value for blue	d0	wr
28	7:0	BS(15:8) – Internal sensor PWM value for blue	d0	wr
29	7:0	BS(7:0) – Internal sensor PWM value for blue	d0	wr

3.9.4 Current Versus Sensor Table Mailbox (I^2C: 5Ah)

3.9.4.1 Internal Sensor Word

Bytes	Bits	Description	Reset	Type
0	7:0	Internal sensor value for red (MSB)	d0	wr
1	7:0	Internal sensor value for red (LSB)		
2	7:0	Internal sensor value for green (MSB)		
3	7:0	Internal sensor value for green (LSB)		
4	7:0	Internal sensor value for blue (MSB)		
5	7:0	Internal sensor value for blue (LSB)		

3.9.4.2 Current Versus Sensor Table Mailbox Sub-Address Map

Word Address	Word
0	Internal sensor word – 0
1	Internal sensor word – 1
2	Internal sensor word – 2
3	Internal sensor word – 3
4	Internal sensor word – 4
5	Internal sensor word – 5
6	Internal sensor word – 6
7	Internal sensor word – 7
8	Internal sensor word – 8
9	Internal sensor word – 9
10	Internal sensor word – 10
11	Internal sensor word – 11
12	Internal sensor word – 12

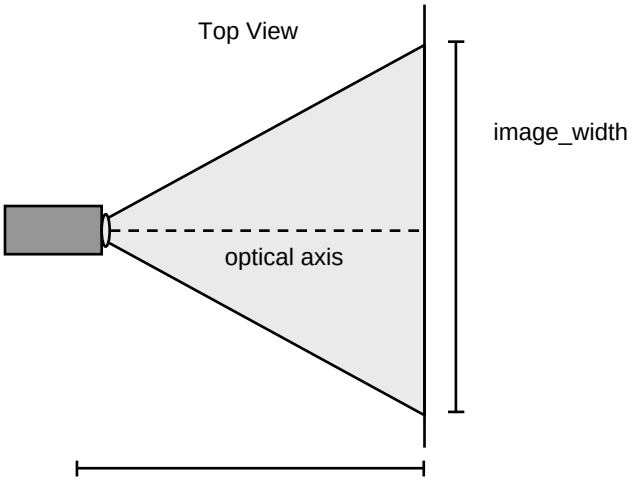
Word Address	Word
13	Internal sensor word – 13
14	Internal sensor word – 14
15	Internal sensor word – 15
16	Internal sensor word – 16
17	Internal sensor word – 17

3.10 Keystone Effect

Keystone correction allows the user to digitally compensate for distorted images when the projector is tilted up or down. Keystone correction is specified by the pitch angle between the optical axis and level ground. This angle is specified in degrees and in s7.8 format. The typical range is ±40°, but decreases if an optical configuration operates outside of the DLPC6401 product specification limits.

For both full screen images and sub-images, the entire DMD area is corrected.

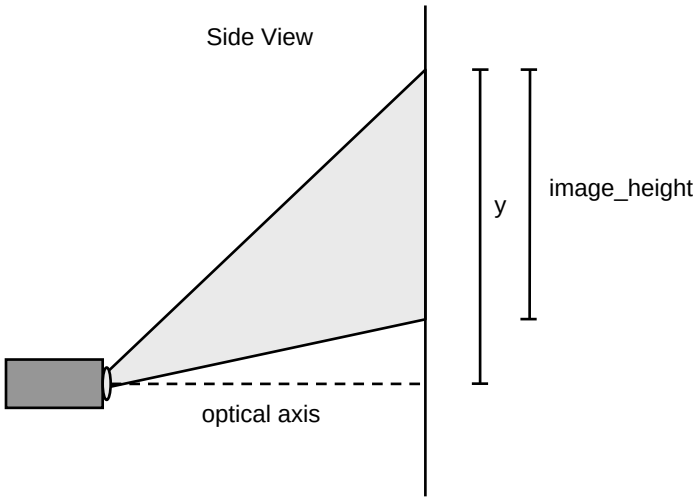
Keystone correction in DLPC6401 corrects for both overall and local area aspect ratio distortion. However, the vertical offset and throw ratio specified to the keystone command must match the optics of the projector. Figure 3-8 and Figure 3-9 show the calculation of these values. .



Throw Ratio = (distance / image_width)

Note this value must be left shifted 8 bits
(multiply by 256) to convert to 8.8 format

Figure 3-8. Calculating Throw Ratio for Keystone



$$\text{Vertical Offset} = (2 \times y / \text{image\_height}) - 1.0$$

Note this value must be left shifted 8 bits
(multiply by 256) to convert to s7.8 format

Figure 3-9. Calculating Vertical Offset for Keystone

3.10.1 Keystone Command (I²C: 60h)

Word	Bits	Description	Reset	Type
0	15:0	**Pitch** Limit –40° to 40° Input format is signed 7.8, with 7 significant bits, 8 fractional bits, and one sign bit (bit 15)	x0	wr
1	15:0	**Throw ratio** Format is unsigned 8.8, with 8 significant bits and 8 fractional bits		
2	15:0	**DMD offset** Format is signed 7.8, with 7 significant bits, 8 fractional bits, and one sign bit (Bit 15)		

3.10.2 Auto Keystone Control (I²C: 61h)

Table 3-41. (1)

Byte	Bits	Description	Reset	Type
0	7:1	Reserved	x0	wr
	0	**Enable or Disable Auto Keystone** 0 – Disable 1 – Enable		

(1) If tilt sensor is used, this command can be used to enable or disable automatic keystone.

3.11 Program Mode Flash Programming Commands

Flash programming commands apply only when the DLPC6401 system operates in programming mode. To enable programming write a value of 0x01 to the programming mode register.

3.11.1 Programming Mode (I²C: 30h)

Read the system status byte verify when programming mode is active.

Table 3-42.

Byte	Bits	Description	Reset	Type
0	7:1	Others: reserved	x0	wr
	0	**Programming Mode** 1 - Enter flash programming mode[1]		

(1) Writing a 1 to the register causes the system to go through a partial system shutdown and enter flash programming mode. Programming mode allows OEMs to program the flash memory using special commands.

3.12 3D

For 3D operation in the DLPC6401, set the external input source frame rate to 120 Hz and set the display frame rate to 120 Hz. Then select the BrilliantColor™ feature Control look (I2C: 32h) command to select a look containing 3D sequence.

3.12.1 3D Frame Synchronization Command (I²C: 62h)

There is a case where a 3D integrated circuit can convert some 3D signals into 120-Hz signals that the DLPC6401 device can accept. However, DLPC6401 device does not automatically record which frame is for the left or right eye. The sequence for 3D has two sub-sequences, one for the left eye and one for the right eye. Connect left and right synchronization signal to one of the GPIO pins in the DLPC6401 device. See Section 3.7.5 for more details about pin configuration for frame synchronization.

Table 3-43. Needs a Title

Byte	Bits	Description	Reset	Type
0	7:1	Reserved	x0	wr
	0	**Enable Left or Right Frame Synchronization** 0 – Disable 1 – Enable		
1	7:1	Reserved		
	0	**Left or Right Frame Order** 0 – Right to left 1 – Left to right		
2	7:0	GPIO pin number to use Pin number must be between 0 to 37. Please see Section 3.7.5 for details		

3.13 Temperature Control

Temperature control shuts down the projector when it is too hot. The user can set a threshold temperature between –50°C and 120°C. When temperature control is enabled, the system status overtemperature warning bit determines if the temperature is over the threshold.

3.13.1 Temperature Control Command (I²C: 63h)

Table 3-44. (1)

Byte	Bits	Description	Reset	Type
0	7:2	Others: reserved	x0	wr
	0:0	**Temperature Control Enable** 0 – Disable temperature control 1 – Enable temperature control		
	1:1	**Threshold Over Temperature Use** 0 **-** No action 1 **-** Shutdown projector when temperature is over the threshold		
1		Temperature Threshold in Degree Celsius (MSB) Limit –50°C to 120°C Input format is signed 8.4, with 8 significant bits, 4 fractional bits, and one sign bit (bit 15)		
2		Temperature Threshold in Degree Celsius (LSB) Limit –50°C to 120°C		

(1) If temperature control is disabled, the temperature threshold setting has no effect.

3.13.2 Read Temperature Command (I^2C: 64h)

Word	Bits	Description	Reset	Type
0	15:0	Temperature in Celsius Format is signed 8.4, with 8 significant bits, 4 fractional bits, and one sign bit (bit 15). The host must convert raw temp data to degrees in Celsius using the following equation: Temp (C) = Raw Temp Data / 10h	x0	r

3.14 RGB Level Check

The *RGB level check* command sets the threshold for red, green, and blue levels. If the maximum level of channels goes below the threshold for a particular number of frames specified by frame threshold, then the LEDs switch off. Disabling the command switches on the LEDs.

LEDs switch on when the levels are above the threshold for a particular number of frames (frame threshold).

3.14.1 RGB Level Check Command (I²C: 65h)

Table 3-45.

Byte	Bits	Description	Reset	Type
0	7:1	Reserved	x0	wr
	0	**Enable RGB Level Check** 0 – Disable 1 – Enable		
1		Red threshold MSB (0 – 3)		
2		Red threshold LSB (0 – 0xff)		
3		Green threshold MSB (0 – 3)		
4		Green threshold LSB (0 – 0xff)		
5		Blue threshold MSB (0 – 3)		
6		Blue threshold LSB (0 – 0xff)		
7		Frame threshold MSB (0 – 3)		
8		Frame threshold LSB (0 – 0xff)		

3.15 DLPC6401 OSD

The DLPC6401 ASIC allows the projector designer to create a unique on-screen display (OSD). The details of the display can be customized to meet the needs of the individual projector or OEM. The DLPC6401 device OSD is a character-based system. The OSD hardware draws characters (including both text and icons) and rectangles (windows). Rectangles may overlap rectangles and characters may overlap rectangles, but characters can never overlap characters. Also, the systsem always draws characters on top of rectangles. The system organizes characters into lines by the OSD design tool. The OSD hardware uses the character lines to draw characters on the image correctly.

The DLPC6401 software includes a standard OSD design, which contains three types of OSD:

- Pull-down menu OSD: this type of OSD has a number of items the user can highlight and scroll through from top to bottom for the selections.
- Slider bar OSD has a progress bar and digits changing dynamically based on user input. An example of this is display brightness change.
- Information OSD such as overtemperature warning, low battery warning.

The OSD commands defined in this section operate only with the standard OSD, and each command requires a menu or item ID parameter to affect the individual menus. For the menu and item ID information, contact the engine manufacturer or design house that provided the DLPC6401 device software flash image.

3.15.1 Position OSD Display Command (I²C: 70h)

The *position OSD display* command positions the OSD at certain locations within the DMD. The following are possible positions for the OSD:

- Top Left = 0
- Top Right = 1
- Center = 2
- Bottom Left = 3

- Bottom Center = 4
- Bottom Right = 5

The OSD design tool is required to design OSD. The OSD header file is required to locate the particular OSD menu ID for input. This command applies to *slider*, *pull-down*, and *information* OSD.

Table 3-46.

Byte	Bits	Description	Reset	Type
0	7:0	Menu ID (MSB) – look for corresponding menu ID in OSD header file	x0	wr
1	7:0	Menu ID (LSB)		
2	7:0	**Position** 0 (Top left) 1 (Top right) 2 (Center) 3 (Bottom left) 4 (Bottom center) 5 (Bottom right)		

3.15.2 Slider Bar Display Start OSD Command (I²C: 71h)

The *slider bar display start* OSD command displays the slider bar OSD adjustment with the option to enable the slider bar display, or not. If set to no initial display, the user can use command 72h to set any desired slider bar position. For initial display (set to TRUE), the slider bar always sets and displays the initial position.

Table 3-47.

Word	Bits	Description	Reset	Type
0	15:0	Menu ID – look for corresponding slider bar menu ID in OSD header file	x0	wr
1	15:0	Option for slider bar initial display 1 – TRUE: Display slider bar default initial position 0 – FALSE: Do not display slider bar initial position		

3.15.3 Update Slider Progress OSD Command (I²C: 72h)

The *update slider progress* OSD command allows updates to the slider program bar and corresponding digits. Figure 3-10 shows an example of a slider bar. The user must input a 32-bit minimum and maximum value. For example ,a temperature value of 32° with minimum temperature = 0 and maximum temperature = 100, Min ≤ Value ≤ Max.

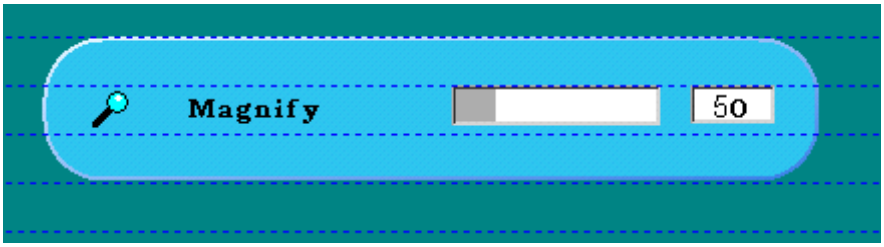


Figure 3-10. Slider Bar Example

Word	Bits	Description	Reset	Type
0	15:0	Item ID – look for corresponding slider bar item ID within the menu ID in the OSD header file	x0	wr
1	15:0	Value (MSB word)		
2	15:0	Value (LSB word)		
3	15:0	Min value (MSB word)		
4	15:0	Min value (LSB word)		
5	15:0	Max value (MSB word)		
6	15:0	Max value (MSB word)		

3.15.4 Pull-Down Menu Display Start OSD Command (I^2C: 73h)

The *pull-down menu display start* OSD command allows the user to do the pull-down menu start display. The first menu item is highlighted when displayed, if default item ID is used; otherwise, the menu item is highlighted by the item ID entered by the user. Figure 3-11 shows an example.

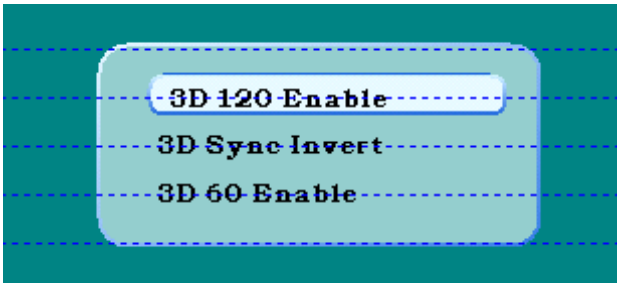


Figure 3-11. Pull-Down Menu Example

Table 3-48.

Word	Bits	Description	Reset	Type
0	15:0	Menu ID – look for corresponding pull-down menu ID in OSD header file	x0	wr
1	15:0	Item ID – this item ID causes the corresponding menu item to be highlighted. Look for corresponding item ID in OSD header file. item ID = 0 is default. This causes the first menu item to be highlighted		

3.15.5 Pull-Down Navigation OSD Command (I^2C: 74h)

The *pull-down navigation* OSD command allows the user to move the menu item up or down. Any target menu item becomes highlighted. When the user reaches the last menu item, the selection moves to the first menu item when the down key is pressed. Be sure to include a direction input (either up or down) as part of the input. The DLPC6401 device supports a vertical menu style only.

- Move up = 0
- Move down = 1

Table 3-49.

Byte	Bits	Description	Reset	Type
0	7:0	Menu ID (MSB) – look for corresponding pull-down menu ID in OSD header file	x0	wr
1	7:0	Menu ID (LSB)		
2	7:0	Current item ID (MSB) – look for corresponding pull-down item ID within the menu ID in OSD header file.		
3	7:0	Current item ID (LSB)		
4	7:0	Current window (MSB)		
5	7:0	Current window (LSB)		
6	7:0	**Pull-Down Menu Move Direction** 0 = Move up 1 = Move down		

3.15.6 Display Exit OSD Command (I²C: 75h)

The *display exit OSD* command exits from the OSD menu display for all OSD types, including information display OSD.

Word	Bits	Description	Reset	Type
0	15:0	Menu ID – look for corresponding menu ID in OSD header file	x0	wr

3.15.7 Information Display Start OSD Command (I²C: 76h)

The *information display start OSD* command shows the static information display. Use *display exit OSD* command (75h) to exit the OSD. Figure 3-12 shows an information display example.

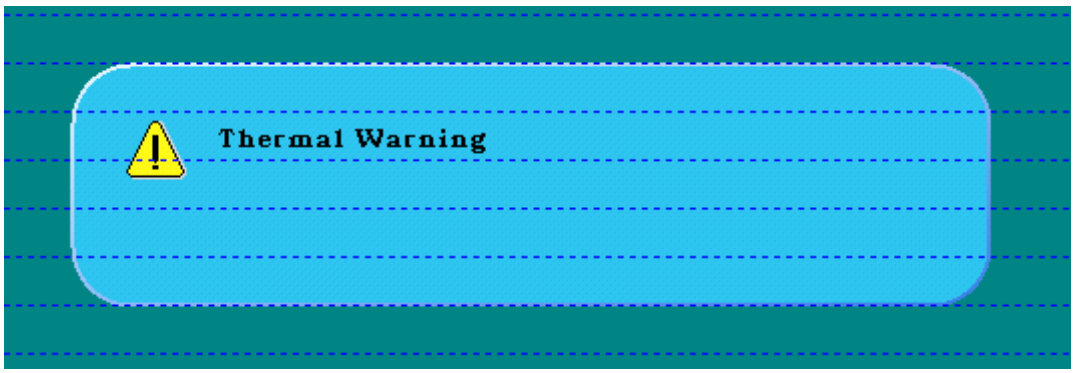


Figure 3-12. Information Display OSD Example

Word	Bits	Description	Reset	Type
0	15:0	Menu ID – look for corresponding menu ID in OSD header file	x0	wr

3.15.8 Get Current Displayed Menu ID Command (I²C: 77h)

The *get current displayed menu ID* command allows the user to retrieve the currently active OSD Menu ID. Only one OSD is active at a time.

Word	Bits	Description	Reset	Type
0	15:0	Menu ID – retrieve current displayed OSD Menu ID	x0	r

3.15.9 Set or Get Language ID Command (I²C: 78h)

The *set or get language ID* command allows the user to set the language. It also displays the current language ID selected.

Byte	Bits	Description	Reset	Type
0	7:0	Lang ID (MSB) – look for corresponding language ID in OSD header file	x0	wr
1	7:0	Lang ID (LSB)		

3.15.10 Set List Index as Dynamic Field Text Command (I²C: 79h)

The *set list index as dynamic field text* command allows the user to text defined in a list for any style of OSD, including pull-down menu OSD. Enter the list field ID defined in corresponding OSD style in the OSD header file. The list index refers to the specific text or icon availble for display. If the user changes the language, the corresponding text changes for that language as well. Figure 3-13 illustrates a language menu where a list is created (using list tool in OSD design) for six list items indexed from 0 to 5. When the user chooses “Display”, the user can refer to this list field ID (0x3085) and index 0, or “Keystone” for index 1, and so forth.

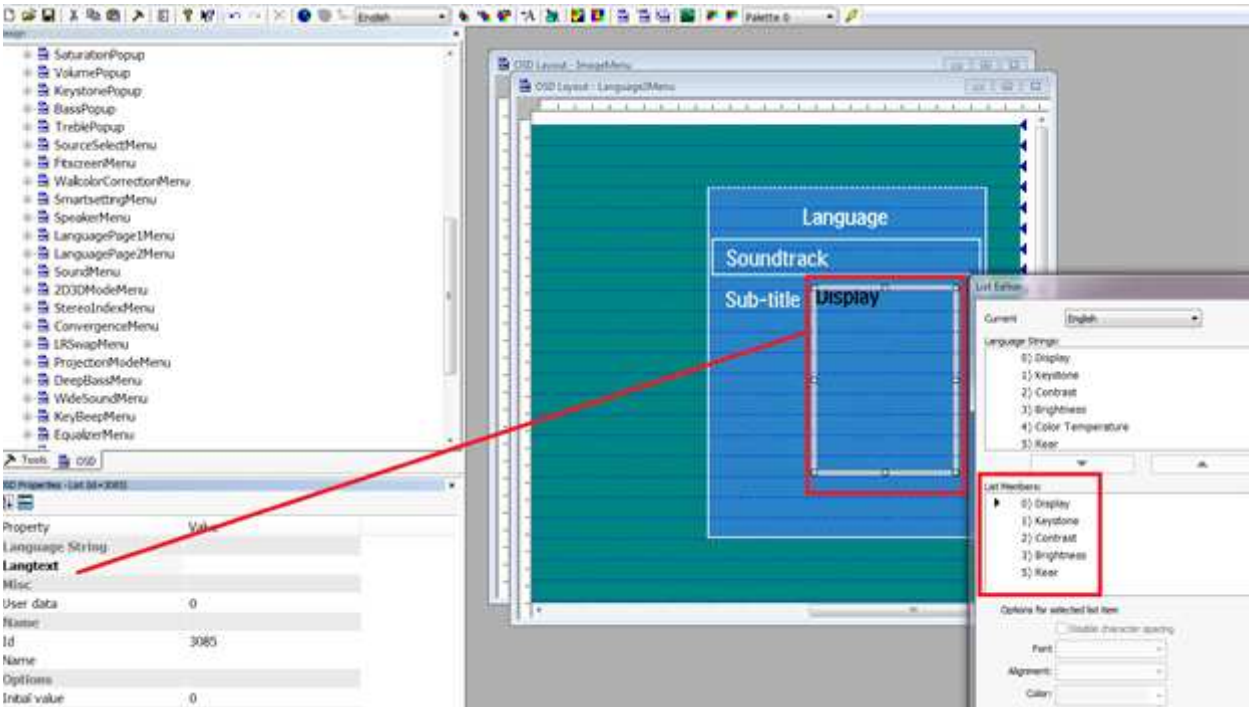


Figure 3-13.

Word	Bits	Description	Reset	Type
0	15:0	List field ID – look for the corresponding field ID in a menu in OSD header file	x0	w
1	15:0	List index as integer starting from 0 to the end of list. The index points to a specific text the user created in the list.		

Revision History

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

Changes from Original (December 2013) to A Revision	Page
• Updated *Figure 2-1*	9
• Added *Section 2.6 Power Down Sequence*	10

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 (www.ti.com/legal/termsofsale.html) 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