DLPC3470 and DLPC3478 Software Programmer's Guide

Programmer's Guide

Literature Number: DLPU075A
October 2018–Revised April 2019

Contents

List of Figures

List of Tables

DLPC3470 and DLPC3478 Software Programmer's Guide

1 Introduction

This document details the software interface requirements for a DLPC347x DLP® Pico™ Light Controller based system. The DLPC347x DLP Pico Light Controllers support Display, 3D Light Control, and 3D printing applications with the use of several TRP DMD chips. Target markets include:

- 3D Optical Inspection
- 3D Measurement
- 3D Facial and Fingerprint Recognition
- 3D Printing
- Robotic Vision
- Machine Vision

The DLPC347x has three modes for use in Light Control applications:

- External Pattern Streaming Mode
- Internal Pattern Streaming Mode
- Splash Pattern Mode

External Pattern Streaming mode displays pattern data received via an external video port. Figure 1 shows a system configured for this mode.

Internal Pattern Streaming mode displays 1D pattern data stored in flash. Figure 2 shows a system configured for this mode.

Splash Pattern mode displays 2D patterns stored as images in flash. Figure 2 shows a system configured for this mode.

Figure 1. DLPC347x External Pattern Streaming Mode configuration

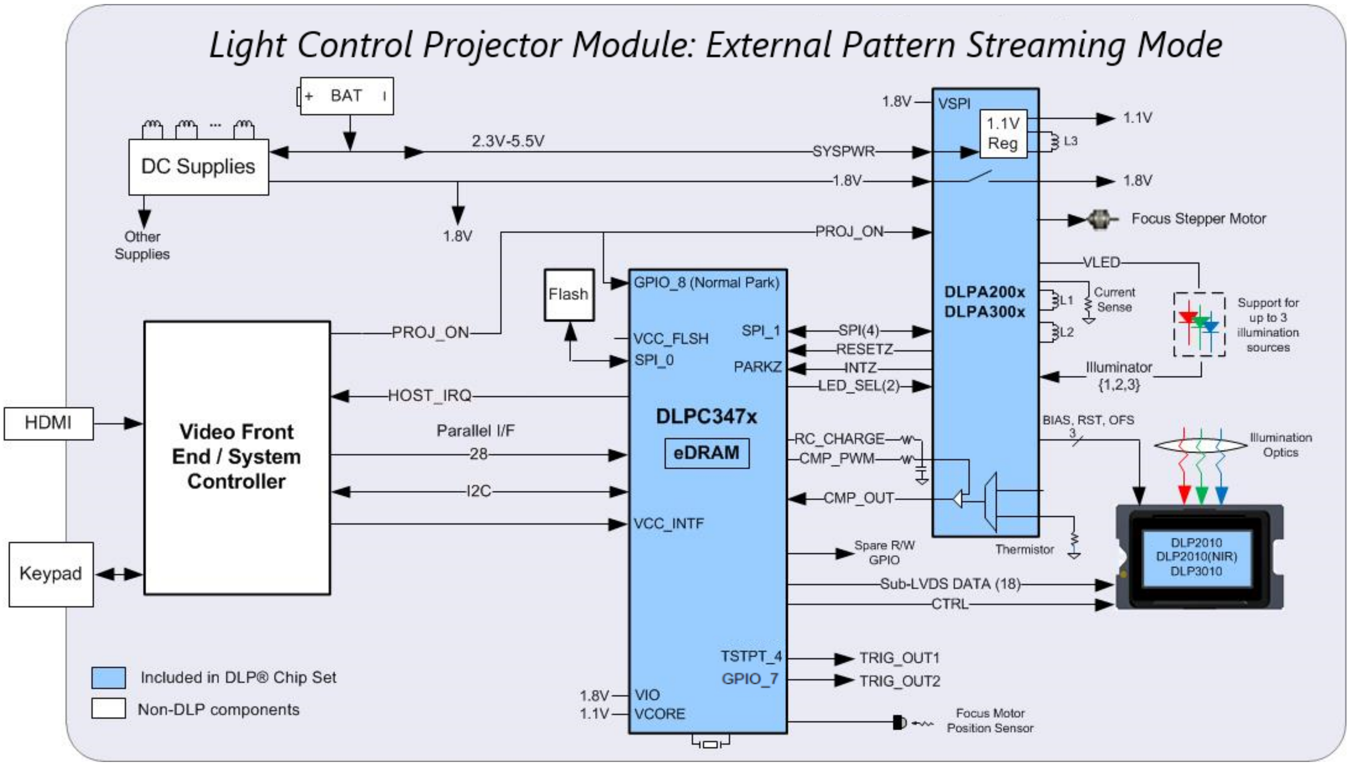
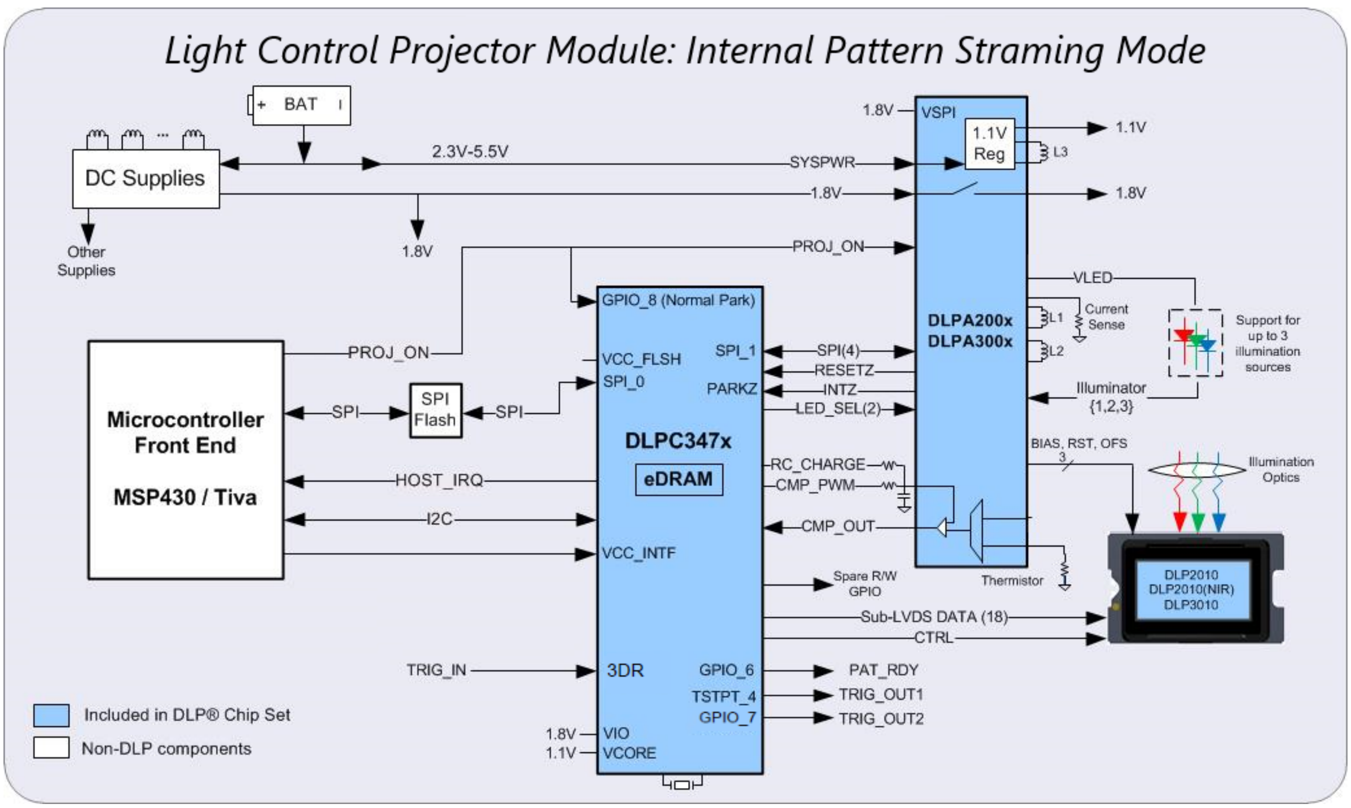


Figure 2. DLPC347x Internal Pattern Streaming Mode / Splash Pattern Mode configuration



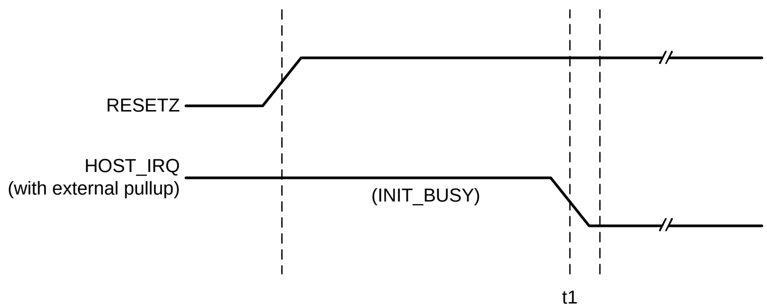
(1) Splash Pattern Mode does not support TRIG_IN and PAT_RDY signals.

2 System Overview

2.1 Initialization

The DLPC347x employs a boot ROM and associated boot software. This resident boot code consists of the minimum code necessary to load the software from flash to internal RAM for execution. For most DLPC347x product configurations, an external flash device can store the main application code, along with the other configuration and operational data required by the system for normal operation.

The HOST_IRQ signal provides the completion status of the DLPC347x system initialization when the system is powered on. Once PROJ_ON is high, HOST_IRQ is tri-stated with an external pull-up. After RESETZ is applied, the controller drives HOST_IRQ high while initializing and then drives it low once initialization has completed. The initialization period is determined by the boot configuration and can vary from one system to another.



t1: the first falling edge of HOST_IRQ indicates auto-initialization completion

Figure 3. HOST_IRQ Timing Diagram

NOTE: Make sure that I²C access to the DLPC347x does not start until HOST_IRQ goes low. Sending an I²C command while HOST_IRQ is high can prevent the system from booting.

2.2 I²C Interface Specification

The protocol used in communicating information to DLPC347x consist of a serial data bus conforming to the Philips I²C specification, up to 100 kHz. Commands are executed using I²C, where the DLPC347x behaves as a slave.

The supported I²C transaction type for both writes and reads is shown in Table 1. The I²C interface supports variable-size transactions (i.e. variable number of bytes as parameters) depending on the command. The list of supported commands are discussed in the next section.

Table 1. I²C Write and Read Transactions

Transaction	Address (One byte)(1)	Sub-Address (One byte)(2)	Remaining Data Bytes(3)
Write or Read Request	36h (or 3Ah)	Command Opcode	Parameter values (0 → N bytes)
Read Response	37h (or 3Bh)		

(1) The address corresponds to the chip address of the controller.
(2) The subaddress corresponds to a command.
(3) The data (if present) corresponds to any required command parameters.

The standard parameter byte format is shown below:

msb	Parameter Byte						lsb
b7	b6	b5	b4	b3	b2	b1	b0

3 System Write/Read Commands

Table 2. List of System Write/Read Software Commands

Command Type	Command Description	OpCode (hex)	Reference
General Operation Commands			
Write	Write Operating Mode Select	05	Section 3.1.1
Read	Read Operating Mode Select	06	Section 3.1.2
Write	Write External Video Source Format Select	07	Section 3.1.3
Read	Read External Video Source Format Select	08	Section 3.1.4
Write	Write Test Pattern Select	0B	Section 3.1.5
Read	Read Test Pattern Select	0C	Section 3.1.6
Write	Write Splash Screen Select	0D	Section 3.1.7
Read	Read Splash Screen Select	0E	Section 3.1.8
Read	Read Splash Screen Header	0F	Section 3.1.9
Write	Write Image Crop	10	Section 3.1.10
Read	Read Image Crop	11	Section 3.1.11
Write	Write Display Size	12	Section 3.1.12
Read	Read Display Size	13	Section 3.1.13
Write	Write Display Image Orientation	14	Section 3.1.14
Read	Read Display Image Orientation	15	Section 3.1.15
Write	Write Display Image Curtain	16	Section 3.1.16
Read	Read Display Image Curtain	17	Section 3.1.17
Write	Write Image Freeze	1A	Section 3.1.18
Read	Read Image Freeze	1B	Section 3.1.19
Write	Write Look Select	22	Section 3.1.20
Read	Read Look Select	23	Section 3.1.21
Read	Read Sequence Header Attributes	26	Section 3.1.22
Read	Read DMD Sequencer Sync Mode	2C	Section 3.1.23
Write	Write Execute Batch File	2D	Section 3.1.24
Write	Write Input Image Size	2E	Section 3.1.25
Read	Read Input Image Size	2F	Section 3.1.26
Write	Write Splash Screen Execute	35	Section 3.1.27
Illumination Control Commands			
Write	Write LED Output Control Method	50	Section 3.2.1
Read	Read LED Output Control Method	51	Section 3.2.2
Write	Write RGB LED Enable	52	Section 3.2.3
Read	Read RGB LED Enable	53	Section 3.2.4
Write	Write RGB LED Current PWM	54	Section 3.2.5
Read	Read RGB LED Current PWM	55	Section 3.2.6
Read	Read CAIC LED Max Available Power	57	Section 3.2.7
Write	Write RGB LED Max Current PWM	5C	Section 3.2.8
Read	Read RGB LED Max Current PWM	5D	Section 3.2.9
Read	Read CAIC RGB LED Current PWM	5F	Section 3.2.10
Image Processing Control Commands			
Write	Write Local Area Brightness Boost Control	80	Section 3.3.1
Read	Read Local Area Brightness Boost Control	81	Section 3.3.2
Write	Write CAIC Image Processing Control	84	Section 3.3.3
Read	Read CAIC Image Processing Control	85	Section 3.3.4
Write	Write Keystone Correction Control	88	Section 3.3.5

Table 2. List of System Write/Read Software Commands (continued)

Command Type	Command Description	OpCode (hex)	Reference
Read	Read Keystone Correction Control	89	Section 3.3.6
Light Control Commands			Section 3.4
Write	Write Trigger In Configuration	90	Section 3.4.1
Read	Read Trigger In Configuration	91	Section 3.4.2
Write	Write Trigger Out Configuration	92	Section 3.4.3
Read	Read Trigger Out Configuration	93	Section 3.4.4
Write	Write Pattern Ready Configuration	94	Section 3.4.5
Read	Read Pattern Ready Configuration	95	Section 3.4.6
Write	Write Pattern Configuration	96	Section 3.4.7
Read	Read Pattern Configuration	97	Section 3.4.8
Write	Write Pattern Order Table Entry	98	Section 3.4.9
Read	Read Pattern Order Table Entry	99	Section 3.4.10
Read	Read Light Control Sequence Version	9B	Section 3.4.11
Read	Read Validate Exposure Time	9D	Section 3.4.12
Write	Write Internal Pattern Control	9E	Section 3.4.13
Read	Read Internal Pattern Status	9F	Section 3.4.14
General Setup Commands			
Write	Write Border Color	B2	Section 3.5.1
Read	Read Border Color	B3	Section 3.5.2
Write	Write Keystone Projection Pitch Angle	BB	Section 3.5.3
Read	Read Keystone Projection Pitch Angle	BC	Section 3.5.4
Administrative Commands			
Read	Read Short Status	D0	Section 3.6.1
Read	Read System Status	D1	Section 3.6.2
Read	Read System Software Version	D2	Section 3.6.3
Read	Read Communication Status	D3	Section 3.6.4
Read	Read Controller Device ID	D4	Section 3.6.5
Read	Read DMD Device ID	D5	Section 3.6.6
Read	Read System Temperature	D6	Section 3.6.7
Read	Read Flash Build Version	D9	Section 3.6.8
Write	Write Flash Batch File Delay	DB	Section 3.6.9
Flash Update Commands			Section 3.7
Read	Read Flash Update PreCheck	DDh	Section 3.7.1
Write	Write Flash Data Type Select	DEh	Section 3.7.2
Write	Write Flash Data Length	DFh	Section 3.7.3
Write	Write Erase Flash Data	E0h	Section 3.7.4
Write	Write Flash Start	E1h	Section 3.7.5
Write	Write Flash Continue	E2h	Section 3.7.6

The following sections describe each of the above listed commands in detail.

3.1 General Operation Commands

3.1.1 Write Operating Mode Select (05h)

This command selects the operating mode of the system.

3.1.1.1 Write Parameters

Byte 1	Operating Mode
00h	Display - External Video Mode
01h	Display - Test Pattern Generator Mode
02h	Display - Splash Screen Mode
03h	Light Control – External Pattern Streaming Mode
04h	Light Control – Internal Pattern Streaming Mode
05h	Light Control – Splash Pattern Mode
06h - FEh	*Reserved*
FFh	Standby Mode

The Standby mode disables illumination power and sets the DMD in a 50-50 refresh duty cycle, where the mirrors are on 50% of the time and off during the remaining time. This 50-50 refresh state helps in prolonging the life of the DMD.

The other operating modes have associated commands which are only applicable to that mode and must be run to properly configure the selected mode. The associated commands are listed below:

- Display - External Video Port:
  - *Write External Input Image Size* - Section 3.1.25
  - *Write External Video Source Format Select* - Section 3.1.3
- Display - Test Pattern Generator:
  - *Write Test Pattern Select (0Bh)* - Section 3.1.5
- Display - Splash Screen:
  - *Write Splash Screen Select (0Dh)* - *Section 3.1.7*
  - *Write Splash Screen Execute (35h)* - Section 3.1.27
- When selecting *Light Control – External Pattern Streaming mode*, prior to setting the operating mode, the external source must be configured and locked, the pattern configuration defined using *Write Pattern Configuration* (Section 3.4.7) and the output trigger signals configured using *Write Trigger Out Configuration* (Section 3.4.3).
- When selecting *Light Control – Internal Pattern Streaming mode*, prior to setting the operating mode, the pattern configuration must be defined using *Write Pattern Configuration* (Section 3.4.7) , output trigger signals configured using *Write Trigger Out Configuration* (Section 3.4.3) and input trigger and pattern ready signals configured using *Write Trigger In Configuration* (Section 3.4.1) and *Write Pattern Ready Configuration* (Section 3.4.5).
- *Light Control – Splash Pattern mode* displays patterns from the flash images similar to splash screens but with image processing disabled. They are also subject to frame rate, illumination, duty cycle and trigger adjustments like other pattern modes. Prior to selecting Splash Pattern as the operating mode, the command *Write Splash Screen Select* (Section 3.1.7) must have been sent. The command*Write Splash Screen Execute* (Section 3.1.27) must be sent afterwards.

The table below show the source specific associated commands, where 'Y' represents a valid source selection and 'N' implies that the command is not supported by the selected source.

Table 3. Source Specific Associated Commands

Source Specific Associated Commands	Display Modes			Light Control Modes		
	External Video Port	Test Pattern Generator	Splash Screen	External Pattern Streaming	Internal Pattern Streaming	Splash
Write External Video Source Format Select	Y	N	N	Y	N	N
Write External Input Image Size	Y	N	N	Y	N	N
Write Test Pattern Select	N	Y	N	N	N	N
Write Splash Screen Select	N	N	Y	N	N	Y
Write Splash Screen Execute	N	N	Y(1)	N	N	Y(1)
Write Pattern Configuration	N	N	N	Y	Y	Y
Write Trigger Out Configuration	N	N	N	Y	Y	Y
Write Trigger In Configuration	N	N	N	N	Y	N
Write Pattern Ready Configuration	N	N	N	N	Y	N

(1) The Write Splash Screen Execute command is special in that there is no maintained state or history. Thus, this command has no "settings" to be stored and reused by the system.

It is recommended that the source associated commands be sent prior to sending the *Write Operating Mode Select* command. These commands (except for *Write Splash Screen Execute*) describe the unique characteristics of their associated source, and once these settings have been defined, they are stored in a volatile manner. When source associated commands are sent when that source is not active, the controller software saves the new settings, but does not execute these commands. When that source becomes active (via the *Write Operating Mode Select* command), the controller applies these settings. Each time an operating mode selection is made , the system retrieves the settings defined previously and automatically applies them. As such, the user only needs to send these associated commands when the source first needs to be defined, or when the source characteristics for that port need to be changed. It is important to note that the appropriate associated commands must be updated when source characteristics change. Refer to for examples to configure different operating modes.

The rest of the commands that apply to image setup are those commands whose settings are applicable across all source selections, and indeed, these command settings would typically remain the same across the different Operating Mode selections. Sometimes the values for these commands are the same across the different input source types, but this does not indicate that hardware settings have not changed. For example, if the display image size is set to 720p and the external port input source size is set to VGA, the input scales up to the display size of 720p. When splash screen is selected, the controller modifies the settings specified by the *Write Image Crop* command (Section 3.1.10) . The controller displays these settings at the resolution specified by the *Write Display Size* command (Section 3.1.12). Therefore, the scale settings must be changed. The controller software manages the underlying hardware settings.

Refer to *Write Image Freeze* (Section 3.1.18.1) for information on hiding on-screen artifacts when selecting an input source.

3.1.2 Read Operating Mode Select (06h)

This command reads the operating mode of the system.

3.1.2.1 Read Parameters

This command has no parameters.

3.1.2.2 Return Parameters

Byte 1	Operating Mode
00h	Display - External Video Mode
01h	Display - Test Pattern Generator Mode
02h	Display - Splash Screen Mode
03h	Light Control – External Pattern Streaming Mode
04h	Light Control – Internal Pattern Streaming Mode
05h	Light Control – Splash Pattern Mode
06h - FEh	*Reserved*
FFh	Standby Mode

This command works in all operating modes and has no effect on the current system configuration.

3.1.3 Write External Video Source Format Select (07h)

This command specifies the active external video port and the source data type.

3.1.3.1 Write Parameters

CMD Parameter	Port	Bits/Pixel	Data Type	Bus Width	Clks/Pixel	Notes
Parallel Port User Selection						
40h	Parallel	16	RGB565	16	1	Auto-select RGB CSC
41h	Parallel	18	RGB 666	18	1	Auto-select RGB CSC
42h	Parallel	24	RGB 888	8	3	Auto-select RGB CSC
43h	Parallel	24	RGB 888	24	1	Auto-select RGB CSC
50h	Parallel	18	YCbCr 666	18	1	Auto-select YCbCr CSC
51h	Parallel	24	YCbCr 888	24	1	Auto-select YCbCr CSC
60h	Parallel	16	YCbCr 4:2:2 88	8	2	Auto-select YCbCr CSC - 4:2:2 or 4:4:4
61h	Parallel	16	YCbCr 4:2:2 88	16	1	Auto-select YCbCr CSC - 4:2:2 or 4:4:4

Default: 43h

- This command is used in conjunction with the *Write Operating Mode Select* command (Section 3.1.1). This command specifies which input port is to be displayed when the *Write Operating Mode Select* command selects External Video Port as the image source. The controller retains the settings for this command until they are changed using this command. The controller automatically applies these settings each time the External Video Port is selected.
- When the external video port is selected as the input source, software will automatically select and load the proper CSC based on the selected parameter of this command (appropriate matrix for RGB, selected matrix for YCbCr including offset). It will also automatically select the appropriate data path for 4:2:2 vs. 4:4:4 processing.
- *This command is a source associated command. Please review the notes for the Write Operating Mode Select command (Section 3.1.1) to understand the concept of source associated commands, which determines when these commands are executed by the system.*

3.1.4 Read External Video Source Format Select (08h)

This command reads the state of the active external video port and the source data type for the display module.

3.1.4.1 Read Parameters

This command has no parameters.

3.1.4.2 Return Parameters

CMD Parameter	Port	Bits/Pixel	Data Type	Bus Width	Clks/Pixel	Notes
Parallel Port User Selection						
40h	Parallel	16	RGB565	16	1	Auto-select RGB CSC
41h	Parallel	18	RGB 666	18	1	Auto-select RGB CSC
42h	Parallel	24	RGB 888	8	3	Auto-select RGB CSC
43h	Parallel	24	RGB 888	24	1	Auto-select RGB CSC
50h	Parallel	18	YCbCr 666	18	1	Auto-select YCbCr CSC
51h	Parallel	24	YCbCr 888	24	1	Auto-select YCbCr CSC
60h	Parallel	16	YCbCr 4:2:2 88	8	2	Auto-select YCbCr CSC - 4:2:2 or 4:4:4
61h	Parallel	16	YCbCr 4:2:2 88	16	1	Auto-select YCbCr CSC - 4:2:2 or 4:4:4

3.1.5 Write Test Pattern Select (0Bh)

This command specifies an internal test pattern for display on the display module.

3.1.5.1 Write Parameters

Parameter Bytes	Description
Byte 1	Test Pattern Generator (TPG) pattern select
Byte 2	Foreground and background color (see Table 4)
Byte 3	Parameter 1 (see Table 5)
Byte 4	Parameter 2 (see Table 5)
Byte 5	Parameter 3 (see Table 5)
Byte 6	Parameter 4 (see Table 5)

Byte 1 **TPG pattern select**

b(7) Test pattern border:
- 00h: Disabled
- 01h: Enabled

b(6:4) Reserved

b(3:0) Pattern select:
- 00h: Solid field
- 01h: Fixed step horizontal ramp
- 02h: Fixed step vertical ramp
- 03h: Horizontal lines
- 04h: Diagonal lines
- 05h: Vertical lines
- 06h: Horizontal and vertical grid
- 07h: Checkerboard
- 08h: Color bars
- 09h-0Fh: Reserved

Byte 2 **Foreground color**

b(7) Reserved

b(6:4)
- 0h: Black
- 1h: Red
- 2h: Green
- 3h: Blue
- 4h: Cyan
- 5h: Magenta
- 6h: Yellow
- 7h: White

Background color

b(3:0) Reserved

b(2:0)
- 0h: Black
- 1h: Red
- 2h: Green
- 3h: Blue
- 4h: Cyan
- 5h: Magenta
- 6h: Yellow
- 7h: White

Table 4. Foreground and Background Color Use

Pattern	Byte 2	
	Foreground Color	Background Color
Solid field	Yes	No
Fixed step horizontal ramp	Yes	No
Fixed step vertical ramp	Yes	No
Horizontal lines	Yes	Yes
Vertical lines	Yes	Yes
Diagonal lines	Yes	Yes
Grid lines	Yes	Yes
Checkerboard	Yes	Yes
Color bars	No	No

Table 5. Descriptions and Bit Assignments for Parameters 1-4

Pattern	Byte 6 (Parameter 4)		Byte 5 (Parameter 3)		Byte 4 (Parameter 2)		Byte 3 (Parameter 1)	
	Description	Bits	Description	Bits	Description	Bits	Description	Bits
Solid field	N/A		N/A		N/A		N/A	
Fixed step horizontal ramp	N/A		N/A		Brightest pixel value	8	Darkest pixel value	8
Fixed step vertical ramp	N/A		N/A		Brightest pixel value	8	Darkest pixel value	8
Horizontal lines	N/A		N/A		Background line width	8	Foreground line width	8
Vertical lines	N/A		N/A		Background line width	8	Foreground line width	8
Diagonal lines	N/A		N/A		Vertical spacing	8	Horizontal spacing	8
Grid lines	Vertical background line width	8	Vertical foreground line width	8	Horizontal background line width	8	Horizontal foreground line width	8
Checkerboard	Number of vertical checkers (MSB)	3	Number of vertical checkers (LSB)	8	Number of horizontal checkers (MSB)	3	Number of horizontal checkers (LSB)	8
Color bars	N/A		N/A		N/A		N/A	

1. This command is used in conjunction with the *Write Operating Mode Select* command (Section 3.1.1). This command specifies which test pattern is to be displayed when the *Write Operating Mode Select* command selects Test Pattern Generator as the image source. The settings for this command are to be retained until changed using this command. The controller automatically applies these settings each time the Test Pattern Generator is selected.
2. Batch files (Section 3.1.24) can be created and stored in Flash and used to recall the settings for predefined test patterns.
3. The controller creates Test Patterns at the resolution of the display (DMD). Hhowever, the *Write Image Crop* command (Section 3.1.10 can modify them. The controller displays them at the resolution specified by the *Write Display Size* command (Section 3.1.12).
4. The Test Pattern border selection creates a single pixel wide and tall white border around the specified test pattern.
5. *It is important that the user review the notes for the Write Operating Mode Select command (Section 3.1.1) to understand the concept of source associated commands. This concept determines when source associated commands are executed by the system. Note that this command is a source associated command.*
6. When a Foreground or Background Color is not used, the controller ignores the corresponding bit values. The number of parameter bytes to be used depends on the selected pattern. Table 6 shows the number of bytes to be used based on the specified pattern.

Table 6. Number of Bytes Required based on Pattern Selection

Specified Pattern	Solid Field	Fixed Step Horz Ramp	Fixed Step Vert Ramp	Horz Lines	Vert Lines	Diag Lines	Grid Lines	Checker board	Color Bars
Number of Bytes Required	2	4	4	4	4	4	6	6	1

7. As noted in Table 4, the color for the Solid Field pattern is specified using the Foreground color. An example of a Solid Field pattern is shown in Figure 4.

Figure 4. Example of Solid Field Test Pattern (Red)

8. As noted in Table 4, the color for the fixed step horizontal ramp pattern is specified using the foreground color. As noted in Table 5, the user specifies the start value and the stop value for the ramp. For this pattern, the system automatically determines the step size based on the start and stop values and the size of the display (DMD). The minimum start value is 0, the maximum stop value is 255, and the start value must always be smaller than the stop value. For example, if the start value = 0, the stop value = 255, and the DMD resolution is 1280 wide, the step size would be 5 (1280 pixels / 256 values = 5). Thus every gray shade value from 0 to 255 would have a step size of 5 pixels (such that each step would have 5 columns of pixels with the same gray scale value). The gray scale value always increments by 1 for each step between the start and stop values. An example of a fixed step horizontal ramp pattern is shown in Figure 5.

Figure 5. Example of Fixed Step Horizontal Ramp Test Pattern

9. As noted in Table 4, the color for the fixed step vertical ramp pattern is specified using the foreground color. As noted in Table 5, the user specifies the start value and the stop value for the ramp. For this pattern, the system automatically determines the step size based on the start and stop values and the size of the display (DMD). The minimum start value = 0, the maximum stop value = 255, and the start value must always be smaller than the stop value. For example, if the start value = 0, the stop value = 255, and the DMD resolution is 768 tall, then the step size would be 3 (768 pixels / 256 values = 3). Thus every value from 0 to 255 would have a step size of 3 pixels (such that each step would have 3 rows of pixels with the same gray scale value). The gray scale value always increments by 1 for each step between the start and stop values. An example of a fixed step vertical ramp pattern is shown in Figure 6.

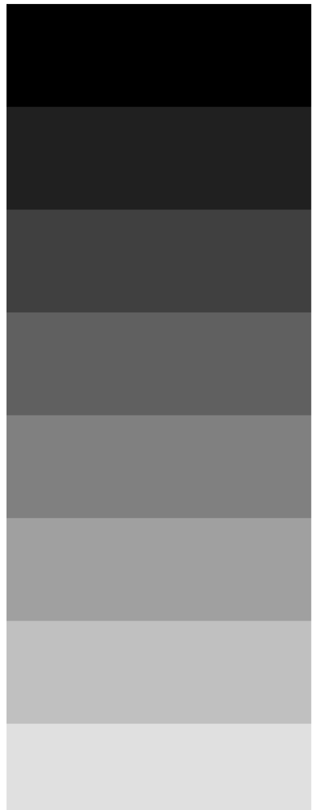


Figure 6. Example of Fixed Step Vertical Ramp Test Pattern

10. As noted in Table 4, the colors for the horizontal lines pattern are specified using both the foreground and background colors. The foreground color is used for the horizontal lines, and the background color is used for the space between the lines. As noted in Table 5, the user specifies the foreground line width, as well as the background line width. The user must determine the line spacing for each resolution display. For example, if the foreground line width = 1, and the background line width = 9, there would be a single pixel horizontal line on every tenth line. An example of a horizontal lines pattern is shown in Figure 7.

Figure 7. Example of Horizontal Lines Test Pattern

11. As noted in Table 4, the colors for the vertical lines pattern are specified using both the foreground and background colors. The foreground color is used for the vertical lines, and the background color is used for the space between the lines. As noted in Table 5, the user specifies the foreground line width, as well as the background line width. The user must determine the line spacing for each resolution display. For example, if the foreground line width = 1, and the background line width = 9, there would be a single pixel vertical line on every tenth line. An example of a vertical lines pattern is shown in Figure 8.

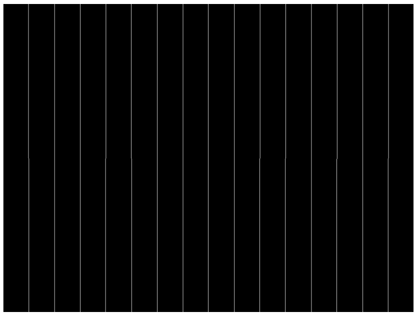


Figure 8. Example of Vertical Lines Test Pattern

12. As noted in Table 4, the colors for the diagonal lines pattern are specified using both the foreground and background colors. The foreground color is used for the diagonal lines, and the background color is used for the space between the lines. As noted in Table 5, the user specifies the horizontal and vertical line spacing. The line width is always one pixel. The user determines the line spacing for each resolution display. Both horizontal and vertical line spacing must use the same value, and are limited to values of 3, 7, 15, 31, 63, 127, and 255. Invalid values result in a communication error (invalid command parameter). An example of a diagonal lines pattern is shown in Figure 9.

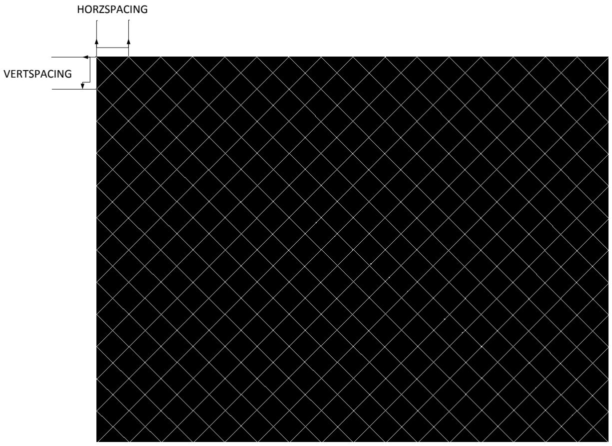


Figure 9. Example of Diagonal Lines Test Pattern

13. As noted in Table 4, the colors for the grid lines pattern are specified using both the foreground and background colors. The foreground color is used for the grid lines, and the background color is used for the space between the lines. As noted in Table 5, the user specifies the horizontal foreground and background line width, as well as the vertical foreground and background line width. The user determines the line spacing for each resolution display. For example, if the horizontal foreground line width = 1, and background line width = 9, there would be a single pixel horizontal line on every tenth line. If the vertical foreground line width = 1, and background line width = 9, there would be a single pixel vertical line on every tenth line. An example of a grid lines pattern is shown in Figure 10.

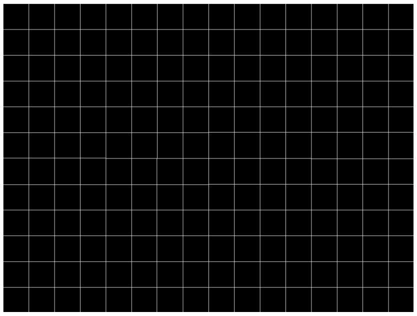


Figure 10. Example of Grid Lines Test Pattern

14. As noted in Table 4, the colors for the checkerboard pattern are specified using both the foreground and background colors. The foreground color is used for one of the checkers, and the background color is used for the alternating checker. As noted in Table 5, the user specifies the number of horizontal checkers and the number of vertical checkers. For this pattern, the system automatically determines the checker size in each direction based on the number of checkers and the size of the display (DMD). For example, if the number of horizontal checkers = 4, the number of vertical checkers = 4, and the DMD resolution is 1280x720, the size of the horizontal checkers is 320 pixels, and the size of the vertical checkers is 180 pixels (1280 pixels / 4 checkers = 320 pixels: 720 pixels / 4 checkers = 180 pixels). An example of a checkerboard pattern (16 checkers by 12 checkers) is shown in Figure 11.

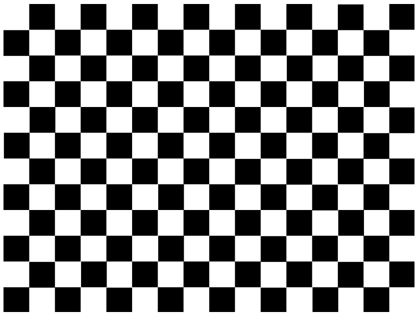


Figure 11. Example of Checkerboard Test Pattern

15. As noted in Table 4 and Table 5, there is no user programmability associated the color bars test pattern. This pattern is made up of eight vertical color bars: white, yellow, cyan, green, magenta, red, blue, and black. For this pattern, the system automatically determines the width for each color bar based on the size of the display (DMD). An example of the color bars pattern is shown in Figure 12.

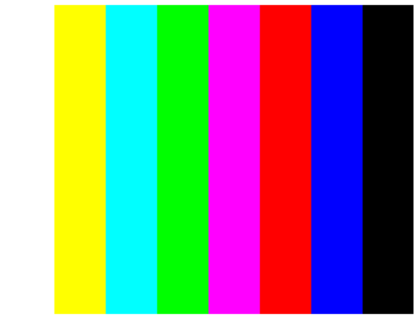


Figure 12. Example of Color Bars Test Pattern

3.1.6 Read Test Pattern Select (0Ch)

This command reads the state of the test pattern selected for the display module.

3.1.6.1 Read Parameters

This command has no parameters.

3.1.6.2 Return Parameters

Parameter Bytes	Description
Byte 1	TPG pattern select
Byte 2	Foreground and background color (see Table 4)
Byte 3	Parameter 1 (see Table 5)
Byte 4	Parameter 2 (see Table 5)
Byte 5	Parameter 3 (see Table 5)
Byte 6	Parameter 4 (see Table 5)

This command always returns six bytes as the host does not know how many bytes are valid until the pattern is selected. All unnecessary bytes (see Table 6) are set to '0'.

3.1.7 Write Splash Screen Select (0Dh)

3.1.7.1 Write

This command selects a stored splash screen to be displayed.

3.1.7.2 Write Parameters (0Dh)

Parameter Bytes	Description
Byte 1	Splash screen reference number (integer)

1. This command is used in conjunction with the *Write Operating Mode Select* (Section 3.1.1) and the *Write Splash Screen Execute* (Section 3.1.27) commands. It specifies which splash screen is to be displayed when the *Input Source Select* command selects splash screen as the image source. The controller retains the settings for this command until changed using this command.
2. The steps required to display a splash screen are: select the desired splash screen (this command), change the input source to splash screen (using *Write Operating Mode Select*), and start the splash screen retrieval process (using *Write Splash Screen Execute*).
3. The Splash Screen is a unique source since it is read from Flash and sent down the processing path of the controller one time, to be stored in memory for display at the end of the processing path. As such, the user must set *all image processing settings (e.g. image crop, image orientation, display size, splash screen select, look select, splash screen as input source) before executing the Write Splash Screen Execute command.*
4. *It is important that the user review the notes for the Write Operating Mode Select command (Section 3.1.1) to understand the concept of source associated commands. This concept determines when source associated commands are executed by the system. Note that this command is a source associated command.*
5. The availability of splash screens is limited by the available space in flash memory.
6. All splash screens must be *landscape* oriented.
7. For single controller applications which support DMD resolutions up to 1280 x 720, the minimum splash image size allowed for flash storage is 427 x 240, with the maximum being the resolution of the DMD. Typical splash image sizes for flash are 427 x 240 and 640 x 360. The full resolution size is typically used to support an “Optical Test” splash screen.
8. The user is responsible for specifying how the splash image displays on the screen. Key commands for this are *Write Image Crop* (Section 3.1.10) and *Write Display Size* (Section 3.1.12).
9. When this command is received while Splash Screen is the active source, other than storing the specified splash screen value, the only action of the controller software is to obtain the header information from the selected splash screen and store this in internal memory. Then, when the *Write Splash Screen Execute* command is received, the controller software uses this stored information to set up the processing path prior to pulling the splash data from flash.

3.1.8 Read Splash Screen Select (0Eh)

This command reads the last selected splash screen index.

3.1.8.1 Read Parameters

This command has no parameters.

3.1.8.2 Return Parameters

Parameter Bytes	Description
Byte 1	Splash screen index

3.1.9 Read Splash Screen Header (0Fh)

This command reads the splash screen header information for the selected splash screen.

3.1.9.1 Read Parameters

Parameter Bytes	Description
Byte 1	Splash screen reference number (integer)

The read parameter is used to specify the splash screen for which the header parameters are to be returned. If a splash screen value is provided for which there is no splash screen available, the controller issues an error (invalid command parameter value – communication status) and the command does not execute.

3.1.9.2 Return Parameters

Parameter Bytes	Description
Byte 1	Splash image width in pixels (LSByte)
Byte 2	Splash image width in pixels (MSByte)
Byte 3	Splash image height in pixels (LSByte)
Byte 4	Splash image height in pixels (MSByte)
Byte 5	Splash image size in bytes (LSByte)
Byte 6	Splash image size in bytes
Byte 7	Splash image size in bytes
Byte 8	Splash image size in bytes (MSByte)
Byte 9	Pixel format
Byte 10	Compression type
Byte 11	Color order
Byte 12	Chroma order
Byte 13	Byte order

Parameter definitions referenced are in Table 7.

Table 7. Splash Screen Header Definitions

Parameter	Values
Pixel format	'0h' = 24-bit RGB unpacked (not used) '1h' = 24-bit RGB packed (not used) '2h' = 16-bit RGB 5-6-5 '3h' = 16-bit YCbCr 4:2:2
Compression type	'0h' = Uncompressed '1h' = RGB RLE compressed '2h' = User-defined (not used) '3h' = YUV RLE compressed
Color order	'0h' = 00RRGGBB '1h' = 00GGRRBB
Chroma order	'0h' = Cr is first pixel '1h' = Cb is first pixel
Byte order	'0h' = Little endian '1h' = Big endian

3.1.10 Write Image Crop (10h)

This command specifies which portion of the input image is captured and output from the cropping function of the display module.

3.1.10.1 Write Parameters

Parameter Bytes	Description
Byte 1	Capture start pixel (LSByte)
Byte 2	Capture start pixel (MSByte)
Byte 3	Capture start line (LSByte)
Byte 4	Capture start line (MSByte)
Byte 5	Pixels per line (LSByte)
Byte 6	Pixels per line (MSByte)
Byte 7	Lines per frame (LSByte)
Byte 8	Lines per frame (MSByte)

1. This command applies to all sources including test patterns, splash screens, and external sources. Making a change to the source or port does not impact the application of this command.
2. The Capture Start parameters for this command are referenced to active data, and are '0' based (i.e. specifying the capture start pixel to be a value of zero would specify the 1^{st} active pixel of a line). The Pixel/Line and Lines/Frame parameters are '1' based (i.e. specifying the pixels/line value to be a value of 640 would specify 640 pixels to be captured).
3. Cropping is done prior to the scaling function in the display module. As such, the size difference between the crop size and display size (Section 3.1.12) in effect determine the amount of scaling to be done in both dimensions.
4. The required scaling may not be achievable depending on other factors such as keystone correction. In this case, the system performs the request even if this results in a broken image. The user is ultimately responsible for providing the appropriate input settings to meet their display needs.
5. If a crop size parameter exceeds the size of the input image, the controller uses the input image size minus the Capture Start Pixel/Line (as appropriate) (as shown in Figure 13). Regardless, the crop size parameters returned by the Read Image Crop command always equal the values specified by the Write Image Crop command.

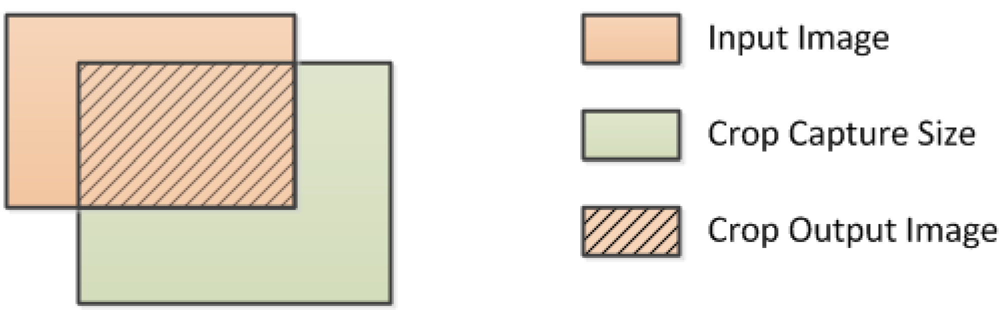


Figure 13. Cropping Rules when Crop Size exceeds Input Size

3.1.11 Read Image Crop (11h)

This command reads the state of the image crop setting.

3.1.11.1 Read Parameters

This command has no parameters.

3.1.11.2 Return Parameters

Parameter Bytes	Description
Byte 1	Capture start pixel (LSByte)
Byte 2	Capture start pixel (MSByte)
Byte 3	Capture start line (LSByte)
Byte 4	Capture start line (MSByte)
Byte 5	Pixels per line (LSByte)
Byte 6	Pixels per line (MSByte)
Byte 7	Lines per frame (LSByte)
Byte 8	Lines per frame (MSByte)

All parameters for this command are referenced to active data, and are '1' based. (i.e. specifying the capture start pixel to be a value of one would specify the 1^{st} active pixel of a line).

3.1.12 Write Display Size (12h)

This command specifies the size of the active image to be displayed on the display module.

3.1.12.1 Write Parameters

Parameter Bytes	Description
Byte 1	Start Pixel (LSByte)
Byte 2	Start Pixel (MSByte)
Byte 3	Start Line (LSByte)
Byte 4	Start Line (MSByte)
Byte 5	Pixels per line (LSByte)
Byte 6	Pixels per line (MSByte)
Byte 7	Lines per frame (LSByte)
Byte 8	Lines per frame (MSByte)

Default: DMD resolution.

1. This command specifies the size of the non-keystone corrected image to be output from the scaler function, which equals size of the active displayed image. Start pixel and Start line are default to 0.
2. The parameter values are to be ‘1’ based. (i.e. a value of 1280 pixels displays 1280 pixels per line).
3. All sub-images (images smaller than the DMD display) can start on any active area of the DMD. There is no sub image rotation and keystone support. The subimage display range is

   Horizontal range: DMD width – Subimage Pixels per line

   Workable x coordinate is 0 to (DMD width – Subimage Pixels per line – 1)

   Vertical range: (DMD height – Subimage Lines per frame)

   Workable y coordinate is 0 to (DMD height – Subimage Lines per frame -1)

   Example: If the display subimage us 600x600 and DMD size is 1280 × 720.

   Horizontal range: 1280 – 600 = 680, the workable x coordinate is 0 to 679.

   Vertical range: 720-600 = 120, the workable y coordinate is 0 to 119.
4. If the display size exceeds the resolution of the DMD, the controller issues an error (invalid command parameter value – communication status) and the command does not execute. Specifically, the controller compares the display size parameters to the DMD resolution in both rotation image orientations (non-rotated and rotated), and if the DMD resolution is exceeded in both of these orientations, the controller issues an error. Note that the system does not check for proper image orientation setup.

   DMD resolution = 854 × 480:
   - Example 1: Display size parameter = 480 × 854 (not an error)
   - Example 2: Display size parameter = 900 × 320 (error)
   - Example 3: Display size parameter = 500 × 600 (error)
5. If the source, crop, and display parameter combinations exceed the capabilities of the scaler, the system tries to implement what is requested by the user, and a *broken* image may be displayed. The user must provide updated parameters to correct the image.

3.1.13 Read Display Size (13h)

This command reads the specified display size.

3.1.13.1 Read Parameters

This command has no parameters.

3.1.13.2 Return Parameters

Parameter Bytes	Description
Byte 1	Start Pixel (LSByte)
Byte 2	Start Pixel (MSByte)
Byte 3	Start Line (LSByte)
Byte 4	Start Line (MSByte)
Byte 5	Pixels per line (LSByte)
Byte 6	Pixels per line (MSByte)
Byte 7	Lines per frame (LSByte)
Byte 8	Lines per frame (MSByte)

The parameter values are to be ‘1’ based. (i.e. a value of 1280 pixels displays 1280 pixels per line).

3.1.14 Write Display Image Orientation (14h)

This command specifies the image orientation of the displayed image.

3.1.14.1 Write Parameters

Parameter Byte	
b(7:3)	*Reserved*
b(2)	Short axis image flip: • 0: Image not flipped. • 1: Image flipped.
b(1)	Long axis image flip: • 0: Image not flipped. • 1: Image flipped.
b(0)	Image rotation (for portrait source only): • 0: No rotation • 1: Minus 90° rotation

Figure 14. Short-Axis Flip

Figure 15. Long-Axis Flip

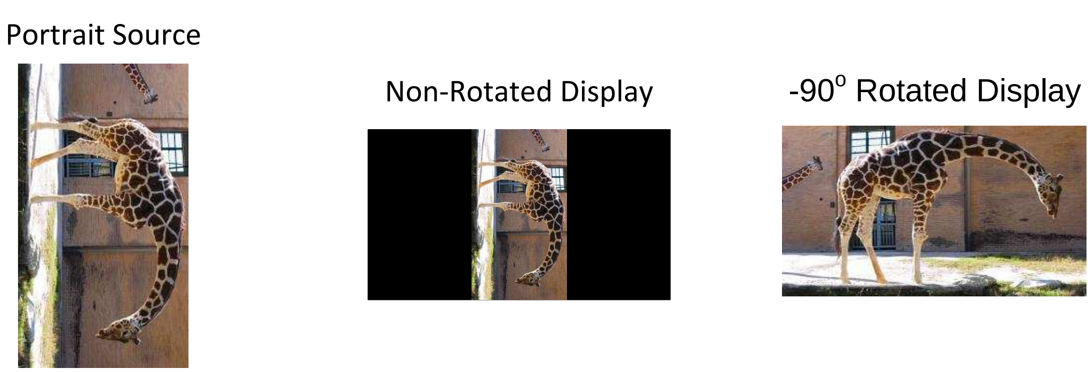


Figure 16. Rotation and Non-Rotation of Portrait Source

3.1.15 Read Display Image Orientation (15h)

This command reads the specified displayed image orientation.

3.1.15.1 Read Parameters

This command has no parameters.

3.1.15.2 Return Parameters

Parameter Byte	
b(7:3)	*Reserved*
b(2)	Short axis image flip: • 0: Image not flipped. • 1: Image flipped.
b(1)	Long axis image flip: • 0: Image not flipped. • 1: Image flipped.
b(0)	Image rotation (for portrait source only): • 0: No rotation • 1: Minus 90° rotation

3.1.16 Write Display Image Curtain (16h)

This command fills the entire display with a user-specified color.

3.1.16.1 Write Parameters

Parameter Byte	
b(7:4)	*Reserved*
b(3:1)	Select curtain color: • 0h: Black • 1h: Red • 2h: Green • 3h: Blue • 4h: Cyan • 5h: Magenta • 6h: Yellow • 7h: White
b(0)	Curtain enable: • '0': Curtain disabled • '1': Curtain enabled

- The Image Curtain fills the entire display with a user specified color.
- The curtain color specified by this command is separate from the border color defined in the *Write Border Color* command (Section 3.5.1), even though both are displayed using the curtain capability.

3.1.17 Read Display Image Curtain (17h)

This command reads the state of the image curtain control function.

3.1.17.1 Read Parameters

This command has no parameters.

3.1.17.2 Return Parameters

Parameter Byte	
b(7:4)	*Reserved*
b(3:1)	Select curtain color: • 0h: Black • 1h: Red • 2h: Green • 3h: Blue • 4h: Cyan • 5h: Magenta • 6h: Yellow • 7h: White
b(0)	Curtain enable: • '0': Curtain disabled • '1': Curtain enabled

3.1.18 Write Image Freeze (1Ah)

This command enables or disables the image freeze function.

3.1.18.1 Write Parameters

Parameter Byte	
b(7:1)	*Reserved*
b(0)	Image freeze: • '0': Image freeze disabled • '1': Image freeze enabled

Default: 00h

1. Normal use of the Image Freeze capability typically has two main functions. The first function is to allow the end user to freeze the current image on the screen for their own uses. The second function is to allow the user to reduce/prevent system changes from showing up on the display as visual artifacts. In this second case, the image would be frozen, system changes would be made, and when complete, the image is unfrozen. In all cases, when the image is unfrozen, the display starts showing the most resent input image. Thus input data between the freeze point and the unfreeze point is lost. Suggestions to the host system for the types of image changes likely to necessitate the use of the image freeze command to hide artifacts are discussed in Section 3.1.18.2.
2. The controller software never (either *automatically* nor *under-the-hood*) freezes or unfreezes the image. This applies when software is making updates to the system on its own volition, as well as for any operation commanded via I^{2}C. The controller software does not freeze or unfreeze the image for any reason except when explicitly commanded by the *Write Image Freeze* command.
3. If the user chooses not to make use of Image Freeze, is recommended that they change the source itself before changing image parameters to minimize transition artifacts.

3.1.18.2 Use of Image Freeze to Reduce On-Screen Artifacts

Commands that take a long time to process, require a lot a data to be loaded from flash, or change the frame timing of the system may create on-screen artifacts. The *Write Image Freeze* command can try and minimize, if not eliminate, these artifacts. The process is:

1. Send a *Write Image Freeze* command to enable freeze.
2. Send commands with the potential to create image artifacts.
3. Send a *Write Image Freeze* command to disable freeze.

As the system processes commands to the controller serially, it requires no special timing or delay between these commands. Make sure that the number of commands placed between the freeze and unfreeze is small, as it is likely not desirable for the image to be frozen for a "long" period of time. A list of commands that may produce image artifacts is listed in Table 8. This is not an all-inclusive list, however, and the user is responsible for determining if and when use of the image freeze command meets their product needs.

Table 8. Partial List of Commands that May Benefit from the Use of Image Freeze

Command	Command OpCode
Write Input Source Select	05h
Write External Video Source Format Select[(1)]	07h
Write Test Pattern Select[(1)]	0Bh
Write Splash Screen Select[(1)]	0Dh
Write Look Select	22h

(1) If changed while this source is the active source

Table 9 and Table 10 show a few examples of how to use the image freeze command.

Table 9. Test Pattern Generator Example Using Image Freeze

Command	Notes
Write Image Freeze = Freeze	
Write Image Crop, Write Display Size, Write Display Image Orientation, Write Test Pattern Select, Keystone Correction Control Enable, Keystone angle.	Potential data processing commands that may be required for proper display of test pattern image. These would be used as appropriate. It is recommended that these be set before the Write Operating Mode Select command.
Write Operating Mode Select = Test Pattern Generator	
Write Image Freeze = Unfreeze	

Table 10. Splash Screen Example Using Image Freeze

Command	Notes
Write Display Image Curtain = enable	May want to apply curtain if already displaying an unwanted image (i.e. a broken source)
Write Image Freeze = freeze	
Write Image Crop, Write Display Size, Write Display Image Orientation, Write Look Select	Potential data processing commands that may be required for proper display of Splash Image. These would be used as appropriate. These must be set prior to Write Splash Screen Execute command to affect the splash screen image.
Write Splash Screen Select Write Input Source Select = Splash	These must be set prior to Write Splash Screen Execute
Write Splash Screen Execute	Retrieves the desired splash screen image for display
Write Image Freeze = unfreeze	

The controller displays a new splash image when it executes the *Write Splash Screen Execute* command, regardless of the state of the *Write Image Freeze* function (due to the one time nature of the Splash image). *Write Image Freeze* = Unfreeze must still be executed.

3.1.19 Read Image Freeze (1Bh)

This command reads the state of the image freeze function.

3.1.19.1 Read Parameters

This command has no parameters.

3.1.19.2 Return Parameters

Parameter Byte	
b(7:1)	*Reserved*
b(0)	Image freeze: • '0': Image freeze disabled • '1': Image freeze enabled

3.1.20 Write Look Select (22h)

This command specifies the Look for the image on the display module.

3.1.20.1 Write Parameters

Parameter Byte	Description
Byte 1	Look number

Default: Firmware defined

- A Look typically specifies a target white point.
- This command allows the host to select a Look (target white point) from a number of looks stored in flash. Based on the look selected, along with other parameters, software automatically selects and loads the most appropriate sequence and duty cycle set available in the Look to get as close as possible to the target white point.
- Looks are specified in this byte by an enumerated value (i.e. 0,1,2,3 etc.).
- There must always be at least one look, the enumerated value of which is ‘0’.
- The number of Looks available may be limited by the available space in flash memory.

3.1.21 Read Look Select (23h)

This command returns the currently selected Look parameters.

3.1.21.1 Read Parameters

This command has no parameters.

3.1.21.2 Return Parameters

Parameter Bytes	Description
Byte 1	Look number
Byte 2	Sequence number
Byte 3	Current sequence frame rate (LSB)
Byte 4	Current sequence frame rate
Byte 5	Current sequence frame rate
Byte 6	Current sequence frame rate (MSB)

Refer to Section 3.1.20 and Section 3.1.22 to understand the concepts of Looks and sequences respectively.

3.1.22 Read Sequence Header Attributes (26h)

This command reads the header information of the active sequence.

3.1.22.1 Read Parameters

This command has no parameters.

3.1.22.2 Return Parameters

Parameter Bytes	Description		Flash Structure
Byte 1	Red duty cycle	LSByte	Look Structure
Byte 2	Red duty cycle	MSByte	Look Structure
Byte 3	Green duty cycle	LSByte	Look Structure
Byte 4	Green duty cycle	MSByte	Look Structure
Byte 5	Blue duty cycle	LSByte	Look Structure
Byte 6	Blue duty cycle	MSByte	Look Structure
Byte 7	Maximum frame count	LSByte	Look Structure
Byte 8	Maximum frame count		Look Structure
Byte 9	Maximum frame count		Look Structure
Byte 10	Maximum frame count	MSByte	Look Structure
Byte 11	Minimum frame count	LSByte	Look Structure
Byte 12	Minimum frame count		Look Structure
Byte 13	Minimum frame count		Look Structure
Byte 14	Minimum frame count	MSByte	Look Structure
Byte 15	Max number of sequence vectors	b(3:0)	Look Structure
Byte 16	Red duty cycle	LSByte	Sequence Structure
Byte 17	Red duty cycle	MSByte	Sequence Structure
Byte 18	Green duty cycle	LSByte	Sequence Structure
Byte 19	Green duty cycle	MSByte	Sequence Structure
Byte 20	Blue duty cycle	LSByte	Sequence Structure
Byte 21	Blue duty cycle	MSByte	Sequence Structure
Byte 22	Maximum frame count	LSByte	Sequence Structure
Byte 23	Maximum frame count		Sequence Structure
Byte 24	Maximum frame count		Sequence Structure
Byte 25	Maximum frame count	MSByte	Sequence Structure
Byte 26	Minimum frame count	LSByte	Sequence Structure
Byte 27	Minimum frame count		Sequence Structure
Byte 28	Minimum frame count		Sequence Structure
Byte 29	Minimum frame count	MSByte	Sequence Structure
Byte 30	Max number of sequence vectors	b(3:0)	Sequence Structure

Table 11. Bit Weight and Bit Order for Duty Cycle Data

MSB	Byte 2						LSB	*MSB*	Byte 1						LSB
b15 2^{7}	b14 2^{6}	b13 2^{5}	b12 2^{4}	b11 2^{3}	b10 2^{2}	b9 2^{1}	b8 2^{0}	b7 2^{-1}	b6 2^{-2}	b5 2^{-3}	b4 2^{-4}	b3 2^{-5}	b2 2^{-6}	b1 2^{-7}	b0 2^{-8}

1. A sequence is a set of instructions that control how data is loaded and displayed on the DMD device.
2. The system stores sequence header data in two separate Flash data structures (the Look Structure and the Sequence Structure), and the values from each structure match unless an error has occurred.
3. The duty cycle data is specified as each colors percent of the frame time. The sum of the three duty cycles must add up to 100 (ex. R = 30.5 = 1E80h , G = 50 = 3200h, B = 19.5 = 1380h).
4. Sequence frame counts are specified in units of 66.67ns (based on the internal 15MHz clock used to time between input frame syncs). These are specified in this way to enable software to make fast and simple comparisons with the frame count.

3.1.23 Read DMD Sequencer Sync Mode (2Ch)

This command reads the state of the DMD sequencer sync mode function.

3.1.23.1 Read Parameters

This command has no parameters.

3.1.23.2 Return Parameters

Byte 1	
b(7:2)	*Reserved*
b(1)	System Auto-Sync Setting 0h: Lock to External VSYNC (Auto-Sync) 1h: Lock to Internal VSYNC (Auto-Sync)
b(0)	DMD Sequencer Sync Mode 0h: Auto-Sync 1h: Force Lock to Internal VSYNC

- System Auto-Sync Setting response is only valid when the DMD Sequencer Sync Mode is set to Auto-Sync (otherwise set to ‘0’). The Lock to External VSYNC (Auto-Sync) option indicates that the system is using the externally provided VSYNC to drive the display module. The Lock to Internal VSYNC option indicates that the system is using the internal VSYNC generator to drive the display module.

3.1.24 Write Execute Flash Batch File (2Dh)

This command executes a batch file stored in flash.

3.1.24.1 *Write Parameters*

Parameter Bytes	Description
Byte 1	Batch File Number

1. Most system *Write* commands specified in this document that can be sent by itself can be grouped together with other system commands or command parameters into a Flash batch file. See Table 12 for a list of commands which cannot be used in batch files.
2. One example for a Flash batch file might be the commands and command parameters required for initialization of the system after power-up.
3. The Flash batch file numbers to be specified in this byte are enumerated values (0,1,2,3...).
4. Flash batch file 0 is a special Auto-Init batch file that is run automatically by the DLPC347X software immediately after system initialization has been completed. The controller does not typically call the Flash batch file 0 using the *Write Execute Flash Batch File* command (although the system does allow it). This special Flash batch file typically specifies the default operating mode the system initializes to.
5. Embedding Flash batch file calls within a Flash batch file is not allowed (i.e. calling another batch file from within a batch file is not allowed). Multiple batch files can be executed consecutively by sending multiple execute batch file commands.
6. The system provides the ability to add an execution delay between commands within a Flash batch file. This is done using the *Write Flash Batch File Delay (DBh)* command (See Section 3.6.9).
7. The order of command execution for commands within a Flash batch file is the same as if the commands had been received over the I^2C port.

Table 12. List of Commands Excluded from Batch File Use

Command	Op-Code
Write Execute Flash Batch File	2D
Write Flash Data Type Select	DE
Write Flash Data Length	DF
Write Erase Flash Data	E0
Write Flash Start	E1
Write Flash Continue	E2
All *Read* commands	Various

3.1.25 Write Input Image Size (2Eh)

This command specifies the active data size of the input image.

3.1.25.1 Write Parameters

Parameter Bytes	Description
Byte 1	Pixels per line (LSByte)
Byte 2	Pixels per line (MSByte)
Byte 3	Lines per frame (LSByte)
Byte 4	Lines per frame (MSByte)

Default: DMD resolution

1. This command is used in conjunction with the *Write Operating Mode Select* command. This command specifies the active data size of the input image to the system for all external video interfaces when the *Write Operating Mode Select* command selects External Video Port as the image source. The settings for this command are to be retained until changed using this command. The controller automatically applies these settings each time the External Video Port is selected.
2. The parameter values are to be '1' based. (a value of 1280 pixels specifies 1280 pixels per line).
3. *It is important that the user review the notes for the Write Operating Mode Select command (Section 3.1.1) to understand the concept of source associated commands. This concept determines when source associated commands are executed by the system. Note that this command is a source associated command.*
4. The maximum and minimum input values are shown in Table 13. The controller flags values outside of these ranges as an error (invalid command parameter), and does not execute the command.

Table 13. Input Source Limits for Active Data

Parameter	Minimum Value	Maximum Value
Input source active pixels per line	320	1280
Input source active lines per frame	200	800

3.1.26 Read Input Image Size (2Fh)

This command reads the specified data size of the external input image.

3.1.26.1 Read Parameters

This command has no parameters.

3.1.26.2 Return Parameters

Parameter Bytes	Description
Byte 1	Pixels per line (LSByte)
Byte 2	Pixels per line (MSByte)
Byte 3	Lines per frame (LSByte)
Byte 4	Lines per frame (MSByte)

1. The parameter values are to be ‘1’ based. (a value of 1280 pixels specifies 1280 pixels per line).
2. This command returns the value specified by the Write External Input Image Size command (Section 3.1.25).

3.1.27 Write Splash Screen Execute (35h)

This command starts the process of retrieving a splash screen from flash for display.

3.1.27.1 Write Parameters

This command has no parameters.

Some important points to be noted about this command:

1. This command is used in conjunction with the Write Operating Mode Select (Section 3.1.1) and the Write Splash Screen Select (Section 3.1.8) commands. It is used to start the process of retrieving a splash screen from Flash for display.
2. The Splash Screen is a unique source as it is read from Flash and sent down the processing path of the controller one time, to be stored in memory for display at the end of the processing path. Set *all image processing settings* (image size, image crop, image orientation, display size, splash screen select, look select, splash screen as input source) *before executing this command*. Any data path processing changed after the splash screen has been executed requires this command to be re-executed before the controller displays the result. This way, the controller repeats the splash screen retrieval process each time it receives this command. See also the Write Image Freeze command (Section 3.1.18) for information on hiding on-screen artifacts when selecting and retrieving a splash image.
3. The process of retrieving the splash screen from SPI Flash can take a significant amount of time depending on the size of the compressed image stored in flash. During this period, the controller will not accept any new I2C commands. The user must ensure that the splash screen has been successfully displayed before sending any further commands.
4. When this command is processed in Display - Splash Screen mode, the system automatically initializes the system color processing based on the splash header information prior to sending the splash image down the data path. However, in Light Control - Splash Pattern mode, no color processing is performed and the stored bitmap is displayed as is, irrespective of whether the image is stored in RGB565 format or YCrCb (16-bit) format. Therefore, to get an accurate representation of the input image in splash pattern mode, the images must be stored in RGB565 format in flash.
5. *It is important that the user review the notes for the Write Operating Mode Select command (Section 3.1.1) to understand the concept of source associated commands. This concept determines when source associated commands are executed by the system. Note that this command is a source associated command; however, this command is special in that there is no maintained state or history. Thus, this command has no "settings" to be stored or reused by the system.*

3.2 *Illumination Control Commands*

3.2.1 Write LED Output Control Method (50h)

This command specifies the method for controlling the LED outputs for the display module.

3.2.1.1 Write Parameters

Byte 1	
b(7:2)	Reserved
b(1:0)	LED control method: • 00: Manual RGB LED PWM (CAIC algorithm disabled) • 01: CAIC (automatic) RGB LED PWM control (CAIC algorithm enabled) • 10: *Reserved* • 11: *Reserved*

Default: Firmware specified

- The Manual RGB LED PWM method provides for manual control of the LED PWM parameters, and disables the CAIC algorithm.
- The CAIC (Automatic) RGB LED PWM Control method provides automatic control of the LED PWM parameters using the CAIC algorithm.

3.2.2 Read LED Output Control Method (51h)

This command reads the selected LED output control method.

3.2.2.1 Read Parameters

This command has no parameters.

3.2.2.2 Return Parameters

Byte 1	
b(7:2)	Reserved
b(1:0)	LED control method: • 00: Manual RGB LED PWM (CAIC algorithm disabled) • 01: CAIC (automatic) RGB LED PWM control (CAIC algorithm enabled) • 10: *Reserved* • 11: *Reserved*

3.2.3 Write RGB LED Enable (52h)

This command enables the LEDs.

3.2.3.1 Write Parameters

Byte 1	
b(7:3)	*Reserved*
b(2)	Blue LED enable: • 0: Blue LED disabled • 1: Blue LED enabled
b(1)	Green LED enable: • 0: Green LED disabled • 1: Green LED enabled
b(0)	Red LED enable: • 0: Red LED disabled • 1: Red LED enabled

Default: Firmware specified

3.2.4 Read RGB LED Enable (53h)

This command reads the state of the LED enables.

3.2.4.1 Read Parameters

This command has no parameters.

3.2.4.2 Return Parameters

Byte 1	
b(7:3)	*Reserved*
b(2)	Blue LED enable: • 0: Blue LED disabled • 1: Blue LED enabled
b(1)	Green LED enable: • 0: Green LED disabled • 1: Green LED enabled
b(0)	Red LED enable: • 0: Red LED disabled • 1: Red LED enabled

3.2.5 Write RGB LED Current PWM (54h)

This command sets the PWM values for the red, green, and blue LEDs.

3.2.5.1 *Write Parameters*

Parameter Bytes	Description	
Byte 1	Red LED PWM parameter	LSByte
Byte 2	Red LED PWM parameter	MSByte
Byte 3	Green LED PWM parameter	LSByte
Byte 4	Green LED PWM parameter	MSByte
Byte 5	Blue LED PWM parameter	LSByte
Byte 6	Blue LED PWM parameter	MSByte

1. When an all-white image is being displayed, this command allows the system white point to be adjusted while also establishing the total LED power. This is true whether the CAIC algorithm is enabled or disabled.
2. The parameters specified by this command have a resolution of 10 bits, and are defined by the appropriate PMIC specification.
3. When the CAIC algorithm is disabled, this command directly sets the LED PWM parameters (the controller sends the R, G, and B values directly to the PMIC device) regardless of the image being displayed.
4. When the CAIC algorithm is enabled:

- This command directly sets the LED PWM parameters when an all-white image is displayed. If the image is changed from an all-white image, depending on the image the CAIC algorithm may alter one or more of the LED PWM parameters from those specified by this command, and the total LED power may drop. The *Read CAIC RGB LED PWM (5Fh)* command reads the actual LED PWM parameters for the image currently displayed.
- In the case of an all-white image, the values read by the *Read CAIC RGB PWM (5Fh)* command closely matches, but may not exactly match, those requested using the *Write RGB LED PWM (54h)* command. For an all-white image, the *Read CAIC RGB LED PWM (5Fh)* command returns PWM parameters within ±4 PMIC device PWM steps for each LED color relative to those requested by the *Write RGB LED PWM (54h)* command.
- When the *Write RGB LED PWM (54h)* command is used to change the LED PWM parameters, the LED PWM for any color must not be changed by more than ±25% from the nominal PWM used for that color when the CAIC LUTs were created. Do not set an LED to a PWM value beyond the maximum value supported in the CAIC intensity-to-current LUT for the corresponding color.
- The maximum total LED power for any displayed image occurs for an all-white image because in this case the CAIC algorithm requests the CAIC LED maximum available power. The maximum available LED power for CAIC is controlled by the *Write RGB LED PWM (54h)* command, because this command controls PWM parameters for an all-white image. After the PWM parameters are adjusted, the *Read CAIC LED Max Available Power (57h)* command can be used to see the max power in Watts that the CAIC algorithm derived.

3.2.6 Read RGB LED Current PWM (55h)

This command reads the PWM parameters for the red, green, and blue LEDs of the display module.

3.2.6.1 Read Parameters

This command has no parameters.

3.2.6.2 Return Parameters

Parameter Bytes	Description	
Byte 1	Red LED PWM parameter	LSByte
Byte 2	Red LED PWM parameter	MSByte
Byte 3	Green LED PWM parameter	LSByte
Byte 4	Green LED PWM parameter	MSByte
Byte 5	Blue LED PWM parameter	LSByte
Byte 6	Blue LED PWM parameter	MSByte

See *Section 3.2.5* for a detailed description of the return parameters.

Unused most significant bits are set to '0'.

3.2.7 Read CAIC LED Max Available Power (57h)

This command reads the specified maximum LED power allowed for the display module at the LED PWM settings set by the *Write RGB LED Current PWM (54h)* command.

3.2.7.1 Read Parameters

This command has no parameters.

3.2.7.2 Return Parameters

Parameter Bytes	Description	
Byte 1	Maximum LED power	LSByte
Byte 2	Maximum LED power	MSByte

1. The controller specifies the value in Watts × 100 (example: 25.75 W = A0Fh)
2. This command is only applicable when CAIC is enabled.
3. The maximum available LED power associated with the CAIC algorithm is specific to an all white displayed image where the LED PWM parameters are set by the *Write RGB LED Current PWM (54h)*. The calculation is:

Max Avail Pwr = (Rdc * Rledc * Rledv) + (Gdc * Gledc * Gledv) + (Bdc * Bledc * Bledv)

i. Rdc = Red Duty Cycle; Rledc = Red LED PWM; Rledv = Red LED Voltage

ii. Gdc = Green Duty Cycle; Gledc = Green LED PWM; Gledv = Green LED Voltage

iii. Bdc = Blue Duty Cycle; Bledc = Blue LED PWM; Bledv = Blue LED Voltage

Example: (.30 * .49A * 2.0V) + (.50 * .39A * 3.1V) + (.20 * .39A * 3.1V) = 1.140W

For example: (.30 × .49 A × 2.0 V) + (.50 × .39 A × 3.1 V) + (.20 × .39 A × 3.1 V) = (.30 × .980 W) + (.50 × 1.209 W) + (.20 × 1.209 W) = 1.140 W

3.2.8 Write RGB LED Max Current PWM (5Ch)

This command specifies the maximum LED PWM allowed for each LED in the display module when CAIC is disabled.

3.2.8.1 Write Parameters

Parameter Bytes	Description	
Byte 1	Maximum Red LED PWM	LSByte
Byte 2	Maximum Red LED PWM	MSByte
Byte 3	Maximum Green LED PWM	LSByte
Byte 4	Maximum Green LED PWM	MSByte
Byte 5	Maximum Blue LED PWM	LSByte
Byte 6	Maximum Blue LED PWM	MSByte

Default: Firmware specified

1. The parameters specified by this command have a resolution of 10 bits, and are defined by the appropriate PMIC specification.
2. This command sets the maximum LED PWM parameters that can be used when the CAIC algorithm is disabled. When the CAIC algorithm is enabled, the maximum LED PWM parameters are determined by the CAIC algorithm LUTs stored in Flash.
3. For further information about LED PWM and the CAIC algorithm, see the notes for the *Write RGB LED PWM (54h)* command.
4. Set unused most significant bits to ‘0’.

3.2.9 Read RGB LED Max Current PWM (5Dh)

This command reads the specified maximum LED PWM allowed for each LED.

3.2.9.1 Read Parameters

This command has no parameters.

3.2.9.2 Return Parameters

Parameter Bytes	Description	
Byte 1	Maximum Red LED PWM	LSByte
Byte 2	Maximum Red LED PWM	MSByte
Byte 3	Maximum Green LED PWM	LSByte
Byte 4	Maximum Green LED PWM	MSByte
Byte 5	Maximum Blue LED PWM	LSByte
Byte 6	Maximum Blue LED PWM	MSByte

1. See the Write *RGB LED PWM Control* command for a detailed description of the return parameters.
2. Unused most significant bits are set to‘0’.

3.2.10 Read CAIC RGB LED Current PWM (5Fh)

This command reads the specified PWM parameters for the red, green, and blue LEDs of the display module.

3.2.10.1 Read Parameters

This command has no parameters.

3.2.10.2 Return Parameters

Parameter Bytes	Description	
Byte 1	Red LED PWM parameter	LSByte
Byte 2	Red LED PWM parameter	MSByte
Byte 3	Green LED PWM parameter	LSByte
Byte 4	Green LED PWM parameter	MSByte
Byte 5	Blue LED PWM parameter	LSByte
Byte 6	Blue LED PWM parameter	MSByte

1. The parameters returned by this command have a resolution of 10 bits, and are defined by the appropriate PMIC specification.
2. When the CAIC algorithm is enabled using the *LED Output Control Method* command.
   - The *Write RGB LED PWM* command directly sets the LED PWM parameters when the controller diplays an all white image. If the image changes from an all white image, depending on the image, the CAIC algorithm may alter one or more of the LED PWM parameters from those specified the *Write RGB LED PWM* command and the total LED power may also drop. The actual LED PWM parameters for the image currently being displayed can be read using this command (the *Read CAIC RGB LED PWM (5Fh)* command).
   - In the case of an all white image, the values returned by this command closely match, but may not exactly match, those specified using the *Write RGB LED PWM* command. For an all white image, this command provides values within ±4 PMIC device current steps for each LED color relative to those specified with the *Write RGB LED PWM* command.
   -
   -
3. Use of this command is only appropriate when the *LED Output Control Method* is set to *CAIC (Automatic) RGB LED PWM Control*.
4. Unused most significant bits are set to‘0’.

3.3 Image Processing Control Commands

3.3.1 Write Local Area Brightness Boost Control (80h)

This command controls the local area brightness boost image processing functionality for Display modes.

3.3.1.1 Write Parameters

Parameter Bytes	Description
Byte 1	LABB Control
Byte 2	LABB Strength Setting

Byte 1	LABB Control
b(7:4)	Sharpness strength
b(3:2)	*Reserved*
b(1:0)	LABB control: • 0h: Disabled • 1h: Enabled: Manual strength control (no light sensor) • 2h: Enabled: Automatic strength control (uses light sensor) • 3h: *Reserved*

Default: 0001h

1. The key function of the LABB is to adaptively gain up darker parts of the image to achieve an overall brighter image.
2. For automatic strength control, the controller uses a light sensor to automatically adjust the applied image strength based on the measured black level of the screen, or the ambient lighting level of the room.
3. Sharpness strength can range from 0 to 15, with 0 indicating sharpness disabled, and 15 indicating the maximum sharpness. The LABB function must be enabled (either Manual or Automatic) to make use of Sharpness.
4. For LABB Strength, 0 indicates no boost applied, and 255 indicates the maximum boost that is considered viable in a product. The strength is not a direct indication of the gain because the gain varies depending on image content.
5. LABB is supported in TPG, Splash, External Input Display Modes, but auto-disabled in curtain mode.

3.3.2 Read Local Area Brightness Boost Control (81h)

This command reads the state of the local area brightness boost image processing functionality for Display modes.

3.3.2.1 Read Parameters

This command has no parameters.

3.3.2.2 Return Parameters

Parameter Bytes	Description
Byte 1	LABB Control
Byte 2	LABB Gain Value

Byte 1	LABB Control
b(7:4)	Sharpness strength
b(3:2)	*Reserved*
b(1:0)	LABB control: • 0h: Disabled • 1h: Enabled: Manual strength control (no light sensor) • 2h: Enabled: Automatic strength control (uses light sensor) • 3h: *Reserved*

Table 14 shows the bit order and weighting for the LABB Gain value, which can range from 1 to almost 8 (controller software typically limits the lower value to 1).

Table 14. Bit Weight Definition for LABB Gain Value

b7	b6	b5	b4	b3	b2	b1	b0
2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}

3.3.3 Write CAIC Image Processing Control (84h)

This command controls the CAIC Image Processing functionality applicable in Display modes.

3.3.3.1 Write Parameters

Parameter Bytes	Description
Byte 1	Reserved
Byte 2	CAIC Maximum Lumens Gain
Byte 3	CAIC Clipping Threshold

Default: Firmware specified

1. The CAIC algorithm (Content Adaptive Illumination Control) provides adaptive control of the LED PWMs and the digital gain applied to the image.
2. The CAIC algorithm is enabled or disabled based on the method of LED PWM control using the *Write LED Output Control Method* command. When enabled, the CAIC algorithm provides automatic control of the LED PWMs as specified by this command and the *Write LED Output Control Method* command.
3. Table 15 shows the bit order and weighting for the CAIC Maximum Lumens Gain value, which has a valid range from 1.0 to 4.0. The controller considers values outside of this range an error (invalid command parameter value – communication status) and does not execute the command.

Table 15. Bit Weight Definition for the CAIC Maximum Gain Value

b7	b6	b5	b4	b3	b2	b1	b0
2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}

4. The CAIC Maximum Lumens Gain parameter sets the maximum lumens gain that a pixel can have as a result of both digital gain and increasing LED PWM parameters. It also serves to bias the CAIC algorithm towards either Constant Power (variable brightness) or Constant Lumens (variable power). Some examples are listed below:
   - Maximum Gain value = 1.0: This biases performance to Constant Lumens. In this case, LED power is reduced for those images where this is possible, but lumens do not increase or decrease.
   - Maximum Lumens Gain value = 4.0: This biases performance to Constant Power. In this case, power is held constant for most images, while the lumens are gained up. For the small percent of images where the gain exceeds 4.0, lumens stop increasing and the power reduces instead.
5. Table 16 shows the bit order and weighting for the CAIC Clipping Threshold value, which has a valid range from 0.0% to 2.0%. The controller considers values outside of this range an error (invalid command parameter value – communication status) and does not execute the command.

Table 16. Bit Weight Definition for the CAIC Clipping Threshold Value

b7	b6	b5	b4	b3	b2	b1	b0
2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}

6. The CAIC Clipping Threshold parameter sets the percentage of pixels that can be clipped by the CAIC algorithm over the full frame of active data due to the digital gain being applied by the CAIC algorithm.
7. CAIC can be enabled in TPG and External Input mode, but auto-disabled in Splash and Curtain mode.

Table 17. LABB and CAIC Modes

Feature	TPG	Splash	Curtain	External Input
LABB	Supported	Supported	Auto-disabled	Supported
CAIC	Supported	Auto-disabled	Auto-disabled	Supported

3.3.4 Read CAIC Image Processing Control (85h)

This command reads the state of the CAIC Image Processing functionality applicable in Display modes.

3.3.4.1 Read Parameters

This command has no parameters.

3.3.4.2 Return Parameters

Parameter Bytes	Description
Byte 1	Reserved
Byte 2	CAIC Maximum Lumens Gain
Byte 3	CAIC Clipping Threshold

Information on these parameters can be found in the notes for the *Write CAIC Image Processing Control* command.

3.3.5 Write Keystone Correction Control (88h)

This command controls the keystone correction image processing functionality.

3.3.5.1 Write Parameters

Parameter Bytes	Description	
Byte 1	See Below	
Byte 2	Optical Throw Ratio	LSByte
Byte 3	Optical Throw Ratio	MSByte
Byte 4	Optical DMD Offset	LSByte
Byte 5	Optical DMD Offset	MSByte

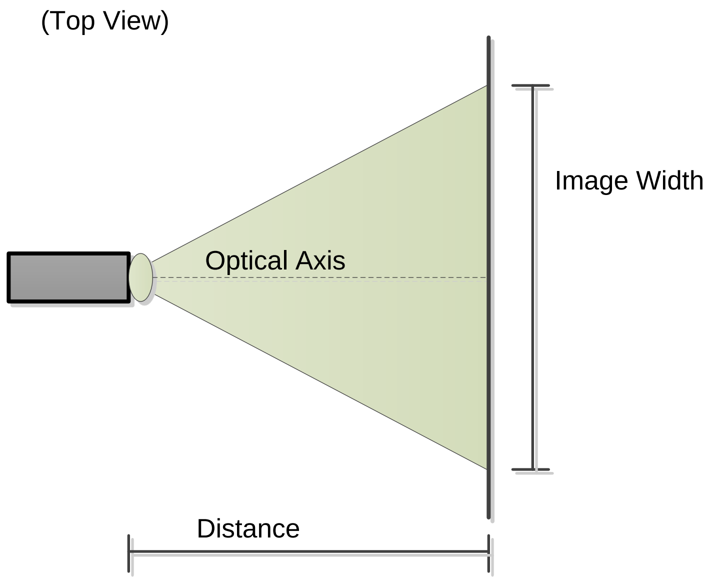
Byte 1	
b(7:1)	*Reserved*
b(0)	Keystone correction enable: • 0: Disabled • 1: Enabled

Default: All bytes: 00h

1. 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 (described in the *Write Keystone Projection Pitch Angle* command), based on the throw ratio, vertical offset, and projector orientation. Each of these parameters are provided by this command. With this information, keystone corrects for both overall and local area aspect ratio distortion. For both full screen images and sub-images, the full active area of the DMD uses keystone correction.
2. When keystone correction is enabled, use the *Write Border Color* command to set the border color to black. Any other color setting produces undesirable results.
3. Image rotation is allowed while keystone correction is enabled; however, it may not be appropriate for all situations or configurations. The user is responsible for determining if the result meets their needs.
4. Table 18 shows the bit order and weighting for the optical throw ratio data. Figure 17 defines how this data is determined.

Table 18. Bit Weight Definition for the Optical Throw Ratio Data

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}



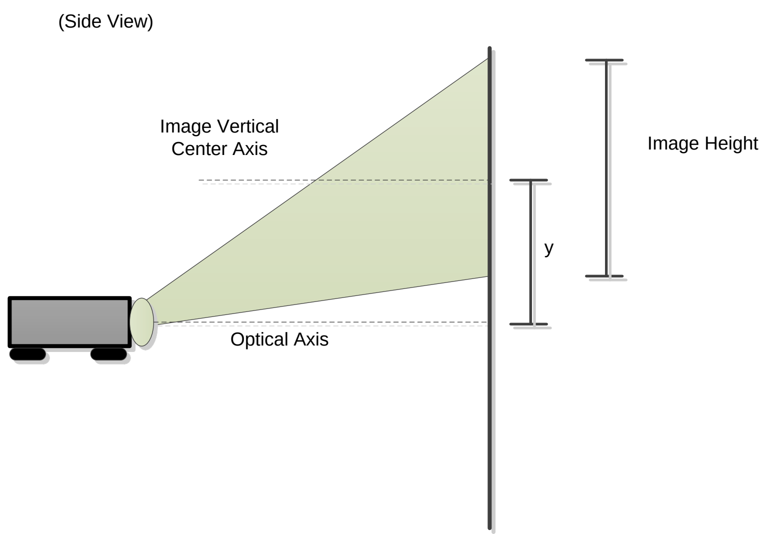
Throw Ratio = Distance / Image Width
Throw Ratio Register Value = 256 × Throw Ratio

Figure 17. Visual Definition and Calculation for Optical Throw Ratio Data

5. Table 19 shows the bit order and weighting for the 2's complement optical DMD offset data. Figure 18 shows how this data is calculated, while Figure 19 shows how the sign of the offset data is determined. The user needs to insure that both the value and the sign of the offset data are correctly determined.

Table 19. Bit Weight Definition for the Optical DMD Offset Data

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}



Vertical Offset = 2 × y / Image Height
Vertical Offset Register Value = 256 × Vertical Offset
(Image Height is always a positive value, while 'y' can be positive or negative)

Figure 18. Method for Calculation for Optical DMD Offset Data

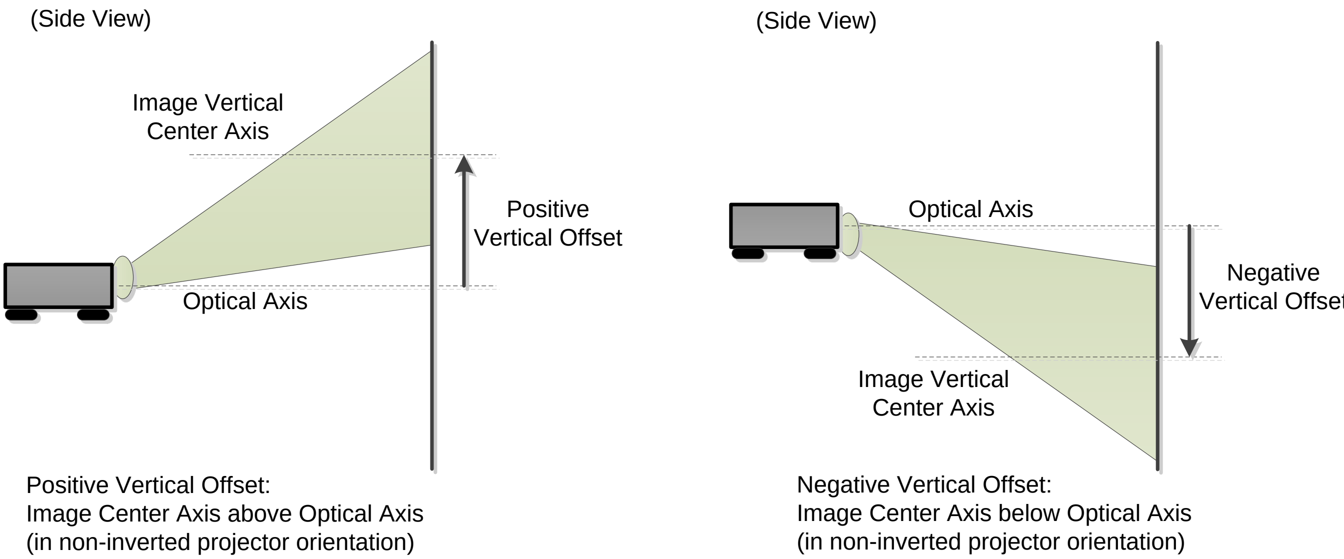


Figure 19. Sign Determination for Optical DMD Offset Data

6. Figure 20 shows examples of Non-Inverted and Inverted projector orientation. This information is required for Byte 1 of this command.

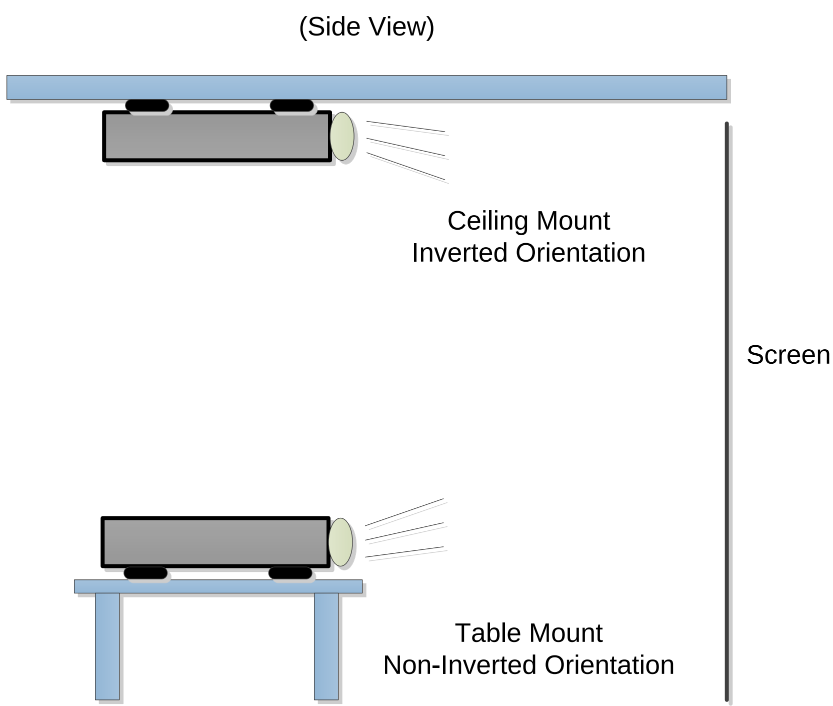


Figure 20. Examples of Non-Inverted and Inverted Projector Orientations

3.3.6 Read Keystone Correction Control (89h)

This command reads the state of the keystone correction image processing functionality.

3.3.6.1 Read Parameters

This command has no parameters.

3.3.6.2 Return Parameters

Parameter Bytes	Description	
Byte 1	See Below	
Byte 2	Optical Throw Ratio	LSByte
Byte 3	Optical Throw Ratio	MSByte
Byte 4	Optical DMD Offset	LSByte
Byte 5	Optical DMD Offset	MSByte

Byte 1	
b(7:1)	*Reserved*
b(0)	Keystone correction enable: • 0: Disabled • 1: Enabled

The bit order and weighting of the Optical Throw Ratio and Optical DMD Offset data is defined in *Write Keystone Correction Control (88h)* .

3.4 Light Control Commands

This section describes the commands used in the Light Control modes for pattern streaming. The following commands must be executed (in the same order as shown below) to successfully set up pattern streaming modes:

1. **External Pattern Streaming**
   - Write Trigger Out Configuration (see Section 3.4.3)
   - Write Pattern Configuration (see Section 3.4.7)
   - Write Operating Mode Select (see Section 3.1.1)
2. **Internal Pattern Streaming**
   - Write Trigger Out Configuration (see Section 3.4.3)
   - Write Trigger In Configuration (see Section 3.4.1)
   - Write Pattern Ready Configuration (see Section 3.4.5)
   - Write Operating Mode Select (see Section 3.1.1)
   - Write Internal Pattern Control (see Section 3.4.13)
3. **Splash Pattern mode**
   - Write Trigger Out Configuration (see Section 3.4.3)
   - Write Pattern Configuration (see Section 3.4.7)
   - Write Operating Mode Select (see Section 3.1.1)
   - Write Splash Screen Select (see Section 3.1.7)
   - Write Display Size (see Section 3.1.12)
   - Write Input Image Size (see Section 3.1.25)
   - Write Image Crop (see Section 3.1.10)
   - Write Splash Screen Execute (see Section 3.1.27)

The following sections describe these commands (and other optional Light Control commands) in detail.

3.4.1 Write Trigger In Configuration (90h)

This command is used to define the Trigger In configuration. Trigger In signal is used to indicate the start of each pattern in a pattern set and is applicable only for internal pattern streaming mode.

3.4.1.1 Write Parameters (90h)

Parameter Byte	
b(7:2)	*Reserved*
b(1)	Polarity '0': Active Low '1': Active High
b(0)	Enable '0': Disable '1': Enable

Default: Flash settings defined

1. Trigger In settings are only applied when the user sends *Write Operating Mode Select* command with a Light Control mode selected.
2. The controller retains and uses the Trigger In setting until next *Write Trigger In Configuration* command.
3. The controller uses Trigger In Enable to indicate in which Internal Pattern Streaming mode the system operates:
   - Enable: Trigger In Mode, the system waits for an external signal to initiate the next pattern in a sequence.
   - Disable: Free Running Mode, the controller needs no external signal to display the next pattern in a sequence.

3.4.2 Read Trigger In Configuration (91h)

This command reads the Trigger In configuration.

3.4.2.1 Read Parameters

This command has no parameters.

3.4.2.2 Return Parameters

Parameter Byte	
b(7:2)	*Reserved*
b(1)	Polarity '0': Active Low '1': Active High
b(0)	Enable '0': Disable '1': Enable

3.4.3 Write Trigger Out Configuration (92h)

This command defines the Trigger Out characteristics.

3.4.3.1 Write Parameters (92h)

Parameter Bytes	Description	
Byte 1	Trigger Configuration	
Byte 2	Delay (in µs)	LSByte
Byte 3	Delay (in µs)	
Byte 4	Delay (in µs)	
Byte 5	Delay (in µs)	MSByte

Byte 1	Trigger Configuration
b(7:3)	*Reserved*
b(2)	Trigger Inversion '0': Not Inverted '1': Inverted
b(1)	Trigger Enable '0': Disable '1': Enable
b(0)	Trigger Select '0': Trigger Out 1 '1': Trigger Out 2

Default: Flash settings defined

1. Trigger Out settings are only applied when the user sends *Write Operating Mode Select* command with a Light Control mode selected.
2. The controller retains and uses the Trigger Out setting until next *Write Trigger Out Configuration* command.
3. Delay range of Trigger Out 1: [0, Pattern Period]
4. Delay range of Trigger Out 2: [-Pre-Illumination Dark Time, Pattern Period]
5. Trigger Out 2 supports negative values, meaning the trigger can be sent in advance. The delay is processed as signed 16-bit integer.

3.4.4 Read Trigger Out Configuration (93h)

This command reads the Trigger Out configuration of the specified trigger.

3.4.4.1 Read Parameters

Parameter Byte	
b(7:1)	*Reserved*
b(0)	Trigger Select '0': Trigger Out 1 '1': Trigger Out 2

3.4.4.2 Return Parameters

Parameter Bytes	Description	
Byte 1	Trigger Configuration	
Byte 2	Delay (in µs)	LSByte
Byte 3	Delay (in µs)	
Byte 4	Delay (in µs)	
Byte 5	Delay (in µs)	MSByte

Byte 1	Trigger Configuration
b(7:3)	*Reserved*
b(2)	Trigger Inversion '0': Not Inverted '1': Inverted
b(1)	Trigger Enable '0': Disable '1': Enable
b(0)	Trigger Select '0': Trigger Out 1 '1': Trigger Out 2

Trigger Out 2 supports negative values, meaning the trigger can be sent in advance. The delay is processed as signed 16-bit integer.

3.4.5 Write Pattern Ready Configuration (94h)

This command defines the configuration of the Pattern Ready signal. Pattern Ready indicates when the Internal Patterns are ready to be displayed in Trigger In Mode.

3.4.5.1 Write Parameters

Parameter Byte	
b(7:2)	*Reserved*
b(1)	Trigger Inversion '0': Not Inverted '1': Inverted
b(0)	Enable '0': Disable '1': Enable

Default: Flash settings defined

3.4.6 Read Pattern Ready Configuration (95h)

This command reads the Pattern Ready configuration.

3.4.6.1 Read Parameters

This command has no parameters.

3.4.6.2 Return Parameters

Parameter Byte	
b(7:2)	*Reserved*
b(1)	Trigger Inversion '0': Not Inverted '1': Inverted
b(0)	Enable '0': Disable '1': Enable

3.4.7 Write Pattern Configuration (96h)

This command defines the pattern characteristics. The settings defined by this command are applied only when the user sends the *Write Operating Mode Select* command (Section 3.1.1) with either External Pattern Streaming mode or Splash Pattern mode selected.

3.4.7.1 Write Parameters

Parameter Bytes	Description	Byte Order
Byte 1	Sequence Type	
Byte 2	Number of Patterns	
Byte 3	Illumination Select	
Byte 4	Illumination Time (µs)	LSByte
Byte 5	Illumination Time (µs)	
Byte 6	Illumination Time (µs)	
Byte 7	Illumination Time (µs)	MSByte
Byte 8	Pre-illumination Dark Time (µs)	LSByte
Byte 9	Pre-illumination Dark Time (µs)	
Byte 10	Pre-illumination Dark Time (µs)	
Byte 11	Pre-illumination Dark Time (µs)	MSByte
Byte 12	Post-illumination Dark Time (µs)	LSByte
Byte 13	Post-illumination Dark Time (µs)	
Byte 14	Post-illumination Dark Time (µs)	
Byte 15	Post-illumination Dark Time (µs)	MSByte

Byte 1	**Sequence Type**
	0h: 1-bit mono
	1h: 1-bit RGB
	2h: 8-bit mono
	3h: 8-bit RGB
	4h–FFh: *Reserved*

Byte 2	**Number of Patterns**	
	Splash:	External:
	Up to 16 for 1-bit patterns	Up to 24 for 1-bit patterns
	Up to 2 for 8-bit patterns	Up to 3 for 8-bit patterns

Byte 3	**Illumination Select**		
b(7:3)	*Reserved*		
b(2)	Blue LED	'0': Disable	'1': Enable
b(1)	Green LED	'0': Disable	'1': Enable
b(0)	Red LED	'0': Disable	'1': Enable

3.4.8 Read Pattern Configuration (97h)

This command reads the pattern configuration.

3.4.8.1 Read Parameters

This command has no parameters.

3.4.8.2 Return Parameters

Parameter Bytes	Description	Byte Order
Byte 1	Sequence Type	
Byte 2	Number of Patterns	
Byte 3	Illumination Select	
Byte 4	Illumination Time (µs)	LSByte
Byte 5	Illumination Time (µs)	
Byte 6	Illumination Time (µs)	
Byte 7	Illumination Time (µs)	MSByte
Byte 8	Pre-illumination Dark Time (µs)	LSByte
Byte 9	Pre-illumination Dark Time (µs)	
Byte 10	Pre-illumination Dark Time (µs)	
Byte 11	Pre-illumination Dark Time (µs)	MSByte
Byte 12	Post-illumination Dark Time (µs)	LSByte
Byte 13	Post-illumination Dark Time (µs)	
Byte 14	Post-illumination Dark Time (µs)	
Byte 15	Post-illumination Dark Time (µs)	MSByte

Refer to *Write Pattern Configuration (96h)* command (Section 3.4.7) for descriptions of the return parameters.

3.4.9 Write Pattern Order Table Entry (98h)

This command defines a new Pattern Order Table or reloads one stored in flash (assuming there is one). It is only applicable for Internal Pattern Streaming mode.

3.4.9.1 *Write Parameters (98h)*

Parameter Bytes	Description	Byte Order
Byte 1	Write Control	
Byte 2	Pattern Set Entry Index	
Byte 3	Number of patterns to display	
Byte 4	Illumination Select	
Byte 5	Pattern Invert	LSWord LSByte
Byte 6	Pattern Invert	LSWord
Byte 7	Pattern Invert	LSWord
Byte 8	Pattern Invert	LSWord MSByte
Byte 9	Pattern Invert	MSWord LSByte
Byte 10	Pattern Invert	MSWord
Byte 11	Pattern Invert	MSWord
Byte 12	Pattern Invert	MSWord MSByte
Byte 13	Illumination Time (µs)	LSByte
Byte 14	Illumination Time (µs)	
Byte 15	Illumination Time (µs)	
Byte 16	Illumination Time (µs)	MSByte
Byte 17	Pre-illumination Dark Time (µs)	LSByte
Byte 18	Pre-illumination Dark Time (µs)	
Byte 19	Pre-illumination Dark Time (µs)	
Byte 20	Pre-illumination Dark Time (µs)	MSByte
Byte 21	Post-illumination Dark Time (µs)	LSByte
Byte 22	Post-illumination Dark Time (µs)	
Byte 23	Post-illumination Dark Time (µs)	
Byte 24	Post-illumination Dark Time (µs)	MSByte

Byte 1 Write Control

0h: Continue
1h: Start
2h: Reload from flash
3h–FFh: *Reserved*

1. To define a Pattern Order Table at run-time, send this command with Start selected as the Write Control option. This indicates to SW that this is the first entry in the new Pattern Order Table.
2. To continue adding to the existing Pattern Order Table, send this command with Continue as the Write Control option. This creates a new entry and appends to the end of the existing table.
3. To use the Pattern Order Table stored in flash, send this command with Reload from flash as the Write Control option, in which case the controller ignores the remaining parameter values.

Byte 4	Illumination Select		
b(7:3)	*Reserved*		
b(2)	Blue LED	'0': Disable	'1': Enable
b(1)	Green LED	'0': Disable	'1': Enable
b(0)	Red LED	'0': Disable	'1': Enable

1. Users have the option of loading the Pattern Order Table from flash (refer to Section 3.7 for updating pattern data stored in flash) or loading the table entries directly to internal memory during run-time using this command.
2. The maximum number of patterns that can be stored at a time in internal memory is shown below.

Table 20. Maximum number of patterns per pattern set

Controller	Horizontal Patterns		Vertical Patterns	
	1-bit	8-bit	1-bit	8-bit
DLPC3470	64	8	64	8
DLPC3478	64	8	51	6

3. Each pattern is mapped to a bit in the Pattern Invert Register (bit 0 corresponds to pattern 0, and so on). A bit value of 1 inverts the pattern.
4. Up to 128 entries can be added in the Pattern Order Table.

3.4.10 Read Pattern Order Table Entry (99h)

This command reads the Pattern Order Table entry for the given Table Entry Index.

3.4.10.1 Read Parameters

Parameter Bytes	Description
Byte 1	Pattern Order Table Entry Index

If invalid index (or index that does not exist), software returns all 0s.

3.4.10.2 Return Parameters

Parameter Bytes	Description	Byte Order
Byte 1	Write Control	
Byte 2	Pattern Set Entry Index	
Byte 3	Number of patterns to display	
Byte 4	Illumination Select	
Byte 5	Pattern Invert	LSWord LSByte
Byte 6	Pattern Invert	LSWord
Byte 7	Pattern Invert	LSWord
Byte 8	Pattern Invert	LSWord MSByte
Byte 9	Pattern Invert	MSWord LSByte
Byte 10	Pattern Invert	MSWord
Byte 11	Pattern Invert	MSWord
Byte 12	Pattern Invert	MSWord MSByte
Byte 13	Illumination Time (µs)	LSByte
Byte 14	Illumination Time (µs)	
Byte 15	Illumination Time (µs)	
Byte 16	Illumination Time (µs)	MSByte
Byte 17	Pre-illumination Dark Time (µs)	LSByte
Byte 18	Pre-illumination Dark Time (µs)	
Byte 19	Pre-illumination Dark Time (µs)	
Byte 20	Pre-illumination Dark Time (µs)	MSByte
Byte 21	Post-illumination Dark Time (µs)	LSByte
Byte 22	Post-illumination Dark Time (µs)	
Byte 23	Post-illumination Dark Time (µs)	
Byte 24	Post-illumination Dark Time (µs)	MSByte

Refer to *Write Pattern Order Table Entry (98h)* command above for a description of the return parameters.

3.4.11 Read Light Control Sequence Version (9Bh)

3.4.11.1 Read Parameters

This command has no parameters.

3.4.11.2 Return Parameters

Parameter Bytes	Description
Byte 1	*Reserved*
Byte 2	Patch
Byte 3	Minor
Byte 4	Major

3.4.12 Read Validate Exposure Time (9Dh)

This command is used to check whether the requested exposure (or illumination) time is supported for the given pattern mode and bit depth. If requested exposure time is supported, the minimum pre-exposure and post-exposure times are returned.

3.4.12.1 Read Parameters

Parameter Bytes	Description
Byte 1	Pattern Mode
Byte 2	Bit Depth
Byte 3	Requested Exposure Time (µs) LSB
Byte 4	Requested Exposure Time (µs)
Byte 5	Requested Exposure Time (µs)
Byte 6	Requested Exposure Time (µs) MSB

Byte 1 Pattern Mode

00h: External
01h: Internal
02h: Splash
03h–FFh: *Reserved*

Byte 2 Bit Depth

00h: 1-bit mono
01h: 1-bit RGB
02h: 8-bit mono
03h: 8-bit RGB
04h–FFh: *Reserved*

3.4.12.2 Return Parameters

Parameter Bytes	Description
Byte 1	Exposure Time Support and Zero Dark Time Support
Byte 2	Exposure Time (µs) LSB
Byte 3	Exposure Time (µs)
Byte 4	Exposure Time (µs)
Byte 5	Exposure Time (µs) MSB
Byte 6	Minimum Pre-Exposure Dark Time (µs) LSB
Byte 7	Minimum Pre-Exposure Dark Time (µs)
Byte 8	Minimum Pre-Exposure Dark Time (µs)
Byte 9	Minimum Pre-Exposure Dark Time (µs) MSB
Byte 10	Minimum Post-Exposure Dark Time (µs) LSB
Byte 11	Minimum Post-Exposure Dark Time (µs)
Byte 12	Minimum Post-Exposure Dark Time (µs)
Byte 13	Minimum Post-Exposure Dark Time (µs) MSB

Byte 1

b(1) Zero Dark Time Support
'0': Not Supported
'1': Supported

b(0) Exposure Time Support
'0': Not Supported
'1': Supported

The returned timing parameters must be interpreted based on the Exposure Time Support parameter as follows:

Exposure Time Support	Returned Exposure Time	Returned Dark Times
'0': Not supported	**Minimum Exposure Time**	Junk Value
'1': Supported	Requested Exposure Time	**Minimum Dark Times**

3.4.13 Write Internal Pattern Control (9Eh)

This command controls the execution of Internal Patterns and is applicable only in that mode.

3.4.13.1 Write Parameters

Parameter Bytes	Description
Byte 1	Internal Pattern Control
Byte 2	Repeat Count

Byte 1 Internal Pattern Control

0h: Start
1h: Stop
2h: Pause
3h: Step
4h: Resume
5h: Reset

Byte 2 Repeat Count

0h: Do not repeat (run once)
1h: Repeat one additional time (run twice)
2h: Repeat two additional times (run three times)
…
…
FEh: Repeat 254 additional times
FFh: Repeat indefinitely

Internal Pattern Control

- *Start* begins to dsplay the first entry in the Pattern Order Table. Byte 2 of this command is used only when Start is selected. It indicates how many times to repeat the Pattern Order Table after the first execution.
- *Stop* stops execution of the Pattern Order Table and turns off illuminators.
- *Pause* pauses execution of the current Pattern Order Table entry. The last displayed pattern continues to be displayed.
- *Step* goes to the next pattern in the current Pattern Order Table entry being displayed, assuming that the user has already Paused the system.
- *Resume* continues to execute the current Pattern Order Table entry, assuming that the user has already Paused the system.
- *Reset* begins to dsplay the first pattern referenced in the current Pattern Order Table.

3.4.14 Read Internal Pattern Status (9Fh)

3.4.14.1 Read

This command returns the status of current pattern execution and is applicable only for Internal Pattern Streaming mode.

3.4.14.2 Read Parameters

This command has no parameters.

3.4.14.3 Return Parameters

Parameter Bytes	Description
Byte 1	Pattern Ready Status
Byte 2	Number of Pattern Order Table Entries
Byte 3	Current Pattern Order Table Entry Index
Byte 4	Current Pattern Set Index
Byte 5	Number of Patterns in the current Pattern Set
Byte 6	Number of Patterns displayed from current Pattern Set
Byte 7	Next Pattern Set Index

Byte 1 Pattern Ready Status

'0': Pattern not ready

'1': Pattern ready

3.5 General Setup Commands

3.5.1 Write Border Color (B2h)

This command specifies the onscreen border color for the display module.

3.5.1.1 Write Parameters

Byte1	Display border color:
	0h: Black
	1h: Red
	2h: Green
	3h: Blue
	4h: Cyan
	5h: Magenta
	6h: Yellow
	7h: White

Whenever the display image size is smaller than the active area of the DMD, the border color is used for all non-image pixels. Some examples using a border include a window box, pillar box (see Figure 21), or letterbox image.

The border color specified by this command is separate from the curtain color defined in the *Display Image Curtain* command, even though both use the curtain capability.

When using the keystone capability, set the border color to black. Setting this parameter to any other color when keystone is enabled produces undesirable results.

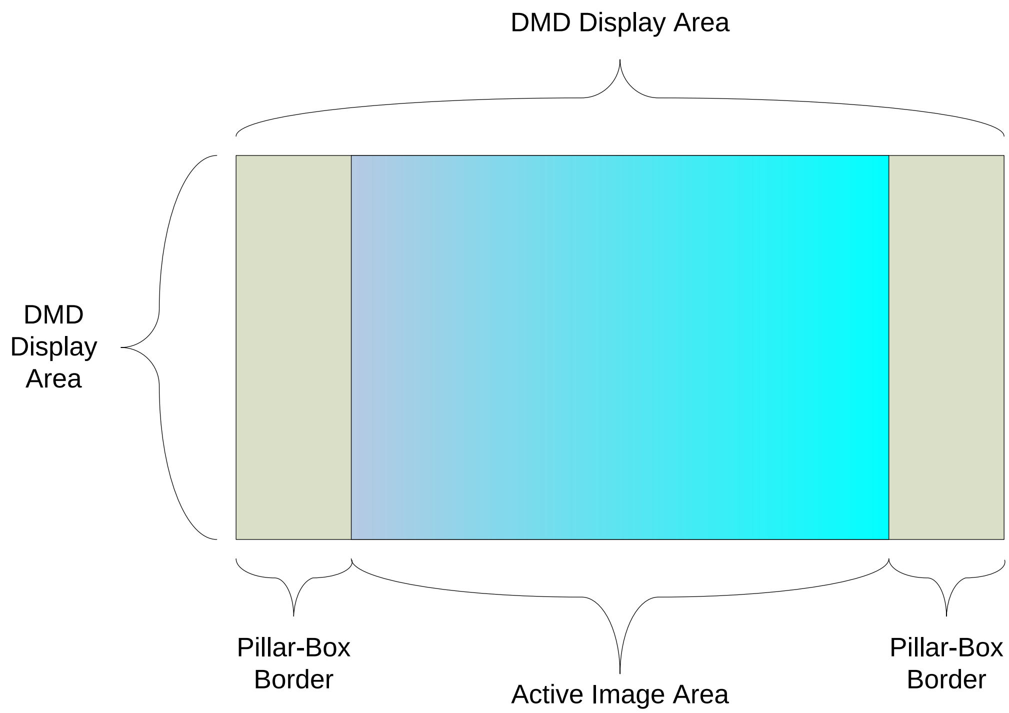


Figure 21. Pillar-Box Border Example

3.5.2 Read Border Color (B3h)

This command reads the state of the onscreen border color configuration.

3.5.2.1 Read Parameters

This command has no parameters.

3.5.2.2 Return Parameters

Byte 1	
b(7)	Pillar-box border color source: 0h: Defined by this command 1h: Flash defined 24-bit color
b(6:3)	*Reserved*
b(2:0)	Display border color: 0h: Black 1h: Red 2h: Green 3h: Blue 4h: Cyan 5h: Magenta 6h: Yellow 7h: White

3.5.3 Write Keystone Projection Pitch Angle (BBh)

This command specifies the projection pitch angle for the display module.

3.5.3.1 Write Parameters

Parameter Bytes	Description
Byte 1	Projection pitch angle (LSByte)
Byte 2	Projection pitch angle (MSByte)

Default: 0000h

Table 21 shows the bit order and weighting for the 2's complement projection pitch angle data.

Table 21. Bit Weight Definition for the Projection Pitch Angle Data

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}

This command is used in conjunction with the *Write Keystone Correction Control* (Section 3.3.5) command.

The projection pitch angle is limited to the range of –40 to 40 degrees. Figure 22 shows examples of the projection pitch angle.

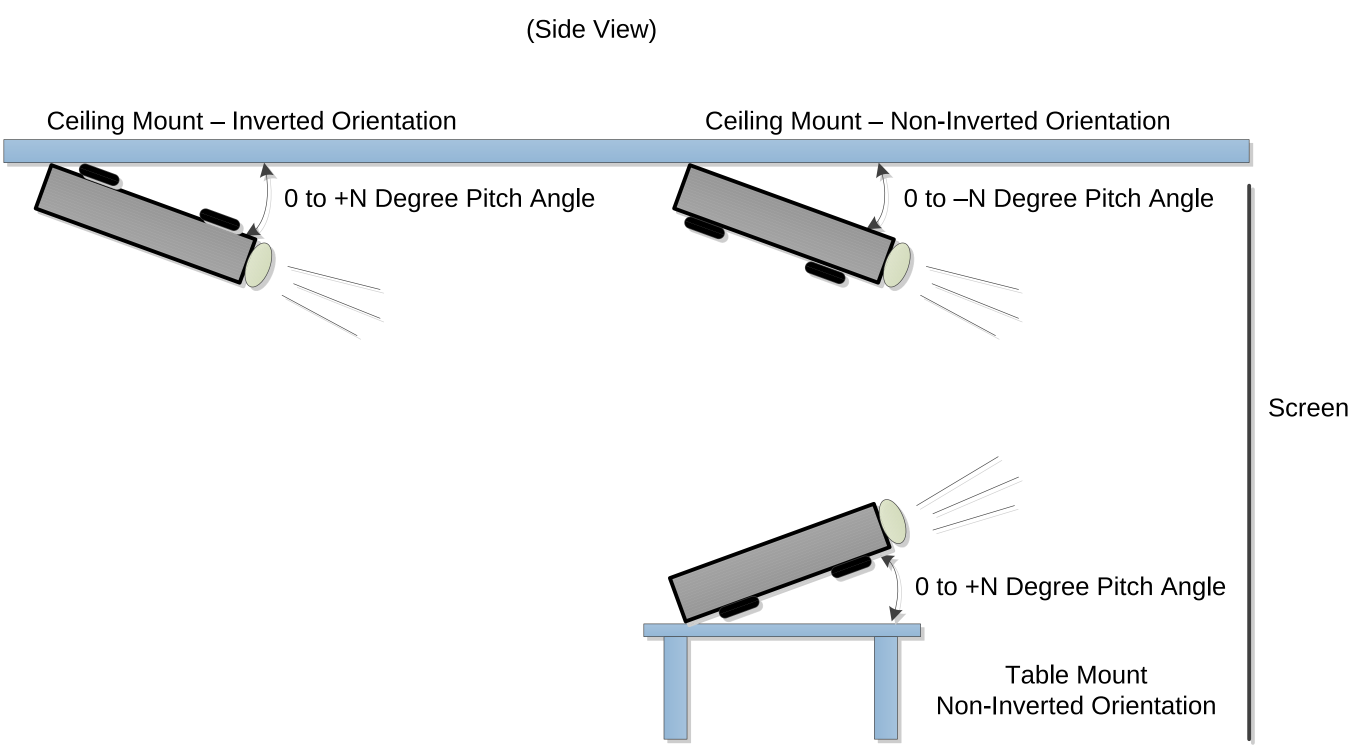


Figure 22. Examples of Projection Pitch Angle

3.5.4 Read Keystone Projection Pitch Angle (BCh)

This command reads the specified projection pitch angle.

3.5.4.1 Read Parameters

This command has no parameters.

3.5.4.2 Return Parameters

Parameter Bytes	Description
Byte 1	Projection pitch angle (LSByte)
Byte 2	Projection pitch angle (MSByte)

3.6 Administrative Commands

3.6.1 Read Short Status (D0h)

This command provides a short system status.

3.6.1.1 Read Parameters

This command has no parameters.

3.6.1.2 Return Parameters

Parameter Bytes	Description
Byte 1	Short System Status

Byte 1			
b(7)	Boot/Main Application '0': Boot '1': Main	b(3)	System Error '0': No Error '1': Error
b(6)	Light Control Sequence Error '0': No Error '1': Error	b(2)	*Reserved* '0': No Error '1': Error
b(5)	Flash Error '0': No Error '1': Error	b(1)	Communication Error '0': No Error '1': Error
b(4)	Flash Erase Complete '0': Complete '1': Not Complete	b(0)	System Initialization '0': Not Complete '1': Complete

1. The controller sets the *Flash Erase Complete* status bit at the start of the Flash erase process, and clears the bit when it completes the erase process. The flash status can be obtained during or after the erase process. To obtain this status during the erase process, only this command can be sent after the start of the flash erase. If the user sends any other command during the erase process, the controller holds the command without processing until it completes the flash erase process (thus blocking any following status requests until it processes the previously sent command).
2. The *Flash Error* bit indicates an error during any Flash operation. For Flash writes, the controller updates this bit at the end of each write transaction. However, when the controller detects an error, this bit remains in the error state until cleared. This error state allows the user the option of checking the status between each write transaction, or at the end of the update. When a write transaction starts, the flash status (and this error bit) is not accessible until the write transaction has completed.
3. The *Communication Error* bit indicates an error on the I^2C command interface. The *Read Communication Status* command gives specific details about communication errors.
4. The *System Error* bit indicates any errors other than *Flash Error* and *Communication Error*. The *Read Communication Status* command gives specific details about communication errors.
5. The controller clears the *Flash Error*, *Communication Error*, and *System Error* bits after executing the *Read Short Status* command.
6. When the *Light Control Sequence Error* bit is set, read the *System Status (D1h)* for specific errors in the selected Light Control mode.
7. Check the *Read Short Status* command *periodically*, *not continuously*. Continuous access can severely degrade system performance.

3.6.2 Read System Status (D1h)

This command reads system status information.

3.6.2.1 Read Parameters

This command has no parameters.

3.6.2.2 Return Parameters

Parameter Bytes	Description
Byte 1	Light Control and DMD interface status
Byte 2	LED status
Byte 3	Internal interrupt status
Byte 4	Miscellaneous status

All system status error bits are cleared by the *Read System Status* command.

Byte 1	Light Control and DMD interface status
b(7:3)	Light Control Error (Only applicable for Light Control Operating Modes) 0h: No Error 1h: illumination time not supported 2h: Pre-illumination dark time not supported 3h: Post-illumination dark time not supported 4h: Trigger Out 1 delay not supported 5h: Trigger Out 2 delay not supported
b(2)	*Reserved* '0': No Error '1': Error
b(1)	DMD Interface Error '0': No Error '1': Error
b(0)	DMD Device Error '0': No Error '1': Error

1. The system sets the DMD Interface Error when there are power management setup conflicts on this interface.
2. The system sets the DMD Device Error for the following conditions:
   - The system cannot read the DMD Device ID from the DMD
   - The firmware specified DMD Device ID does not match the actual DMD Device ID

Byte 2	LED Status		
b(7:6)	*Reserved*		
b(5)	Blue LED Error	b(2)	Blue LED State
b(4)	Green LED Error	b(1)	Green LED State
b(3)	Red LED Error	b(0)	Red LED State
	'0': No Error '1': Error		'0': Off '1': On

Byte 3	Internal interrupt status		
b(7)	*Reserved* '0': No Error '1': Error	b(3)	*Reserved* '0': No Error '1': Error
b(6)	*Reserved* '0': No Error '1': Error	b(2)	DC Power Supply '0': Supply voltage is in normal range '1': Supply voltage is low
b(5)	*Reserved* '0': No Error '1': Error	b(1)	Sequence Error '0': No Error '1': Error
b(4)	*Reserved* '0': No Error '1': Error	b(0)	Sequence Abort Error '0': No Error '1': Error

The DC power supply voltage status reported in b(2) is based on DLPA300x PMIC chip's monitoring of the DC power supply voltage by using the chip's BAT_LOW_WARN feature. The status of b(2) matches the value of the BAT_LOW_WARN bit in the DLPA300x PMIC chip's status/interrupt register.

Byte 4	Miscellaneous Status		
b(7)	*Reserved* '0': No Error '1': Error	b(3)	Master or Slave Operation '0': Master '1': Slave
b(6)	*Reserved* '0': No Error '1': Error	b(2)	Single or Dual Controller Configuration '0': Single '1': Dual
b(5)	Watchdog Timer Timeout '0': No Timeout '1': Timeout	b(1)	*Reserved* '0': No Error '1': Error
b(4)	Product Configuration Error '0': No Error '1': Error	b(0)	*Reserved* '0': No Error '1': Error

1. The system sets the SPI Flashless Data Request Error bit if the display does not start sending the requested data before the SPI flashless data request timeout is exceeded. After the timeout is exceeded, the display aborts the current request, and then attempts the request again.
2. The system sets the SPI Flashless Communication Error bit if the display has three consecutive SPI Flashless Data Request Errors. If this happens, it is assume that the SPI communication link is not operational, and system operations stops. The system requires a reset before operation restarts.
3. The system sets the Master or Slave bit as appropriate in both single and dual controller configurations.
4. The system sets the Product Configuration Error bit if it determines that some piece of the product configuration is not correct. Some examples are:
   - Invalid Controller/DMD Combination
   - Invalid Controller/DLPAx000 Combination
   - Invalid Flash build for current Controller, DMD, and/or DLPAx000 configuration
5. The system sets the Watchdog Timer Timeout bit if the system has been reset due to a watchdog timer timeout.

3.6.3 Read System Software Version (D2h)

This command reads the main application software version information.

3.6.3.1 Read Parameters

This command has no parameters.

3.6.3.2 Return Parameters

Parameter Bytes	Description		
Byte 1	Controller main application software version	Patch	LSByte
Byte 2	Controller main application software version	Patch	MSByte
Byte 3	Controller main application software version	Minor	
Byte 4	Controller main application software version	Major	

3.6.4 Read Communication Status (D3h)

3.6.4.1 Read Parameters

Parameter Bytes	Description
Byte 1	Command bus selection

Byte 1	**Command bus selection**		
b(7:2)	*Reserved*	b(1:0)	'10': I^2C

3.6.4.2 Return Parameters

Parameter Bytes	Description
Bytes 1 - 4	*Reserved*
Byte 5	I^2C Communication status
Byte 6	Aborted/ Invalid Command op-code

The system clears all communication status error bits when this command is read.

Byte 5	**I^2C Communication status**		
b(7)	*Reserved* '0': No Error '1': Error	b(3)	Flash Batch File Error '0': No Error '1': Error
b(6)	I^2C Bus Timeout Error '0': No Error '1': Error	b(2)	Command Processing Error '0': No Error '1': Error
b(5)	Invalid Number of Command Parameters '0': No Error '1': Error	b(1)	Invalid Command Parameter Value '0': No Error '1': Error
b(4)	Read Command Error '0': No Error '1': Error	b(0)	Invalid Command Error '0': No Error '1': Error

1. The system sets the Invalid Command Error bit when it does not recognize the command op-code. Byte 6 will contain the invalid op-code received by the controller.
2. The system sets the Invalid Command Parameter Error bit when the it detects that the value of a command parameter is not valid (e.g. out of allowed range).
3. The system sets the Command Processing Error bit when a fault is detected while processing a command. In this case, the system aborts the command and moves to the next command. Byte 6 will contain the op-code of the aborted command.
4. The system sets the Flash Batch File Error bit when an error occurs during the processing of a flash batch file. When the systems sets this bit, it typically sets another bit to indicate what kind of error was detected (such as: Invalid Command Error).
5. The system sets the Read Command Error bit when the host terminates the read operation before all of the requested data has been provided, or if the host continues to request read data after all of the requested data has been provided.
6. The system sets the Invalid Number of Command Parameters Error bit when too many or too few command parameters are received. In this case, the system aborts the command and moves to the next command. Byte 6 will contain the op-code of the aborted command.
7. The system sets the Bus Timeout Error bit when it releases control of the bus because the bus timeout value was exceeded.

3.6.5 Read Controller Device ID (D4h)

This command reads the Controller Device ID.

3.6.5.1 *Read Parameters*

This command has no parameters.

3.6.5.2 *Return Parameters*

Byte 1	
b(7:4)	*Reserved*
b(3:0)	Controller Device ID

Table 22. Controller Device ID Decode

Controller Device ID	Device Number	DMD Resolution	Number of Controllers	Package	LED Driver
0Fh	DLPC3470	854x480	1	13mm x 13mm (0.8mm pitch)	DLPA200x/DLPA3000
0Bh	DLPC3478	1280x720	1	13mm x 13mm (0.8mm pitch)	DLPA200x/DLPA300x

Unused controller device ID values are reserved.

3.6.6 Read DMD Device ID (D5h)

This command reads the DMD device ID.

3.6.6.1 Read Parameters

Byte 1	DMD Register Selection	
b(7:1)	*Reserved*	
b(0)	'0':	DMD Device ID
	'1':	*Reserved*

3.6.6.2 Return Parameters

Parameter Bytes	Description
Bytes 1 – 4	See Table 23

Table 23. DMD Device ID Reference Table

DMD Device ID				Device Description
Byte 1 (Identifier)	**Byte 2 (Byte Count)**	**Byte 3 (ID-MSByte)**	**Byte 4 (ID-LSByte)**	**Resolution and Type**
60h	0Dh	00h	64h / 69h / 8Dh	0.2 WVGA (854x480, Sub-LVDS)
60h	0Dh	00h	68h / 72h / 87h	0.3 720p (1280x720, Sub-LVDS)

3.6.7 Read System Temperature (D6h)

This command is used to read the system temperature using an external thermistor (if available).

3.6.7.1 Read Parameters

The command has no read parameters.

3.6.7.2 Read Parameters

Parameter Bytes	Description
Byte 1	LSByte
Byte 2	MSByte

Figure 23 shows the bit order and definition for the signed magnitude system temperature data, which will be returned in degrees C. The unspecified msbits (bits 15:12) will be set to ‘0’.

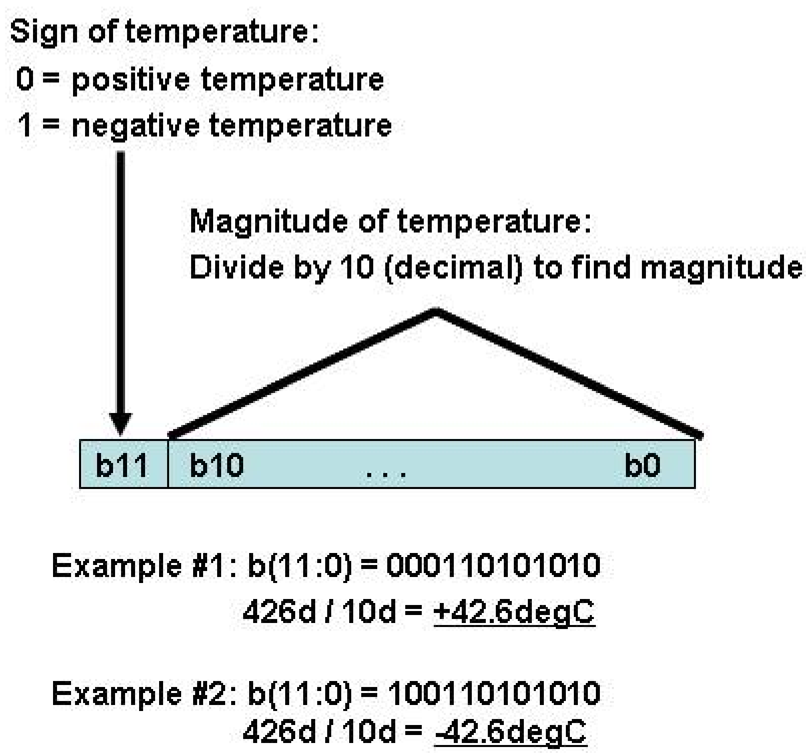


Figure 23. Bit Order and Definition for System Temperature

3.6.8 Read Flash Build Version (D9h)

This command reads the controller flash version.

3.6.8.1 Read Parameters

The command has no read parameters.

3.6.8.2 Return Parameters

Parameter Bytes	Description		
Byte 1	Flash build version	Patch	LSByte
Byte 2	Flash build version	Patch	MSByte
Byte 3	Flash build version	Minor	
Byte 4	Flash build version	Major	

3.6.9 Write Flash Batch File Delay (DBh)

This command is used to specify an execution time delay within a Flash batch file.

3.6.9.1 Write Parameters

Parameter Bytes	Description
Byte 1	Flash Batch File Delay (LSB)
Byte 2	Flash Batch File Delay (MSB)

1. The Flash batch file delay is to be specified in units of 1ms (e.g. 500ms = 01F4h).
2. This command is used to specify an execution delay time within a Flash batch file. It can only be used within a Flash batch file, and is not a valid command on the I2C interface.
3. Typical use of this command will be in the Auto-Init Flash batch file (batch file 0), but is valid for use in any batch file (See Section 3.1.24).

3.7 Flash Update Commands

This section describes the commands required to update internal pattern data stored in flash. The following steps must be followed to successfully update pattern data:

1. Generate your pattern data using the DLP® Display and Light Control EVM GUI. The .bin file generated by the “Save Pattern Data” button in GUI contains the pattern data to be sent through I2C.
2. Ensure internal patterns are not currently running. If the controller is in internal pattern mode, use the *Write Internal Pattern Control* command (Section 3.4.13) to stop running internal patterns.
3. Set flash data type to pattern data block using *Write Flash Data Type Select (DEh)* command.
4. Check whether the generated .bin file can fit within the currently existing pattern data flash block using the *Read Flash Update PreCheck (DDh)* command.
5. Erase existing pattern data using *Write Erase Flash Data (E0h)* command.
6. Set flash data length to 1024 bytes using the *Write Flash Data Length (DFh)* command.
7. Write the first 1024 bytes of pattern data using *Write Flash Start (E1h)* command.
8. Write the remaining data in blocks of 1024 bytes using the *Write Flash Continue (E2h)* command.

NOTE: If the size of the pattern data is not a multiple of 1024 bytes, the last Write Flash Continue command will have less than 1024 bytes. In that case, use the *Write Flash Data Length (DFh)* command to update the Flash Data Length prior to the last Write Flash Continue command.

Note that the flash commands described in this section cannot be used within batch files.

3.7.1 Read Flash Update PreCheck (DDh)

This command is used to verify that a pending flash update is appropriate for the specified flash data type.

3.7.1.1 Read Parameters

Parameter Bytes	Description
Byte 1	Flash Build Data Size (LSB)
Byte 2	Flash Build Data Size
Byte 3	Flash Build Data Size
Byte 4	Flash Build Data Size (MSB)

3.7.1.2 Return Parameters

Parameter Bytes	Description
Byte 1	Flash PreCheck Result

Byte 1 - Flash PreCheck Result

b(0) Package Size Error
'0': No Error
'1': Error

b(7:1) *Reserved*

1. This command is used in conjunction with the *Write Flash Data Type Select (DEh)* command. This command would be sent after the flash data type has been selected, but before any other flash operation. The purpose is to verify that the desired flash update is compatible, and will fit within the existing flash space, for the current flash configuration.
2. The Flash Build Data Size specifies the size of the flash update package in bytes.
3. A Package Size error indicates that the flash package is too large to fit into the specified location. For example, this bit will be set if the size of the new internal pattern data is larger than that of the data currently present in flash.
4. If an error is returned by this command, the user is responsible for correcting the error before updating the flash. If the user chooses to ignore this error and proceed with updating the flash, the system will allow this. In this case, the user is responsible for any problems or system behaviors that arise thereafter.

3.7.2 Write Flash Data Type Select (DEh)

This command is used to specify the type of data that will be written to the Flash. In DLPC347x the only flash data type supported for update through I2C is internal pattern data.

3.7.2.1 Write Parameters

Parameter Bytes	Description
Byte 1	Flash Data Type (D0h for pattern data)
Byte 2	*Reserved* (00h for pattern data)
Byte 3	*Reserved* (00h for pattern data)
Byte 4	*Reserved* (00h for pattern data)

3.7.3 Write Flash Data Length (DFh)

This command is used to specify the length of the data that will be written to the Flash in bytes.

3.7.3.1 Write Parameters

Parameter Bytes	Description
Byte 1	Flash Data Length (LSB)
Byte 2	Flash Data Length (MSB)

Default: 0000h

1. Flash data length must be in multiples of four bytes.
2. The flash data length applies to each write transaction, not to the length of the data type selected.
3. The maximum data length allowed for each write transaction is 1024 bytes.

3.7.4 Write Erase Flash Data (E0h)

This command erases the specified block of data in Flash.

3.7.4.1 Write Parameters

Parameter Bytes	Description
Byte 1	Signature: Value = AAh
Byte 2	Signature: Value = BBh
Byte 3	Signature: Value = CCh
Byte 4	Signature: Value = DDh

1. When this command is executed, the system will erase all sectors associated with the data type specified by the *Write Flash Data Type Select* (Section 3.7.2) command. As such, this command does not make use of the Flash Data Length parameter.
2. The signature bytes are used to minimize unintended flash erases. The command OpCode and four signature bytes must be received correctly before this command will be recognized and executed.

NOTE: Since the process of erasing Flash sectors can take a significant amount of time, the Flash Erase Complete status bit in the *Read Short Status* command (Section 3.6.1) should be checked periodically (not continuously) to determine when this task has been completed. This bit will be set at the start of the erase process, and will be cleared when the erase process is complete. Flash writes should not be started before the erase process has been completed.

3.7.5 2.3.5.49 Write Flash Start (E1h)

This command is used to start writing data to Flash.

3.7.5.1 Write Parameters

Parameter Bytes	Description
Byte 1	Data Byte 1
Byte 2	Data Byte 2
Byte 3	Data Byte 3
Byte 4	Data Byte 4
Byte 5 … n	Data Byte 5 … n

1. The *Write Flash Data Length* command must be used to specify how many bytes of data will be sent by this command.
2. The *Write Flash Start* command is used to write up to 1024 bytes of data starting at the first address of the data type selected. If more than 1024 bytes are to be written, the *Write Flash Continue* command must be used. Up to 1024 bytes of data can be written with each *Write Flash Continue* command, which starts at the end of the last data written.
3. The Flash Error bit of the *Read Short Status* command will indicate if the Flash update was successful. This bit will be set for an error at the end of each write transaction, however, once an error has been detected, this bit will remain in the error state until a new data type is selected (selecting a new data type will clear this bit). This will allow the user the option of checking the status between each write transaction, or at the end of the update of a specific data type. Once a write transaction has started, the flash status (and this error bit) will not be accessible until the write transaction has completed.

3.7.6 2.3.5.49 Write Flash Continue (E2h)

This command is used if more than 1024 bytes of data has to be written to Flash.

3.7.6.1 Write Parameters

Parameter Bytes	Description
Byte 1	Data Byte 1
Byte 2	Data Byte 2
Byte 3	Data Byte 3
Byte 4	Data Byte 4
Byte 5 … n	Data Byte 5 … n

1. The *Write Flash Data Length* command must be used to specify how many bytes of data will be sent by this command.
2. The *Write Flash Start* command is used to write up to 1024 bytes of data starting at the first address of the data type selected. If more than 1024 bytes are to be written, the *Write Flash Continue* command must be used. Up to 1024 bytes of data can be written with each *Write Flash Continue* command, which starts at the end of the last data written.
3. The Flash Error bit of the *Read Short Status* command will indicate if the Flash update was successful. This bit will be set for an error at the end of each write transaction, however, once an error has been detected, this bit will remain in the error state until a new data type is selected (selecting a new data type will clear this bit). This will allow the user the option of checking the status between each write transaction, or at the end of the update of a specific data type. Once a write transaction has started, the flash status (and this error bit) will not be accessible until the write transaction has completed.

Revision History

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

Changes from Original (October 2018) to A Revision	Page
• Updated Figure 3	7
• Added Flash Update Commands to Section 3.7	9
• Corrected title of command 06h	12
• Deleted legacy source formats in Section 3.1.3.1	13
• Added 4 missing parameter bytes to Section 3.1.12.1	27
• Corrected Description of Return Parameters in Section 3.1.13.2	28
• Added Table 12	41
• Added section on setting up Light Control modes - Section 3.4	63
• Added Zero Dark Time Support bit in 9Dh Return Parameters	76
• Added new DMD Device IDs	89
• Added command D6h to Read System Temperature using external thermistor	90
• Added command DBh to specify Flash Batch File Delay	92

IMPORTANT NOTICE AND DISCLAIMER