



## Abstract

This is the programmer's guide for the [DLPC3439](#) controller used with the [DLP4710](#) DMD. This guide primarily discusses the I<sup>2</sup>C interface of these controllers. For additional information please visit the desired device product folder on ti.com.

## Table of Contents

<b>Abstract</b> .....	1
<b>1 Introduction</b> .....	2
1.1 System Overview.....	2
1.2 Software Overview.....	3
1.3 Related Documentation.....	4
<b>2 System Initialization</b> .....	5
2.1 Boot ROM.....	5
2.2 Device Startup.....	5
<b>3 I<sup>2</sup>C Commands</b> .....	7
3.1 General Operation Commands.....	10
3.2 Illumination Control Commands.....	41
3.3 Image Processing Control Commands.....	47
3.4 General Setup Commands.....	52
3.5 Administrative Commands.....	56
3.6 Flash Update Commands.....	66
<b>4 Revision History</b> .....	73

## Trademarks

Pico™ is a trademark of Texas Instruments.  
DLP® is a registered trademark of Texas instruments.  
Arm® and Cortex® are registered trademarks of Arm Limited.  
All trademarks are the property of their respective owners.



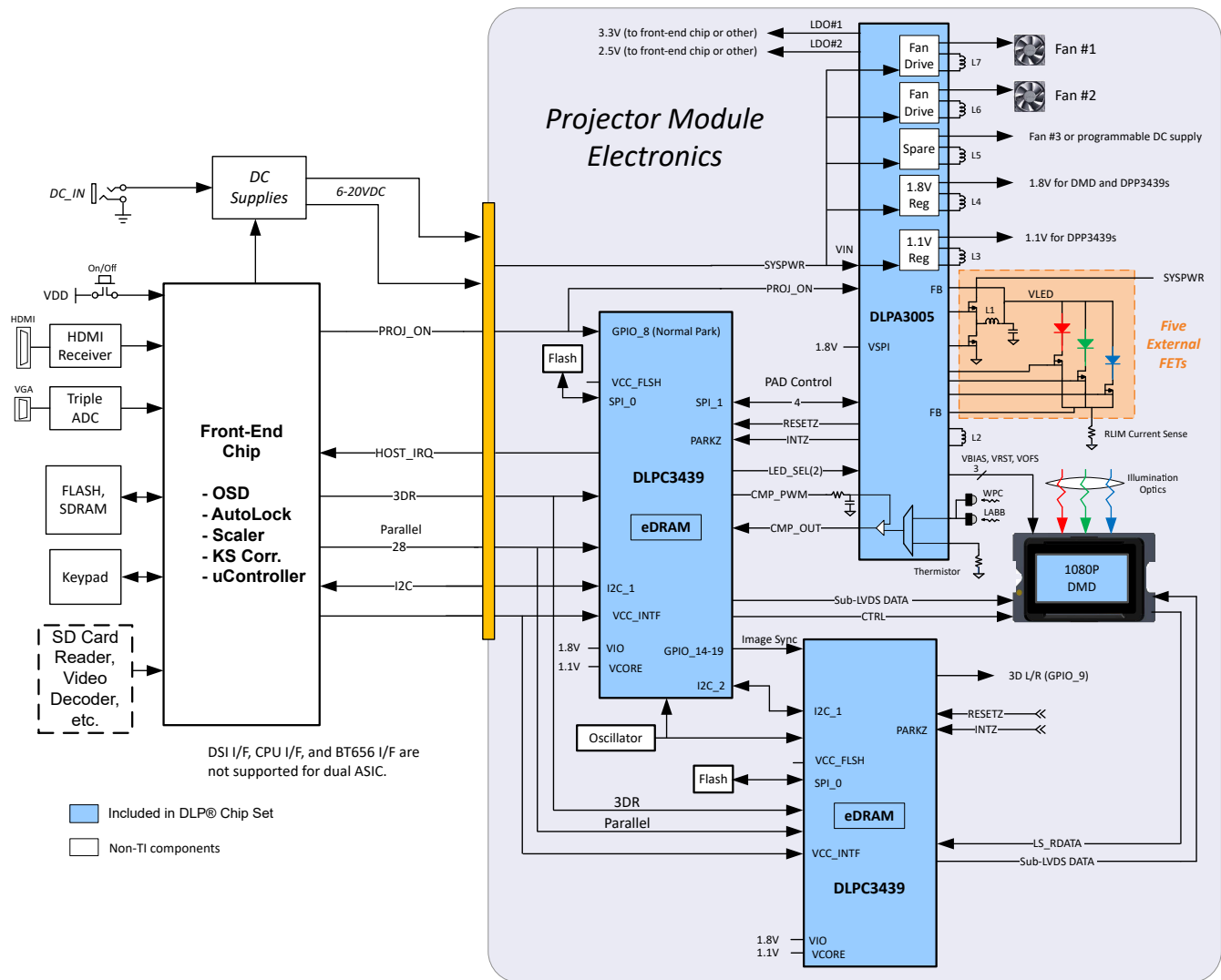


Figure 1-2. DLPC3439 Accessory Configuration with DLPA3005

## 1.2 Software Overview

The DLPC343x controller contains an Arm® Cortex®-M3 processor with additional functional blocks to enable video processing and control. TI provides software as a firmware image. The firmware consists of the main application code (used by the Arm processor) along with other configuration and operational data required by the system for normal operation. The controller and its accompanying DLP chipset components require this proprietary software to operate.

The firmware must be programmed into the SPI flash memory. The DLPC343x controller loads the main application into the Arm processor which periodically accesses the operational data. The available controller functions depend on the firmware version installed. Different firmware is required for different chipset combinations (such as when using different PMIC devices). Visit the applicable controller product folder on ti.com, visit the [DLP Pico Firmware Selector](#), or contact TI for the latest firmware.

### 1.2.1 I<sup>2</sup>C Overview

The protocol used in communicating information to DLPC343x controller consist of a serial data bus conforming to the Philips I<sup>2</sup>C specification. The controller can be configured at runtime by using these I<sup>2</sup>C commands. The DLPC343x behaves as an I<sup>2</sup>C follower operating at up to 100 kHz.

### 1.2.2 I<sup>2</sup>C Transactions

Since all I<sup>2</sup>C commands are processed by software, only one type of I<sup>2</sup>C transaction is supported. This transaction type is shown in [Table 1-1](#) for both writes and reads. The I<sup>2</sup>C interface supports variably-sized transactions (for example, a one byte transaction or a nine byte transaction) to match the commands discussed later in this document.

**Table 1-1. I<sup>2</sup>C Write and Read Transactions**

Transaction		Address <sup>(1)</sup>	Sub-Address <sup>(2)</sup>	Remaining Data Bytes <sup>(3)</sup>
Write	Size	8-bits	8-bits	8-bit parameter bytes (0 → N)
	Value	36h (or 3Ah)	Command value	Parameter values
Read Request	Size	8-bits	8-bits	8-bit parameter bytes (0 → N)
	Value	36h (or 3Ah)	Command value	Parameter values
Read Response	Size	8-bits		8-bit parameter bytes (0 → N)
	Value	37h (or 3Bh)		Parameter values

- (1) The address corresponds to the chip address of the controller. The address is dependent upon the firmware image with the default value of 36h.
- (2) The subaddress corresponds to a TI command.
- (3) The data (if present) corresponds to any required command parameters.

### 1.2.3 Data Flow Control

While the I<sup>2</sup>C interface inherently supports flow control by holding the clock, this is not sufficient for all transactions (for example, sequence and some other updates). In this case, the host software will need to execute the *Read Short Status* command to determine if the system is busy.

### 1.3 Related Documentation

- [DLPC3439 Display Controller Data Sheet](#)
- [DLP4710 0.47 1080p DMD Data Sheet](#)
- [DLPA3000 Power Management and LED/Lamp Driver IC Data Sheet](#)
- [DLPA3005 Power Management and LED/Lamp Driver IC Data Sheet](#)

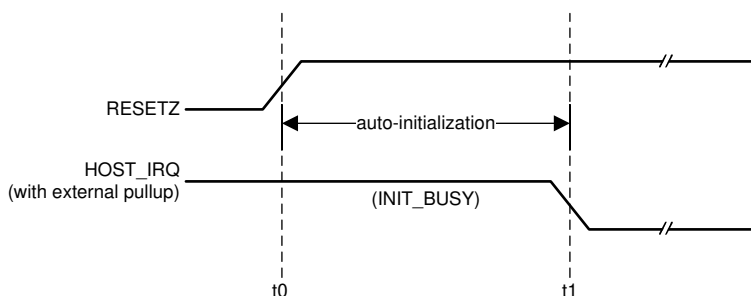
## 2 System Initialization

### 2.1 Boot ROM

The DLPC343x 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 DLPC343x 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.

### 2.2 Device Startup

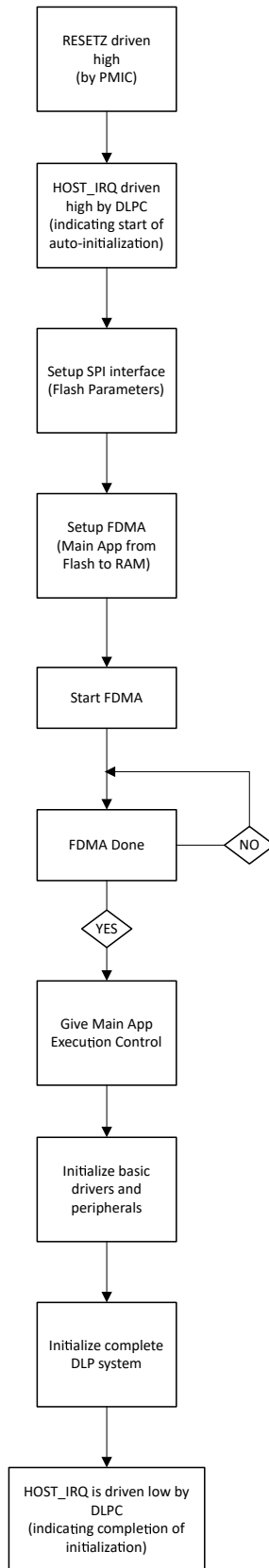
- The HOST\_IRQ signal is provided to indicate when the system has completed auto-initialization.
- While reset is applied, HOST\_IRQ is tri-stated (an external pullup resistor pulls the line high).
- HOST\_IRQ remains tri-stated (pulled high externally) until the boot process completes. While the signal is pulled high, this indicates that the controller is performing boot-up and auto-initialization.
- As soon as possible after the controller boots-up, the controller drives HOST\_IRQ to a logic high state to indicate that the controller is continuing to perform auto-initialization (no real state changes occur on the external signal).
- The software sets HOST\_IRQ to a logic low state at the completion of the auto-initialization process. At the falling edge of the signal, the initialization is complete.
- The DLPC34xx controller is ready to receive commands through I<sup>2</sup>C or accept video over the DSI or the parallel interface only after auto-initialization is complete.
- The controller initialization typically completes (HOST\_IRQ goes low) within 500 ms of RESETZ being asserted. However, this time may vary depending on the software version and the contents of the user configurable auto initialization file.



t0: rising edge of RESETZ; auto-initialization begins

t1: falling edge of HOST\_IRQ; auto-initialization is complete

**Figure 2-1. HOST\_IRQ Timing**



**Figure 2-2. Startup Flow**

### 3 I<sup>2</sup>C Commands

**Table 3-1. List of System Write/Read Software Commands**

Command Type	Command Description	OpCode (hex)	Reference
<b>General Operation Commands</b>			
Write	Write Input Source Select	05	<a href="#">Write Input Source Select (05h)</a>
Read	Read Input Source Select	06	<a href="#">Read Input Source Select (06h)</a>
Write	Write External Video Source Format Select	07	<a href="#">Write External Video Source Format Select (07h)</a>
Read	Read External Video Source Format Select	08	<a href="#">Read External Video Source Format Select (08h)</a>
Write	Write External Video Chroma Processing Select	09	<a href="#">Write External Video Chroma Processing Select (09h)</a>
Read	Read External Video Chroma Processing Select	0A	<a href="#">Read External Video Chroma Processing Select (0Ah)</a>
Write	Write Test Pattern Select	0B	<a href="#">Write Test Pattern Select (0Bh)</a>
Read	Read Test Pattern Select	0C	<a href="#">Read Test Pattern Select (0Ch)</a>
Write	Write Splash Screen Select	0D	<a href="#">Write Splash Screen Select (0Dh)</a>
Read	Read Splash Screen Select	0E	<a href="#">Read Splash Screen Select (0Eh)</a>
Read	Read Splash Screen Header	0F	<a href="#">Read Splash Screen Header (0Fh)</a>
Write	Write Display Size	12	<a href="#">Write Display Size (12h)</a>
Read	Read Display Size	13	<a href="#">Read Display Size (13h)</a>
Write	Write Display Image Orientation	14	<a href="#">Write Display Image Orientation (14h)</a>
Read	Read Display Image Orientation	15	<a href="#">Read Display Image Orientation (15h)</a>
Write	Write Display Image Curtain	16	<a href="#">Write Display Image Curtain (16h)</a>
Read	Read Display Image Curtain	17	<a href="#">Read Display Image Curtain (17h)</a>
Write	Write Image Freeze	1A	<a href="#">Write Image Freeze (1Ah)</a>
Read	Read Image Freeze	1B	<a href="#">Read Image Freeze (1Bh)</a>
Write	Write 3-D Control	20	<a href="#">Write 3-D Control (20h)</a>
Read	Read 3-D Control	21	<a href="#">Read 3-D Control (21h)</a>
Write	Write Look Select	22	<a href="#">Write Look Select (22h)</a>
Read	Read Look Select	23	<a href="#">Read Look Select (23h)</a>
Read	Read Sequence Header Attributes	26	<a href="#">Read Sequence Header Attributes (26h)</a>
Write	Write Degamma/CMT Select	27	<a href="#">Write Degamma/CMT Select (27h)</a>
Read	Read Degamma/CMT Select	28	<a href="#">Read Degamma/CMT Select (28h)</a>
Write	Write CCA Select	29	<a href="#">Write CCA Select (29h)</a>
Read	Read CCA Select	2A	<a href="#">Read CCA Select (2Ah)</a>
Read	Read DMD Sequencer Sync Mode	2C	<a href="#">Read DMD Sequencer Sync Mode (2Ch)</a>
Write	Write Execute Flash Batch File	2D	<a href="#">Write Execute Flash Batch File (2Dh)</a>
Write	Write Input Image Size	2E	<a href="#">Write Input Image Size (2Eh)</a>
Read	Read Input Image Size	2F	<a href="#">Read Input Image Size (2Fh)</a>
Write	Write 3-D Reference	30	<a href="#">Write 3-D Reference (30h)</a>
Write	Write Splash Screen Execute	35	<a href="#">Write Splash Screen Execute (35h)</a>
Write	Write Parallel Data Mask Control	37	<a href="#">Write Parallel Data Mask Control (37h)</a>
Read	Read Parallel Data Mask Control	38	<a href="#">Read Parallel Data Mask Control (38h)</a>
Write	Write Mirrors Lock Command	39	<a href="#">Write Mirrors Lock Command (39h)</a>
Read	Read Mirrors Lock Command	3A	<a href="#">Read Mirrors Lock Command (3Ah)</a>
<b>Illumination Control Commands</b>			
Write	Write LED Output Control Method	50	<a href="#">Write LED Output Control Method (50h)</a>

**Table 3-1. List of System Write/Read Software Commands (continued)**

Command Type	Command Description	OpCode (hex)	Reference
Read	Read LED Output Control Method	51	<a href="#">Read LED Output Control Method (51h)</a>
Write	Write RGB LED Enable	52	<a href="#">Write RGB LED Enable (52h)</a>
Read	Read RGB LED Enable	53	<a href="#">Read RGB LED Enable (53h)</a>
Write	Write RGB LED Current	54	<a href="#">Write RGB LED Current (54h)</a>
Read	Read RGB LED Current	55	<a href="#">Read RGB LED Current (55h)</a>
Read	Read CAIC LED Max Available Power	57	<a href="#">Read CAIC LED Max Available Power (57h)</a>
Write	Write RGB LED Max Current	5C	<a href="#">Write RGB LED Max Current (5Ch)</a>
Read	Read RGB LED Max Current	5D	<a href="#">Read RGB LED Max Current (5Dh)</a>
Read	Read CAIC RGB LED Current	5F	<a href="#">Read CAIC RGB LED Current (5Fh)</a>
<b>Image Processing Control Commands</b>			
Write	Write Local Area Brightness Boost Control	80	<a href="#">Write Local Area Brightness Boost Control (80h)</a>
Read	Read Local Area Brightness Boost Control	81	<a href="#">Read Local Area Brightness Boost Control (81h)</a>
Write	Write CAIC Image Processing Control	84	<a href="#">Write CAIC Image Processing Control (84h)</a>
Read	Read CAIC Image Processing Control	85	<a href="#">Read CAIC Image Processing Control (85h)</a>
Write	Write Color Coordinate Adjustment Control	86	<a href="#">Write Color Coordinate Adjustment Control (86h)</a>
Read	Read Color Coordinate Adjustment Control	87	<a href="#">Read Color Coordinate Adjustment Control (87h)</a>
<b>General Setup Commands</b>			
Write	Write Border Color	B2	<a href="#">Write Border Color (B2h)</a>
Read	Read Border Color	B3	<a href="#">Read Border Color (B3h)</a>
Write	Write Parallel Interface Sync Polarity	B6	<a href="#">Write Parallel Interface Sync Polarity (B6h)</a>
Read	Read Parallel Interface Sync Polarity	B7	<a href="#">Read Parallel Interface Sync Polarity (B7h)</a>
Read	Read Auto Framing Information	BA	<a href="#">Read Auto Framing Information (BAh)</a>
<b>Administrative Commands</b>			
Read	Read Short Status	D0	<a href="#">Read Short Status (D0h)</a>
Read	Read System Status	D1	<a href="#">Read System Status (D1h)</a>
Read	Read System Software Version	D2	<a href="#">Read System Software Version (D2h)</a>
Read	Read Communication Status	D3	<a href="#">Read Communication Status (D3h)</a>
Read	Read Controller Device ID	D4	<a href="#">Read Controller Device ID (D4h)</a>
Read	Read DMD Device ID	D5	<a href="#">Read DMD Device ID (D5h)</a>
Read	Read System Temperature	D6	<a href="#">Read System Temperature (D6h)</a>
Read	Read Flash Build Version	D9	<a href="#">Read Flash Build Version (D9h)</a>
Write	Write Flash Batch File Delay	DB	<a href="#">Write Flash Batch File Delay (DBh)</a>
Read	Read DMD I/F Training Data	DC	<a href="#">Read DMD I/F Training Data (DCh)</a>
<b>Flash Update Commands</b>			
Read	Read Flash Update PreCheck	DD	<a href="#">Read Flash Update PreCheck (DDh)</a>
Write	Write Flash Data Type Select	DE	<a href="#">Write Flash Data Type Select (DEh)</a>
Write	Write Flash Data Length	DF	<a href="#">Write Flash Data Length (DFh)</a>
Write	Write Erase Flash Data	E0	<a href="#">Write Erase Flash Data (E0h)</a>
Write	Write Flash Start	E1	<a href="#">Write Flash Start (E1h)</a>
Write	Write Flash Continue	E2	<a href="#">Write Flash Continue (E2h)</a>



**Table 3-1. List of System Write/Read Software Commands (continued)**

Command Type	Command Description	OpCode (hex)	Reference
Read	Read Flash Start	E3	<a href="#">Read Flash Start (E3h)</a>
Read	Read Flash Continue	E4	<a href="#">Read Flash Continue (E4h)</a>

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

### 3.1 General Operation Commands

#### 3.1.1 Write Input Source Select (05h)

This command selects the input source of the system.

##### 3.1.1.1 Write Parameters

Table 3-2 describes the write parameters.

**Table 3-2. Write Parameters**

MSB	Byte 1						LSB
b7	b6	b5	b4	b3	b2	b1	b0

Byte 1	Operating Mode
00h	External Video Mode
01h	Test Pattern Generator Mode
02h	Splash Screen Mode
03h - FFh	Reserved

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:

- External Video Port:
  - [Write Input Image Size \(2Eh\)](#)
  - [Write External Video Source Format Select \(07h\)](#)
  - [Write External Video Chroma Processing Select \(09h\)](#)
- Test Pattern Generator:
  - [Write Test Pattern Select \(0Bh\)](#)
- Splash Screen:
  - [Write Splash Screen Select \(0Dh\)](#)
  - [Write Splash Screen Execute \(35h\)](#)

---

**Note**

Only RGB (and not YCbCr) format is supported in splash pattern mode.

---

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 3-3. Source Specific Associated Commands**

Source Specific Associated Commands	Display Modes		
	External Video Port	Test Pattern Generator	Splash Screen
Write External Video Source Format Select	Y	N	N
Write External Video Chroma Processing Select	Y	N	N
Write External Input Image Size	Y	N	N
Write Test Pattern Select	N	Y	N
Write Splash Screen Select	N	N	Y
Write Splash Screen Execute	N	N	Y <sup>(1)</sup>

- (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. See below 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 Write Image Crop (10h). The controller displays these settings at the resolution specified by Write Display Size (12h). Therefore, the scale settings must be changed. The controller software manages the underlying hardware settings. An example is shown below:

1. User sends the following commands (active Input Source = Test Pattern Generator)
  - a. Write Image Freeze = Freeze
  - b. Write External Video Source Format Select (settings stored, command not executed)
  - c. Write External Video Chroma Processing Select (settings stored, command not executed)
  - d. Write External Input Image Size (settings stored, command not executed)
  - e. Write Operating Mode Select = External Port (see the next step)
  - f. Write Image Freeze = Unfreeze
2. When the Write Operating Mode Select command is received, software applies the settings from these External Video Port associated commands.
  - a. External Video Source Format Select
  - b. External Video Chroma Processing Select
  - c. External Input Image Size
  - d. External Parallel Manual Image Framing (as appropriate – that is, if parallel port selected)
  - e. External CPU Video Sync Mode (as appropriate – that is, if CPU port selected)

If source associated commands are sent for a source that is already active, the controller software will execute these commands when received. An example is shown below:

1. User sends the following commands (active Input Source = External Video Port)
  - a. Write Image Freeze = Freeze
  - b. Write External Video Source Format Select (command executed)
  - c. Write External Video Chroma Processing Select (command executed)
  - d. Write Image Freeze = Unfreeze

Refer to Write Image Freeze (1Ah) for information on hiding on-screen artifacts when selecting an input source.

### 3.1.2 Read Input Source Select (06h)

This command reads the input source of the system.

#### 3.1.2.1 Read Parameters

This command has no read parameters.

#### 3.1.2.2 Return Parameters

Table 3-5 describes the return parameters.

Byte 1	Operating Mode
00h	External Video Mode
01h	Test Pattern Generator Mode
02h	Splash Screen Mode
03h - FFh	Reserved

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

Table 3-4 describes the write parameters.

**Table 3-4. Write Parameters**

Parameter Bytes			Description			
Byte 1			External Video Source Format			
CMD Parameter	Port	Bits/Pixel	Data Type	Bus Width	Clks/Pixel	Notes
40h	Parallel	16	RGB 565	16	1	Auto-select RGB CSC
41h	Parallel	18	RGB 666	18	1	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 Auto-select 4:2:2 → 4:4:4
61h	Parallel	16	YCbCr 4:2:2 88	16	1	Auto-select YCbCr CSC Auto-select 4:2:2 → 4:4:4

This command is used in conjunction with the *Write Input Source Select* command. This command specifies which input port displays when the *Write Input Source Select* command selects external video port as the image source. The settings for this command are retained until changed using this command. These settings are automatically applied each time the external video port is selected.

When the external video port is selected as the input source, the software automatically selects and loads the proper CSC, based on the selected parameter of this command (appropriate matrix for RGB, selected matrix for YCbCr including offset). The appropriate data path is also automatically selected for 4:2:2 versus 4:4:4 processing.

The user must review the notes for the *Write Input Source Select* command to understand the concept of source-associated commands. This concept determines when source-associated commands are executed by the system. This command is a source-associated command.

### 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 read parameters.

#### 3.1.4.2 Return Parameters

Table 3-5 describes the return parameters.

**Table 3-5. Return Parameters**

Parameter Bytes			Description			
Byte 1			External Video Source Format			
Command Parameter	Port	Bits/Pixel	Data Type	Bus Width	Clks/Pixel	Notes
40h	Parallel	16	RGB 565	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

Command Parameter	Port	Bits/Pixel	Data Type	Bus Width	Clks/Pixel	Notes
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 Auto-select 4:2:2 → 4:4:4
61h	Parallel	16	YCbCr 4:2:2 88	16	1	Auto-select YCbCr CSC Auto-select 4:2:2 → 4:4:4

### 3.1.5 Write External Video Chroma Processing Select (09h)

This command specifies the active external video chroma processing type for the display module.

#### 3.1.5.1 Write Parameters

Table 3-6 describes the write parameters.

**Table 3-6. Write Parameters**

Parameter Bytes	Description
Byte 1	Chroma
Byte 2	CSC coefficient set

MSB	Byte 1						LSB
b7	b6	b5	b4	b3	b2	b1	b0

b(7:5)	Reserved
b(4)	Chroma Interpolation Method: <ul style="list-style-type: none"> <li>0: Chroma Interpolation</li> <li>1: Chroma Copy</li> </ul>
b(3)	Reserved
b(2)	Chroma Channel Swap: <ul style="list-style-type: none"> <li>0: CbCr</li> <li>1: CrCb</li> </ul>
b(0:1)	Reserved

MSB	Byte 2						LSB
b7	b6	b5	b4	b3	b2	b1	b0

b(7:2)	Reserved
b(1:0)	CSC Coefficient Set (color space)

#### Note

CSC coefficient sets are specified in Byte 2 by an enumerated value (i.e. 0,1,2, or 3). The set stored in '0' is ITU-R BT. Rec. 601. The other three sets are user definable

### 3.1.6 Read External Video Chroma Processing Select (0Ah)

This command is used to specify the characteristics of the selected YCbCr source, as well as specifying the type of chroma processing to be used for this YCbCr source by the display module.

#### 3.1.6.1 Read Parameters

This command has no read parameters.

### 3.1.6.2 Return Parameters

Table 3-7 describes the return parameters.

**Table 3-7. Return Parameters**

Parameter Bytes		Description	
Byte 1		Chroma	
Byte 2		CSC coefficient set	

MSB	Byte 1						LSB
b7	b6	b5	b4	b3	b2	b1	b0
b(7:5)	Reserved						
b(4)	Chroma Interpolation Method: <ul style="list-style-type: none"> <li>0: Chroma Interpolation</li> <li>1: Chroma Copy</li> </ul>						
b(3)	Reserved						
b(2)	Chroma Channel Swap: <ul style="list-style-type: none"> <li>0: CbCr</li> <li>1: CrCb</li> </ul>						
b(0:1)	Reserved						

MSB	Byte 2						LSB
b7	b6	b5	b4	b3	b2	b1	b0
b(7:2)	Reserved						
b(1:0)	CSC Coefficient Set (color space)						

#### Note

CSC coefficient sets are specified in Byte 2 by an enumerated value (i.e. 0,1,2, or 3). The set stored in '0' is ITU-R BT. Rec. 601. The other three sets are user definable

### 3.1.7 Write Test Pattern Select (0Bh)

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

#### 3.1.7.1 Write Parameters

Table 3-8 describes the write parameters.

**Table 3-8. Write Parameters**

Parameter Bytes	Description
Byte 1	TPG pattern select
Byte 2	Foreground and background color (see Table 3-11)
Byte 3	Parameter 1 (see Table 3-12)
Byte 4	Parameter 2 (see Table 3-12)
Byte 5	Parameter 3 (see Table 3-12)
Byte 6	Parameter 4 (see Table 3-12)

**Table 3-9. Byte 1 Write Parameter**

MSB	Byte 1						LSB
b7	b6	b5	b4	b3	b2	b1	b0

b(7)	Test pattern border: <ul style="list-style-type: none"> <li>• 00h: Disabled</li> <li>• 01h: Enabled</li> </ul>
b(6:4)	Reserved
b(3:0)	Left pattern select: <ul style="list-style-type: none"> <li>• 00h: Solid field</li> <li>• 01h: Fixed step horizontal ramp</li> <li>• 02h: Fixed step vertical ramp</li> <li>• 03h: Horizontal lines</li> <li>• 04h: Diagonal lines</li> <li>• 05h: Vertical lines</li> <li>• 06h: Horizontal and vertical grid</li> <li>• 07h: Checkerboard</li> <li>• 08h: Color bars</li> <li>• 09h-0Fh: Reserved</li> </ul>

**Table 3-10. Byte 2 Write Parameter**

MSB	Byte 2						LSB
	b7	b6	b5	b4	b3	b2	
b(7)	Reserved						
b(6:4)	Foreground color: <ul style="list-style-type: none"> <li>• 0h: Black</li> <li>• 1h: Red</li> <li>• 2h: Green</li> <li>• 3h: Blue</li> <li>• 4h: Cyan</li> <li>• 5h: Magenta</li> <li>• 6h: Yellow</li> <li>• 7h: White</li> </ul>						
b(3)	Reserved						
b(2:0)	Background color: <ul style="list-style-type: none"> <li>• 0h: Black</li> <li>• 1h: Red</li> <li>• 2h: Green</li> <li>• 3h: Blue</li> <li>• 4h: Cyan</li> <li>• 5h: Magenta</li> <li>• 6h: Yellow</li> <li>• 7h: White</li> </ul>						

**Table 3-11. 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

**Table 3-11. Foreground and Background Color Use (continued)**

Pattern	Byte 2	
	Foreground Color	Background Color
Color bars	No	No

**Table 3-12. 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		End value	8	Start value	8
Fixed step vertical ramp	N/A		N/A		End value	8	Start 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	3	Number of vertical checkers	8	Number of horizontal checkers	3	Number of horizontal checkers	8
Color bars	N/A		N/A		N/A		N/A	

This command is used in conjunction with the *Write Input Source Select* command. This command specifies which test pattern displays when the *Write Input Source Select* command selects test pattern generator as the image source. The settings for this command are retained until changed using this command. These settings automatically apply each time the test pattern generator is selected.

Batch files are created and stored in flash, and recall the settings for predefined test patterns.

Test patterns are created at the resolution of the display (DMD), are modified by the *Write Image Crop* command, and displayed at the resolution specified by the *Write Display Size* command.

Test patterns display at the default frame rate 60 Hz.

The *Test Pattern Border Selection* creates a white border, a single pixel wide and tall, around the specified test pattern.

The user must review the notes for the *Write Input Source Select* command to understand the concept of source-associated commands. This concept determines when source-associated commands are executed by the system. This command is a source-associated command.

When a foreground or background color is not used, the bit values are ignored (see [Table 3-11](#)). If both foreground and background color are not used, or when a parameter byte (bytes 3 thru 6) is not used, the byte must not be sent. [Table 3-13](#) shows the number of bytes required, based on the specified pattern.

**Table 3-13. Number of Bytes Required based on Pattern Selection**

Specified Pattern	Number of Bytes Required
Solid field	2
Fixed step horizontal ramp	4
Fixed step vertical ramp	4
Horizontal lines	4
Vertical lines	4
Diagonal lines	4
Grid lines	6
Checkerboard	6



**Table 3-13. Number of Bytes Required based on Pattern Selection (continued)**

Specified Pattern	Number of Bytes Required
Color bars	1

As noted in [Table 3-11](#), the color for the solid field pattern is specified using the foreground color. An example of a solid field pattern is shown in [Figure 3-1](#).



**Figure 3-1. Example of Solid Field Test Pattern (Red)**

As noted in [Table 3-11](#), the color for the fixed step horizontal ramp pattern is specified using the foreground color. As noted in [Table 3-12](#), 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 is 5 (1280 pixels / 256 values = 5). Thus, every gray shade value from 0 to 255 has a step size of 5 pixels (such that each step has 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 3-2](#).



**Figure 3-2. Example of Fixed Step Horizontal Ramp Test Pattern**

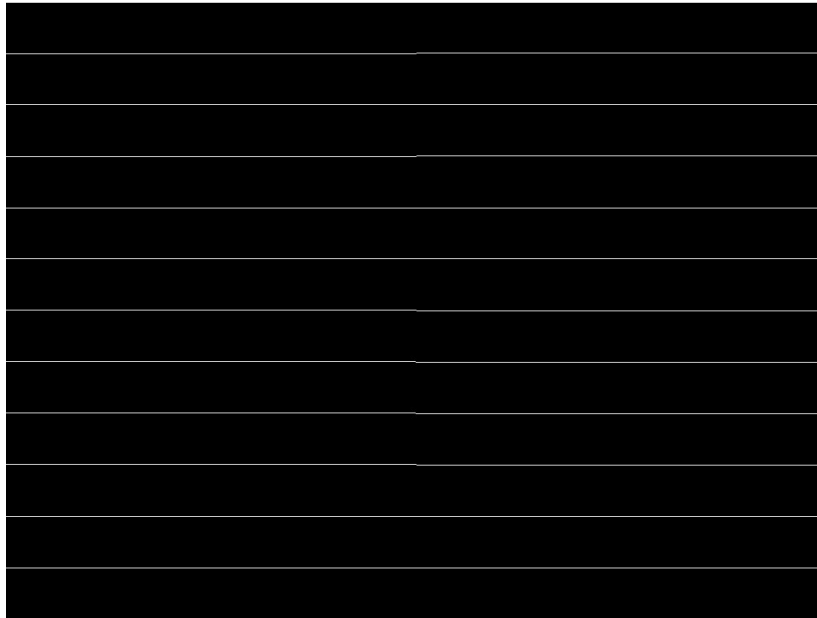
As noted in [Table 3-11](#), the color for the fixed step vertical ramp pattern is specified using the foreground color. As noted in [Table 3-12](#), the user specifies the start value and the stop value for the ramp. For this pattern, the system automatically determines the step size based on the start 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 is 3 (768 pixels / 256 values = 3). Thus, every value from 0 to 255 has a step size of 3 pixels (such that each step has 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 3-3](#).



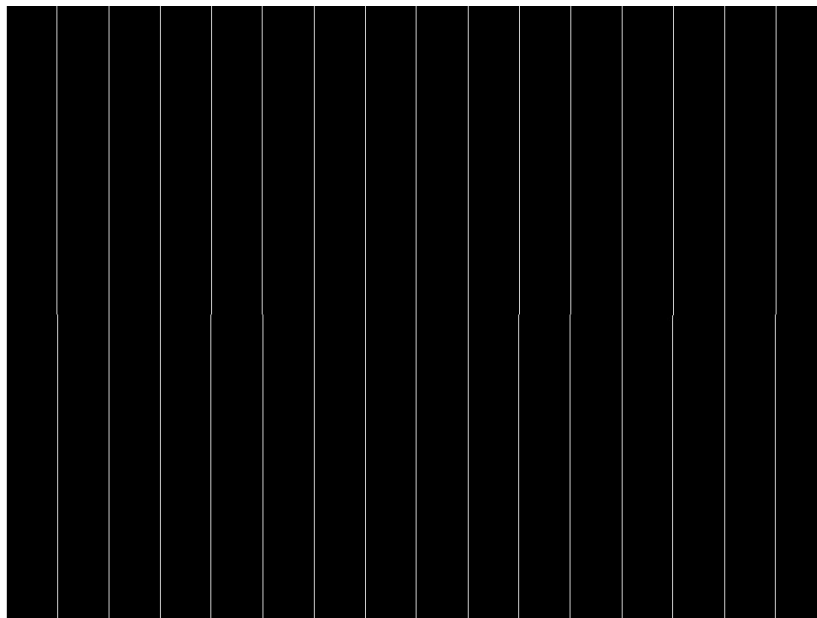
**Figure 3-3. Example of Fixed Step Vertical Ramp Test Pattern**

As noted in [Table 3-11](#), 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 3-12](#), 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 is a single pixel horizontal line on every tenth line. An example of a horizontal lines pattern is shown in [Figure 3-4](#).



**Figure 3-4. Example of Horizontal Lines Test Pattern**

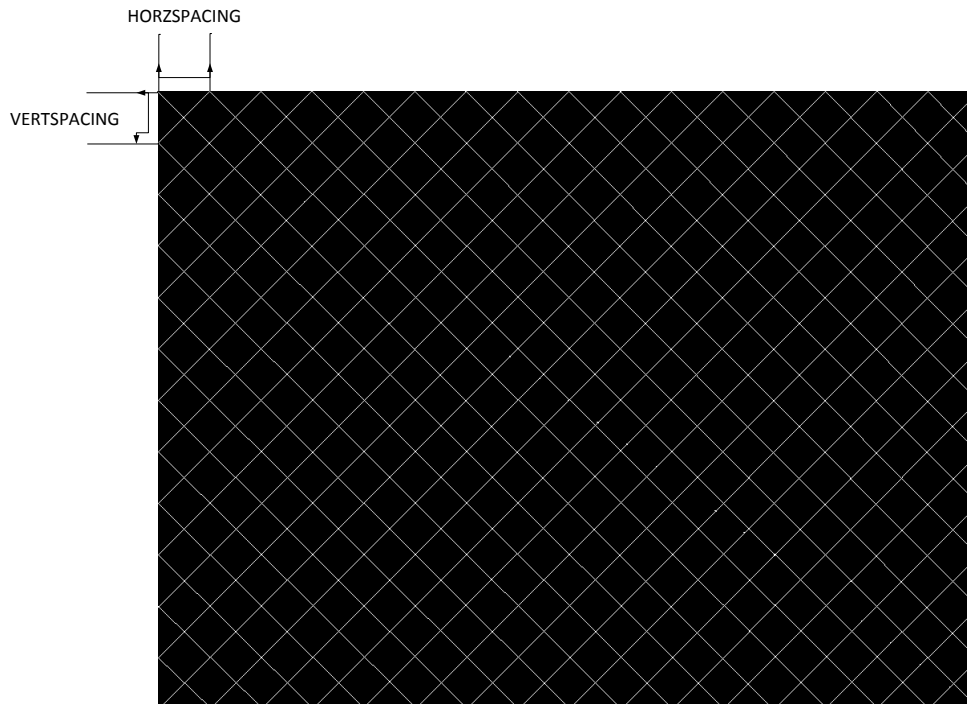
As noted in [Table 3-11](#), the colors for the vertical lines pattern are specified using both the foreground and background colors. The foreground color is used for the vertical lines, and the background color is used for the space between the lines. As noted in [Table 3-12](#), the user specifies 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 is a single pixel vertical line on every tenth line. An example of a vertical lines pattern is shown in [Figure 3-5](#).



**Figure 3-5. Example of Vertical Lines Test Pattern**

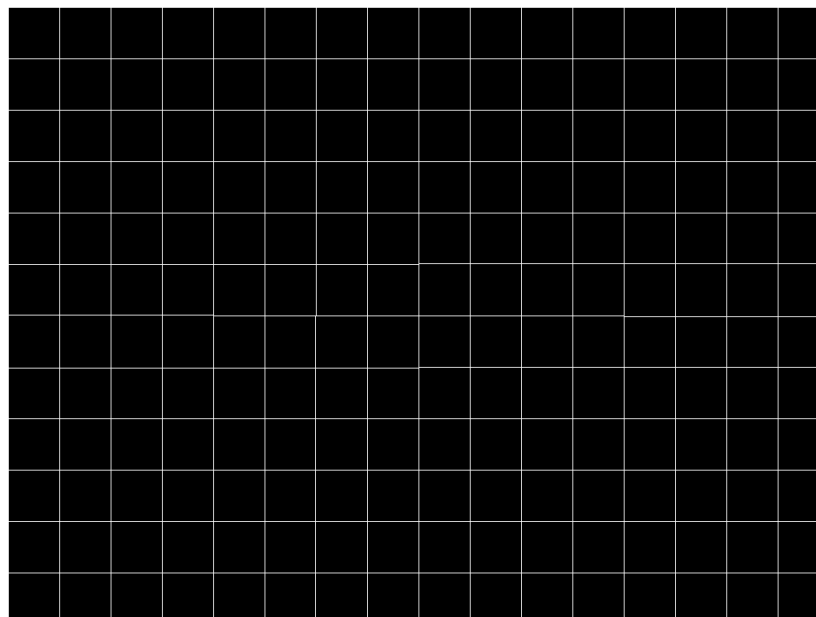
As noted in [Table 3-11](#), the colors for the diagonal lines pattern are specified using both the foreground and background colors. The foreground color is used for the diagonal lines, and the background color is used for the space between the lines. As noted in [Table 3-12](#), the user specifies 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 3-6](#).



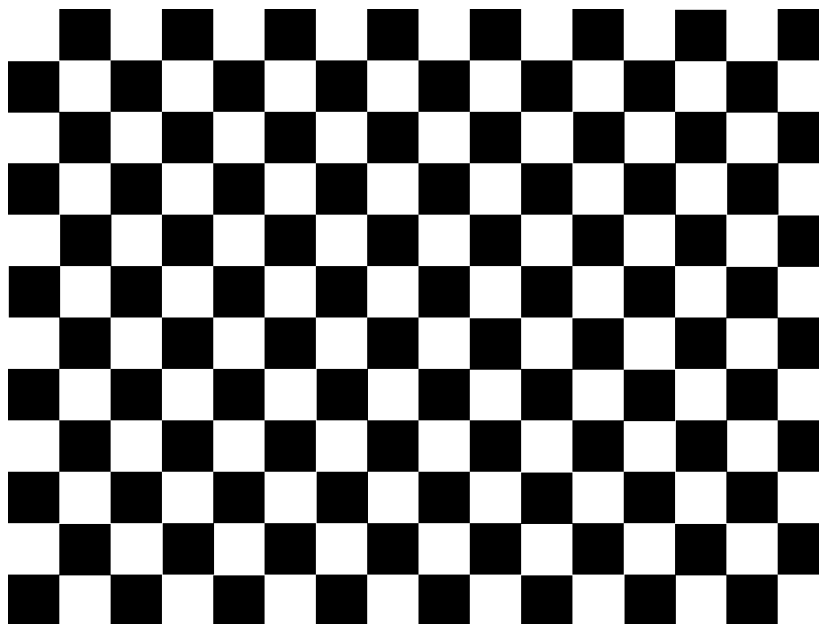
**Figure 3-6. Example of Diagonal Lines Test Pattern**

As noted in [Table 3-11](#), 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 3-12](#), 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 is a single pixel horizontal line on every tenth line. If the vertical foreground line width = 1, and background line width = 9, there is a single pixel vertical line on every tenth line. An example of a grid lines pattern is shown in [Figure 3-7](#).



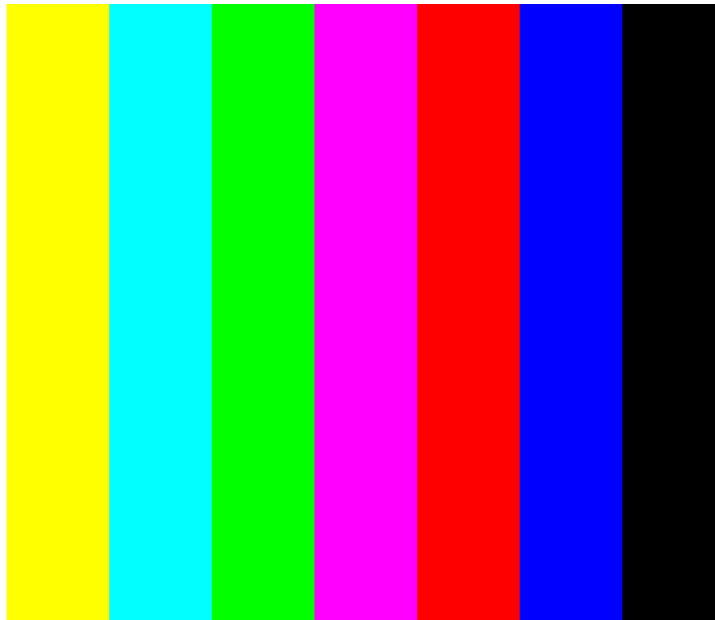
**Figure 3-7. Example of Grid Lines Test Pattern**

As noted in [Table 3-11](#), 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 3-12](#), 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 3-8](#).



**Figure 3-8. Example of Checkerboard Test Pattern**

As noted in [Table 3-11](#) and [Table 3-12](#), 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 3-9](#).



**Figure 3-9. Example of Color Bars Test Pattern**

### 3.1.8 Read Test Pattern Select (0Ch)

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

#### 3.1.8.1 Read Parameters

This command has no read parameters.

#### 3.1.8.2 Return Parameters

[Table 3-14](#) describes the return parameters.

**Table 3-14. Return Parameters**

MSB	Byte 1						LSB
b7	b6	b5	b4	b3	b2	b1	b0

**Table 3-15. Parameter Bytes**

Parameter Bytes	Description
Byte 1	TPG pattern select
Byte 2	Foreground and background color (see <a href="#">Table 3-11</a> )
Byte 3	Parameter 1 (see <a href="#">Table 3-12</a> )
Byte 4	Parameter 2 (see <a href="#">Table 3-12</a> )
Byte 5	Parameter 3 (see <a href="#">Table 3-12</a> )
Byte 6	Parameter 4 (see <a href="#">Table 3-12</a> )

This command always returns six bytes, since the host does not know how many bytes are valid until the pattern is selected. All unnecessary bytes (see [Table 3-13](#)) are set to 0.

If a batch file is used to specify the parameters of the test pattern generator, those parameters are returned by this command.

### 3.1.9 Write Splash Screen Select (0Dh)

This command selects a stored splash screen to be displayed on the display module.

#### 3.1.9.1 Write Parameters

[Table 3-16](#) describes the write parameters.

**Table 3-16. Write Parameters**

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

This command is used in conjunction with the *Write Input Source Select* and the *Write Splash Screen Execute* commands, and specifies which splash screen is selected by the *Input Source Select* command. The settings for this command are retained until changed using this command.

The steps required to display a splash screen are:

1. Select the desired splash screen (using this command)
2. Change the input source to splash screen (using *Write Input Source Select*)
3. Start the splash screen retrieval process (using *Write Splash Screen Execute*).

The splash screen is read from flash and sent down the processing path of the controller once, to be stored in memory for display at the end of the processing path. As such, all image processing settings (such as image crop, image orientation, display size, splash screen select, splash screen as input source, and so forth) must be set by the user before executing the *Write Splash Screen Execute* command.

The user must review the notes for the *Write Input Source Select* command to understand the concept of source-associated commands. This concept determines when source-associated commands are executed by the system. This command is a source-associated command.

The availability of the splash screen is limited by the available space in flash memory. All splash screens must be landscape oriented.

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.

The user must specify how the splash image is displayed on the screen. Key commands for this are *Write Image Crop* and *Write Display Size*.

When this command is received while splash screen is the active source, other than storing the specified splash screen value, the only action taken by the controller software is to obtain the header information from the selected splash screen and store this in internal memory. 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.

### 3.1.10 Read Splash Screen Select (0Eh)

This command reads the state of the *Splash Screen Select* command of the display module.

#### 3.1.10.1 Read Parameters

This command has no read parameters.

#### 3.1.10.2 Return Parameters

[Table 3-17](#) describes the return parameters.

**Table 3-17. Return Parameters**

Parameter Bytes	Description
Byte 1	Splash screen selected (integer)

### 3.1.11 Read Splash Screen Header (0Fh)

This command reads the splash screen header information for the selected splash screen of the display module.

### 3.1.11.1 Read Parameters

The read parameter specifies the splash screen for which the header parameters are returned. If a splash screen value is provided for an unavailable splash screen, this is considered an error (invalid command parameter value – communication status).

**Table 3-18. Read Parameters**

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

### 3.1.11.2 Return Parameters

Table 3-19 describes the return parameters.

**Table 3-19. 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 are referenced in Table 3-20.

**Table 3-20. 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

### 3.1.12 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

Table 3-21 describes the write parameters.

**Table 3-21. 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 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. (that is, 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 is 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 read parameters.

#### 3.1.13.2 Return Parameters

Table 3-22 describes the return parameters.

**Table 3-22. Return Parameters**

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

**Table 3-22. Return Parameters (continued)**

Parameter Bytes	Description
Byte 4	Lines per frame (MSByte)

The parameter values are to be '1' based. (that is, 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

Table 3-23 describes the write parameters.

**Table 3-23. Write Parameters**

MSB	Byte 1						LSB
b7	b6	b5	b4	b3	b2	b1	b0
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)		Reserved					



**Figure 3-10. Long-Axis Flip**

Figure 3-11 shows the short-axis flip.



**Figure 3-11. Short-Axis Flip**

### 3.1.15 Read Display Image Orientation (15h)

This command reads the state of the displayed image orientation function for the display module.

#### 3.1.15.1 Read Parameters

This command has no read parameters.

### 3.1.15.2 Return Parameters

Table 3-24 describes the return parameters.

**Table 3-24. Return Parameters**

MSB	Byte 1						LSB
b7	b6	b5	b4	b3	b2	b1	b0
b(7:3)	Reserved						
b(2)	Short-axis image flip:						
	<ul style="list-style-type: none"> <li>• 0: Image not flipped.</li> <li>• 1: Image flipped.</li> </ul>						
b(1)	Long-axis image flip:						
	<ul style="list-style-type: none"> <li>• 0: Image not flipped.</li> <li>• 1: Image flipped.</li> </ul>						
b(0)	Reserved						

### 3.1.16 Write Display Image Curtain (16h)

This command controls the display image curtain for the display module.

#### 3.1.16.1 Write Parameters

Table 3-25 shows the write parameters.

**Table 3-25. Write Parameters**

MSB	Byte 1						LSB
b7	b6	b5	b4	b3	b2	b1	b0
b(7:4)	Reserved						
b(3:1)	Select curtain color:						
	<ul style="list-style-type: none"> <li>• 0h: Black</li> <li>• 1h: Red</li> <li>• 2h: Green</li> <li>• 3h: Blue</li> <li>• 4h: Cyan</li> <li>• 5h: Magenta</li> <li>• 6h: Yellow</li> <li>• 7h: White</li> </ul>						
b(0)	Curtain enable:						
	<ul style="list-style-type: none"> <li>• 0: Curtain disabled</li> <li>• 1: Curtain enabled</li> </ul>						

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, 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 for the display module.

#### 3.1.17.1 Read Parameters

This command has no read parameters.

#### 3.1.17.2 Return Parameters

Table 3-26 describes the return parameters.

**Table 3-26. Return Parameters**

MSB	Byte 1						LSB
b7	b6	b5	b4	b3	b2	b1	b0
b(7:4)	Reserved						
b(3:1)	Select curtain color: <ul style="list-style-type: none"> <li>• 0h: Black</li> <li>• 1h: Red</li> <li>• 2h: Green</li> <li>• 3h: Blue</li> <li>• 4h: Cyan</li> <li>• 5h: Magenta</li> <li>• 6h: Yellow</li> <li>• 7h: White</li> </ul>						
b(0)	Curtain enable: <ul style="list-style-type: none"> <li>• 0: Curtain disabled</li> <li>• 1: Curtain enabled</li> </ul>						

### 3.1.18 Write Image Freeze (1Ah)

This command enables or disables the image freeze function for the display module.

#### 3.1.18.1 Write Parameters

[Table 3-27](#) describes the write parameters.

**Table 3-27. Write Parameters**

MSB	Byte 1						LSB
b7	b6	b5	b4	b3	b2	b1	b0
b(7:1)	Reserved						
b(0)	Image freeze: <ul style="list-style-type: none"> <li>• 0: Image freeze disabled</li> <li>• 1: Image freeze enabled</li> </ul>						

The image freeze capability has two main uses. The first use is to simply freeze the current image on the screen. The second use is to assist the user in reducing display artifacts during system changes. In this second case, the image is frozen, system changes are made, and the image is unfrozen when complete. In all cases, when the image is unfrozen, the display shows the most resent input image. Input data between the freeze point and the unfreeze point is lost.

The controller software does not freeze or unfreeze the image except when explicitly commanded by the *Write Image Freeze* command. The controller software may execute the *Write Image Freeze* when the software is making updates to the system on its own volition, and for any operation commanded via the I<sup>2</sup>C interface.

The user must review the notes for the *Write Input Source Select* command to understand the concept of source-associated commands. This concept determines when source-associated commands are executed by the system.

If the user chooses not to make use of image freeze, they must change the source before changing the 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. Embedded software executes *Write Image Freeze* command to disable freeze.

Because commands to the controller process serially, no special timing or delay is required between these commands. The number of commands placed between the freeze and unfreeze must be small, as it is 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 3-28](#), which is not an all-inclusive list. The user is responsible for determining the correct use of the image freeze command.

**Table 3-28. Partial List of Commands that May Benefit from the Use of Image Freeze**

Command	Command OpCode	Notes
Write Input Source Select	05h	
Write External Video Source Format Select	07h	If changed while this source is the active source, embedded software disables image freeze after this command is executed
Write Test Pattern Select	0Bh	If changed while this source is the active source
Write Splash Screen Select	0Dh	If changed while this source is the active source, embedded software disables image freeze after this command is executed
Write Look Select	22h	

[Table 3-29](#) and [Table 3-30](#) show a few examples of how to use the image freeze command.

**Table 3-29. Splash Screen Example Using Image Freeze**

Command	Notes
Write Display Image Curtain = enable	May want to apply curtain if already displaying an unwanted image (such as a broken source)
Write Image Freeze = freeze	
Write Image Crop, Write Display Size, Write Display Image Orientation	Potential data processing commands that may be required for proper display of splash image. 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

The new splash image displays when the *Write Splash Screen Execute* command executes, regardless of the state of the *Write Image Freeze* command (due to the one time nature of the splash image). Embedded software disables image freeze after this command is executed.

**Table 3-30. 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	Potential data processing commands that may be required for proper display of test pattern image. These must be set before the Write Input Source Select command.
Write Input Source Select = test pattern generator	
Write Image Freeze = unfreeze	

### 3.1.19 Read Image Freeze (1Bh)

This command reads the state of the image freeze function for the display module.

#### 3.1.19.1 Read Parameters

This command has no read parameters.

### 3.1.19.2 Return Parameters

Table 3-31 describes the return parameters.

**Table 3-31. Return Parameters**

MSB	Byte 1						LSB
b7	b6	b5	b4	b3	b2	b1	b0
b(7:1)	Reserved						
b(0)	Image freeze:						
	<ul style="list-style-type: none"> <li>• 0: Image freeze disabled</li> <li>• 1: Image freeze enabled</li> </ul>						

### 3.1.20 Write 3-D Control (20h)

This command controls 3-D functionality for the display module.

#### 3.1.20.1 Write Parameters

Table 3-32 describes the write parameters.

**Table 3-32. Write Parameters**

msb	Byte 1						lsb
b7	b6	b5	b4	b3	b2	b1	b0
b(7)	<i>Reserved</i>						
b(6)	Polarity of 3-D Reference (External Only)						
	<ul style="list-style-type: none"> <li>• 0h: Correct – No Inversion Required</li> <li>• 1h: Incorrect – Inversion Required</li> </ul>						
b(5)	Frame Dominance						
	<ul style="list-style-type: none"> <li>• 0h: Left Dom. (Data sent left eye first)</li> <li>• 1h: Right Dom. (Data sent right eye first)</li> </ul>						
b(4:2)	<i>Reserved</i>						
b(1)	Source of 3-D Reference						
	<ul style="list-style-type: none"> <li>• 0h: Internal Reference Generator NOT supported</li> <li>• 1h: External (SLT_3DR Pin)</li> </ul>						
b(0)	<i>Reserved</i>						
Default: 00h							

When appropriate, the system will automatically enable 3-D operation based on source frame rate and whether 3-D sequences are loaded in flash. The 3-D parameters specified by this command will take effect following the next VSYNC.

3-D image data must always be sent frame sequential. Syncs and blanking are sent between every eye frame at frame rates greater than approximately 94Hz. DLPC3439 does not support frame rate multiplication.

Internal reference generator is not supported on dual ASIC DLPC3439.

The 3-D Reference is used to specify whether a frame of data contains left eye data or right eye data. This 3-D reference can be provided to the display by an external hardware signal. Table 3-33 shows which 3-D Reference source can be used with which image data port.

When using the external hardware signal as the reference, it must be provided for every frame of data. If the external 3-D Reference is misaligned with the data, it can be corrected using the *Polarity of 3-D Reference (External Only)* parameter. As noted, the *Polarity of 3-D Reference* parameter is only applicable when the External Signal is selected as the 3-D Reference source.

**Table 3-33. 3-D Reference Source Applicability for Display Data Ports**

Display Data Port	3-D Reference Source	Applicable	Notes
Parallel	External Hardware Signal	Yes	Recommended
Parallel	Internal Reference Generator	No	

The [Write 3-D Reference \(30h\)](#) command should be use with this selection.

For frame sequential 3-D, Frame Dominance determines which eye frames in the data stream go together to create a single 3-D image. Left Dominance indicates that the first eye frame of a pair is Left and the second eye frame is Right. Right Dominance indicates that the first eye frame of a pair is Right and the second eye frame is Left. Frame dominance is essential for correct operation of display histograms, which span both eye frames of a single image. When the image is frozen, proper Frame Dominance ensures the correct display of two eye frames together. Frame dominance control must not be used to attempt correction for misalignment of the 3-D reference signal to the image data.

### 3.1.21 Read 3-D Control (21h)

This command reads the state of the 3-D control function for the display module.

#### 3.1.21.1 Read Parameters

This command has no read parameters.

#### 3.1.21.2 Return Parameters

[Table 3-34](#) describes the return parameters.

**Table 3-34. Return Parameters**

msb	Byte 1						lsb
	b7	b6	b5	b4	b3	b2	
b(7)	<i>Reserved</i>						
b(6)	Polarity of 3-D Reference (External Only) <ul style="list-style-type: none"> <li>0h: Correct – No Inversion Required</li> <li>1h: Incorrect – Inversion Required</li> </ul>						
b(5)	Frame Dominance <ul style="list-style-type: none"> <li>0h: Left Dom. (Data sent left eye first)</li> <li>1h: Right Dom. (Data sent right eye first)</li> </ul>						
b(4:2)	<i>Reserved</i>						
b(1)	Source of 3-D Reference <ul style="list-style-type: none"> <li>0h: Internal Reference Generator NOT supported</li> <li>1h: External (SLT_3DR Pin)</li> </ul>						
b(0)	3-D Mode Control <ul style="list-style-type: none"> <li>0h: 2-D Operation</li> <li>1h: 3-D Operation</li> </ul>						

The system automatically enables and disables 3-D operation. Bit(0) will indicate the state of 2-D/3-D operation.

### 3.1.22 Write Look Select (22h)

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

#### 3.1.22.1 Write Parameters

[Table 3-35](#) describes the write parameters.

**Table 3-35. Write Parameters**

MSB	Byte 1						LSB
b7	b6	b5	b4	b3	b2	b1	b0
b(7:0)		Look number					

In this product, a Look typically specifies a target white point. The number of Looks available may be limited by the available space in flash memory.

This command allows the host to select a Look (target white point) from a number of Looks stored in flash.

Looks are specified in this byte by an enumerated value (such as 0,1,2,3). There must always be at least one Look, with an enumerated value of 0.

There are two other items that the host must specify when determining a white point. These are:

- A desired degamma curve, achieved by selecting the appropriate degamma/CMT, which has the desired degamma curve and correct bit weights for the sequence selected.
- The desired color points, achieved by selecting the appropriate CCA parameters using the CCA select command.

### 3.1.23 Read Look Select (23h)

This command reads the state of the Look select command for the display module.

#### 3.1.23.1 Read Parameters

This command has no read parameters.

#### 3.1.23.2 Return Parameters

[Table 3-36](#) describes the return parameters.

**Table 3-36. Return Parameters**

Parameter Bytes	Description
Byte 1	Look Number. See the following notes.
Byte 2	Sequence number. See the following notes.
Byte 3	Current Sequence Frame Rate (LSB). See the following notes.
Byte 4	Current Sequence Frame Rate.
Byte 5	Current Sequence Frame Rate.
Byte 6	Current Sequence Frame Rate (MSB).

MSB	Byte 1 and 2						LSB
b7	b6	b5	b4	b3	b2	b1	b0

**Table 3-37. Byte 1 Read Look Select Register Field Descriptions**

Bit	Type	Description
7-0	R	Look number

**Table 3-38. Byte 2 Read Look Select Register Field Descriptions**

Bit	Type	Description
7-0	R	Sequence number

Looks are specified by an enumerated value (such as 0, 1, 2, 3).

Sequences are specified by an enumerated value (such as 0, 1, 2, 3). The value returned by this command is the sequence currently selected by the Look algorithm when the command is received.

The current sequence frame rate is returned as a count specified in units of 66.67 ns. This is based on the internal 15-MHz clock used to time between input frame syncs. The frame rate is specified in this manner to



enable the software to make quick and simple comparisons to the frame count. Either the controller software or the user can make sequence frame rate and duty cycle selections.

### 3.1.24 Read Sequence Header Attributes (26h)

This command reads sequence header information for the active sequence of the display module.

#### 3.1.24.1 Read Parameters

This command has no read parameters.

#### 3.1.24.2 Return Parameters

Table 3-39 describes the return parameters.

**Table 3-39. Return Parameters**

Parameter Bytes	Description
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	Maximum number of sequence vectors (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	Maximum number of sequence vectors (Sequence structure)

The sequence header data is stored in two separate flash data structures (the Look structure and the sequence structure), and the values from each must match.

The bit weight and bit order for the duty cycle data is shown in Table 3-40.

**Table 3-40. Bit Weight and Bit Order for Duty Cycle Data**

MSB	Byte 2						LSB	MSB	Byte 1						LSB
b15 2 <sup>7</sup>	b14 2 <sup>6</sup>	b13 2 <sup>5</sup>	b12 2 <sup>4</sup>	b11 2 <sup>3</sup>	b10 2 <sup>2</sup>	b9 2 <sup>1</sup>	b8 2 <sup>0</sup>	b7 2 <sup>-1</sup>	b6 2 <sup>-2</sup>	b5 2 <sup>-3</sup>	b4 2 <sup>-4</sup>	b3 2 <sup>-5</sup>	b2 2 <sup>-6</sup>	b1 2 <sup>-7</sup>	b0 2 <sup>-8</sup>

The duty cycle data is specified as each color percent of the frame time. The sum of the three duty cycles must add up to 100. An example possibility is, R = 30.5 = 1E80h, G = 50 = 3200h, and B = 19.5 = 1380h)

The sequence maximum and minimum frame counts are specified in units of 66.67 ns. This is based on the internal 15-MHz clock used to time between input frame syncs. The frame rate is specified in this manner to enable the software to make quick and simple comparisons to the frame count.

The maximum number of sequence vectors byte is defined in [Table 3-41](#).

**Table 3-41. Maximum Number of Sequence Vectors**

MSB	Byte 15 and 30						LSB
b7	b6	b5	b4	b3	b2	b1	b0
b(7:4)	Reserved						
b(3:0)	Maximum number of sequence vectors						

### 3.1.25 Write Degamma/CMT Select (27h)

This command is used to select a specific Degamma/CMT LUT for the display module.

#### 3.1.25.1 Write Parameters

[Table 3-42](#) describes the write parameters.

**Table 3-42. Write Parameters**

MSB	Byte 1						LSB
b7	b6	b5	b4	b3	b2	b1	b0
b(7:0)	Degamma/CMT LUT index number (0-255)						

### 3.1.26 Read Degamma/CMT Select (28h)

This command is used to read the Degamma/CMT LUT index for the display module.

#### 3.1.26.1 Read Parameters

This command has no read parameters.

#### 3.1.26.2 Return Parameters

[Table 3-43](#) describes the return parameters.

**Table 3-43. Return Parameters**

MSB	Byte 1						LSB
b7	b6	b5	b4	b3	b2	b1	b0
b(7:0)	Degamma/CMT LUT index number						

### 3.1.27 Write CCA Select (29h)

This command specifies which set of CCA (color coordinate adjustment) parameters to use.

#### 3.1.27.1 Write Parameters

[Table 3-44](#) describes the write parameters.

**Table 3-44. Write Parameters**

MSB	Byte 1						LSB
b7	b6	b5	b4	b3	b2	b1	b0
b(7:0)	CCA Parameter Set Index						

One or more CCA parameter sets may be included in a firmware image. Each CCA parameter set is intended to specify a target color point in the system. This command enables selection from multiple CCA sets by an enumerated value (such as 0,1,2,3). A CCA parameter set must exist for it to be selected. This command may be used in conjunction with *Write Color Coordinate Adjustment Control*.

### 3.1.28 Read CCA Select (2Ah)

This command reads the state of the CCA Parameter Set Index.

#### 3.1.28.1 Read Parameters

This command has no read parameters.

#### 3.1.28.2 Return Parameters

[Table 3-45](#) describes the return parameters.

**Table 3-45. Return Parameters**

MSB	Byte 1						LSB
b7	b6	b5	b4	b3	b2	b1	b0
b(7:0)	CCA Parameter Set Index						

### 3.1.29 Read DMD Sequencer Sync Mode (2Ch)

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

#### 3.1.29.1 Read Parameters

This command has no read parameters.

#### 3.1.29.2 Return Parameters

[Table 3-46](#) describes the return parameters.

**Table 3-46. Return Parameters**

MSB	Byte 1						LSB
b7	b6	b5	b4	b3	b2	b1	b0
b(7:2)	Reserved						
b(1)	System auto-sync setting: <ul style="list-style-type: none"> <li>0h: Lock to external VSYNC (auto-sync)</li> <li>1h: Lock to internal VSYNC (auto-sync)</li> </ul>						
b(0)	DMD sequencer sync mode: <ul style="list-style-type: none"> <li>0h: Auto-sync</li> <li>1h: Force lock to internal VSYNC</li> </ul>						

The DMD sequencer sync mode response indicates the setting specified by the *Write DMD Mode* command.

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 the external VSYNC (auto-sync) option indicates that the system is using the externally provided VSYNC to drive the display module. The lock to the internal VSYNC option indicates that the system is using the internal VSYNC generator to drive the display module.

### 3.1.30 Write Execute Flash Batch File (2Dh)

This command executes a flash batch file for the display module.

#### 3.1.30.1 Write Parameters

[Table 3-47](#) describes the write parameters.

**Table 3-47. Write Parameters**

Parameter Bytes	Description
Byte 1	Batch file number

This command executes a batch file stored in the flash of the display module. Any system write command that can be sent by itself can be grouped together with other system commands or write parameters into a flash batch file, with the exception of all read commands.

The flash batch file numbers specified in this byte are enumerated values (such as 0,1,2,3). Flash batch file 0 is a special auto-initialization batch file that runs automatically by the DLPC34xx software immediately after system initialization is complete. The flash batch file 0 is typically not called using the *Write Execute Batch File* command (although the system allows it). This special flash batch file specifies the source to be used (such as splash screen or data port) once the system initializes.

Embedding flash batch file calls within a flash batch file is not allowed (for example, calling another batch file from within a batch file is not allowed). To execute two batch files back to back, use back to back execute batch file commands.

The system allows adding an execution delay between commands within a flash batch file, which is done using the [Write Flash Batch File Delay \(DBh\)](#) command.

The order of command execution for commands within a flash batch file is the same as if the commands are received over the I<sup>2</sup>C port.

### 3.1.31 Write Input Image Size (2Eh)

This command specifies the active data size of the external input image to the display module.

#### 3.1.31.1 Write Parameters

[Table 3-48](#) describes the write parameters.

**Table 3-48. 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)

This command is used in conjunction with the [Write Input Source Select \(05h\)](#) command. When external video is selected as the image source, this command specifies the active data size of the input image to the system for all external video interfaces. The settings for this command are retained until changed with subsequent Write Input Image Size commands. The settings for this command are retained until changed using this command. These settings automatically apply each time the external video port is selected.

The parameter values are 1-based (for example, a value of 1280 pixels specifies 1280 pixels per line).

The user must review the notes for the [Write Input Source Select \(05h\)](#) command to understand the concept of source-associated commands. This concept determines when source-associated commands are executed by the system. This command is a source-associated command.

The maximum and minimum input values are shown in [Table 3-49](#). The controller flags values outside of these ranges as an error (invalid write parameter), and does not execute the command.

**Table 3-49. Input Image Size**

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

### 3.1.32 Read Input Image Size (2Fh)

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

#### 3.1.32.1 Read Parameters

This command has no read parameters.

#### 3.1.32.2 Return Parameters

[Table 3-50](#) describes the return parameters.

**Table 3-50. 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)

The parameter values are 1-based (for example, a value of 1280 pixels specifies 1280 pixels per line).

This command returns the value specified by the *Write External Input Image Size* command.

### 3.1.33 Write 3-D Reference (30h)

This command selects a 3-D reference for the display module.

#### 3.1.33.1 Write Parameters

[Table 3-51](#) describes the write parameters.

**Table 3-51. Write Parameters**

<i>msb</i>	<i>Byte 1</i>						<i>lsb</i>
b7	b6	b5	b4	b3	b2	b1	b0
b(7:1)	<i>Reserved</i>						
b(0)	3-D Reference <ul style="list-style-type: none"> <li>• 0h: Next Frame Left</li> <li>• 1h: Next Frame Right</li> </ul>						
Default: 00h							

The 3-D Reference is used to specify whether a frame of data contains left eye data or right eye data. The 3-D reference is provided to the display as either a GPIO hardware signal or through sending this command. The selection is made using the Write 3-D Control command. When using this command as the reference, it is recommended that the command be sent every frame, or at the minimum, at the start of each eye pair. For example, the command can be sent before each left eye frame. At a minimum, it must be sent once at the start of 3-D operation. If the 3-D Reference is misaligned with the data, it can be corrected using this command or by using the *Polarity of 3-D Reference* parameter in the [Write 3-D Control \(20h\)](#) command.

When the Write 3-D Reference command is received, its parameter value will be applied to the image data following the next VSYNC or Start of Frame command.

When this command is received, software must set up the internal 3-D reference generator. If the command is sent every frame, software will ensure that the output of the internal 3-D reference generator remains accurate.

### 3.1.34 Write Splash Screen Execute (35h)

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

#### 3.1.34.1 Write Parameters

This command has no write parameters.

This command is used in conjunction with the *Write Input Source Select* and the *Write Splash Screen Select* commands to start the process of retrieving a splash screen from flash for display.

The splash screen is read from flash and sent down the processing path of the controller to be stored in memory for display at the end of the processing path. All image processing settings, such as image crop, image orientation, display size, splash screen select, and splash screen as input source, must be set by the user 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 result is seen on the display. Thus, the splash screen retrieval process repeats each time this command is received. See the *Write Image Freeze* command for more information on hiding on-screen artifacts when selecting and retrieving a splash image.

The user must review the notes for the *Write Input Source Select* command to understand the concept of source-associated commands. This concept determines when source-associated commands are executed by the system. This command is a source-associated command; however, this command has no maintained state or history and has no settings to be stored or reused by the system.

When this command is processed the system automatically initializes the system color processing based on the splash header information prior to sending the splash image down the data path.

#### Note

If an incorrect image size is set for the splash image prior to splash screen execute, then the system requires power cycle to recover

### 3.1.35 Write Parallel Data Mask Control (37h)

This command is used to control the masking function for the external parallel port I/F of the DLPC343x.

#### 3.1.35.1 Write Parameters

Table 3-52 describes the write parameters.

**Table 3-52. Write Parameters**

MSB	Byte 1						LSB
b7	b6	b5	b4	b3	b2	b1	b0
b(7:2)	Reserved						
b(1)	Polarity Select for Data Mask Control: <ul style="list-style-type: none"> <li>0: Unmasked = 0, Masked = 1</li> <li>1: Unmasked = 1, Masked = 0</li> </ul>						
b(0)	Data Mask Enable: <ul style="list-style-type: none"> <li>0: Mask Disable</li> <li>1: Mask Enable</li> </ul>						

When the parallel data mask is enabled, the DLPC343x input PDM\_CVS\_TE pin functions as a data mask control for the video data on the parallel port interface. Therefore, when this functionality is enabled and the mask control is active, input image frames will be ignored and the source image will not be propagated to the display. During image frames that are masked, the last unmasked image frame received will continue to be displayed. The mask control signal (PDM\_CVS\_TE) should only be updated during vertical blanking.

The Polarity Select specifies the active state for the mask control signal. The polarity should only be updated when the mask function is disabled (via this command).

### 3.1.36 Read Parallel Data Mask Control (38h)

This command reads the state of the parallel data mask control.

#### 3.1.36.1 Read Parameters

This command has no read parameters.

#### 3.1.36.2 Return Parameters

[Table 3-53](#) describes the return parameters.

**Table 3-53. Return Parameters**

MSB	Byte 1						LSB
b7	b6	b5	b4	b3	b2	b1	b0
b(7:2)	Reserved						
b(1)	Polarity Select for Data Mask Control: <ul style="list-style-type: none"> <li>0: Unmasked = 0, Masked = 1</li> <li>1: Unmasked = 1, Masked = 0</li> </ul>						
b(0)	Data Mask Enable: <ul style="list-style-type: none"> <li>0: Mask Disabled</li> <li>1: Mask Enabled</li> </ul>						

### 3.1.37 Write Mirrors Lock Command (39h)

This command writes the Mirrors Lock command to lock/unlock the DMD interface for Optical Alignment.

#### 3.1.37.1 Write Parameters

[Table 3-54](#) describes the write parameters.

**Table 3-54. Write Parameters**

MSB	Byte 1						LSB
b7	b6	b5	b4	b3	b2	b1	b0
7-2	Reserved						
1-0	Mirror Lock State Selection 0h = Reserved 1h = DMD Interface Lock 2h = DMD Interface Unlock 3h = DMD Interface Unlock, Delay 100ms, DMD Interface Lock						

This command is only used in factory to prevent DMD damage.

### 3.1.38 Read Mirrors Lock Command (3Ah)

This command reads the status of the Mirrors Lock.

#### 3.1.38.1 Read Parameters

This command has no read parameters.

#### 3.1.38.2 Return Parameters

[Table 3-55](#) describes the return parameters.

**Table 3-55. Return Parameters**

MSB	Byte 1						LSB
b7	b6	b5	b4	b3	b2	b1	b0
7-2	Reserved						

1-0	Mirror Lock State Selection 0h = Reserved 1h = DMD Interface Lock 2h = DMD Interface Unlock 3h = DMD Interface Unlock, Delay 100ms, DMD Interface Lock
-----	--



## 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

Table 3-56 describes the write parameters.

**Table 3-56. Write Parameters**

MSB	Byte 1						LSB
b7	b6	b5	b4	b3	b2	b1	b0
b(7:2)	Reserved						
b(1:0)	LED control method: <ul style="list-style-type: none"> <li>• 00: Manual RGB LED currents (disables CAIC algorithm)</li> <li>• 01: CAIC (automatic) RGB LED power (enables CAIC algorithm)</li> <li>• 10: Reserved</li> <li>• 11: Reserved</li> </ul>						

This command selects the method used to control the output of the red, green, and blue LEDs. Based on the method chosen, a specific set of commands are available for controlling the LED outputs. These are shown in Table 3-57.

The manual RGB LED currents method provides for manual control of the LED currents, and disables the CAIC algorithm. The CAIC (automatic) RGB LED current control method provides automatic control of the LED currents using the CAIC algorithm.

**Table 3-57. Available Commands Based on LED Control Method**

LED Control Method	Available Commands
Manual RGB LED current control	Write RGB LED Enable Read RGB LED Enable Write Manual RGB LED Current Read Manual RGB LED Current Write Manual RGB LED Max Current (see <a href="#">Write RGB LED Max Current (5Ch)</a> ) Read Manual RGB LED Max Current (see <a href="#">Read RGB LED Max Current (5Dh)</a> )
CAIC (automatic) RGB LED current control	Write RGB LED Enable Read RGB LED Enable Read CAIC LED Max Available Power Read CAIC LED RGB Current

### 3.2.2 Read LED Output Control Method (51h)

This command reads the selected LED output control method for the display module.

#### 3.2.2.1 Read Parameters

This command has no read parameters.

#### 3.2.2.2 Return Parameters

Table 3-58 describes the return parameters.

**Table 3-58. Return Parameters**

MSB	Byte 1						LSB
b7	b6	b5	b4	b3	b2	b1	b0

b(7:2)	Reserved
b(1:0)	LED control method: <ul style="list-style-type: none"> <li>• 00: Manual RGB LED currents (CAIC algorithm disabled)</li> <li>• 01: CAIC (automatic) RGB LED current control (CAIC algorithm enabled)</li> <li>• 10: Reserved</li> <li>• 11: Reserved</li> </ul>

### 3.2.3 Write RGB LED Enable (52h)

This command enables the LEDs for the display module.

#### 3.2.3.1 Write Parameters

Table 3-59 describes the write parameters.

**Table 3-59. Write Parameters**

MSB	Byte 1						LSB
b7	b6	b5	b4	b3	b2	b1	b0
b(7:3)	Reserved						
b(2)	Blue LED enable: <ul style="list-style-type: none"> <li>• 0: Blue LED disabled</li> <li>• 1: Blue LED enabled</li> </ul>						
b(1)	Green LED enable: <ul style="list-style-type: none"> <li>• 0: Green LED disabled</li> <li>• 1: Green LED enabled</li> </ul>						
b(0)	Red LED enable: <ul style="list-style-type: none"> <li>• 0: Red LED disabled</li> <li>• 1: Red LED enabled</li> </ul>						

### 3.2.4 Read RGB LED Enable (53h)

This command reads the state of the LED enables for the display module.

#### 3.2.4.1 Read Parameters

This command has no read parameters.

#### 3.2.4.2 Return Parameters

Table 3-60 describes the return parameters.

**Table 3-60. Return Parameters**

MSB	Byte 1						LSB
b7	b6	b5	b4	b3	b2	b1	b0
b(7:3)	Reserved						
b(2)	Blue LED enable: <ul style="list-style-type: none"> <li>• 0: Blue LED disabled</li> <li>• 1: Blue LED enabled</li> </ul>						
b(1)	Green LED enable: <ul style="list-style-type: none"> <li>• 0: Green LED disabled</li> <li>• 1: Green LED enabled</li> </ul>						

b(0)	Red LED enable: <ul style="list-style-type: none"> <li>• 0: Red LED disabled</li> <li>• 1: Red LED enabled</li> </ul>
------	---

### 3.2.5 Write RGB LED Current (54h)

This command sets the current for the red, green, and blue LEDs of the display module.

#### 3.2.5.1 Write Parameters

Table 3-61 describes the write parameters.

**Table 3-61. Write Parameters**

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

When an all-white image is displayed, this command allows the system white point to be adjusted while establishing the total LED power whether the CAIC algorithm is enabled or disabled.

The parameters specified by this command have a resolution of 10 bits, and are defined by the appropriate DLPA200x specification.

When the CAIC algorithm is disabled, this command directly sets the LED currents (the R, G, and B values provided are sent directly to the DLPA200x device) regardless of the image being displayed.

When the CAIC algorithm is enabled:

- This command directly sets the LED currents 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 currents from those specified by this command, and the total LED power may drop. The *Read CAIC RGB LED Current* command can read the actual LED currents for the image currently displayed.
- In the case of an all-white image, the values read by the *Read CAIC RGB LED Current* command closely match, but may not exactly match, those requested using the *Write RGB LED Current* command. For an all-white image, the *Read CAIC RGB LED Current* command gives currents within +/-4 DLPA200x device current steps for each LED color relative to those requested by the *Write RGB LED Current* command.
- When the *Write RGB LED Current* command changes the LED currents, the LED current for any color must not be changed by more than +/-25% from the nominal current used for that color when the CAIC LUTs were created. No LED current can be set to a current 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, since the CAIC algorithm requests the CAIC LED maximum available power. The maximum available LED power for the CAIC is controlled by the *Write RGB LED Current* command, as this command controls currents for an all-white image. After the currents are adjusted, the *Read CAIC LED Max Available Power* command is used to see the maximum power in watts derived from the CAIC.

### 3.2.6 Read RGB LED Current (55h)

This command reads the state of the current for the red, green, and blue LEDs of the display module.

#### 3.2.6.1 Read Parameters

This command has no read parameters.

#### 3.2.6.2 Return Parameters

Table 3-62 describes the return parameters.

**Table 3-62. Return Parameters**

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

See [Write RGB LED Current \(54h\)](#) 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.

#### 3.2.7.1 Read Parameters

This command has no read parameters.

#### 3.2.7.2 Return Parameters

[Table 3-63](#) describes the return parameters.

**Table 3-63. Return Parameters**

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

The value is specified in watts × 100 (for example: 25.75 W = A0Fh). This command is only applicable when CAIC is enabled.

The CAIC maximum available LED power pertains if an all-white image is displayed where LED currents are set by the *Write RGB LED Current* command. The calculation is:

$$R \text{ duty cycle} \times R \text{ LED current} \times R \text{ LED voltage} + G \text{ duty cycle} \times G \text{ LED current} \times G \text{ LED voltage} + B \text{ duty cycle} \times B \text{ LED current} \times B \text{ LED voltage}$$

For example:  $(.30 \times .49 \text{ A} \times 2.0 \text{ V}) + (.50 \times .39 \text{ A} \times 3.1 \text{ V}) + (.20 \times .39 \text{ A} \times 3.1 \text{ V}) = (.30 \times .980 \text{ W}) + (.50 \times 1.209 \text{ W}) + (.20 \times 1.209 \text{ W}) = 1.140 \text{ W}$

### 3.2.8 Write RGB LED Max Current (5Ch)

This command specifies the maximum LED current allowed for each LED in the display module.

#### 3.2.8.1 Write Parameters

[Table 3-64](#) describes the write parameters.

**Table 3-64. Write Parameters**

Parameter Bytes	Description
Byte 1	Maximum red LED current (LSByte)
Byte 2	Maximum red LED current (MSByte)
Byte 3	Maximum green LED current (LSByte)
Byte 4	Maximum green LED current (MSByte)
Byte 5	Maximum blue LED current (LSByte)
Byte 6	Maximum blue LED current (MSByte)

This command sets the maximum LED currents that can be used when CAIC is enabled or disabled. When CAIC is enabled, the maximum LED currents may be further limited by the CAIC LUTs stored in the flash. The parameters specified by this commands have a resolution of 10 bits. The unused most significant bits should be set to 0.

### 3.2.9 Read RGB LED Max Current (5Dh)

This command reads the specified maximum LED current allowed for each LED in the display module.

#### 3.2.9.1 Read Parameters

This command has no read parameters.

#### 3.2.9.2 Return Parameters

[Table 3-65](#) describes the return parameters.

**Table 3-65. Return Parameters**

Parameter Bytes	Description
Byte 1	Maximum red LED current (LSByte)
Byte 2	Maximum red LED current (MSByte)
Byte 3	Maximum green LED current (LSByte)
Byte 4	Maximum green LED current (MSByte)
Byte 5	Maximum blue LED current (LSByte)
Byte 6	Maximum blue LED current (MSByte)

See the [Write RGB LED Current \(54h\)](#) for a detailed description of the return parameters.

Unused most significant bits are set to 0.

### 3.2.10 Read CAIC RGB LED Current (5Fh)

This command reads the state of the current for the red, green, and blue LEDs of the display module.

#### 3.2.10.1 Read Parameters

This command has no read parameters.

#### 3.2.10.2 Return Parameters

[Table 3-66](#) describes the return parameters.

**Table 3-66. Return Parameters**

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

The parameters returned by this command have a resolution of 10 bits, and are defined by the appropriate DLPA200x specification.

When the CAIC algorithm is enabled using the *LED Output Control Method* command:

- The *Write RGB LED Current* command directly sets the LED currents when an all-white image is displayed. If the image changes from an all-white image, depending on the image, the CAIC algorithm may alter one or more of the LED currents from those specified by the *Write RGB LED Current* command, and the total LED power may drop. The actual LED currents for the image currently displayed are read using the *Read CAIC RGB LED Current* 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 Current* command. For an all-white image, this command provides values within +/- 4 DLPA200x device current steps for each LED color relative to those specified with the *Write RGB LED Current* command.

Use of this command is only appropriate when the LED output control method is set to CAIC (automatic) RGB LED current control.

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 LABB image processing functionality for the display module.

##### 3.3.1.1 Write Parameters

Table 3-67 describes the write parameters.

**Table 3-67. Write Parameters**

Parameter Bytes	Description
Byte 1	See Table 3-68
Byte 2	LABB strength setting

**Table 3-68. Byte 1 Return Parameters**

MSB	Byte 1						LSB
b7	b6	b5	b4	b3	b2	b1	b0
b(7:4)	Sharpness strength						
b(3:2)	Reserved						
b(1:0)	LABB control: <ul style="list-style-type: none"> <li>• 0h: Disabled</li> <li>• 1h: Enabled: Manual strength control</li> <li>• 2h: Reserved</li> <li>• 3h: Reserved</li> </ul>						

The key function of the LABB is to adaptively gain up darker parts of the image to achieve an overall brighter image.

For LABB strength, 0 indicates no boost applied and 255 indicates the maximum boost viable in a product. The strength is not a direct indication of the gain, since the gain varies depending on the image content.

Sharpness strength ranges from 0 to 15, with 0 indicating sharpness disabled and 15 indicating the maximum sharpness. The LABB function must be enabled to make use of sharpness.

LABB is supported in TPG, splash, and external input mode, but auto-disabled in curtain mode.

#### 3.3.2 Read Local Area Brightness Boost Control (81h)

This command reads the state of the LABB image processing functionality for the display module.

##### 3.3.2.1 Read Parameters

This command has no read parameters.

##### 3.3.2.2 Return Parameters

Table 3-69 describes the return parameters.

**Table 3-69. Return Parameters**

Parameter Bytes	Description
Byte 1	See Table 3-70
Byte 2	LABB strength setting
Byte 3	LABB gain value

**Table 3-70. Byte 1 Return Parameters**

MSB	Byte 1						LSB
b7	b6	b5	b4	b3	b2	b1	b0
b(7:4)	Sharpness strength						

b(3:2)	Reserved
b(1:0)	LABB control: <ul style="list-style-type: none"> <li>• 0h: Disabled</li> <li>• 1h: Enabled: Manual strength control</li> <li>• 2h: Reserved</li> <li>• 3h: Reserved</li> </ul>

[Table 3-71](#) shows the bit order and weighting for the LABB gain value, which ranges from 1 to 8 (the controller software limits the lower value to 1).

**Table 3-71. Bit Weight Definition for LABB Gain Value**

b7	b6	b5	b4	b3	b2	b1	b0
2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>	2 <sup>-1</sup>	2 <sup>-2</sup>	2 <sup>-3</sup>	2 <sup>-4</sup>	2 <sup>-5</sup>

The software equation to calculate LABB Gain as a fixed point value is shown below:

$$\text{LABB\_gain} = \text{add\_8lsb(APL)} / \text{pre\_LABB\_APL} \quad (//\text{add 8 LSBS (u8.0 / u8.0 = u8.8 / u8.0 = u8.8)})$$

### 3.3.3 Write CAIC Image Processing Control (84h)

This command controls the CAIC functionality for the display module.

#### 3.3.3.1 Write Parameters

[Table 3-72](#) describes the write parameters.

**Table 3-72. Write Parameters**

Parameter Bytes	Description
Byte 1	See <a href="#">Table 3-73</a>
Byte 2	CAIC maximum lumens gain
Byte 3	CAIC clipping threshold

**Table 3-73. Byte 1 Write Parameters**

MSB	Byte 1						LSB
b7	b6	b5	b4	b3	b2	b1	b0
b(7)	CAIC gain display enable: <ul style="list-style-type: none"> <li>• 0h: Disabled</li> <li>• 1h: Enabled</li> </ul>						
b(6)	CAIC gain display scale: <ul style="list-style-type: none"> <li>• 0h: 100% = 1024 pixels</li> <li>• 1h: 100% = 512 pixels</li> </ul>						
b(5:0)	Reserved						

The CAIC algorithm (Content Adaptive Illumination Control) provides adaptive control of the LED currents and the digital gain applied to the image.

The CAIC algorithm is enabled or disabled based on the method of LED current control selected by the user using the *Write LED Output Control Method* command. When enabled, the CAIC algorithm provides automatic control of the LED currents as specified by this command and the *Write LED Output Control Method* command.

The CAIC gain display provides a visual presentation of the instantaneous gain provided by the CAIC algorithm. The CAIC gain display is typically used as a debug tool and to show the performance of the algorithm. The CAIC gain display must never be used for normal operation. The display is composed of five bars, with the bottom three bars (green, red, and blue) showing the respective CAIC gain for each color. The top two bars are for TI debug use only.



Table 3-74 shows the bit order and weighting for the CAIC maximum lumens gain value, which has a valid range from 1.0 to 4.0. Values outside of this range are considered an error (invalid write parameter value – communication status) and the command does not execute.

**Table 3-74. Bit Weight Definition for the CAIC Maximum Gain Value**

b7	b6	b5	b4	b3	b2	b1	b0
2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>	2 <sup>-1</sup>	2 <sup>-2</sup>	2 <sup>-3</sup>	2 <sup>-4</sup>	2 <sup>-5</sup>

The CAIC maximum lumens gain parameter sets the maximum lumens gain for a pixel as a result of both digital gain and increasing LED currents. The CAIC maximum lumens gain parameter 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 is reduced.

Table 3-75 shows the bit order and weighting for the CAIC clipping threshold value, which has a valid range from 0.0% to 4.0%. Values outside of this range are considered an error (invalid write parameter value – communication status) and the command does not execute.

**Table 3-75. Bit Weight Definition for the CAIC Clipping Threshold Value**

b7	b6	b5	b4	b3	b2	b1	b0
2 <sup>1</sup>	2 <sup>0</sup>	2 <sup>-1</sup>	2 <sup>-2</sup>	2 <sup>-3</sup>	2 <sup>-4</sup>	2 <sup>-5</sup>	2 <sup>-6</sup>

The CAIC clipping threshold parameter sets the percentage of pixels clipped by the CAIC algorithm over the full frame of active data, due to the digital gain applied by the CAIC algorithm.

Table 3-76 shows the bit order and weighting for the CAIC RGB intensity gain values, which have a valid range from 0.0 to almost 1.0. Values outside of this range are considered an error (invalid write parameter value – communication status) and the command does not execute.

**Table 3-76. Bit Weight Definition for the CAIC RGB Intensity Gain Values**

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Res	Res	Res	Res	Res	Res	2 <sup>-1</sup>	2 <sup>-2</sup>	2 <sup>-3</sup>	2 <sup>-4</sup>	2 <sup>-5</sup>	2 <sup>-6</sup>	2 <sup>-7</sup>	2 <sup>-8</sup>	2 <sup>-9</sup>	2 <sup>-10</sup>

CAIC can be enabled in TPG and external input mode, but auto-disabled in splash and curtain mode.

**Table 3-77. LABB and CAIC Modes**

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

### 3.3.4 Read CAIC Image Processing Control (85h)

This command reads the state of the CAIC functionality within the display module.

#### 3.3.4.1 Read Parameters

This command has no read parameters.

#### 3.3.4.2 Return Parameters

Table 3-78 describes the return parameters.

**Table 3-78. Return Parameters**

Parameter Bytes	Description
Byte 1	See <a href="#">Table 3-79</a>
Byte 2	CAIC maximum lumens gain
Byte 3	CAIC clipping threshold

**Table 3-79. Byte 1 Return Parameters**

MSB	Byte 1						LSB
b7	b6	b5	b4	b3	b2	b1	b0
b(7)	CAIC gain display enable: <ul style="list-style-type: none"> <li>0h: Disabled</li> <li>1h: Enabled</li> </ul>						
b(6)	CAIC gain display scale: <ul style="list-style-type: none"> <li>0h: 100% = 1024 pixels</li> <li>1h: 100% = 512 pixels</li> </ul>						
b(5:0)	Reserved						

Information on these parameters can be found in [Write CAIC Image Processing Control \(84h\)](#).

### 3.3.5 Write Color Coordinate Adjustment Control (86h)

This command controls the CCA image processing functionality for the display module.

#### 3.3.5.1 Write Parameters

[Table 3-80](#) describes the write parameters.

**Table 3-80. Write Parameters**

MSB	Byte 1						LSB
b7	b6	b5	b4	b3	b2	b1	b0
b(7:1)	Reserved						
b(0)	CCA enable: <ul style="list-style-type: none"> <li>0: Disabled</li> <li>1: Enabled</li> </ul>						

This function must remain enabled during normal operation.

When CCA is disabled, an identity matrix is used.

### 3.3.6 Read Color Coordinate Adjustment Control (87h)

This command reads the state of the CCA image processing within the display module.

#### 3.3.6.1 Read Parameters

This command has no read parameters.

#### 3.3.6.2 Return Parameters

[Table 3-81](#) describes the return parameters.

**Table 3-81. Return Parameters**

MSB	Byte 1						LSB
b7	b6	b5	b4	b3	b2	b1	b0
b(7:1)	Reserved						

b(0)

CCA enable:

- 0: Disabled
- 1: Enabled

### 3.4 General Setup Commands

#### 3.4.1 Write Border Color (B2h)

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

##### 3.4.1.1 Write Parameters

Table 3-82 describes the write parameters.

**Table 3-82. Write Parameters**

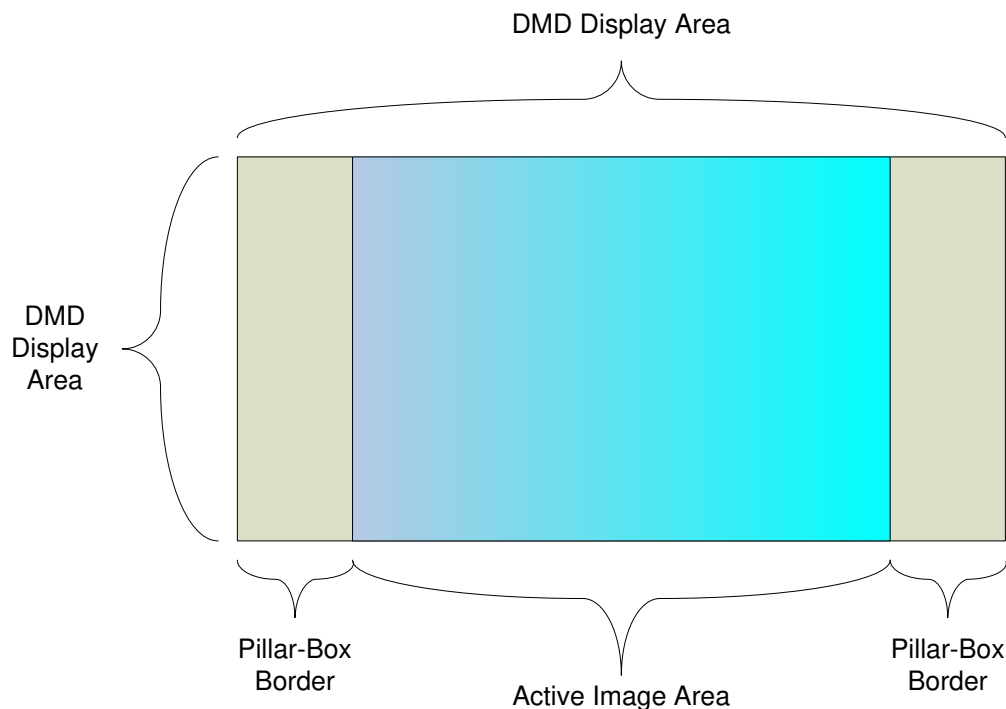
MSB	Byte 1						LSB
b7	b6	b5	b4	b3	b2	b1	b0
b(7:3)	Reserved						
b(2:0)	Display border color: <ul style="list-style-type: none"> <li>• 0h: Black</li> <li>• 1h: Red</li> <li>• 2h: Green</li> <li>• 3h: Blue</li> <li>• 4h: Cyan</li> <li>• 5h: Magenta</li> <li>• 6h: Yellow</li> <li>• 7h: White</li> </ul>						

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, or letterbox image.

To display a pillar box image (see Figure 3-12), the user can use the border color defined by this command.

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

Whenever the keystone capability is used, the user should set the border color to black. Setting this parameter to any other color when keystone is enabled produces undesirable results.



**Figure 3-12. Pillar-Box Border Example**

### 3.4.2 Read Border Color (B3h)

This command reads the state of the onscreen border color for the display module.

#### 3.4.2.1 Read Parameters

This command has no read parameters.

#### 3.4.2.2 Return Parameters

[Table 3-83](#) describes the return parameters.

**Table 3-83. Return Parameters**

MSB	Byte 1						LSB
	b7	b6	b5	b4	b3	b2	
b(7)	Pillar-box border color source: <ul style="list-style-type: none"> <li>• 0h: Defined by this command</li> <li>• 1h: Flash defined 24-bit color</li> </ul>						
b(6:3)	Reserved						
b(2:0)	Display border color: <ul style="list-style-type: none"> <li>• 0h: Black</li> <li>• 1h: Red</li> <li>• 2h: Green</li> <li>• 3h: Blue</li> <li>• 4h: Cyan</li> <li>• 5h: Magenta</li> <li>• 6h: Yellow</li> <li>• 7h: White</li> </ul>						

For the special case of a pillar box image (see [Figure 3-12](#)), the user can use the border color defined by the *Write Border Color* command.

### 3.4.3 Write Parallel Interface Sync Polarity (B6h)

This command specifies the VSYNC and HSYNC polarity for the DLPC343x parallel interface.

#### 3.4.3.1 Write Parameters

[Table 3-84](#) describes the write parameters.

**Table 3-84. Write Parameters**

MSB	Byte 1						LSB
	b7	b6	b5	b4	b3	b2	
b(7:3)	Reserved						
b(2)	Manual Mode – Parallel Interface HSYNC Polarity: <ul style="list-style-type: none"> <li>• 0: Falling Edge Active (Negative Pulse)</li> <li>• 1: Rising Edge Active (Positive Pulse)</li> </ul>						
b(1)	Manual Mode – Parallel Interface VSYNC Polarity: <ul style="list-style-type: none"> <li>• 0: Falling Edge Active (Negative Pulse)</li> <li>• 1: Rising Edge Active (Positive Pulse)</li> </ul>						
b(0)	Parallel Interface Sync Polarity Mode: <ul style="list-style-type: none"> <li>• 0: Automatic Mode</li> <li>• 1: Manual Mode</li> </ul>						

This command may be needed when using the parallel interface port. This command is not applicable for BT656 sources. In Automatic mode, the system can typically determine the appropriate polarity of the syncs. In Manual mode, the user is allowed to specify these polarities.

### 3.4.4 Read Parallel Interface Sync Polarity (B7h)

This command reads the state of the VSYNC and HSYNC polarity of the DLPC343x parallel interface.

#### 3.4.4.1 Read Parameters

This command has no read parameters.

#### 3.4.4.2 Return Parameters

Table 3-85 describes the return parameters.

**Table 3-85. Return Parameters**

MSB	Byte 1						LSB
b7	b6	b5	b4	b3	b2	b1	b0
b(7:2)	Reserved						
b(1)	Manual Mode – Parallel Interface HSYNC Polarity:						
	<ul style="list-style-type: none"> <li>• 0: Falling Edge Active (Negative Pulse)</li> <li>• 1: Rising Edge Active (Positive Pulse)</li> </ul>						
b(0)	Manual Mode – Parallel Interface VSYNC Polarity:						
	<ul style="list-style-type: none"> <li>• 0: Falling Edge Active (Negative Pulse)</li> <li>• 1: Rising Edge Active (Positive Pulse)</li> </ul>						

### 3.4.5 Read Auto Framing Information (BAh)

This command is used to read data from the flash for the display module.

#### 3.4.5.1 Read Parameters

This command has no read parameters.

#### 3.4.5.2 Return Parameters

Table 3-86 describes the return parameters.

**Table 3-86. Return Parameters**

Parameter Bytes	Description
Byte 1	External Input VSYNC Rate (LSByte)
Byte 2	External Input VSYNC Rate
Byte 3	External Input VSYNC Rate
Byte 4	External Input VSYNC Rate (MSByte)
Byte 5	External Input Total Pixels per Line (LSByte)
Byte 6	External Input Total Pixels per Line (MSByte)
Byte 7	External Input Total Lines per Frame (LSByte)
Byte 8	External Input Total Lines per Frame (MSByte)
Byte 9	External Input Active Pixels per Line (LSByte)
Byte 10	External Input Active Pixels per Line (MSByte)
Byte 11	External Input Active Lines per Frame (LSByte)
Byte 12	External Input Active Lines per Frame (MSByte)
Byte 13	Pixel/Line Reference Clock Rate (LSByte)
Byte 14	Pixel/Line Reference Clock Rate (MSByte)

- In most cases, the above data can be measured by the system (even when manual data framing is used).  
*This data is provided for debug purposes only.*
- The external input frame rate is returned as a count that is specified in units of 66.67ns (based on the internal 15MHz clock used to time between input frame syncs).
- The pixels per line and lines per frame parameters are to be '1' based. (that is, a value of 1280 active pixels indicates that there are 1280 active pixels per line).
- The pixels per line and lines per frame parameters are based on an internal sample clock (for DSI and CPU Bus) or a measurement of the actual input pixel clock (for the Parallel Bus). This clock rate is returned as the *Pixel/Line Reference Clock Rate*. This parameter value is the clock rate multiplied by 100 in MHz (for example, 60.00MHz = 1770h).

## 3.5 Administrative Commands

### 3.5.1 Read Short Status (D0h)

This command provides a short system status for the display module.

#### 3.5.1.1 Read Parameters

This command has no read parameters.

#### 3.5.1.2 Return Parameters

Table 3-87 describes the return parameters.

**Table 3-87. Return Parameters**

Parameter Bytes	Description
Byte 1	Short System Status

**Table 3-88. Byte 1 Return Parameters**

MSB	Byte 1 – General Status						LSB
b7	b6	b5	b4	b3	b2	b1	b0
b(7)	Boot/main application: <ul style="list-style-type: none"> <li>• 0: Boot</li> <li>• 1: Main</li> </ul>						
b(6)	Sensing Sequence error: <ul style="list-style-type: none"> <li>• 0: No error</li> <li>• 1: Error</li> </ul>						
b(5)	Flash error: <ul style="list-style-type: none"> <li>• 0: No error</li> <li>• 1: Error</li> </ul>						
b(4)	Flash erase Complete: <ul style="list-style-type: none"> <li>• 0: Not Complete</li> <li>• 1: Complete</li> </ul>						
b(3)	System error: <ul style="list-style-type: none"> <li>• 0: No error</li> <li>• 1: Error</li> </ul>						
b(2)	Reserved						
b(1)	Communication error: <ul style="list-style-type: none"> <li>• 0: No error</li> <li>• 1: Error</li> </ul>						
b(0)	System initialization: <ul style="list-style-type: none"> <li>• 0: Not complete</li> <li>• 1: Complete</li> </ul>						

The communication error bit indicates any error on the I<sup>2</sup>C command interfaces. Specific details about communication errors are available using the *Read Communication Status* command. Any errors other than a communication error are indicated by the system error bit. Specific details about system errors are available using the *Read System Status* command.

The communication error, and system error bits are cleared when the *Read Short Status* is read. The *Read Short Status* command must only be checked periodically, not continuously. Continuous access may severely impact system performance.



### 3.5.2 Read System Status (D1h)

This command reads system status information for the display module.

#### 3.5.2.1 Read Parameters

This command has no read parameters.

#### 3.5.2.2 Return Parameters

Table 3-89 describes the return parameters.

**Table 3-89. Return Parameters**

Parameter Bytes	Description
Byte 1	DMD interface status
Byte 2	LED status
Byte 3	Internal interrupt status
Byte 4	Misc. status

All system status error bits are cleared when the read system status is read.

**Table 3-90. Byte 1 Return Parameters**

MSB	Byte 1 – DMD Interface Status						LSB
b7	b6	b5	b4	b3	b2	b1	b0
b(7:3)	Reserved						
b(2)	DMD training error: <ul style="list-style-type: none"> <li>0h: No error</li> <li>1h: Error</li> </ul>						
b(1)	DMD interface error: <ul style="list-style-type: none"> <li>0h: No error</li> <li>1h: Error</li> </ul>						
b(0)	DMD device error: <ul style="list-style-type: none"> <li>0h: No error</li> <li>1h: Error</li> </ul>						

The system sets the DMD device error for the following conditions:

- The system cannot read the DMD device ID from the DMD.
- The system-specified DMD device ID does not match the actual DMD device ID.

The system sets the DMD interface error when there are power management setup conflicts on this interface.

The system sets the DMD training error when the training algorithm cannot find a data eye that meets the specified requirements. For controller pins selected for training, DMD training will fail if less than 20 out of 50 continuous response values pass.

**Table 3-91. Byte 2 Return Parameters**

MSB	Byte 2 – LED Status						LSB
b7	b6	b5	b4	b3	b2	b1	b0
b(7)	Reserved						
b(6)	LED no connection error: <ul style="list-style-type: none"> <li>0h: No error</li> <li>1h: Error</li> </ul>						
b(5:3)	Reserved						

b(2)	Blue LED state: <ul style="list-style-type: none"> <li>• 0h: Off</li> <li>• 1h: On</li> </ul>
b(1)	Green LED state: <ul style="list-style-type: none"> <li>• 0h: Off</li> <li>• 1h: On</li> </ul>
b(0)	Red LED state: <ul style="list-style-type: none"> <li>• 0h: Off</li> <li>• 1h: On</li> </ul>

**Table 3-92. Byte 3 Return Parameters**

MSB	Byte 3 – Internal Interrupt Status						LSB
b7	b6	b5	b4	b3	b2	b1	b0
b(7:3)	Reserved						
b(2)	DC Power Low Voltage: <ul style="list-style-type: none"> <li>• 0h: Supply voltage is normal</li> <li>• 1h: Supply voltage is low</li> </ul>						
b(1)	Sequence error: <ul style="list-style-type: none"> <li>• 0h: No error</li> <li>• 1h: Error</li> </ul>						
b(0)	Sequence abort error: <ul style="list-style-type: none"> <li>• 0h: No error</li> <li>• 1h: Error</li> </ul>						

The system reports **Sequence abort error** if Pulse Width Modulation (PWM) sequencer aborts the operation due to an error.

The system reports **Sequence error** if PWM sequencer encounter an error.

**Table 3-93. Byte 4 Return Parameters**

MSB	Byte 4 – Misc. Status						LSB
b7	b6	b5	b4	b3	b2	b1	b0
b(7:6)	Reserved						
b(5)	Watchdog timer timeout: <ul style="list-style-type: none"> <li>• 0h: No timeout</li> <li>• 1h: Timeout</li> </ul>						
b(4)	Product configuration error: <ul style="list-style-type: none"> <li>• 0h: No error</li> <li>• 1h: Error</li> </ul>						
b(3)	Leader versus follower operation: <ul style="list-style-type: none"> <li>• 0h: Leader</li> <li>• 1h: Follower</li> </ul>						
b(2)	Single versus dual controller configuration: <ul style="list-style-type: none"> <li>• 0h: Single</li> <li>• 1h: Dual</li> </ul>						
b(1:0)	Reserved						

The DLP3439 chip set is a dual controller solution. One controller is configured as leader and the other as follower.

The system should report "Dual" configuration. Bit (3) will be Leader or Follower depending on which controller is responding to the command.

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 or DMD combination
- Invalid controller or Power Management IC (DLPA200X or DLPA300X) combination
- Invalid flash build for the current controller, DMD, or Power Management IC configuration

The system sets the watchdog timer timeout bit if the system has been reset due to a watchdog timer timeout.

### 3.5.3 Read System Software Version (D2h)

This command reads the main application software version information for the display module.

#### 3.5.3.1 Read Parameters

This command has no read parameters.

#### 3.5.3.2 Return Parameters

[Table 3-94](#) describes the return parameters.

**Table 3-94. 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
Bytes 5 - 8	Reserved

### 3.5.4 Read Communication Status (D3h)

This command reads system status information for the display module.

#### 3.5.4.1 Read Parameters

The read parameters are described in [Table 3-95](#).

**Table 3-95. Read Parameters**

Parameter Bytes	Description
Byte 1	Command bus status selection

**Table 3-96. Byte 1 Read Parameters**

MSB	Byte 1 – Command Bus Status Selection						LSB
b7	b6	b5	b4	b3	b2	b1	b0
b(7:2)	Reserved						
b(1:0)	Command bus status selection:						
	<ul style="list-style-type: none"> <li>• 00: Reserved</li> <li>• 01: Reserved</li> <li>• 10: I<sup>2</sup>C only</li> <li>• 11: Reserved</li> </ul>						

This command returns the communication status for the specified command bus. For I<sup>2</sup>C only: This selection returns status bytes 5 through 6.

### 3.5.4.2 Return Parameters

Table 3-97 describes the return parameters.

**Table 3-97. Return Parameters**

Parameter Bytes	Description
Byte 1 - 4	Reserved
Byte 5	Communication status
Byte 6	Aborted op-code

All communication status error bits are cleared when the *Read Communication Status* is read.

**Table 3-98. Byte 5 Return Parameters**

MSB	Byte 5 – Communication Status						LSB
b7	b6	b5	b4	b3	b2	b1	b0
b(7)	Reserved						
b(6)	Bus timeout by display error: <ul style="list-style-type: none"> <li>• 0: No error</li> <li>• 1: Error</li> </ul>						
b(5)	Invalid number of command parameters: <ul style="list-style-type: none"> <li>• 0: No error</li> <li>• 1: Error</li> </ul>						
b(4)	Read command error: <ul style="list-style-type: none"> <li>• 0: No error</li> <li>• 1: Error</li> </ul>						
b(3)	Flash batch file error: <ul style="list-style-type: none"> <li>• 0: No error</li> <li>• 1: Error</li> </ul>						
b(2)	Command processing error: <ul style="list-style-type: none"> <li>• 0: No error</li> <li>• 1: Error</li> </ul>						
b(1)	Invalid command parameter value: <ul style="list-style-type: none"> <li>• 0: No error</li> <li>• 1: Error</li> </ul>						
b(0)	Invalid command error: <ul style="list-style-type: none"> <li>• 0: No error</li> <li>• 1: Error</li> </ul>						

The system sets the invalid command error bit when it does not recognize the command op-code. The invalid command op-code is reported in the I<sup>2</sup>C CMD error op-code byte of this status.

The system sets the invalid command parameter error bit when it detects that the value of a command parameter is not valid (for example, out of the allowed range).

The system sets the command processing error bit when a fault is detected when processing a command. In this case, the command aborts and the system moves on to the next command. The op-code for the aborted command is reported in the I<sup>2</sup>C CMD error op-code byte of this status.

The system sets the flash batch file error bit when an error occurs during the processing of a flash batch file. When this bit is set, typically another bit is set to indicate what kind of error was detected (for example, an invalid command error).

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.

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 command aborts and the system moves on to the next command. The op-code for the aborted command is reported in the I<sup>2</sup>C CMD error op-code byte of this status.

The system sets the bus timeout by display error bit when the display releases control of the bus after the bus timeout value is exceeded.

**Table 3-99. Byte 6 Return Parameters**

MSB	Byte 6 – CMD Error Op-Code						LSB
b7	b6	b5	b4	b3	b2	b1	b0
b(7:0)		I <sup>2</sup> C CMD error op-code					

The CMD error op-code is associated with various I<sup>2</sup>C communication status bits, and reports the op-code for an I<sup>2</sup>C command as noted.

### 3.5.5 Read Controller Device ID (D4h)

This command reads the Controller Device ID for the display module.

#### 3.5.5.1 Read Parameters

This command has no read parameters.

#### 3.5.5.2 Return Parameters

[Table 3-100](#) describes the return parameters.

**Table 3-100. Return Parameters**

MSB	Byte 1						LSB
b7	b6	b5	b4	b3	b2	b1	b0
b(7:4)		Reserved					
b(3:0)		Controller device ID					

The controller device ID are decoded using [Table 3-101](#).

**Table 3-101. Controller Device ID Decode**

Controller Device ID	Device Number	Application
09h	DLPC3439	Standalone (SD)

Unused controller device ID values are reserved.

### 3.5.6 Read DMD Device ID (D5h)

This command reads the DMD device ID for the display module.

#### 3.5.6.1 Read Parameters

The read parameters are described in [Table 3-102](#).

**Table 3-102. Read Parameters**

MSB	Byte 1 – DMD Register Selection						LSB
b7	b6	b5	b4	b3	b2	b1	b0
b(7:3)		Reserved					

b(2:0)	DMD data selection: <ul style="list-style-type: none"> <li>• 0h: DMD device ID</li> <li>• 1h – 7h: Reserved</li> </ul>
--------	--

### 3.5.6.2 Return Parameters

Table 3-103 describes the return parameters.

**Table 3-103. 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	68h/73h	0.47 1080P* (1920x1080, Sub-LVDS)

### 3.5.7 Read System Temperature (D6h)

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

#### 3.5.7.1 Read Parameters

The command has no read parameters.

#### 3.5.7.2 Return Parameters

Table 3-104 describes the return parameters.

**Table 3-104. Return Parameters**

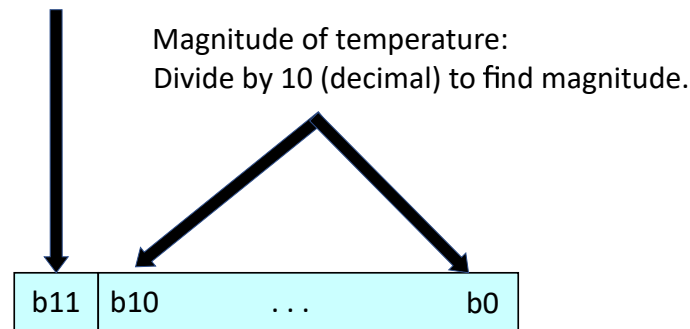
Parameter Bytes	Description
Byte 1	LSByte
Byte 2	MSByte

Figure 3-13 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'.

Sign of temperature:

0 = positive temperature:

1 = negative temperature:



Example #1: b(11:0) = 000110101010  
426d/10d = +42.6°C

Example #1: b(11:0) = 100110101010  
426d/10d = -42.6°C

**Figure 3-13. Bit Order and Definition for System Temperature**

### 3.5.8 Read Flash Build Version (D9h)

This command reads the controller flash version for the display module.

#### 3.5.8.1 Read Parameters

The command has no read parameters.

#### 3.5.8.2 Return Parameters

[Table 3-105](#) describes the return parameters.

**Table 3-105. Return Parameters**

Parameter Bytes	Description
Byte 1	Flash build version – patch LSBByte
Byte 2	Flash build version – patch MSByte
Byte 3	Flash build version – Minor
Byte 4	Flash build version – Major

The user specifies a version number for the controller flash build in the format specified by this command. This command allows the user to read back this version information.

### 3.5.9 Write Flash Batch File Delay (DBh)

This command is used to specify an execution time delay within a flash batch file for the display module.

#### 3.5.9.1 Write Parameters (DBh)

[Table 3-106](#) describes the write parameters.

**Table 3-106. Write Parameters**

Parameter Bytes	Description
Byte 1	Flash batch file delay (LSB)
Byte 2	Flash batch file delay (MSB)

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 I<sup>2</sup>C interfaces.

The flash batch file delay is to be specified in units of 1 ms (for example, 500 ms = 1F4h).

Typical use of this command is in the auto-init flash batch file (batch file 0), but is valid for use in any batch file (See [Write Execute Flash Batch File \(2Dh\)](#)).

### 3.5.10 Read DMD I/F Training Data (DCh)

This command is used to read back the DMD interface training data for the display module.

#### 3.5.10.1 Read Parameters

[Table 3-107](#) describes the command read parameters.

**Table 3-107. Read Parameters**

Parameter Bytes	Description
Byte 1	DMD I/F training data selection (see below)

MSB	Byte 1 – DMD I/F Data Selection						LSB
b7	b6	b5	b4	b3	b2	b1	b0

**Table 3-108. Byte 1 Read DMD I/F Training Data (DCh) Register Field Descriptions**

Bit	Type	Description
7-5	R	Reserved

**Table 3-108. Byte 1 Read DMD I/F Training Data (DCh) Register Field Descriptions (continued)**

Bit	Type	Description
4	R	Training data selection 0h = High/Low/Selected 1h = Full profile
3-0	R	Controller pin pair selection 0h = A 1h = B 2h = C 3h = D 4h = E 5h = F 6h = G 7h = H 8h - Fh = Reserved

This command returns the DMD I/F training data specified for the controller pin pair specified.

- High/Low/Selected: This selection returns bytes 1 through 4
- Full profile: This selection returns bytes 5 through 11

### 3.5.10.2 Return Parameters

Table 3-109 describes the return parameters.

**Table 3-109. DMD I/F Training Data Return Parameters**

Parameter Bytes	Description
Byte 1	High/Low/Selected (see below) (LSB)
Byte 2	High/Low/Selected (see below)
Byte 3	High/Low/Selected (see below)
Byte 4	High/Low/Selected (see below) (MSB)
Byte 5	Full profile (bits 7-0) (LSB)
Byte 6	Full profile (bits 15-8)
Byte 7	Full profile (bits 23-16)
Byte 8	Full profile (bits 31-24)
Byte 9	Full profile (bits 39-32)
Byte 10	Full profile (bits 47-40)
Byte 11	Full profile (bits 50-48) (MSB)

MSB	Byte 1 - 4						LSB
b7	b6	b5	b4	b3	b2	b1	b0

**Table 3-110. Byte 1 Read DMD I/F Training Data (DCh) Register Field Descriptions**

Bit	Type	Description
7-6	R	Reserved
5	R	Training error 0h = No error 1h = Error
4	R	Pin pair selected for training 0h = No 1h = Yes



**Table 3-110. Byte 1 Read DMD I/F Training Data (DCh) Register Field Descriptions (continued)**

Bit	Type	Description
3-0	R	Controller pin pair selection 0h = A 1h = B 2h = C 3h = D 4h = E 5h = F 6h = G 7h = H 8h - Fh = Reserved

**Table 3-111. Byte 2 Read DMD I/F Training Data (DCh) Register Field Descriptions**

Bit	Type	Description
7-6	R	Reserved
5-0	R	Selected DLL (delay-locked loop) value

**Table 3-112. Byte 3 Read DMD I/F Training Data (DCh) Register Field Descriptions**

Bit	Type	Description
7-6	R	Reserved
5-0	R	Low pass DLL value

**Table 3-113. Byte 4 Read DMD I/F Training Data (DCh) Register Field Descriptions**

Bit	Type	Description
7-6	R	Reserved
5-0	R	High pass DLL value

This command is typically used for debug or characterization of the controller to DMD interface.

The return data is specified by the read parameter data.

DMD I/F training tests/calibrates the DLL that is associated with each controller pin pair, trying each of the DLL parameter values (0 to 50), looking for a pass ('0') or fail ('1') response for each value. Thus, the full training profile for each pin pair is made up of a 51 bit pass/fail result. This result is provided on full profile bits 50:0.

The full profile response has a region of passing DLL values. The highest DLL value for this region is returned as the high pass DLL value, the smallest DLL value is returned as the low pass DLL value, and the algorithm selected value as the selected DLL value.

This command does not run the DMD I/F training algorithm. This is done automatically by the system. This command returns the result from the most recent training event.

### 3.6 Flash Update Commands

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

#### 3.6.1 Read Flash Update PreCheck (DDh)

This command is used to verify that a pending flash update (write) is appropriate for the specified block of the display module flash.

##### 3.6.1.1 Read Parameters

Table 3-114 describes the read parameters.

**Table 3-114. 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)

##### 3.6.1.2 Return Parameters

Table 3-115 describes the return parameters.

**Table 3-115. Return Parameters**

MSB	Byte 1 - Flash PreCheck Results						LSB
b7	b6	b5	b4	b3	b2	b1	b0

**Table 3-116. Flash Update PreCheck Register Field Descriptions**

Bit	Type	Description
7-3	R	Reserved
2	R	Package configuration (identifier) 0h = No error 1h = Error
1	R	Package configuration (collapsed) 0h = No error 1h = Error
0	R	Package size 0h = No error 1h = Error

This command is used in conjunction with the flash data type select 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.

The flash build data size specifies the size of the flash update package in bytes.

When the controller software receives the flash build data size, it will verify that the package is appropriate for the specified location. This includes size, identifier, sequence build type, and so forth.

A package size error indicates that the flash package is too large to fit into the specified location. A few examples are listed:

- If replacing the entire flash, the size of the flash build exceeds the size of the flash device in the system.
- If replacing the entire flash except for the user blocks, the size of the flash build will either overwrite some portion of the existing user blocks, or exceed the size of the flash device in the system.
- If replacing the look block, the size of the flash build exceeds the size of the existing look block in the flash.
- If replacing a single sequence (for example, a partial update), the size of the flash build exceeds the size of the existing splash screen.

A package configuration error indicates that the flash package is not appropriate for the flash update requested. An example is listed below.

- If replacing a single splash screen (for example, a partial update), and the specified splash screen index value (identifier) is not being used in the flash build. Partial updates can only replace an existing flash entity.

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 the error and update the flash anyway, the system will allow this. In this case, the user is responsible for any problems or system behaviors that arise from this. It should also be noted that this pre-check does *not* cover all possible mismatches that might arise when replacing blocks or partial blocks in the flash.

### 3.6.2 Write Flash Data Type Select (DEh)

This command is used to specify the type of data that will be written to or read from the flash of the display module.

#### 3.6.2.1 Write Parameters

[Table 3-117](#) describes the write parameters.

**Table 3-117. Write Parameters**

Parameter Bytes		Description	
Byte 1		Flash data type (See below)	
Byte 2		Optional: Partial data identifier (See Byte 1 Below)	
Byte 3		Optional: Partial data identifier (See Byte 1 Below)	
Byte 4		Optional: Partial data identifier (See Byte 1 Below)	

MSB	Byte 1						LSB
b7	b6	b5	b4	b3	b2	b1	b0

**Table 3-118. Flash Data Type Select Register Field Descriptions**

Bit	Type	Description - Flash data type
7-0	W	<p><b>Entire flash</b> 00h = Entire flash 01h = Reserved 02h = Entire flash except user calibration data and user scratchpad data 03h - 0Fh = Reserved</p> <p><b>TI software</b> 10h = Main software application 11h - 1Fh = Reserved</p> <p><b>TI application data</b> 20h = TI application data set (AOM) 21h - 2Fh = Reserved</p> <p><b>User batch files</b> 30h = User batch files 31h - 3Fh = Reserved</p> <p><b>Look data</b> 40h = Look data set 41h - 4Fh = Reserved</p> <p><b>Sequence data</b> 50h = Entire sequence data set 51h = Entire sequence data set (Reads only) 52h - 5Fh = Reserved</p> <p><b>Degamma/CMT data</b> 60h = Entire degamma/CMT data set 61h = Partial degamma/CMT data set (reads only) 62h - 6Fh = Reserved</p> <p><b>CCA data</b> 70h = CCA data set 71h - 7Fh = Reserved</p> <p><b>General LUT data</b> 80h = CCA data set</p>

The flash data type command must be provided each time a new flash write or read operation is desired to ensure that the appropriate data type parameters are provided. The system expects four parameter bytes regardless of whether all four bytes are needed. Any unused bytes should be set to zero.

The flash data length must be provided to indicate the amount of flash data that will be provided for each write or read transaction.

The specified flash data will be written to or read from flash using the [Write Flash Start \(E1h\)](#), [Write Flash Continue \(E2h\)](#), [Read Flash Start \(E3h\)](#), and [Read Flash Continue \(E4h\)](#) commands.

While all of the flash data sets indicated can be written/replaced in their entirety, a few will also support partial writes/updates. Partial update command parameters will use an “odd” command number (for example, 91h, B1h) which will indicate that one to three additional command parameter bytes of information must be provided to specify which subset of data is to be updated. The additional command parameter data required is described below.

**Table 3-119. Command Parameters for Partial Flash Data Set**

Data Type (Writes Only)	2nd CMD Parameter (Byte 2)	3rd CMD Parameter (Byte 2)	4th CMD Parameter (Byte 2)	Comments
Partial user splash screen set	Splash number	N/A	N/A	A splash screen will be specified by its splash screen number
Partial user scratchpad data set	Sector number	N/A	N/A	If this data set is allocated more than one sector, each sector can be specified (0 = 1st sector, 1 = 2nd sector, and so forth)

**Table 3-119. Command Parameters for Partial Flash Data Set (continued)**

Data Type (Writes Only)	2nd CMD Parameter (Byte 2)	3rd CMD Parameter (Byte 2)	4th CMD Parameter (Byte 2)	Comments
Partial sequence data set	Look number	Sequence index number		A sequence data set will be specified by its sequence index number.
Partial CMT data set	Look number	Sequence index number		A CMT data set will be specified by its CMT index number.
Partial user splash screen set	Splash number	N/A	N/A	A Splash screen will be specified by its Splash screen number.
Partial user scratchpad data set	Splash number	Sub-sector address (LSB)	Sub-sector address (MSB)	If this data set is allocated more than one sector, each sector can be specified (0 = 1st sector, 1 = 2nd sector, and so forth) The host is also allowed to specify the start address within the sector specified in byte 2. This address needs to be a relative address within the specified sector (that is, the value can range from 0 to 4096), and must be a 32-bit aligned byte address.

While all of the flash data sets indicated can be read starting at the beginning of the data set, a few will also support read starts at the beginning of a data subset. The partial update command parameters which use an “odd” command number (for example, 41h, 43h, 75h) will indicate that one to three additional command parameter bytes must be provided to specify the start location for these reads. The additional command parameter data required is described in the previous table.

It is expected that all TI formatted factory calibration data, including the golden ratio, the power-up RGB currents, and the user thermister LUT trim data, will be stored in the user calibration block of the flash. It will be the responsibility of the user to manage updates to this block, which may require the user to read the entire block, modify, and then rewrite the entire block when making an update within the block.

While flash processing requires that flash commands be executed in the proper order (for example, flash must be erased prior to being written), due to the flexibility provided for flash updates, command order checking is not provided.

It is recommended that the user make use of the [Read Flash Update PreCheck \(DDh\)](#) command before updating an existing flash build.

The system allows the user to allocate up to four separable blocks of flash space for their own use (user scratchpad data). The user can also specify the size of each of these blocks, where each block can be one or more sectors in (one sector = 4 kB). This is all defined via the GUI. It is the responsibility of the user to manage these data sets, including updates, which may require the user to read an entire sector, modify, and then rewrite the entire sector when making an update within a sector. References to an unavailable data set will result in an invalid command parameter value error in the communication status.

### 3.6.3 Write Flash Data Length (DFh)

This command is used to specify the length of the data that will be written to or read from the flash of the display module.

#### 3.6.3.1 Write Parameters

[Table 3-120](#) describes the write parameters.

**Table 3-120. Write Parameters**

Parameter Bytes	Description
Byte 1	Flash data length (LSB)
Byte 2	Flash data length (MSB)

Flash data length must be a multiple of four bytes.

The flash data length applies to each write or read transaction, not to the length of the data type selected.

The maximum data length allowed for each write transaction is 1024 bytes. The maximum data length allowed for each read transaction is 256 bytes.

While flash processing requires that flash commands be executed in the proper order (for example, flash must be erased prior to being written), due to the flexibility provided for flash updates, command order checking is not provided.

### 3.6.4 Write Erase Flash Data (E0h)

This command directs the display module to erase the specified flash data.

#### 3.6.4.1 Write Parameters

Table 3-121 describes the write parameters.

**Table 3-121. 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

When this command is executed, the system will erase all sectors associated with the data type specified by the flash data type select command. As such, this command does not make use of the flash data length parameter

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 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.

While flash processing requires that flash commands be executed in the proper order (for example, flash must be erased prior to being written), due to the flexibility provided for flash updates, command order checking is not provided.

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.

### 3.6.5 Write Flash Start (E1h)

This command is used to write data to the flash for the display module.

#### 3.6.5.1 Write Parameters

Table 3-122 describes the write parameters.

**Table 3-122. 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

The flash data length command must be used to specify how much data will be sent by the write flash start command.

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.

The flash error bit of the write 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.

While flash processing requires that flash commands be executed in the proper order (for example, flash must be erased prior to being written), due to the flexibility provided for flash updates, command order checking is not provided.

### 3.6.6 Write Flash Continue (E2h)

This command is used if more than 1024 bytes of data has to be written to flash of the display module.

#### 3.6.6.1 Write Parameters

Table 3-123 describes the write parameters.

**Table 3-123. 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

The flash data length command must be used to specify how much data will be sent by the write flash start command.

The [Write Flash Start \(E1h\)](#) 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.

The flash error bit of the write 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.

While flash processing requires that flash commands be executed in the proper order (for example, flash must be erased prior to being written), due to the flexibility provided for flash updates, command order checking is not provided.

### 3.6.7 Read Flash Start (E3h)

This command is used to read data from the flash for the display module.

#### 3.6.7.1 Read Parameters

The command has no read parameters.

#### 3.6.7.2 Return Parameters

Table 3-124 describes the return parameters.

**Table 3-124. Return Parameters**

Parameter Bytes	Description
Byte 1	Data byte 1
Byte 2	Data byte 2
Byte 3	Data byte 3

**Table 3-124. Return Parameters (continued)**

Parameter Bytes	Description
Byte 4	Data byte 4
Byte 5 ... n	Data byte 5 ... n

The flash data length command must be used to specify how much data is to be read by the read flash start command.

The read flash start command is used to read up to 256 bytes of data starting at the specified address, or at the first address of the data type selected. If more than 256 bytes are to be read, the read flash continue command must be used. Up to 256 bytes of data can be read with each read flash continue command, which starts at the end of the last data read.

While flash processing requires that flash commands be executed in the proper order (for example, flash must be erased prior to being written), due to the flexibility provided for flash updates, command order checking is not provided.

### 3.6.8 Read Flash Continue (E4h)

This command is used to continue to read data (if more than 256 bytes) from the flash for the display module.

#### 3.6.8.1 Read Parameters

The command has no read parameters.

#### 3.6.8.2 Return Parameters

[Table 3-125](#) describes the return parameters.

**Table 3-125. Return 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

The flash data length command must be used to specify how much data is to be read by the read flash continue command.

The read flash start command is used to read up to 256 bytes of data starting at the specified address, or at the first address of the data type selected. If more than 256 bytes are to be read, the read flash continue command must be used. Up to 256 bytes of data can be read with each read flash continue command, which starts at the end of the last data read.

While flash processing requires that flash commands be executed in the proper order (for example, flash must be erased prior to being written), due to the flexibility provided for flash updates, command order checking is not provided.



## 4 Revision History

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

<b>Changes from Revision * (August 2015) to Revision A (December 2023)</b>	<b>Page</b>
• Updates throughout document for consistency across related products.....	<a href="#">2</a>

## IMPORTANT NOTICE AND DISCLAIMER

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

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

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

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

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2023, Texas Instruments Incorporated