DLPC3437 Software Programmer's Guide

Programmer's Guide

Literature Number: DLPU062A
March 2018–Revised May 2019

Contents

List of Figures

List of Tables

(1)(2)

Revision History

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

Changes from Original (March 2018) to A Revision	Page
• Deleted section "Electrical Interface"	12
• Updated Section 2.1	12
• Updated Figure 2-1	12
• Changed section "I²C Transactions" with new section "I²C Interface Specification"	13
• Deleted 5EH entry in Table 3-1	15
• Changed reference link "maximum number of sequence vector bytes" in Figure 3-3	39
• Changed PAD to Power Management IC (PMIC)	67
• Changed PAD with PMIC globally	67
• Deleted section for 5Eh in Section 3.1.	70

(1) LightCrafter is a trademark of Texas Instruments.
(2) DLP is a registered trademark of Texas Instruments.

DLPC3437 Software Programmer's Guide

1.1 Introduction

1.1.1 Software Programmer's Guide Overview

This guide details the software interface requirements for a DLPC3437 dual controller based system. It defines all applicable communication protocols including I^2C, initialization, default settings and timing. The DLPC3437 system can be used in accessory products with Power Management IC DLPA3000 in Figure 1-1.

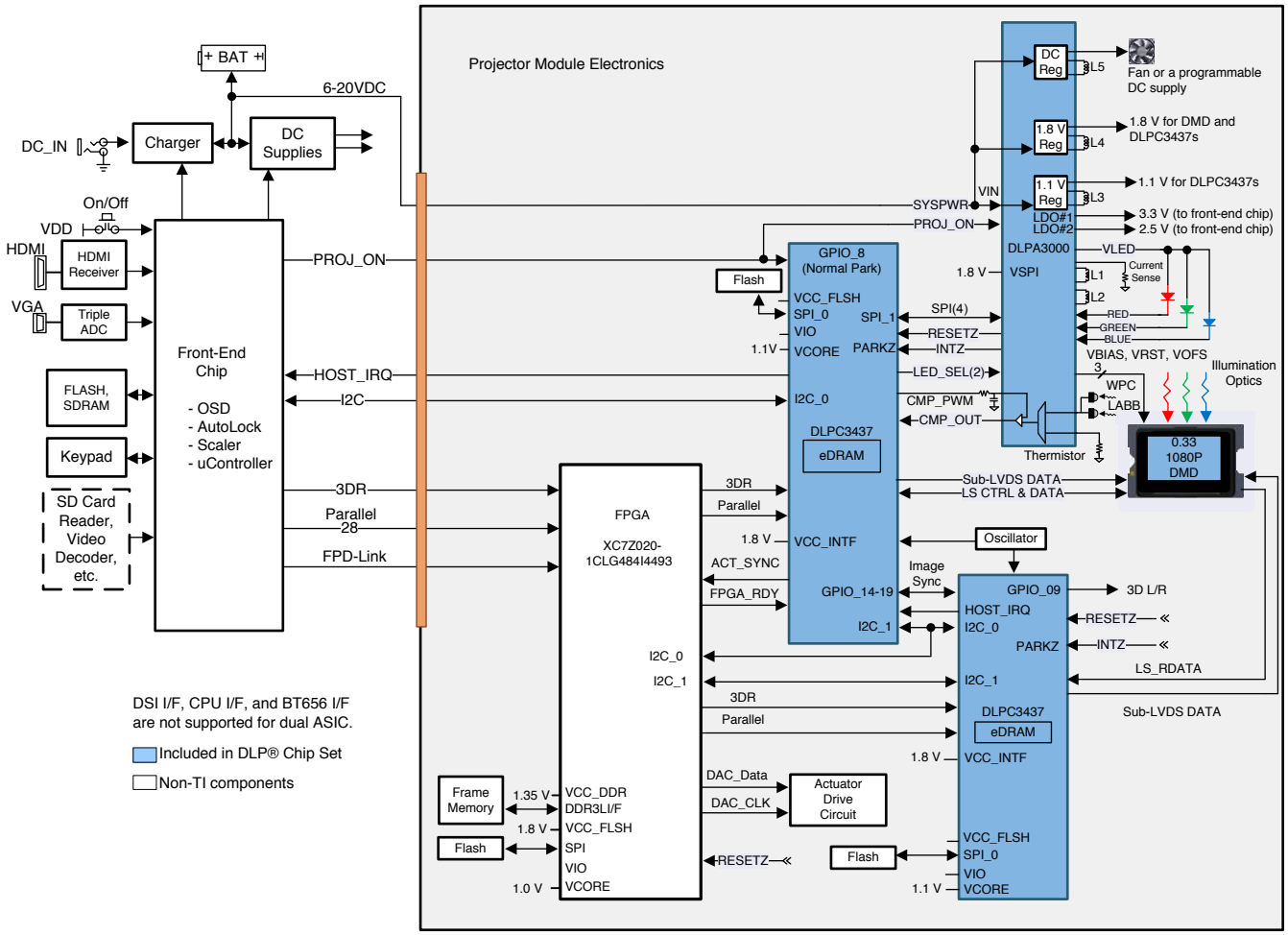

Figure 1-1. DLPC343x Embedded Configuration

1.1.1.1 I²C-Based Command Data Interface

The legacy interface configurations make use of an I²C interface for commands (conforming to the Philips I²C specification, up to 400 KHz) and a 24-bit parallel interface.

NOTE: I²C interface speeds up to a maximum of 100 kHz are supported.

1.1.2 Reference Documents

Table 1-1. Reference Documents

Document Number	Document Description
DLP® LightCrafter™ Display 3310 EVM User's Guide (DLPU063)	Overview of DLP Evaluation Module
DLPC3437 Display Controller (DLPS084)	Data sheet for DLP Display controller for DLP3310 DMD
DLPA3000 PMIC and High-Current LED Driver IC (DLPS052)	Data sheet for DLPA3000 PMIC/LED Driver
DLPA3005 PMIC and High-Current LED Driver IC (DLPS071)	Data sheet for DLPA3005 PMIC/LED Driver
DLP3310 0.33 1080p DMD (DLPS077)	Data sheet for DLP3310 0.33-inch DMD

2.1 System Initialization

This section describes the methodology used for system initialization.

2.1.1 Boot ROM Concept

The DLPC343x employs a boot ROM and associated boot software. This resident boot code consists of the minimum code necessary to complete the program loading. 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.

2.1.2 Resident Boot Software

The resident boot code consists of the minimum code necessary to load the ARM software from flash to internal RAM for execution.

2.1.3 HOST_IRQ Initialization Sequence

HOST_IRQ is a signal indicating the status of DLPC343x initialization. While reset is applied, HOST_IRQ resets to tri-state (an external pullup pulls the line high). HOST_IRQ remains tri-state (pulled high externally) until the microprocessor boot completes. While the signal is pulled high, the controller performs boot-up and auto-initialization

Immediately after boot-up, the microprocessor drives HOST_IRQ to a logic high state to indicate that the controller is performing auto-initialization (no real state change occurs on the external signal). Upon completion of auto-initialization, ARM software sets HOST_IRQ to a logic low state to indicate the completion of auto-initialization. At the falling edge, the system is said to enter the INIT_DONE state.

After auto-initialization completes, HOST_IRQ generates a logic high interrupt pulse to the host through software control; this interrupt indicates that the controller detects an error condition or requires service.

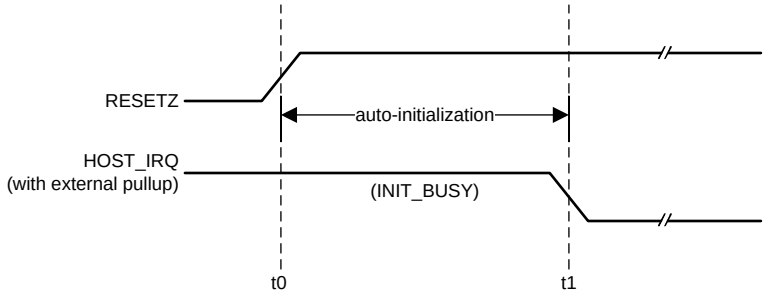

Figure 2-1. Host_IRQ Timing Diagram

2.2 Software Interface

In general, the DLPC3437 DUAL controller supports one set of software commands. This custom set of TI specific commands is applicable for use on I^2C command interface.

2.2.1 Software Command Philosophy

With DLPC3437, software processes all I²C interface commands. As such, no commands directly address or access controller registers, controller mailboxes, or any attached flash parts. All commands are high level, more abstract nature, decoupling the OEM from the internal hardware of the controller.

2.2.2 I²C Considerations

2.2.2.1 I²C Interface Specification

The protocol used in communicating information to DLPC3437 consist of a serial data bus conforming to the Philips I²C specification, up to 100 kHz. Commands are executed using I²C, where the DLPC3437 behaves as a slave.

The supported I²C transaction type for both writes and reads is shown in Table 2-1. The I²C interface supports variable-size transactions (i.e. variable number of bytes as parameters) depending on the command. The list of supported commands are discussed in the next section.

Table 2-1. I²C Write and Read Transactions

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

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

The standard parameter byte format is shown below:

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

2.2.2.2 Data Flow Control

While the I²C interface inherently supports flow control by holding the clock, this support is likely not sufficient for all transactions (sequence and CMT updates for example). In this case, the host software should make use of the Read Short Status to determine if the system is busy.

List of System Write/Read Software Commands

The commands supported by the I²C interface are discussed in the following sections.

Table 3-1. Supported TI Generic Commands

Offset	Command Type	Command Description	Reset Value	Default Action	Section
05h	Write	Write Input Source Select	1h	Test pattern	Section 3.1.1
06h	Read	Read Input Source Select			Section 3.1.2
09h	Write	Write External Video Processing Select	0h	Chroma interpolation	Section 3.1.3
0Ah	Read	Read External Video Chroma Processing Select			Section 3.1.4
0Dh	Write	Write Splash Screen Select		User specified	Section 3.1.5
0Eh	Read	Read Splash Screen Select			Section 3.1.6
0Fh	Read	Read Splash Screen Header			Section 3.1.7
14h	Write	Write Display Image Orientation		User specified	Section 3.1.8
15h	Read	Read Display Image Orientation			Section 3.1.9
16h	Write	Write Display Image Curtain	1h	Black	Section 3.1.10
17h	Read	Read Display Image Curtain			Section 3.1.11
1Ah	Write	Write Image Freeze	0h	No freeze	Section 3.1.12
1Bh	Read	Read Image Freeze			Section 3.1.13
20h	Write	Write 3-D Control	0h	Automatic	Section 3.1.14
21h	Read	Read 3-D Control			Section 3.1.15
22h	Write	Write LOOK Select		User specified	Section 3.1.16
23h	Read	Read LOOK Select			Section 3.1.17
26h	Read	Read Sequence Header Attributes			Section 3.1.18
27h	Write	Write Degamma/CMT Select		User specified	Section 3.1.19
28h	Read	Read Degamma/CMT Select			Section 3.1.20
29h	Write	Write CCA Select		User specified	Section 3.1.21
2Ah	Read	Read CCA Select			Section 3.1.22
2Dh	Write	Write Execute Batch File	0h		Section 3.1.23
30h	Write	Write 3-D Reference	0h	Next frame left	Section 3.1.24
31h	Write	Write GPIO [19:00] Control		User specified	Section 3.1.25
32h	Read	Read GPIO [19:00] Control			Section 3.1.26
33h	Write	Write GPIO [19:00] Outputs		User specified	Section 3.1.27
34h	Read	Read GPIO [19:00] Outputs			Section 3.1.28
35h	Write	Write Splash Screen Execute			Section 3.1.29
36h	Read	Read GPIO [19:00] Inputs			Section 3.1.30
4Bh	Write	Write FPD Pixel Map Mode			Section 3.1.31
4Ch	Read	Read FPD Pixel Map Mode			Section 3.1.32
4Dh	Write	Write FPGA Input Video Chroma Processing Select			Section 3.1.33
4Eh	Read	Read FPGA Input Video Chroma Processing Select			Section 3.1.34

Table 3-1. Supported TI Generic Commands (continued)

Offset	Command Type	Command Description	Reset Value	Default Action	Section
50h	Write	Write LED Output Control Method		User specified	Section 3.1.35
51h	Read	Read LED Output Control Method			Section 3.1.36
52h	Write	Write RGB LED Enable	7h	Enabled	Section 3.1.37
53h	Read	Read RGB LED Enable			Section 3.1.38
54h	Write	Write RGB LED Current		User specified	Section 3.1.39
55h	Read	Read RGB LED Current			Section 3.1.40
57h	Read	Read CAIC LED Max Available Power			Section 3.1.41
5Ch	Write	Write RGB LED Max Current		User specified	Section 3.1.42
5Dh	Read	Read RGB LED Max Current			Section 3.1.43
5Fh	Read	Read CAIC RGB LED Current			Section 3.1.44
60h	Write	Write XPR FPGA Input Image Size			Section 3.1.45
61h	Read	Read XPR FPGA Input Image Size			Section 3.1.46
62h	Write	Write XPR FPGA Source Select			Section 3.1.47
63h	Read	Read XPR FPGA Source Select			Section 3.1.48
64h	Read	Read XPR FPGA Version			Section 3.1.49
67h	Write	Write XPR FPGA Test Pattern Select			Section 3.1.50
68h	Read	Read XPR FPGA Test Pattern Select			Section 3.1.51
6Bh	Write	Write XPR FPGA Parallel Video Control			Section 3.1.52
6Ch	Read	Read XPR FPGA Parallel Video Control			Section 3.1.53
6Dh	Write	Write XPR FPGA Video Format Select			Section 3.1.54
6Eh	Read	Read XPR FPGA Video Format Select			Section 3.1.55
6Fh	Read	Read XPR FPGA Status			Section 3.1.56
70h	Write	Write Actuator Latency			Section 3.1.57
71h	Read	Read Actuator Latency			Section 3.1.58
72h	Write	Write Actuator Gain			Section 3.1.59
73h	Read	Read Actuator Gain			Section 3.1.60
74h	Write	Write Segment Length			Section 3.1.61
75h	Read	Read Segment Length			Section 3.1.62
76h	Write	Write Manual Actuator Sync Delay			Section 3.1.63
77h	Read	Read Manual Actuator Sync Delay			Section 3.1.64
78h	Write	Write Manual Actuator Offset			Section 3.1.65
79h	Read	Read Manual Actuator Offset			Section 3.1.66
7Ah	Write	Write Actuator Fixed Output			Section 3.1.67
7Bh	Read	Read Actuator Fixed Output			Section 3.1.68
7Ch	Write	Write Actuator Direction			Section 3.1.69
7Dh	Read	Read Actuator Direction			Section 3.1.70
7Eh	Write	Write Actuator Enable			Section 3.1.71
7Fh	Read	Read Actuator Enable			Section 3.1.72

Table 3-1. Supported TI Generic Commands (continued)

Offset	Command Type	Command Description	Reset Value	Default Action	Section
80h	Write	Write Local Area Brightness Boost Control	1h	Manual strength control	Section 3.1.73
81h	Read	Read Local Area Brightness Boost Control			Section 3.1.74
84h	Write	Write CAIC Image Processing Control		User specified	Section 3.1.75
85h	Read	Read CAIC Image Processing Control			Section 3.1.76
86h	Write	Write CCA Control	1h	Enabled	Section 3.1.77
87h	Read	Read CCA Control			Section 3.1.78
88h	Write	Write Keystone Correction Control			Section 3.1.79
89h	Read	Read Keystone Correction Control			Section 3.1.80
B2h	Write	Write Border Color	0h	Black	Section 3.1.81
B3h	Read	Read Border Color			Section 3.1.82
BBh	Write	Write Keystone Projection Pitch Angle			Section 3.1.83
BCh	Read	Read Keystone Projection Pitch Angle			Section 3.1.84
D0h	Read	Read Short Status			Section 3.1.85
D1h	Read	Read System Status			Section 3.1.86
D2h	Read	Read System Software Version			Section 3.1.87
D3h	Read	Read Communication Status			Section 3.1.88
D4h	Read	Read Controller Device ID			Section 3.1.89
D5h	Read	Read DMD Device ID			Section 3.1.90
D6h	Read	Read System Temperature			Section 3.1.91
D9h	Read	Read Flash Build Version			Section 3.1.92
DBh	Write	Write Batch File Delay		User specified	Section 3.1.93
DCh	Read	Read DMD I/F Training Data			Section 3.1.94
DDh	Read	Flash Update PreCheck		Entire flash	Section 3.1.95
DEh	Write	Flash Data Type Select			Section 3.1.96
DFh	Write	Flash Data Length			Section 3.1.97
E0h	Write	Erase Flash Data			Section 3.1.98
E1h	Write	Write Flash Start			Section 3.1.99
E2h	Write	Write Flash Continue			Section 3.1.100
E3h	Read	Read Flash Start			Section 3.1.101
E4h	Read	Read Flash Continue			Section 3.1.102

3.1 System Write/Read Commands

3.1.1 Write Input Source Select (Offset = 05h) [Reset = 01h]

This command is used to control the idle mode algorithms for the display module.

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

Table 3-2. Write Input Source Select (05h) Register Field Descriptions

Bit	Type	Description
7-2	R	Reserved
1-0	W	Input Source 0h = Parallel Video Port from XPR FPGA. The XPR FPGA can take LVDS, RGB video as input or generate TPG and convert them to parallel RGB video. 1h = Reserved. DLPC3437 internal test pattern is not supported. 2h = Splash Screen only in non-XPR mode. 3h = Reserved Write Splash Screen Select (0Dh)

NOTE: When selecting the Parallel Video Port, there is a set of associated commands that are only applicable to this source selection. These associated commands are the Write XPR FPGA Source Select (Section 3.1.47) and the Write External Video Chroma Processing Select (Section 3.1.3) command.

When selecting the Test Pattern Generator from FPGA, there is one associated command that is only applicable to this source selection. This associated command is the Write XPR FPGA Test Pattern Select (Section 3.1.50) command

When selecting the Splash Screen, there are two associated commands that are only applicable to this source selection. These associated commands are the Write Splash Screen Select (Section 3.1.5) and Write Splash Screen Execute (Section 3.1.29) commands.

These associations are also shown in Table 3-3.

Table 3-3. Source Specific Associated Commands

Source Specific Associated Commands	Input Source Select Options		
	External Video Port	Test Pattern Generator	Splash Screen[(1)]
Write External Video Source Format Select	Only	N/A	N/A
Write External Video Chroma Processing Select	Only	N/A	N/A
Write External Parallel I/F Manual Image Framing	Only	N/A	N/A
Write Test Pattern Select	N/A	Only	N/A
Write Splash Screen Select	N/A	N/A	Only
Write Splash Screen Execute	N/A	N/A	Special

(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.

These commands (other than *Write Splash Screen Execute*) describe the characteristics of their associated source, and once these settings are defined the system stores them. Afterwards, each time an input source selection is made (using the *Write Input Source Select* command), the system remembers the settings described by the commands associated with the selected source, and automatically applies them. The user only needs to send these associated commands when the source is first defined, or when the source characteristics for that port must be changed. The appropriate associated commands must be updated when source characteristics change.

The user can send source-associated commands every time they make an input source selection. The source associated commands should be sent prior to sending the *Write Input Source Select* command. 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 Input Source Select* command), the controller applies these new settings, as in the following example:

1. The user sends the following commands (active input source = test pattern generator):
   - *Write Image Freeze* = Freeze
   - *Write External Video Source Format Select* (settings stored, command not executed)
   - *Write External Video Chroma Processing Select* (settings stored, command not executed)
   - *Write Input Source Select* = external port (see step 2 below)
   - *Write Image Freeze* = unfreeze
2. When the *Write Input Source Select* command is received, the software applies the settings from these external video port-associated commands:
   - *External Video Source Format Select*
   - *External Video Chroma Processing Select*
   - *External Input Image Size*
   - *External Parallel Manual Image Framing (as appropriate – for example, if parallel port selected)*
   - *External CPU Video Sync Mode (as appropriate – for example, if CPU port selected)*

If source-associated commands are sent for a source that is already active, the controller software executes these commands when received, as in the following example:

- The user sends the following commands (active input source = external video port):
  - *Write Image Freeze* = freeze
  - *Write External Video Source Format Select* (command executed)
  - *Write External Video Chroma Processing Select* (command executed)
  - *Write Image Freeze* = unfreeze

The rest of the commands that apply to image setup have settings applicable across all source selections, and typically remain the same across the three input source selections. A few examples are *Write Display Size* and *Write Display Image Orientation*. A representative list of these commands is shown in Table 3-4.

Table 3-4. Common Commands

Common Commands	Input Source Select Options		
	External Video Port	Test Pattern Generator	Splash Screen
Write Display Image Orientation	Common	Common	Common
Write Display Image Curtain	Common	Common	Common
Write Look Select	Common	Common	Common
Write Sequence Select	Common	Common	Common
Write Local Area Brightness Boost Control	Common	Common	Common
Write CAIC Image Processing Control	Common	Common	Common

While the values for these commands may be the same across the different input source types, the hardware settings may change (for example: Display image size = 1080p = DMD size – the external port input source size is WXGA, which is scaled up to the display size of 1080p. If the user changes to the TPG Input Source, the size of the test pattern must match the size of the DMD. Therefore, the scaler settings must to be changed). The controller software manages the underlying hardware settings. This also applies to those commands which specify automatic operation (for example, Write Idle Mode Select = Auto Idle Mode Enable). While the automatic setting remains the same, the underlying algorithm might change its settings based on the characteristic of the selected source.

- The user is required to specify the active data size for all external input sources using the Write Input Image Size command. In addition, for input image data on the Parallel bus that doesn't provide data framing information, the user is required to provide manual framing data using the Parallel I/F Manual Image Framing command.
- The software generates a selected test pattern at the resolution of the DMD.
- The user typically sees the *Write Image Freeze* command for information on hiding on-screen artifacts when selecting an input source.

3.1.2 Read Input Source Select (Offset = 06h)

This command reads the state of the image input source for the display module.

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

Table 3-5. Read Input Source Select (06h) Register Field Descriptions

Bit	Type	Description
7-2	R	Reserved
1-0	R	Input source 0h = Parallel Video Port from FPGA 1h = Reserved 2h = Splash Screen only supported in non-XPR 3h = Reserved

3.1.3 Write External Video Chroma Processing Select (Offset = 09h) [Reset = 0h]

This command is used to specify the CSC Coefficient Set parameter.

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

Table 3-6. Write External Video Chroma Processing Select (09h) Register Field Descriptions

Bit	Type	Description
7-2	R	Reserved
1-0	W	CSC Coefficient Set (Color Space)

NOTE: This command is used in conjunction with the Write Input Source Select command (Section 3.1.1). The software retains the settings for this command until changed using this command. These settings are automatically applied each time the External Video Port is selected.

NOTE: CSC coefficient sets are stored in Flash until needed.

NOTE: CSC coefficient sets are specified in Byte 1 by an enumerated value (for example, 0, 1, 2, or 3). The set stored in '0' is ITU-R BT. Rec. 601 standard. The other three sets are customer definable via User.

3.1.4 Read External Video Chroma Processing Select (Offset = 0Ah)

This command is used to read the CSC Coefficient Set parameter.

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

Table 3-7. Read External Video Chroma Processing Select (0Ah) Register Field Descriptions

Bit	Type	Description
7-2	R	Reserved
1-0	R	CSC Coefficient Set (Color Space)

3.1.5 Write Splash Screen Select (Offset = 0Dh) [Reset = User specified]

This command is used to select a stored splash screen to be displayed on the display module.

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

This command is used in conjunction with the Write Input Source Select (Section 3.1.1) and the Write Splash Screen Execute (Section 3.1.29) commands. It specifies which splash screen is to be displayed when the Input Source Select command selects splash screen as the image source. 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 (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)

NOTE: The Splash Screen is a unique source because it is read from Flash and sent down the processing path of the controller one time, to be stored in memory for display at the end of the processing path. As such, all image processing settings (for example image crop, image orientation, display size, splash screen select, splash screen as input source, and so forth) should be set appropriately by the user before executing the Write Splash Screen Execute command.

NOTE: *It is important that the user review the notes for the Write Input Source Select command in Section 3.1.1 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.*

NOTE: The availability of splash screens is limited by the available space in flash memory.

NOTE: All splash screens must be landscape oriented.

NOTE: 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 product 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.

NOTE: For dual controller applications which support DMD resolutions up to 1980 x 1080, the minimum splash image size allowed for flash storage is 854 x 480, with the maximum being the resolution of the product DMD. Typical splash image sizes for flash are 854 x 480. The full resolution size is typically used to support an “Optical Test” splash screen.

NOTE: The user is responsible for specifying how the splash image displays on the screen.

NOTE: When this command is received while Splash Screen is the active source, other than storing the specified splash screen value, the only action the controller software takes is to obtain the header information from the selected splash screen and store this in internal memory. Then, when the Write Splash Screen Execute command is received, the controller software uses this stored information to set up the processing path prior to pulling the splash data from flash.

3.1.6 Read Splash Screen Select (Offset = 0Eh)

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

Table 3-8. Return Parameters

Parameter Bytes	Description
Byte 1	Splash screen selected (integer)

3.1.7 Read Splash Screen Header (0Fh)

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

3.1.7.1 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) and the command is executed.

Table 3-9. Read Parameters

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

3.1.7.2 Return Parameters

Table 3-10 describes the return parameters.

Table 3-10. 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 3-11.

Table 3-11. 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

3.1.8 Write Display Image Orientation (Offset = 14h) [Reset = User specified]

This command specifies the image orientation of the displayed image for the display module.

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

Table 3-12. Write Display Image Orientation (14h) Register Field Descriptions

Bit	Type	Description
7-3	R	Reserved
2	W	Short axis image flip 0h = Image not flipped. 1h = Image flipped.
1	W	Long axis image flip 0h = Image not flipped. 1h = Image flipped.
0	R	Reserved

Figure 3-1 shows the short-axis flip.

Figure 3-1. Long-Axis Flip

Figure 3-2 shows the short-axis flip.

Figure 3-2. Short-Axis Flip

3.1.9 Read Display Image Orientation (Offset = 15h)

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

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

Table 3-13. Read Display Image Orientation (15h) Register Field Descriptions

Bit	Type	Description
7-3	R	Reserved
2	R	Short axis image flip 0h = Image not flipped. 1h = Image flipped.
1	R	Long axis image flip 0h = Image not flipped. 1h = Image flipped.
0	R	Reserved

3.1.10 Write Display Image Curtain (Offset = 16h) [Reset = 01h]

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

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

Table 3-14. Write Display Image Curtain (16h) Register Field Descriptions

Bit	Type	Description
7-4	R	Reserved
3-1	W	Select curtain color 0h = Black 1h = Red 2h = Green 3h = Blue 4h = Cyan 5h = Magenta 6h = Yellow 7h = White
0	W	Curtain enable 0h = Curtain disabled 1h = Curtain enabled

NOTE: 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.

NOTE: The curtain color specified by this command is separate from the border color defined in the Write Border Color command (Section 3.1.81), even though they are both displayed using the curtain capability.

3.1.11 Read Display Image Curtain (Offset = 17h)

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

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

Table 3-15. Read Display Image Curtain (17h) Register Field Descriptions

Bit	Type	Description
7-4	R	Reserved
3-1	R	Select curtain color 0h = Black 1h = Red 2h = Green 3h = Blue 4h = Cyan 5h = Magenta 6h = Yellow 7h = White
0	R	Curtain enable 0h = Curtain disabled 1h = Curtain enabled

3.1.12 Write Image Freeze (Offset = 1Ah) [Reset = 0h]

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

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

Table 3-16. Write Image Freeze (1Ah) Register Field Descriptions

Bit	Type	Description
7-1	R	Reserved
0	W	Image freeze 0h = Image freeze disabled 1h = Image freeze enabled

Normal use of the Image Freeze capability typically has two main functions. The first function is to allow the end user to freeze the current image on the screen for their own uses. The second function is to allow the user (host system/OEM) to reduce/prevent system changes from showing up on the display as visual artifacts. In this second case, the image would be frozen, system changes would be made, and when complete, the image is unfrozen. In all cases, when the image is unfrozen, the display starts showing the most recent input image. Thus input data between the freeze point and the unfreeze point is lost. Suggestions to the host system for the types of image changes likely to necessitate the use of the image freeze command to hide artifacts are discussed in Section 3.1.12.1.

The controller software does not freeze or unfreeze the image for any reason except when explicitly commanded by the Write Image Freeze command.

NOTE: It is important that the user review the notes for the Write Input Source Select command in Section 3.1.1 to understand the concept of source associated commands. This concept determines when source associated commands are executed by the system. Note, Freeze command doesn't work on Splash screen on a dual DLPC3437 system.

NOTE: If the OEM chooses not to make use of Image Freeze, it is recommended that they change the source itself before changing image parameters to minimize transition artifacts.

3.1.12.1 Use of Image Freeze to Reduce On-Screen Artifacts

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

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

Because commands to the controller are processed serially, no special timing or delay is required between these commands. It is suggested that the number of commands placed between the freeze and unfreeze be kept small, as it is likely not desirable for the image to be frozen for a “long” period of time. A list of commands that may produce image artifacts are listed in Table 3-17. However, this is not an all-inclusive list, and the user is ultimately responsible for determining if and when use of the image freeze command meets requirements of the application.

Table 3-17. Partial List of Commands that May Benefit from the Use of Image Freeze

Command	Command Offset	Notes
Write Input Source Select	05h	
Write External Video Source Format Select	07h	Reference source not found.
Write Look Select	22h	

Table 3-18 and Table 3-19 show a few examples of how to use the image freeze command.

Table 3-18. 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 Display Image Orientation	Potential data processing commands that may be required for proper display of TPG.
Write Test Patern Generator Select	Set up TPG.
Write Image Freeze = unfreeze	

Table 3-19. Test Pattern Generator Example Using Image Freeze

Command	Notes
Write Image Freeze = freeze	
Write Display Image Orientation, Write Test Pattern Select	Potential data processing commands that may be required for proper display of test pattern image. These should be set before the Write Input Source Select command.
Write Input Source Select = test pattern generator	
Write Image Freeze = unfreeze	

3.1.13 Read Image Freeze (Offset = 1Bh)

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

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

Table 3-20. Read Image Freeze (1Bh) Register Field Descriptions

Bit	Type	Description
7-1	R	Reserved
0	R	Image freeze 0h = Image freeze disabled 1h = Image freeze enabled

3.1.14 Write 3-D Control (Offset = 20h) [Offset = 0h]

This command is used to control 3-D functionality for the display module.

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

Table 3-21. Write 3-D Control (20h) Register Field Descriptions

Bit	Type	Description
7	R	Reserved
6	W	Polarity of 3-D Reference (External Only) 0h = Correct – No Inversion Required. 1h = Incorrect – Inversion Required.
5	W	Frame Dominance 0h = Left Dominant. (Data sent left eye first)
4-2	R	Reserved
1	W	Source of 3-D Reference 0h = Internal Reference Generator NOT supported 1h = External (SLT_3DR Pin)
0	R	Reserved

NOTE: The system automatically enables 3-D operation when appropriate, basing this decision on the source frame rate, and whether 3-D sequences are available to the system (loaded in flash, for example). The 3-D parameters specified by this command takes effect following the next VSYNC.

NOTE: 3-D image data must always be sent frame sequential (that is, syncs and blanking to be sent between every eye frame), at frame rates greater than approximately 94 Hz (controller does not support frame rate multiplication). Internal Reference Generator is not supported in Dual controller system.

NOTE: Internal reference generator is not supported on dual controller DLPC3437.

NOTE: 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 3-22 shows which 3-D Reference source can be used with each 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 3-22. 3D Control

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

NOTE: The Write 3-D Reference command should be use with this selection.

NOTE: For frame sequential 3-D, Frame Dominance determines which eye frames in the data stream go together to make up a single 3-D image. Left dominance indicates that the first eye frame of a pair is left, the second eye frame is right. Right dominance indicates that the first eye frame of a pair is right, the second eye frame is left). This is important for proper operation of display histograms (which span both eye frames of a single image), and when the image is frozen, as we want to be sure we display the correct two eye frames together. The frame dominance control must not be used to attempt correction for misalignment of the 3-D reference signal to the image data.

3.1.15 Read 3-D Control (Offset = 21h)

This command is used to read the state of the 3-D control function for the display module.

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

Table 3-23. Read 3-D Control (21h) Register Field Descriptions

Bit	Type	Description
7	R	Reserved
6	R	Polarity of 3-D Reference (External Only) 0h = Correct – No Inversion Required. 1h = Incorrect – Inversion Required.
5	R	Frame Dominance 0h = Left Dominant. (Data sent left eye first) 1h = Right Dominant. (Data sent right eye first)
4-2	R	Reserved
1	R	Source of 3-D Reference 0h = Internal Reference Generator NOT supported 1h = External (SLT_3DR Pin)
0	R	3-D Mode Control 0h = 2-D Operation 1h = 3-D Operation

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

3.1.16 Write LOOK Select (Offset = 22h) [Reset = User specified]

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

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

Table 3-24. Write LOOK Select (22h) Register Field Descriptions

Bit	Type	Description
7-0	W	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. Based on the LOOK selected and measured data obtained from an appropriate light sensor, the software automatically selects and loads the most appropriate sequence or duty cycle set available in the LOOK, to get as close as possible to the target white point.

LOOKs are specified in this byte by an enumerated value (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 should specify in addition to the LOOK. 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.

3.1.17 Read LOOK Select (Offset = 23h)

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

Table 3-25. 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 3-26. Byte 1 Read LOOK Select (23h) Register Field Descriptions

Bit	Type	Description
7-0	R	LOOK number

Table 3-27. Byte 2 Read LOOK Select (23h) Register Field Descriptions

Bit	Type	Description
7-0	R	Sequence number

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

NOTE: Sequences are specified by an enumerated value (that is, 0, 1, 2, 3, and so forth), and the value returned by this command is the sequence currently selected by the LOOK algorithm when this command is received.

NOTE: The current sequence frame rate is returned as a count that is specified in units of 66.67 ns (based on the internal 15-MHz clock used to time between input frame syncs), and is valid regardless of whether controller software made the sequence/duty cycle selection, or the user made the selection. The frame rate is specified in this way to enable fast and simple comparisons to the frame count by the software.

3.1.18 Read Sequence Header Attributes (Offset = 26h)

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

3.1.18.1 Return Parameters

Table 3-28 describes the return parameters.

Table 3-28. 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	Max 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	Max 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 should match.

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

Figure 3-3. Bit Weight and Bit Order for Duty Cycle Data

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

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

The sequence maximum and minimum frame counts are specified in units of 66.67 ns (based on the internal 15-MHz clock used to time between input frame syncs). These are specified in this way to enable fast and simple comparisons to the frame count by software.

The maximum number of sequence vector bytes is defined in Table 3-29.

MSB	Byte 15 and 30						LSB
b7	b6	b5	b4	b3	b2	b1	b0

Table 3-29. Read Sequence Header Attributes (26h) Register Field Descriptions

Bit	Type	Description
7-4	R	Reserved
3-0	R	Maximum number of sequence vectors

3.1.19 Write Gamma/CMT Select (Offset = 27h) [Reset = User specified]

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

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

Table 3-30. Write Gamma/CMT Select(27h) Register Field Descriptions

Bit	Type	Description
7-0	W	Degamma/CMT LUT Index Number

NOTE: Degamma/CMT LUTs are stored in Flash until needed.

NOTE: The Degamma/CMT LUT Number specified by the user determines the degamma applied by the system.

NOTE: For TI software purposes, this Degamma/CMT LUT number is the CMT Index number in the Flash structure. Thus, if there is a degamma of 1.5 (for example) at CMT Index 0, then every sequence generates a CMT Index of 0 that references a degamma of 1.5 that is appropriate for each respective sequence.

3.1.20 Read Gamma/CMT Select (Offset = 28h)

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

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

Table 3-31. Read Gamma/CMT Select (28h) Register Field Descriptions

Bit	Type	Description
7-0	R	Degamma/CMT LUT Index Number

3.1.21 Write CCA Select (Offset = 29h) [Reset = User specified]

This command is used to select a specific set of CCA parameters (to specify the color points) for the display module.

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

Table 3-32. Write CCA Select (29h) Register Field Descriptions

Bit	Type	Description
7-0	W	CCA Parameter Set

NOTE: CCA parameter sets are used to set a target color points for the system. The sets are stored in Flash until needed.

NOTE: CCA parameter sets are specified in this byte by an enumerated value (that is, 0, 1, 2, 3, and so forth). This number specifies the actual CCA number reference in the flash structure.

3.1.22 Read CCA Select (Offset = 2Ah)

This command is used to read the status of the CCA select command for the display module.

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

Table 3-33. Read CCA Select (2Ah) Register Field Descriptions

Bit	Type	Description
7-0	R	CCA Parameter Set

3.1.23 Write Execute Flash Batch File (Offset = 2Dh)

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

Table 3-34. Write Parameters

Parameter Bytes	Description
Byte 1	Batch file number

This command is used to command the execution of 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 command parameters into a Flash batch file, with the exception of those listed in Table 3-35. Flash batch files are created using the GUI tool, and then stored in the Flash build. One example for a Flash batch file might be the commands and command parameters required for initialization of the system after power-up.

NOTE: The Flash batch file numbers to be specified in this byte are enumerated values (that is, 0, 1, 2, 3, and so forth).

NOTE: Flash batch file 0 is a special Auto-Init batch file that is run automatically by the DLPC3437 software immediately after system initialization has been completed. As such, Flash batch file 0 is not typically called using the Write Execute Batch File command (although the system allows it). This special Flash batch file would typically be used to specify the source to be used (for example, splash screen, data port) once the system is initialized.

NOTE: 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). If it is desired to have two batch files executed back to back, they should be called by back to back execute batch file commands.

NOTE: The system provides the ability to add an execution delay between commands within a Flash batch file. This is done using the Write Flash Batch File Delay command (Section 3.1.93).

NOTE: The order of command execution for commands within a Flash batch file is the same as if the commands had been received over the I^2C port.

Table 3-35. Flash Batch File Operations

Command	Offset	Applicable
Write Command Synchronization	N/A	Reference source not found.
Write Execute Flash Batch File	2D	No
Flash Data Type Select	DE	
Flash Data Length	DF	
Erase Flash Data	E0	
Write Flash Start	E1	
Write Flash Continue	E2	
Write Internal Mailbox Address	E8	
Write Internal Mailbox	E9	
All Read commands	Various	Various

3.1.24 Write 3-D Reference (Offset = 30h) [Reset = 0h]

This command is used to provide a 3-D reference for the display module.

Table 3-36. Write Parameters

Parameter Bytes	Description
Byte 1	Batch file number

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

Table 3-37. Write Execute Flash Batch File (2Dh) Register Field Descriptions

Bit	Type	Description
7-1	R	Reserved
0	W	3-D Reference 0h = Next Frame Left 1h = Next Frame Right

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 can be provide to the display as a hardware signal or by using this command (selection is made using the Write 3-D Control command in Section 3.1.14). When using this command as the reference, it is recommend that the command be sent every frame, or at least at the start of each eye pair (for example, 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 command.

NOTE: When the software received the Write 3-D Reference command, it applies the parameter value at the next VSYNC (that is, it applies the parameter value to the image data following the next VSYNC or Start of Frame command).

NOTE: When the softwarereceived this command, it sets up the internal controller 3-D reference generator. If the command is sent every frame, software can monitor to ensure that the output of the internal controller 3-D reference generator is still correct.

3.1.25 Write GPIO[19:00] Control (Offset = 31h) [Reset = User specified]

This command is used to specify how GPIO(19:09) and GPIO(07:04) of the display module are to be used.

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

Table 3-38. Byte 1 Write GPIO[19:00] Control (31h) Register Field Descriptions

Bit	Type	Description
7-6	W	GPIO (12) 0h = User Defined 1h = Input 2h = Output (Standard) 3h = Output (Open Drain)
5-4	W	GPIO (11) 0h = User Defined 1h = Input 2h = Output (Standard) 3h = Output (Open Drain)
3-2	W	GPIO (10) 0h = User Defined 1h = Input 2h = Output (Standard) 3h = Output (Open Drain)
1-0	W	GPIO (09) 0h = User Defined 1h = Input 2h = Output (Standard) 3h = Output (Open Drain)

Table 3-39. Byte 2 Write GPIO[19:00] Control (31h) Register Field Descriptions

Bit	Type	Description
7-6	W	GPIO (16) 0h = User Defined 1h = Input 2h = Output (Standard) 3h = Output (Open Drain)
5-4	W	GPIO (15) 0h = User Defined 1h = Input 2h = Output (Standard) 3h = Output (Open Drain)
3-2	W	GPIO (14) 0h = User Defined 1h = Input 2h = Output (Standard) 3h = Output (Open Drain)
1-0	W	GPIO (13) 0h = User Defined 1h = Input 2h = Output (Standard) 3h = Output (Open Drain)

Table 3-40. Byte 3 Write GPIO[19:00] Control (31h) Register Field Descriptions

Bit	Type	Description
7-6	R	Reserved
5-4	W	GPIO (19) 0h = User Defined 1h = Input 2h = Output (Standard) 3h = Output (Open Drain)
3-2	W	GPIO (18) 0h = User Defined 1h = Input 2h = Output (Standard) 3h = Output (Open Drain)
1-0	W	GPIO (17) 0h = User Defined 1h = Input 2h = Output (Standard) 3h = Output (Open Drain)

Table 3-41. Byte 4 Write GPIO[19:00] Control (31h) Register Field Descriptions

Bit	Type	Description
7-6	W	GPIO (07) 0h = User Defined 1h = Input 2h = Output (Standard) 3h = Output (Open Drain)
5-4	W	GPIO (06) 0h = User Defined 1h = Input 2h = Output (Standard) 3h = Output (Open Drain)
3-2	W	GPIO (05) 0h = User Defined 1h = Input 2h = Output (Standard) 3h = Output (Open Drain)
1-0	W	GPIO (04) 0h = User Defined 1h = Input 2h = Output (Standard) 3h = Output (Open Drain)

NOTE: GPIO(19:09) and GPIO(07:04) can individually function as defined by the OEM in the GUI, or their function can be redefined on-the-fly using this command. When their functions are redefined by this command, the Write GPIO[19:00] Outputs (

Section 3.1.27) and Read GPIO[19:00] Inputs (Section 3.1.30) commands are used to write or read the redefined GPIO. GPIO(08) has a fixed function, and the functions of GPIO(03:00) can only be specified via the GUI.

NOTE: The OEM ensures that signal conflicts do not arise when switching GPIO signal directions (for example, external signal driving a GPIO that is configured as an output).

3.1.26 Read GPIO[19:00] Control (Offset = 32h)

This command is used to read back the control state for GPIO(19:09) and GPIO(07:04) of the display module.

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

Table 3-42. Byte 1 Read GPIO[19:00] Control (32h) Register Field Descriptions

Bit	Type	Description
7-6	R	GPIO (12) 0h = User Defined 1h = Input 2h = Output (Standard) 3h = Output (Open Drain)
5-4	R	GPIO (11) 0h = User Defined 1h = Input 2h = Output (Standard) 3h = Output (Open Drain)
3-2	R	GPIO (10) 0h = User Defined 1h = Input 2h = Output (Standard) 3h = Output (Open Drain)
1-0	R	GPIO (09) 0h = User Defined 1h = Input 2h = Output (Standard) 3h = Output (Open Drain)

Table 3-43. Byte 2 Read GPIO[19:00] Control (32h) Register Field Descriptions

Bit	Type	Description
7-6	R	GPIO (16) 0h = User Defined 1h = Input 2h = Output (Standard) 3h = Output (Open Drain)
5-4	R	GPIO (15) 0h = User Defined 1h = Input 2h = Output (Standard) 3h = Output (Open Drain)
3-2	R	GPIO (14) 0h = User Defined 1h = Input 2h = Output (Standard) 3h = Output (Open Drain)
1-0	R	GPIO (13) 0h = User Defined 1h = Input 2h = Output (Standard) 3h = Output (Open Drain)

Table 3-44. Byte 3 Read GPIO[19:00] Control (32h) Register Field Descriptions

Bit	Type	Description
7-6	R	Reserved
5-4	R	GPIO (19) 0h = User Defined 1h = Input 2h = Output (Standard) 3h = Output (Open Drain)
3-2	R	GPIO (18) 0h = User Defined 1h = Input 2h = Output (Standard) 3h = Output (Open Drain)
1-0	R	GPIO (17) 0h = User Defined 1h = Input 2h = Output (Standard) 3h = Output (Open Drain)

Table 3-45. Byte 4 Read GPIO[19:00] Control (32h) Register Field Descriptions

Bit	Type	Description
7-6	R	GPIO (07) 0h = User Defined 1h = Input 2h = Output (Standard) 3h = Output (Open Drain)
5-4	R	GPIO (06) 0h = User Defined 1h = Input 2h = Output (Standard) 3h = Output (Open Drain)
3-2	R	GPIO (05) 0h = User Defined 1h = Input 2h = Output (Standard) 3h = Output (Open Drain)
1-0	R	GPIO (04) 0h = User Defined 1h = Input 2h = Output (Standard) 3h = Output (Open Drain)

3.1.27 Write GPIO[19:00] Outputs (Offset = 33h) [Reset = User specified]

This command is used to specify how GPIO(19:09) and GPIO(07:00) of the display module are to be used.

Table 3-46. Write Parameters

Parameter Bytes	Description
Byte 1	GPIO Mask (See Below)
Byte 2	GPIO Mask (See Below)
Byte 3	GPIO Mask (See Below)
Byte 4	GPIO Mask (See Below)
Byte 5	GPIO Mask (See Below)
Byte 6	GPIO Mask (See Below)

MSB	Byte 1 - 6						LSB
b7	b6	b5	b4	b3	b2	b1	b0

Table 3-47. Byte 1 Write GPIO[19:00] Outputs (33h) Register Field Descriptions

Bit	Type	Description
7	W	GPIO (7) 0h = Note Selected 1h = Selected
6	W	GPIO (6) 0h = Note Selected 1h = Selected
5	W	GPIO (5) 0h = Note Selected 1h = Selected
4	W	GPIO (4) 0h = Note Selected 1h = Selected
3	W	GPIO (3) 0h = Note Selected 1h = Selected
2	W	GPIO (2) 0h = Note Selected 1h = Selected
1	W	GPIO (1) 0h = Note Selected 1h = Selected
0	W	GPIO (0) 0h = Note Selected 1h = Selected

Table 3-48. Byte 2 Write GPIO[19:00] Outputs (33h) Register Field Descriptions

Bit	Type	Description
7	W	GPIO (16) 0h = Note Selected 1h = Selected
6	W	GPIO (15) 0h = Note Selected 1h = Selected
5	W	GPIO (14) 0h = Note Selected 1h = Selected
4	W	GPIO (13) 0h = Note Selected 1h = Selected
3	W	GPIO (12) 0h = Note Selected 1h = Selected
2	W	GPIO (11) 0h = Note Selected 1h = Selected
1	W	GPIO (10) 0h = Note Selected 1h = Selected
0	W	GPIO (9) 0h = Note Selected 1h = Selected

Table 3-49. Byte 3 Write GPIO[19:00] Outputs (33h) Register Field Descriptions

Bit	Type	Description
7-3	R	Reserved
2	W	GPIO (19) 0h = Note Selected 1h = Selected
1	W	GPIO (18) 0h = Note Selected 1h = Selected
0	W	GPIO (17) 0h = Note Selected 1h = Selected

Table 3-50. Byte 4 Write GPIO[19:00] Outputs (33h) Register Field Descriptions

Bit	Type	Description
7	W	GPIO (7) 0h = Note Selected 1h = Selected
6	W	GPIO (6) 0h = Note Selected 1h = Selected
5	W	GPIO (5) 0h = Note Selected 1h = Selected
4	W	GPIO (4) 0h = Note Selected 1h = Selected
3	W	GPIO (3) 0h = Note Selected 1h = Selected
2	W	GPIO (2) 0h = Note Selected 1h = Selected
1	W	GPIO (1) 0h = Note Selected 1h = Selected
0	W	GPIO (0) 0h = Note Selected 1h = Selected

Table 3-51. Byte 5 Write GPIO[19:00] Outputs (33h) Register Field Descriptions

Bit	Type	Description
7	W	GPIO (16) 0h = Note Selected 1h = Selected
6	W	GPIO (15) 0h = Note Selected 1h = Selected
5	W	GPIO (14) 0h = Note Selected 1h = Selected
4	W	GPIO (13) 0h = Note Selected 1h = Selected
3	W	GPIO (12) 0h = Note Selected 1h = Selected
2	W	GPIO (11) 0h = Note Selected 1h = Selected
1	W	GPIO (10) 0h = Note Selected 1h = Selected
0	W	GPIO (9) 0h = Note Selected 1h = Selected

Table 3-52. Byte 6 Write GPIO[19:00] Outputs (33h) Register Field Descriptions

Bit	Type	Description
7-3	R	Reserved
2	W	GPIO (19) 0h = Note Selected 1h = Selected
1	W	GPIO (18) 0h = Note Selected 1h = Selected
0	W	GPIO (17) 0h = Note Selected 1h = Selected

NOTE: GPIO(19:09) and GPIO(07:04) can function as defined by the OEM in the GUI, or their function can be redefined on-the-fly using the Write GPIO[19:00] Control command (Section 3.1.25).

GPIO(08) is not re-definable on-the-fly, and is not available for use as an OEM GPIO.

GPIO(03:00) can only function as defined by the OEM. One of the choices in the GUI is to define one or more of these GPIO to be OEM GPIO Outputs.

NOTE: When one or more of GPIO(19:09) and GPIO(07:04) are defined as output signals (using the Write GPIO[19:00] Control command), or when one or more of GPIO(03:00) are defined as output signals by the OEM in the GUI, this command is used to specify the values of those output signals. The software retains all of these values for later application if required, and for read back using the Read GPIO[19:00] Outputs command (Section 3.1.28).

This command has no effect on GPIO(19:09) and GPIO(07:04) that are not defined as output signals by the Write GPIO[19:00] Control command, although any values entered for these GPIO are retained and used when these GPIO are later specified to be outputs.

This command has no effect on GPIO(03:00) that are not defined as output signals by the GUI. Even so, any values entered for these GPIO are retained for read back (although they are not applied to the GPIO).

NOTE: In order to set the value of a GPIO, the GPIO must be selected using bytes 1 to 3 of this command, with the appropriate value specified using bytes 3 to 6.

3.1.28 Read GPIO[19:00] Outputs (Offset = 34h)

This command is used to read the output values for GPIO(19:09) and GPIO(07:00) of the display module.

Table 3-53. Read Parameters

Parameter Bytes	Description
Byte 1	GPIO Value (See Below)
Byte 2	GPIO Value (See Below)
Byte 3	GPIO Value (See Below)

MSB	Byte 1 - 3						LSB
b7	b6	b5	b4	b3	b2	b1	b0

Table 3-54. Byte 1 Read GPIO[19:00] Outputs (34h) Register Field Descriptions

Bit	Type	Description
7	R	GPIO (7)
6	R	GPIO (6)
5	R	GPIO (5)
4	R	GPIO (4)
3	R	GPIO (3)
2	R	GPIO (2)
1	R	GPIO (1)
0	R	GPIO (0)

Table 3-55. Byte 2 Read GPIO[19:00] Outputs (34h) Register Field Descriptions

Bit	Type	Description
7	R	GPIO (16)
6	R	GPIO (15)
5	R	GPIO (14)
4	R	GPIO (13)
3	R	GPIO (12)
2	R	GPIO (11)
1	R	GPIO (10)
0	R	GPIO (9)

Table 3-56. Byte 3 Read GPIO[19:00] Outputs (34h) Register Field Descriptions

Bit	Type	Description
7-3	R	Reserved
2	R	GPIO (19)
1	R	GPIO (18)
0	R	GPIO (17)

NOTE: This command returns the values specified by the Write GPIO[19:00] Outputs command (Section 3.1.27). Any GPIO not having a value specified by the Write GPIO[19:00] Outputs command returns a value of zero. The value returned may or may not be the value at the GPIO. See the Write GPIO[19:00] Control (Section 3.1.25) and Read GPIO[19:00] Inputs (Section 3.1.30) command for further information.

NOTE: When one or more of GPIO(19:09) are defined as input signals (using the Write GPIO[19:00] Control command), or one or more of GPIO(07:00) are defined as input signals by the OEM in the GUI, this command is used to read back the current value of those specific GPIO. Each time a read request is made, the ARM software de-bounces, samples, and returns the current value on these GPIO. This command returns a zero for any GPIO(19:09) and GPIO(07:00) not defined as an input signal.

3.1.29 Write Splash Screen Execute (Offset = 35h)

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

This command has no write parameters.

NOTE: This command is used in conjunction with the Write Input Source Select (Section 3.1.1) and the Write Splash Screen Select (Section 3.1.5) commands. It is used to start the process of retrieving a splash screen from Flash for display.

The Splash Screen is a unique source because it is read from Flash and sent down the processing path of the controller one time, to be stored in memory for display at the end of the processing path. As such, all image processing settings (for example, image crop, image orientation, display size, splash screen select, splash screen as input source, and so forth) should be set appropriately 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 displays. Thus, the splash screen retrieval process repeats each time this command is received. See also the Write Image Freeze command (Section 3.1.12) for information on hiding on-screen artifacts when selecting and retrieving a splash image.

NOTE: It is important that the user review the notes for the Write Input Source Select command in Section 3.1.1 to understand the concept of source associated commands. This concept determines when source associated commands are executed by the system. Note that this command is a source associated command; however, this command is special in that there is no maintained state or history. Thus, this command has no “settings” to be stored or reused by the system.

NOTE: When this command is processed, the system automatically sets up the system color processing based on the splash header information prior to sending the splash image down the data path.

3.1.30 Read GPIO[19:00] Inputs (Offset = 36h)

This command is used to read the output values for GPIO(19:09) and GPIO(07:00) of the display module.

MSB	Byte 1 - 3						LSB
b7	b6	b5	b4	b3	b2	b1	b0

Table 3-57. Byte 1 Read GPIO[19:00] Inputs (36h) Register Field Descriptions

Bit	Type	Description
7	R	GPIO (7)
6	R	GPIO (6)
5	R	GPIO (5)
4	R	GPIO (4)
3	R	GPIO (3)
2	R	GPIO (2)
1	R	GPIO (1)
0	R	GPIO (0)

Table 3-58. Byte 2 Read GPIO[19:00] Inputs (36h) Register Field Descriptions

Bit	Type	Description
7	R	GPIO (16)
6	R	GPIO (15)
5	R	GPIO (14)
4	R	GPIO (13)
3	R	GPIO (12)
2	R	GPIO (11)
1	R	GPIO (10)
0	R	GPIO (9)

Table 3-59. Byte 3 Read GPIO[19:00] Inputs (36h) Register Field Descriptions

Bit	Type	Description
7-3	R	Reserved
2	R	GPIO (19)
1	R	GPIO (18)
0	R	GPIO (17)

NOTE: GPIO(19:09) can function as defined by the OEM, or their function can be redefined on-the-fly using the Write GPIO[19:00] Control command (Section 3.1.25). GPIO(08) is not re-definable on-the-fly, and is not available for use as an OEM GPIO. GPIO(07:00) can only function as defined by the OEM. One of the choices in the GUI is to define one or more of these GPIO to be OEM GPIO Inputs.

NOTE: When one or more of GPIO(19:09) are defined as input signals (using the Write GPIO[19:00] Control command), or one or more of GPIO(07:00) are defined as input signals by the OEM, this command is used to read back the current value of those specific GPIO. Each time a read request is made the ARM software de-bounces, samples, and returns the current value on these GPIO. This command returns zero for any GPIO(19:09) and GPIO(07:00) not defined as an input signal.

3.1.31 Write FPD Link Data Mode (Offset = 4Bh)

This command is used to configure the FPD link display bit rate and Map mode.

Table 3-60. FPD Link Data Parameters

Parameter Bytes	Description
Byte 0	Bit rate in Mbps (lsb)
Byte 1	Bit rate in Mbps (lsb)
Byte 2	See below

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

Table 3-61. Write FPD Link Data Mode (4Bh) Register Field Descriptions

Bit	Type	Description
7-4	R	Reserved
3-0	W	Pixel Map Mode 1h = Mode #1 2h = Mode #2 3h = Mode #3 4h = Mode #4 5h = Mode #5 6h = Mode #6 7h = Mode #7 8h = Mode #8

Input video data is encoded into the FPD data buses as indicated in the following tables.

Table 3-62. FPD LVDS Data Bus Encoding

	Mode 1	Mode 2	Mode 3	Mode 4
FPD Bus A - Data_A Channel				
FPD_A_DATA_A_6	Green_4	Green_2	Green_0	Green_4
FPD_A_DATA_A_5	Red_9	Red_7	Red_5	Red_9
FPD_A_DATA_A_4	Red_8	Red_6	Red_4	Red_8
FPD_A_DATA_A_3	Red_7	Red_5	Red_3	Red_7
FPD_A_DATA_A_2	Red_6	Red_4	Red_2	Red_6
FPD_A_DATA_A_1	Red_5	Red_3	Red_1	Red_5
FPD_A_DATA_A_0	Red_4	Red_2	Red_0	Red_4
FPD Bus A - Data_B Channel				
FPD_A_DATA_B_6	Blue_5	Blue_3	Blue_1	Blue_5
FPD_A_DATA_B_5	Blue_4	Blue_2	Blue_0	Blue_4
FPD_A_DATA_B_4	Green_9	Green_7	Green_5	Green_9
FPD_A_DATA_B_3	Green_8	Green_6	Green_4	Green_8
FPD_A_DATA_B_2	Green_7	Green_5	Green_3	Green_7
FPD_A_DATA_B_1	Green_6	Green_4	Green_2	Green_6
FPD_A_DATA_B_0	Green_5	Green_3	Green_1	Green_5

Table 3-62. FPD LVDS Data Bus Encoding (continued)

	Mode 1	Mode 2	Mode 3	Mode 4
FPD Bus A - Data_C Channel				
FPD_A_DATA_C_6	DEN	DEN	DEN	DEN
FPD_A_DATA_C_5	VSYNC	VSYNC	VSYNC	VSYNC
FPD_A_DATA_C_4	HSYNC	HSYNC	HSYNC	HSYNC
FPD_A_DATA_C_3	Blue_9	Blue_7	Blue_5	Blue_9
FPD_A_DATA_C_2	Blue_8	Blue_6	Blue_4	Blue_8
FPD_A_DATA_C_1	Blue_7	Blue_5	Blue_3	Blue_7
FPD_A_DATA_C_0	Blue_6	Blue_4	Blue_2	Blue_6
FPD Bus A - Data_D Channel				
FPD_A_DATA_D_6	Map to Field	Map to Field	Map to Field	Map to Field
FPD_A_DATA_D_5	Blue_3	Blue_9	Blue_7	not used
FPD_A_DATA_D_4	Blue_2	Blue_8	Blue_6	not used
FPD_A_DATA_D_3	Green_3	Green_9	Green_7	not used
FPD_A_DATA_D_2	Green_2	Green_8	Green_6	not used
FPD_A_DATA_D_1	Red_3	Red_9	Red_7	not used
FPD_A_DATA_D_0	Red_2	Red_8	Red_6	not used
FPD Bus A - Data_E Channel				
FPD_A_DATA_E_6	Map to Field	Map to Field	Map to Field	Map to Field
FPD_A_DATA_E_5	Blue_1	Blue_1	Blue_9	not used
FPD_A_DATA_E_4	Blue_0	Blue_0	Blue_8	not used
FPD_A_DATA_E_3	Green_1	Green_1	Green_9	not used
FPD_A_DATA_E_2	Green_0	Green_0	Green_8	not used
FPD_A_DATA_E_1	Red_1	Red_1	Red_9	not used
FPD_A_DATA_E_0	Red_0	Red_0	Red_8	not used
FPD Bus B				
FPD Bus B is unused in Modes 1 through 6				

3.1.32 Read FPD Pixel Map Mode (Offset = 4Ch)

This command is used to read the FPD link display Pixel Map mode.

Table 3-63. Return Parameters

Parameter Bytes	Description
Byte 0	Bit rate in Mbps (lsb)
Byte 1	Bit rate in Mbps (lsb)
Byte 2	See below

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

Table 3-64. Read FPD Pixel Map Mode (4Ch) Register Field Descriptions

Bit	Type	Description
7-4	R	Reserved
3-0	R	Pixel Map Mode 1h = Mode #1 2h = Mode #2 3h = Mode #3 4h = Mode #4 5h = Mode #5 6h = Mode #6 7h = Mode #7 8h = Mode #8

3.1.33 Write FPD Input Video Chroma Processing Select (Offset = 4Dh)

This command is used to specify Chroma processing select for the YUV422 source input to the FPGA.

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

Table 3-65. Write FPD Input Video Chroma Processing Select (4Dh) Register Field Descriptions

Bit	Type	Description
7-4	R	Reserved
3	W	Chroma Channel Swap 0h = CbCr 1h = CrCb
2-0	R	Reserved

3.1.34 Read FPGA Input Video Chroma Processing Select (Offset = 4Eh)

This command is used to read the Chroma processing select for the YUV422 source input to the FPGA.

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

Table 3-66. Read FPGA Input Video Chroma Processing Select (4Eh) Register Field Descriptions

Bit	Type	Description
7-4	R	Reserved
3	R	Chroma Channel Swap 0h = CbCr 1h = CrCb
2-0	R	Reserved

3.1.35 Write LED Output Control Method (Offset = 50h) [Reset = User specified]

This command is used to specify the method for controlling the LED outputs for the display module.

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

Table 3-67. Write LED Output Control Method (50h) Register Field Descriptions

Bit	Type	Description
7-2	R	Reserved
1-0	W	LED control method 0h = Manual RGB LED currents (disables CAIC algorithm) 1h = CAIC (automatic) RGB LED power (enables CAIC algorithm) 2h = Reserved 3h = Reserved

NOTE: This command selects the method to be 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 3-68.

Table 3-68. Available Commands Based on LED Control Method

LED Control Method	Available Commands
Manual RGB LED current control (CAIC Disabled)	Write RGB LED Enable (52h) Read RGB LED Enable (53h) Write RGB LED Current (54h) Read RGB LED Current (55h) Write RGB LED Max Current (5Ch) Read RGB LED Max Current (5Dh)
CAIC (automatic) RGB LED current control (CAIC Enabled)	Write RGB LED Enable (52h) Read RGB LED Enable (53h) Write RGB LED Current (54h) Read RGB LED Current (55h) Read CAIC LED Max Available Power (57h) Read CAIC LED RGB Current (5Fh)

NOTE: The Manual RGB LED Currents method provides for manual control of the LED currents, and as such, the CAIC algorithm (Section 3.1.75) is disabled.

NOTE: The CAIC (Automatic) RGB LED Current Control method provides automatic control of the LED currents using the CAIC algorithm.

3.1.36 Read LED Output Control Method (Offset = 51h)

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

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

Table 3-69. Read LED Output Control Method (51h) Register Field Descriptions

Bit	Type	Description
7-2	R	Reserved
1-0	W	LED control method 0h = Manual RGB LED currents (disables CAIC algorithm) 1h = CAIC (automatic) RGB LED power (enables CAIC algorithm) 2h = Reserved 3h = Reserved

3.1.37 Write RGB LED Enable (Offset = 52h) [Reset = 07h]

This command enables the LEDs for the display module.

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

Table 3-70. Write RGB LED Enable (52h) Register Field Descriptions

Bit	Type	Description
7-3	R	Reserved
2	W	Blue LED enable 0h = Blue LED disabled 1h = Blue LED enabled
1	W	Green LED enable 0h = Green LED disabled 1h = Green LED enabled
0	W	Red LED enable 0h = Red LED disabled 1h = Red LED enabled

3.1.38 Read RGB LED Enable (Offset = 53h)

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

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

Table 3-71. Read RGB LED Enable (53h) Register Field Descriptions

Bit	Type	Description
7-3	R	Reserved
2	W	Blue LED enable 0h = Blue LED disabled 1h = Blue LED enabled
1	W	Green LED enable 0h = Green LED disabled 1h = Green LED enabled
0	W	Red LED enable 0h = Red LED disabled 1h = Red LED enabled

3.1.39 Write RGB LED Current (Offset = 54h) [Reset = User specified]

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

Table 3-72. 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)

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

NOTE: The parameters specified by this command have a resolution of 10 bits, and are to be as defined by the appropriate Power Management IC (PMIC) specification.

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

When 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 also drop. Command Read CAIC RGB LED Current (5Fh) can be used to read the actual LED currents for the image currently being displayed.
- In the case of an all-white image, the values read by the command Read CAIC RGB LED Current (5Fh) closely matches but may not exactly match those requested using command Write RGB LED Current (54h). For an all-white image command Read CAIC RGB LED Current (5Fh) gives currents within ±4 PMIC device current steps for each LED color relative to those requested by command Write RGB LED Current (54h).
- When command Write RGB LED Current (54h) is used to change the LED currents, the LED current for any color should not be changed by more than ±25% from the nominal current used for that color when the CAIC LUTs were created. Furthermore, no LED current should 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 because in this case the CAIC algorithm requests the CAIC LED maximum available power. The maximum available LED power for CAIC is controlled by the command Write RGB LED Current because this command controls currents for an all-white image. After the currents are adjusted, command Read CAIC LED Maximum Available Power (57h) can be used to see the maximum power in Watts that CAIC derived.

3.1.40 Read RGB LED Current (Offset = 55h)

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

Table 3-73. 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)

NOTE: See *Section 3.1.39* for a detailed description of the return parameters.

NOTE: Unused most significant bits are set to 0.

3.1.41 Read CAIC LED Max Available Power (Offset = 57h)

This command is used to read the maximum LED power allowed for the display module at the LED current settings set by the Write RGB LED Current (54h) command.

Table 3-74. Return Parameters

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

NOTE: The value is specified in Watts * 100 (Example: 25.75 W = A0Fh).

NOTE: 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 equation is:

Maximum Available Power = R duty cycle × R LED current × R LED voltage + G duty cycle × G LED current × G LED voltage + B duty cycle × B LED current × B LED voltage (1)

For example: (0.30 × 0.49 A × 2.0 V) + (0.50 × 0.39 A × 3.1 V) + (0.20 × 0.39 A × 3.1 V) = (0.30 × 0.980 W) + (0.50 × 1.209 W) + (0.20 × 1.209 W) = 1.140 W

3.1.42 Write RGB LED Max Current (Offset = 5Ch) [Reset = User specified]

This command is used to specify the maximum LED current allowed for each LED in the display module when CAIC is disabled.

Table 3-75. 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)

NOTE: The parameters specified by this command have a resolution of 10 bits, and are to be as defined by the appropriate PMIC specification.

NOTE: This command sets the maximum LED currents that can be used when the CAIC algorithm is disabled. When the CAIC algorithm is enabled, the maximum LED currents are determined by the CAIC algorithm LUTs stored in Flash.

NOTE: For further information about LED current and the CAIC algorithm, see the notes for the Write RGB LED Current (54h) command.

NOTE: Unused most significant bits should be set to ‘0’.

3.1.43 Read RGB LED Max Current (Offset = 5Dh)

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

Table 3-76. 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)

NOTE: See the Write RGB LED Current Control command for a detailed description of the return parameters.

NOTE: Unused most significant bits are set to ‘0’.

3.1.44 Read CAIC RGB LED Current (Offset = 5Fh)

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

Table 3-77. 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)

NOTE: The parameters returned by this command have a resolution of 10 bits, and are defined by the appropriate PMIC 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 being 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 the Write RGB LED current command and the total LED power may also drop. The actual LED currents for the image currently being displayed can be read using this command (the Read CAIC RGB LED Current (5Fh) command)
- In the case of an all white image, the values returned by this command closely matches, 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 PMIC device current steps for each LED color relative to those specified with the Write RGB LED Current command.

NOTE: Use of this command is only appropriate when the LED Output Control Method is set to CAIC (Automatic) RGB LED Current Control.

NOTE: Unused most significant bits are set to ‘0’.

3.1.45 Write XPR FPGA Input Image Size (Offset = 60h)

This command is used to specify the active data size of the external input image that goes to the XPR FPGA.

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)

3.1.46 Read XPR FPGA Input Image Size (Offset = 61h)

This command is used to read specified data size of the external input image to the display module.

3.1.47 Write XPR FPGA Source Select (Offset = 62h)

This command is used to specify XPR FPGA input source.

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

Table 3-78. Write XPR FPGA Source Select (62h) Register Field Descriptions

Bit	Type	Description
7-2	R	Reserved
1-0	W	Input source 0h = TPG 1h = External Parallel Video 2h = FPD-Link or LVDS 3h = Reserved

3.1.48 Read XPR FPGA Source Select (Offset = 63h)

This command is used to read selected XPR FPGA input source.

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

Table 3-79. Read XPR FPGA Source Select (63h) Register Field Descriptions

Bit	Type	Description
7-2	R	Reserved
1-0	W	Input source 0h = TPG 1h = External Parallel Video 2h = FPD-Link or LVDS 3h = Reserved

3.1.49 Read XPR FPGA Version (Offset = 64h)

This command is used to read the XPR FPGA software and bitstream version.

Parameter Bytes	Description
Byte 1	FPGA ARM software version - Minor (LSByte)
Byte 2	FPGA ARM software version - Major (MSByte)
Byte 3:6	• b(31:28) = FPGA Firmware Version – Build Level • b(27:20) = FPGA Firmware Version – Minor • b(19:12) = FPGA Firmware Version – Major • b(11:0) = FPGA Firmware Version – Build Number

3.1.50 Write XPR FPGA Test Pattern Select (Offset = 67h)

This command is used to specify an internal test pattern from XPR FPGA for display on the display module.

Parameter Bytes	Description
Byte 1	TPG pattern select (LSByte)
Byte 2	Size of pattern (MSByte)

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

Table 3-80. Write XPR FPGA Test Pattern Select (67h) Register Field Descriptions

Bit	Type	Description
7	W	Test Pattern Boarder 0h = Disabled (default) 1h = Enabled
6-4	W	Color 0h = Black 1h = Blue 2h = Red 3h = Magenta 4h = Green 5h = Cyan 6h = Yellow 7h = White
3-0	W	Pattern Select 0h = Solid Field 1h = Grids 2h = Horizontal Ramp 4h = Checkerboard 5h = Horizontal Lines 6h = Vertical Lines 7h = Diagonal Lines 8h = Actuator Calibration Pattern 9h = 3D Test Pattern Ah - Fh = Reserved

Byte 2 Size of the pattern: for the patterns where size is not required (e.g solid field), this should be set to 0.

NOTE: Test Pattern Vertical Lines is not supporting border enable.

3.1.51 Read XPR FPGA Test Pattern Select (Offset = 68h)

This command is used to an internal test pattern from XPR FPGA.

Parameter Bytes	Description
Byte 1	TPG pattern select (LSByte)
Byte 2	Size of pattern (MSByte)

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

Table 3-81. Read XPR FPGA Test Pattern Select (68h) Register Field Descriptions

Bit	Type	Description
7	R	Test Pattern Boarder 0h = Disabled (default) 1h = Enabled
6-4	R	Color 0h = Black 1h = Blue 2h = Red 3h = Magenta 4h = Green 5h = Cyan 6h = Yellow 7h = White
3-0	R	Pattern Select 0h = Solid Field 1h = Grids 2h = Horizontal Ramp 4h = Checkerboard 5h = Horizontal Lines 6h = Vertical Lines 7h = Diagonal Lines 8h = Actuator Calibration Pattern 9h = 3D Test Pattern Ah - Fh = Reserved

Byte 2: Size of pattern.

NOTE: Test Pattern Vertical Lines is not supporting border enable.

3.1.52 Write XPR FPGA Parallel Video Control (Offset = 6Bh)

This command is used to configure polarity of syncs and sampling edge of the pixel clock in XPR FPGA.

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

Table 3-82. Write XPR FPGA Parallel Video Control (6Bh) Register Field Descriptions

Bit	Type	Description
7-4	R	Reserved
3	W	VSync Polarity 0h = Active Low 1h = Active High
2	W	HSync Polarity 0h = Active Low 1h = Active High
1	W	IValid Polarity 0h = Active Low 1h = Active High
0	W	Pixel Clock Sampling Edge 0h = Falling Edge 1h = Rising Edge

3.1.53 Read XPR FPGA Parallel Video Control (Offset = 6Ch)

This command is used to read XPR FPGA video format.

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

Table 3-83. Read XPR FPGA Parallel Video Control (6Ch) Register Field Descriptions

Bit	Type	Description
7-4	R	Reserved
3	R	VSync Polarity 0h = Active Low 1h = Active High
2	R	HSync Polarity 0h = Active Low 1h = Active High
1	R	IValid Polarity 0h = Active Low 1h = Active High
0	R	Pixel Clock Sampling Edge 0h = Falling Edge 1h = Rising Edge

3.1.54 Write XPR FPGA Video Format Select (Offset = 6Dh)

This command is used to specify XPR FPGA video format.

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

Table 3-84. Write XPR FPGA Video Format Select (6Dh) Register Field Descriptions

Bit	Type	Description
7-2	R	Reserved
1-0	W	Input source format • 0h = RGB888 • 1h = RGB565 • 2h = RGB666 • 3h = YCbCr422 • 4h = YCbCr444 • 5h = YCbCr565 • 6h = YCbCr666

3.1.55 Read XPR FPGA Video Format Select (Offset = 6Eh)

This command is used to read XPR FPGA video format.

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

Table 3-85. Read XPR FPGA Video Format Select (6Eh) Register Field Descriptions

Bit	Type	Description
7-2	R	Reserved
1-0	W	Input source format • 0h = RGB888 • 1h = RGB565 • 2h = RGB666 • 3h = YCbCr422 • 4h = YCbCr444 • 5h = YCbCr565 • 6h = YCbCr666

3.1.56 Read XPR FPGA Status (Offset = 6Fh)

This command is used to read XPR FPGA status.

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

Table 3-86. Read XPR FPGA Status (6Fh) Register Field Descriptions

Bit	Type	Description
7-2	R	Reserved
1	R	Display Mode 0h = Non-XPR Mode 1h = XPR Mode
0	R	FPGA Keying Status 0h = Failed 1h = Passed

3.1.57 Write Actuator Latency (Offset = 70h)

This command is used to specify the Actuator Latency. This command is required for Actuator calibration. The reset value is the latency value in the sequence header.

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

Table 3-87. Write Actuator Latency (70h) Register Field Descriptions

Bit	Type	Description
7-1	R	Reserved
0	W	Manual Latency Enable 0h = Manual Latency disabled – Latency value stored in the sequence header is used. 1h = Latency value is used provided in Byte 1-3.

3.1.58 Read Actuator Latency (Offset = 71h)

This command is used to read the Actuator Latency. This command is required for Actuator calibration. The reset value is the latency value in the sequence header.

Parameter Bytes	Description
Byte 1	Latency (LSByte)
Byte 2	Latency
Byte 3	Latency (MSByte)
Byte 4	See below

Table 3-88. Read Actuator Latency (71h) Register Field Descriptions

Bit	Type	Description
7-1	R	Reserved
0	R	Manual Latency Enable 0h = Manual Latency disabled – Latency value stored in the sequence header is used. 1h = Latency value are used provided in Byte 1-3.

3.1.59 Write Actuator Gain (Offset = 72h)

This command is used to specify the Actuator Gain parameter.

Parameter Bytes	Description
Byte 1	Actuator Gain

NOTE: Value is presented in fixed point format.

1 = 0.007813

Valid range (0 to 1.9921875)

3.1.60 Read Actuator Gain (Offset = 73h)

This command is used to read the Actuator Gain parameter.

NOTE: Value is presented in fixed point format.

1 = 0.007813

Valid range (0 to 1.9921875)

3.1.61 Write Segment Length (Offset = 74h)

This command is used to specify the Actuator Segment Length parameter.

Parameter Bytes	Description
Byte 1	Segment Length (LSByte)
Byte 2	Segment Length (MSByte)

NOTE: Valid segment length is 2 to 65535.

3.1.62 Read Segment Length (Offset = 75h)

This command is used to read the Actuator Segment Length parameter.

Parameter Bytes	Description
Byte 1	Segment Length (LSByte)
Byte 2	Segment Length (MSByte)

NOTE: Valid segment length is 2 to 65535.

3.1.63 Write Manual Actuator Sync Delay (Offset = 76h)

This command is used to specify the Actuator Sync Delay parameter.

The reset value is pre-configured in the sequence header.

Parameter Bytes	Description
Byte 1	Actuator Sync Delay (LSByte)
Byte 2	Actuator Sync Delay
Byte 3	Actuator Sync Delay (MSByte)
Byte 4	Manual / Auto Actuator Sync Delay enable

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

Table 3-89. Write Manual Actuator Sync Delay (76h) Register Field Descriptions

Bit	Type	Description
7-2	R	Reserved
1	W	Auto-scaling enable. Applicable only when manual Actuator Sync Delay override mode is enabled, b(0)=1. 0h = No scaling is performed. Actuator Sync delay is applied as defined in Byte 1-3 1h = Auto scaling is performed with frame rate change.
0	W	Manual Actuator Sync Delay override enable 0h = Actuator Sync Delay defined in Byte 1 to 3 will not be applied only when this bit is disabled. Instead, the Actuator Sync Delay defined in the flash as part of the sequence data is applied. 1h = Actuator Sync Delay defined in Byte 1 to 3 are applied only when this bit is enabled.

NOTE: This command is executed in conjunction with Write Actuator Latency command. Latency corrections are always made to the Actuator delay before writing to the hardware register. In case Latency correction is not required, then Latency should be set to 0.

3.1.64 Read Manual Actuator Sync Delay (Offset = 77h)

This command is used to read Manual Actuator Sync Delay parameter.

The reset value is pre-configured in the sequence header.

Parameter Bytes	Description
Byte 1	Actuator Sync Delay (LSByte)
Byte 2	Actuator Sync Delay
Byte 3	Actuator Sync Delay (MSByte)
Byte 4	Manual / Auto Actuator Sync Delay enable

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

Table 3-90. Read Manual Actuator Sync Delay (77h) Register Field Descriptions

Bit	Type	Description
7-2	R	Reserved
1	R	Auto-scaling enable. Applicable only when manual Actuator Sync Delay override mode is enabled, b(0)=1. 0h = No scaling is performed. Actuator Sync delay is applied as defined in Byte 1-3 1h = Auto scaling is performed with frame rate change.
0	R	Manual Actuator Sync Delay override enable 0h = Actuator Sync Delay defined in Byte 1 to 3 will not be applied only when this bit is disabled. Instead, the Actuator Sync Delay defined in the flash as part of the sequence data is applied. 1h = Actuator Sync Delay defined in Byte 1 to 3 are applied only when this bit is enabled.

NOTE: This command is executed in conjunction with Write Actuator Latency command. Latency corrections are always made to the Actuator delay before writing to the hardware register. In case Latency correction is not required, then Latency should be set to 0.

3.1.65 Write Manual Actuator Offset (Offset = 78h)

This command is used to specify the Manual Actuator Offset parameter.

Parameter Bytes	Description
Byte 1	Manual Actuator Offset (LSByte)
Byte 2	Manual Actuator Offset (MSByte)

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

Table 3-91. Write Manual Actuator Offset (78h) Register Field Descriptions

Bit	Type	Description
7-4	R	Reserved
3-0	W	Manual Actuator Offset

NOTE: This Actuator Manual Offset is presented in fixed point format (1 = 00.0078130)

Valid values of Actuator Manual Offset is 0 – 31.9921875.

3.1.66 Read Manual Actuator Offset (Offset = 79h)

This command is used to read the Manual Actuator Offset parameter.

Parameter Bytes	Description
Byte 1	Manual Actuator Offset (LSByte)
Byte 2	Manual Actuator Offset (MSByte)

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

Table 3-92. Read Manual Actuator Offset (79h) Register Field Descriptions

Bit	Type	Description
7-4	R	Reserved
3-0	R	Manual Actuator Offset

NOTE: This Actuator Manual Offset is presented in fixed point format (1 = 00.0078130)

Valid values of Actuator Manual Offset is 0 – 31.9921875.

3.1.67 Write Actuator Fixed Output (Offset = 7Ah)

This command is used to specify the Actuator Fixed Output parameter.

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

Table 3-93. Write Actuator Fixed Output (7Ah) Register Field Descriptions

Bit	Type	Description
7-1	R	Reserved
0	W	Enable Fixed Output 0h = Disable Fixed Output (means switch to auto output mode) 1h = Enable Fixed Output (Fixed output state is defined in flash)

3.1.68 Read Actuator Fixed Output (Offset = 7Bh)

This command is used to read the Actuator Fixed Output parameter.

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

Table 3-94. Read Actuator Fixed Output (7Bh) Register Field Descriptions

Bit	Type	Description
7-1	R	Reserved
0	W	Enable Fixed Output 0h = Disable Fixed Output (means switch to auto output mode) 1h = Enable Fixed Output (Fixed output state is defined in flash)

3.1.69 Write Actuator Direction (Offset = 7Ch)

This command is used to specify the Actuator Waveform Direction parameter.

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

Table 3-95. Write Actuator Direction (7Ch) Register Field Descriptions

Bit	Type	Description
7-1	R	Reserved
0	W	Actuator Waveform Direction 0h = Normal 1h = Reverse

3.1.70 Read Actuator Direction (Offset = 7Dh)

This command is used to read the Actuator Waveform Direction parameter.

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

Table 3-96. Read Actuator Direction (7Dh) Register Field Descriptions

Bit	Type	Description
7-1	R	Reserved
0	R	Actuator Waveform Direction 0h = Normal 1h = Reverse

3.1.71 Write Actuator Enable (Offset = 7Eh)

This command is used to specify the Actuator Enable parameter.

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

Table 3-97. Write Actuator Enable (7Eh) Register Field Descriptions

Bit	Type	Description
7-1	R	Reserved
0	W	Actuator Waveform Direction 0h = Disabled 1h = Enabled

3.1.72 Read Enable (Offset = 7Fh)

This command is used to read the Actuator Enable parameter.

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

Table 3-98. Read Enable (7Fh) Register Field Descriptions

Bit	Type	Description
7-1	R	Reserved
0	R	Actuator Waveform Direction 0h = Disabled 1h = Enabled

3.1.73 Write Local Area Brightness Boost Control (Offset = 80h) [Reset = 1h]

This command controls the local area brightness boost image processing functionality for the display module.

Table 3-99. Write Parameters

Parameter Bytes	Description
Byte 1	See below
Byte 2	LABB strength setting

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

Table 3-100. Write Local Area Brightness Boost Control (80h) Register Field Descriptions

Bit	Type	Description
7-4	W	Sharpness strength
3-2	R	Reserved
1-0	W	LABB control 0h = Disabled 1h = Enabled: Manual strength control (no light sensor) 2h = Enabled: Automatic strength control (uses light sensor) 3h = Reserved

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

NOTE: For automatic strength control, a light sensor are used to automatically adjust the applied image strength based on the measured black level of the screen, or the ambient lighting level of the room.

NOTE: For LABB Strength, 0 indicates no boost applied, and 255 indicates the maximum boost that is considered viable in a product. The strength does not directly indicate the gain bacause the gain varies depending on image content.

NOTE: Sharpness strength can range from 0 to 15, with 0 indicating sharpness disabled, and 15 indicating the maximum sharpness. The LABB function must be enabled (either Manual or Automatic) to make use of Sharpness.

NOTE: LABB is supported in TPG, Splash, External Input mode, but auto-disabled in curtain mode.

3.1.74 Read Local Area Brightness Boost Control (Offset = 81h)

This command reads the state of the local area brightness boost image processing functionality for the display module.

Figure 3-4. Return Parameters

Parameter Bytes	Description
Byte 1	See below
Byte 2	LABB strength setting
Byte 3	LABB gain value

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

Table 3-101. Read Local Area Brightness Boost Control (81h) Register Field Descriptions

Bit	Type	Description
7-4	R	Sharpness strength
3-2	R	Reserved
1-0	R	LABB control 0h = Disabled 1h = Enabled: Manual strength control (no light sensor) 2h = Enabled: Automatic strength control (uses light sensor) 3h = Reserved

Figure 3-5 shows the bit order and weighting for the LABB gain value, which ranges from 1 to 8 (the controller software should limit the lower value to 1).

Figure 3-5. Bit Weight Definition for LABB Gain Value

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

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

```
LABB_gain = add_8lsb(APL) / pre_LABB_APL      (//add 8 LSBs (u8.0 / u8.0 = u8.8 / u8.0 = u8.8)
```

3.1.75 Write CAIC Image Processing Control (Offset = 84h) [Reset = User specified]

This command controls the CAIC functionality for the display module.

Table 3-102. Write Parameters

Parameter Bytes	Description
Byte 1	See below
Byte 2	CAIC maximum lumens gain
Byte 3	CAIC clipping threshold

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

Table 3-103. Write CAIC Image Processing Control (84h) Register Field Descriptions

Bit	Type	Description
7	W	CAIC gain display enable 0h = Disabled 1h = Enabled
6	W	CAIC gain display scale 0h = 100% = 1024 pixels 1h = 100% = 512 pixels
5-3	R	Reserved
2-0	W	CAIC WPC control 0h = White point correction disabled 1h = White point correction enabled

NOTE: The CAIC algorithm (Content Adaptive Illumination Control) provides adaptive control of the LED currents and the digital gain applied to the image. In addition, when an external sensor is provided by the OEM (and when WPC is enabled by this command), the algorithm will provide automatic white point correction.

NOTE: The CAIC algorithm is enabled or disabled based on the method of LED current control selected by the OEM 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.

NOTE: The CAIC Gain Display provides a visual presentation of the instantaneous gain provided by the CAIC algorithm. This is typically used as a debug tool and to show the performance of the algorithm. It should never be used for normal operation. The display is made up of 5 bars, where the bottom three bars (green, red, and blue) show the respective CAIC gain for each color. The top two bars are for TI debug use only. For SW, the CAIC Gain Display Enable is controlled by CAIC_DEBUG_MODE (2:0), where Disabled = 0h, and Enabled = 3h. The Display Scale is set using CAIC_DEBUG_MODE(3).

NOTE: Figure 3-6 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 command parameter value – communication status) and the command will not be executed.

Figure 3-6. Bit Weight Definition for the CAIC Maximum Gain Value

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

The CAIC Maximum Lumens Gain parameter sets the maximum lumens gain that a pixel can have as a result of both digital gain and increasing LED currents. It also serves to bias the CAIC algorithm towards either Constant Power (variable brightness) or Constant Lumens (variable power). Some examples are listed below:

- Maximum Gain value = 1.0h = This biases CAIC 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 = 4.0h = This biases CAIC performance to Constant Power. In this case, power is held constant for most images, while the lumens are gained up. It should be noted that for the small percent of images where the gain would exceed 4.0, lumens will stop increasing and the power is reduced instead.

.

NOTE: Figure 3-7 shows the bit order and weighting for the CAIC Clipping Threshold value, which has a valid range from 0.0% to 2.0%. Values outside of this range are considered an error (invalid command parameter value – communication status) and the command will not be executed.

Figure 3-7. Bit Weight Definition for the CAIC Clipping Threshold Value

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

NOTE: The CAIC Clipping Threshold parameter sets the percentage of pixels that can be clipped by the CAIC algorithm over the full frame of active data due to the digital gain being applied by the CAIC algorithm.

NOTE: The below 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 command parameter value – communication status) and the command will not be executed.

Figure 3-8. 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^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}	2^{-9}	2^{-10}

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

Table 3-104. LABB and CAIC Modes

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

3.1.76 Read CAIC Image Processing Control (Offset = 85h)

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

Table 3-105. Return Parameters

Parameter Bytes	Description
Byte 1	See below
Byte 2	CAIC maximum lumens gain
Byte 3	CAIC clipping threshold

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

Table 3-106. Read CAIC Image Processing Control (85h) Register Field Descriptions

Bit	Type	Description
7	R	CAIC gain display enable 0h = Disabled 1h = Enabled
6	R	CAIC gain display scale 0h = 100% = 1024 pixels 1h = 100% = 512 pixels
5-3	R	Reserved
2-0	R	CAIC WPC control 0h = White point correction disabled 1h = White point correction enabled

Information on these parameters can be found in *Write CAIC Image Processing Control* Section 3.1.75.

3.1.77 Write Color Coordinate Adjustment Control (Offset = 86h) [Reset = 1h]

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

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

Table 3-107. Write Color Coordinate Adjustment Control (86h) Register Field Descriptions

Bit	Type	Description
7-1	R	Reserved
0	R	CCA enable 0h = Disabled 1h = Enabled

This command is for TI debug purposes only. This function should remain enabled during normal operation.

When CCA is disabled, use an identity matrix.

3.1.78 Read Color Coordinate Adjustment Control (Offset = 87h)

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

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

Table 3-108. Read Color Coordinate Adjustment Control (87h) Register Field Descriptions

Bit	Type	Description
7-1	R	Reserved
0	R	CCA enable 0h = Disabled 1h = Enabled

3.1.79 Write Keystone Correction Control (Offset = 88h) [Reset = 0h]

This command controls the keystone correction image processing functionality for the display module.

Table 3-109. Write Parameters

Parameter Bytes	Description
Byte 1	See below
Byte 2	Optical throw ratio (LSByte)
Byte 3	Optical throw ratio (MSByte)
Byte 4	Optical DMD offset (LSByte)
Byte 5	Optical DMD offset (MSByte)

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

Table 3-110. Write Keystone Correction Control (88h) Register Field Descriptions

Bit	Type	Description
7-1	R	Reserved
0	W	Keystone correction enable 0h = Disabled 1h = Enabled

NOTE: Refer to “Keystone Parameters Supported Range” for valid range of Keystone Control Parameters.

3.1.80 Read Keystone Correction Control (Offset = 89h)

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

Table 3-111. Return Parameters

Parameter Bytes	Description
Byte 1	See
Byte 2	Optical throw ratio (LSByte)
Byte 3	Optical throw ratio (MSByte)
Byte 4	Optical DMD offset (LSByte)
Byte 5	Optical DMD offset (MSByte)

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

Table 3-112. Read Keystone Correction Control (89h) Register Field Descriptions

Bit	Type	Description
7-1	R	Reserved
0	R	Keystone correction enable 0h = Disabled 1h = Enabled

3.1.81 Write Border Color (Offset = B2h) [Reset = 0h]

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

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

Table 3-113. Write Border Color (B2h) Register Field Descriptions

Bit	Type	Description
7-3	R	Reserved
2-0	W	Display border color 0h = Black 1h = Red 2h = Green 3h = Blue 4h = Cyan 5h = Magenta 6h = Yellow 7h = White

NOTE: Whenever the display image size is smaller than the active area of the DMD, this border color are used for all non image pixels. Some examples where a border might come into play would be for a Window Box, Pillar Box, or Letterbox image.

NOTE: For the special case of displaying a pillar box image (Figure 3-9), the OEM can make use of the border color defined by this command, or make use of a dithered 24-bit border color. The definition of this dithered 24-bit border color, as well as the decision whether to use it, or use the color selected by this command is made with the DLP GUI software tool and stored in flash.

NOTE: The border color specified by this command is separate from the curtain color defined in the Write Display Image Curtain command (Section 3.1.10), even though they are both displayed using the curtain capability.

NOTE: The dithered 24-bit border color is specified in the VGP/CCP.

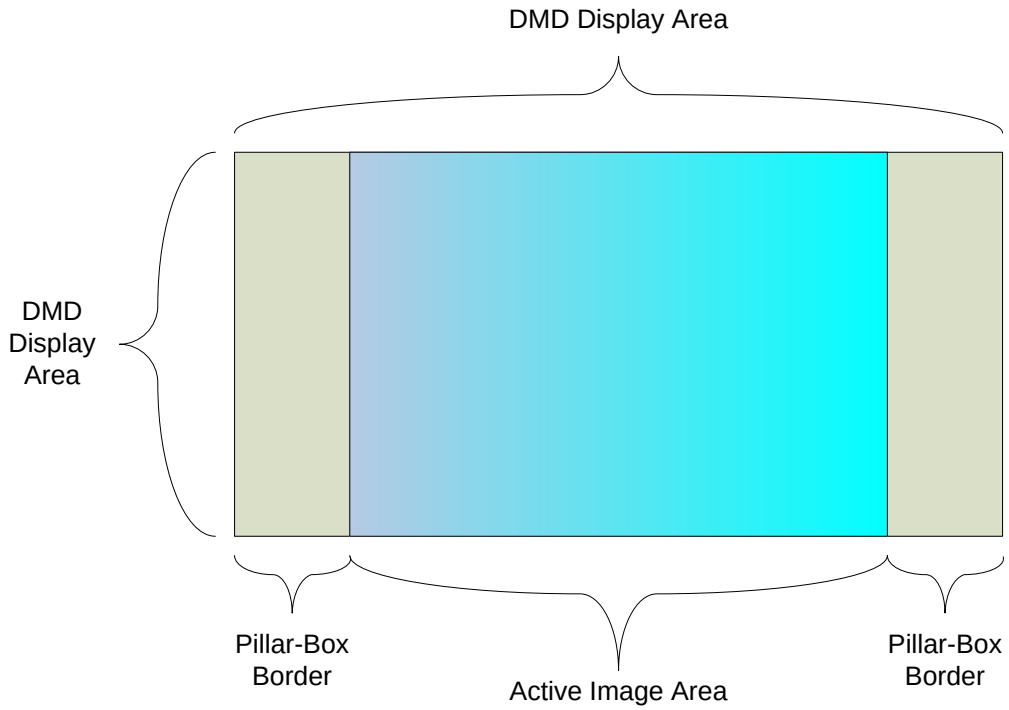

Figure 3-9. Pillar-Box Border Example

3.1.82 Read Border Color (Offset = B3h)

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

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

Table 3-114. Read Border Color (B3h) Register Field Descriptions

Bit	Type	Description
7	R	Pillar-box border color source 0h = Defined by this command 1h = Flash defined 24-bit color
6-3	R	Reserved
2-0	R	Display border color 0h = Black 1h = Red 2h = Green 3h = Blue 4h = Cyan 5h = Magenta 6h = Yellow 7h = White

NOTE: For the special case of a pillar box image (Figure 3-9), the OEM can make use of the border color defined by the Write Border Color command (Section 3.1.81), or make use of a dithered 24-bit border color. The definition of this dithered 24-bit border color, as well as the decision whether to use it or use the color selected by this command, is made with the DLP GUI software tool and stored in flash. The use decision stored in flash is shown by bit-7 of this command.

3.1.83 Write Keystone Projection Pitch Angle (Offset = BBh) [Reset = 0h]

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

Table 3-115. Write Parameters

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

Figure 3-10 shows the bit order and weighting for the 2's-complement projection pitch angle data.

Figure 3-10. Bit Weight Definition for the Projection Pitch Angle Data

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

This command is used in conjunction with the *Write Keystone Correction Control* command.

Format of pitch angle is defined as: Projection pitch angle = s7.8.

Refer to "Keystone Parameters Supported Range" for valid range of keystone control parameters. Figure 3-11 shows examples of the projection pitch angle.

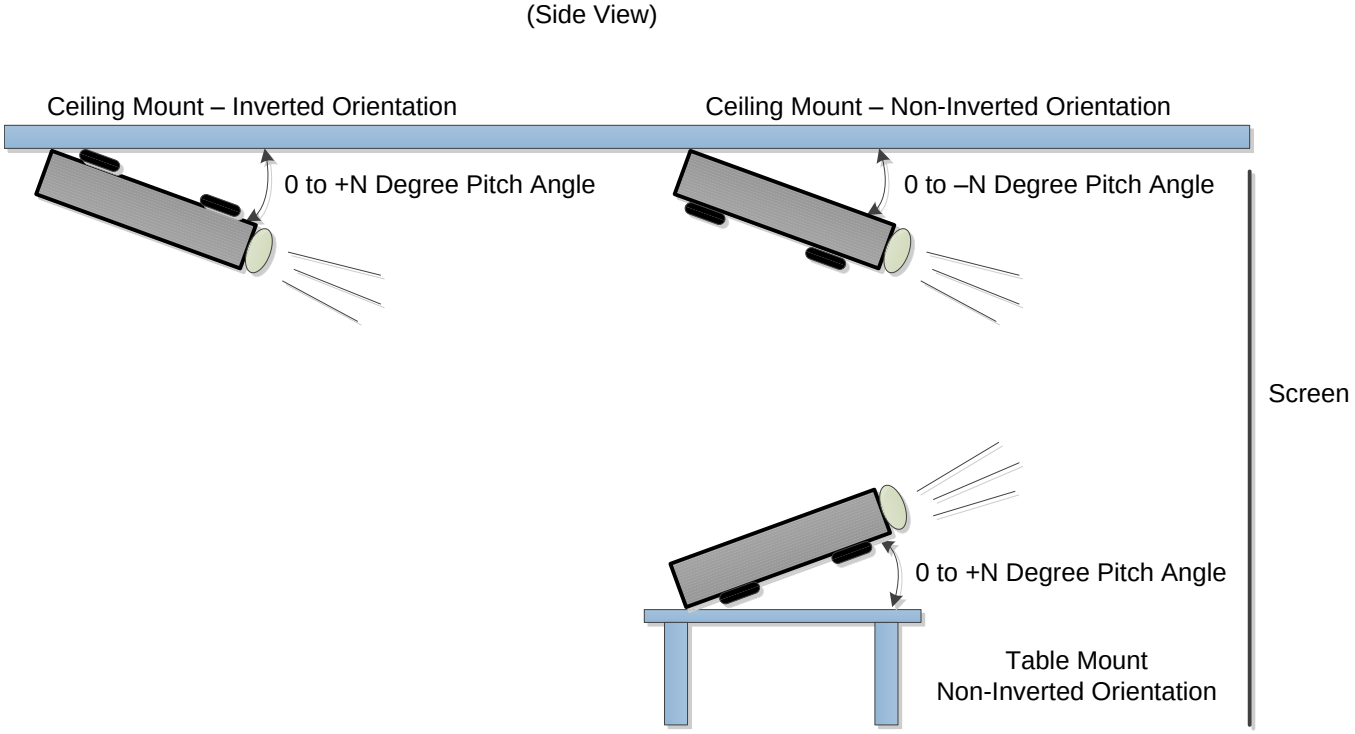

Figure 3-11. Examples of Projection Pitch Angle

3.1.84 Read Keystone Projection Pitch Angle (Offset = BCh)

This command reads the specified projection pitch angle for the display module.

Table 3-116. Return Parameters

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

3.1.85 Read Short Status (Offset = D0h)

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

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

Table 3-117. Read Short Status (D0h) Register Field Descriptions

Bit	Type	Description
7	R	Boot/main application 0h = Boot 1h = Main
6	R	Reserved
5	R	Flash error 0h = No error 1h = Error
4	R	Flash erase complete 0h = Complete 1h = Not complete
3	R	System error 0h = No error 1h = Error
2	R	Reserved
1	R	Communication error 0h = No error 1h = Error
0	R	System initialization 0h = Not complete 1h = Complete

NOTE: The flash erase complete status bit are set at the start of the flash erase process and are cleared when the erase process is complete. The flash status can be obtained during or after the erase process. To obtain this status during the erase process, only this command can be sent after the start of the flash erase. If any other command is sent during the erase process, it are held without processing until the flash erase has completed (thus blocking any following status requests until the previously sent command is processed).

NOTE: The flash error bit is used to indicate an error during any flash operation. For flash writes, this bit are updated at the end of each write transaction, however, once an error has been detected, this bit will remain in the error state until cleared. This will allow the OEM the option of checking the status between each write transaction, or at the end of the update. Once a write transaction has started, the flash status (and this error bit) will not be accessible until the write transaction has completed.

NOTE: The communication error bit is used to indicate any error on the I^2C command interfaces. Specific details about communication errors are available using the read communication status command.

NOTE: Any errors other than flash error and communication error are indicated by the system error bit. Specific details about system errors are available using the read system status command.

NOTE: The flash error, communication error, and system error bits are cleared when the read short status is read.

NOTE: The read short status command should only be checked periodically, not continuously. It is likely that continuous access will severely impact system performance.

3.1.86 Read System Status (Offset = D1h)

This command reads system status information for the display module.

Table 3-118. Return Parameters

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

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

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

Table 3-119. Byte 1 Read System Status (D1h) Register Field Descriptions

Bit	Type	Description
7-3	R	Reserved
2	R	DMD training error 0h = No error 1h = Error
1	R	DMD interface error 0h = No error 1h = Error
0	R	DMD device error 0h = No error 1h = Error

NOTE: The system will set the DMD device error for the following conditions:

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

NOTE: The system will set the DMD interface error when there are power management setup conflicts on this interface.

NOTE: The system will set the DMD training error when the training algorithm can't find a data eye that will meet the specified requirements.

Table 3-120. Byte 2 Read System Status (D1h) Register Field Descriptions

Bit	Type	Description
7-6	R	Reserved
5	R	Blue LED error 0h = No error 1h = Error
4	R	Green LED error 0h = No error 1h = Error
3	R	Red LED error 0h = No error 1h = Error
2	R	Blue LED state 0h = Off 1h = On
1	R	Green LED state 0h = Off 1h = On
0	R	Red LED state 0h = Off 1h = On

Table 3-121. Byte 3 Read System Status (D1h) Register Field Descriptions

Bit	Type	Description
7-2	R	Reserved
1	R	Sequence error 0h = No error 1h = Error
0	R	Sequence abort error 0h = No error 1h = Error

Table 3-122. Byte 4 Read System Status (D1h) Register Field Descriptions

Bit	Type	Description
7-6	R	Reserved
5	R	Watchdog timer timeout 0h = No timeout 1h = Timeout
4	R	Product configuration error 0h = No error 1h = Error
3	R	Master versus slave operation 0h = Master 1h = Slave
2	R	Single versus dual controller configuration 0h = Single 1h = Dual
1	R	SPI flashless communication error 0h = No error 1h = Error
0	R	SPI flashless data request error 0h = No error 1h = Error

NOTE: The system will set the SPI flashless data request error bit if the display does not start sending the requested data before the SPI flashless data request timeout is exceeded. Once the timeout is exceeded, the display will abort the current request, and then try again.

NOTE: The system will set the SPI flashless communication error bit if the display has three consecutive SPI flashless data request errors. If this happens, it is assumed that the SPI communication link is not operational, and system operations will halt. A reset are required to restart operations.

NOTE: The system will set the master versus slave bit as appropriate in both single and dual controller configurations.

NOTE: The system will set the product configuration error bit if it determines that some piece of the product configuration is not correct. Some examples are:

- Invalid controller/DMD combination
- Invalid controller/DLPA300x combination
- Invalid flash build for current controller, DMD, or DLPA300x configuration

NOTE: The system will set the watchdog timer timeout bit if the system has been reset due to a watchdog timer timeout.

3.1.87 Read System Software Version (Offset = D2h)

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

Table 3-123. Return Parameters

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

3.1.88 Read Communication Status (Offset = D3h)

This command reads system status information for the display module.

3.1.88.1 Read Parameters

Table 3-124 describes the read parameters.

Table 3-124. Read Parameters

Parameter Bytes	Description
Byte 1	Command bus status selection

MSB	Byte 1 – Command Bus Status Selection						LSB
b7	b6	b5	b4	b3	b2	b1	b0

Table 3-125. Read Communication Status (D3h) Register Field Descriptions

Bit	Type	Description
7-2	R	Reserved
1-0	R	Command bus status selection 00h = Reserved 01h = Reserved 10h = I^2C only 11h = Reserved

NOTE: This command will return the communication status for the command bus specified.

- Reserved: This selection will return status bytes 1 through 6
- Reserved: This selection will return status bytes 1 though 4
- I^2C only: This selection will return status bytes 5 though 6

3.1.88.2 Return Parameters

The return parameters are described below.

Table 3-126. Return Parameters

Parameter Bytes	Description
Byte 1	Reserved
Byte 2	Reserved
Byte 3	Reserved
Byte 4	Reserved
Byte 5	I^2C communication status
Byte 6	I^2C aborted offset

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

MSB	Byte 5 – Communication Status						LSB
b7	b6	b5	b4	b3	b2	b1	b0

Table 3-127. Byte 5 Read Communication Status (D3h) Register Field Descriptions

Bit	Type	Description
7	R	Reserved
6	R	Bus timeout by display error 0h = No error 1h = Error
5	R	Invalid number of command parameters 0h = No error 1h = Error
4	R	Read command error 0h = No error 1h = Error
3	R	Flash batch file error 0h = No error 1h = Error
2	R	Command processing error 0h = No error 1h = Error
1	R	Invalid command parameter value 0h = No error 1h = Error
0	R	Invalid command error 0h = No error 1h = Error

The system will set the invalid command error bit when it does not recognize the command offset. The invalid command offset are reported in the I^2C CMD error offset byte of this status.

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

The system will set the command processing error bit when a fault is detected when processing a command. In this case, the command is aborted with the system moving on to the next command. The offset for the aborted command is reported in the I^2C CMD error offset byte of this status.

The system will set 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, invalid command error).

The system will set 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 will set the Invalid number of command parameters error bit when too many or too few command parameters are received. In this case, the command is aborted with the system moving on to the next command. The offset for the aborted command is reported in the I^2C CMD error offset byte of this status.

The system will set the bus timeout by display error bit when the display releases control of the bus because the bus timeout value was exceeded.

MSB	Byte 6 – CMD Error Offset						LSB
b7	b6	b5	b4	b3	b2	b1	b0

Table 3-128. Read Communication Status (D3h) Register Field Descriptions

Bit	Type	Description
7-0	R	I^2C CMD error offset

The CMD error offset is associated with various I^2C communication status bits, and reports the offset for an I^2C command as noted.

3.1.89 Read Controller Device ID (Offset = D4h)

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

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

Table 3-129. Read Controller Device ID (D4h) Register Field Descriptions

Bit	Type	Description
7-4	R	Reserved
3-0	R	Controller device ID

The controller device ID can be decoded using Table 3-130.

Table 3-130. Controller Device ID Decode

Controller Device ID	Device Number	DMD Resolution	# of Controllers	Package	LED Driver
00h	DLPC3430	<1280x720	1	7mm x 7mm (0.4mm pitch)	DLPA200x
01h	DLPC3433	<1280x720	1	7mm x 7mm (0.4mm pitch)	DLPA200x/ DLPA3000
04h	DLPC3435	<1280x720	1	13mm x 13mm (0.8mm pitch)	DLPA200x
05h	DLPC3438	<1280x720	1	13mm x 13mm (0.8mm pitch)	DLPA200x/ DLPA3000
09h	DLPC3437	1920x1080	2	13mm x 13mm (0.8mm pitch)	DLPA3000/ 3005

3.1.90 Read DMD Device ID (Offset = D5h)

This command is used to read the DMD device ID for the display module.

3.1.90.1 Read Parameters

The read parameters are described below.

MSB	Byte 1 – DMD Register Selection						LSB
b7	b6	b5	b4	b3	b2	b1	b0

Table 3-131. Read DMD Device ID (D5h) Register Field Descriptions

Bit	Type	Description
7-3	R	Reserved
2-0	R	DMD data selection 0h = DMD device ID 1h – 7h = Reserved

3.1.90.2 Return Parameters

Table 3-132 describes the return parameters.

Table 3-132. DMD Device ID Reference Table

DMD Device ID			
Byte 1 (Identifier)	Byte 2 (Byte Count)	Byte 3 (ID-msbyte)	Byte 4 (ID-lsbyte)

3.1.91 Read System Temperature (Offset = D6h)

This command is used to read the system temperature for the display module.

Parameter Bytes	Description
Byte 1	See below (LSByte)
Byte 2	See below (MSByte)

Figure 3-12 shows the bit order and definition for the signed magnitude system temperature data, which is returned in °C. The unspecified msbits (bits 15:12) is set to '0'.

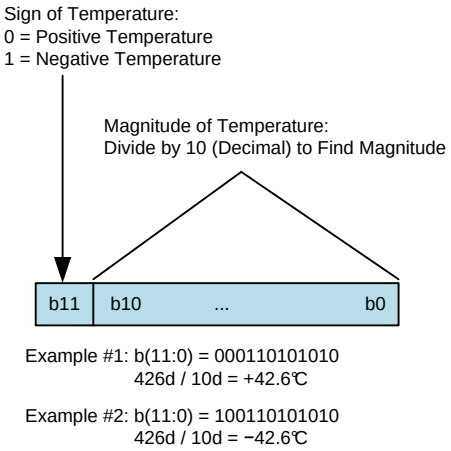

Figure 3-12. Bit Order and Definition

3.1.92 Read Flash Build Version (Offset = D9h)

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

Table 3-133. Return Parameters

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

The OEM is allowed to specify a version number for the controller flash build in the format specified by this command. This command allows the OEM to read back this version information.

3.1.93 Write Flash Batch File Delay (Offset = DBh) [Reset = User specified]

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

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^2C 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).

Software should make use of the available hardware timers.

3.1.94 Read DMD I/F Training Data (Offset = DCh)

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

3.1.94.1 Read Parameters

The command parameters are described below.

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

Table 3-134. Byte 1 Read DMD I/F Training Data (DCh) Register Field Descriptions

Bit	Type	Description
7-5	R	Reserved
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 will return the DMD I/F training data specified for the controller pin pair specified.

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

3.1.94.2 Return Parameters

The return parameters are described below.

Table 3-135. 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 3-136. 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
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 3-137. 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 3-138. 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 3-139. 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 contoller 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 should have 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.

3.1.95 Flash Update PreCheck (Offset = DDh)

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

3.1.95.1 Read Parameters

The command parameters are described below.

Table 3-140. Return 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)

3.1.95.2 Return Parameters

The return parameters are described below.

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

Table 3-141. Flash Update PreCheck (DDh) 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 OEM blocks, the size of the flash build will either overwrite some portion of the existing OEM 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 OEM is responsible for correcting the error before updating the flash. If the OEM chooses to ignore the error and update the flash anyway, the system will allow this. In this case, the OEM 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.

3.1.96 Flash Data Type Select (Offset = DEh) [Offset = 0h]

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

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 3-142. Flash Data Type Select (DEh) Register Field Descriptions

Bit	Type	Description
7-0	W	Flash data type Entire flash 00h = Entire flash 01h = Reserved 02h = Entire flash except OEM calibration data and OEM scratchpad data 03h - 0Fh = Reserved TI software 10h = Main software application 11h - 1Fh = Reserved TI application data 20h = TI application data set (AOM) 21h - 2Fh = Reserved OEM batch files 30h = OEM batch files 31h - 3Fh = Reserved Look data 40h = Look data set 41h - 4Fh = Reserved Sequence data 50h = Entire sequence data set 51h = Entire sequence data set (Reads only) 52h - 5Fh = Reserved Degamma/CMT data 60h = Entire degamma/CMT data set 61h = Partial degamma/CMT data set (reads only) 62h - 6Fh = Reserved CCA data 70h = CCA data set 71h - 7Fh = Reserved General LUT data 80h = CCA data set 81h - 8Fh = Reserved

Table 3-142. Flash Data Type Select (DEh) Register Field Descriptions (continued)

Bit	Type	Description
7-0	W	OEM Splash screen data 90h = Entire OEM splash screen data set 91h = Partial OEM splash screen data set 92h - 9Fh = Reserved OEM Calibration data A0h = OEM calibration data set A1h - AFh = Reserved OEM scratchpad data B0h = Entire OEM scratchpad data set 0 B1h = Partial OEM scratchpad data set 0 B2h = Entire OEM scratchpad data set 1 B3h = Partial OEM scratchpad data set 1 B4h = Entire OEM scratchpad data set 2 B5h = Partial OEM scratchpad data set 2 B6h = Entire OEM scratchpad data set 3 B7h = Partial OEM scratchpad data set 3 B8h - BFh = Reserved

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 is provided for each write or read transaction.

The specified flash data is written to or read from flash using the write flash start, write flash continue, read flash start, and read flash continue 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 3-143. 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 OEM splash screen set	Splash number	N/A	N/A	A splash screen is specified by its splash screen number
Partial OEM 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)
Partial sequence data set	Look number	Sequence index number		A sequence data set is specified by its sequence index number.
Partial CMT data set	Look number	Sequence index number		A CMT data set is specified by its CMT index number.
Partial OEM splash screen set	Splash number	N/A	N/A	A Splash screen is specified by its Splash screen number.
Partial OEM 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 OEM thermister LUT trim data, is stored in the OEM calibration block of the flash. It is the responsibility of the OEM to manage updates to this block, which may require the OEM 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 OEM make use of the flash update pre-check command before updating an existing flash build.

The system allows the OEM to allocate up to four separable blocks of flash space for their own use (OEM scratchpad data). The OEM 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 OEM to manage these data sets, including updates, which may require the OEM 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.

3.1.97 Flash Data Length (Offset = DFh) [Reset = 0h]

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

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.

3.1.98 Erase Flash Data (Offset = E0h) [Reset = 0h]

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

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

Because 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 is set at the start of the erase process, and is 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 offset and four signature bytes must be received correctly before this command is recognized and executed.

3.1.99 Write Flash Start (Offset = E1h)

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

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 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 is 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 OEM 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.

3.1.100 Write Flash Continue (Offset = E2h)

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

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 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 is 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 OEM 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.

3.1.101 Read Flash Start (Offset = E3h)

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

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 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.

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

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.

3.1.102 Read Flash Continue (E4h)

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

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.

Appendix

A.1 Legal Notice

Important Notice

Texas Instruments and its subsidiaries (TI) reserve the right to make changes to their products or to discontinue any product or service without notice, and advise customers to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgment, including those pertaining to warranty, patent infringement, and limitation of liability.

TI warrants performance of its products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements. Customers are responsible for their applications using TI components.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards must be provided by the customer to minimize inherent or procedural hazards. TI assumes no liability for applications assistance or customer product design. TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such products or services might be or are used. TI's publication of information regarding any third party's products or services does not constitute TI's approval, license, warranty or endorsement thereof.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations and notices. Representation or reproduction of this information with alteration voids all warranties provided for an associated TI product or service, is an unfair and deceptive business practice, and TI is not responsible nor liable for any such use.

Resale of TI's products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service, is an unfair and deceptive business practice, and TI is not responsible nor liable for any such use.

Also see: Standard Terms and Conditions of Sale for Semiconductor Products.

IMPORTANT NOTICE AND DISCLAIMER

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

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

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

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