DLPC3436 Software Programmer's Guide

User's Guide

Literature Number: DLPU078
July 2019

Contents

List of Figures

List of Tables

Abstract

This guide details the software interface requirements for a DLPC3436 dual controller based system. It defines all applicable communication protocols including I^2C, initialization, default settings and timing.

Related Documentation

These data sheets describe the DLPC3436 chipset components.

- DLPC3436 (DLPS156) DLP Display controller for DLP230NP DMD
- DLPA2005 PMIC and High-Current LED Driver (DLPS047)
- DLPA2000 PMIC and High-Current LED Driver (DLPS043)
- DLPA3000 PMIC and High-Current LED Driver (DLPS052)
- DLP230NP 0.23 1080p DMD (DLPS144)

Trademarks

DLPC3436 Software Programmer's Guide

1.1 Introduction

1.1.1 Software Programmer's Guide Overview

This guide details the software interface requirements for a DLPC3436 dual controller based system. It defines all applicable communication protocols including I²C, initialization, default settings and timing. The DLPC3436 system can be used with associated devices in Figure 1-1.

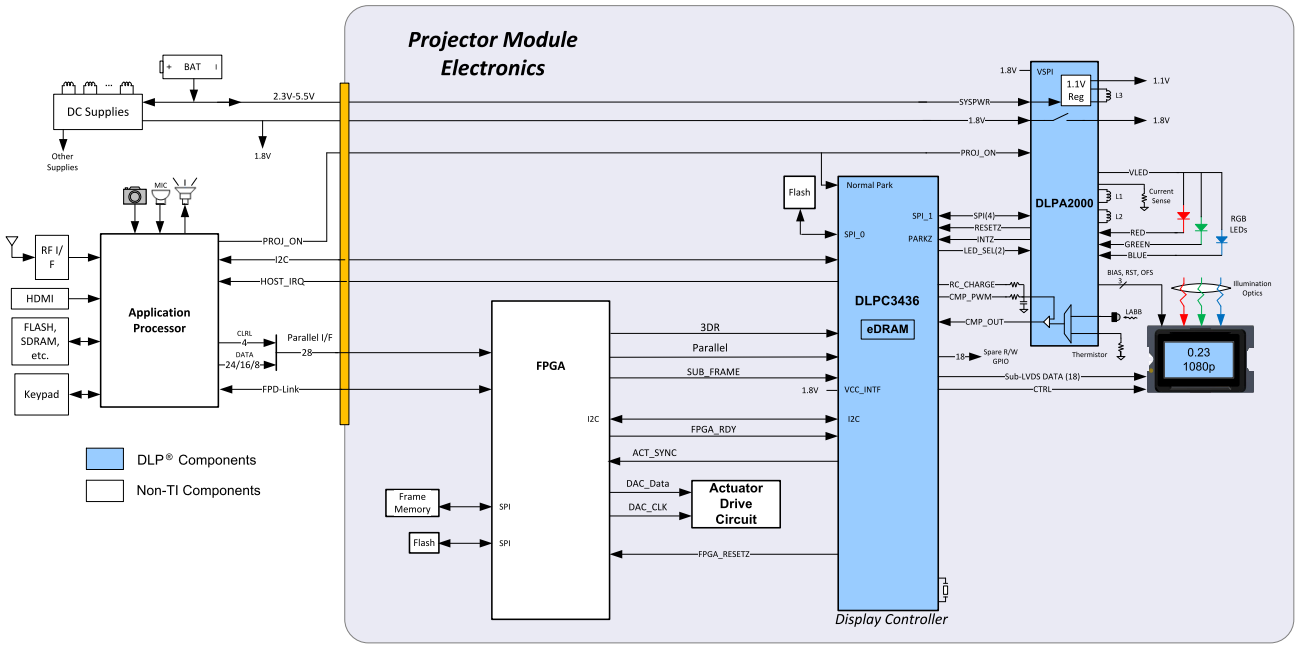

Figure 1-1. DLPC343x Embedded Configuration

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 100 KHz) and a 24-bit parallel interface.

Interface Specification

2.1 Electrical Interface

This section discusses the requirements for a number of interface signals that are not command or data busses. These signals allow different boot options.

2.1.1 System Power-up Associated Signals

2.1.1.1 EXT-BOOT-EN

The controller hardware uses the EXT-BOOT-EN signal to determine which application to use during the controller initialization process of the system power-up sequence. Either the internal boot application, or an external boot application (located in FLASh), can be used. See also Section 2.2.

2.1.1.2 DIS-PGM-LD

The boot application uses the DIS-PGM-LD signal to direct the function of the system boot application during the controller initialization process of the system power-up sequence. See also Section 2.2.

2.1.1.3 SPI-FLS-EN

The boot application uses the SPI-FLS-EN signal to direct the function of the system boot application during the controller initialization process of the system power-up sequence. See also Section 2.2.

2.1.1.4 High Level Definition

As noted, a more detailed discussion of these signals is provided in Section 2.2; however, a brief summary is in Table 2-1.

Table 2-1. Summary of Settings for Power-up Associated Signals

EXT-BOOT-EN	DIS-PGM-LD	SPI-FLS-EN	USE/DEFINITION
0	0	0	Normal Flash Operation
0	0	1	SPI Flashless Operation
0	1	0	Bad Flash Flashless Operation
0	1	1	N/A
1	0	0	TI Debug
1	0	1	N/A
1	1	0	TI Debug
1	1	1	N/A

NOTE: The DLPC3436 device supports normal flash operation only.

2.2 System Initialization

This section discusses system initialization methodologies.

2.2.1 Boot ROM Concept

The DLPC343x controller includes boot ROM, and associated boot software. This *resident* boot code consist of the minimum code needed to complete the various tasks required based on the state of the DIS-PGM-LD (disable program load) pin and the SPI-FLS-EN (SPI flashless enable) pin.

2.2.2 Internal vs. External Boot Software

In the DLPC343x controller, the state of the EXT-BOOT-EN (external boot enable) pin allows the external user to specify the microprocessor hardware target. The target options are:

- the internal boot ROM for the boot application (EXT-BOOT-EN = ‘0’)
- an external FLASH for the boot application (EXT-BOOT-EN = ‘1’)

The system allows for the use of an external boot program in FLASH for debuging and boot code development purposes only because microprocessor code execution out of serial flash is extremely slow.

2.2.3 Flash and Flashless Product Configurations

In most DLPC343x family product configurations, an external FLASH device stores the main application code, along with all of the other configuration and operational data required by the system for normal operation.

Specific applications eliminate the external FLASH component to reduce the cost. In these flashless configurations, the boot application typically downloads the main application code (by command) to iRAM via the SPI port. the Main Application code obtains all other configuration and operational data (normally obtained from flash) via the SPI port.

All discussions in this document, unless stated otherwise, assume that the system includces an external FLASH device.

2.2.4 Resident Boot Software (EXT-BOOT-EN = 0)

The resident boot code consists of the minimum code required based on the state of the DIS-PGM_LD and SPI-FLS_EN pins (with EXT-BOOT-EN = ‘0’). Figure 2-1 shows an overview of these tasks. .

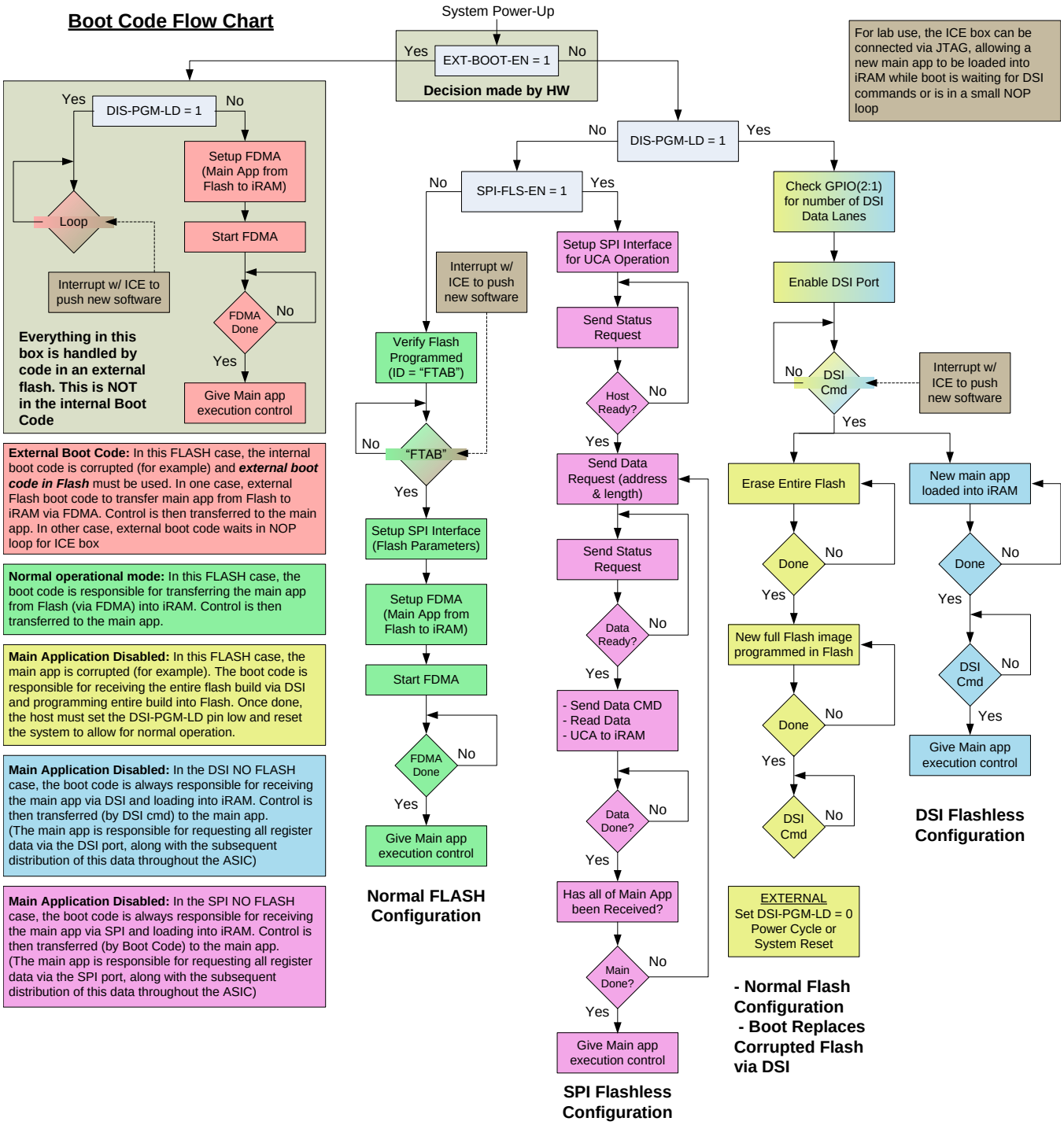

Figure 2-1. Boot Code Flow Chart

2.2.4.1 DIS-PGM-LD = 0 (with EXT-BOOT-EN = 0)

2.2.4.1.1 SPI-FLS-EN = 0

SPI-FLS-EN = 0 sets normal operational mode of the boot application for a flash-based product configuration.

2.2.4.1.2 SPI-FLS-EN = 1

SPI-FLS-EN = 1 sets normal operational mode of the boot application for an SPI-based (flashless) product configuration. In this case, the boot application expects to get the main application from the host via the SPI port in response to TI command requests. The only SPI interface instructions that are supported by the boot code are associated with requesting and read data from the host via this port.

2.3 Software Interface

In general, the DLPC3436 controller supports one set of software commands. This custom set of TI specific commands apply for use over the I^2C command interface.

2.3.1 Software Command Philosophy

In the DLPC3436 controller, software processes all commands via I^2C. No commands directly address or access controller registers, controller mailboxes, or any attached flash parts. All commands are high level, and of a more abstract nature, and as such decouple the OEM from the internal hardware of the controller.

2.3.2 I^2C Considerations

2.3.2.1 I^2C Transactions

Because software processes all I^2C commands, there is only one type of I^2C transaction that requires support.. Table 2-2 lists the transaction type for both write and read functions. The I^2C interface supports variably sized transactions (for example, both a one byte transaction, and a nine byte transaction) to match the TI commands that are discussed later in this document.

Table 2-2. I^2C Write and Read Transactions

Transaction	Address[1]	Sub-Address[2]	Remaining Data Bytes[3]
Write	8-bits	8-bits	8-bit parameter bytes (0 → N)
	36h (or 3Ah)	Command value	Parameter values
Read request	8-bits	8-bits	8-bit parameter bytes (0 → N)
	36h (or 3Ah)	Command value	Parameter values
Read response	8-bits		8-bit parameter bytes (0 → N)
	37h (or 3Bh)		Parameter values

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

2.3.2.2 Data Flow Control

I^2C interface inherently supports flow control by holding the clock. However this control is most likely not sufficient for all transactions (sequence and CMT updates for example). In this case, the host software can use of the Read Short Status command to determine if the system is busy.

List of System Write/Read Software Commands

This section describes the commands supported by the I^2C interface.

Table 3-1. Supported TI Generic Commands

Offset	Command Type	Command Description	Reset Value	Default Action	Section
05h	Write	Write Source Select	1h	Test pattern	Section 3.1.1
06h	Read	Read Source Select			Section 3.1.2
0Dh	Write	Write Splash Screen Select		User specified	Section 3.1.3
0Eh	Read	Read Splash Screen Select			Section 3.1.4
0Fh	Read	Read Splash Screen Header			Section 3.1.5
14h	Write	Write Display Image Orientation		User specified	Section 3.1.6
15h	Read	Read Display Image Orientation			Section 3.1.7
16h	Write	Write Display Image Curtain	1h	Black	Section 3.1.8
17h	Read	Read Display Image Curtain			Section 3.1.9
1Ah	Write	Write Image Freeze	0h	No freeze	Section 3.1.10
1Bh	Read	Read Image Freeze			Section 3.1.11
20h	Write	Write 3-D Control	0h	Automatic	Section 3.1.12
21h	Read	Read 3-D Control			Section 3.1.13
22h	Write	Write Look Select		User specified	Section 3.1.14
23h	Read	Read Look Select			Section 3.1.15
26h	Read	Read Sequence Header Attributes			Section 3.1.16
27h	Write	Write Degamma/CMT Select		User specified	Section 3.1.17
28h	Read	Read Degamma/CMT Select			Section 3.1.18
29h	Write	Write CCA Select		User specified	Section 3.1.19
2Ah	Read	Read CCA Select			Section 3.1.20
2Dh	Write	Write Execute Batch File	0h		Section 3.1.21
30h	Write	Write 3-D Reference	0h	Next frame left	Section 3.1.22

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

Offset	Command Type	Command Description	Reset Value	Default Action	Section
39h	Write	Write Mirror Lock Control	0h		Section 3.1.23
3Ah	Read	Read Mirror Lock Control			Section 3.1.24
4Bh	Write	Write FPD Link Pixel Map Mode			Section 3.1.25
4Ch	Read	Read FPD Link Pixel Map Mode			Section 3.1.26
4Dh	Write	Write FPGA Input Video Chroma Processing Select			Section 3.1.27
4Eh	Read	Read FPGA Input Video Chroma Processing Select			Section 3.1.28
50h	Write	Write LED Output Control Method		User specified	Section 3.1.29
51h	Read	Read LED Output Control Method			Section 3.1.30
52h	Write	Write RGB LED Enable	7h	Enabled	Section 3.1.31
53h	Read	Read RGB LED Enable			Section 3.1.32
54h	Write	Write RGB LED Current		User specified	Section 3.1.33
55h	Read	Read RGB LED Current			Section 3.1.34
57h	Read	Read CAIC LED Max Available Power			Section 3.1.35
5Ch	Write	Write RGB LED Max Current		User specified	Section 3.1.36
5Dh	Read	Read RGB LED Max Current			Section 3.1.37
5Eh	Read	Read Measured LED Parameters			Section 3.1.38
5Fh	Read	Read CAIC RGB LED Current			Section 3.1.39
60h	Write	Write XPR FPGA Input Image Size			Section 3.1.40
61h	Read	Read XPR FPGA Input Image Size			Section 3.1.41
64h	Read	Read XPR FPGA Version			Section 3.1.42
67h	Write	Write XPR FPGA Test Pattern Select			Section 3.1.43
68h	Read	Read XPR FPGA Test Pattern Select			Section 3.1.44
6Bh	Write	Write XPR FPGA Parallel Video Control			Section 3.1.45
6Ch	Read	Read XPR FPGA Parallel Video Control			Section 3.1.46
6Dh	Write	Write XPR FPGA Video Format Select			Section 3.1.47
6Eh	Read	Read XPR FPGA Video Format Select			Section 3.1.48
6Fh	Read	Read XPR FPGA Status			Section 3.1.49

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

Offset	Command Type	Command Description	Reset Value	Default Action	Section
70h	Write	Write Actuator Latency			Section 3.1.50
71h	Read	Read Actuator Latency			Section 3.1.51
72h	Write	Write Actuator Gain			Section 3.1.52
73h	Read	Read Actuator Gain			Section 3.1.53
74h	Write	Write Segment Length			Section 3.1.54
75h	Read	Read Segment Length			Section 3.1.55
76h	Write	Write Manual Actuator Sync Delay			Section 3.1.56
77h	Read	Read Manual Actuator Sync Delay			Section 3.1.57
78h	Write	Write Manual Actuator Offset			Section 3.1.58
79h	Read	Read Manual Actuator Offset			Section 3.1.59
80h	Write	Write Local Area Brightness Boost Control	1h	Manual strength control	Section 3.1.60
81h	Read	Read Local Area Brightness Boost Control			Section 3.1.61
84h	Write	Write CAIC Image Processing Control		User specified	Section 3.1.62
85h	Read	Read CAIC Image Processing Control			Section 3.1.63
86h	Write	Write CCA Control	1h	Enabled	Section 3.1.64
87h	Read	Read CCA Control			Section 3.1.65
88h	Write	Write Keystone Correction Control			Section 3.1.66
89h	Read	Read Keystone Correction Control			Section 3.1.67
A0h	Write	Write Actuator Number of Segments			Section 3.1.68
A1h	Read	Read Actuator Number of Segments			Section 3.1.69
A2h	Write	Write Actuator Configuration Select			Section 3.1.70
A3h	Read	Read Actuator Configuration Select			Section 3.1.71
A4h	Write	Write Actuator Fixed Level Value			Section 3.1.72
A5h	Read	Read Actuator Fixed Level Value			Section 3.1.73
A6h	Write	Write Actuator Period Stretch Value			Section 3.1.74
A7h	Read	Read Actuator Period Stretch Value			Section 3.1.75
A8h	Write	Write Actuator Reference Value			Section 3.1.76
A9h	Read	Read Actuator Reference Value			Section 3.1.77

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

Offset	Command Type	Command Description	Reset Value	Default Action	Section
AAh	Write	Write Actuator Output Select			Section 3.1.78
ABh	Read	Read Actuator Output Select			Section 3.1.79
ACh	Write	Write Actuator Edge Table Address Mode			Section 3.1.80
ADh	Read	Read Actuator Edge Table Address Mode			Section 3.1.81
AEh	Write	Write Actuator DAC Enable			Section 3.1.82
AFh	Read	Read Actuator DAC Enable			Section 3.1.83
BAh	Read	Read Auto Framing Information			Section 3.1.84
BBh	Write	Write Keystone Projection Pitch Angle			Section 3.1.85
BCh	Read	Read Keystone Projection Pitch Angle			Section 3.1.86
C2h	Write	Write Actuator Watchdog Window Width			Section 3.1.87
C3h	Read	Read Actuator Watchdog Window Width			Section 3.1.88
C4h	Write	Write Actuator Subframe Filter Width			Section 3.1.89
C5h	Read	Read Actuator Subframe Filter Width			Section 3.1.90
C6h	Write	Write Actuator Stepped/Fixed Output Invert Enable			Section 3.1.91
C7h	Read	Read Actuator Stepped/Fixed Output Invert Enable			Section 3.1.92
C8h	Write	Write Actuator Orientation			Section 3.1.93
C9h	Read	Read Actuator Orientation			Section 3.1.94
D0h	Read	Read Short Status			Section 3.1.95
D1h	Read	Read System Status			Section 3.1.96
D2h	Read	Read System Software Version			Section 3.1.97
D3h	Read	Read Communication Status			Section 3.1.98
D4h	Read	Read Controller Device ID			Section 3.1.99
D5h	Read	Read DMD Device ID			Section 3.1.100
D6h	Read	Read System Temperature			Section 3.1.101
D9h	Read	Read Flash Build Version			Section 3.1.102
DBh	Write	Write Batch File Delay		User specified	Section 3.1.103
DCh	Read	Read DMD I/F Training Data			Section 3.1.104

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

Offset	Command Type	Command Description	Reset Value	Default Action	Section
DDh	Read	Flash Update PreCheck		Entire flash	Section 3.1.105
DEh	Write	Flash Data Type Select			Section 3.1.106
DFh	Write	Flash Data Length			Section 3.1.107
E0h	Write	Erase Flash Data			Section 3.1.108
E1h	Write	Write Flash Start			Section 3.1.109
E2h	Write	Write Flash Continue			Section 3.1.110
E3h	Read	Read Flash Start			Section 3.1.111
E4h	Read	Read Flash Continue			Section 3.1.112

3.1 System Write/Read Commands

3.1.1 Write Source Select (05h)

Use this command to control the system display module.

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

Table 3-2. Byte 1 Write Source Select Register Field Descriptions

Bit	Type	Description
7-3	R	Reserved
2-0	W	Input Source 0h = Test Pattern Generator from XPR FPGA 1h = External Parallel Video from XPR FPGA 2h = FPD-Link or LVDS Source from XPR FPGA 3h = Internal Controller Splash Screen 4h = Internal Controller Test Pattern 5h - 7h = Reserved

The Write XPR FPGA Test Pattern Select (Section 3.1.43) command applies only when selecting the test pattern generator from FPGA .

The Write Splash Screen Select (Section 3.1.3) command applies only when selecting the Splash Screen.

Enabling the External Calibration command reconfigures the system to disable all FPGA image processing. This result allows the user to apply a custom test pattern for calibration purposes.

The External Video Port, Test Pattern Generator and Splash Screen input select options share the following commands:

- Write Display Image Orientation
- Write Display Image Curtain
- Write Look Select
- Write Local Area Brightness Boost Control
- Write CAIC Image Processing Control

While the values for these commands may be the same across the different input source types, the hardware settings may change. For example, if the user changes to a Test Pattern Generator input source, the size of the test pattern must match the size of the DMD. Therefore, the display scaler settings may need to be adjusted.

The user must specify the active data size for all external input sources using the Write Input Image Size command. For input image data on the Parallel bus that doesn't provide data framing information, the user must provide manual framing data using the Parallel I/F Manual Image Framing command.

The software generates selected test patterns at the resolution of the DMD.

The *Write Image Freeze* command offers options for hiding on-screen artifacts when during the input source selection process.

3.1.2 Read Source Select (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 3-3. Byte 1 Read Source Select Register Field Descriptions

Bit	Type	Description
7-3	R	Reserved
2-0	W	Input Source 0h = Test Pattern Generator from XPR FPGA 1h = External Parallel Video from XPR FPGA 2h = FPD-Link or LVDS Source from XPR FPGA 3h = Internal Controller Test Pattern Generator 4h = Internal Controller Splash Screen 5h - 7h = Reserved

3.1.3 Write Splash Screen Select (0Dh)

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 Source Select (Section 3.1.1) command. It specifies which splash screen to display when the Source Select command selects splash screen as the image source. The software retains the settings for this command 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 Source Select)

Follow these guidelins for proper operation.

- The Splash Screen is a unique source since it is read from Flash and sent down the processing path of the controller one time, to be stored in memory for display at the end of the processing path. As such, 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 Source Select command.
- *It is important that the user review the notes for the Write 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.*
- The availability of splash screens is limited by the available space in flash memory.
- All splash screens must be landscape oriented.
- For single controller applications which support DMD resolutions up to 1280 × 720, the minimum splash image size allowed for flash storage is 427 × 240, with the maximum being the resolution of the product DMD. For this system the maximum splash screen size is 960 × 540. Typical splash image sizes for flash are 427 × 240 and 640 × 360. The full resolution size is typically used to support an Optical Test splash screen.
- For dual controller applications which support DMD resolutions up to 1980 × 1080, the minimum splash image size allowed for flash storage is 854 × 480, with the maximum being the resolution of the product DMD. Typical splash image sizes for flash are 854 × 480. The full resolution size is typically used to support an Optical Test splash screen.
- The user must specifying how the splash image displays on the screen.
- Because the splash selection loads the frame memory only once, it must be re-selected after changing image display settings such as image orientation.

3.1.4 Read Splash Screen Select (0Eh)

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

Table 3-4. Return Parameters

Parameter Bytes	Description
Byte 1	Splash screen selected (integer)

3.1.5 Read Splash Screen Header (0Fh)

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

3.1.5.1 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 3-5. Read Parameters

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

3.1.5.2 Return Parameters

Table 3-6 describes the return parameters.

Table 3-6. 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

Table 3-7 lists the parameter definitions.

Table 3-7. Splash Screen Header Definitions

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

3.1.6 Write Display Image Orientation (14h)

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 3-8. Write Display Image Orientation 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

3.1.7 Read Display Image Orientation (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 3-9. Read Display Image Orientation 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

3.1.8 Write Display Image Curtain (16h)

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

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

Table 3-10. Write Display Image Curtain 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

3.1.9 Read Display Image Curtain (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 3-11. Read Display Image Curtain 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

3.1.10 Write Image Freeze (1Ah)

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 3-12. Write Image Freeze 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. Section 3.1.10.1 discusses suggestions to the host system for the types of image changes likely to necessitate the use of the image freeze command to hide artifacts.

The controller software never freezea or unfreezea the image, neither automatically nor under-the-hood.The controller software does not freeze or unfreeze the image for any reason except when explicitly commanded by the Write Image Freeze command. If the OEM chooses not to make use of the Image Freeze function, make sure to change the source itself before changing image parameters to minimize transition artifacts.

NOTE: Review the notes for the Write Source Select command in Section 3.1.1 to understand the concept of source associated commands. This concept determines when the system executes source associated commands.

3.1.10.1 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 minimizes, if not eliminates, 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 the controller processes commands serially, it requires no special timing or delay between these commands. Ensure that only a small number of commands are placed between the freeze and unfreeze, so that the image is not frozen for a long period of time. Table 3-13 lists the commands that may produce image artifacts. However, this list is not an inclusive. Users must determining if and when use of the image freeze command meets the application requirements.

Table 3-13. Partial List of Commands that May Benefit from the Use of Image Freeze

Command	Command Offset	Notes
Write Source Select	05h	
Write XPR FPGA Video Source Format Select	6Dh	
Write Look Select	22h	

Table 3-14 and Table 3-15 show a few examples of how to use the image freeze command.

Table 3-14. 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 3-15. 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 Source Select command.
Write Source Select = test pattern generator	
Write Image Freeze = unfreeze	

3.1.11 Read Image Freeze (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 3-16. Read Image Freeze Register Field Descriptions

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

3.1.12 Write 3-D Control (20h)

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

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

Table 3-17. Write 3-D Control 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

Follow these guidelines:

- 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 take effect following the next VSYNC.
- 3-D image data must always be sent as frame sequential (that is, syncs and blanking commands are sent between every eye frame), at frame rates greater than approximately 94 Hz, The controller does not support frame rate multiplication. The Dual controller system configuration does not support Internal Reference Generator.
- The 3-D Reference specifies whether a frame of data contains left eye data or right eye data. An external hardware signal can provide this 3-D reference to the display. Table 3-18 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. The Polarity of 3-D Reference parameter is applicable only when the use selects the External Signal as the 3-D Reference source.

Table 3-18. 3D Control

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

Use the Write 3-D Reference commandwith this selection.

For frame sequential 3-D, Frame Dominance determines which eye frames in the data stream combine to create 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 equence is important for proper operation of display histograms (which span both eye frames of a single image), and when the image is frozen, to ensure display of the correct two eye frames together.

Do not use the frame dominance control to attempt correction for misalignment of the 3-D reference signal to the image data.

3.1.13 Read 3-D Control (21h)

This command reads 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 3-19. Read 3-D Control 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.

3.1.14 Write Look Select (22h)

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 3-20. Write Look Select 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.

3.1.15 Read Look Select (23h)

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

Table 3-21. Return Parameters

Parameter Bytes	Description
Byte 1	Look Number.
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 3-22. Byte 1 Read Look Select Register Field Descriptions

Bit	Type	Description
7-0	R	look number

Table 3-23. Byte 2 Read Look Select Register Field Descriptions

Bit	Type	Description
7-0	R	Sequence number

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

Sequences are specified by an enumerated value (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.

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.

3.1.16 Read Sequence Header Attributes (26h)

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

3.1.16.1 Return Parameters

Table 3-24 describes the return parameters.

Table 3-24. 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.

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

Table 3-25. Read Sequence Header Attributes Register Field Descriptions

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

3.1.17 Write Gamma/CMT Select (27h)

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 3-26. Write Gamma/CMT SelectRegister Field Descriptions

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

Degamma/CMT LUTs are stored in Flash until needed.

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

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 has a CMT Index of 0 that references a degamma of 1.5 that is appropriate for each respective sequence.

3.1.18 Read Gamma/CMT Select (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 3-27. Read Gamma/CMT Select Register Field Descriptions

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

3.1.19 Write CCA Select (29h)

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 3-28. Write CCA Select Register Field Descriptions

Bit	Type	Description
7-0	W	CCA Parameter Set

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

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.

3.1.20 Read CCA Select (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 3-29. Read CCA Select Register Field Descriptions

Bit	Type	Description
7-0	R	CCA Parameter Set

3.1.21 Write Execute Flash Batch File (2Dh)

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

Table 3-30. 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 3-31. 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.

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

Flash batch file 0 is a special Auto-Init batch file that is run automatically by the DLPC3436 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.

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.

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 3.1.103).

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 3-31. 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	
Write External PAD Address	EB	
Write External PAD Data	EC	
All Read commands	Various	Various

3.1.22 Write 3-D Reference (30h)

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

Table 3-32. Write Parameters

Parameter Bytes	Description
Byte 1	Batch file number

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

Table 3-33. Write Execute Flash Batch File 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 3.1.12). 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.

When the Write 3-D Reference command is received, teh device applies its parameter value at the next VSYNC (that is, the parameter value is applied to the image data following the next VSYNC or Start of Frame command).

When this command is received, software establishes 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.

3.1.23 Write Mirror Lock Control (39h)

This command is used to set the lock state of the DMD interface for optical alignment purposes.

Table 3-34. Write Parameters

Parameter Bytes	Description
Byte 1	Batch file number

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

Table 3-35. Write Execute Flash Batch File Register Field Descriptions

Bit	Type	Description
7-2	R	Reserved
1-0	W	Mirror Lock State Selection 0h = Reserved 1h = DMD Interface Lock 2h = DMD Interface Unlock 3h = DMD Interface Unlock, Delay 100ms, DMD Interface Lock

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

3.1.24 Read Mirror Lock Control (3Ah)

This command is used to read the lock state of the DMD interface.

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 Register Field Descriptions

Bit	Type	Description
7-2	R	Reserved
1-0	R	Mirror Lock State 0h = DMD Interface is Locked 1h = DMD Interface is Unlocked

3.1.25 Write FPD Link Pixel Map Mode (4Bh)

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

Table 3-38. FPD Link Data Parameters

Parameter Bytes	Description
Byte 0	Reserved
Byte 1	Reserved
Byte 2	See below

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

Table 3-39. Write FPD Link Pixel Map Mode 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 3-40. FPD LVDS Data Bus Encoding

	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6
FPD Bus A - Data_A Channel						
FPD_A_DATA_A_6	Green_4	Green_2	Green_0	Green_4	Green_0	Green_2
FPD_A_DATA_A_5	Red_9	Red_7	Red_5	Red_9	Red_5	Red_7
FPD_A_DATA_A_4	Red_8	Red_6	Red_4	Red_8	Red_4	Red_6
FPD_A_DATA_A_3	Red_7	Red_5	Red_3	Red_7	Red_3	Red_5
FPD_A_DATA_A_2	Red_6	Red_4	Red_2	Red_6	Red_2	Red_4
FPD_A_DATA_A_1	Red_5	Red_3	Red_1	Red_5	Red_1	Red_3
FPD_A_DATA_A_0	Red_4	Red_2	Red_0	Red_4	Red_0	Red_2
FPD Bus A - Data_B Channel						
FPD_A_DATA_B_6	Blue_5	Blue_3	Blue_1	Blue_5	Blue_1	Blue_3
FPD_A_DATA_B_5	Blue_4	Blue_2	Blue_0	Blue_4	Blue_0	Blue_2
FPD_A_DATA_B_4	Green_9	Green_7	Green_5	Green_9	Green_5	Green_7
FPD_A_DATA_B_3	Green_8	Green_6	Green_4	Green_8	Green_4	Green_6
FPD_A_DATA_B_2	Green_7	Green_5	Green_3	Green_7	Green_3	Green_5
FPD_A_DATA_B_1	Green_6	Green_4	Green_2	Green_6	Green_2	Green_4
FPD_A_DATA_B_0	Green_5	Green_3	Green_1	Green_5	Green_1	Green_3

Table 3-40. FPD LVDS Data Bus Encoding (continued)

	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6
FPD Bus A - Data_C Channel						
FPD_A_DATA_C_6	DEN	DEN	DEN	DEN	DEN	DEN
FPD_A_DATA_C_5	VSYNC	VSYNC	VSYNC	VSYNC	VSYNC	VSYNC
FPD_A_DATA_C_4	HSYNC	HSYNC	HSYNC	HSYNC	HSYNC	HSYNC
FPD_A_DATA_C_3	Blue_9	Blue_7	Blue_5	Blue_9	Blue_5	Blue_7
FPD_A_DATA_C_2	Blue_8	Blue_6	Blue_4	Blue_8	Blue_4	Blue_6
FPD_A_DATA_C_1	Blue_7	Blue_5	Blue_3	Blue_7	Blue_3	Blue_5
FPD_A_DATA_C_0	Blue_6	Blue_4	Blue_2	Blue_6	Blue_2	Blue_4
FPD Bus A - Data_D Channel						
FPD_A_DATA_D_6	Map to Field	Map to Field	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	Blue_7	Blue_1
FPD_A_DATA_D_4	Blue_2	Blue_8	Blue_6	not used	Blue_6	Blue_0
FPD_A_DATA_D_3	Green_3	Green_9	Green_7	not used	Green_7	Green_1
FPD_A_DATA_D_2	Green_2	Green_8	Green_6	not used	Green_6	Green_0
FPD_A_DATA_D_1	Red_3	Red_9	Red_7	not used	Red_7	Red_1
FPD_A_DATA_D_0	Red_2	Red_8	Red_6	not used	Red_6	Red_0
FPD Bus A - Data_E Channel						
FPD_A_DATA_E_6	Map to Field	Map to Field	Map to Field	Map to Field	not used	not used
FPD_A_DATA_E_5	Blue_1	Blue_1	Blue_9	not used	not used	not used
FPD_A_DATA_E_4	Blue_0	Blue_0	Blue_8	not used	not used	not used
FPD_A_DATA_E_3	Green_1	Green_1	Green_9	not used	not used	not used
FPD_A_DATA_E_2	Green_0	Green_0	Green_8	not used	not used	not used
FPD_A_DATA_E_1	Red_1	Red_1	Red_9	not used	not used	not used
FPD_A_DATA_E_0	Red_0	Red_0	Red_8	not used	not used	not used
FPD Bus B						
FPD Bus B is unused in Modes 1 through 8						

Mode 7 (AM8280 Mode 9-1)				
FPD Bus A - Data_A Channel			**FPD Bus B - Data_A Channel**	
FPD_A_DATA_A_6	Odd_Green_0		FPD_B_DATA_A_6	Even_Green_0
FPD_A_DATA_A_5	Odd_Red_5		FPD_B_DATA_A_5	Even_Red_5
FPD_A_DATA_A_4	Odd_Red_4		FPD_B_DATA_A_4	Even_Red_4
FPD_A_DATA_A_3	Odd_Red_3		FPD_B_DATA_A_3	Even_Red_3
FPD_A_DATA_A_2	Odd_Red_2		FPD_B_DATA_A_2	Even_Red_2
FPD_A_DATA_A_1	Odd_Red_1		FPD_B_DATA_A_1	Even_Red_1
FPD_A_DATA_A_0	Odd_Red_0		FPD_B_DATA_A_0	Even_Red_0
FPD Bus A - Data_B Channel			**FPD Bus B - Data_B Channel**	
FPD_A_DATA_B_6	Odd_Blue_1		FPD_B_DATA_B_6	Even_Blue_1
FPD_A_DATA_B_5	Odd_Blue_0		FPD_B_DATA_B_5	Even_Blue_0
FPD_A_DATA_B_4	Odd_Green_5		FPD_B_DATA_B_4	Even_Green_5
FPD_A_DATA_B_3	Odd_Green_4		FPD_B_DATA_B_3	Even_Green_4
FPD_A_DATA_B_2	Odd_Green_3		FPD_B_DATA_B_2	Even_Green_3
FPD_A_DATA_B_1	Odd_Green_2		FPD_B_DATA_B_1	Even_Green_2
FPD_A_DATA_B_0	Odd_Green_1		FPD_B_DATA_B_0	Even_Green_1
FPD Bus A - Data_C Channel			**FPD Bus B - Data_C Channel**	
FPD_A_DATA_C_6	DEN		FPD_B_DATA_C_6	DEN
FPD_A_DATA_C_5	VSYNC		FPD_B_DATA_C_5	VSYNC
FPD_A_DATA_C_4	HSYNC		FPD_B_DATA_C_4	HSYNC
FPD_A_DATA_C_3	Odd_Blue_5		FPD_B_DATA_C_3	Even_Blue_5
FPD_A_DATA_C_2	Odd_Blue_4		FPD_B_DATA_C_2	Even_Blue_4
FPD_A_DATA_C_1	Odd_Blue_3		FPD_B_DATA_C_1	Even_Blue_3
FPD_A_DATA_C_0	Odd_Blue_2		FPD_B_DATA_C_0	Even_Blue_2
FPD Bus A - Data_D Channel			**FPD Bus B - Data_D Channel**	
FPD_A_DATA_D_6	Map to Field		FPD_B_DATA_D_6	Map to Field
FPD_A_DATA_D_5	Odd_Blue_7		FPD_B_DATA_D_5	Even_Blue_7
FPD_A_DATA_D_4	Odd_Blue_6		FPD_B_DATA_D_4	Even_Blue_6
FPD_A_DATA_D_3	Odd_Green_7		FPD_B_DATA_D_3	Even_Green_7
FPD_A_DATA_D_2	Odd_Green_6		FPD_B_DATA_D_2	Even_Green_6
FPD_A_DATA_D_1	Odd_Red_7		FPD_B_DATA_D_1	Even_Red_7
FPD_A_DATA_D_0	Odd_Red_6		FPD_B_DATA_D_0	Even_Red_6
FPD Bus A - Data_E Channel			**FPD Bus B - Data_E Channel**	
FPD_A_DATA_E_6	not used		FPD_B_DATA_E_6	not used
FPD_A_DATA_E_5	not used		FPD_B_DATA_E_5	not used
FPD_A_DATA_E_4	not used		FPD_B_DATA_E_4	not used
FPD_A_DATA_E_3	not used		FPD_B_DATA_E_3	not used
FPD_A_DATA_E_2	not used		FPD_B_DATA_E_2	not used
FPD_A_DATA_E_1	not used		FPD_B_DATA_E_1	not used
FPD_A_DATA_E_0	not used		FPD_B_DATA_E_0	not used

Mode 8 (AM8280 Mode 9-2)				
FPD Bus A - Data_A Channel			**FPD Bus B - Data_A Channel**	
FPD_A_DATA_A_6	Odd_Green_2		FPD_B_DATA_A_6	Even_Green_2
FPD_A_DATA_A_5	Odd_Red_7		FPD_B_DATA_A_5	Even_Red_7
FPD_A_DATA_A_4	Odd_Red_6		FPD_B_DATA_A_4	Even_Red_6
FPD_A_DATA_A_3	Odd_Red_5		FPD_B_DATA_A_3	Even_Red_5
FPD_A_DATA_A_2	Odd_Red_4		FPD_B_DATA_A_2	Even_Red_4
FPD_A_DATA_A_1	Odd_Red_3		FPD_B_DATA_A_1	Even_Red_3
FPD_A_DATA_A_0	Odd_Red_2		FPD_B_DATA_A_0	Even_Red_2
FPD Bus A - Data_B Channel			**FPD Bus B - Data_B Channel**	
FPD_A_DATA_B_6	Odd_Blue_3		FPD_B_DATA_B_6	Even_Blue_3
FPD_A_DATA_B_5	Odd_Blue_2		FPD_B_DATA_B_5	Even_Blue_2
FPD_A_DATA_B_4	Odd_Green_7		FPD_B_DATA_B_4	Even_Green_7
FPD_A_DATA_B_3	Odd_Green_6		FPD_B_DATA_B_3	Even_Green_6
FPD_A_DATA_B_2	Odd_Green_5		FPD_B_DATA_B_2	Even_Green_5
FPD_A_DATA_B_1	Odd_Green_4		FPD_B_DATA_B_1	Even_Green_4
FPD_A_DATA_B_0	Odd_Green_3		FPD_B_DATA_B_0	Even_Green_3
FPD Bus A - Data_C Channel			**FPD Bus B - Data_C Channel**	
FPD_A_DATA_C_6	DEN		FPD_B_DATA_C_6	DEN
FPD_A_DATA_C_5	VSYNC		FPD_B_DATA_C_5	VSYNC
FPD_A_DATA_C_4	HSYNC		FPD_B_DATA_C_4	HSYNC
FPD_A_DATA_C_3	Odd_Blue_7		FPD_B_DATA_C_3	Even_Blue_7
FPD_A_DATA_C_2	Odd_Blue_6		FPD_B_DATA_C_2	Even_Blue_6
FPD_A_DATA_C_1	Odd_Blue_5		FPD_B_DATA_C_1	Even_Blue_5
FPD_A_DATA_C_0	Odd_Blue_4		FPD_B_DATA_C_0	Even_Blue_4
FPD Bus A - Data_D Channel			**FPD Bus B - Data_D Channel**	
FPD_A_DATA_D_6	Map to Field		FPD_B_DATA_D_6	Map to Field
FPD_A_DATA_D_5	Odd_Blue_1		FPD_B_DATA_D_5	Even_Blue_1
FPD_A_DATA_D_4	Odd_Blue_0		FPD_B_DATA_D_4	Even_Blue_0
FPD_A_DATA_D_3	Odd_Green_1		FPD_B_DATA_D_3	Even_Green_1
FPD_A_DATA_D_2	Odd_Green_0		FPD_B_DATA_D_2	Even_Green_0
FPD_A_DATA_D_1	Odd_Red_1		FPD_B_DATA_D_1	Even_Red_1
FPD_A_DATA_D_0	Odd_Red_0		FPD_B_DATA_D_0	Even_Red_0
FPD Bus A - Data_E Channel			**FPD Bus B - Data_E Channel**	
FPD_A_DATA_E_6	not used		FPD_B_DATA_E_6	not used
FPD_A_DATA_E_5	not used		FPD_B_DATA_E_5	not used
FPD_A_DATA_E_4	not used		FPD_B_DATA_E_4	not used
FPD_A_DATA_E_3	not used		FPD_B_DATA_E_3	not used
FPD_A_DATA_E_2	not used		FPD_B_DATA_E_2	not used
FPD_A_DATA_E_1	not used		FPD_B_DATA_E_1	not used
FPD_A_DATA_E_0	not used		FPD_B_DATA_E_0	not used

3.1.26 Read FPD Link Pixel Map Mode (4Ch)

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

Table 3-41. Return Parameters

Parameter Bytes	Description
Byte 0	Reserved
Byte 1	Reserved
Byte 2	See below

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

Table 3-42. Read FPD Link Pixel Map Mode 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

3.1.27 Write FPGA Input Video Chroma Processing Select (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 3-43. Write FPD Input Video Chroma Processing Select Register Field Descriptions

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

3.1.28 Read FPGA Input Video Chroma Processing Select (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 3-44. Read FPGA Input Video Chroma Processing Select Register Field Descriptions

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

3.1.29 Write LED Output Control Method (50h)

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 3-45. Write LED Output Control Method 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 3-46.

Table 3-46. 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)

The Manual RGB LED Currents method provides for manual control of the LED currents, and as such, the CAIC algorithm (Section 3.1.62) is disabled.

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

3.1.30 Read LED Output Control Method (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 3-47. Read LED Output Control Method 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

3.1.31 Write RGB LED Enable (52h)

This command enables the LEDs for the display module.

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

Table 3-48. Write RGB LED Enable 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

3.1.32 Read RGB LED Enable (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 3-49. Read RGB LED Enable 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

3.1.33 Write RGB LED Current (54h)

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

Table 3-50. Write Parameters

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

When an all white image is 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.

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

When the CAIC algorithm is disabled, this command directly sets the LED currents (that is, the R, G, and B values provided are sent directly to the PAD 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) allow currents within ±4 PAD 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 since 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 since 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.

During low battery operation, the DLPC3436 reverts its RGB LED current settings automatically to their default boot-up value. This feature is only available in select system configurations.

3.1.34 Read RGB LED Current (55h)

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

Table 3-51. Return Parameters

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

See *Section 3.1.33* for a detailed description of the return parameters.

Unused most significant bits are set to 0.

3.1.35 Read CAIC LED Max Available Power (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 3-52. Return Parameters

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

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

This command is only applicable when CAIC is enabled.

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

3.1.36 Write RGB LED Max Current (5Ch)

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

Table 3-53. 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)

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

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.

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

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

3.1.37 Read RGB LED Max Current (5Dh)

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

Table 3-54. Return Parameters

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

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

Unused most significant bits is set to ‘0’.

3.1.38 Read Measured LED Parameters (5Eh)

This command is used to read the measured values for a number of LED based parameters in the display module.

Table 3-55. Return Parameters(1)(2)(3)

Parameter Bytes	Description
Byte 1	Measured Red LED Current (LSByte)
Byte 2	Measured Red LED Current (MSByte)
Byte 3	Measured Green LED current (LSByte)
Byte 4	Measured Green LED Current (MSByte)
Byte 5	Measured Blue LED current (LSByte)
Byte 6	Measured Blue LED current (MSByte)
Byte 7	Measured Red LED Voltage (LSByte)
Byte 8	Measured Red LED Voltage (MSByte)
Byte 9	Measured Green LED Voltage (LSByte)
Byte 10	Measured Green LED Voltage (MSByte)
Byte 12	Measured Blue LED Voltage (LSByte)
Byte 13	Measured Blue LED Voltage (MSByte)
Byte 14	Measured Red LED Power (LSByte)
Byte 15	Measured Red LED Power (MSByte)
Byte 16	Measured Green LED Power (LSByte)
Byte 17	Measured Green LED Power (MSByte)
Byte 18	Measured Blue LED Power (LSByte)
Byte 19	Total LED Power (LSByte)
Byte 20	Total LED Power (MSByte)

(1) Current is be specified as Milliamps × 2, with the maximum value = 32767.5 mA. (Example: 1287.5 mA = 0A0Fh).
(2) Voltage is specified as Voltage × 1700, with the maximum value = 38.550 V. (Example: 1.548 mA = 0A48h).
(3) Power is specified as Watts × 325, with the maximum value = 201.64 W. (Example: 7.923 W = A0Fh).

3.1.39 Read CAIC RGB LED Current (5Fh)

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

Table 3-56. Return Parameters(1)(2)(3)

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)

(1) The parameters returned by this command have a resolution of 10 bits, and are defined by the appropriate PAD specification.
(2) Use of this command is only appropriate when the LED Output Control Method is set to CAIC (Automatic) RGB LED Current Control.
(3) Unused most significant bits are set to '0'.

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 match (but may not exactly match) those specified using the Write RGB LED Current command. For an all white image, this command provides values within ±4 PAD device current steps for each LED color relative to those specified with the Write RGB LED Current command.

3.1.40 Write XPR FPGA Input Image Size (60h)

This command is used to specify the active data size of the external input image that goes to the XPR FPGA. Resolutions supported by the system are 1920×1080 and 960×540.

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)

3.1.41 Read XPR FPGA Input Image Size (61h)

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

3.1.42 Read XPR FPGA Version (64h)

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

Parameter Bytes	Description
Byte 4:1	• 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
Byte 5	b(7:0) = FPGA ECO Revision
Byte 6	• b(7:0) = FPGA ARM Software Version - Major

3.1.43 Write XPR FPGA Test Pattern Select (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	TPG pattern options (MSByte)

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

Table 3-57. Byte 1 Write XPR FPGA Test Pattern Select 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 = Color Bars Bh = Frame & Cross Ch - Fh = Reserved

Byte 2: Varies depending on configuration selected. It is ignored for Solid Field, Grids, Horizontal, Vertical and Diagonal lines, 3D test patterns and Color Bars.

For Horizontal and Vertical Ramps, Byte 2 represents the pixel level intensity at the brightest part of the ramp ranging from 0-255.

For Checkerboards, Byte 2 specifies the size of each checker in 4 pixel resolution so a value of 10 would generate 40 pixel checkers.

For Actuator Calibration test patterns, Byte 2 specifies the sub-frames to be displayed:

- Byte 2 (7:0) = 0: Actuator Calibration Pattern - Sub-Frame 0 and 1 Only (HD only)
- Byte 2 (7:0) = 3: Actuator Calibration Pattern - Sub-Frames 0, 1, 2, 3 (Full HD only)

For Frame & Cross, Byte 2 is divided into two nibbles. Each nibble is a pixel position from the upper left corner of the image with a resolution of (Pixel Count / 16). So 720p has 80 pixel increments horizontally and 45 pixel increments vertically. 1080p has 120 pixel increments horizontally and 68 pixel increments vertically.

- MS-Nibble (7:4): Horizontal position
- LS-Nibble (3:0): Vertical position

3.1.44 Read XPR FPGA Test Pattern Select (68h)

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

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

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

Table 3-58. Read XPR FPGA Test Pattern Select 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 = Color Bars Bh = Frame & Cross Ch - Fh = Reserved

Byte 2: Varies depending on configuration selected. It is ignored for Solid Field, Grids, Horizontal, Vertical and Diagonal lines, 3D test patterns and Color Bars.

For Horizontal and Vertical Ramps, Byte 2 represents the pixel level intensity at the brightest part of the ramp ranging from 0-255.

For Checkerboards, Byte 2 specifies the size of each checker in 4 pixel resolution so a value of 10 would generate 40 pixel checkers.

For Actuator Calibration test patterns, Byte 2 specifies the sub-frame(s) to be displayed:

- Byte 2 (7:0) = 0: Actuator Calibration Pattern - Sub-Frame 0 and 1 Only (HD only)
- Byte 2 (7:0) = 3: Actuator Calibration Pattern - Sub-Frames 0, 1, 2, 3 (Full HD only)

For Frame & Cross, Byte 2 is divided into two nibbles. Each nibble is a pixel position from the upper left corner of the image with a resolution of (Pixel Count / 16). So 720p has 80 pixel increments horizontally and 45 pixel increments vertically. 1080p has 120 pixel increments horizontally and 68 pixel increments vertically.

- MS-Nibble (7:4): Horizontal position

- LS-Nibble (3:0): Vertical position

3.1.45 Write XPR FPGA Parallel Video Control (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 3-59. Write XPR FPGA Parallel Video Control 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

3.1.46 Read XPR FPGA Parallel Video Control (6Ch)

This command is used to read XPR FPGA video format.

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

Table 3-60. Read XPR FPGA Parallel Video Control 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

3.1.47 Write XPR FPGA Video Format Select (6Dh)

This command is used to specify XPR FPGA video format.

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

Table 3-61. Write XPR FPGA Video Format Select 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

When adjusting the XPR FPGA Video Format selection, the parallel video data input must be properly aligned with the 24-bit bus of the FPGA. The appropriate data encoding format is provided in Figure 3-4.

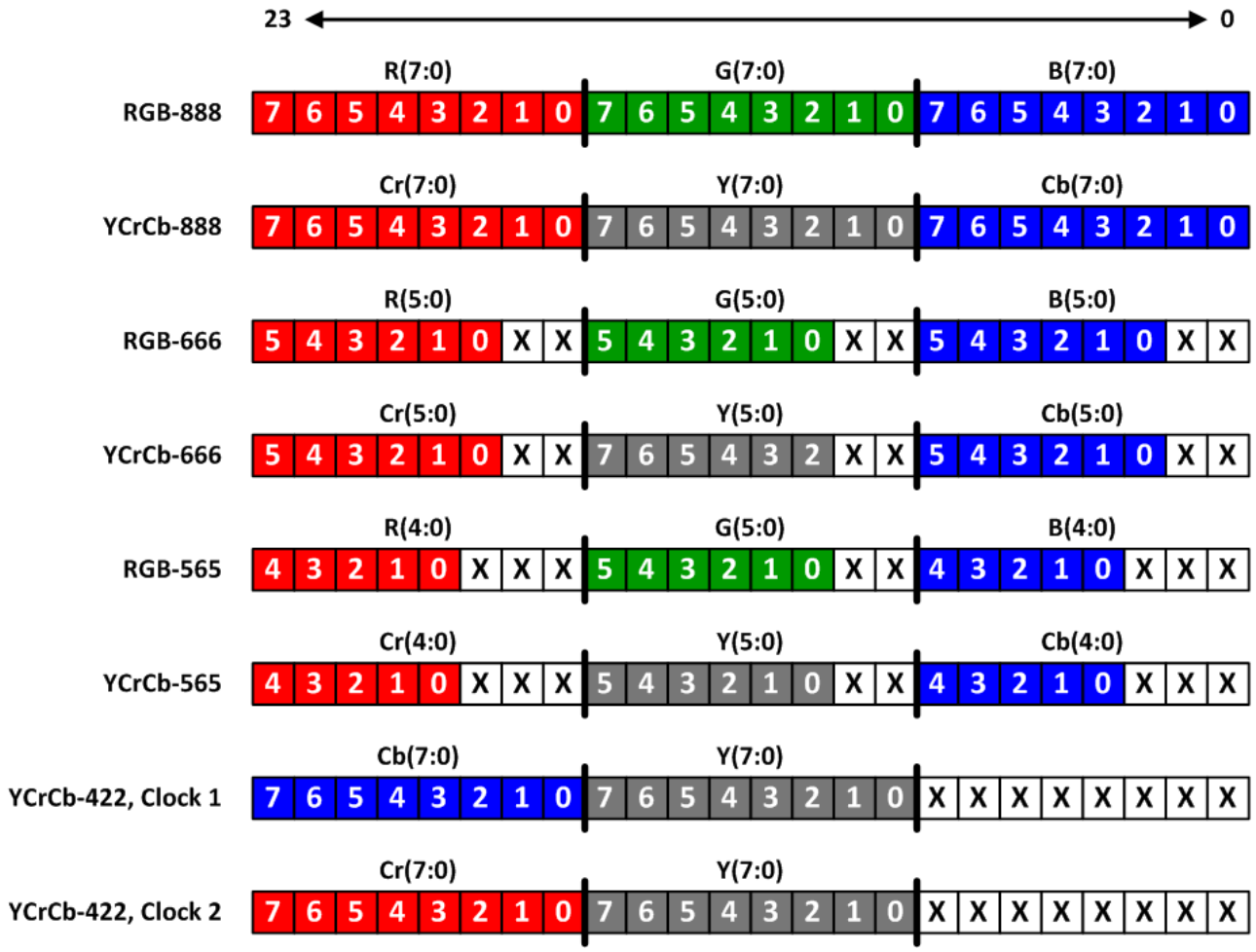

Figure 3-4. Parallel Data Bus (23:0) Encoding Options

3.1.48 Read XPR FPGA Video Format Select (6Eh)

This command is used to read XPR FPGA video format.

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

Table 3-62. Read XPR FPGA Video Format Select 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

3.1.49 Read XPR FPGA Status (6Fh)

This command is used to read XPR FPGA status.

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

Table 3-63. Read XPR FPGA Status 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

3.1.50 Write Actuator Latency (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.

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

NOTE: Valid latency input values range from 000h to 3FFh, with a step size of 133.333 ns. This yields a minimum latency of 0 ns and a maximum latency of 34952312.619 ns.

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

Table 3-64. Write Actuator Latency Register Field Descriptions

Bit	Type	Description
7-2	R	Reserved
1	W	Latency Auto Scaling Enable 0h = Auto scaling disabled -- No clock dropping scaling is applied 1h = Auto scaling enabled
0	W	Manual Latency Enable 0h = Manual Latency disabled – Latency value stored in the sequence header is used. 1h = Latency value iused provided in Byte 1-3.

3.1.51 Read Actuator Latency (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 3-65. Read Actuator Latency Register Field Descriptions

Bit	Type	Description
7-2	R	Reserved
1	W	Latency Auto Scaling Enable 0h = Auto scaling disabled -- No clock dropping scaling is applied 1h = Auto scaling enabled
0	W	Manual Latency Enable 0h = Manual Latency disabled – Latency value stored in the sequence header will be used. 1h = Latency value used provided in Byte 1-3.

3.1.52 Write Actuator Gain (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)

3.1.53 Read Actuator Gain (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)

3.1.54 Write Segment Length (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.

3.1.55 Read Segment Length (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.

3.1.56 Write Manual Actuator Sync Delay (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 3-66. Write Manual Actuator Sync Delay 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 is not 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 is 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.

3.1.57 Read Manual Actuator Sync Delay (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 3-67. Read Manual Actuator Sync Delay 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 is not 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 is 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.

3.1.58 Write Manual Actuator 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)
Byte 3	Reserved
Byte 4	See Below

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

Table 3-68. Byte 4 Write Manual Actuator Offset Register Field Descriptions

Bit	Type	Description
7-1	R	Reserved
0	W	Auto DC Offset Enable 0h = Auto DC offset disabled 1h = Auto DC offset enabled

NOTE: This Actuator Manual Offset is presented in 16-bit signed 9.7 format (01h = 00.0078130)

Valid values of Actuator Manual Offset range from -255 to +255.

The sum of the Auto-DC offset and the Manual offset cannot be less than -255 or greater than +255.

3.1.59 Read Manual Actuator 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)
Byte 3	Reserved
Byte 4	See Below

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

Table 3-69. Byte 4 Read Manual Actuator Offset Register Field Descriptions

Bit	Type	Description
7-1	R	Reserved
0	W	Auto DC Offset Enable 0h = Auto DC offset disabled 1h = Auto DC offset enabled

NOTE: This Actuator Manual Offset is presented in 16-bit signed 9.7 format (01h = 00.0078130)

Valid values of Actuator Manual Offset range from -256 to 256.

3.1.60 Write Local Area Brightness Boost Control (80h)

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

Table 3-70. 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 3-71. Write Local Area Brightness Boost Control Register Field Descriptions (1)(2)(3)(4)(5)

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

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

(2) For automatic strength control, a light sensor is 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.

(3) For LABB Strength, 0 indicates no boost applied, and 255 indicates the maximum boost that is considered viable in a product. The strength is not a direct indication of the gain because the gain varies depending on image content.

(4) 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.

(5) LABB is supported in TPG, Splash, External Input mode, but auto-disabled in curtain mode.

3.1.61 Read Local Area Brightness Boost Control (81h)

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

Figure 3-5. 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 3-72. Read Local Area Brightness Boost Control 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 3-6 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 3-6. 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)
```

3.1.62 Write CAIC Image Processing Control (84h)

This command controls the CAIC functionality for the display module.

Table 3-73. 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 3-74. Write CAIC Image Processing Control Register Field Descriptions[(1)(2)(3)(4)]

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

(1) 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 provides automatic white point correction.

(2) 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.

(3) 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).

(4) Figure 3-7 shows the bit order and weighting for the CAIC Maximum Lumens Gain value, which has a valid range from 1.0 to 4.0. The device considers values outside of this range as an error (invalid command parameter value – communication status) anddoes not execute the command.

Figure 3-7. 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 typical case, power holds constant for most images, while the lumens gain. In the rare case where the gain exceeds 4.0, lumens stop increasing and the power reduces.

.

Figure 3-8 shows the bit order and weighting for the CAIC Clipping Threshold value, which has a valid range from 0.0% to 2.0%. The device considers values outside of this range as an error (invalid command parameter value – communication status) and the device does not execute this command.

Figure 3-8. Bit Weight Definition for the CAIC Clipping Threshold Value(1)(2)

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

(1) 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.

(2) Figure 3-9 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. The device considers values outside of this range as an error (invalid command parameter value – communication status) and the device does not execute the command.

Figure 3-9. 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 3-75. LABB and CAIC Modes

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

3.1.63 Read CAIC Image Processing Control (85h)

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

Table 3-76. 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 3-77. Read CAIC Image Processing Control 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 3.1.62.

3.1.64 Write Color Coordinate Adjustment Control (86h)

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 3-78. Write Color Coordinate Adjustment Control 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 must remain enabled during normal operation. When CCA is disabled, use an identity matrix.

3.1.65 Read Color Coordinate Adjustment Control (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 3-79. Read Color Coordinate Adjustment Control Register Field Descriptions

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

3.1.66 Write Keystone Correction Control (88h)

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

Table 3-80. 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 3-81. Write Keystone Correction Control Register Field Descriptions

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

NOTE: Refer to Table 3-102 for valid range of Keystone Control Parameters.

3.1.67 Read Keystone Correction Control (89h)

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

Table 3-82. 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 3-83. Read Keystone Correction Control Register Field Descriptions

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

3.1.68 Write Actuator Number of Segments (A0h)

Defines the number of steps (or levels) in the actuator waveform.

Table 3-84. Write Parameters

Parameter Bytes	Description
Byte 1	Number of Segments (Range is 2 to 255)

This command is applied to the default voice coil or the one most recently selected via the *Write Actuator Configuration Select* command.

The command is programmed when transmitted and therefore applies to any waveform programmed until this command is sent again.

3.1.69 Read Actuator Number of Segments (A1h)

Returns the number of steps (levels) in the actuator waveform as specified by default or by the most recent *Write Actuator Number of Segments* command.

Table 3-85. Read Parameters

Parameter Bytes	Description
Byte 1	Number of Segments (Range is 2 to 255)

This command is associated with the default voice coil or the one most recently selected via the *Write Actuator Configuration Select* command.

3.1.70 Write Actuator Configuration Select (A2h)

Specifies which voice coil and which tilt axis orientation are to be configured by subsequent commands.

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

Table 3-86. Write Actuator Configuration Select Register Field Descriptions

Bit	Type	Description
7-4	R	Reserved
3-2	W	Tilt Axis Orientation 0h = Axis 1 1h = Axis 2 2h = Axis 3
1-0	W	Voice Coil 0h = Voice Coil A 1h = Voice Coil B

NOTE: This command simply stores the selected actuator configuration information in global data. When a subsequent command arrives that depends on an axis and / or voice coil selection, this global data is used to configure the appropriate actuator component(s).

3.1.71 Read Actuator Configuration Select (A3h)

Specifies which voice coil and which tilt axis orientation are to be configured by subsequent commands.

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

Table 3-87. Read Actuator Configuration Select Register Field Descriptions

Bit	Type	Description
7-4	R	Reserved
3-2	W	Tilt Axis Orientation 0h = Axis 1 1h = Axis 2 2h = Axis 3
1-0	W	Voice Coil 0h = Voice Coil A 1h = Voice Coil B

3.1.72 Write Actuator Fixed Level Value (A4h)

Specifies the fixed value to be output by the actuator waveform generator.

Table 3-88. Write Parameters

Parameter Bytes	Description
Byte 1	Fixed Level Value (Range is 0 to 255)

This command is applied to the default voice coil or the one most recently selected via the *Write Actuator Configuration Select* command.

The command is programmed when transmitted and therefore applies to any waveform programmed until this command is sent again.

3.1.73 Read Actuator Fixed Level Value(A5h)

Returns the fixed value to be output by the actuator waveform generator.

Table 3-89. Read Parameters

Parameter Bytes	Description
Byte 1	Fixed Level Value (Range is 0 to 255)

This command is associated with the default voice coil or the one most recently selected via the *Write Actuator Configuration Select* command.

3.1.74 Write Actuator Period Stretch Value(A6h)

Defines the high and low time for the Waveform Generator DAC clock output.

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

Table 3-90. Write Actuator Period Stretch Value Register Field Descriptions

Bit	Type	Description
7-3	R	Reserved
2-0	W	Clock Stretch Value (Range is 0 to 7). The clock period equals: 2 × (Clock Stretch Value+1).

The DAC Clock Generator function generates a 50% duty cycle clock signal for driving the external digital to analog converter (DAC). The LSB of this configuration parameter represents the DAC input clock period so a DAC output clock period of 2 (high+low) to 16 times the DAC input clock period is supported.

This command is applied to the default axis or the one most recently selected via the *Write Actuator Configuration Select* command.

3.1.75 Read Actuator Period Stretch Value(A7h)

Returns the high and low time for the Waveform Generator DAC clock output as specified by default or by the most recent *Write Actuator Period Stretch* command.

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

Table 3-91. Read Actuator Period Stretch Value Register Field Descriptions

Bit	Type	Description
7-3	R	Reserved
2-0	W	Clock Stretch Value (Range is 0 to 7). The clock period equals: 2 × (Clock Stretch Value+1).

This return value is associated with the default voice coil or the one most recently selected via the *Write Actuator Configuration Select* command.

3.1.76 Write Actuator Reference Value (A8h)

Specifies the Reference DAC fixed output value.

Table 3-92. Write Parameters

Parameter Bytes	Description
Byte 1	Reference Value (Range is 0 to 255)

3.1.77 Read Actuator Reference Value (A9h)

Returns the Reference DAC fixed output value.

Table 3-93. Read Parameters

Parameter Bytes	Description
Byte 1	Reference Value (Range is 0 to 255)

3.1.78 Write Actuator Output Select (AAh)

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

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

Table 3-94. Write Actuator Fixed Output 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)

3.1.79 Read Actuator Output Select (ABh)

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

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

Table 3-95. Read Actuator Fixed Output 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)

3.1.80 Write Actuator Edge Table Address Mode (ACh)

This command is used to specify the Actuator Waveform Address Mode parameter.

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

Table 3-96. Write Actuator Direction Register Field Descriptions

Bit	Type	Description
7-2	R	Reserved
1-0	W	Actuator Edge Table Address Mode 0h = Read Down & Read Up 1h = Read Up Inverted & Read Up 2h = Read Up & Read Down 3h = Read Up & Read Up Inverted

3.1.81 Read Actuator Edge Table Address Mode (ADh)

This command is used to read the Actuator Waveform Address Mode parameter.

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

Table 3-97. Read Actuator Direction Register Field Descriptions

Bit	Type	Description
7-2	R	Reserved
1-0	W	Actuator Edge Table Address Mode 0h = Read Down & Read Up 1h = Read Up Inverted & Read Up 2h = Read Up & Read Down 3h = Read Up & Read Up Inverted

3.1.82 Write Actuator DAC Enable (AEh)

This command is used to specify the Actuator DAC Enable parameter.

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

Table 3-98. Write Actuator Enable Register Field Descriptions

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

3.1.83 Read Actuator DAC Enable (AFh)

This command is used to read the Actuator DAC Enable parameter.

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

Table 3-99. Read Enable Register Field Descriptions

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

3.1.84 Read Auto Framing Information (BAh)

This command is used to read the VSYNC rate, total pixel, and total lines for the displayed image.

Table 3-100. Read Auto Framing Information Fields

Parameter Bytes	Description
Byte 1	Read VSYNC Rate (LSByte)
Byte 2	Read VSYNC Rate
Byte 3	Read VSYNC Rate
Byte 4	Read VSYNC Rate (MSByte)
Byte 5	Read Total Pixels Per Line (LSByte)
Byte 6	Read Total Pixels Per Line (MSByte)
Byte 7	Read Total Lines Per Frame (LSByte)
Byte 8	Read Total Lines Per Frame (MSByte)
Byte 9	Read Active Pixels Per Line (LSByte)
Byte 10	Read Active Pixels Per Line (MSByte)
Byte 12	Read Active Lines Per Frame (LSByte)
Byte 13	Read Active Lines Per Frame (MSByte)
Byte 14	Read Reference Clock Rate (LSByte)
Byte 15	Read Reference Clock Rate (MSByte)

3.1.85 Write Keystone Projection Pitch Angle (BBh)

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

Table 3-101. 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 Table 3-102 for valid range of keystone control parameters. Figure 3-11 shows examples of the projection pitch angle.

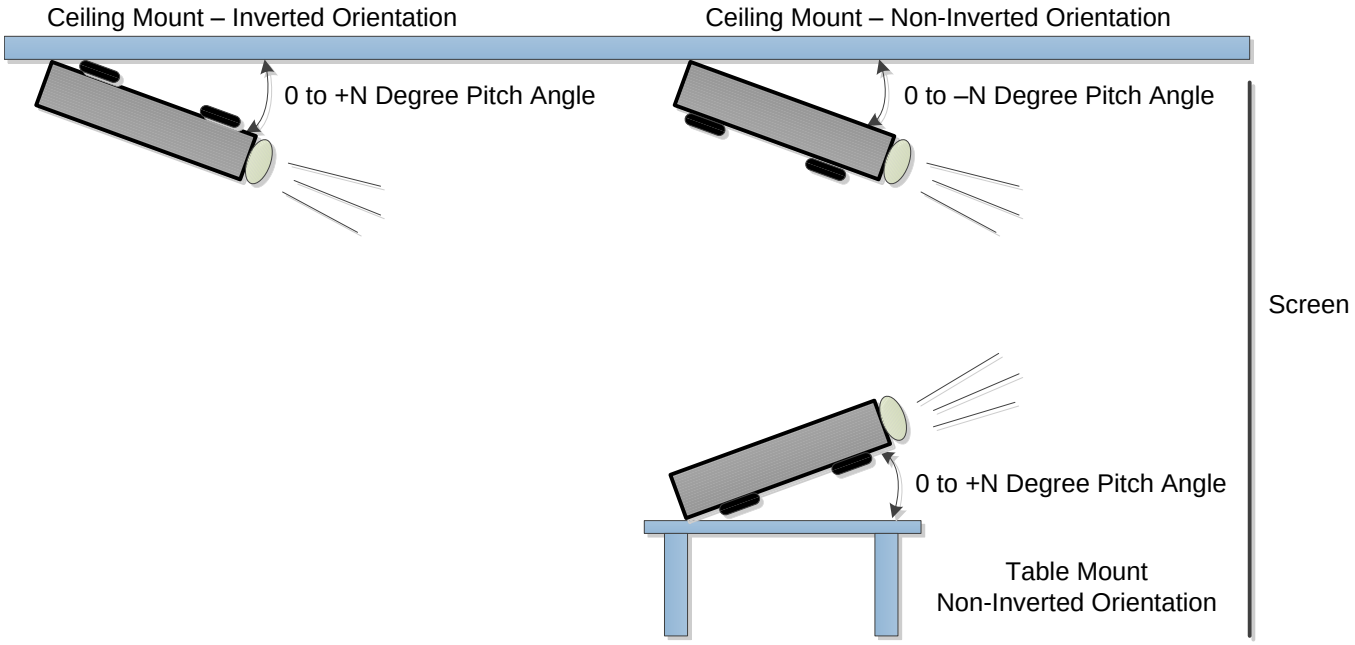

Figure 3-11. Examples of Projection Pitch Angle

Table 3-102. Keystone Parameters Supported Range

Parameter	Solution Space 1	Solution Space 2	Solution Space 3	Solution Space 4
Throw Ratio	0.2 to 0.3	0.3 to 0.4	0.4 to 0.6	0.6 to 2.0
Pitch (degrees)	-29 to +14 degrees	-29 to +20 degrees	-39 to +27 degrees	-39 to +36 degrees
Offset	0% to 150%	0% to 150%	0% to 150%	0% to 150%

3.1.86 Read Keystone Projection Pitch Angle (BCh)

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

Table 3-103. Return Parameters

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

3.1.87 Write Actuator Watchdog Window Width (C2h)

This command specifies the amount of time that must pass before edges of the INT_SUBFRAME signal, which are artificially inserted ensuring the edges never stop completely. The input parameters are applied separately to the axes defined by AWG_12 and AWG_34.

Table 3-104. Write Actuator Watchdog Window Width - Byte 0 and 1

Bit	Type	Description
15-10	W	Reserved
9-0	W	Watchdog window width value (LSB = 100 us) applied to AWG_12

Table 3-105. Write Actuator Watchdog Window Width - Byte 2 and 3

Bit	Type	Description
15-10	W	Reserved
9-0	W	Watchdog window width value (LSB = 100 us) applied to AWG_34

When either of the AWG values equals zero, the device disables the auto-calculation for Actuator Subframe Filter Width and Actuator Watchdog Window Width. If a system error is encountered as a result of incorrectly configuring either of these values, ensure all fields are nonzero to resume auto-calculation mode. Flash download cannot be initiated while this system error is present.

3.1.88 Read Actuator Watchdog Window Width (C3h)

This command returns the value for the watchdog window width for each axis (AWG_12 and AWG_34).

Table 3-106. Read Actuator Watchdog Window Width - Byte 0 and 1

Bit	Type	Description
15-10	R	Reserved
9-0	R	Watchdog window width value (LSB = 100 us) applied to AWG_12

Table 3-107. Read Actuator Watchdog Window Width - Byte 2 and 3

Bit	Type	Description
15-10	R	Reserved
9-0	R	Watchdog window width value (LSB = 100 us) applied to AWG_34

NOTE: When either of the AWG values equals zero, the device disables the auto-calculation for Actuator Subframe Filter Width and Actuator Watchdog Window Width. If a system error is encountered as a result of incorrectly configuring either of these values, ensure all fields are nonzero to resume auto-calculation mode. Flash download cannot be initiated while this system error is present.

3.1.89 Write Actuator Subframe Filter Width (C4h)

This command specifies the amount of time that must pass before edges of the INT_SUBFRAME signal, which are allowed to toggle. This function suppresses an edge that occurs too close to the previous edge. The input parameters are applied separately to the axes defined by AWG_12 and AWG_34.

Table 3-108. Write Actuator Subframe Filter Width - Byte 0 and 1

Bit	Type	Description
15-10	W	Reserved
9-0	W	Subframe filter width value (LSB = 100 us) applied to AWG_12

Table 3-109. Write Actuator Subframe Filter Width - Byte 2 and 3

Bit	Type	Description
15-10	W	Reserved
9-0	W	Subframe filter width value (LSB = 100 us) applied to AWG_34

When either of the AWG values equals zero, the device disables the auto-calculation for Actuator Subframe Filter Width and Actuator Watchdog Window Width. If a system error is encountered as a result of incorrectly configuring either of these values, ensure all fields are nonzero to resume auto-calculation mode. Flash download cannot be initiated while this system error is present.

3.1.90 Read Actuator Subframe Filter Width (C5h)

This command returns the value of the subframe filter width for each axis (AWG_12 and AWG_34).

Table 3-110. Read Actuator Subframe Filter Width - Byte 0 and 1

Bit	Type	Description
15-10	R	Reserved
9-0	R	Subframe filter width value (LSB = 100 us) applied to AWG_12

Table 3-111. Read Actuator Subframe Filter Width - Byte 2 and 3

Bit	Type	Description
15-10	R	Reserved
9-0	R	Subframe filter width value (LSB = 100 us) applied to AWG_34

When either of the AWG values equals zero, the device disables the auto-calculation for Actuator Subframe Filter Width and Actuator Watchdog Window Width. If a system error is encountered as a result of incorrectly configuring either of these values, ensure all fields are nonzero to resume auto-calculation mode. Flash download cannot be initiated while this system error is present.

3.1.91 Write Actuator Stepped/Fixed Output Invert Enable (C6h)

This command inverts the value of the stepped/fixed output (2's complement).

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

Table 3-112. Write Actuator Stepped Output Invert Field Descriptions

Bit	Type	Description
7-2	R	Reserved
1	W	Actuator Stepped Output Invert Enable 0h = Disable (Stepped output is not inverted) 1h = Enable (Stepped output is inverted)
0	W	Actuator Fixed Output Invert Enable 0h = Disable (Fixed output is not inverted) 1h = Enable (Fixed output is inverted)

3.1.92 Read Actuator Stepped/Fixed Output Invert Enable (C7h)

This command is used to read the state of the stepped/fixed output invert function for the default coil or the coil most recently selected via the Write Actuator Configuration (A2h).

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

Table 3-113. Read Actuator Stepped Output Invert Field Descriptions

Bit	Type	Description
7-2	R	Reserved
1	R	Actuator Stepped Output Invert Enable 0h = Disable (Stepped output is not inverted) 1h = Enable (Stepped output is inverted)
0	R	Actuator Fixed Output Invert Enable 0h = Disable (Fixed output is not inverted) 1h = Enable (Fixed output is inverted)

3.1.93 Write Actuator Orientation (C8h)

This command specifies the actuator orientation value. The command input value represents 1 of the 24 possible combinations in which 4 subframes can be ordered. Internally, the input is further modified by the current image orientation. Therefore, the input maps to 4 different subframe orderings.

Parameter Bytes	Description
Byte 1	Actuator Subframe - Decode
Byte 2	Actuator Subframe - No Flip
Byte 3	Actuator Subframe - Horizontal Flip
Byte 4	Actuator Subframe - Vertical Flip
Byte 5	Actuator Subframe - Horizontal/Vertical Flip

NOTE: Value ranges from 0 to 23 for each byte.

3.1.94 Read Actuator Orientation (C9h)

This command is used to read the specified actuator orientation value.

Parameter Bytes	Description
Byte 1	Actuator Subframe - Decode
Byte 2	Actuator Subframe - No Flip
Byte 3	Actuator Subframe - Horizontal Flip
Byte 4	Actuator Subframe - Vertical Flip
Byte 5	Actuator Subframe - Horizontal/Vertical Flip

NOTE: Value ranges from 0 to 23 for each byte.

3.1.95 Read Short Status (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 3-114. Read Short Status 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

The flash erase complete status bit is set at the start of the flash erase process and is 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, the device holds it without processing until the flash erase has completed (thus blocking any following status requests until the previously sent command is processed).

The flash error bit is used to indicate an error during any flash operation. For flash writes, this bit updates at the end of each write transaction, however, once an error has been detected, this bit remains in the error state until cleared. This function allows the OEM the option of checking the status between each write transaction, or at the end of the update. After a write transaction starts, the flash status (and this error bit) is not accessible until the write transaction completes.

The communication error bit indicates any error on the I^2C command interfaces. The device reports specific details about communication errors using the read communication status command.

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.

The flash error, communication error, and system error bits clear when the read short status is read.

Don't check the read short status command continuously, but only periodically. It is likely that continuous access severely reduces system performance.

3.1.96 Read System Status (D1h)

This command reads system status information for the display module.

Table 3-115. 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 3-116. Byte 1 Read System Status 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

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

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

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

The system sets the DMD training error when the training algorithm can't find a data eye that meets the specified requirements.

Table 3-117. Byte 2 Read System Status 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 3-118. Byte 3 Read System Status Register Field Descriptions

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

Table 3-118. Byte 3 Read System Status Register Field Descriptions (continued)

Bit	Type	Description
0	R	Sequence abort error 0h = No error 1h = Error

Table 3-119. Byte 4 Read System Status Register Field Descriptions

Bit	Type	Description
7-6	R	Reserved
5	R	Actuator Watchdog timer timeout 0h = No timeout 1h = Timeout
4	R	Actuator WFG/PWM Configuration Error 0h = No error 1h = Error
3-2	R	Reserved
1	R	Actuator PWM Mode 0h = Actuator is WFG mode 1h = Actuator is PWM mode
0	R	Actuator Drive status 0h = Actuator Drive Disabled 1h = Actuator Drive Enabled

The system sets the SPI flashless data request error bit if the display does not start sending the requested data before the SPI flashless data request timeout is exceeded. After the timeout is exceeded, the display aborts the current request, and then reattempts the request.

The system sets the SPI flashless communication error bit if the display has three consecutive SPI flashless data request errors. If this happens, it is assumed that the SPI communication link is not operational, and system operations halt. The system requires a reset to restart operations.

The system sets the master versus slave bit as appropriate in both single and dual controller configurations.

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

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

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

3.1.97 Read System Software Version (D2h)

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

Table 3-120. 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

3.1.98 Read Communication Status (D3h)

This command reads system status information for the display module.

3.1.98.1 Read Parameters

The read parameters are described below.

Table 3-121. 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 3-122. Read Communication Status 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

This command returns the communication status for the command bus specified.

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

3.1.98.2 Return Parameters

The return parameters are described below.

Table 3-123. 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 3-124. Byte 5 Read Communication Status 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 sets the invalid command error bit when it does not recognize the command offset. The invalid command offset is reported in the I^2C CMD error offset byte of this status.

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

The system sets the read command error bit when the host terminates the read operation before all of the requested data has been provided, or if the host continues to request read data after all of the requested data has been provided.

The system sets the Invalid number of command parameters error bit when too many or too few command parameters are received. In this case, the command abortes 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 sets 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 3-125. Read Communication Status 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.

3.1.99 Read Controller Device ID (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 3-126. Read Controller Device ID 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 3-127.

Table 3-127. 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/ DLPA3000
01h	DLPC3433	<1280x720	1	7mm x 7mm (0.4mm pitch)	DLPA200x/ DLPA3000
02h	DLPC3432	960x540	1	7mm x 7mm (0.4mm pitch)	DLPA200x/ DLPA3000
03h	DLPC3434	960x540	1	7mm x 7mm (0.4mm pitch)	DLPA200x/ DLPA3000
04h	DLPC3435	<1280x720	1	13mm x 13mm (0.8mm pitch)	DLPA200x/ DLPA3000
05h	DLPC3438	<1280x720	1	13mm x 13mm (0.8mm pitch)	DLPA200x/ DLPA3000
06h	DLPC3436	960x540	1	7mm x 7mm (0.4mm pitch)	DLPA200x/ DLPA3000
07h	DLPC3437	1366x768	2	13mm x 13mm (0.8mm pitch)	DLPA3000/ 3005
09h	DLPC3439	1920x1080	2	13mm x 13mm (0.8mm pitch)	DLPA3000/ 3005

3.1.100 Read DMD Device ID (D5h)

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

3.1.100.1 Read Parameters

The read parameters are described below.

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

Table 3-128. Read DMD Device ID Register Field Descriptions

Bit	Type	Description
7-3	R	Reserved
2-0	R	DMD data selection 0h = DMD device ID 1h = TI Fuse Group Data 1 2h = TI Fuse Group Data 2 3h = TI Fuse Group Data 3 4h = TI Fuse Group Data 4 5h – 7h = Reserved

3.1.100.2 Return Parameters

Table 3-129 describes the return parameters.

Table 3-129. DMD Device ID Reference Table

DMD Device ID				Device Description
Byte 1 (Identifier)	Byte 2 (Byte Count)	Byte 3 (ID-msbyte)	Byte 4 (ID-lsbyte)	Resolution and Type
60h	0Dh	00h	89h	0.23 qHD (960x540, Sub-LVDS)

3.1.101 Read System Temperature (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 (in °C). The unspecified msbits (bits 15:12) are set to '0'.

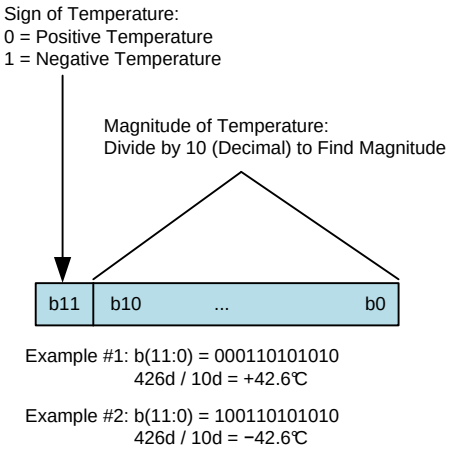

Figure 3-12. Bit Order and Definition

3.1.102 Read Flash Build Version (D9h)

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

Table 3-130. 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.

3.1.103 Write Flash Batch File Delay (DBh)

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

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) .

Applications typically use this command in the auto-init flash batch file (batch file 0), but it is valid for use in any batch file (See write execute flash batch file).

Ensure that the software uses the available hardware timers.

3.1.104 Read DMD I/F Training Data (DCh)

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

3.1.104.1 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 3-131. Byte 1 Read DMD I/F Training Data 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

3.1.104.2 Return Parameters

The return parameters are described below.

Table 3-132. 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 3-133. Byte 1 Read DMD I/F Training Data 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 3-134. Byte 2 Read DMD I/F Training Data Register Field Descriptions

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

Table 3-135. Byte 3 Read DMD I/F Training Data Register Field Descriptions

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

Table 3-136. Byte 4 Read DMD I/F Training Data 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.

3.1.105 Flash Update PreCheck (DDh)

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

3.1.105.1 Read Parameters

The command parameters are described below.

Table 3-137. 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)

3.1.105.2 Return Parameters

The return parameters are described below.

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

Table 3-138. Flash Update PreCheck Register Field Descriptions

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

This command is used in conjunction with the flash data type select command. This command would be sent after the flash data type has been selected, but before any other flash operation. The purpose is to verify that the desired flash update is compatible, and will fit within the existing flash space, for the current flash configuration.

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

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

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

- If replacing the entire flash, the size of the flash build exceeds the size of the flash device in the system.
- If replacing the entire flash except for the 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.

3.1.106 Flash Data Type Select (DEh)

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

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 3-139. Flash Data Type Select 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 3-139. Flash Data Type Select 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 LUT data sets C0h = Entire CAIC LUT Data Set D0h = Entire FPGA LUT Data Set E0h = Entire Actuator Calibration Data Set

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

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

The specified flash data will be written to or read from flash using the write flash start, 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 3-140. 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 will be 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 will be specified by its sequence index number.
Partial CMT data set	Look number	Sequence index number		A CMT data set will be specified by its CMT index number.
Partial OEM splash screen set	Splash number	N/A	N/A	A Splash screen will be 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, will be stored in the OEM calibration block of the flash. It will be 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.

3.1.107 Flash Data Length (DFh)

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

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.

3.1.108 Erase Flash Data (E0h)

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

Since the process of erasing flash sectors can take a significant amount of time, the flash erase complete status bit in the read short status command should be checked periodically (not continuously) to determine when this task has been completed. This bit will be set at the start of the erase process, and will be cleared when the erase process is complete. Flash writes should not be started before the erase process has been completed.

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

The signature bytes are used to minimize unintended flash erases. The command offset and four signature bytes must be received correctly before this command will be recognized and executed.

3.1.109 Write Flash Start (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 will be sent by the write flash start command.

The write flash start command is used to write up to 1024 bytes of data starting at the first address of the data type selected. If more than 1024 bytes are to be written, the write flash continue command must be used. Up to 1024 bytes of data can be written with each write flash continue command, which starts at the end of the last data written.

The flash error bit of the write short status command will indicate if the flash update was successful. This bit will be set for an error at the end of each write transaction, however, once an error has been detected, this bit will remain in the error state until a new data type is selected (selecting a new data type will clear this bit). This will allow the 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.

NOTE: To prevent system error, avoid issuing I2C commands or manipulating the display hardware connections while a flash write sequence is in process. Wait until after rebooting the system to issue new I2C commands or modifying display connections.

3.1.110 Write Flash Continue (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 will be sent by the write flash start command.

The write flash start command is used to write up to 1024 bytes of data starting at the first address of the data type selected. If more than 1024 bytes are to be written, the write flash continue command must be used. Up to 1024 bytes of data can be written with each write flash continue command, which starts at the end of the last data written.

The flash error bit of the write short status command will indicate if the flash update was successful. This bit will be set for an error at the end of each write transaction, however, once an error has been detected, this bit will remain in the error state until a new data type is selected (selecting a new data type will clear this bit). This will allow the 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.

NOTE: To prevent system error, avoid issuing I2C commands or manipulating the display hardware connections while a flash write sequence is in process. Wait until after rebooting the system to issue new I2C commands or modifying display connections.

3.1.111 Read Flash Start (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.

3.1.112 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