Programmer's Guide

DLPC900 Programmer's Guide

TEXAS INSTRUMENTS

Table of Contents

Read This First 4
About This Manual 4
Related Documents from Texas Instruments 4
If You Need Assistance 4
1 Interface Protocol 5
1.1 I²C Interface 5
1.1.1 I²C Transaction Structure 5
1.1.1.1 I²C START Condition 5
1.1.1.2 I²C STOP Condition 5
1.1.1.3 DLPC900 Slave Address 5
1.1.1.4 DLPC900 Sub-Address and Data Bytes 5
1.1.2 Example I²C Read Command Sequence 6
1.1.2.1 Read Command Example with Parameters 6
1.1.3 Example I²C Write Command Sequence 7
1.2 USB Interface 7
1.2.1 USB Transaction Sequence 7
1.2.2 USB Read Transaction Sequence Example 9
1.2.3 USB Write Transaction Sequence Example 10
1.3 INIT_DONE Signal 10
2 DLPC900 Control Commands 11
2.1 DLPC900 Status Commands 11
2.1.1 Hardware Status 12
2.1.2 System Status 13
2.1.3 Main Status 13
2.1.4 Retrieve Firmware Version 14
2.1.5 Reading Hardware Configuration and Firmware Tag Information 14
2.1.6 Read Error Code 15
2.1.7 Read Error Description 15
2.2 DLPC900 Firmware Programming Commands 16
2.2.1 Read Status 16
2.2.2 Enter Program Mode 18
2.2.3 Exit Program Mode 18
2.2.4 Read Control 19
2.2.5 Start Address 19
2.2.6 Erase Sector 20
2.2.7 Download Flash Data Size 20
2.2.8 Download Data 20
2.2.9 Calculate Checksum 21
2.2.10 Controller Enable/Disable Command 21
2.3 Chipset Control Commands 22
2.3.1 Chipset Configuration Commands 22
2.3.1.1 Power Mode 22
2.3.1.2 DMD Park/Unpark 23
2.3.1.3 Curtain Color 23
2.3.2 Parallel Interface Configuration 24
2.3.2.1 Parallel Port Configuration 24
2.3.2.2 Input Data Channel Swap 25
2.3.3 Input Source Commands 25
2.3.3.1 Port and Clock Configuration 25
2.3.3.2 Input Source Configuration 26
2.3.3.3 Input Pixel Data Format 27

2.3.3.4 Internal Test Pattern Select....27
2.3.3.5 Internal Test Patterns Color....28
2.3.3.6 Load Image....29
2.3.4 Image Flip....29
2.3.4.1 Long-Axis Image Flip....29
2.3.4.2 Short Axis Image Flip....30
2.3.5 IT6535 Power Mode....30
2.3.6 Gamma Configuration and Enable....31
2.3.7 LED Driver Commands....33
2.3.7.1 LED Enable Outputs....33
2.3.7.2 LED Driver Current....35
2.3.7.3 Minimum LED Pulse Width in microseconds (µs)....36
2.3.7.4 Minimum LED Pulse Width in nanoseconds (ns)....36
2.3.7.5 Get Minimum LED Pattern Exposure in microseconds (µs)....36
2.3.7.6 Get Minimum LED Pattern Exposure in nanoseconds (ns)....36
2.3.8 GPIO Commands....37
2.3.8.1 GPIO Configuration....37
2.3.8.2 GPIO Clock Configuration....38
2.3.8.3 GPIO Busy....38
2.3.9 Pulse Width Modulated (PWM) Control....38
2.3.9.1 PWM Setup....39
2.3.9.2 PWM Enable....39
2.3.10 Batch File Commands....40
2.3.10.1 Batch File Name....40
2.3.10.2 Batch File Execute....40
2.3.10.3 Batch File Delay....40
2.3.10.4 Batch File Example....41
2.4 Display Mode Commands....41
2.4.1 Display Mode Selection....45
2.4.1.1 Video Mode Resolution....46
2.4.1.2 Input Display Resolution....46
2.4.1.3 DMD Block Load....46
2.4.1.4 Minimum Exposure Times....47
2.4.1.5 DMD Idle Mode....48
2.4.2 Image Header....49
2.4.3 Pattern Image Compression....49
2.4.3.1 Run-Length Encoding....49
2.4.3.2 Enhanced Run-Length Encoding....50
2.4.4 Pattern Display Commands....51
2.4.4.1 Trigger Commands....51
2.4.4.2 LED Enable Delay Commands....54
2.4.4.3 Pattern Display Commands....56
2.4.4.4 Pattern On-The-Fly Commands....60
2.4.4.5 I²C Pass Through Commands....62
3 DLPC900 Fault Status....64
3.1 DLPC900 FAULT_STATUS Location(s)....64
3.2 DLPC900 FAULT_STATUS Interpretation....64
4 Power-Up and Power-Down and Initialization Considerations....65
4.1 Power Up....65
4.2 Power Down....65
4.3 Power-Up Auto-Initialization....65
5 Command Examples....66
5.1 Video Pattern Mode Example....66
5.2 Pre-Stored Pattern Mode Example....66
5.3 Pattern On-The-Fly Example....67
5.4 I²C Pass Through Write Example....67
5.5 I²C Pass Through Read Example....67
A Register Quick Reference....68
A.1 I²C Register Quick Reference....68
A.2 Command Guide....70
B Batch File Command Descriptors....72
B.1 Command Descriptors....72
Revision History....74

List of Figures

Figure 1-1. I2C Read Command Waveform Diagram.... 6
Figure 1-2. I2C Read Data Waveform Diagram.... 6
Figure 1-3. I2C Write Command Waveform Diagram.... 7
Figure 1-4. USB HID Protocol.... 8
Figure 1-5. USB Multi-Transfer Transaction.... 8
Figure 2-1. Image Long-Axis Flip Example.... 30
Figure 2-2. Image Short-Axis Flip Example.... 30
Figure 2-3. Gamma Chart.... 32
Figure 2-4. DLPC900 Single Controller System Block Diagram.... 42
Figure 2-5. DLPC900 Dual Controller System Block Diagram.... 43
Figure 2-6. Bit-Planes of a 24-Bit RGB Image.... 44
Figure 2-7. Bit Partition in a Frame for an 8-Bit Monochrome Image.... 44
Figure 2-8. Video Pattern Mode Timing Diagram Example.... 52
Figure 2-9. Pre-Stored Pattern Mode Timing Diagram Example.... 52
Figure 3-1. DLPC900 FAULT_STATUS Format.... 64

Trademarks

LightCrafter™ and E2E™ are trademarks of Texas Instruments.
DLP® are registered trademarks of Texas Instruments.
All trademarks are the property of their respective owners.

Read This First

About This Manual

This document specifies the command and control interface to the DLPC900 controller and defines all applicable commands, default settings, and control register bit definitions.

Related Documents from Texas Instruments

- DLPC900 Data Sheet, DLPS037
- DLP500YX Data Sheet, DLPS193
- DLP6500FLQ Data Sheet, DLPS040
- DLP6500FYE Data Sheet, DLPS053
- DLP670S Data Sheet, DLPS194
- DLP9000 Data Sheet, DLPS036
- DLP® LightCrafter™ Single DLPC900 Controller Evaluation Module (EVM) User's Guide, DLPU101
- DLP® LightCrafter™ Dual DLPC900 Controller Evaluation Module (EVM) User's Guide, DLPU102

If You Need Assistance

Visit the TI E2E™ support forums at DLP Products and MEMS TI E2E Community.

1 Interface Protocol

This chapter describes the interface protocol between the DLPC900 and a host processor. The DLPC900 supports two host interface protocols: I²C and USB 1.1 slave interfaces.

1.1 I²C Interface

The DLPC900 controller uses the I²C protocol to exchange commands and data with a host processor. The I²C protocol is a two-wire serial data bus. One wire, SCL, serves as a serial clock, while the second wire, SDA, serves as serial data. Several different devices can be connected together in an I²C bus. Each device is software addressable by a unique address. Communication between devices occurs in a simple I²C "master-to-slave" relationship.

1.1.1 I²C Transaction Structure

All I²C transactions are composed of a number of bytes, combined in the following order:

START Condition, 7-Bit Slave Address Byte + 1 R/W Bit, Sub-Address Byte, N-Data Bytes, STOP Condition

Where N in *N-Data Bytes* varies based on the sub-address.

1.1.1.1 I²C START Condition

All I²C transactions begin with a START condition. A START condition is defined by a high-to-low transition on the SDA line, followed by a high-to-low transition on the SCL line.

1.1.1.2 I²C STOP Condition

All I²C transactions end with a STOP condition. A STOP condition is defined by a low-to-high transition on the SDA line, followed by a low-to-high transition on the SCL line.

1.1.1.3 DLPC900 Slave Address

The DLPC900 offers a programmable slave address. Refer to the App Defaults Settings found in the DLPC900 LightCrafter GUI Firmware tab to set a different slave address. The default I²C settings are shown in Table 1-1. The Write Slave Address must be an even 7-bit address, and the Read Slave Address must be the Write Slave Address plus 1.

Table 1-1. I²C Slave Settings

ADDRESSING MODE	DEFAULT WRITE ADDRESS	DEFAULT READ ADDRESS	MAXIMUM CLOCK RATE (kHz)
7-bit	0x34	0x35	400

1.1.1.4 DLPC900 Sub-Address and Data Bytes

The DLPC900 I²C sub-address corresponds to the byte address of the DLPC900 commands described in Appendix A. Most I²C sub-addresses have a Read and Write command pair where the Write command equals the Read command with the most significant bit set. For example, Table 1-2 and Table 1-4 show the Input Data Channel Swap sub-address command pair is (0x04,0x84), where the Write sub-address command 0x84 is the Read sub-address command 0x04 with the most significant bit set. Each sub-address command requires a certain number of data bytes, and each command is followed by variable length data where the **least significant byte is first for each parameter**.

Note

The DLPC900 I²C command data is formatted with the least significant byte first for each parameter in the data. This maintains the same format with the USB protocol.

The DLPC900 internal command buffer has a maximum of 512 bytes and it is shared between the Read and Write commands; therefore, whenever a Read command is executed it must be followed by I²C operation with the Read Slave Address to retrieve the data otherwise the data is overwritten by the next command executed. See Section 1.1.2 for a Read command example.

1.1.2 Example I²C Read Command Sequence

To execute a command to read the Input Data Channel Swap setting, the host builds a sequence of bytes containing the slave address, the sub-address, and the data (if any), and performs the following steps:

1. The host performs the required START condition followed by sending the sequence of bytes.
2. The DLPC900 holds the SCL line low to indicate it is busy.
3. The host waits for the DLPC900 to release the SCL line.
4. Once the SCL line goes high, the host performs a STOP condition.
5. The host then performs a START condition followed by sending the Read Slave Address (0x35), and then reads the required number of bytes and concludes with a STOP condition.

An example of the above read command sequence is shown in Table 1-2, and a waveform diagram of a host executing this read sequence is shown in Figure 1-1 and Figure 1-2.

Table 1-2. Read Command Sequence Example (1)

SLAVE ADDRESS	SUB-ADDRESS	DATA
34	04	
35		03

(1) All values shown are in HEX notation.

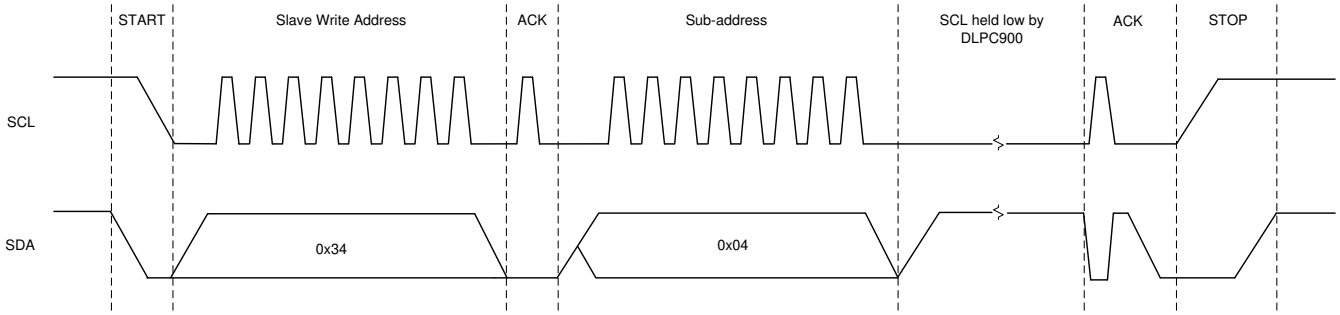

Figure 1-1. I²C Read Command Waveform Diagram

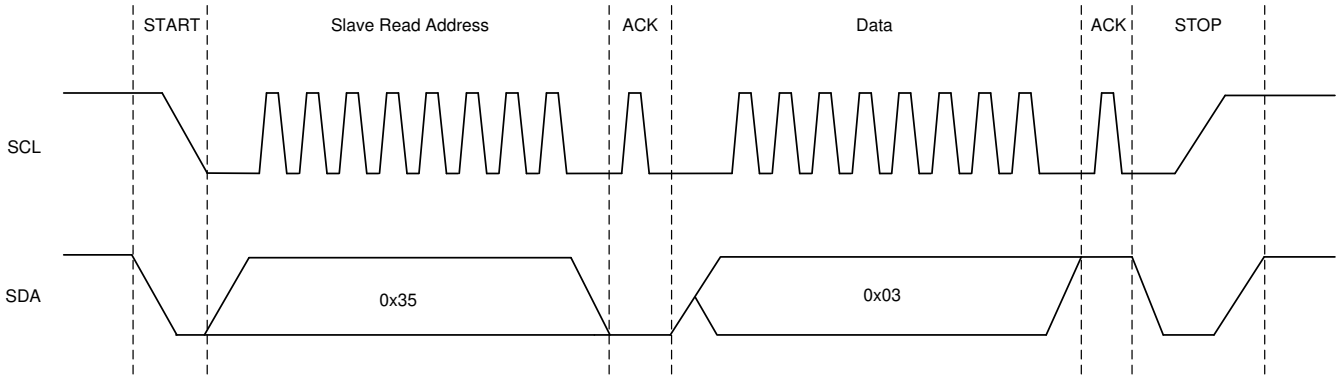

Figure 1-2. I²C Read Data Waveform Diagram

1.1.2.1 Read Command Example with Parameters

Some Read sub-address commands require a parameter(s) to be included in the sequence. For example, the command in Section 2.3.8.1 has multiple GPIO to choose from. Therefore, the GPIO selection parameter must be included in the Read byte sequence in order to retrieve the configuration for the GPIO chosen. Table 1-3 shows the two I²C operations, where the first row contains the parameter data 06 which indicates GPIO 6. The second row is the returned data of 06 03, where 06 was the chosen GPIO 6 and has a configuration of 03.

Table 1-3. Read Command with Parameter Sequence Example (1)

SLAVE ADDRESS	SUB-ADDRESS	DATA
34	44	06
35		06 03

(1) All values shown are in HEX notation.

1.1.3 Example I²C Write Command Sequence

To execute a command to set the Input Data Channel Swap value, the host builds a sequence of bytes containing the slave address, the sub-address, and the data, and performs the following steps.

1. The host performs the required START condition followed by sending the sequence of bytes.
2. The host performs a STOP condition.

An example of the above write command sequence is shown in Table 1-4, and a waveform diagram of a host executing this write sequence is shown in Figure 1-3.

Table 1-4. Write Command Sequence Example (1)

SLAVE ADDRESS	SUB-ADDRESS	DATA
34	84	02

(1) All values shown are in HEX notation.

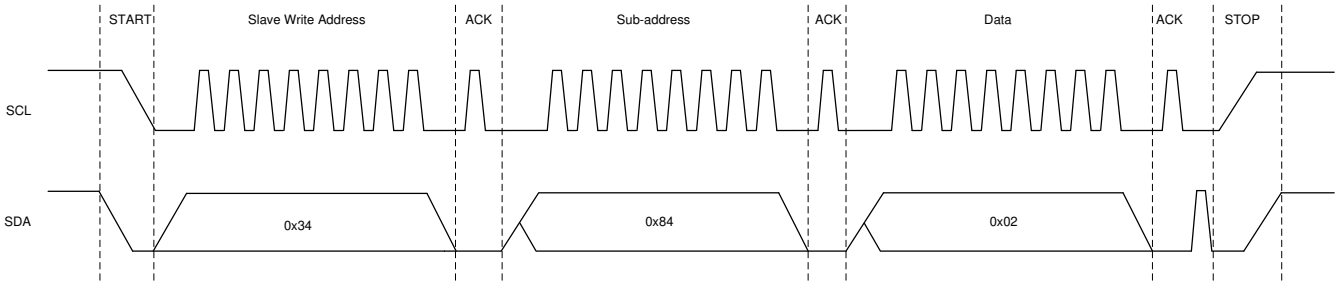

Figure 1-3. I²C Write Command Waveform Diagram

1.2 USB Interface

The DLPC900 controller also supports the USB 1.1 human interface device (HID) to exchange commands and data with a host processor. The USB commands are variable length data packets that are sent with the **least significant byte first for each parameter**.

1.2.1 USB Transaction Sequence

The USB 1.1 HID protocol has the structure shown in Figure 1-4. The host must build a stream of bytes that consist of the Report ID, Header, and the payload. The following is a description of these three parts.

Report ID: The Report ID is always set to 0 and always the leading byte of all transfers.

Header: The header consists of four bytes.

1) Flag Byte: Shown in Figure 1-4 and described in the Read and Write examples in Section 1.2.2 and Section 1.2.3.

2) Sequence Byte: The sequence byte can be a rolling counter. It is used primarily when the host wants a response from the DLPC900. The DLPC900 responds with the same sequence byte that the host sent. The host can then match the sequence byte from the command it sent with the sequence byte from the DLPC900 response.

3) Length: Two bytes in length, this denotes the number of data bytes in the Payload only.

Payload Bytes: The payload bytes consist of the USB command followed by the data that is associated with the command.

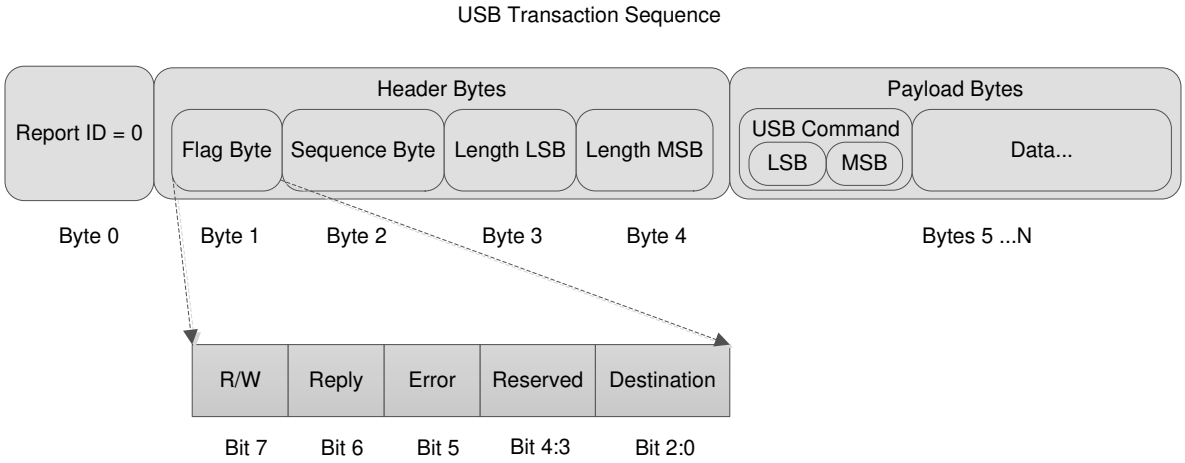

Figure 1-4. USB HID Protocol

During a Write operation, the host transmits the entire transaction sequence to the DLPC900, and the DLPC900 performs the operation associated with the Write command. During a Read operation, the host transmits the entire transaction sequence to the DLPC900, and the DLPC900 performs the operation associated with the Read command. Therefore, both Write and Read transactions are considered *writes* to the DLPC900 where the host performs an API level *Writefile* to the HID driver. The difference is when the DLPC900 executes a Read operation, where the DLPC900 places the response into its internal buffer and waits for the host to perform an API level *Readfile* to the HID driver and only then does the DLPC900 transmit the response data back to the host.

The DLPC900 internal command buffer has a maximum of 512 bytes and it is shared between both the Write and Read operations; therefore, whenever the host performs a Read operation, it must be followed by the *Readfile* to the HID driver to get the response otherwise the response data is overwritten by the next Write or Read operation.

The HID protocol is limited to 64 byte transfers in both directions. Therefore, commands that are larger than 64 bytes require multiple transfers. Whenever such a command is used, only the very first transfer requires the Header and the USB Command. The Report ID is always the leading byte of all transfers. Figure 1-5 shows an example of a Write command that contains 76 bytes and requires two transfers. Notice that the first transfer contains 65 bytes, which is correct. The host hardware level HID driver extracts the Report ID before transmitting or receiving the data over the USB bus.

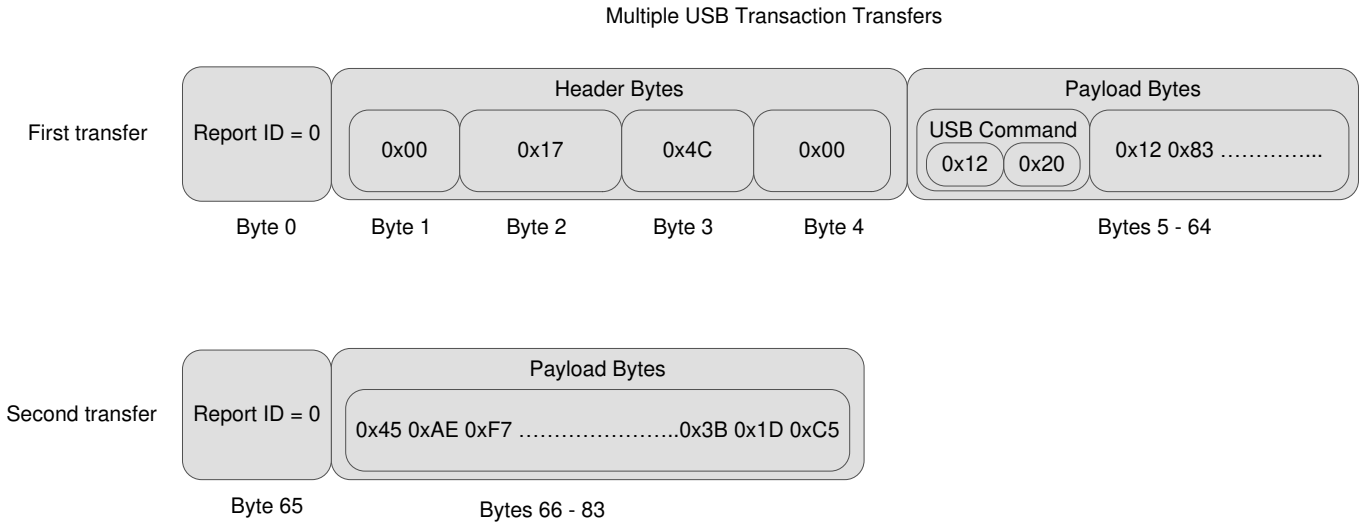

Figure 1-5. USB Multi-Transfer Transaction

1.2.2 USB Read Transaction Sequence Example

To perform a Read operation on the DLPC900, the host must assemble a sequence of bytes that corresponds to the command being used. The following Table 1-5 shows an example on how to read the curtain color intensity of each color.

Table 1-5. Read Operation Example (1)

REPORT ID BYTE	FLAG BYTE	SEQUENCE BYTE	LENGTH (2)	USB COMMAND (2)
00	C0	11	02 00	00 11

(1) All values shown are in HEX notation.
(2) LSB precedes the MSB for each parameter.

1. Report ID byte: Always set to 0.
2. Flag byte. Where:
   - Bits 2:0 are set to 0x0 for regular DLPC900 operation.
   - Bit 6 is set to 0x1 to indicate the host wants a reply from the device.
   - Bit 7 is set to 0x1 to indicate a read transaction.
3. Sequence byte: The sequence byte can be a rolling counter. It is used primarily when the host wants a response from the DLPC900. The DLPC900 responds with the same sequence byte that the host sent. The host can then match the sequence byte from the command it sent with the sequence byte from the DLPC900 response.
4. Length: Two bytes in length, this denotes the number of data bytes in the sequence and excludes the number of bytes in steps 1 through 4. It denotes the total number of bytes sent in steps 5 (command bytes).
5. USB Command: Two byte USB command.
6. Once the host transmits the data over the USB interface, the DLPC900 responds to the Read operation by placing the response data in its internal buffer. The host must then perform a HID driver read operation. Table 1-6 shows the response data sent back from the DLPC900.
   a. Report ID: Always set to 0.
   b. Flag byte: The same as was sent plus error bit. The host may check the error flag (bit 5) as follows.
      i. 0 = No errors.
      ii. 1 = Command not found or command failed.
   c. Sequence byte: The same as was sent. The host may match the sent sequence byte with the response sequence byte.
   d. Length: Number of data bytes. The host must assemble the data according to the definition of the command.

Table 1-6. Read Response Example (1)

REPORT ID BYTE	FLAG BYTE	SEQUENCE BYTE	LENGTH (2)	DATA (2)
00	C0	11	06 00	FF 01 FF 01 FF 01

(1) All values shown are in HEX notation.
(2) LSB precedes the MSB for each parameter.

1.2.3 USB Write Transaction Sequence Example

To perform a Write operation on the DLPC900, the host must assemble a sequence of bytes that corresponds to the command being used. The following Table 1-7 shows an example on how to set the curtain color intensity of each color to 511.

Table 1-7. Write Operation Example (1)

REPORT ID BYTE	FLAG BYTE	SEQUENCE BYTE	LENGTH (2)	USB COMMAND (2)	DATA (2)
00	00	12	08 00	00 11	FF 01 FF 01 FF 01

(1) All values shown are in HEX notation.
(2) LSB precedes the MSB for each parameter.

1. Report ID byte: Always set to 0.
2. Flag byte. Where:
   - Bits 2:0 are set to 0x0 for regular DLPC900 operation.
   - Bit 6 is set to 0x0 to indicate the host does not want a reply from the device. This bit is set to 0x1 only if a reply is needed, which is usually not required.
   - Bit 7 is set to 0x0 to indicate a write transaction.
3. Sequence byte: The sequence byte can be a rolling counter. It is used primarily when the host wants a response from the DLPC900. Normally during a write operation, the DLPC900 does not respond; however, the host can continue to increment the sequence byte for the next command operation.
4. Length: Two bytes in length, this denotes the number of data bytes in the sequence and excludes the number of bytes in steps 1 through 4. It denotes the total number of bytes sent in steps 5 (command bytes) and 6 (data bytes).
5. USB Command: Two byte USB command.
6. Data: The data appropriate to the command.

1.3 INIT_DONE Signal

The DLPC900 does not have a dedicated INIT_DONE signal output to indicate that it has completed its power-up initialization and is ready to accept commands. The user may configure one of the nine GPIOs available as an INIT_DONE signal output simply by adding the GPIO configuration into the default batch file that is executed at power-up. A 10-kΩ pull-down resistor must be connected to the GPIO that is used.

The following is an example of adding the configuration for GPIO_08 to a batch file, where GPIO_08 is configured as an output and the signal is set high. When this command is added to the top of the batch file, the GPIO output goes high in approximately 800 ms from the time POSENSE goes high:

GPIO_CONFIG: 0x8 0x3

2 DLPC900 Control Commands

This chapter lists the DLPC900 control commands.

The following sections list the supported control commands of the DLPC900. In the *Type* column, 'wr' type is a writeable field through I^2C or USB write transactions. Data can also be read through I^2C or USB read transactions for 'wr' type bits. Type *r* is read-only. Write transactions to read-only fields are ignored.

The Reset column in all of the following command tables is the default value after power up. These values may be overwritten after power up.

Note

Reserved bits and registers. When writing to valid command bit fields, set all bits marked as unused or reserved to 0, unless specified otherwise.

Note

Momentary Image Corruption During Command Writes. Certain commands may cause brief visual artifacts in the display image under some circumstances. Command data values may always be read without impacting displayed image. To avoid momentary image corruption due to a command, disable the LEDs prior to the command write, then re-enable the LEDs after all commands have been issued.

Note

Writing or reading from undocumented registers is NOT recommended.

2.1 DLPC900 Status Commands

The DLPC900 has the following set of status commands:

Hardware Status

System Status

Main Status

Retrieve Firmware Version

Read Error Codes

2.1.1 Hardware Status

The hardware status command provides status information on the sequencer, digital micromirror device (DMD) controller, and initialization of DLPC900.

Table 2-1. Hardware Status Command

I²C	USB
Read	0x1A0A
0x20	

Table 2-2. Hardware Status Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
0	0	Internal Initialization 0 = Error 1 = Successful	d1	r
	1	0 = No Error 1 = Incompatible Controller or DMD **Note** This error also occurs if the wrong firmware for the system has been loaded.	d0	r
	2	DMD Reset Controller Error 0 = No error has occurred 1 = Multiple overlapping bias or reset operations are accessing the same DMD block.	d0	r
	3	Forced Swap Error 0 = No error has occurred. 1 = Forced Swap Error occurred.	d0	r
	4(1)	0 = No Slave Controller Present 1 = Slave Controller Present and Ready	d0 (single controller DMD) d1 (dual controller DMD)	r
	5	Reserved	d0	r
	6	Sequencer Abort Status Flag 0 = No error has occurred 1 = Sequencer has detected an error condition that caused an abort	d0	r
	7	Sequencer Error 0 = No error has occurred. 1 = Sequencer detected an error.	d0	r

(1) When the DLPC900 is combined with a single controller DMD, this bit is 0. When two DLPC900 controllers are combined with a dual controller DMD, this bit must be 1 for proper operation. If the bit is 0 and the DLPC900 is combined with a dual controller DMD, this indicates a malfunction in one or both controllers.

Note

Any error condition indicates a fault condition and it must be corrected.

2.1.2 System Status

The system status command provides the DLPC900 status on internal memory tests.

Table 2-3. System Status Command

I²C	USB
Read	0x1A0B
0x21	

Table 2-4. System Status Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
0	0	Internal Memory Test 0 = Internal Memory Test failed 1 = Internal Memory Test passed	d1	r
	1:7	Reserved	d0	r

2.1.3 Main Status

The main status command provides the status of DMD park and DLPC900 sequencer, frame buffer, and gamma correction.

Table 2-5. Main Status Command

I²C	USB
Read	0x1A0C
0x22	

Table 2-6. Main Status Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
0	0	DMD Park Status 0 = DMD micromirrors are not parked 1 = DMD micromirrors are parked	d1	r
	1	Sequencer Run Flag 0 = Sequencer is stopped 1 = Sequencer is running normally	d0	r
	2	Video Frozen Flag 0 = Video is running (Normal frame change) 1 = Video is frozen (Displaying single frame)	d0	r
	3	External video source locked 0 = External source not locked 1 = External source locked	d0	r
	4	Port 1 syncs valid 0 = Port 1 syncs not valid 1 = Port 1 syncs valid	d0	r
	5	Port 2 syncs valid 0 = Port 2 syncs not valid 1 = Port 2 syncs valid	d0	r
	7:6	Reserved	d0	r

2.1.4 Retrieve Firmware Version

This command reads the version information of the DLPC900 firmware.

Table 2-7. Retrieve Firmware Version Command

I²C	USB
Read	0x0205
0x11	

Table 2-8. Get Version Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
3:0		Application software revision:	Matches firmware version read from stored firmware.	r
	15:0	Application software patch number		
	23:16	Application software minor revision		
	31:24	Application software major revision		
7:4		API software revision:	d0	r
	15:0	API patch number		
	23:16	API minor revision		
	31:24	API major revision		
11:8		Software configuration revision:	d0	r
	15:0	Software configuration patch number		
	23:16	Software configuration minor revision		
	31:24	Software configuration major revision		
15:12		Sequencer configuration revision:	d0	r
	15:0	Sequencer configuration patch number		
	23:16	Sequencer configuration minor revision		
	31:24	Sequencer configuration major revision		

2.1.5 Reading Hardware Configuration and Firmware Tag Information

This command reads the hardware configuration of the system and also returns the 31 byte ASCII firmware tag information.

Table 2-9. Reading Hardware Configuration and Firmware Tag Information Command

I²C	USB
Read	0x0206
0x12	

Table 2-10. Reading Hardware Configuration and Firmware Tag Command Response

BYTE	VALUE	DESCRIPTION	RESET	TYPE
0	0x00	Unknown	Read from firmware	r
	0x01	DLP6500 hardware		
	0x02	DLP9000 hardware		
	0x03	DLP670S hardware (1)		
	0x04	DLP500YX hardware (1)		
	0x05 - 0xFF	Reserved		
32:1		31 byte ASCII firmware tag information		r

(1) Firmware version 6.x introduces support for two new DMD devices. These are shown in Table 2-10

2.1.6 Read Error Code

This command retrieves the error code number from the DLPC900 of the last executed command.

Table 2-11. Read Error Code Command

I²C	USB
Read	0x0100
0x32	

Table 2-12. Read Error Code Command Definition

BYTE	VALUE	DESCRIPTION	RESET	TYPE
0	0	No error	0	r
	1	Batch file checksum error		
	2	Device failure		
	3	Invalid command number		
	4	Incompatible controller / DMD		
	5	Command not allowed in current mode		
	6	Invalid command parameter		
	7	Item referred by the parameter is not present		
	8	Out of resource (RAM / Flash)		
	9	Invalid BMP compression type		
	10	Pattern bit number out of range		
	11	Pattern BMP not present in flash		
	12	Pattern dark time is out of range		
	13	Signal delay parameter is out of range		
	14	Pattern exposure time is out of range		
	15	Pattern number is out of range		
	16	Invalid pattern definition (errors other than 9-15)		
	17	Pattern image memory address is out of range		
	18-254	Not defined		
	255	Internal Error		

2.1.7 Read Error Description

This command retrieves the error descriptive string from the DLPC900 of the last executed command. The string is composed of character bytes ending with a null termination character.

Table 2-13. Read Error Description Command

I²C	USB
Read	0x0101
0x33	

Table 2-14. Read Error Description Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
127:0	All	Error description for the last executed command. 0 terminated string of character bytes.	0	r

2.2 DLPC900 Firmware Programming Commands

The Programming commands manage downloading a new firmware image into flash memory. This can be done over I²C or USB interfaces. The commands in the DLPC900 Programming Commands section are only valid in **program mode** except for Enter Program Mode (**I²C**: 0x30 or **USB** 0x3001), which exits normal mode and enters program mode. Once in program mode, the user must issue the proper Exit Program Mode (**I²C**: 0x30 or **USB** 0x0030) command to return to normal mode. **While in program mode, commands outside of this section do not work.**

Flash memory has the address layout shown in Table 2-15. The design is for up to a single 128 megabyte flash device for storing the firmware.

Note

Depending on memory needs the design can also be used for a single 16, 32, or 64 megabyte flash memory.

The firmware consists of the bootloader, the main application, any sequences/images stored in flash (optional), and 1 Megabyte of reserved space. This area (0x9000000 - 0x9FFFFFF) must not be overwritten.

The bootloader is located at the *beginning* of flash memory block 0. The size of the bootloader is 128 kilobytes, beginning at address 0xF9000000. The bootloader is necessary for operation. If the bootloader becomes corrupted in some way it may render the device inoperable requiring JTAG to reprogram. The bootloader is followed by the main application and 1 megabyte of reserved space. Pattern/sequence data starts at 0x9240000. Patterns may not span memory block boundaries.

Note

Writing across memory block boundaries is not permitted. Patterns must not span across block boundaries between blocks because of the extended addressing schema. If a pattern does not fit in a given block, the entire, 24-bit image (or composite image) must be moved into next block.

Table 2-15. Flash Device Layout

Memory Block	Address Space (Start and End)	Single Flash Memory Addressed	Megabytes	Contents
0	0xF9000000 - 0xF9FFFFFF **Reserved:** *Bootloader* 0xF9000000 - 0xF901FFFF *Application binary* 0xF9020000 - 0xF913FFFF *1 megabyte* 0xF9140000 - 0xF923FFFF	0x00000000 - 0x00FFFFFF	0 - 15	• **Bootloader** • **Application binary** • **1 megabyte reserved space** • Sequences/patterns
1	0xFA000000 - 0xFAFFFFFF	0x01000000 - 0x01FFFFFF	16 - 31	Patterns only
2	0xF8000000 - 0xF8EFFFFF	0x02000000 - 0x02FFFFFF	32 - 47	Patterns only
3	0x03000000 - 0x03FFFFFF	0x03000000 - 0x03FFFFFF	48 - 63	Patterns only
4	0x04000000 - 0x04FFFFFF	0x4000000 - 0x04FFFFFF	64 - 79	Patterns only
5	0x05000000 - 0x05FFFFFF	0x05000000 - 0x05FFFFFF	80 - 95	Patterns only
6	0x06000000 - 0x06FFFFFF	0x06000000 - 0x06FFFFFF	96 - 111	Patterns only
7	0x07000000 - 0x07FFFFFF	0x07000000 - 0x07FFFFFF	112 - 127	Patterns only

2.2.1 Read Status

This command indicates if the flash is ready to be programmed and also if a flash operation is in progress.

Table 2-16. Read Status Command

I²C	USB
Read	0x0000
0x23	

Table 2-17. Read Status Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
0	0	Master ready 0 = Master not ready 1 = Master ready	d1	r
	1	Slave ready (Valid only on Dual DLPC900 board) 0 = Slave not ready 1 = Slave ready	d0 (single controller DMD) d1 (dual controller DMD)	
	2	Slave controller flash busy (Valid only on Dual DLPC900 board) 0 = Slave not busy 1 = Slave busy	d0	
	3	Master controller flash busy 0 = Master not busy 1 = Master busy	d0	
	4	Reserved	d0	
	5	Slave controller present (Valid only on Dual DLPC900 board) 0 = Slave not present 1 = Slave present	d0 (single controller DMD) d1 (dual controller DMD)	
	6	Slave controller program mode (Valid only on Dual DLPC900 board) 0 = Slave not in program mode 1 = Slave in program mode	d0 (single controller DMD) d1 (dual controller DMD)	
	7	Master controller program mode 0 = Master not in program mode 1 = Master in program mode	d1	
1	3:0	Major Version	x	
	7:4	Minor version	x	
2	7:0	Patch version	x	
3	7:0	Controller ID	x52	
4	7:0	Bootloader ID 65h = Single DLPC900 90h = Dual DLPC900	x65 (Single DLPC900) x90 (Dual DLPC900)	
5	7:0	Bytes 1 - 15 are from Master or Slave 0 = Bytes 1 - 15 are from Slave 1 = Bytes 1 - 15 are from Master	d1	
6	7:0	Data (LSB)	d0	
7	7:0	Data	d0	
8	7:0	Data	d0	
9	7:0	Data (MSB)	d0	
10	7:0	Reserved	x3	
11	7:0	Reserved	d0	
12	7:0	Data (LSB)	d0	
13	7:0	Data	d0	
14	7:0	Data	d0	
15	7:0	Data (MSB)	d0	

2.2.2 Enter Program Mode

This command tells the controller to enter its programming mode and jump to the boot loader. If the boot loader receives this command, then the command has no effect.

Table 2-18. Enter Program Mode Command

I²C	USB
Write	0x3001
0x30	

Table 2-19. Enter Program Mode Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
0	1:0	Program Mode 1 = Enter Program Mode – Jump to boot loader	d0	w
	7:2	Reserved		

2.2.3 Exit Program Mode

This command tells the controller to exit its programming mode. If the application receives the exit command, the command has no effect.

Table 2-20. Exit Program Mode Command

I²C	USB
Write	0x0030
0x30	

Table 2-21. Exit Program Mode Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
0	1:0	Program Mode 2 = Exit Program Mode – Reset controller and run application	d0	w
	7:2	Reserved		

2.2.4 Read Control

This command reads the Flash Manufacturer and Device IDs, as well as the Checksum, after the Calculate Checksum command is executed.

Table 2-22. Read Control Command

I²C	USB
Read	0x0015
0x15	

Table 2-23. Query Flash IDs Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
0	3:0	ID 0 = Request Checksum xB = Request Number of flash present xC = Requests Flash Manufacturer ID xD = Requests Flash Device ID	d0	r
	7:4	Reserved		

2.2.5 Start Address

The Start Address command serves three purposes.

1) Specifies the start address of the flash download write operation. It is the responsibility of the user to ensure that the start address is on a sector boundary in the current flash device.

2) Specifies the start address where checksum operation begins.

3) Specifies the sector address to be erased. The address must be the start of a sector.

The Flash Data Size command always follows 1 and 2 above, which defines how many bytes to be downloaded or how many bytes to include for the checksum operation.

The user must avoid erasing the first 128 kilobytes of the boot flash as this contains the boot image. The user must also avoid erasing other sectors that contain firmware that are required for proper controller operation.

Table 2-24. Start Address Command

I²C	USB
Read	0x0032
0x32	

Table 2-25. Start Address Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
3:0	31:0	4 byte flash address. Byte 0 is LSB, byte 4 is MSB. Valid Range: 0x03000000 - 0x07FFFFFF 0xF8000000 – 0xFAEFFFFF 0xF9000000 – 0xF9FFFFFF	x0	w

2.2.6 Erase Sector

This is a system write command to erase a sector of flash memory. **Do not execute this command until valid data has been written to the Flash Start Address. Users are responsible for ensuring that a valid address has been written.** The Busy bit is set in the Boot Loader status byte while the sector erase is in progress. No data is associated with this command.

Table 2-26. Erase Sector Command

I²C	USB
Write	0x0028
0x28	

Note

TI cautions against erasing the boot sector of the device as this contains key initialization parameters and the flash programming functionality. Only the sector that contains the start address is erased, not all sectors from the start address to the end of the device. Users must either pre-erase all sectors to be programmed, or erase and program each sector individually.

2.2.7 Download Flash Data Size

System write command to specify the size of the following flash download. The data size is sent to tell the Boot Loader how many bytes to expect to program into the flash device. It is also used for specifying the checksum range when requesting that operation.

Table 2-27. Download Flash Data Size Command

I²C	USB
Write	0x0033
0x33	

Table 2-28. Download Data Size Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
3:0	31:0	4 Byte flash size. Valid Range 4 - 0x2FFFFFF. Byte 0 is LSB, byte 3 is MSB.	x0	w

2.2.8 Download Data

This command contains the flash data to be programmed. The maximum data size which can be sent in each command is 512 bytes, which corresponds to a data length of 514. The number of bytes downloaded by consecutive download data commands must match the predefined Flash Data Size for the operation to be successful.

Table 2-29. Download Data Command

I²C	USB
Write	0x0025
0x25	

Table 2-30. Download Data Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
0	7:0	Length LSB	x0	w
1	7:0	Length MSB		
513:2	4095:0	Up to 512 Data Bytes		
514	7:0	Checksum		

2.2.9 Calculate Checksum

This command calculates the checksum. Executing this command causes the Boot Loader to read the data in the flash memory and calculate a 4-byte 8-bit checksum. The Busy bit is set in the Boot Loader status byte while the checksum computation is in progress. After completion, the 4-byte checksum can be read back through the Read Control command. The data range to be summed is specified by writing appropriate data with the Flash Start Address and Flash Data Size commands. There is no data associated with this command.

Table 2-31. Calculate Checksum Command

I²C	USB
Write	0x0026
0x26	

2.2.10 Controller Enable/Disable Command

This command stops the given controller from executing any further commands until enabled by the same command. This command is intended to be used when two DLPC900 controllers are combined with a dual controller DMD, where one controller is the I²C master and the other is the I²C slave.

Table 2-32. Controller Enable/Disable Command

I²C	USB
Write	0x0031
0x31	

Table 2-33. Controller Enable/Disable Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
0	0	1 – Disable Master Controller	x0	w
		0 – Enable Master Controller		
	1	1 – Disable Slave Controller	x0	w
		0 – Enable Slave Controller		
	7:2	Reserved	x0	w

2.3 Chipset Control Commands

The DLPC900 I²C and USB control commands are accepted in any order, except when special sequencing is required (for example, setting up the flash). Each control command is validated for sub-address and parameter errors as it is received. Commands failing validation are ignored. On power up, it is necessary to wait for DLPC900 to complete its initialization before sending any I²C or USB commands.

2.3.1 Chipset Configuration Commands

The Chipset Configuration commands enable control of the power mode, DMD park state and image curtain display.

2.3.1.1 Power Mode

The Power Control places the DLPC900 in a standby state and powers down the DMD interface. Enter Standby mode prior to any planned system power shutdowns to safely park the micro-mirrors. Enable Standby mode only after all data for the last frame to be displayed has been transferred to the DLPC900. Standby mode must be disabled prior to sending any new data. After executing this command, the host may poll the system status using I²C commands 0x20, x21, and 0x22 or USB commands 0x1A0A, 0x1A0B, and 0x1A0C to attain status.

Table 2-34. Power Mode Command

I²C		USB
Read	Write	0x0200
0x07	0x87	

Table 2-35. Power Mode Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
0	1:0	Power Mode 0 = Return to Normal mode from Standby Mode of operation. In Normal mode, the selected external source is displayed. 1 = Enter Standby mode to place the DMD in a standby state. Standby disables the front end input data interfaces and park and power down parts of the DMD. **Note** Standby mode requires a wait period of two minutes to run 50/50 and park the DMD. Three additional seconds are needed to complete all operations before issuing a Return to Normal command. **Note** Once Standby has been issued the only command accepted by the DLPC900 controller board is Return to Normal (0). Any other command can result in unexpected behavior. Particularly a Park/ Unpark is not accepted until Normal mode is resumed.	d0	wr
		2 = Perform a software reset If a software reset is performed in a batch file, no further commands in the batch file are processed resulting in an "idx" (batch file line index number) error. A USB or I²C connection must be established before starting a new batch file. **Note** If a hardware reset is performed while the system is in standby mode the system needs to poll the status again to recognize that normal operation has resumed.		w
		3 = Reserved		r
	7:2	Reserved	d0	r

2.3.1.2 DMD Park/Unpark

The DMD Park commands parks the mirrors of the DMD. The DMD Unpark command unparks the mirrors of the DMD. Because the Standby command execution includes the parking of the DMD, the Park command is unneeded for planned power down events. Therefore using the isolated Park/Unpark command is no longer recommended except for debugging purposes.

If the system is not going to be powered down it is recommended to set the DMD to Idle Mode (see Section 2.4.1.5).

Note

Except for Video Mode it is required to issue a Stop command before issuing a Park command. If the Display Mode is set to Pre-stored pattern mode, Video pattern mode or Pattern On-The-Fly mode, the pattern sequence must be stopped prior to execution of this command. The device stops the pattern sequence by calling Pattern Display Start/Stop (Section 2.4.4.3.1).

It is only Video Mode that resumes without further command after the device issues an Unpark command to returns mirrors to operation. For all other mode, the device must resend a Pattern Display LUT Configuration command and Pattern Display LUT Reorder Configuration command before restarting display of patterns sequence.

Table 2-36. DMD Park and Unpark Command

I²C		USB
Read	Write	0x0609
0x14	0x94	

Table 2-37. DMD Park and Unpark Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
0	0	0 = Unpark DMD. Controller returns to the last commanded operating mode. 1= Park DMD. Mirrors go to parked state. LED outputs are disabled.	d0	wr
	7:1	Reserved	d0	r

2.3.1.3 Curtain Color

This register provides image curtain control. When enabled and the input source is set to external video with no video source connected, a solid color field is displayed on the entire DMD display. The Display Curtain Control provides an alternate method of masking temporary source corruption from reaching the display due to on-the-fly reconfiguration. It is also useful for optical test and debug support.

Table 2-38. Curtain Color Command

I²C		USB
Read	Write	0x1100
0x06	0x86	

Table 2-39. Display Curtain Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
1:0	9:0	Red color intensity in a scale from 0 to 1023	d0	wr
	15:10	Reserved	d0	r
3:2	9:0	Green color intensity in a scale from 0 to 1023	d0	wr
	15:10	Reserved	d0	r
5:4	9:0	Blue color intensity in a scale from 0 to 1023	d1023	wr
	15:10	Reserved	d0	r

2.3.2 Parallel Interface Configuration

The Parallel Interface Configuration manages the operation of the RGB parallel interface.

2.3.2.1 Parallel Port Configuration

The Parallel Port Configuration command reads the parallel port configuration details.

Table 2-40. Parallel Port Configuration Command

I²C		USB
Read	Write	0x1A3C
0x64	N/A	

Table 2-41. Parallel Port Configuration Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
1:0	32:0	Total DMD Area - pixels per line **Note** This is the entire width in pixels of the native DMD resolution	Read from EDID	r
3:2	32:0	Total DMD Area - lines per frame	Read from EDID	r
5:4	32:0	Active Area - pixels per line **Note** For dual DLPC900 DMDs this is half of the width of the native DMD resolution	Read from EDID	r
7:6	32:0	Active Area - lines per frame	Read from EDID	r
9:8	32:0	Active Area - first pixel	Read from EDID	r
11:10	32:0	Active Area - first line	Read from EDID	r
13:12	32:0	Bottom Field - first line	Read from EDID	r
17:14	64:0	Pixel Clock - frequency in Hz	Read from EDID	r

2.3.2.2 Input Data Channel Swap

The Input Data Channel Swap commands configure the specified input data ports and maps the data sub-channels. The DLPC900 interprets channel A as Green, channel B as Red, and channel C as Blue.

Table 2-42. Input Data Channel Swap Command

I²C		USB
Read	Write	0x1A37
0x04	0x84	

Table 2-43. Input Data Channel Swap Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
0	0	Port Number 0 – Port 1 1 – Port 2	0	w
	3:1	Swap parallel interface data sub-channel: 0 - ABC = ABC No swapping of data sub-channels 1 - ABC = CAB Data sub-channels are right shifted and circularly rotated 2 - ABC = BCA Data sub-channels are left shifted and circularly rotated 3 - ABC = ACB Data sub-channels B and C are swapped 4 - ABC = BAC Data sub-channels A and B are swapped 5 - ABC = CBA Data sub-channels A and C are swapped 6 & 7 - Reserved	d4	wr
	7:4	Reserved	d0	r

2.3.3 Input Source Commands

The Input Source Selection determines the input source for the DLPC900 data display.

2.3.3.1 Port and Clock Configuration

This command selects which port the RGB data is on and which pixel clock, data enable, and syncs to use. The user must select the correct port and clock configuration according to the PCB layout routing.

Table 2-44. Port and Clock Configuration Command

I²C		USB
Read	Write	0x1A03
0x03	0x83	

Table 2-45. Port and Clock Configuration Command Definition

BYTE	BITS	DESCRIPTION (1) (2)	RESET	TYPE
0	1:0	0 - Data Port 1, Single Pixel mode 1 - Data Port 2, Single Pixel mode 2 - Data Port 1-2, Dual Pixel mode. Even pixel on port 1, Odd pixel on port 2 3 - Data Port 2-1, Dual Pixel mode. Even pixel on port 2, Odd pixel on port 1	d0	wr
	3:2	0 - Pixel Clock 1 1 - Pixel Clock 2 2 - Pixel Clock 3 3 - Reserved		
	4	0 - Data Enable 1 1 - Data Enable 2		
	5	0 - P1 VSync and P1 HSync 1 - P2 VSync and P2 HSync		
	7:6	Reserved		

(1) Single Pixel refers to the parallel data that is connected to port 1 or port 2 and the input source pixel clock that is less than 175 MHz. Both ports cannot be used simultaneously in single pixel mode.

(2) Dual Pixel refers to the parallel data that is connected to port 1 and port 2 and the input source pixel clock that is less than 141 MHz.

2.3.3.2 Input Source Configuration

The Input Source Configuration command selects the input source to be displayed by the DLPC900: 30-bit parallel port, Internal Test Pattern or flash memory. After executing this command, the host may poll the system status using I^2C commands: 0x20, 0x21, and 0x22, or the respective USB commands: 0x1A0A, 0x1A0b, and 0x1A0C.

Table 2-46. Input Source Configuration Command

I^2C		USB
Read	Write	0x1A00
0x00	0x80	

Table 2-47. Input Source Configuration Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
0	2:0	Select the input source and interface mode: 0 = Primary parallel interface with 16-bit, 20-bit, 24-bit, or 30-bit RGB or YUV data formats. 1 = Internal test pattern generator. 2 = Flash. Images are 24-bit single-frame, still images stored in flash that are uploaded on command. 3 = Solid curtain.	d0	wr
	4:3	Parallel Interface bit depth 0 = 30 bits 1 = 24 bits 2 = 20 bits 3 = 16 bits	d1	wr
	7:5	Reserved	d0	r

Note

All Pattern modes only use up to 24 bits. If a 30 bit video stream is input, the last two bits of data are for each color are not used.

2.3.3.3 Input Pixel Data Format

The Input Pixel Data Format command defines the pixel data input format to the DLPC900.

Table 2-48. Input Pixel Data Format Command

I²C		USB
Read	Write	0x1A02
0x02	0x82	

Table 2-49. Input Pixel Data Format Command Definition

BYTE	BITS	DESCRIPTION				RESET	TYPE
0	3:0	Select the pixel data format:	Supported Pixel Formats vs. Source Type			d0	wr
			Parallel	Test Pattern	Flash Image		
		0 - RGB (24 or 30 bit)	Yes	Yes	Yes		
		1 - YCrCb 4:4:4 (30 bit)	Yes	No	No		
		2 - YCrCb 4:2:2	Yes	No	Yes		
	7:4	Reserved				d0	r

2.3.3.4 Internal Test Pattern Select

When the internal test pattern is the selected input, the Internal Test Pattern Select defines the test pattern displayed on the screen. These test patterns are internally generated; therefore, all image processing is performed on the test images. The resolution of the Test Pattern is native to the attached DMD.

Table 2-50. Internal Test Pattern Select Command

I²C		USB
Read	Write	0x1203
0x0A	0x8A	

Table 2-51. Internal Test Patterns Select Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
0	3:0	Internal Test Patterns Select:	d10	wr
		0 = Solid field		
		1 = Horizontal ramp		
		2 = Vertical ramp		
		3 = Horizontal lines		
		4 = Diagonal lines		
		5 = Vertical lines		
		6 = Grid		
		7 = Checkerboard		
		8 = RGB ramp		
		9 = Color bars		
		10 = No pattern		
		11 - 15 = Reserved		
	7:4	Reserved		

2.3.3.5 Internal Test Patterns Color

When the internal test pattern is the selected input, the Internal Test Patterns Color Control defines the colors of the test pattern displayed on the screen. These test patterns are internally generated; therefore, all image processing is performed on the test images. Set up all command registers as if the test images are input from an RGB 8:8:8 external source. The foreground color setting affects all test patterns. The background color setting affects those test patterns that have a foreground and background component, such as Horizontal Lines, Diagonal Lines, Vertical Lines, Grid, and Checkerboard.

Table 2-52. Internal Test Patterns Color Command

I²C		USB
Read	Write	0x1204
0x1A	0x9A	

Table 2-53. Internal Test Patterns Color Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
1:0	9:0	Red Foreground Color intensity in a scale from 0 to 1023 0x0 = No Red Foreground color intensity ... 0x3FF = Full Red Foreground color intensity	x3FF	wr
3:2	9:0	Green Foreground Color intensity in a scale from 0 to 1023 0x0 = No Green Foreground color intensity ... 0x3FF = Full Green Foreground color intensity	x3FF	wr
5:4	9:0	Blue Foreground Color intensity in a scale from 0 to 1023 0x0 =No Blue Foreground color intensity ... 0x3FF = Full Blue Foreground color intensity	x3FF	wr
7:6	9:0	Red Background Color intensity in a scale from 0 to 1023 0x0 = No Red Background color intensity ... 0x3FF = Full Red Background color intensity	x0	wr
9:8	9:0	Green Background Color intensity in a scale from 0 to 1023 0x0 = No Green Background color intensity ... 0x3FF = Full Green Background color intensity	x0	wr
11:10	9:0	Blue Background Color intensity in a scale from 0 to 1023 0x0 = No Blue Background color intensity ... 0x3FF = Full Blue Background color intensity	x0	wr

2.3.3.6 Load Image

This command loads an image from flash memory and then displays it on the DMD. After executing this command, the host may poll the system status using I²C commands: 0x20, 0x21, and 0x22, or using the respective USB commands: 0x1A0A, 0x1A0B, and 0x1A0C.

Note

The Load Image command is a blocking command. No other commands other than the polling commands are accepted by the system until the load has finished

Table 2-54. Load Image Command

I²C		USB
Read	Write	0x1A39
0x7F	0xFF	

Table 2-55. Load Image Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
0	7:0	Image Index. Loads the image at this index. Reading this back provides the index that was loaded most recently through this command.	d0	wr

2.3.4 Image Flip

The DLPC900 supports long- and short-axis image flips to support rear- and front-projection, as well as table- and ceiling-mounted projection.

Note

If showing image from Flash, load image (I²C: 0x7F, USB: 0x1A39), this must be called to update the image flip setting.

2.3.4.1 Long-Axis Image Flip

Note

The DLPC900 only supports long axis image flip for single controller DMDs. This command is not supported when combined with a dual controller DMD.

The Long-Axis Image Flip defines whether the input image is flipped across the long axis of the DMD. If this parameter is changed while displaying a still image, then re-send the still image. If the image is not re-sent, the output image might be slightly corrupted. Figure 2-1 shows an example of a long-axis image flip. In Structured Light mode, the image flip takes effect on the next bit-plane, image, or video frame load.

Table 2-56. Long-Axis Image Flip Command

I²C		USB
Read	Write	0x1008
0x08	0x88	

Table 2-57. Long Axis Image Flip Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
0	0	Flips image along the long side of the DMD: 0 = Disable flip 1 = Enable flip	d0	wr
	7:1	Reserved	d0	r

Figure 2-1. Image Long-Axis Flip Example

2.3.4.2 Short Axis Image Flip

The Short-Axis Image Flip defines whether the input image is flipped across the short axis of the DMD. If this parameter is changed while displaying a still image, re-send the input still image. If the image is not re-sent, the output image might be slightly corrupted. Figure 2-2 shows an example of a short axis image flip. In Structured Light mode, the image flip takes effect on the next bit-plane, image, or video frame load.

Table 2-58. Short Axis Image Flip Command

I²C		USB
Read	Write	0x1009
0x09	0x89	

Table 2-59. Short-Axis Image Flip Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
0	0	Flips image along the short side of the DMD: 0 - Disable flip 1 - Enable flip	d0	wr
	7:1	Reserved	d0	r

Figure 2-2. Image Short-Axis Flip Example

2.3.5 IT6535 Power Mode

The IT6535 Power Mode command allows the user to power-down and tri-state the IT6535 digital receiver data and sync outputs. This command is ignored if the IT6535 is not present or has been disabled.

Table 2-60. IT6535 Power Mode Command

I²C		USB
Read	Write	0x1A01
0x0C	0x8C	

Table 2-61. IT6535 Power Mode Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
0	1:0	0 = Power-Down. (Outputs are tri-stated) 1 = Power-Up for HDMI input. 2 = Power-Up for DisplayPort input. 3 = Reserved	d0	wr
	7:2	Reserved.	d0	r

2.3.6 Gamma Configuration and Enable

In firmware 6.x a new command to enable and disable Gamma and select one of three pre-defined gamma tables was introduced.

Table 2-62. Gamma Configuration and Enable Command

I²C		USB
Read	Write	0x1A3B
N/A	0x61	

Table 2-63. Gamma Configuration and Enable Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
0	0	Disables or enables gamma table 0 - Disable gamma 1 - Enable gamma	d0	w
	7:1	Reserved	d0	w
1	1:0	**Stored Gama Tables (0 - 4)** [see Figure 2-3] 0 = Linear (ɣ = 1.0) 1 = Power Law 2.22 (ɣ = 2.22) 2 = Photo 3 = Enhanced 4 = Max Brightness 5 - 7 - Reserved	d0	w
	7:2	Reserved.	d0	w

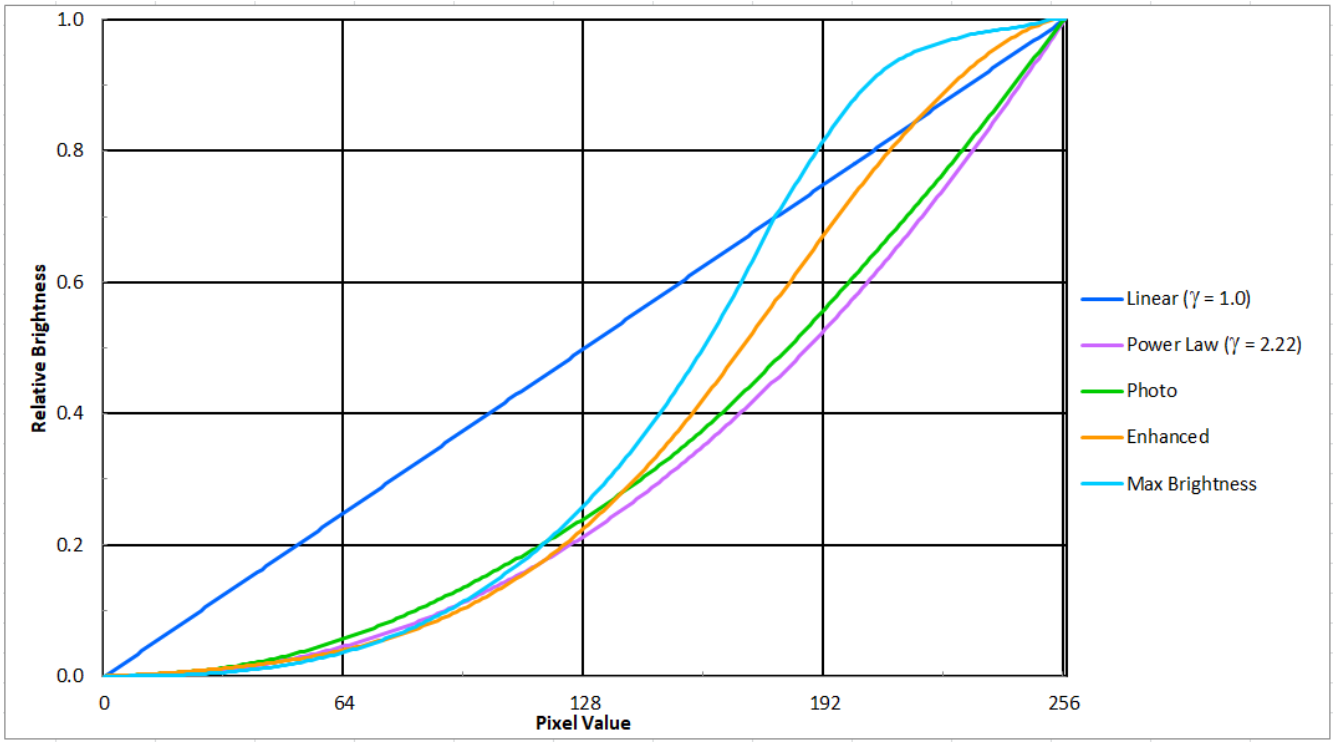

Figure 2-3. Gamma Chart

2.3.7 LED Driver Commands

LED driver operation is a function of the individual red, green, and blue LED-enable software-control parameters. The recommended order for initializing LED drivers is to:

1. Program the individual red, green, and blue LED driver currents.
2. Program the LED PWM polarity.
3. Enable the individual LED enable outputs.
4. Turn ON the DLP display sequence (see Section 2.4.1).

The LED-current software-control parameters define PWM values that drive corresponding LED current. The LED enables indicate which LEDs are activated.

CAUTION

Careful control of LED current is needed to prevent damage to LEDs. Follow all LED manufacturer recommendations and maintain LED current levels within recommended operating conditions. The setting of the LED current depends on many system and application parameters (including projector thermal design, LED specifications, selected display mode, and so forth). Therefore, the recommended and absolute-maximum settings vary greatly.

2.3.7.1 LED Enable Outputs

The DLPC900 offers three sets of pins to control the LED enables:

- LEDR_EN for the red LED
- LEDG_EN for the green LED
- LEDB_EN for the blue LED

After reset, all LED enables are placed in the inactive state until the board initializes.

Table 2-64. LED Enable Outputs Command

I²C		USB
Read	Write	0x1A07
0x10	0x90	

Table 2-65. LED Enable Outputs Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
0	0	Red LED Enable 0 - Red LED is disabled 1 - Red LED is enabled	d0	wr
	1	Green LED Enable 0 - Green LED is disabled 1 - Green LED is enabled	d0	wr
	2	Blue LED Enable 0 - Blue LED is disabled 1 - Blue LED is enabled	d0	wr
	3	LED Enable Control 0 - All LED enables are controlled by bits 2:0 and ignore Sequencer control 1 - All LED enables are controlled by the Sequencer	d1	wr
	7:4	Reserved	d0	r

2.3.7.1.1 LED PWM Polarity

The LED PWM Polarity command sets the polarity of all PWM signals. This command must be issued before powering up the LED drivers.

Table 2-66. LED PWM Polarity Command

I²C		USB
Read	Write	0x1A05
0x0B	0x8B	

Table 2-67. LED PWM Polarity Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
0	1:0	Polarity of PWM signals (1) 0 - Normal polarity. PWM 0 value corresponds to no current while PWM 255 value corresponds to maximum current. 1 - Inverted polarity. PWM 0 value corresponds to maximum current while PWM 255 value corresponds to no current.	d0	wr
	7:2	Reserved	d0	r

(1) Depending on the LED driver design, the polarity chosen may have an opposite effect.

2.3.7.2 LED Driver Current

This parameter controls the pulse duration of the specific LED PWM modulation output pin. The resolution is 8 bits and corresponds to a percentage of the LED current. The PWM value can be set from 0 to 100% in 256 steps. If the LED PWM polarity is set to normal polarity, a setting of 0xFF gives the maximum PWM current. The LED current is a function of the specific LED driver design.

Table 2-68. LED Driver Current Command

I²C		USB
Read	Write	0x0B01
0x4B	0xCB	

CAUTION

Take care when using this command. Improper use of this command can lead to damage to the system. The setting of the LED current depends on many system and application parameters (including thermal design, LED specifications, selected display mode, and so forth). Therefore, recommended and absolute-maximum settings vary greatly.

Table 2-69. LED Driver Current Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
0	7:0	Red LED PWM current control Valid range, assuming normal polarity of PWM signals, is: 0x00 (0% duty cycle → Red LED driver generates no current) to 0xFF (100% duty cycle → Red LED driver generates maximum current) The current level corresponding to the selected PWM duty cycle is a function of the specific LED driver design and thus varies by design.	x97	wr
1	7:0	Green LED PWM current control Valid range, assuming normal polarity of PWM signals, is: 0x00 (0% duty cycle → Green LED driver generates no current) to 0xFF (100% duty cycle → Green LED driver generates maximum current) The current level corresponding to the selected PWM duty cycle is a function of the specific LED driver design and thus varies by design.	x78	wr
2	7:0	Blue LED PWM current control Valid range, assuming normal polarity of PWM signals, is: 0x00 (0% duty cycle → Blue LED driver generates no current) to 0xFF (100% duty cycle → Blue LED driver generates maximum current) The current level corresponding to the selected PWM duty cycle is a function of the specific LED driver design and thus varies by design.	x7D	wr

2.3.7.3 Minimum LED Pulse Width in microseconds (µs)

This parameter gets/sets the minimum LED pulse width restriction in microseconds (µs) for the implementation of high speed illumination-modulated 8-bit patterns. Setting a value of 0 indicates that no illumination modulation is performed.

Table 2-70. Minimum LED Pulse Width in microseconds (µs) Command

I²C		USB
Read	Write	0x1A41
N/A	0x62	

Table 2-71. Set Minimum LED Pulse Width in microseconds (µs) Command Definition Table

BYTE	BITS	DESCRIPTION	RESET	TYPE
0	7:0	Minimum pulse width in microseconds	d0	wr

2.3.7.4 Minimum LED Pulse Width in nanoseconds (ns)

Introduced in firmware 6.x, this parameter gets/sets the minimum LED pulse width restriction in nanoseconds (ns) for DMDs that support the implementation of high speed illumination-modulated 16-bit patterns. Setting a value of 0 indicates that no illumination modulation is performed.

Table 2-72. Set Minimum LED Pulse Width in ns Command

I²C		USB
Read	Write	0x1A43
N/A	0x67	

Table 2-73. Set Minimum LED Pulse Width in ns Command Definition Table

BYTE	BITS	DESCRIPTION	RESET	TYPE
0	7:0	Minimum pulse width in nanoseconds	d0	wr

2.3.7.5 Get Minimum LED Pattern Exposure in microseconds (µs)

This parameter gets the stored minimum LED pattern exposure, in microseconds.

Table 2-74. Get Minimum LED Pattern Exposure in µs Command

I²C		USB
Read	Write	0x1A41
0x63	N/A	

Table 2-75. Get Minimum LED Pattern Exposure in µs Command Definition Table

BYTE	BITS	DESCRIPTION	RESET	TYPE
0-15	7:0	16 bytes are returned. Each two bytes represent the Minimum Pattern Exposure for each bit depth (from 0 to 8) in microseconds.	d0	r

2.3.7.6 Get Minimum LED Pattern Exposure in nanoseconds (ns)

This parameter gets the stored minimum LED pattern exposure, in nanoseconds.

Table 2-76. Get Minimum LED Pattern Exposure in ns Command

I²C		USB
Read	Write	0x1A43
0x65	N/A	

Table 2-77. Get Minimum LED Pattern Exposure in ns Command Definition Table

BYTE	BITS	DESCRIPTION	RESET	TYPE
0-15	7:0	16 bytes are returned. Each two bytes represent the Minimum Pattern Exposure for each bit depth (from 0 to 8) in microseconds.	d0	r

2.3.8 GPIO Commands

DLPC900 offers 9 general-purpose input/output pins (GPIO). Some of these pins can be configured for PWM output, PWM input, or clock output functionality. By default, all pins are configured as GPIO inputs.

2.3.8.1 GPIO Configuration

The GPIO Configuration command enables GPIO functionality on a specific set of DLPC900 pins. The command sets their direction, output buffer type, and output state.

Table 2-78. GPIO Configuration Command

I²C		USB
Read	Write	0x1A38
0x44	0xC4	

Table 2-79. GPIO Configuration Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
0	7:0	GPIO selection. See Table 2-80 for description of available pins	d0	wr
1	0	Output state 0 = Low 1 = High	d0	wr
	1	0 – Configure pin as input 1 – Configure pin as output	d0	wr
	2	0 – Configure as normal mode 1 – Configure as open drain mode	d0	wr
	7:3	Reserved	d0	r

Table 2-80. GPIO Selection

GPIO Selection	DLPC900 GPIO Pin	Function	Alternate Function
0	GPIO_PWM_0	GPIO	PWM Output
1	GPIO_PWM_1	GPIO	PWM Output
2	GPIO_PWM_2	GPIO	PWM Output
3	GPIO_PWM_3	GPIO	PWM Output
4	GPIO_4	GPIO	None
5	GPIO_5	GPIO	None
6	GPIO_6	GPIO	None
7	GPIO_7	GPIO	None
8	GPIO_8	GPIO	None

2.3.8.2 GPIO Clock Configuration

DLPC900 supports one clock output capability. The OCLKA Clock Configuration command enables the clock output functionality and sets the clock frequency.

Table 2-81. GPIO Clock Configuration Command

I²C		USB
Read	Write	0x0807
0x48	0xC8	

Table 2-82. GPIO Clock Configuration Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
0	0	Clock Selection 0 = OCLKA 1 = Reserved	d0	wr
	7:1	Reserved	d0	r
1	0	Clock Functionality Disable 0 = Disable clock functionality on selected pin 1 = Enable clock functionality on selected pin	d0	wr
	7:1	Reserved	d0	r
2	7:0	Clock Divider. Allowed values in the range of 2 to 127. Output frequency = 100 MHz / (Clock Divider) 0x0 = Reserved 0x1 = Reserved 0x2 = 2 ... 0x7F = 127 0xFF:0x80 = Reserved	x7F	wr

2.3.8.3 GPIO Busy

Added in FW 6.x the GPIO Busy command queries the system to poll if the GPIO subsystem is busy.

Table 2-83. GPIO Busy Command

I²C		USB
Read	Write	0x1A5E
0x5E	N/A	

Table 2-84. GPIO Configuration Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
0	0	0 = not busy 1 = busy	d0	r
	7:1	Reserved	d0	r

2.3.9 Pulse Width Modulated (PWM) Control

DLPC900 provides four general-purpose PWM channels that can be used for a variety of control applications, such as fan speed. If the PWM functionality is not needed, these signals can be programmed as GPIO pins. To enable the PWM signals:

1. Program the PWM signal using the PWM Setup command.
2. Enable the PWM signal with the PWM Enable command.

2.3.9.1 PWM Setup

The PWM Setup command sets the clock period and duty cycle of the specified PWM channel. The PWM frequency and duty cycle is derived from an internal 18.67 MHz clock. To calculate the desired PWM period, divide the desired clock frequency from the internal 18.67 MHz clock. For example, a PWM frequency of 2 kHz, requires a 18666667 / 2000 = 9333 or 0x2475.

Table 2-85. PWM Setup Command

I²C		USB
Read	Write	0x1A11
0x41	0xC1	

Table 2-86. PWM Setup Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
0	1:0	PWM Channel Output Select 0 - PWM channel 0 (GPIO_PWM_0) 1 - PWM channel 1 (GPIO_PWM_1) 2 - PWM channel 2 (GPIO_PWM_2) 3 - PWM channel 3 (GPIO_PWM_3)	d0	wr
	7:2	Reserved	d0	r
4:1	31:0	Clock Period in increments of 53.57 ns. Clock Period = (value + 1) × 53.5 ns	d0	wr
5	6:0	Duty Cycle = (value + 1)% Value range is 1% to 99%	d0	wr
	7	Reserved	d0	r

2.3.9.2 PWM Enable

After the PWM Setup command configures the clock period and duty cycle, the PWM Enable command activates the PWM signals.

Table 2-87. PWM Enable Command

I²C		USB
Read	Write	0x1A10
0x40	0xC0	

Table 2-88. PWM Enable Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
0	1:0	PWM Channel Output Select 0 - PWM channel 0 (GPIO_PWM_0) 1 - PWM channel 1 (GPIO_PWM_1) 2 - PWM channel 2 (GPIO_PWM_2) 3 - PWM channel 3 (GPIO_PWM_3)	d0	wr
	6:2	Reserved	d0	r
	7	PWM Channel Enable 0 - Disable selected PWM Channel 1 - Enable selected PWM Channel	d0	wr

2.3.10 Batch File Commands

During power-up and initialization or during normal operation, the DLPC900 can be commanded to execute a batch file containing a set of commands. The set of commands are created and saved in a text file. The text file then becomes an additional part of the firmware and is uploaded into the flash memory. The user can also specify a default batch file that the DLPC900 executes during its power-up sequence.

2.3.10.1 Batch File Name

This is a read command that returns the name of the given batch file index. This is useful for listing the set of batch files available for the user to execute. To list all the batch file names, iterate through all numbers from 0 to n until an error is returned, which identifies the end of the list.

Table 2-89. Batch File Name Command

I²C	USB
Read	0x1A14
0x3A	

Table 2-90. Batch File Name Command Definition

BYTES	BITS	DESCRIPTION	RESET	TYPE
0	7:0	Batch Command Index (Read parameter)	d0	w
15:1	All	Batch Command Name String (Read result)	d0	r

2.3.10.2 Batch File Execute

This command executes all the commands in a given batch file at the given index.

Table 2-91. Batch File Execute Command

I²C	USB
Write	0x1A15
0xBB	

Table 2-92. Batch File Execute Command Definition

BYTES	BITS	DESCRIPTION	RESET	TYPE
0	7:0	Batch command index to be executed.	d0	w

2.3.10.3 Batch File Delay

This command is useful for introducing the given amount of delay between batch commands within the same batch file. This command by itself does not perform any action.

Table 2-93. Batch File Delay Command

I²C	USB
Write	0x1A16
0xBC	

Table 2-94. Batch File Delay Command Definition

BYTES	BITS	DESCRIPTION	RESET	TYPE
3:0	31:0	Delay to be introduced in milliseconds	d0	w

2.3.10.4 Batch File Example

The following table shows an example of a batch file. Only command descriptors with parameters are allowed in the batch file.

Table 2-95. Batch File Example

COMMAND DESCRIPTOR	PARAMETERS	DESCRIPTION
VIDEO_CONT_SEL	0x01	Power on the IT6535 for HDMI input.
DELAY	0xC8	Delay 200 ms.
CHANNEL_SWAP	0x04	Select input data channel swap to ABC = BAC
FLIP_LONG	0x01	Flip the image on long axis.

When saving the batch file to a text file, only save the command descriptor and the parameters as shown below with a colon after the command descriptor and space delimited. See Appendix B for a list of the supported command descriptors. Once the batch file has been created and saved as a text file, see the DLPC900 LightCrafter Dual Controller or Single Controller EVM User's Guide on how to add batch files to the firmware.

```
VIDEO_CONT_SEL: 0x01
DELAY: 0xC8
CHANNEL_SWAP: 0x04
FLIP_LONG: 0x01
```

2.4 Display Mode Commands

The DLPC900 display consists of several parameters which dictate the loading of the DMD and the control of PWM to the LEDs. The DLPC900 supports four main display modes:

- Video mode
- Video Pattern mode
- Pre-Stored Pattern mode
- Pattern On-The-Fly mode

The Display Mode Selection command (Section 2.4.1) selects between these modes.

In Video mode, the DLPC900 30-bit RGB interface supports up to the native resolution of the attached DMD. The DLPC900 processes the digital input image and converts the data into the appropriate format.

The DLPC900 offers scaling and cropping functions to appropriately display resolutions on single controller DMDs.

The DLPC900 combined with dual controller DMDs does not support scaling or cropping functions.

In the latter three modes, the DLPC900 provides high-speed pattern rates. These modes support only 24-bit data input through the DLPC900 RGB interface, from flash memory, or dynamically loaded using Pattern On-The-Fly modes. These modes are well-suited for techniques such as structured light, additive manufacturing, or digital exposure. The DLPC900 also has the capability to display a set of patterns and signal a camera to capture when these patterns are displayed. Figure 2-4 shows the DLPC900 Single Controller DMD block diagram and Figure 2-5 shows the DLPC900 Dual Controller DMD block diagram. The main functional blocks for the four display modes are shown in these diagrams.

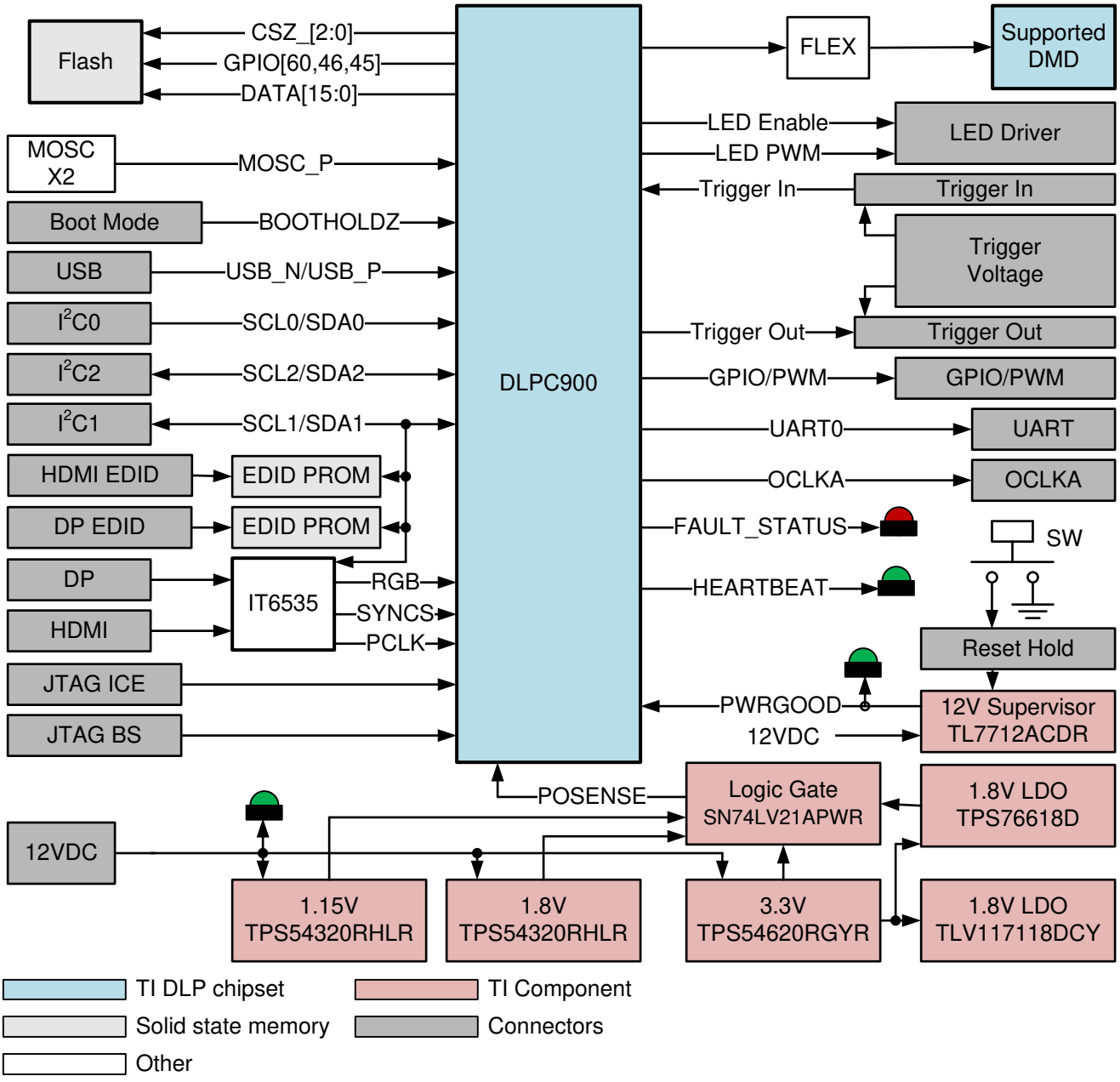

Figure 2-4. DLPC900 Single Controller System Block Diagram

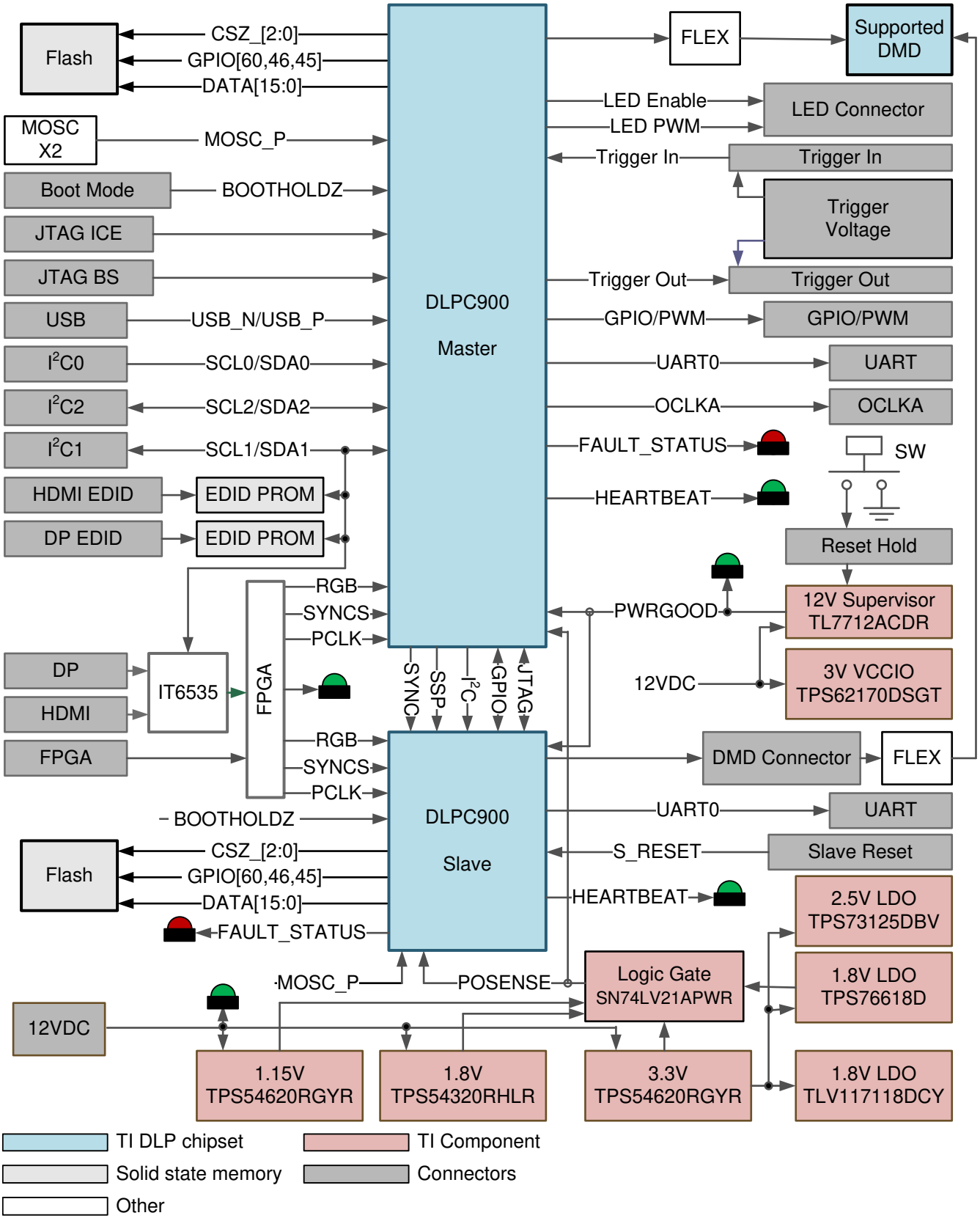

Figure 2-5. DLPC900 Dual Controller System Block Diagram

In Video mode, the DLPC900 operates on a per-frame basis where it takes the input data and appropriately allocates it in a frame. For example, a 24-bit RGB input image is allocated into a 60-Hz frame by dividing each color (red, green, and blue) into specific percentages of the frame. Therefore, for a 40% red, 45% green, and 15% blue ratio, the red, green, and blue colors would have a 6.67-, 7.5-, and 2.54-ms time slot allocated, respectively. Because each color has an 8-bit depth, each color time slot is further divided into bit-planes, as shown in Figure 2-6. A bit-plane is the two-dimensional arrangement of one bit extracted from all the pixels in the full color 2D image.

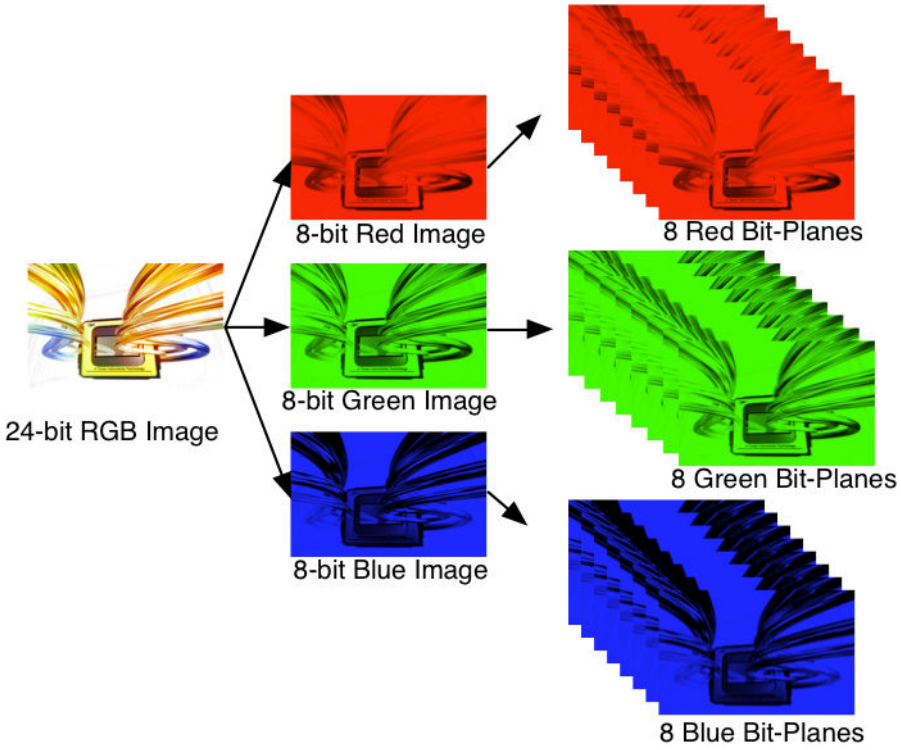

Figure 2-6. Bit-Planes of a 24-Bit RGB Image

The length of each bit-plane in the time slot is weighted by the corresponding power of two of its binary representation. This provides a binary pulse-width modulation of the image. For example, a 24-bit RGB input has three colors with 8-bit depth each. Each color time slot is divided into eight bit-planes, with the sum of the weight of all bit planes in the time slot equal to 255. See Figure 2-7 for an illustration of this partition of the bits in a frame.

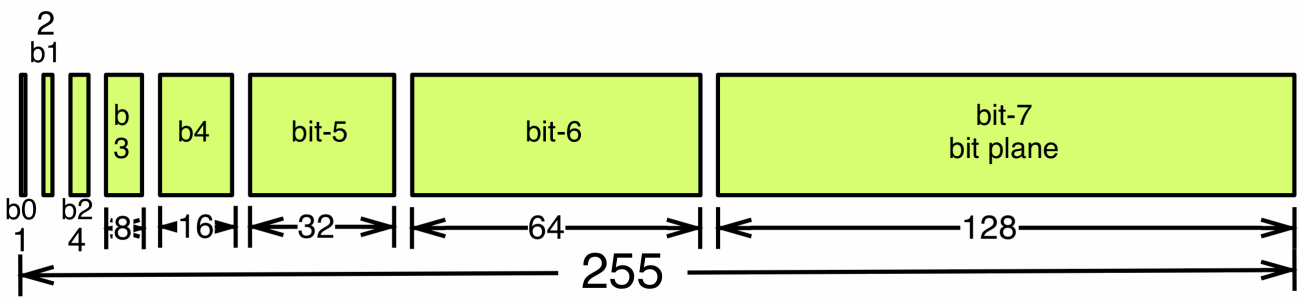

Figure 2-7. Bit Partition in a Frame for an 8-Bit Monochrome Image

Therefore, a single video frame is composed of a series of bit-planes. Because the DMD mirrors can be either on or off, an image is created by turning on the mirrors corresponding to the bit set in a bit-plane. With binary pulse width modulation, the intensity level of the color is reproduced by controlling the amount of time the mirror is on. For a 24-bit RGB frame image inputted to the DLPC900 controller, the DLPC900 controller creates 24 bit-planes, stores them in internal embedded DRAM, and sends them to the DMD, one bitplane at a time. The bit weight controls the illumination intensity of the bit-plane where smaller the bit weight is the less intense the bit-plane becomes. To improve image quality in video frames, these bit-planes, time slots, and color frames are shuffled and interleaved within the pixel processing functions of the DLPC900 controller.

For other applications where one-to-one pixel mapping to the DMD micromirror is required, the scaling, cropping, and pixel processing functions are disabled and a specific set of patterns is used. The bit-depth of the pattern is then allocated into the corresponding binary weighted time slots. Furthermore, output trigger signals are also synchronized with these time slots to indicate when the image is displayed. For structured light applications, this mechanism provides the capability to display a set of patterns and signal a camera to capture these patterns overlaid on an object.

2.4.1 Display Mode Selection

The Display Mode Selection command switches the internal image processing functions of the DLPC900 to operate in the mode selected. After executing this command, the host may poll the system status using I^2C commands: 0x20, 0x21, and 0x22 or the respective USB commands: 0x1A0A, 0x1A0B, and 0x1A0C.

Table 2-96. Display Mode Selection Command

I^2C		USB
Read	Write	0x1A1B
0x69	0xE9	

Table 2-97. Display Mode Selection Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
0	1:0	0 = Video mode 1 = Pre-stored pattern mode (Images from flash) 2 = Video pattern mode 3 = Pattern On-The-Fly mode (Images loaded through USB/I^2C)	d1	wr
	7:2	Reserved	d0	r

To change to Video pattern mode (2), the system must first change to Video mode (0) with the desired source enabled and sync must be locked before switching to Video Pattern mode. Once sync lock is achieved it takes about 300 ms to complete the transition to Video Pattern mode. If the display mode is read back before this time, it may not return the correct mode.

2.4.1.1 Video Mode Resolution

When Display Mode is set to Video Mode see the DLPC900 data sheet for resolutions supported with various DLPC900 / DMD combinations and any reduced blanking requirements.

2.4.1.2 Input Display Resolution

The Input Display Resolution command defines the active input resolution and active output (displayed) resolution. This command provides the option to define a subset of active input frame data using pixel (column) and line (row) counts relative to the source-data enable signal (DATEN). In other words, this feature allows the source image to be cropped as the first step in the processing chain. After executing this command, the host may poll the system status using I^2C commands: 0x20, 0x21, and 0x22, or the respective USB commands: 0x1A0A, 0x1A0B, and 0x1A0C. **This command is not supported when DLPC900 is combined with a Dual DLPC900 DMD**.

Table 2-98. Input Display Resolution Command

I^2C		USB
Read	Write	0x1000
0x7E	0xFE	

Table 2-99. Input Display Resolution Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
1:0	15:0	Input image, first active pixel (column) of cropped area	d0	wr
3:2	15:0	Input image, first active line (row) of cropped area	d0	
5:4	15:0	Input image vertical resolution, pixels (columns) per line (row) of cropped area	d0	
7:6	15:0	Input image horizontal resolution, lines (rows) per frame of cropped area	d0	
9:8	15:0	Output image, first active pixel (column) of displayed image	d0	
11:10	15:0	Output image, first active line (row) of displayed image	d0	
13:12	15:0	Output image horizontal resolution, pixels (columns) per line (row)	d (1)	
15:14	15:0	Output image vertical resolution, lines (rows) per frame	d (2)	

(1) Maximum horizontal resolution depends on attached DMD.
(2) Maximum vertical resolution depends on attached DMD.

2.4.1.3 DMD Block Load

The DMD Block Load command allows the user to specify which of the DMD blocks are active. Only adjacent blocks are allowed. Mirrors in blocks that are not active are set to their off state prior to the pattern sequence running. Selecting a reduced number of active DMD blocks allows for an increase in pattern speeds. See Table 2-102.

Block Load is only applicable for 1-bit depth patterns. The entire 1-bit pattern data must be sent to the controller when using video pattern mode, pattern on the fly, or pre-stored pattern mode. The controller loads the selected block(s) based on the rows selected in Block Load.

Note

The performance of mirrors in blocks that are not active are affected by prolonged use of being in the off state. To optimize the mirrors, enable DMD Idle Mode as often as possible. This mode provides a 50/50 duty cycle across the entire DMD mirror array, where the mirrors are continuously flipped between the on and off states. See command in Section 2.4.1.5.

Table 2-100. DMD Block Load Command

I^2C		USB
Read	Write	0x1A40
0x60	0xE0	

Table 2-101. DMD Block Load Command Definition

BYTE	BITS	DESCRIPTION (1)	RESET	TYPE
0	4:0	Start block. Range 0x0 - 0x0E on DMDs with 15 blocks or 0x0 - 0xF on DMDs with 16 blocks	0	wr
	7:5	Reserved	0	r
1	4:0	Number of blocks. Range 0x1 - 0x0E on DMDs with 15 blocks or 0x1 - 0xF on DMDs with 16 blocks	0xF or 0x10	wr
	7:5	Reserved	0	r

(1) When short and long axes are disabled, block 0 begins at pixel (0,0) on the DMD.

Table 2-102. DMD Block Load Minimum Exposure Times

NUMBER OF DMD ACTIVE BLOCKS(1)	Block Load Minimum Exposure Time (µs)			
	DLP6500	DLP9000	DLP670S	DLP500YX
1	24	24	27	30
2	45	42	27	30
3	45	42	27	30
4	45	42	33	30
5	48	45	38	34
6	54	51	38	38
7	60	56	49	42
8	66	61	55	46
9	72	67	61	50
10	78	72	66	54
11	84	77	72	58
12	90	83	77	62
13	96	88	83	-
14	101	93	89	
15	105	99	94	
16	-	105	100	

(1) See DMD data sheet for number of blocks and rows per block

2.4.1.4 Minimum Exposure Times

Table 2-103. Minimum Exposure in Any Pattern Mode (1)

BIT DEPTH	Minimum Exposure in Any Pattern Mode (µs)			
	DLP6500	DLP9000	DLP670S	DLP500YX
1	105	105	100	62
2	304	304	343	184
3	394	380	438	269
4	823	733	768	458
5	1215	1215	1299	682
6	1487	1487	1488	807
7	1998	1998	2000	1083
8	4046	4046	4046	2263
10	-	-	-	10363
12				41452
14				165807
16				663225

(1) The maximum pattern rate for pre-loaded patterns with external trigger are slightly less than the listed values for pre-loaded patterns with internal trigger.

2.4.1.5 DMD Idle Mode

It is strongly recommended that anytime the DMD is idle and not actively projecting data that the DMD Idle Mode be enabled to assist in maximizing DMD lifetime. This mode enables a 50/50 duty cycle pattern sequence, where the entire mirror array is continuously flipped periodically between the on and off states. Whenever this mode is enabled, the LED Enable outputs are disabled to prevent illumination on the DMD. When operating with a subset of DMD blocks, **enable this mode as often as possible**. For example, whenever the system is idle, between exposures if the application allows for it, or when the exposure pattern sequence is stopped. To enable this mode, the pattern sequences must first be stopped. To restart the pattern sequence, this mode must be disabled. This mode can be enabled in any operating mode except for Video Mode. This mode can also be enabled to optimize the mirrors that experience prolonged use of being in the on or off states when all DMD blocks are active.

Note

Once DMD Idle Mode is enabled, a minimum of three seconds must elapse before disabling it.

Note

If the system is going to be unused for long periods of time, consider using the Standby state [Power Mode =1] instead (see Table 2-35).

Table 2-104. DMD Idle Mode Command

I^2C		USB
Read	Write	0x0201
0x0D	0x8D	

Table 2-105. DMD Idle Mode Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
0	0	0 - Idle mode disabled 1 - Idle mode enabled	d0	wr
	7:1	Reserved	0	r

2.4.2 Image Header

Precede the image data by the image header (48 Bytes) shown in Table 2-106.

Table 2-106. Image Header

NUMBER OF BYTES	DESCRIPTION
4	Signature (53 70 6C 64)
2	Image width (for dual controller DMDs this is half of the full width) See Section 2.4.4.4.2
2	Image height
4	Number of bytes in the encoded image data
8	Reserved (FF FF FF FF FF FF FF FF)
4	Background color (BB GG RR 00)
1	Reserved (00)
1	Compression 0 – Uncompressed 1 – RLE compression 2 – Enhanced RLE compression
1	Reserved (01)
21	Reserved (00…)

2.4.3 Pattern Image Compression

In order to minimize Flash storage requirements, it is recommended (but not required) that pattern images be stored in a compressed format. The compression format supported by the DLPC900 is a subset of BMP Run-Length Encoding (RLE). The DLPC900 is able to perform the decompression of pattern images as they are loaded from external flash or when using Pattern On-The-Fly mode to its internal memory. The DLPC900 can also perform no decompression if the images are not compressed.

For most efficient storage and compression of images, pack stored images into groups of 24-bit RGB bitmap images.

Note

Compressed images must be stored right side up instead of upside down as in standard BMP format images.

Note

With RLE, there is always a question of whether the compressed image is larger or smaller than the uncompressed image. The method to decide which to use is left up to the programmer.

2.4.3.1 Run-Length Encoding

Table 2-107 defines the RLE Control Bytes recognized by the DLPC900. The DLPC900 firmware automatically decompresses the image when operating in Pre-Stored Pattern Mode or Pattern On-The-Fly Mode.

Table 2-107. RLE Control Bytes

CONTROL BYTE (n)	COLOR BYTE (c)	RESULT
0	0	End-of-Line
0	1	End-of-Image (required)
0	>= 2	Uncompressed. The next c pixels are uncompressed
n > 0	n/a	Repeat; Repeat the next RGB pixel (or the next dual y/c pixel pair) n times

2.4.3.1.1 RLE Compression Example

Table 2-108 shows the hexadecimal values of a 2-line packed 24-bit compressed bitmap. The compressed data on the left is stored sequentially in Flash memory. The DLPC900 firmware automatically expands the data as shown on the right which is stored in internal memory.

Table 2-108. RLE Compression Example

COMPRESSED DATA (HEX)	EXPANDED DATA (HEX)
03 040506	040506 040506 040506
05 777777	777777 777777 777777 777777 777777
00 03 040506 070809 0A0B0C	040506 070809 0A0B0C
02 789ABC	789ABC 789ABC
00 00	(End-of-Line Command)
00 00 00	00 00 00 00 00 00 00 00 (End-of-Line Padding)
07 1D1E1F	1D1E1F 1D1E1F 1D1E1F 1D1E1F 1D1E1F 1D1E1F 1D1E1F
06 212223	212223 212223 212223 212223 212223 212223
00 01	(End-of-File command)
00 01 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 (End-of-Image Padding)

2.4.3.2 Enhanced Run-Length Encoding

To achieve higher compression ratios, this compression format takes advantage of the similarities from line-to-line and uses one or two bytes to encode the length. Table 2-109 defines the RLE Control Bytes recognized by the DLPC900. The DLPC900 firmware automatically decompresses the image when operating in Pre-Stored Pattern Mode or Pattern On-The-Fly Mode.

Table 2-109. Enhanced RLE Control Bytes

CONTROL BYTE 1	CONTROL BYTE 2	CONTROL BYTE 3	RESULT
0	0	n/a	End of Image
0	1	*n*	Copy *n* pixels from previous line
0	*n* > 1	n/a	*n* uncompressed sequence of pixels
n > 1	n/a	n/a	Repeat following pixel *n* times

If *n* is < 128 then encode it with 1 byte.

If *n* is >= 128 then encode it with 2 bytes in the following manner:

- Byte0 = (n & 0x7F) | 0x80
- Byte1 = (n >> 7)
- Example: Number 0x1234 is encoded as 0xB4, 0x24

2.4.3.2.1 Enhanced RLE Compression Example

Table 2-110 shows an example of this RLE compression.

Table 2-110. Enhanced RLE Compression Example

COMPRESSED DATA (HEX)	EXPANDED DATA (HEX)
03 040506	040506 040506 040506
05 777777	777777 777777 777777 777777 777777
00 03 040506 070809 0A0B0C	040506 070809 0A0B0C
82 01 789ABC	789ABC 789ABC … (513 times)
00 00	(End of line) (1)
01 010203	010203
00 01 09	040506 040506 777777 777777 777777 777777 777777 040506 070809
00 01 00	(End-of-Image Padding)

(1) End-of-Line Command and End-of-Line Padding is optional for this RLE compression.

2.4.3.2.2 End of Image Padding

Note

End all padding on a 4 byte boundary.

2.4.4 Pattern Display Commands

In pattern display modes 0, 2, and 3, the DLPC900 supports 1-, 2-, 3-, 4-, 5-, 6-, 7-, and 8- bit images streamed through the 24-bit RGB parallel interface, pre-stored patterns in the flash memory, or dynamically with Pattern On-The-Fly. The following commands are only supported in display modes 1, 2, and 3:

- Trigger Commands
- LED Enable Delay Commands
- Pattern Display Commands
- Pattern On-The-Fly Commands

Note

If the pattern display is already active, it must be stopped using I^2C command 0x65 or USB 0x1A24 before calling these commands.

2.4.4.1 Trigger Commands

To synchronize a camera with the displayed patterns, the DLPC900 supports three pattern modes:

- Video Pattern Mode (applicable when pattern data from RGB parallel port):
  - VSYNC used as trigger input.
  - TRIG_OUT1 frames the exposure time of the pattern.
  - TRIG_OUT2 : marks the beginning of each pattern start with 20-µs pulse. This can be selectively disabled for individual patterns.
- Pre-Stored Pattern Mode (applicable for pattern data from flash):
  - TRIG_IN1 advances to next pattern, while TRIG_IN2 starts and pauses the pattern sequence.
  - TRIG_OUT1 frames the exposure time of the pattern.
  - TRIG_OUT2 : marks the beginning of each pattern start with 20-µs pulse. This can be selectively disabled for individual patterns.
- Pattern On-The-Fly Mode (patterns downloaded over USB/I^2C)
  - Triggers are the same as Pre-Stored Pattern Mode

Figure 2-8 shows an example in video pattern mode. The VSYNC starts the pattern sequence display. The pattern sequence consists of a series of four patterns followed by a series of three patterns and then repeats. The first pattern sequence consists of P1, P2, P3, and P4. The second pattern sequence consists of P5, P6, and P7. TRIG_OUT_1 frames each pattern exposed while TRIG_OUT_2 is user programmable and in this example, indicates the start of each pattern in the sequence. If the pattern sequence is configured without dark time between patterns, then the TRIG_OUT_1 output would be high enough for the entire pattern sequence. This example uses internal triggering, so TRIG_IN signals are not used.

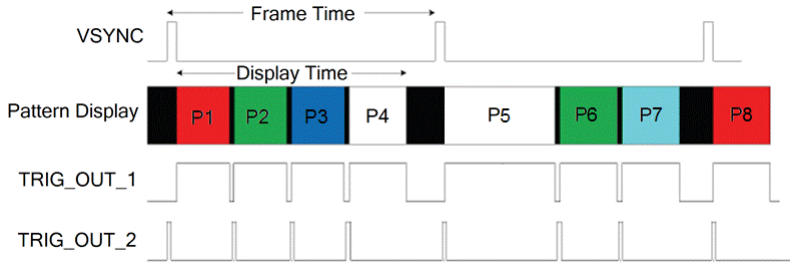

Figure 2-8. Video Pattern Mode Timing Diagram Example

Figure 2-9 shows an example in pre-stored pattern mode. Pattern sequences of four are displayed. TRIG_OUT_1 frames each pattern exposed, while TRIG_OUT_2 is user programmable and in this example, indicates the start of each pattern in the sequence. If the pattern sequence is configured without dark time between patterns, then the TRIG_OUT_1 output would be high for the entire pattern sequence. This example uses internal triggering, so TRIG_IN signals are not used.

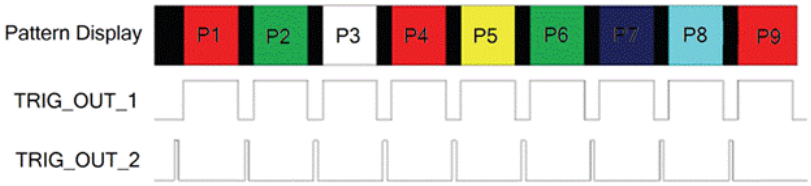

Figure 2-9. Pre-Stored Pattern Mode Timing Diagram Example

2.4.4.1.1 Trigger Out 1

The Trigger Out 1 command sets the polarity, rising edge delay, and falling edge delay of the TRIG_OUT1 signal. The delays are compared to when the pattern is displayed on the DMD. Before executing this command, stop the current pattern sequence.

Table 2-111. Trigger Out 1 Command

I²C		USB
Read	Write	0x1A1D
0x6A	0xEA	

Table 2-112. Trigger Out 1 Command Definition

BYTE	BITS	DESCRIPTION (3)	RESET	TYPE
0	0	0 = Non inverted trigger output (1)	d0	wr
		1 = Inverted trigger output (2)		
	7:1	Reserved	d0	r
2:1	15:0	Trigger output Raising Edge delay in micro seconds (int16 number) Valid Range : -20 to 20000 (4)	d0	wr
4:3	15:0	Trigger output Falling Edge delay in micro seconds (int16 number) Valid Range : -20 to 20000 (4)	d0	wr

(1) When non inverted output is selected, the rising edge must be less than or equal to the falling edge.
(2) When inverted output is selected, the rising edge must be greater than or equal to the falling edge.
(3) Minimum pulse width is 20 µs.
(4) The minimum delay is affected when the number of active blocks is reduced. The formula to calculate the minimum delay is: – (min_exposure – 5) µs. See Table 2-102 for the min_exposure for the number of active DMD blocks.

2.4.4.1.2 Trigger Out 2

The Trigger Out 2 Control command sets the polarity and rising edge delay of the TRIG_OUT2 signal. The delay is compared to when the pattern is displayed on the DMD. Before executing this command, stop the current pattern sequence.

Table 2-113. Trigger Out 2 Command

I²C		USB
Read	Write	0x1A1E
0x6B	0xEB	

Table 2-114. Trigger Out 2 Command Definition

BYTE	BITS	DESCRIPTION (3)	RESET	TYPE
0	0	0 = Non inverted trigger output (1)	d0	wr
		1 = Inverted trigger output (2)		
	7:1	Reserved	d0	r
2:1	15:0	Trigger output Raising Edge delay in micro seconds (int16 number) Valid Range : -20 to 20000 (4)	d0	wr
4:3	15:0	Trigger output Falling Edge delay in micro seconds (int16 number) Valid Range : -20 to 20000 (4)	d0	wr

(1) When non inverted output is selected, the rising edge must be less than the falling edge.
(2) When inverted output is selected, the rising edge must be greater than the falling edge.
(3) Minimum pulse width is 20 µs.
(4) The minimum delay is affected when the number of active blocks is reduced. The formula to calculate the minimum delay is: – (min_exposure – 5) µs. See Table 2-102 for the min_exposure for the number of active DMD blocks.

2.4.4.1.3 Trigger In 1

The Trigger In 1 command sets the rising edge delay of the TRIG_IN1 signal compared to when the pattern is displayed on the DMD. The polarity of TRIG_IN_ is set in the lookup table of the pattern sequence. Before executing this command, stop the current pattern sequence.

Table 2-115. Trigger In 1 Command

I²C		USB
Read	Write	0x1A35
0x79	0xF9	

Table 2-116. Trigger In 1 Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
1:0	15:0	Trigger 1 delay in micro seconds. This is the time after which the pattern is displayed from trigger active edge. A minimum delay of 104 µs is required by the HW	d105	wr
2	0	0 – Pattern advances on rising edge 1 – Pattern advances on falling edge	d0	wr
	7:1	Reserved	d0	r

2.4.4.1.4 Trigger In 2

In Video Pattern and Pre-Stored Pattern modes, the TRIG_IN2 acts as a start or stop signal. If the sequence was not already started by a software command, the rising edge on the TRIG_IN2 signal input starts or resumes the pattern sequence. If the pattern sequence is active, the falling edge on the TRIG_IN2 signal input stops the pattern sequence. Before executing this command, stop the current pattern sequence.

Table 2-117. Trigger In 2 Command

I²C		USB
Read	Write	0x1A36
0x7A	0xFA	

Table 2-118. Trigger In 2 Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
0	0	0 – Pattern started on rising edge stopped on falling edge 1 – Pattern started on falling edge stopped on rising edge	d0	wr
	7:1	Reserved	d0	r

2.4.4.2 LED Enable Delay Commands

The LED Enable Delay commands set the rising and falling edge offsets of the LED enable signals compared to when the pattern is displayed on the DMD. This command is only for Pattern Display mode. When in a video mode, set these delays to 0x0.

2.4.4.2.1 Red LED Enable Delay

The Red LED Enable Delay command sets the rising and falling edge delay of the Red LED enable signal.

Table 2-119. Red LED Enable Delay Command

I²C		USB
Read	Write	0x1A1F
0x6C	0xEC	

Table 2-120. Red LED Enable Delay Command Definition [prior to FW 6.x]

BYTE	BITS	DESCRIPTION (1)	RESET	TYPE
1:0	15:0	LED Enable Rising Edge delay in micro seconds (int16 number) Valid Range : -20 to 20000	d0	wr

Table 2-120. Red LED Enable Delay Command Definition [prior to FW 6.x] (continued)

BYTE	BITS	DESCRIPTION (1)	RESET	TYPE
3:2	15:0	LED Enable Falling Edge delay in micro seconds (int16 number) Valid Range : -20 to 20000	d0	wr

Table 2-121. New Red LED Enable Delay Command Definition [beginning with FW 6.x]

BYTE	BITS	DESCRIPTION (1)	RESET	TYPE
0	0	Invert the Red LED output	d0	wr
	7:1	Reserved	d0	wr
2:1	15:0	LED Enable Rising Edge delay in micro seconds (int16 number) Valid Range : -20 to 20000	d0	wr
4:3	15:0	LED Enable Falling Edge delay in micro seconds (int16 number) Valid Range : -20 to 20000	d0	wr

(1) The minimum delay is affected when the number of active blocks is reduced. The formula to calculate the minimum delay is: – (min_exposure – 5) µs. See Table 2-102 for the min_exposure for the number of active DMD blocks.

Note

The new Red LED Enable Delay Command Definition is NOT backward compatible with FW prior to 6.x.

2.4.4.2.2 Green LED Enable Delay

The Green LED Enable Delay command sets the rising and falling edge delay of the Green LED enable signal.

Table 2-122. Green LED Enable Delay Command

I²C		USB
Read	Write	0x1A20
0x6D	0xED	

Table 2-123. Green LED Enable Command Definition [prior to FW 6.x]

BYTE	BITS	DESCRIPTION (1)	RESET	TYPE
1:0	15:0	LED Enable Raising Edge delay in micro seconds (int16 number) Valid Range : -20 to 20000	d0	wr
3:2	15:0	LED Enable Falling Edge delay in micro seconds (int16 number) Valid Range : -20 to 20000	d0	wr

Table 2-124. New Green LED Enable Delay Command Definition [beginning with FW 6.x]

BYTE	BITS	DESCRIPTION (1)	RESET	TYPE
0	0	Invert the Green LED output	d0	wr
	7:1	Reserved	d0	wr
2:1	15:0	LED Enable Rising Edge delay in micro seconds (int16 number) Valid Range : -20 to 20000	d0	wr
4:3	15:0	LED Enable Falling Edge delay in micro seconds (int16 number) Valid Range : -20 to 20000	d0	wr

(1) The minimum delay is affected when the number of active blocks is reduced. The formula to calculate the minimum delay is: – (min_exposure – 5) µs. See Table 2-102 for the min_exposure for the number of active DMD blocks.

Note

The new Green LED Enable Command Definition is NOT backward compatible with FW prior to 6.x.

2.4.4.2.3 Blue LED Enable Delay

The Blue LED Enable Delay command sets the rising and falling edge delay of the Blue LED enable signal.

Table 2-125. Blue LED Enable Delay Command

I²C		USB
Read	Write	0x1A21
0x6E	0xEE	

Table 2-126. Blue LED Enable Delay Command Definition [prior to FW 6.x]

BYTE	BITS	DESCRIPTION (1)	RESET	TYPE
1:0	15:0	LED Enable Raising Edge delay in micro seconds (int16 number) Valid Range : -20 to 20000	d0	wr
3:2	15:0	LED Enable Falling Edge delay in micro seconds (int16 number) Valid Range : -20 to 20000	d0	wr

Table 2-127. New Blue LED Enable Delay Command Definition [beginning with FW 6.x]

BYTE	BITS	DESCRIPTION (1)	RESET	TYPE
0	0	Invert the Blue LED output	d0	wr
	7:1	Reserved	d0	wr
2:1	15:0	LED Enable Rising Edge delay in micro seconds (int16 number) Valid Range : -20 to 20000	d0	wr
4:3	15:0	LED Enable Falling Edge delay in micro seconds (int16 number) Valid Range : -20 to 20000	d0	wr

(1) The minimum delay is affected when the number of active blocks is reduced. The formula to calculate the minimum delay is: – (min_exposure – 5) µs. See Table 2-102 for the min_exposure for the number of active DMD blocks.

Note

The new Blue LED Enable Delay Command Definition is NOT backward compatible with FW prior to 6.x.

2.4.4.3 Pattern Display Commands

2.4.4.3.1 Pattern Display Start/Stop

The Pattern Display Start/Stop command starts or stops the programmed pattern sequence. After executing this command, the host may poll the system status using I²C commands: 0x20, 0x21, and 0x22 or the respective USB commands: 0x1A0A, 0x1A0B, and 0x1A0C.

Table 2-128. Pattern Display Start/Stop Command

I²C		USB
Read	Write	0x1A24
N/A	0xE5	

Table 2-129. Pattern Display Start/Stop Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
0	1:0	0 = Stop Pattern Display Sequence. The next *Start* command restarts the pattern sequence from the beginning. 1 = Pause Pattern Display Sequence. The next *Start* command starts the pattern sequence by re-displaying the current pattern in the sequence. 2 = Start Pattern Display Sequence 3 = Reserved	d0	w
	7:2	Reserved	d0	w

2.4.4.3.2 Pattern Display Invert Data

The Pattern Display Invert Data command dictates how the DLPC900 interprets a value of 0 or 1 to control mirror position for displayed patterns.

Note

Before executing this command, stop the current pattern sequence. Once the command has been sent to the DLPC900, the Pattern Display LUT Definition for all the patterns must be re-sent to the DLPC900.

Table 2-130. Pattern Display Invert Data Command

I²C		USB
Read	Write	0x1A30
0x74	0xF4	

Table 2-131. Pattern Display Invert Data Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
0	0	Pattern Display Invert Data 0 = Normal operation. A data value of 1 flips the mirrors to output light, while a data value of 0 flips the mirrors to block light 1 = Inverted operation. A data value of 0 l flips the mirrors to output light, while a data value of 1 flips the mirrors to block light	d0	wr
	7:1	Reserved	d0	r

2.4.4.3.3 Pattern Display LUT Configuration

The Pattern Display LUT Configuration command controls the execution of patterns stored in the lookup table (LUT). Before executing this command, stop the current pattern sequence.

Issue this command after any *Pattern Display LUT Definition* command is issued.

NOTES:

- This command makes all *Pattern Display LUT Definition* data effective and sets the *Pattern Display LUT* default order which displays all patterns in the DLPC900 pattern memory in the order they are defined by the pattern index in *Pattern Display LUT Definition*.
- When this command is executed any pattern that precedes a pattern with an input trigger in the *Pattern Display LUT Definition*, has a black pattern loaded at the end. This causes the system to display a darkness until the trigger is received to start the triggered pattern. If the triggered pattern is the very first pattern in the *Pattern Display LUT Definition* then the very last pattern in the in the *Pattern Display LUT Definition* is treated as the pattern preceding it.

Table 2-132. Pattern Display LUT Configuration Command

I²C		USB
Read	Write	0x1A31
0x75	0xF5	

Table 2-133. Pattern Display LUT Configuration Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
1:0	10:0	Number of LUT entries (range 1 through 400) 1 = One entry 2 = Two entries ... 512 = 512 entries	d0	wr
	15:11	Reserved		
5:2	31:0	Number of times to repeat the pattern sequence	d0	wr

2.4.4.3.4 Pattern Display LUT Reorder Configuration

The Pattern Display LUT Reorder Configuration command reorders the lookup table (Pattern Display LUT) so that the patterns stored in memory are displayed in the order defined by this command. Before executing this command, stop the current pattern sequence. This command is only applicable in Pre-stored Pattern Mode and Pattern On-The-Fly Mode.

Patterns can be referenced in any order and can be repeated in the Pattern Display LUT. Moreover, a subset of patterns stored in the DLPC900 pattern memory can be referenced.

NOTES:

- The default display order must be set by issuing a *Pattern Display LUT Configuration* command before this command can be used.
- The pattern index numbers used must be in the set of patterns defined by the *Pattern Display LUT Configuration* command (i.e. Must less than or equal to the number of entries - 1 defined by the *Pattern Display LUT Configuration* command).

Table 2-134. Pattern Display LUT Reorder Configuration Command

I²C		USB
Read	Write	0x1A32
	0xF6	

Table 2-135. Pattern Display LUT Reorder Configuration Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
1:0	10:0	Number of LUT entries (range 1 through 512) 1 = One entry 2 = Two entries … 512 = 512 entries	d0	wr
5:2	31:0	Number of times to repeat the pattern sequence	d0	wr
7:6	15:0	Pattern index number to be displayed first	d0	wr
9:8	15:0	Pattern index number to be displayed second	d0	wr
…	…	…		
…	15:0	Pattern index number to be displayed in Nth position NOTE: N = number of LUT entries (BYTE 1:0) NOTE: the Pattern index to be displayed must be <= number of entries - 1 defined in 0x1A31	d0	wr

Input Trigger considerations:

Patterns with input triggers, as defined by the *Pattern Display LUT Definition* commands, are attached to the pattern. Therefore the system waits for a trigger whenever the pattern index number is referenced in the re-ordered *Display Pattern LUT*. Moreover, the pattern that originally preceded the triggered pattern in the default *Display Pattern LUT* order displays 105 µs of dark time at the end of it regardless of where it appears in the re-ordered *Display Pattern LUT*. In addition, if a pattern that was not preceding the triggered pattern in the default *Display Pattern LUT* order is placed immediately in front of the triggered pattern, the last bit pattern of that pattern displays until the trigger is received.

There are several methods that can be employed to manage this behavior:

- Consider the triggered pattern and the pattern preceding it in the default *Display Pattern LUT* order as a set that must be kept together.
- Add a 1-bit all-black pattern with the shortest duration allowed (105 µs) preceding the triggered pattern in the default *Display Pattern LUT* order. Now consider these patterns as a set that must be kept together.
- Add a 1-bit all-black pattern with the shortest duration allowed (105 µs) followed by a triggered 1-bit all-black pattern (also 105 µs duration), and remove the trigger from the pattern originally to be triggered in the default *Display Pattern LUT* order. These two patterns become a versatile trigger set that can be used repeatedly wherever a trigger is desired. Since the triggered pattern is black, even if your exposure integration begins with the triggered black pattern there is no additional light contributing to your exposure.

2.4.4.3.5 Pattern Display LUT Definition

The Pattern Display LUT Definition contains the definition of each pattern to be displayed during the pattern sequence. **Display Mode must be set before sending any pattern LUT definition data. If the Pattern Display Data Input Source is set to streaming, the image indexes do not need to be set. After any Pattern Display LUT Definition command is issued a Pattern Display LUT Configuration command must be issued**. Regardless of the input source, the pattern definition must be set.

NOTES:

- Pattern definition data can be changed using this command without reloading pattern data into the DLPC900 pattern memory.
- It is possible to use *Pattern Display LUT Definition* commands to change the pattern definitions for some or all of the patterns in a previously set default *Display Pattern LUT* . Only those pattern indices that are to be changed need to entered. (For the changes to take effect a *Pattern Display LUT Configuration* command must again be issued.)

Table 2-136. Pattern Display LUT Definition Command

I²C		USB
Read	Write	0x1A34
N/A	0xF8	

Table 2-137. Pattern Display LUT Definition Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
1:0	15:0	Pattern Index (range 0 through 511)	d0	w
4:2	23:0	Pattern exposure in micro seconds		
5	0	Clear the pattern after exposure. This is only applicable for 1 bit patterns with an external trigger. For other patterns, the clear is automatically handled.		w
	3:1	Bit Depth: Select desired bit-depth (see byte 9, bit = 1 for bit depths from 9-16.) (1) b000 = 1 bit b001 = 2 bit b010 = 3 bit ... b111 = 8 bit		w
	6:4	b000 = All LEDs disabled b001 = Red b010 = Green b011 = Yellow (Green + Red) b100 = Blue b101 = Magenta (Blue + Red) b110 = Cyan (Blue + Green) b111 = White (Blue + Green + Red)		w
	7	1 = Wait for trigger before displaying the pattern 0 = Continue running after previous pattern		w
8:6	23:0	Dark display time following the exposure (in micro seconds)		w
9	0	1 = Disable trigger 2 output for this pattern 0 = Enable trigger 2 output for this pattern		w
	1	0 = 8-bit depth 1 = 16-bit Extended bit depth Introduced in firmware (FW) 6.x -- Extended Bit Depth bit for bit depths 9 - 16. **Actual Bit Depth** = Bit Depth (see byte 5 bits 3:1) + Extended Bit Depth * 8*(1) **Note** This parameter is ignored in firmware versions prior to 6.0		w
	7:2	Reserved		w
11:10	10:0	Image pattern index (Not applicable in video pattern mode) Valid Range 0-255		w
	15:11	Bit position in the image pattern (Frame in video pattern mode) Valid range 0-23		w

(1) The Extended Bit Depth bit was added to a previously reserved part of the LUT definition for backward compatibility with previous versions.

2.4.4.4 Pattern On-The-Fly Commands

These commands allow the user to dynamically upload the pattern images over the I^2C or USB interface and store them directly into internal memory. The user can preview the pattern sequence to verify that the patterns and the pattern sequence are correct before actually writing the patterns to the flash. Only use commands in Pattern On-The-Fly mode and requires **Display Mode to be set before sending any pattern LUT definition data. After any Pattern Display LUT Definition command is issued a Pattern Display LUT Configuration command must be issued.** Section 5.3 shows a Pattern On-The-Fly example.

2.4.4.4.1 Initialize Pattern BMP Load

When the Initialize Pattern BMP Load command is issued, the patterns in the flash are not used until the pattern mode is disabled by command. Follow this command by the Pattern BMP Load command to load the images. Load the images in the reverse order. Suppose there are 3 images 0,1 and 2 then the order for loading the image is 2, 1 and 0. **When the DLPC900 is combined with a Dual Controller DMD, the user must perform the same operation on both the I²C master and slave controllers by choosing the appropriate command in the command table.**

Table 2-138. Initialize Pattern BMP Load Command

Controller	I²C		USB
	Read	Write	
Master	0x2A	0xAA	0x1A2A
Slave	0x2C	0xAC	0x1A2C

Table 2-139. Initialize Pattern BMP Load Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
1:0	4:0	Image Index (0 – 17). In 24 bit format. This be referred in LUT command. Always load the images in reverse order.	d0	wr
	15:5	Reserved		
5:2	31:0	Number of bytes in the compressed image including the 48 byte header	d0	wr

2.4.4.4.2 Pattern BMP Load

This command is used for updating the pattern images on-the-fly. This loads the full compressed 24 bit BMP images into the internal memory of the DLPC900. This command is issued after the Init pattern BMP command and multiple times until all the bytes are sent. Compress images using Run-Length Encoding (RLE). See Section 2.4.3 for a description of the compression formats.

The first line of this command must contain the 48 bytes of the Image Header and the remaining 456 bytes contain the first 456 bytes of the image data. See Section 2.4.2

When the DLPC900 is combined with a dual controller DMD, the user must load the images to both the I²C master and slave controllers by choosing the appropriate command in the command table. The full image must be divided in half where the I²C master controller gets the left half and the I²C slave controller gets the right half. Include the image header in the first line of both image halves where the image "width" is one half of the full array (native resolution) width.

Note

Re-download the images to the DLPC900 whenever changes are made to the number of entries in the Pattern Display LUT Configuration or changing the images, bit depth, image index, or bit position in the Pattern Display LUT Definition

Table 2-140. Pattern BMP Load Command

Controller	I²C		USB
	Read	Write	
Master	0x2B	0xAB	0x1A2B
Slave	0x2D	0xAD	0x1A2D

Table 2-141. Pattern BMP Load Command

BYTES	BITS	DESCRIPTION	RESET	TYPE
1:0	9:0	Number of bytes in this packet	d0	w
	15:10	Reserved		
n:2	All	Compressed BMP Data	d0	w

2.4.4.5 I²C Pass Through Commands

The I²C Pass Through commands allow the user to use I²C port 1 or port of 2 the controller to control external devices.

2.4.4.5.1 I²C Pass Through Configuration

The I²C Pass Through Configuration command configures the I²C port to be used.

Table 2-142. I²C Pass Through Configuration Command

I²C		USB
Read	Write	0x1A4E
N/A	0xC5	

Table 2-143. I²C Pass Through Configuration Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
0	1:0	I²C Port number 1 or 2 0 = Invalid Port 1 = Port 1 2 = Port 2 3 = Invalid Port	d0	w
	3:2	Reserved		
	4	Device addressing mode 0 – 7 bit addressing 1 – 10 bit addressing	d0	w
	7:5	Reserved		
4:1	31:0	I²C Clock rate 100000 – 400000 Hz (Actual rate may not be exactly as entered due to the dividers used in calculating the rate)	d0	w

2.4.4.5.2 I²C Pass Through Write

The I²C Pass Through Write command allows the user to send data to the specified I²C device on the port that was configured by the Pass Through Configuration command.

Table 2-144. I²C Pass Through Write Command

I²C		USB
Read	Write	0x1A4F
N/A	0xCF	

Table 2-145. I²C Pass Through Write Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
1:0	15:0	Number of bytes to write (1 – 512)	d0	w
2	1:0	I²C Port number 1 or 2. (Port configuration of the port being used must have been done prior to using this command) 0 = Invalid Port 1 = Port 1 2 = Port 2 3 = Invalid Port	d0	w
	7:2	Reserved		
4:3	10:0	Slave Address	d0	w
	15:11	Reserved		
n:5	All	Bytes to be written	d0	w

2.4.4.5.3 I²C Pass Through Read

The I²C Pass Through Read command allows the user to read data from the specified I²C device on the port that was configured by the Pass Through Configuration command.

Table 2-146. I²C Pass Through Read Command

I²C		USB
Read	Write	0x1A4F
0x4F	N/A	

Table 2-147. I²C Pass Through Read Command Definition

BYTE	BITS	DESCRIPTION	RESET	TYPE
1:0	15:0	Number of bytes to write (1-512)	d0	w
3:2	15:0	Number of bytes to read (1-512)	d0	w
4	1:0	I²C Port number 1 or 2 (Port configuration of the port being used must have been done prior to using this command)	d0	w
		0 = Invalid Port		
		1 = Port 1		
		2 = Port 2		
		3 = Invalid Port		
	7:2	Reserved		
6:5	10:0	Slave Address	d0	w
	15:11	Reserved		
n:7	All	Data to be written	d0	w
m:0	All	Data bytes read	d0	r

3 DLPC900 Fault Status

3.1 DLPC900 FAULT_STATUS Location(s)

The DLPC900 produces error codes, or fault statuses, under certain error conditions. The FAULT_STATUS pin on the DLPC900 is AC11 (See DLPC900 datasheet DLPS037 for pin details).

3.2 DLPC900 FAULT_STATUS Interpretation

The format of the DLPC900 FAULT_STATUS signal is shown in Figure 3-1. The signal begins with a pulse, or pulses, indicating the critical error type. The number of critical error pulses indicates the critical error type - LLFAULT_MAIN, LLFAULT_SYS, or LLFAULT_EX. The critical error pulses are then followed by a short pause and one or more module error pulses. A longer pause follows the module error type, indicating repetition of the fault status sequence. In Table 3-1, the fault status can be interpreted by finding the status matching the number of critical error pulses followed by the number of module error pulses.

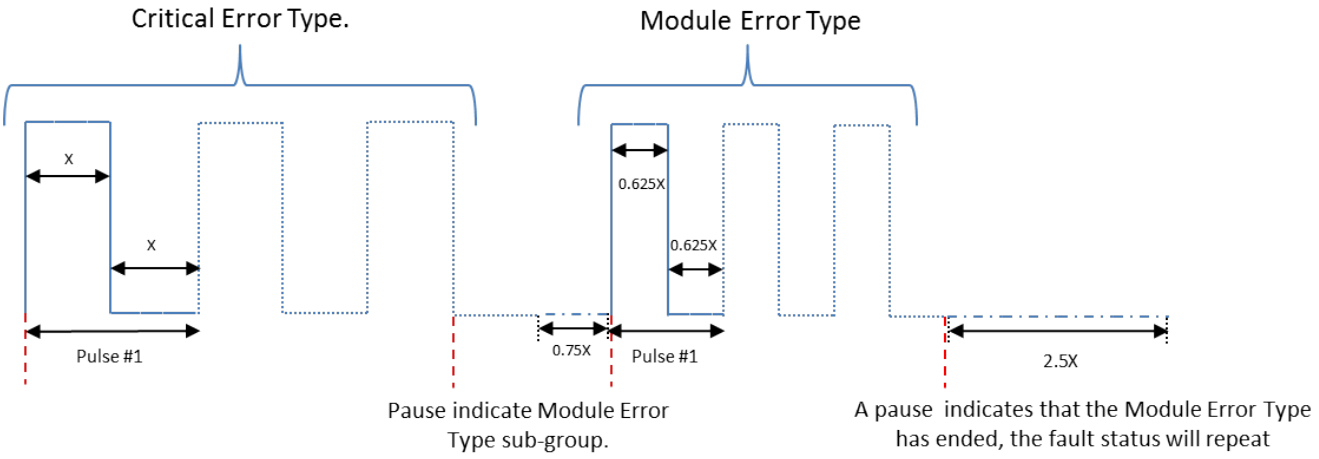

Figure 3-1. DLPC900 FAULT_STATUS Format

Table 3-1. DLPC900 Fault Status Description

Number of Critical Error Pulses	Number of Module Error Pulses	Description
1	1	Missing flash table signature
2	2	Mismatched controller SW configuration versions
3	4	Error in first initialization of I2C
	7	DMD Initialization error
	8	DMD/controller incompatibility fatal error

For any combination of pulses not listed please contact TI for more information.

4 Power-Up and Power-Down and Initialization Considerations

This chapter describes the initial power-up and power-down considerations, as well as other initialization considerations.

4.1 Power Up

The DLPC900 is initialized and ready to process commands sometime after the signal $\overline{\text{RESET}}$ is driven high. Detailed power-up timing is given in the DLPC900 data sheet, DLPS037.

4.2 Power Down

A *Power Standby* command (Section 2.3.1.1)is required at power down of the DLPC900. Please see the Power-Down requirements in the DLPC900 data sheet, DLPS037.

4.3 Power-Up Auto-Initialization

Upon release of system reset, the DLPC900 executes an auto-initialization routine that is automatically uploaded from flash. This initialization process consists of setting specific configurations, uploading specific configuration tables (such as sequence), and displaying a defined splash screen. The goal of the auto-initialization process is to allow the DLPC900 to fully configure itself for default operation with no external I^2C control.

5 Command Examples

5.1 Video Pattern Mode Example

The following table lists the step for a Video Pattern Mode example with two exposures. Start with the system powered on and displaying a video source in Video Mode.

Note

An error occurs if an attempt is made to switch to Video Pattern Mode if there is not a video source with a sync lock.

Note

If the receiver source is switched to another source while in Video Pattern Mode (such as from HDMI to DisplayPort, or RGB input) video sync lock is lost requiring a return to video mode from RGB source to regain sync lock before returning to Video Pattern Mode.

Note

The total exposure time of patterns defined within an input video frame cannot exceed the total frame time or a Sequencer Abort Error occurs. This means that when a sync occurs, if all of the exposure(s) defined within that frame are not finished the sequencer aborts and a sequencer error is thrown. The sequencer attempts to resume on the next frame following the aborted frame.

Table 5-1. Video Pattern Mode Example

STEP	I²C (1)	USB (1)	DATA (1)	DESCRIPTION
1	E9	1A1B	02	Set video pattern mode
2	F8	1A34	00 00 C8 00 00 90 00 00 00 00 00 00	Define pattern 0 (200 µs red 1 bit) and wait for trigger
3	F8	1A34	01 00 90 01 00 21 00 00 00 00 00 08	Define pattern 1 (400 µs green 2 bit)
4	F5	1A31	02 00 00 00 00 00	Number of patterns 2 with indefinite repeat
5	E5	1A24	02	Start running the pattern (2)

(1) All bytes are in HEX notation.
(2) Connect a video source before performing this step.

5.2 Pre-Stored Pattern Mode Example

The following table lists the steps for a Pre-Stored Pattern Mode example with two exposures. Start with the system powered on.

Table 5-2. Pre-Stored Pattern Mode Example

STEP	I²C (1)	USB (1)	DATA (1)	DESCRIPTION
1	E9	1A1B	01	Set pre-stored pattern mode
2	F8	1A34	00 00 C8 00 00 10 00 00 00 00 00 00	Define pattern 0 (200 µs red 1 bit)
3	F8	1A34	01 00 90 01 00 21 00 00 00 00 00 08	Define pattern 1 (400 µs green 2 bit)
4	F5	1A31	02 00 00 00 00 00	Number of patterns 2 and indefinite repeat
5	E5	1A24	02	Start running the pattern (2)

(1) All bytes are in HEX notation.
(2) There must be at least two pattern images in flash memory.

5.3 Pattern On-The-Fly Example

The following table lists the steps for Image On-The-Fly Pattern Mode example with two images. Start with the system powered on. Compress images using Run-Length Encoding (RLE).

Table 5-3. Pattern On-The-Fly Example

STEP	I²C (1)	USB (1)	DATA (1)	DESCRIPTION
1	E9	1A1B	03	Set on-the-fly pattern mode
2	F8	1A34	00 00 C8 00 00 10 00 00 00 00 00 00	Define pattern 0 from image 0 (200 µs red 1 bit)
3	F8	1A34	01 00 90 01 00 21 00 00 00 00 01 08	Define pattern 1 from image 1 (400 µs green 2 bit)
4	AA	1A2A	01 00 E8 03 00 00	Set BMP 1 Size to 1000 (0x03E8)
5	AB	1A2B	F8 01 XX XX …… XX	Load 504 bytes of compressed BMP 1 Data
6	AB	1A2B	E0 01 XX XX ….. XX	Load 496 bytes of compressed BMP 1 Data
7	AA	1A2A	00 00 D0 07 00 00	Set BMP 0 Size to 2000 (0x07D0)
8	AB	1A2B	F8 01 XX XX …… XX	Load 504 bytes of compressed BMP 0 Data
9	AB	1A2B	F8 01 XX XX …… XX	Load 504 bytes of compressed BMP 0 Data
10	AB	1A2B	F8 01 XX XX …… XX	Load 504 byte of compressed BMP 0 Data
11	AB	1A2B	E8 01 XX XX …… XX	Load 488 bytes of compressed BMP 0 Data
12	F5	1A31	02 00 00 00 00 00	Number of patterns 2 and indefinite repeat
13	E5	1A24	02	Start running the pattern

(1) All bytes are in HEX notation.

5.4 I²C Pass Through Write Example

The following table lists the steps to communicate with an external device using one of the DLPC900 I²C ports. The example shows how to write 16 bytes to an EEPROM starting at address location 16.

Table 5-4. I²C Pass Through Write Example

STEP	I²C (1)	USB (1)	DATA (1)	DESCRIPTION
1	C5	1A4E	01 A0 86 01 00	Address mode = 7-bits, port = 1, and clock = 100 kHz
2	CF	1A4F	11 00 01 A0 00 00 10 01 18 01 03 A5 00 00 00 DA 04 85 A0 57 4A 9B 26	Number of bytes = 17, port = 1, device address = A0, EEPROM address location = 16, and 16 bytes of data.

(1) All bytes are in HEX notation.

5.5 I²C Pass Through Read Example

The following table lists the steps to communicate with an external device using one of the DLPC900 I²C ports. The example shows how to read 16 bytes from an EEPROM starting at address location 16.

Table 5-5. I²C Pass Through Read Example

STEP	I²C (1)	USB (1)	DATA (1)	DESCRIPTION
1	C5	1A4E	01 A0 86 01 00	Address mode = 7-bits, port = 1, and clock = 100 kHz
2	4F	1A4F	01 00 10 00 01 A0 00 10	Number of bytes to write = 1, number of bytes to read = 16, port = 1, device address = A0, EEPROM address location = 16
3			01 18 01 03 A5 00 00 00 DA 04 85 A0 57 4A 9B 26	The host performs an I²C read operation to retrieve the data.

(1) All bytes are in HEX notation.

A Register Quick Reference

This appendix provides a quick reference summary of all available sub-address commands.

A.1 I²C Register Quick Reference

Table A-1. Register Quick Reference

I²C SUB-ADDRESS		USB	DESCRIPTION	TYPE	RESET VALUE	DEFAULT ACTION
Read	Write					
0x00	0x80	0x1A00	Input Source Select	WR	0x8	24-bit parallel interface
0x02	0x82	0X1A02	Pixel Format	WR	0x0	RGB
0x03	0x83	0x1A03	Port and Clock Configuration	WR	0x0	Single Pixel, Pixel Clock 1, Data enable 1
0x04	0x84	0x1A37	Channel Swap	WR	0x8	ABC = BAC
0x06	0x86	0x1100	Curtain Color	WR	0x0 0x0 0x0 0x0 0x0 0x0	Curtain is black
0x07	0x87	0x0200	Power Mode	WR	0x0	Normal operation
0x08	0x88	0x1008	Long Axis Flip	WR	0x0	Flip disabled
0x09	0x89	0x1009	Short Axis Flip	WR	0x0	Flip disabled
0x0A	0x8A	0x1203	Test Pattern Select	WR	0x0	Solid Field
0x0B	0x8B	0x1A05	LED PWM Polarity	WR	0x0	Normal polarity
0x0C	0x8C	0x1A01	IT6535 Power Mode	WR	0x0	Power down
0x0D	0x8D	0x0201	DMD Idle Mode	WR	0x0	Disabled
0x10	0x90	0x1A07	LED Enable	WR	0x8	LEDs controlled by Sequencer
0x11	-	0x0205	Get Version	R	Matches firmware version stored in Flash	Matches firmware version
0x12	-	0x0206	Get firmware type	R	Matches firmware type stored in Flash	Matches firmware type of attached DMD
0x14	0x94	0x0609	DMD Park / Unpark	WR	0x0	Unpark DMD
0x1A	0x9A	0x1204	Test Pattern Color	WR	0x3FF 0x3FF 0x3FF 0x0 0x0 0x0	White foreground, black background
0x20	-	0x1A0A	Hardware Status	R	0x1	No errors
0x21	-	0x1A0B	System Status	R	0x1	No errors
0x22	-	0x1A0C	Main Status	R	0x0	No errors
0x32	-	0x0100	Read Error Code	R	0x0	No errors
0x33	-	0x0101	Read Error Code Description	R	0x0	No description
-	0xAA	0x1A2A	Initialize Pattern BMP Load	W	0x0	See Command Description
-	0xAB	0x1A2B	Pattern BMP Load	W	0x0	See Command Description
-	0xAC	0x1A2C	Initialize Pattern BMP Load	W	0x0	See Command Description
-	0xAD	0x1A2D	Pattern BMP Load	W	0x0	See Command Description
0x3A	-	0x1A14	Batch File Name	WR	0x0	Index
-	0xBB	0x1A15	Batch File Execute	W	0x0	Index
-	0xBC	0x1A16	Batch File Delay	W	0x0	Delay
0x40	0xC0	0x1A10	PWM Enable	WR	Channel dependent	Channel dependent
0x41	0xC1	0x1A11	PWM Setup	WR	Channel dependent	Channel dependent
0x44	0xC4	0x1A38	GPIO Configuration	WR	Channel dependent	Channel dependent
0x5E	-	0x1A5E	GPIO Busy	R	0x0	0x0
-	0xC5	0x1A4E	I²C Pass Through Configuration	W	0x0 0x0 0x0 0x0 0x0	See Command Description
0x48	0xC8	0x0807	Clock Configuration	WR	Channel dependent	Channel dependent

Table A-1. Register Quick Reference (continued)

I²C SUB-ADDRESS		USB	DESCRIPTION	TYPE	RESET VALUE	DEFAULT ACTION
0x4B	0xCB	0x0B01	LED Current	WR	0x97 0x78 0x7D	LED PWMs
0x4F	0xCF	0x1A4F	I²C Pass Through Read or Write	WR	See Command Description	See Command Description
0x60	0xE0	0x1A40	DMD Block Load	WR	0x0 0xF or 0x0 0x10	All blocks active
	0x61	0x1A3B	Gamma Configuration and Enable	W	0x0	Gamma disabled
-	0x62	0x1A41	Set Minimum LED Pulse Width in µs	W	0x0	0x0
0x63	-	0x1A42	Get Minimum Pattern Exposure	R	0x0	0x0
-	0x64	0x1A3C	Input Source Configuration Command	W	0x0	0x0
-	0xE5	0x1A24	Pattern Start/Stop	W	0x0	Pattern stopped
-	0x67	0x1A43	Set Minimum LED Pulse Width in ns	W	0x0	0x0
0x69	0xE9	0x1A1B	Display Mode	WR	0x0	Video Mode
0x6A	0xEA	0x1A1D	Trigger Out 1	WR	0x0 0x0 0x0 0x0 0x0	Normal Polarity with no rising or falling delay
0x6B	0xEB	0x1A1E	Trigger Out 2	WR	0x0 0x0 0x0 0x0 0x0	Normal Polarity with no rising delay
0x6C	0xEC	0x1A1F	Red Enable Delay	WR	0x0 0x0 0x0 0x0	No rising or falling delay
0x6D	0xED	0x1A20	Green Enable Delay	WR	0x0 0x0 0x0 0x0	No rising or falling delay
0x6E	0xEE	0x1A21	Blue Enable Delay	WR	0x0 0x0 0x0 0x0	No rising or falling delay
0x74	0xF4	0x1A30	Invert Data	WR	0x0	Normal operation
0x75	0xF5	0x1A31	Pattern LUT Configuration	WR	See Command Description	See Command Description
-	0xF6	0x1A32	Pattern LUT Reorder Configuration	W	See Command Description	See Command Description
-	0xF8	0x1A34	Pattern LUT Definition	W	See Command Description	See Command Description
0x79	0xF9	0x1A35	Trigger In 1	WR	0x69	No delay
0x7A	0xFA	0x1A36	Trigger In 2	WR	0x0	Advance Pattern Pair on Rising Edge (for Trigger Mode 2)
0x7E	0xFE	0x1000	Manual Input Display Resolution	WR	0x0	Output Display Resolution is DMD Dependent
0x7F	0xFF	0x1A39	Image Load	WR	0x0	Image Index

A.2 Command Guide

This section shows which commands can be used in which modes. I²C control and USB commands are accepted in any order, except when special sequencing is required (for example, setting up the flash).

Table A-2. Command Matrix

COMMAND NAME	I²C		USB	NORMAL POWER MODE	STANDBY POWER MODE	VIDEO MODE	VIDEO PATTERN MODE	PRE-STORED PATTERN MODE	PATTERN ON-THE-FLY MODE
	Read	Write							
Input Source Select	0x00	0x80	0x1A00	x		x	x		
Pixel Format	0x02	0x82	0x1A02	x		x	x		
Port and Clock Configuration	0x03	0x83	0x1A03	x		x	x		
Channel Swap	0x04	0x84	0x1A37	x		x	x		
Curtain Color	0x06	0x86	0x1100	x		x			
Power Mode	0x07	0x87	0x0200	x	x (normal only)	x	x	x	x
IT6535 Power Mode	0x0C	0x8C	0x1A01	x		x			
Long Axis Flip	0x08	0x88	0x1008	x		x	x	x	x
Short Axis Flip	0x09	0x89	0x1009	x		x	x	x	x
Test Pattern Select	0x0A	0x8A	0x1203	x					
LED PWM Polarity	0x0B	0x8B	0x1A05	x		x	x	x	x
DMD Idle Mode	0x0D	0x8D	0x0201	x			x	x	x
LED Enable	0x10	0x90	0x1A07	x		x	x	x	x
Get Version	0x11		0x0205	x		x	x	x	x
Test Pattern Color	0x1A	0x9A	0x1204	x					
DMD Park / Unpark	0x14	0x94	0x0609	x		x	x(1)	x(1)	x(1)
Hardware Status	0x20		0x1A0A	x		x	x	x	x
System Status	0x21		0x1A0B	x		x	x	x	x
Main Status	0x22		0x1A0C	x		x	x	x	x
Read Error Code	0x32		0x0100	x		x	x	x	x
Read Error Code Description	0x33		0x0101	x		x	x	x	x
Initialize Pattern BMP Load		0xAA	0x1A2A	x					x
Pattern BMP Load		0xAB	0x1A2B	x					x
Initialize Pattern BMP Load		0xAC	0x1A2C	x					x
Pattern BMP Load		0xAD	0x1A2D	x					x
Batch File Name	0x3A		0x1A14	x		x	x	x	x
Batch File Execute		0xBB	0x1A15	x		x	x	x	x
Batch File Delay		0xBC	0x1A16	x		x	x	x	x
PWM Enable	0x40	0xC0	0x1A10	x		x	x	x	x
PWM Setup	0x41	0xC1	0x1A11	x		x	x	x	x
GPIO Configuration	0x44	0xC4	0x1A38	x		x	x	x	x
GPIO Busy	0x5E		0x1A5E	x		x	x	x	x
I²C Pass Through Configuration		0xC5	0x1A4E	x		x	x	x	x
Clock Configuration	0x48	0xC8	0x0807	x		x	x	x	x
LED Current	0x4B	0xCB	0x0B01	x		x	x	x	x

Table A-2. Command Matrix (continued)

COMMAND NAME	I²C		USB	NORMAL POWER MODE	STANDBY POWER MODE	VIDEO MODE	VIDEO PATTERN MODE	PRE-STORED PATTERN MODE	PATTERN ON-THE-FLY MODE
I²C Pass Through Read or Write	0x4F	0xCF	0x1A4F	x		x	x	x	x
DMD Block Load	0x60	0xE0	0x1A40	x		x	x	x	x
Pattern Start/Stop	0x65	0xE5	0x1A24	x			x	x	x
Display Mode	0x69	0xE9	0x1A1B	x		x	x	x	x
Trigger Out 1	0x6A	0xEA	0x1A1D	x			x	x	x
Trigger Out 2	0x6B	0xEB	0x1A1E	x			x	x	x
Red Enable Delay	0x6C	0xEC	0x1A1F	x			x	x	x
Green Enable Delay	0x6D	0xED	0x1A20	x			x	x	x
Blue Enable Delay	0x6E	0xEE	0x1A21	x			x	x	x
Invert Data	0x74	0xF4	0x1A30	x			x	x	x
Pattern LUT Configuration	0x75	0xF5	0x1A31	x			x	x	x
Pattern LUT Reorder Configuration	-	0xF6	0x1A34	x				x	x
Pattern LUT Definition	-	0xF8	0x1A34	x			x	x	x
Trigger In 1	0x79	0xF9	0x1A35	x			x	x	x
Trigger In 2	0x7A	0xFA	0x1A36	x			x	x	x
Gamma Configuration and Enable	-	0x61	0x1A3B	x		x	x	x	x
Manual Input Display Resolution	0x7E	0xFE	0x1000	x		x			
Image Load	0x7F	0xFF	0x1A39	x		x			

(1) This command can only be used in this mode when the pattern display has been stopped or has not yet been started.

B Batch File Command Descriptors

This appendix provides a quick reference to all supported batch file command descriptors.

B.1 Command Descriptors

Command descriptors are followed by a colon. Each line in the batch file is space delaminated and saved as a text file.

Table B-1. Command Descriptors

COMMAND DESCRIPTOR	DESCRIPTION
SOURCE_SEL	Input Source Select Section 2.3.3.2
PIXEL_FORMAT	Pixel Format Section 2.3.3.3
CLK_SEL	Port and Clock Configuration Section 2.3.3.1
CHANNEL_SWAP	Channel Swap Section 2.3.2.2
POWER_CONTROL	Power Mode Section 2.3.1.1
FLIP_LONG	Long Axis Flip Section 2.3.4.1
FLIP_SHORT	Short Axis Flip Section 2.3.4.2
TPG_SEL	Test Pattern Select Section 2.3.3.4
PWM_INVERT	LED PWM Invert Section 2.3.7.1.1
LED_ENABLE	LED Enable Section 2.3.7.1
PWM_ENABLE	PWM Enable Section 2.3.9.2
PWM_SETUP	PWM Setup Section 2.3.9.1
GPIO_CONFIG	GPIO Configuration Section 2.3.8.1
DE_GAMMA_SET	Gamma Configuration and Enable Command Gamma Configuration
LED_CURRENT	LED Current Section 2.3.7.2
DISP_CONFIG	Display Configuration Section 2.4.1.2
DISP_MODE	Display Mode Section 2.4.1
TRIG_OUT1_CTL	Trigger 1 Output Control Section 2.4.4.1.1
TRIG_OUT2_CTL	Trigger 2 Output Control Section 2.4.4.1.2
RED_LED_ENABLE_DLY	Red LED Enable Delay Section 2.4.4.2.1
GREEN_LED_ENABLE_DLY	Green LED Enable Delay Section 2.4.4.2.2
BLUE_LED_ENABLE_DLY	Blue LED Enable Delay Section 2.4.4.2.3
PAT_START_STOP	Pattern Start, Pause, and Stop Section 2.4.4.3.1
TRIG_IN1_CTL	Trigger Input 1 Control Section 2.4.4.1.3
TRIG_IN2_CTL	Trigger Input 2 Control Section 2.4.4.1.4
INVERT_DATA	Invert Data Section 2.4.4.3.2
PAT_CONFIG	Pattern LUT Configuration Section 2.4.4.3.3
MBOX_ADDRESS	Pattern Display LUT Reorder Configuration Section 2.4.4.3.4
MBOX_DATA	Pattern LUT Definition Section 2.4.4.3.5
SPLASH_LOAD	Image Load Section 2.3.3.6
GPCLK_CONFIG	Clock Output Configuration Section 2.3.8.2
TPG_COLOR	Test Pattern Color Table 2-52
I2C_PASSTHRU	I^2C Pass Through Section 2.4.4.5.2
VIDEO_CONT_SEL	IT6535 Power Mode Section 2.3.5
PATMEM_LOAD_INIT_MASTER	Initialize BMP Pattern On-The-Fly Master Section 2.4.4.4.1
PATMEM_LOAD_DATA_MASTER (1)	Load BMP Pattern On-The-Fly Master Section 2.4.4.4.2
PATMEM_LOAD_INIT_SLAVE	Initialize BMP Pattern On-The-Fly Slave Section 2.4.4.4.1
PATMEM_LOAD_DATA_SLAVE (1)	Load BMP Pattern On-The-Fly Slave Section 2.4.4.4.2
DELAY	Batch File Delay Section 2.3.10.3
I2C_CONFIG	I^2C Pass Through Configuration Section 2.4.4.5.1

Table B-1. Command Descriptors (continued)

COMMAND DESCRIPTOR	DESCRIPTION
CURTAIN_COLOR	Curtain Color Section 2.3.1.3
BATCHFILE_EXECUTE	Batch File Execute Section 2.3.10.2
DMD_BLOCKS	DMD Block Load Section 2.4.1.3
DMD_IDLE	DMD Idle Mode (50/50 Duty Cycle) Section 2.4.1.5

(1) These commands are not allowed to be included in a batch file that is added to the firmware.

Revision History

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

Changes from Revision E (April 2019) to Revision F (June 2020) **Page**

- Added DLP500YX Data Sheet to Related Documents....4
- Added DLP670S and DLP500YX response values to Reading Hardware Configuration and Firmware Tag Command Response....14
- Added table to show new 128 megabyte flash memory address arrangement, more notes on flash memory, and removed image,....16
- Corrected boot flash size to 128 MB in Section 2.2.5 19
- Added two minute wait time for 50/50 display time....22
- Added Requirements for Park command and to resume operation in any Pattern mode 23
- Added Gamma Configuration and Enable Command....31
- Added Minimum LED Pulse Width command in ns to support 16-bit patterns Table 2-72 36
- Added Get Minimum LED Pattern Exposure in nanoseconds (ns)....36
- Added GPIO Busy command to poll if the GPIO subsystem is busy....38
- Changed Single Controller Block Diagram and added Dual Controller Block Diagram....41
- Removed Allowed Pattern Display Combinations Table....41
- Removed mirror array flipped approximately every 105 µs....48
- Added three second wait before disabling Idle Mode....48
- Added consider using Standby Mode for prolonged periods of inactivity....48
- Changed Red LED Enable Delay Command Definition to add one byte with one bit to toggle Red LED output 54
- Changed Green LED Enable Delay Command Definition to add one byte with one bit to toggle Green LED output....55
- Changed Blue LED Enable Delay Command Definition to add one byte with one bit to toggle Blue LED output 56
- Value of I^2C for Pattern Display LUT Reorder Configuration Command corrected from 0x76 to 0xF6....58
- Deleted Reserved bits 24:31 for bytes 4:3 in Pattern Display LUT Definition Command Definition since the field is only 3 bytes (bits 23:0)....59
- Added reference to Extended Bit Depth bit (byte 9, bit 1) for 16-bit pattern support....59
- Deleted Reserved bits 24:31 for bytes 8:6 in Pattern Display LUT Definition Command Definition since the field is only 3 bytes (bits 23:0)....59
- Added Extended Bit Depth bit to support 16-bit patterns....59
- Added DLP670S and DLP500YX to default action of Get firmware type of I^2C Register Quick Reference table 68
- Added Gamma Configuration command to I^2C Register Quick Reference....68
- Added Set Minimum LED Pulse Width in ns command to I^2C Register Quick Reference table. in order to support 16-bit patterns. Original command in µs left for backward compatibility....68
- Added GPIO Busy to Command Matrix....70
- Removed all commands except Normal Power Mode from Standby Power Mode....70
- Added DeGamma Configuration to Batch File Command Descriptors table....72

Changes from Revision D (July 2017) to Revision E (April 2019) **Page**

- Updated E2E link to current forum....4
- Changed Reading "Firmware Type" to "Hardware Configuration" in Section 2.1.5 14
- Added recommendation to put DMD in Idle Mode if it is going to be idle but not powered down....23
- Deleted 4:4:4 from RGB in pixel format since RGB does not support chroma sub-sampling....27
- Added Get Minimum LED Pattern Exposure command in nanoseconds(ns)Table 2-76 36
- Changed note regarding maximum number of patterns from "uncompressed" to "compressed" 41

- Added note about using a user defined solid state illumination sequence with eight 1-bit patterns to achieve maximum 8-bit pattern rates shown in the DMD data sheets. 41
- Added DLP670S and DLP500YX to Minimum Exposure Table..... 47
- Image Header moved from Appendix C to its own section..... 49
- Added notes to the *Pattern Display LUT Configuration* command..... 57
- Added *Pattern Display LUT Reorder Configuration* command..... 58
- Added **Input Trigger Considerations** to the *Pattern Display LUT Reorder Configuration* command..... 58
- Changed From requiring a *Pattern Display LUT Configuration* command before to after using *Pattern Display LUT Definition* commands 59
- Added notes to the *Pattern Display LUT Definition* command..... 59
- Changed requirements for Pattern On-The-Fly Mode to issue a Pattern LUT Configuration after any Patter LUT Definition..... 60
- Removed "Internal error, contact TI for more information" entries from the Table 3-1 table and added a note to contact TI regarding any combinations not listed..... 64
- Changed Power down instruction to include *Power Standby* command..... 65
- Added note required system to have a sync lock by displaying video in Video Mode before switching to Video Pattern mode..... 66
- Corrected hex value in Command Examples to 0x00C8 to match 200 µs pattern display time..... 66
- Deleted "displaying a video source" from Pre-Stored Pattern Mode example..... 66
- Changed pattern data load sizes in Pattern On-The-Fly Example..... 67
- Deleted 4:4:4 from RGB in pixel format since RGB does not support chroma sub-sampling..... 68
- Added Pattern LUT Reorder Configuration to reference tables..... 68
- Added cross references to sections in Batch File Command Descriptors table..... 72

Changes from Revision C (March 2017) to Revision D (July 2017) **Page**

- Moved "Enter Standby mode prior to any planned system power shutdowns" from Table 2-35 22
- Updated Power Mode Command Definition..... 22
- Changed Power Mode 0, 1 descriptive text..... 22
- Added Section DMD Park / Unpark..... 23
- Updated DMD Idle Mode..... 48
- Changed Idle mode recommendation..... 48
- Added 0x14 DMD Park / Unpark..... 68
- Added DMD Park / Unpark..... 70

Changes from Revision B (July 2015) to Revision C (March 2017) **Page**

- Added Section 2.1.5 14
- Added Table 2-9 to Section 2.1.5 14
- Changed "Enter Program Mode" value to "1" in Table 2-19 18
- Changed "Exit Program Mode" value to "2" in Table 2-21 18
- Corrected 128 kilobit to 128 kilobyte to describe the size of the boot flash sector in Section 2.2.5 19
- Changed IT6535 (0x0C) bits to 1:0 and 7:2 in Table 2-61 30
- Added Minimum LED Pulse Width command in µs Table 2-70 36
- Added Get Minimum LED Pattern Exposure command in microseconds (µs) Table 2-74 36
- Changed WQXGA refresh rate from 60 Hz to 120 Hz in Section 2.4 41
- Corrected supported display modes from "0, 2, and 3" to "1, 2, and 3" in Section 2.4.4 51
- Moved Section 3.1 DLPC900 FAULT_STATUS Location(s) and Section 3.2 DLPC900 FAULT_STATUS Interpretation..... 64

Changes from Revision A (November 2014) to Revision B (July 2015) **Page**

- Added EVM User's Guide to related documents..... 4
- General changes to layout..... 4

- Clarified I^2C transaction structure....5
- Corrected *bits* to *byte* in table header....13
- Added bit to check if external video source is locked....13
- Added bit to check if port syncs are valid....13
- Corrected *bits* to *value* in table header....15
- Added bit to check if pattern image memory address is out of range....15
- Added information, image, and notes on flash memory....16
- Updated Read Status Command Definition Table to reflect firmware 2.0....16
- Added bit to request number of flash devices present....19
- Changed location size of the boot loader to match firmware 2.0 and later....19
- Corrected blue curtain color intensity reset value....23
- Removed Allowed Pattern Display Combinations table....41
- Added clarification on how and when block load is used....46
- Corrected minimum exposure values for number of active DMD blocks....46
- Added table with minimum exposure based on bit depth....47
- Added DMD Idle Mode command....48
- Corrected video pattern mode trigger description and timing diagram....51
- Corrected pre-stored pattern mode trigger description and timing diagram....51
- Corrected trigger output1 rising and falling edge minimum delay to -20 µs....53
- Corrected trigger output2 rising and falling edge minimum delay to -20 µs....53
- Corrected Red LED enable rising and falling edge minimum delay to -20 µs....54
- Corrected Red LED enable rising and falling edge minimum delay to -20 µs....54
- Corrected green LED enable rising and falling edge minimum delay to -20 µs....55
- Corrected Green LED enable rising and falling edge minimum delay to -20 µs....55
- Corrected blue LED enable rising and falling edge minimum delay to -20 µs....56
- Corrected Green LED enable rising and falling edge minimum delay to -20 µs....56
- Added that Initialize Pattern BMP Load Command size must include header....61
- Added DLPC900 fault status table....64
- Corrected the step order on the Pattern on the Fly example....67
- PWM Capture register is unavailable and was removed from table....68

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