DLPC230-Q1 and DLPC231-Q1 for Light Control Applications

Programmer's Guide

Literature Number: DLPU048E
AUGUST 2016 – REVISED JULY 2024

Table of Contents

List of Figures

List of Tables

This page intentionally left blank.

Trademarks

All trademarks are the property of their respective owners.

This page intentionally left blank.

Chapter 1

Introduction

1.1 Document Overview

This guide is intended for software programmers developing host software to control DMDs used for light control applications such as the DLP5531-Q1 or DLP4621-Q1 chipset through the DLPC230-Q1 or DLPC231-Q1 controllers. It includes communication protocols, command descriptions, and hardware pins that relate to software functionality. Additional use-case information is provided to explain general command usage scenarios and how software settings can be modified to meet necessary product goals.

1.2 Software Overview

DLPC230-Q1 software is comprised of two components: a boot application and a main application. The BOOTHOLD mode of the controller places the controller into its boot application so that it never reaches the main application. In this mode, the user may program an empty or corrupted flash device. The boot application also allows programming of an empty or corrupted flash, during which the main application would not be available. The majority of system functionality resides in the main application. Both applications share a common hardware-supported SPI and I^2C communication protocol to allow control from a host processor, but each application includes a different set of command definitions to support their functionality.

External SPI flash memory is used to configure all functions of the DLPC230-Q1 software. The flash data contains information including default settings, DMD sequencer instructions, splash image data, and batch command sets.

1.3 Headlight Overview

The following section describes the software-related components and interfaces of the headlight application of the DLP5531-Q1 chipset. More detail on the hardware interfaces can be found in the DLPC230-Q1 Data Sheet and TPS99000-Q1 Data Sheet.

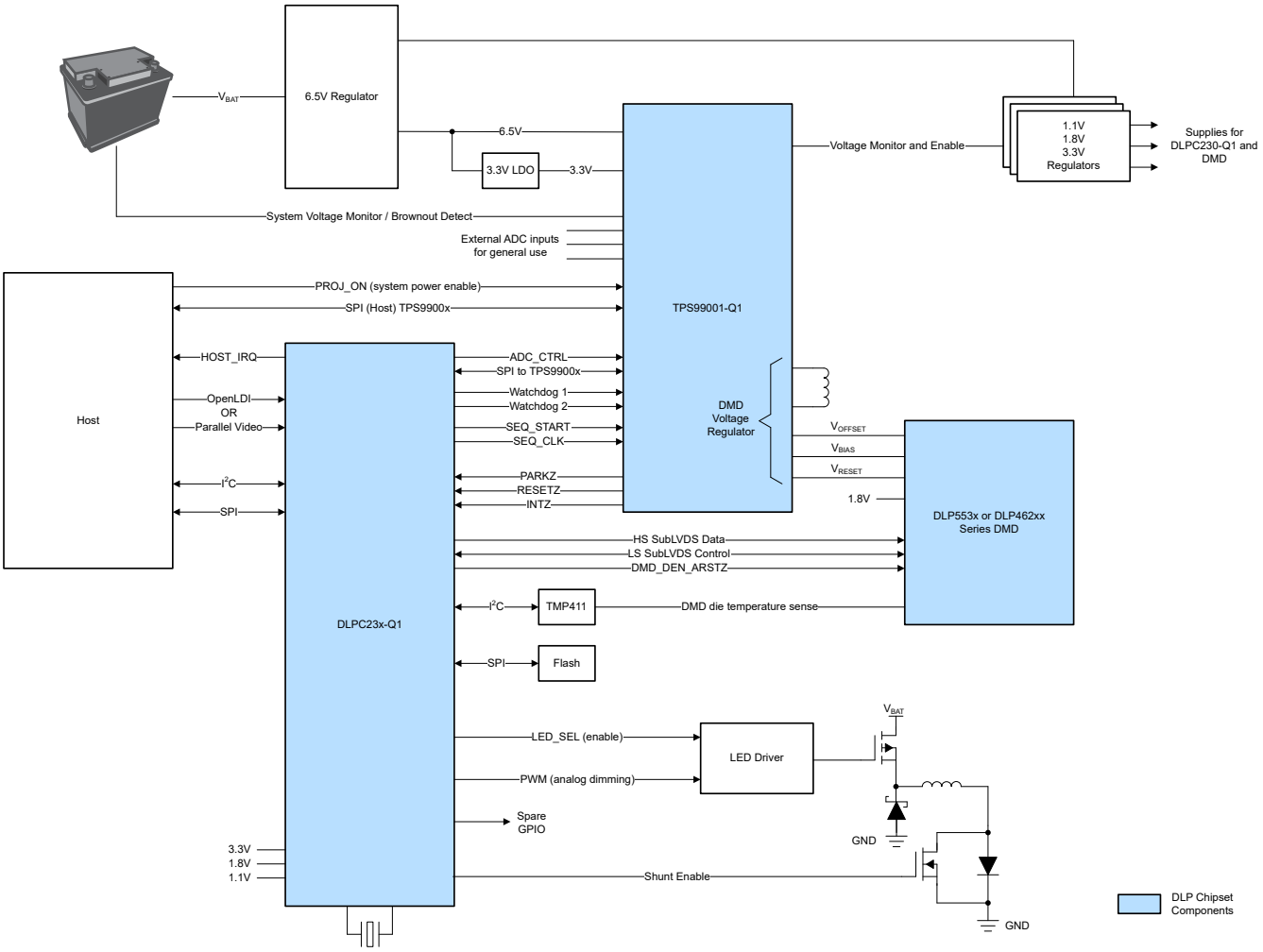


Figure 1-1. Headlight System Block Diagram

SPI DLPC230-Q1 Host Interface - An SPI communication interface that allows the host primary to communicate with the DLPC230-Q1 using the commands described in this guide. If the I^2C interface is used as the host interface, this interface will be configured as a read-only interface to a DLPC230-Q1 diagnostic memory for third-party verification of system data.

I^2C DLPC230-Q1 Host Interface - An I^2C communication interface that allows the host to communicate with the DLPC230-Q1 using the commands described in this guide. If the SPI interface is used as the host interface, this interface will be configured as a read-only interface to a DLPC230-Q1 diagnostic memory for third-party verification of system data.

SPI DLPC230-Q1 to TPS99000-Q1 Interface - An SPI communication interface between the DLPC230-Q1 and TPS99000-Q1. Note that the TPS99000-Q1 is not directly controlled by the host processor during operation.

SPI TPS99000-Q1 Host Interface - An SPI communication interface between the host and TPS99000-Q1. This is a read-only interface to allow a host to confirm TPS99000-Q1 status during operation.

OpenLDI and Parallel Video Interfaces - Either OpenLDI or parallel video may be provided for video input. The DLPC230-Q1 Data Sheet defines supported resolutions and frame rates on these interfaces.

SPI Flash - Contains the main application software as well as all necessary data for the main application to operate. Programming the SPI flash is performed through DLPC230-Q1 commands sent to the DLPC230-Q1 boot application or main application.

TMP411 - Reads the DMD temperature from a temperature sensing diode built into the DMD. The DLPC230-Q1 continuously polls the TMP411 remote channel temperature during operation to take action to park the DMD at extreme temperatures. The DMD temperature may also be read from the DLPC230-Q1.

HOST_IRQ - An interrupt signal that alerts the host that an error caused the DLPC230-Q1 software to switch to Standby Mode and park the DMD. More information on this signal can be found in Section 2.2.1.

PROJ_ON - Host-controlled reset signal for the entire chipset. More information on this signal can be found in the DLPC230-Q1 Data Sheet.

GPIO - The DLPC230-Q1 includes several spare GPIO that may be controlled by the host processor using SPI or I^2C commands.

PWM - The DLPC230-Q1 includes three PWM channels that may be controlled by the host processor using SPI or I^2C commands for various purposes such as illumination control.

Chapter 2

Electrical Interface

2.1 Power-Up Signals

The DLPC230-Q1 includes hardware pins to modify software and hardware functionality. DLPC230-Q1 software reads the value of these pins during its startup procedure to configure communication interfaces and software startup behavior.

2.1.1 STAY-IN-BOOT (TSTPT_0)

The STAY-IN-BOOT signal is used by the boot application to determine whether the system should remain in boot execution or attempt to load and transition to the main application during power-up.

Remaining in the boot application may be desirable for re-programming the SPI flash, including the main application, if any system-level issue prevents re-programming during the execution of the main application.

Table 2-1. STAY-IN-BOOT Configuration

VALUE AT SYSTEM POWER-UP	SYSTEM ACTION
0	Load main application from flash. Normal operation.
1	Stay in boot application and wait for command instructions.

2.1.2 HOST_IF_SEL

The HOST_IF_SEL signal is used by the DLPC230-Q1 hardware during system power-up to determine which interface (SPI or I^2C) is to be used for host control of the system. The interface that is not selected for host control will be configured as a diagnostic interface which can be used to read diagnostic memory information.

Table 2-2. HOST_IF_SEL Configuration

VALUE AT SYSTEM POWER-UP	SYSTEM ACTION
0	SPI = Host Control Interface, I2C = Diagnostic Interface
1	SPI = Diagnostic Interface, I2C = Host Control Interface

2.1.3 CRCZ_CHKSUM_SEL

The CRCZ_CHKSUM_SEL signal is used by the boot application during system power-up to determine whether the host will use an 8-bit CRC or 8-bit checksum to verify command transactions. The main application will also use this value during power-up to determine whether the diagnostic control interface will use an 8-bit CRC or 8-bit checksum.

Table 2-3. CRCZ_CHKSUM_SEL Configuration

VALUE AT SYSTEM POWER-UP	SYSTEM ACTION
0	Host Control Interface uses CRC on all transactions Diagnostic Control Interface uses CRC on all transactions
1	Host Control Interface uses checksum on all transactions Diagnostic Control Interface uses checksum on all transactions

2.1.4 HOST_SPI_MODE

The HOST-SPI-MODE signal is used by the boot application during system power-up to determine what SPI mode (clock polarity and phase) is to be used for Host SPI communications. After the value of this signal is read by the boot application, it is configured as an output and toggled during stages of software initialization for debug usage. This debug usage is described in Section 2.1.4.1.

Table 2-4. HOST_SPI_MODE Configuration

VALUE AT SYSTEM POWER-UP	SYSTEM ACTION
0	Host SPI port may use SPI mode 0 or 3
1	Host SPI port may use SPI mode 1 or 2

The SPI modes are defined in Table 2-5.

Table 2-5. SPI Mode Definitions

MODE	CLOCK POLARITY	CLOCK PHASE
0	0	0
1	0	1
2	1	0
3	1	1

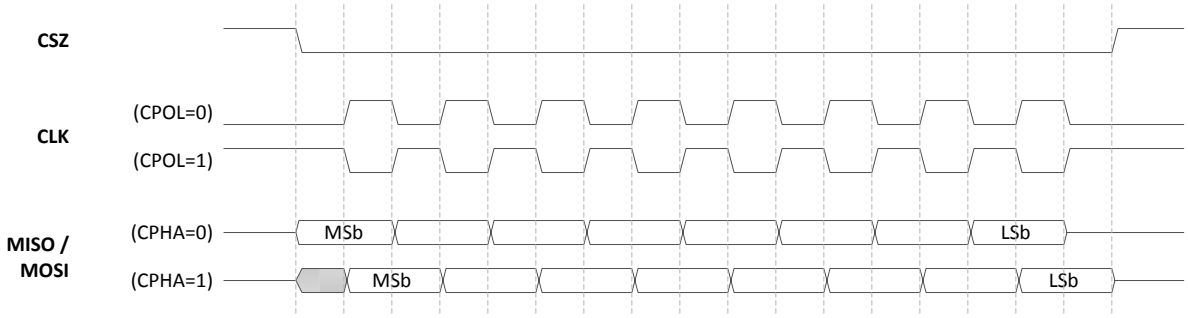


Figure 2-1. SPI Mode Timing

2.1.4.1 Boot Flow Debug

HOST_SPI_MODE is configured as an output and toggled during DLPC230-Q1 software initialization to aid in debug. This begins immediately after the value is read as an input. Figure 2-2 shows the initialization events and the associated HOST_SPI_MODE signal. A toggle will occur for optional events regardless of whether they are enabled or disabled so that events following can still be properly identified.

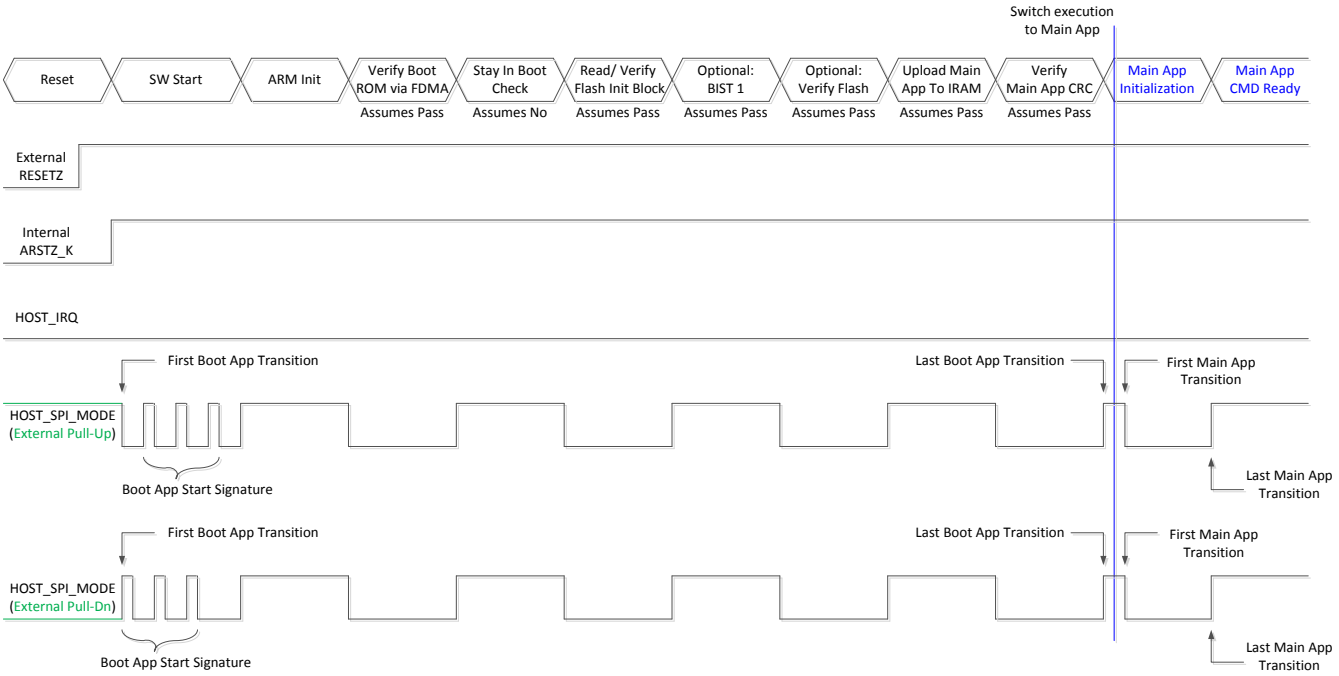


Figure 2-2. Boot Flow Debug

The tests are described in Section 6.4.2. BIST 1 is the DLPC230-Q1 Command and Flash Interface Memory Test.

2.1.5 SPREAD-SPECTRUM-DISABLE (TSTPT_5)

SPREAD-SPECTRUM-DISABLE is controlled by the state of the TSTPT_5 DLPC230-Q1 pin. This can be considered a hardware override of the spread spectrum function. If this is set HIGH, the spread spectrum settings will be read from flash memory. If this is set LOW, the spread spectrum settings will be ignored and spread spectrum will always remain disabled.

Note that a HIGH on this signal does not necessarily enable spread spectrum since the flash settings can also disable it.

Table 2-6. SPREAD-SPECTRUM-DISABLE Configuration

VALUE AT SYSTEM POWER-UP	SYSTEM ACTION
0	Spread spectrum disabled
1	Software will read spread spectrum settings from flash memory

2.2 System Signals

This section describes hardware signals related to software operation.

2.2.1 HOST_IRQ

The HOST_IRQ signal is an interrupt from the DLPC230-Q1 that is used to notify the host that a serious system error has occurred. This would generally result in an emergency shutdown. The HOST_IRQ signal will be set HIGH in this situation. Additional information on emergency shutdown can be found in Section 6.2.

2.2.2 GPIO

The DLPC230-Q1 has 32 GPIO. Some of these GPIO are reserved per product configuration, but the non-reserved pins are available for the host to control based on system needs. The reserved and available GPIO per product are described in the DLPC230-Q1 Data Sheet.

The following commands are used for GPIO control:

- *GPIO Configure - Write* - Configures each GPIO pin as an input or output.
- *GPIO Configure - Read* - Returns whether each GPIO is set as an input, output, or alternate function.
- *GPIO Outputs - Write* - Sets the value of each GPIO.
- *GPIO Outputs - Read* - Reads the value of each GPIO.
- *GPIO Reserved - Read* - Returns whether each GPIO is reserved or available for host use.

During development, the reserved GPIO can be determined by reading the *GPIO Reserved* command. The available GPIO can be configured as inputs or outputs through flash configuration or by sending the *GPIO Configure* command.

The *GPIO Outputs* command should be sent to change the HIGH/LOW state of any GPIOs during system operation. This command includes mask bits for each GPIO so that only the desired GPIO values will be set.

2.2.3 PWM Control

The DLPC230-Q1 includes 3 general purpose PWM signals for use in headlight applications. Example use-cases include illumination magnitude control or thermo-electric cooler control.

These PWM signals operates nominally at 136.7kHz. This nominal frequency will be affected if spread spectrum is enabled. The duty cycle of each PWM signal can be controlled using the *PWM Control* command.

Chapter 3

Communication Protocol

3.1 Command Protocol

The DLPC230-Q1 may be controlled by a host processor using either SPI or I^2C. The general command structure is identical for either communication type. The following sections provide an overview of the command protocol and byte diagrams for read and write commands.

Each command transaction is bounded by an SPI chip select or I^2C Start/Stop condition.

The transmitted command data includes a command op-code, a command tag, the payload length, the payload data, and a CRC or checksum byte. The tag and CRC/checksum are used for error checking on the communication interface for added robustness.

- The command tag is a unique value that can be traced through system execution in the Short Status and Error History to determine whether the command was successful or whether any errors occurred. More information on command tags can be found in Section 3.7.
- The CRC or checksum byte ensures integrity of the data being transmitted in case of electrical noise on the data lines or other byte-level corruption.

The received command data includes the repeated command op-code, repeated command tag, the return payload length, the return payload data, and a CRC or checksum for the returned bytes.

The state of the external signal *CRCZ-CHKSUM-SEL* at system initialization will dictate whether commands require a CRC or checksum of the payload data.

3.2 SPI Specifications

The SPI signal specifications are as follows:

Table 3-1. SPI Configuration

PARAMETER	VALUE
Clock Frequency	10 MHz maximum
Mode (Clock Phase and Polarity)	All 4 combinations of clock phase and polarity are supported. The external signal *HOST-SPI-MODE* must be configured properly at startup to set the desired mode.
Data Word Length	8-bit
Bit Order	Most significant bit first within a byte
Byte Order	Least significant byte first
Chip Select Polarity	Active low
Dummy Byte	From controller(host): 0x00. From peripheral (DLPC230): 0xFF

3.3 I²C Specifications

The I²C signal specifications are as follows:

Table 3-2. I²C Configuration

PARAMETER	VALUE
Clock Frequency	100 kHz, 400 kHz (Fast Mode)
Device Address (Target address and write/read bit)	36/37h
Data Word Length	8-bit
Bit Order	Most significant bit first within a byte
Byte Order	Least significant byte first

3.4 Write

There are two types of write transaction protocols: a short write and a bulk write. The short write is used for data payloads of 64 bytes or less. This is used for almost all boot and main application commands. The bulk write is used for data payloads of 65 bytes to 256 bytes. Bulk write is only used for programming flash data.

Section 3.4.3 defines a command handling flow for writing any data to the DLPC230-Q1.

3.4.1 Short Write

The Short Write protocol is used for write commands with a data payload of 64 bytes or less. Typical operating commands use this protocol.

3.4.1.1 SPI Short Write

Table 3-3, Table 3-4, and Figure 3-1 describe the SPI short write protocol.

MOSI Controller Output, Peripheral input (e.g. Host Output, DLPC230 input):

Table 3-3. SPI Short Write MOSI Bytes

DATA BYTE	DESCRIPTION
CMD	1-byte write command op-code
Tag	1-byte identifier used to uniquely mark each command sent by host.
Length	Length of the payload data.
Data1(a:n)	Payload data to be sent. Length and content varies based on the command op-code, up to 64 bytes.
CRC	CRC or checksum for the write bytes. This value must cover all bytes including op-code, tag, length, and data bytes.

MISO Controller input, Peripheral output (e.g. Host input, DLPC230 output):

Table 3-4. SPI Short Write MISO Bytes

DATA BYTE	DESCRIPTION
CMD (ACK)	Echo of the received command op-code from the host controller.
Tag (ACK)	Echo of the received tag from the host controller.
LengthR	Length of the return data. This is always zero for a short write.
CRC-R	CRC or checksum for the response bytes. This value covers op-code, tag, and length.

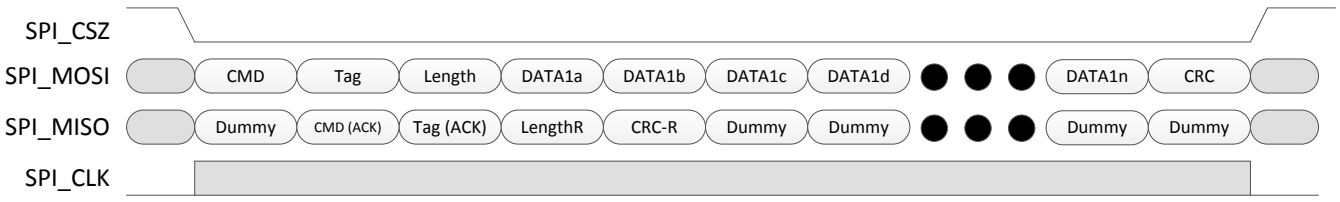


Figure 3-1. SPI Short Write

3.4.1.2 I²C Short Write

Table 3-5 and Figure 3-2 describe the I²C short write protocol.

Data:

Table 3-5. I²C Short Write Bytes

DATA BYTE	DESCRIPTION
CMD	1-byte write command op-code
Tag	1-byte identifier used to uniquely mark each command sent by host.
Length	Length of the payload data.
Data1(a:n)	Payload data to be sent. Length and content varies based on the command op-code, up to 64 bytes.
CRC	CRC or checksum for the write bytes. This value must cover all bytes including op-code, tag, length, and data bytes.

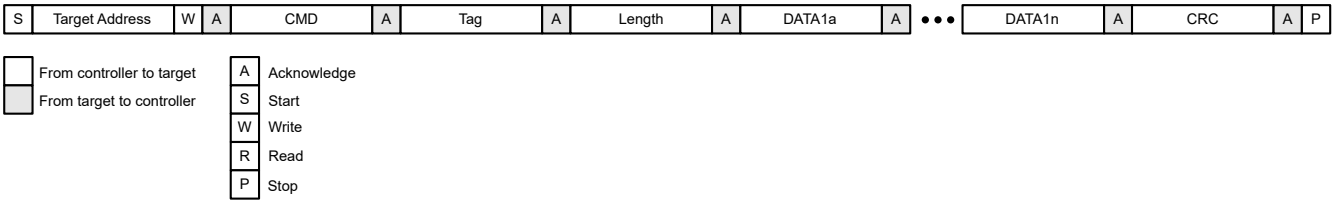


Figure 3-2. I²C Short Write

3.4.2 Bulk Write

The “Bulk Write” protocol is used for write commands with a data payload greater than 64 bytes up to and including 256 bytes.

3.4.2.1 SPI Bulk Write

Table 3-6, Table 3-7, and Figure 3-3 describe the SPI bulk write protocol.

MOSI Controller Output, Peripheral input (e.g. Host Output, DLPC230 input):

Table 3-6. SPI Bulk Write MOSI Bytes

DATA BYTE	DESCRIPTION
CMD	1-byte write command op-code
Tag	1-byte identifier used to uniquely mark each command sent by host.
Length	Length of the bulk header data (Data 1a, Data 1b). This is always 2 for a bulk write.
Data1(a:b)	Bulk header data. The value specifies the length of the bulk payload (1-256 bytes). LSByte = "a".
CRC1	CRC or checksum for the write header. Includes op-code, tag, length, and header data bytes.
Data2(a:n)	Bulk write payload data. The number of bytes will vary, and should match the value stored in Data1(a:b).
CRC2	CRC or checksum for the bulk payload. This only includes the bulk payload (Data2).

MISO Controller input, Peripheral output (e.g. Host input, DLPC230 output):

Table 3-7. SPI Bulk Write MISO Bytes

DATA BYTE	DESCRIPTION
CMD (ACK)	Echo of the received command op-code from the host controller.
Tag (ACK)	Echo of the received tag from the host controller.
LengthR	Length of the return data. This is always zero for a bulk write.
CRC-R	CRC or checksum for the response bytes. This value covers op-code, tag, and length.

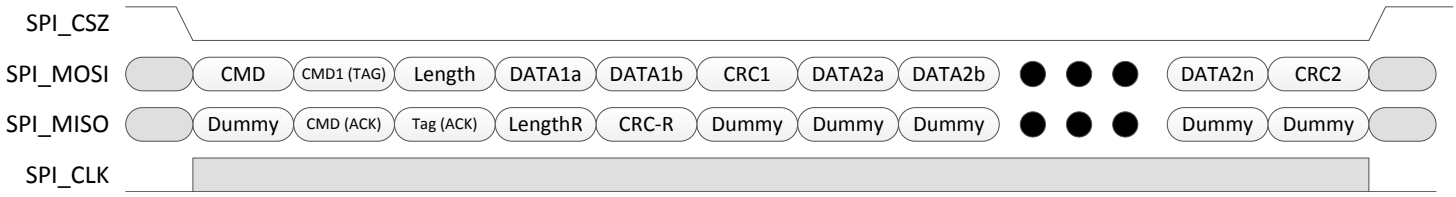


Figure 3-3. SPI Bulk Write

3.4.2.2 I²C Bulk Write

Table 3-8 and Figure 3-4 describe the I²C bulk write protocol.

Data:

Table 3-8. I²C Bulk Write Bytes

DATA BYTE	DESCRIPTION
CMD	1-byte write command op-code
Tag	1-byte identifier used to uniquely mark each command sent by host controller.
Length	Length of the bulk header data (Data 1a, Data 1b). This is always 2 for a bulk write.
Data1(a:b)	Bulk header data. The value specifies the length of the bulk payload (1-256 bytes).
CRC1	CRC or checksum for the write header. Includes op-code, tag, length, and header data bytes (Data1).
Data2(a:n)	Bulk write payload data. The number of bytes will vary, and should match the value stored in Data1(a:b).
CRC2	CRC or checksum for the bulk payload. This only includes the bulk payload (Data2).

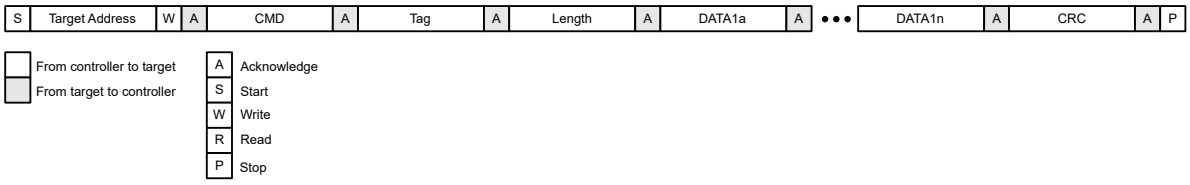


Figure 3-4. I²C Bulk Write

3.4.3 Write Command Handling

A write command handler must check the *Short Status* to ensure that the system is no longer busy from prior transactions and to check whether any errors have occurred. Errors could be related to the current command transaction or other background system processes.Figure 3-5 describes the flow of a command handler for any write command using either short or bulk write protocols.

More detail on the Short Status can be found in Section 8.2.87 (Main Application) and Section 7.2.6 (Boot Application). This command handler flow is valid for both applications.

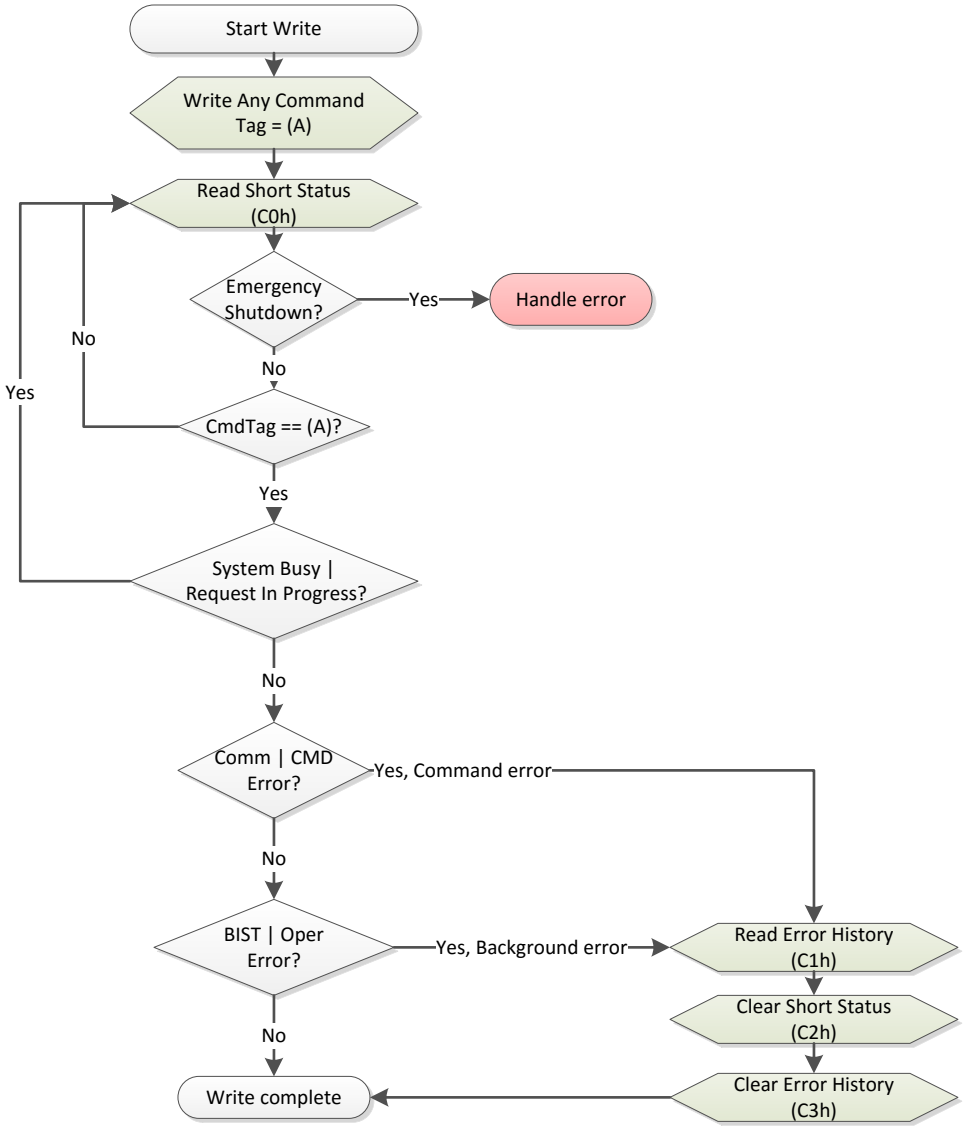


Figure 3-5. Write Command Handler

- **Write Any Command** - Write the desired command using the *Short Write Protocol* or *Bulk Write Protocol* as required for the specific command. Record the tag that was selected for this command (A).
- **Read Short Status** - Read the Short Status using the *Short Status Read Protocol*.
- **Emergency Shutdown?** - This Short Status bit indicates whether an emergency shutdown condition occurred. If this occurs, command handling should cease in order to handle the emergency shutdown. The *HOST_IRQ* signal will also trigger if this occurs. Refer to Section 6.2 for more information on emergency shutdown.

- **CmdTag == (A)?** - Check whether the Short Status contains the tag that was sent with the write command. This indicates that the command was executed by the software, regardless of whether it was successful or not. If this does not match the desired tag (A), continue polling the Short Status until the tag matches.
- **System Busy | Request In Progress?** - Check whether the *System Busy* or *Request In Progress* Short Status bits are set. If these are set, the system is still working on the write command and new commands should not be sent. If one of these bits is set, continue polling the Short Status until both are cleared.
- **COMM | CMD Error?** - Check for *Communication Error* or *Command Error* Short Status bits. If either are set, the write command has failed. In this situation, the *Error History* should be read to determine the specific error details. Once the Error History is read, the Error History and Short Status should be cleared of previous errors prior to the next transaction. The host can then determine what action to take such as re-sending the command or restarting the system.
- **BIST | Oper Error?** - Check for *BIST Error* or *Operational Error* Short Status bits. If either bit is set, a background process has encountered an error. If either error bit is set without the CMD error bit being set, the error is not related to the current command transaction. An example of this is a temperature sensor failure during regular main application polling. In this situation, the Error History should still be read to determine the cause of the error, but it does not mean that the current write command transaction has failed.

Note that only *Command Error* and *Communication Error* Short Status bits should be considered a command failure. If *BIST Error* or *Operational Error* bits are set without Command or Communication error bits, another process received an error but the command was still completed.

3.5 Read

Reading data is accomplished using two primary transactions: a *Read Pre-Fetch* command and a *Read Activate* command. The reason for this two-command read is to prevent the host from being tied to command execution. The host can request data, handle other processes while DLPC230-Q1 embedded software executes the request, and later read the requested data. Section 3.5.4 defines a command handling flow for reading any data from the DLPC230-Q1 using these two transactions and the Short Status.

3.5.1 Read Pre-Fetch

Read Pre-Fetch is a Short Write command used to request a read of a specific command op-code. A successful write of *Read Pre-Fetch* will initiate the read data process. The transmitted data payload is the desired read command op-code and its associated command parameters. The command parameters vary per command and are described in Section 7.2 (Boot Application) and Section 8.2 (Main Application). No read payload is received from this read pre-fetch step.

3.5.1.1 SPI Read Pre-Fetch

Figure 3-6 describes the SPI Read Pre-Fetch protocol.

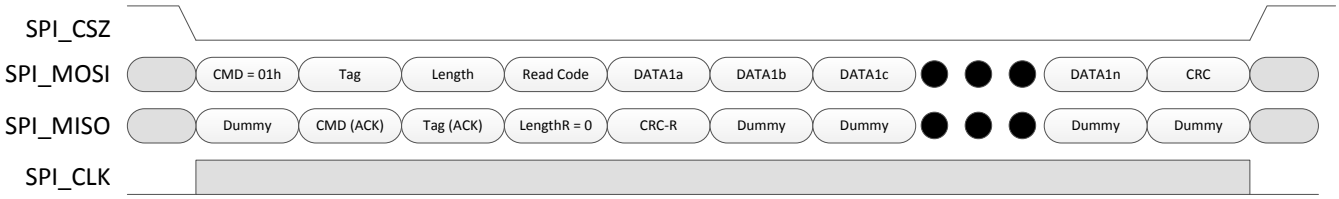


Figure 3-6. SPI Read Pre-Fetch

3.5.1.2 I²C Read Pre-Fetch

Figure 3-7 describes the I²C Read Pre-Fetch protocol.

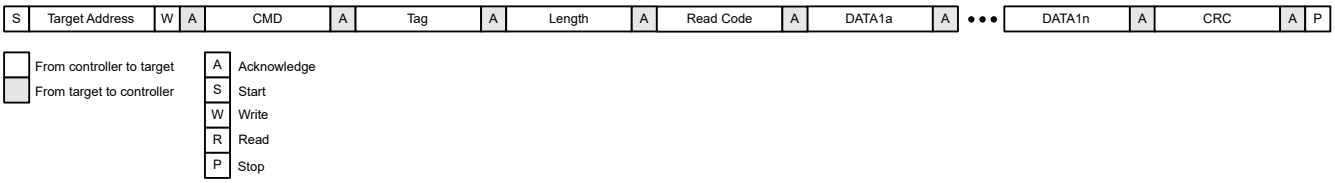


Figure 3-7. I²C Read Pre-Fetch

3.5.2 Read Activate

The *Read Activate* command reads the data that was requested by the *Read Pre-Fetch* command. The same op-code is used regardless of the data that is being read, but the read data length will vary depending on the requested data.

3.5.2.1 SPI Read Activate

Table 3-9, Table 3-10, and Figure 3-8 describe the SPI Read Activate protocol.

MOSI Controller Output, Peripheral input (e.g. Host Output, DLPC230 input):

Table 3-9. SPI Read Activate MOSI Bytes

DATA BYTE	DESCRIPTION
CMD	1-byte write command op-code. This is 02h for Read Activate.
Tag	1-byte identifier used to uniquely mark each command sent by host.
Length	Length of the transmitted data payload. This is always 0 for read activate.
CRC	CRC or checksum for the read header data. This includes the op-code, tag, and length.

MISO Controller input, Peripheral output (e.g. Host input, DLPC230 output):

Table 3-10. SPI Read Activate MISO Bytes

DATA BYTE	DESCRIPTION
CMD (ACK)	Echo of the received command op-code from the host controller.
Tag (ACK)	Echo of the received tag from the host controller.
LengthR	Length of the read header data (Data1a, Data1b). The value is always 2 for read activate.
Data1(a:b)	Read header data. Specifies the length of the read return payload (0-256 bytes). LSByte = "a".
CRC-R1	CRC or checksum for the header response bytes. This includes the op-code, tag, length, and read return header data (Data1).
Data2(a:n)	Read return payload. The number of bytes varies depending on the read command and is specified by the value in Data1(a:b).
CRC-R2	CRC or checksum for the read return payload. This only includes the read return payload (Data2) bytes.

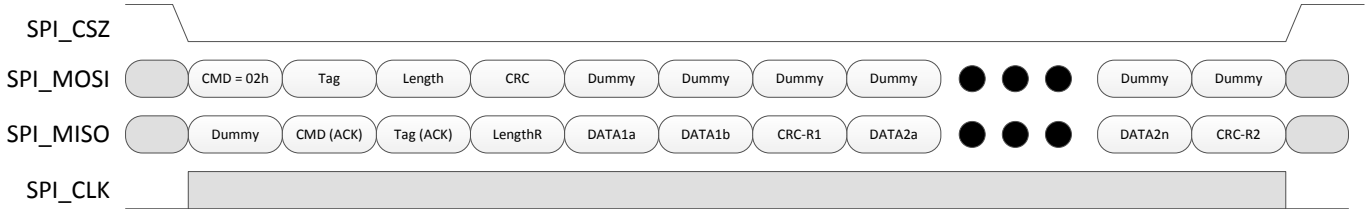


Figure 3-8. SPI Read Activate

3.5.2.2 I²C Read Activate

Table 3-11, Table 3-12, and Figure 3-9 describe the I²C Read Activate protocol.

Data Write:

Table 3-11. I²C Read Activate Write Bytes

DATA BYTE	DESCRIPTION
CMD	1-byte write command op-code
Tag	1-byte identifier used to uniquely mark each command sent by host controller.
Length	Length of the transmitted data payload. This is always 0 for read activate.
CRC	CRC or checksum for the read header data. This includes the op-code, tag, and length.

Data Read:

Table 3-12. I²C Read Activate Read Bytes

DATA BYTE	DESCRIPTION
CMD (ACK)	Echo of the received op-code from the host controller.
Tag (ACK)	Echo of the received tag from the host controller.
LengthR1	Length of the read header data (Data1a, Data1b). This is always 2 for read activate.
Data1(a:b)	Read header data. Specifies the length of the read return payload (0-256 bytes).
CRC-R1	CRC or checksum for the read header data. This includes the op-code, tag, length, and read return header data (Data1).
Data2(a:n)	Read return payload.
CRC-R2	CRC or checksum for the read return payload. This only includes the read return payload (Data2).

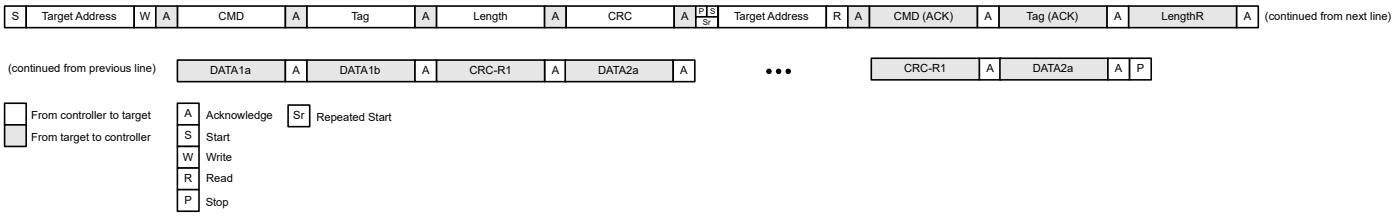


Figure 3-9. I²C Read Activate

3.5.3 Short Status Read

Short Status is a unique read transaction that is hardware supported and does not require embedded software intervention. This means that the status data can be received in one transaction in a fixed amount of time unlike typical read transactions which require two transactions. More details about the contents of the *Short Status* can be found in Section 8.2.87 (Main Application) and Section 7.2.6 (Boot Application). The protocol is similar to a *Read Activate* command, and is shown in Section 3.5.3.1:

3.5.3.1 SPI Short Status Read

Table 3-13, Table 3-14, and Figure 3-10 describe the SPI Short Status read protocol.

MOSI Controller Output, Peripheral input (e.g. Host Output, DLPC230 input):

Table 3-13. SPI Short Status MOSI Bytes

DATA BYTE	DESCRIPTION
CMD1	1-byte write command op-code. This is C0h for Short Status.
Tag	1-byte identifier used to uniquely mark each command sent by host controller.
Length1	Length of the transmitted data payload. This is always 0 for short status.
CRC1	CRC or checksum for the read header data. This includes the op-code, tag, and length.

MISO Controller input, Peripheral output (e.g. Host input, DLPC230 output):

Table 3-14. SPI Short Status MISO Bytes

DATA BYTE	DESCRIPTION
CMD1	Echo of the received command op-code from the host controller.
Tag	Echo of the received tag from the host controller.
LengthR1	Length of the short status. The value is always 4 bytes.
Data1(a:d)	Short status data. This data is defined in Section 8.2.87 (Main Application) and Section 7.2.6 (Boot Application).
CRC-R1	CRC or checksum for the read data. This includes the op-code, tag, length, and short status data.

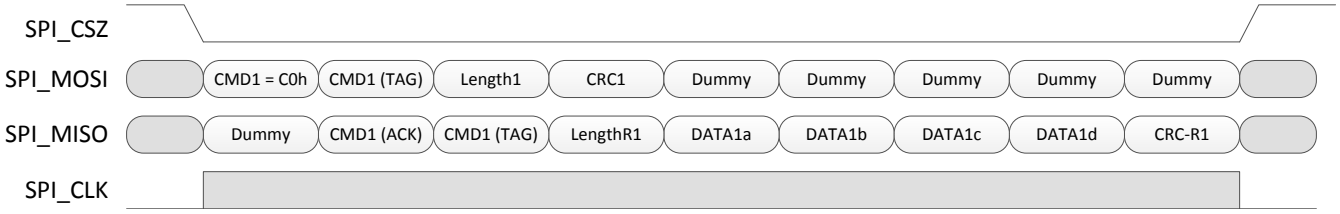


Figure 3-10. SPI Short Status

3.5.3.2 I²C Short Status Read

Table 3-15, Table 3-16, and Figure 3-11 describe the I²C Short Status Read protocol.

Data Write:

Table 3-15. I²C Short Status Write Bytes

DATA BYTE	DESCRIPTION
CMD	1-byte write command op-code. This is C0h for Short Status.
Tag	1-byte identifier used to uniquely mark each command sent by host controller.
Length	Length of the transmitted data payload. This is always 0 for Short Status.
CRC	CRC or checksum for the read header data. This includes the op-code, tag, and length.

Data Read:

Table 3-16. I²C Short Status Read Bytes

DATA BYTE	DESCRIPTION
CMD (ACK)	Echo of the received op-code from the host controller.
Tag (ACK)	Echo of the received tag from the host controller.
LengthR	Length of the short status data. (Data1a, Data1b). This is always 4 for Short Status.
Data1(a:d)	Short Status data. This data is defined in Section 8.2.87 (Main Application) and Section 7.2.6 (Boot Application).
CRC-R	CRC or checksum for the read header data. This includes the op-code, tag, length, and Short Status data.

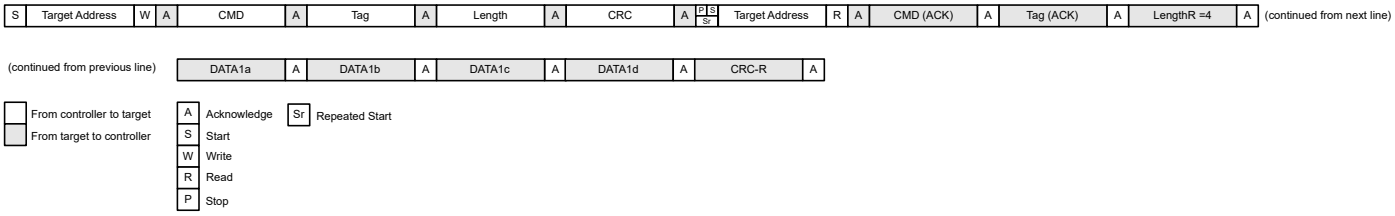


Figure 3-11. I²C Short Status

3.5.4 Read Command Handling

A read command handler must check the *Short Status* to determine whether read data is available and to check whether any errors have occurred. Errors could be related to the current command transaction or other background software processes. Figure 3-12 describes the flow of a command handler for read commands.

More detail on the Short Status can be found in Section 8.2.87 (Main Application) and Section 7.2.6 (Boot Application). This command handler flow is valid for both applications.

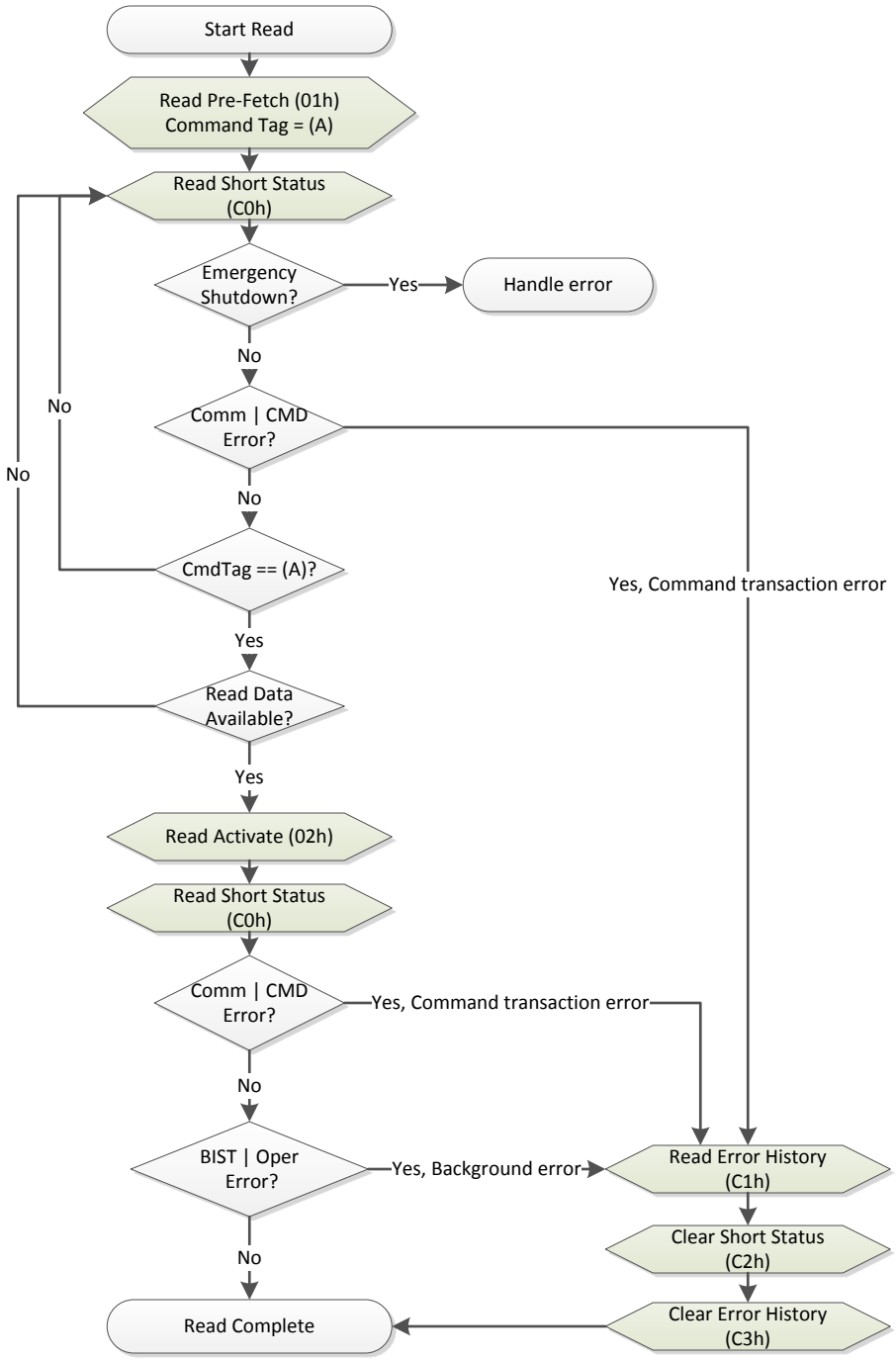


Figure 3-12. Read Command Handler

- **Read Pre-Fetch** - Send a read command op-code and its associated command parameters using the *Read Pre-Fetch Protocol*. Record the tag that was selected for this command (A).
- **Read Short Status** - Read the Short Status using the *Short Status Read Protocol*.
- **Emergency Shutdown?** - This Short Status bit indicates whether an emergency shutdown condition occurred. If this occurs, command handling should cease in order to handle the emergency shutdown. The *HOST_IRQ* signal will also trigger if this occurs. Refer to Section 6.2 for more information on emergency shutdown.
- **COMM | CMD Error?** - Check for *Communication Error* or *Command Error* Short Status bits. If either are set, the Read Pre-Fetch command has failed. In this situation, the *Error History* should be read to determine the specific error details. Once the Error History is read, the Error History and Short Status should be cleared of previous errors prior to the next transaction. The host can then determine what action to take such as re-sending the Read Pre-Fetch command or restarting the system.
- **CmdTag == (A)?** - Check whether the Short Status contains the tag that was sent with the Read Pre-Fetch command. This indicates that the Read Pre-Fetch was executed by software. If this does not match the desired tag (A), continue polling the Short Status until the tag matches.
- **Read Data Available?** - Check whether the *Read Data Available* bit is set in Short Status. If it is set, the requested read data is available. If it is not set, continue polling the Short Status.
- **Read Activate** - Once the read data is available, a Read Activate command may be used to read the requested data using the *Read Activate Protocol*.
- **Read Short Status** - Read the Short Status using the *Short Status Read Protocol* to check for any errors after the Read Activate.
- **COMM | CMD Error?** - Check for *Communication Error* or *Command Error* Short Status bits. If either are set, the read activate command has failed. In this situation, the Error History should be read to determine the specific error details.
- **BIST | Oper Error?** - Check for *BIST Error* or *Operational Error* Short Status bits. If either bit is set, a background process encountered an error. If these error bits are set without the CMD error bit being set, the error is not related to the current command transaction. An example of this is a temperature sensor failure during regular main application polling. In this situation, the Error History should still be read to determine the cause of the error, but it does not mean that the current read command transaction has failed.

Note that only *Command Error* and *Communication Error* Short Status bits should be considered a command failure. If *BIST Error* or *Operational Error* bits are set without Command or Communication error bits, another process received an error but the command still completed.

3.6 CRC and Checksum

The usage of CRC or checksum can be selected at startup based on the state of the external DLPC230-Q1 signal, *CRCZ_CHKSUM_SEL*.

3.6.1 CRC Calculation

The CRC uses CRC-8 CCITT which has the following implementation:

- Polynomial $x^8 + x^2 + x + 1$ (normal polynomial representation = 0x07)
- Initial value is 0xFF

3.6.1.1 CRC Example Implementation

The following pseudo-code demonstrates one possible method for implementing the CRC calculation.

"^" represents bitwise exclusive OR operator.

"IN" is the data byte that is being included in the CRC.

"CRC" is the CRC byte that is being calculated.

```
CRC(0)(7:0)=0xFF; # Initial seed value is 0xFF.
FOR x=1 to N; # Loop through all "N" data bytes that are included in the CRC.
CRC(x)(0)=IN(x)(0)^IN(x)(6)^IN(x)(7)^CRC(x-1)(0)^CRC(x-1)(6)^CRC(x-1)(7);
CRC(x)(1)=IN(x)(0)^IN(x)(1)^IN(x)(6)^CRC(x-1)(0)^CRC(x-1)(1)^CRC(x-1)(6);
CRC(x)(2)=IN(x)(0)^IN(x)(1)^IN(x)(2)^IN(x)(6)^CRC(x-1)(0)^CRC(x-1)(1)^CRC(x-1)(2)^CRC(x-1)(6);
CRC(x)(3)=IN(x)(1)^IN(x)(2)^IN(x)(3)^IN(x)(7)^CRC(x-1)(1)^CRC(x-1)(2)^CRC(x-1)(3)^CRC(x-1)(7);
CRC(x)(4)=IN(x)(2)^IN(x)(3)^IN(x)(4)^CRC(x-1)(2)^CRC(x-1)(3)^CRC(x-1)(4);
CRC(x)(5)=IN(x)(3)^IN(x)(4)^IN(x)(5)^CRC(x-1)(3)^CRC(x-1)(4)^CRC(x-1)(5);
CRC(x)(6)=IN(x)(4)^IN(x)(5)^IN(x)(6)^CRC(x-1)(4)^CRC(x-1)(5)^CRC(x-1)(6);
CRC(x)(7)=IN(x)(5)^IN(x)(6)^IN(x)(7)^CRC(x-1)(5)^CRC(x-1)(6)^CRC(x-1)(7);
END FOR;
```

3.6.1.2 CRC Example

This example demonstrates the values generated by the CRC calculation.

"IN" Data bytes = 0xC0, 0x3A, 0x04, 0x89, 0x39, 0x13, 0x30.

Table 3-17. CRC Example Calculation

IN(x)	CRC(x)	COMMENTS
	0xFF	Initial seed CRC value
0xC0	0xBD	CRC after byte 1
0x3A	0x9C	CRC after byte 2
0x04	0xC1	CRC after byte 3
0x89	0xFF	CRC after byte 4
0x39	0x5C	CRC after byte 5
0x13	0xEA	CRC after byte 6
0x30	0x08	CRC after byte 7

The final CRC of this data is 0x08.

3.6.2 Checksum Calculation

Command checksums can be calculated by summing all of the bytes that are to be included in the checksum. This sum value should be masked with 0xFF to receive the checksum value.

Example checksum calculation of bytes: 0xA0, 0xBD, 0xCF, 0x85.

Sum = 0x2B1.

Checksum = 0xB1.

3.7 Command Tags

A command tag is required to be sent with each command in order to track command execution. The host command interface should only use a specific subset of the available byte values as shown in Table 3-18. Other values are reserved for other system usages, such as commands that are executed from batch command sets stored in flash.

During typical usage, the host should increment the tags from 01h through CFh for each command sent and then loop back to 01h. This provides a unique identifying value for the command transaction so that the host can verify whether the command has completed.

Table 3-18. Command Tag Allocations

TAG SOURCE	TAG RANGE	DESCRIPTION
Reserved	00h	Default value at system initialization and when the Short Status is cleared.
Host	01h - CFh	Host command interface
Batch Command Set	D0h - DFh	Batch command set from flash. These are automatically determined by embedded software.
Diagnostic Interface	E0h - EFh	Diagnostic read-only host interface
Reserved	F0h - FEh	Reserved
Null Tag Response in Short Status	FFh	Tag unknown or corrupted. Should not be used by host.

Chapter 4

System Operation

4.1 Operating Modes

The DLPC230-Q1 embedded software consists of two applications: the boot application and the main application. The main application includes several states, or modes, which determine the system's allowable commands and functionality at any point during operation. Typically, the *Operating Mode Select* command is used to transition between modes, but there are additional system conditions that can automatically trigger a mode transition. The modes and their interactions are described in the following sections.

The state diagram in Figure 4-1 provides an overview of the DLPC230-Q1 power-up procedure and the various software state transitions.

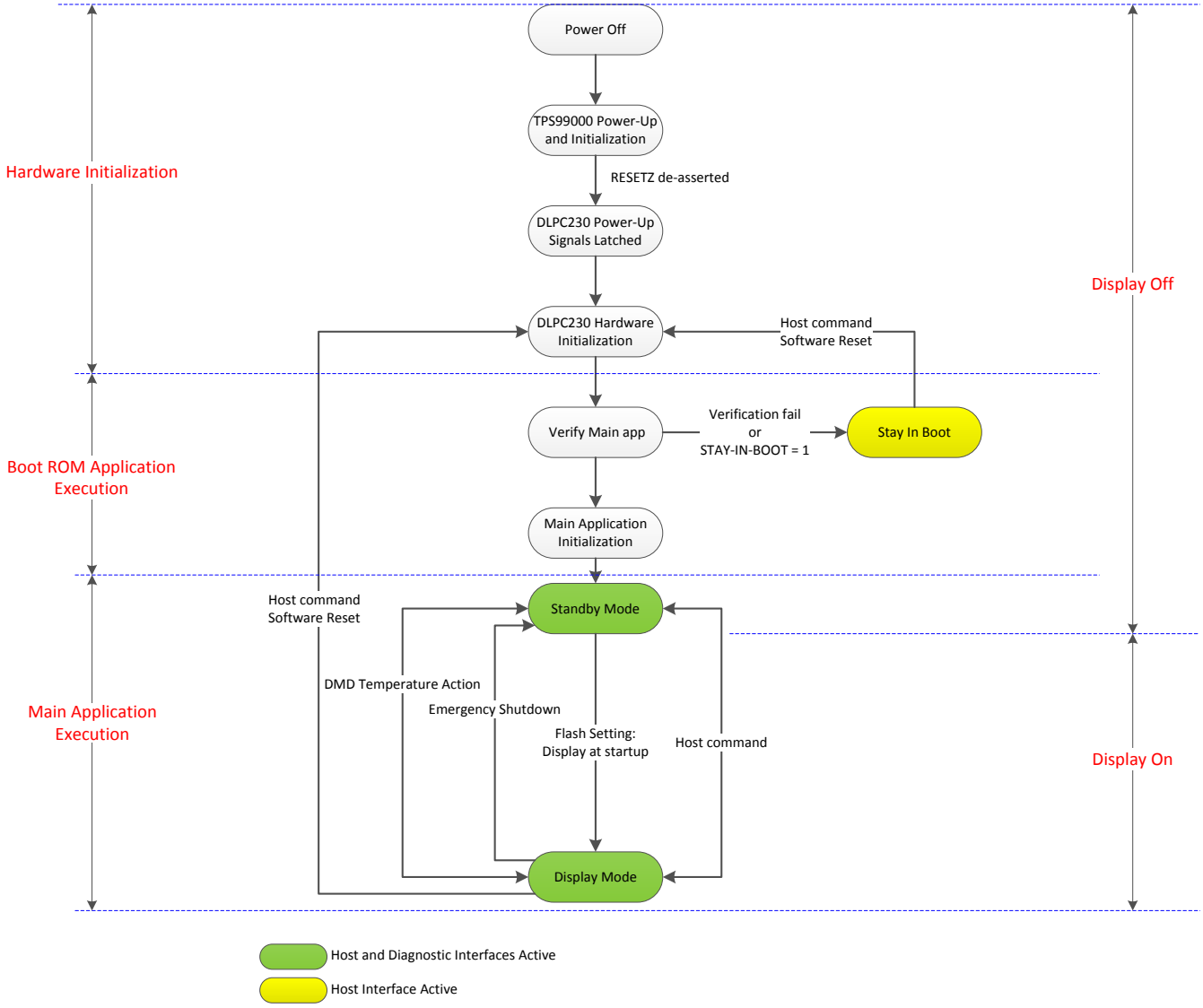


Figure 4-1. DLPC230-Q1 State Diagram

The following table summarizes general feature availability per software application and operating mode. This is not a full feature list and exceptions may exist. This is provided as an initial reference of the general purpose of each mode. The command availability tables in Chapter 7 and Chapter 8 provide more detail on the modes in which commands are available.

Table 4-1. Software Mode Feature Summary

	Boot Application	Main Application - Standby	Main Application - Display
Short Status / Error History	Yes	Yes	Yes
Full Flash Program	Yes	Yes	No
Partial Flash Program	No	Yes	No
Non-Periodic BISTs	No	Yes	No
Source Display	No	No	Yes
Periodic BISTs	No	No	Yes

4.1.1 Standby

Standby Mode is the first mode that the main application enters when the boot application transitions execution to the main application. No video is displayed, illumination is disabled, and the DMD remains parked. The primary features of Standby Mode are flash programming and non-periodic BIST execution, which cannot be executed in other operating modes.

When Standby Mode is entered, a selection of non-periodic BISTs can be automatically executed based on flash options. Once this is completed, a flash option determines whether the software will remain in Standby Mode or automatically transition to Display Mode. If the flash option to transition to Display Mode is not selected, the software will wait in Standby Mode until the host sends a command to transition to another mode.

4.1.2 Display

Display Mode is the primary mode of operation to display video data. Periodic BISTs, such as video source checks, will be executed in this mode. The following commands may not be executed in this mode:

- Flash programming
- Non-Periodic BISTs

4.2 Software Startup Procedure

At startup, the host may begin sending commands to control the device once the *System Initialized* flag has been set in the Short Status. This may be set by either the boot application or the main application. More detail on the Short Status can be found in Section 8.2.87 (Main Application) and Section 7.2.6 (Boot Application). The host should expect to receive NAKs (I^2C) or invalid data (SPI) before the *System Initialized* flag is set. Figure 4-2 demonstrates the actions that the host should take at startup. Section 2.1.4.1 describes a hardware signal that can be used to debug startup timing and progress.

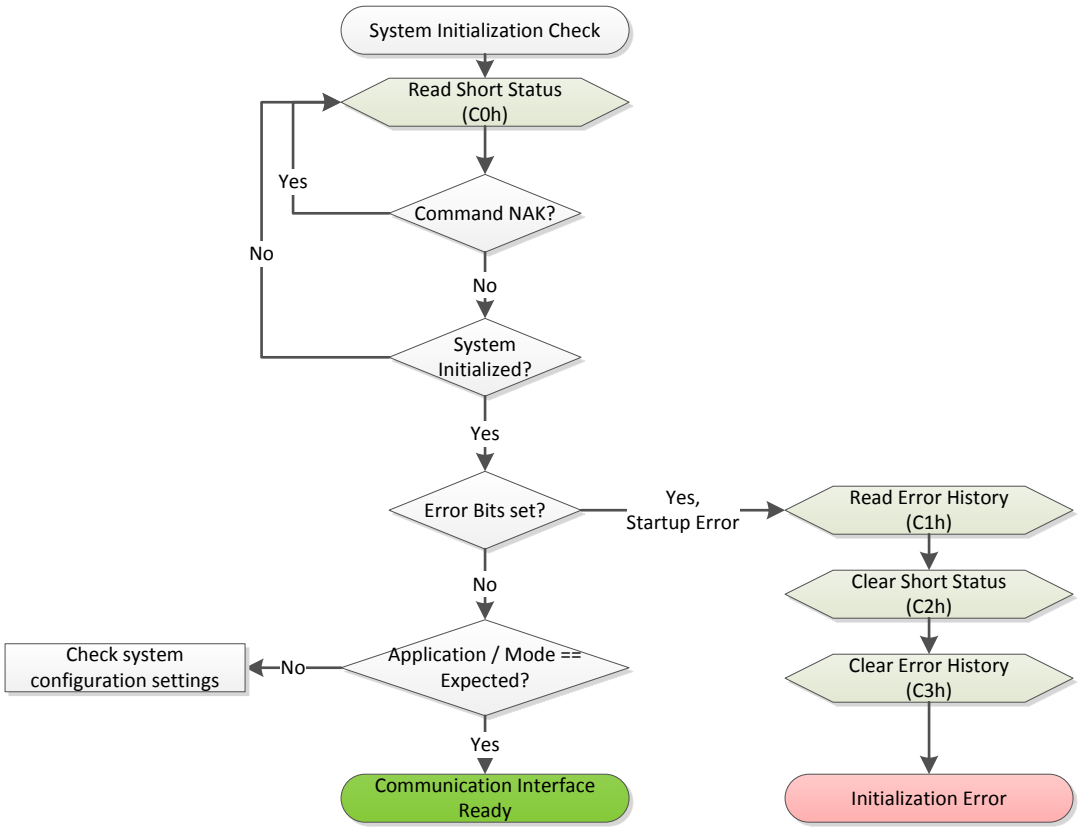


Figure 4-2. Communication Ready Check

- **Read Short Status** - Attempt to read the Short Status from hardware.

- **Command NAK?** - If communication hardware has not yet been initialized, a NAK or invalid data will be received.
- **System Initialized?** - If the Short Status was received successfully, check the *System Initialized* bit. If it is cleared, continue polling the Short Status until it is set.
- **Error Bits Set?** - If any of the Short Status error bits are set, a startup error occurred. Examples of possible startup errors include startup BIST failures or auto-initialization batch command set execution errors. The *Error History* should be read for details on the errors and then the Short Status and Error History should be cleared to clear the prior error flags. The host may then determine what action to take based on the error conditions.
- **Application / Mode == Expected?** - If no error bits are set, the application and operating mode should be checked to confirm that the software is in the expected state. If the application or mode are incorrect and no error is set, there are several system configuration parameters that can be checked. Check that the STAY-IN-BOOT signal is pulled to the desired state. This determines whether the main application is executed or if execution remains in the boot application. Check that the flash header file indicates that the correct main application operating mode (Standby or Display) is selected for the initial startup mode.

The boot application resides in boot ROM in the DLPC230-Q1. The boot application is executed after DLPC230-Q1 hardware has initialized at startup. The purpose of the boot application is to load the main application from flash and verify its contents during start-up. The boot application execution flow is shown in Figure 4-3. It reads the hardware pin signals described in Section 2.1 to configure the host communication interface and then it determines whether *STAY-IN-BOOT* has been set. If *STAY-IN-BOOT* is set, the boot application will enter a loop waiting for host-commanded instructions. If the *STAY-IN-BOOT* signal is not set, the boot application will load the main application from flash and verify it. If verification fails, an error will be recorded in the Error History and the boot application will enter the wait-for-command loop. If verification succeeds, the main application will be executed.

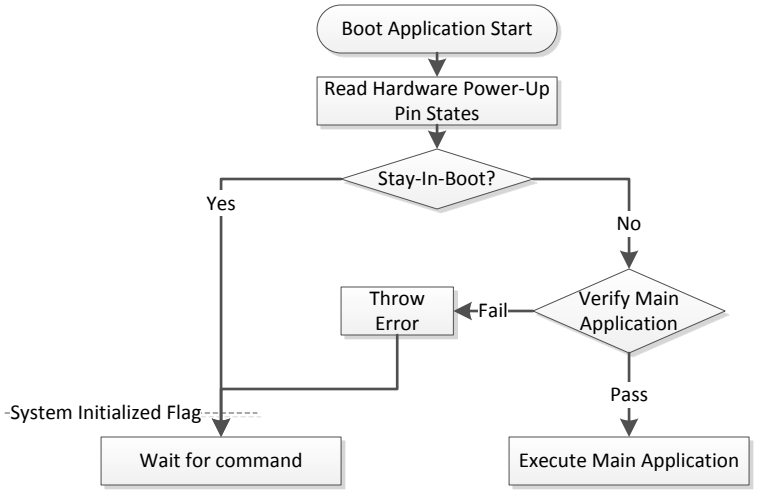


Figure 4-3. Boot Application Execution Flow

The main application execution flow is shown in Figure 4-4. The main application begins by initializing clocks and then retrieves the cause of the last reset from hardware. It then executes any non-periodic BISTs that are selected through flash options. The main application will check a flash option to determine whether it should stay in Standby Mode or automatically switch to Display Mode. Once it has entered the desired mode, the auto-initialization batch command set will be run from flash memory. More information on batch command sets can be found in Section 5.2. The auto-initialization batch command set can be empty if there are no commands to be executed during start-up. The auto-initialization batch command set can also change the operating mode. Once the auto-initialization batch command set is complete, the main application will set the *System Initialized* Short Status flag to indicate that command transactions may be sent from the host.

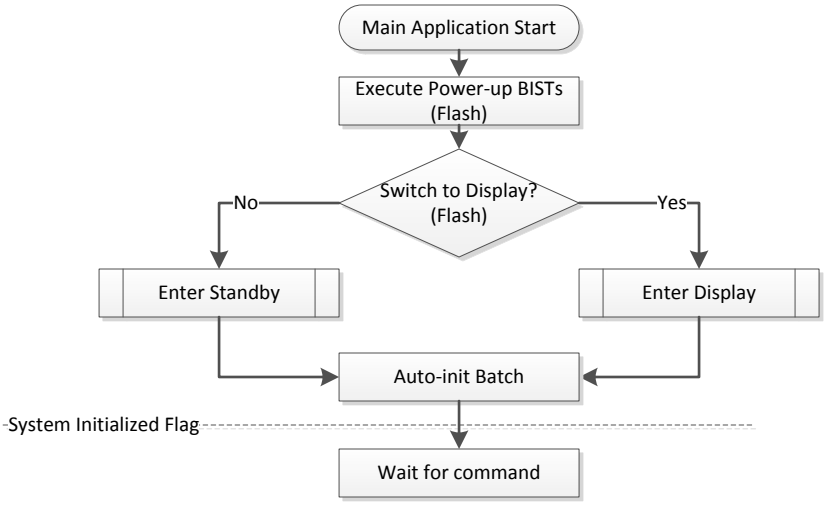


Figure 4-4. Main Application Execution Flow

Chapter 5

Application and Use-Cases

5.1 Display and Source

The DLPC230-Q1 supports several video source types: external parallel and openLDI video interfaces, internal test pattern generation, and splash images stored in flash.

5.1.1 Displaying an Image

The parameters for the available source configurations are stored in batch command sets in flash. To configure a source, the desired batch command set index should be executed. Once the batch command set is executed, the *Prepare for Source Change* command will disable illumination and the software will hold the configuration parameters until the *Source Select* command is sent. This two-command process provides time to prepare the external video source signals without displaying corrupted video data and causing the software to throw video source detection errors. During normal operation, an external video source should be adjusted between these two commands.

An image may only be displayed in Display Mode .

The source change steps are summarized as follows:

1. *Execute Batch Command Set* command - Configures the desired display source parameters. This configuration will be held by software and will not be applied until the *Source Select* command is sent.
2. *Prepare For Source Change* command - Disables illumination and external source error-checking to transition without displaying corrupted video data.
3. Configure external video signals as needed if external source is being displayed. For example, adjust video resolution to match the new batch command set configuration.
4. *Source Select* command - Applies the source configuration. Illumination will then be re-enabled and the desired source will be visible on the display.

If multiple external video inputs are used, the following steps must be used to switch between them. This assumes source 1 is being displayed:

1. *Operating Mode* command - Transition to Standby Mode.
2. *Execute Batch Command Set* command - Configures the desired display source parameters for source 2.
3. Configure external video signals.
4. *Source Select* command - Holds the source configuration until operating mode select.
5. *Operating Mode* command - Transition to Display Mode . Illumination will then be re-enabled and the desired source will be visible on the display.

5.1.2 Supported Image Processing

Some DMDs, such as the DLP5530S-Q1 and the DLP4620S-Q1, have a diamond mirror configuration. These type of DMDs allow the light to enter directly from the bottom of the DMD optically, which can result in smaller and more efficient optical design. Refer to Figure 5-1 as an example.

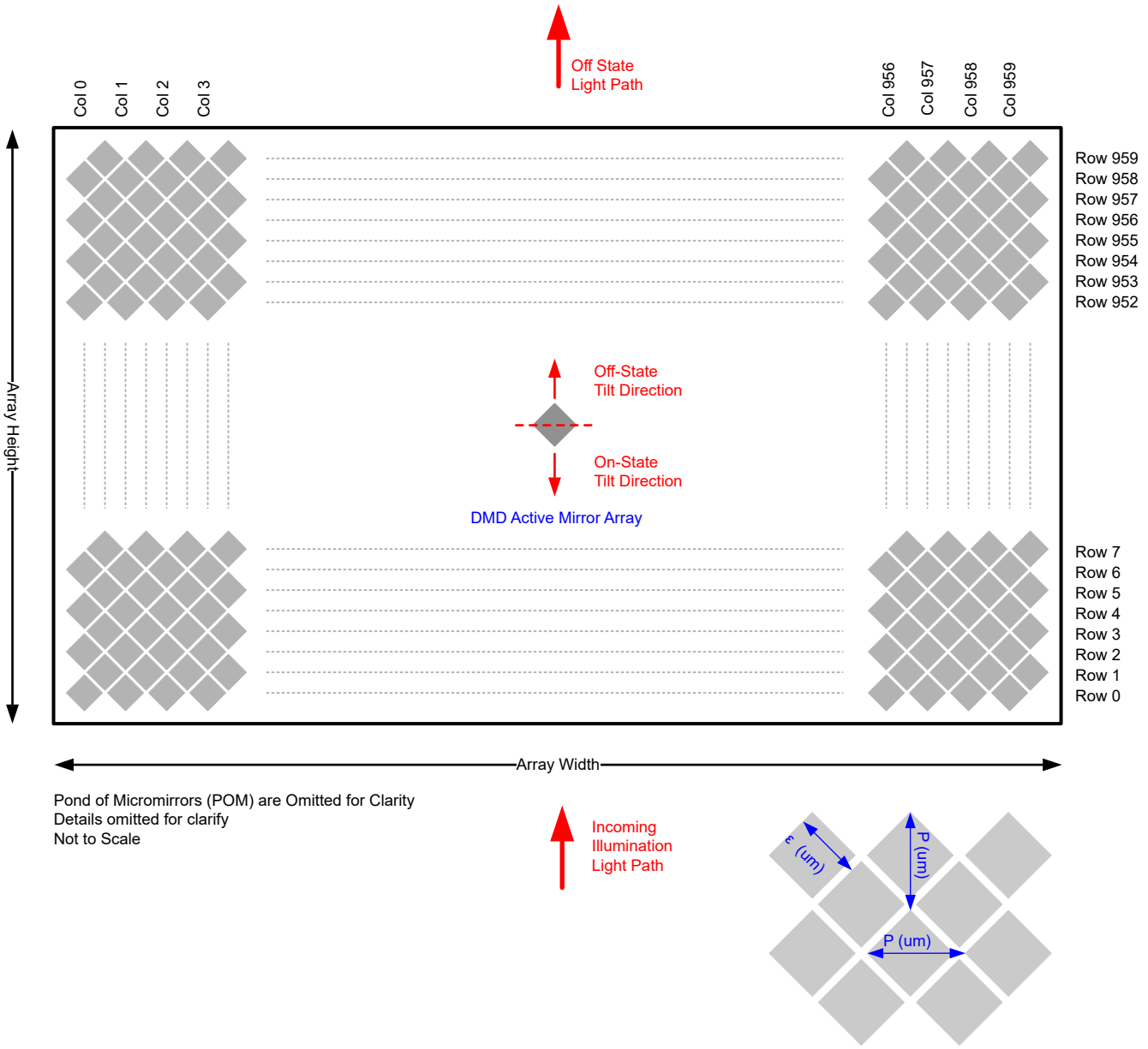


Figure 5-1. DLP4620S-Q1 Mirror Diagram

The diamond mirror configuration enables Quincunx processing which can represent a higher effective resolution than the total number of mirrors. Specifically, the horizontal and vertical representation of data can represent twice as much resolution as the number mirrors. In this fully maximized mode, the diagonal information is sacrificed, however due to the limitations of the human visual system, the reduced resolution in the diagonal is less observable by the typical viewer. Quincunx processing is widely used in consumer electronics to maximize resolution for the fewest number of physical pixels, including smart phones for example.

Different processing modes supported by the DLPC23x-Q1 when combined with the DMD with diamond configuration can be seen in Table 5-1.

Table 5-1. Image Processing Modes

Mode	Input source description	Operations performed by DLPC23x-Q1 and DMD
Quincunx pre-processed data Refer to Figure 5-2 for an example on converting source data from a manhattan configuration to a diamond configuration	Host MCU performs Quincunx processing Example: source data is converted from 1920 x 960 manhattan into 960 x 960 diamond	Data is passed through to display on the DMD without further processing
DLPC23x-Q1 performs Quincunx processing (possible when the internally scaled image is less than 1.5Mbits) Refer to Figure 5-3 for an example on having a higher resolution Manhattan configuration	Example: 1358 x 566 manhattan input source data	DLPC23x-Q1 scales data up to 1920 x 800 and applies a diamond bandpass filter to remove high frequency diagonal content, then the DLP4620S-Q1 deciminates every other pixel where the number of physical mirrors used for display is 960 x 800
Lower resolution Manhattan input that must be scaled up to be displayed on the DMD Refer to Figure 5-4 for an example pertaining to a Manhattan source configuration with resolution lower than the DMD	Example: 960 x 480 input to be display on 960 x 960 diamond DMD	DLPC23x-Q1 scales data up to 960 x 960 where the DMD displays the scaled information

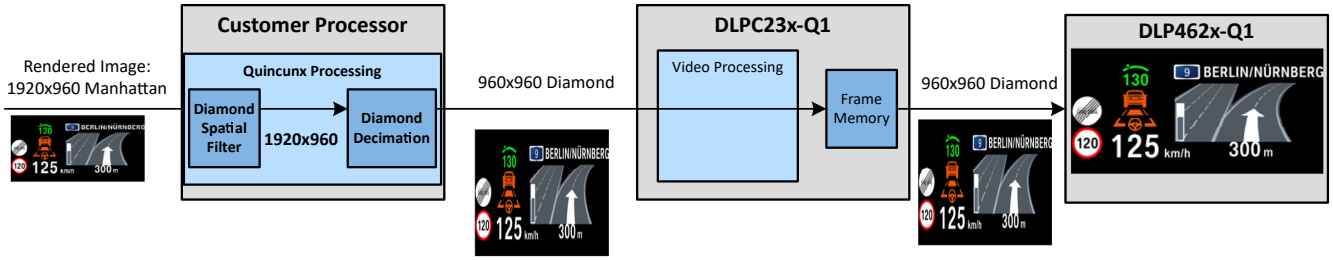


Figure 5-2. Pre-processed Quincunx example

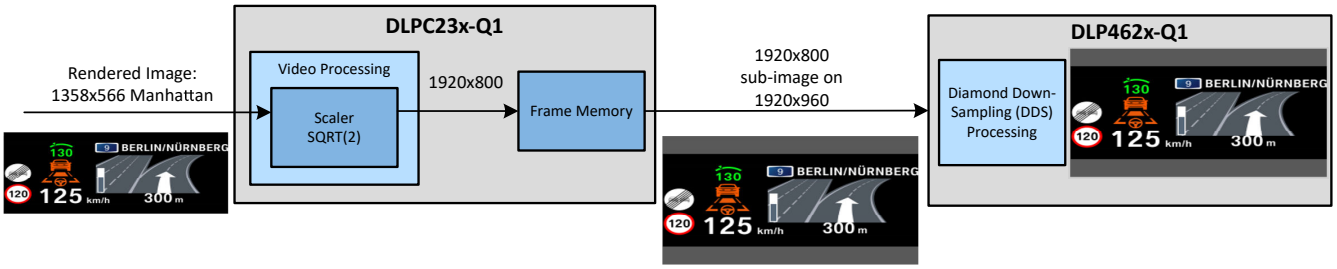


Figure 5-3. High Resolution Manhattan Example

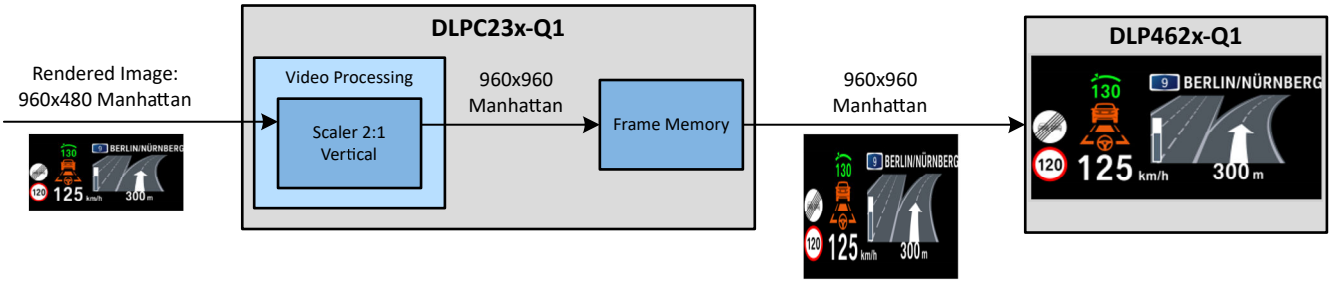


Figure 5-4. Manhattan source with Resolution lower than DMD

Contact TI (E2E)for further explanation of the benefits of Quincunx processing to increase resolution.

5.1.3 External Video

Parallel and openLDI interfaces can be configured at various resolutions. The supported resolutions are described in the DLPC230-Q1 Data Sheet.

Important note for DLP462xx-Q1 series of DMDs. The input resolution selected must be compatible with the DMD chosen. For example, the DLP4620S-Q1 DMD will support input resolution up to 1358x566. However, other DMDs may limit the maximum input resolution. When the resolution selected is higher than the limit for a specific DMD, the system will not transition to external video mode. Please review the specific DMD datasheet to determine resolutions supported for that DMD.

5.1.4 Test Pattern

The DLPC230-Q1 includes an internal test pattern generator which can be used to display basic test images and verify system functionality without an external video source. Figure 5-5 through Figure 5-13 show the test patterns that may be generated.

Note that the test patterns below assume that the system uses red, green, and blue light sources. Test patterns displayed using single-color systems (including full-white light systems) may not appear as shown below.

Solid Color (White, Red, Green, Blue)

Figure 5-5. Test Pattern - Solid Color

Color Bars (8-color)

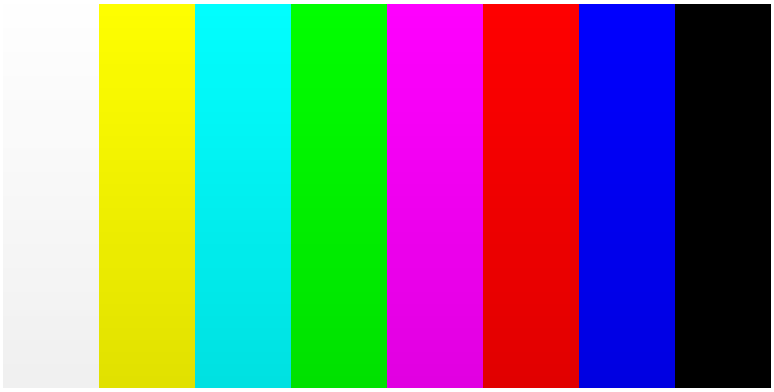


Figure 5-6. Test Pattern - Color Bars

Checkerboard (White/Black)

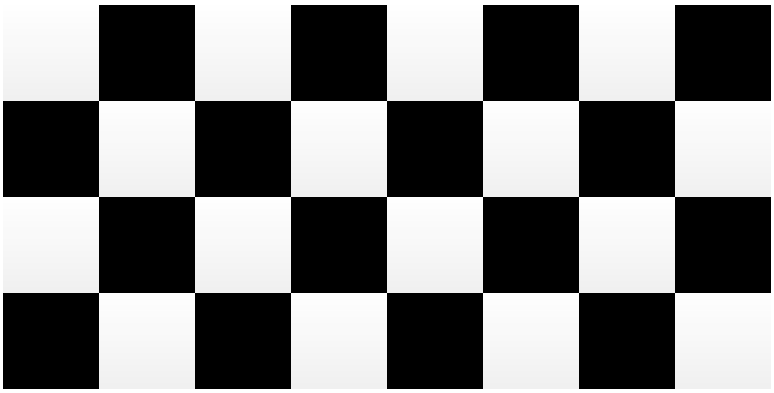


Figure 5-7. Test Pattern - Checkerboard

Horizontal Ramp (White/Black)

Figure 5-8. Test Pattern - Horizontal Ramp

Vertical Ramp (White/Black)

Figure 5-9. Test Pattern - Vertical Ramp

Grid Lines (White, Black)

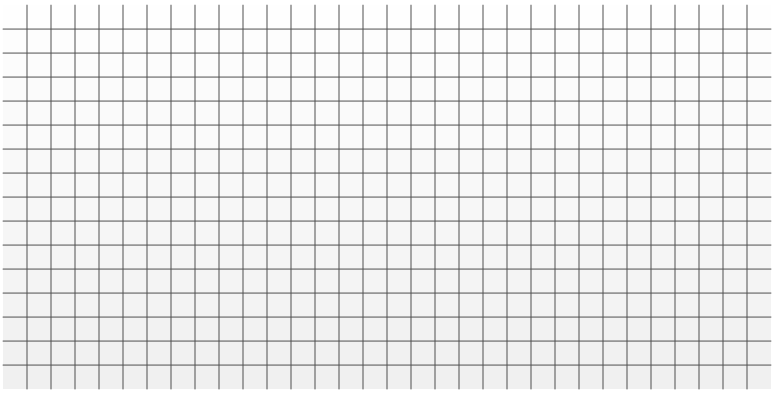


Figure 5-10. Test Pattern - Grid White

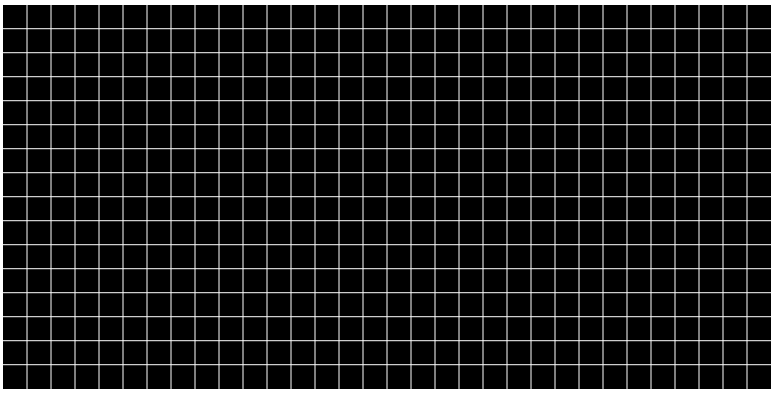


Figure 5-11. Test Pattern - Grid Black

Diagonal Lines (White, Black)

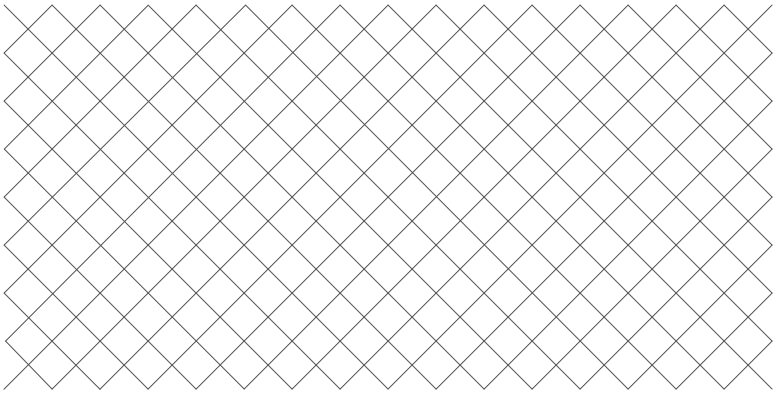


Figure 5-12. Test Pattern - Diagonal White

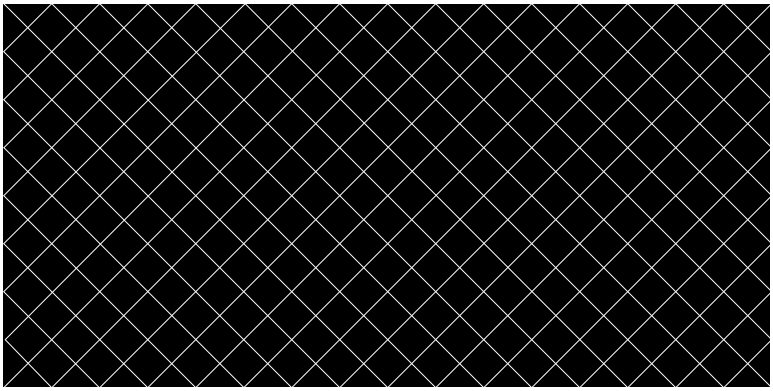


Figure 5-13. Test Pattern - Diagonal Black

5.1.5 Splash Image

Splash images are individual binary images stored in flash memory. A maximum of 8 splash images can be stored in flash if there is sufficient memory space. Splash images require slightly less than 0.5 seconds to load from flash when requested, during which illumination will be disabled.

Depending on the DMD part number, different splash resolutions are supported. For example, for 0.55" DMDs, splash resolutions include:

- 1152 x 576
- 1152 x 1152 (No scaling. 1:1 aspect ratio input image will display with 2:1 aspect ratio on the DMD due to diamond pixel architecture)

For 0.46" DMDs, splash resolutions include:

- 480 x 240
- 960 x 480
- 960 x 960 (No scaling. 1:1 aspect ratio input image will display with 2:1 aspect ratio on the DMD due to diamond pixel architecture)
- 1358 x 566 & 1220 x 610 - For part numbers that support diamond down sampling

Non-supported resolutions may be scaled to a supported resolution, cropped, or bordered with black pixels before being stored in flash memory.

Supported image formats are:

- RGB 888
- YCbCr 422

The YCbCr pixel format results in smaller binary sizes. This can reduce the flash size requirements and flash programming time, but the image conversion may reduce the image quality slightly.

5.1.6 Image Flip

Image flip can be used to compensate for folds in the optical path. The default image flip settings are stored in flash and the values are specified in the flash header file. Image flip values may also be set using the *Display Image Orientation* command. This command is a "Hold Source" command which means that changes will not be applied until the source is changed. To do this, the *Prepare for Source Change* command should be sent followed by the *Source Select* command in order to apply the source changes. These steps are shown as follows:

1. *Display Image Orientation* command - Set the new desired flip setting.
2. *Prepare For Source Change* command - Disable illumination to prepare for a source update without image artifacts.
3. *Source Select* command - Selects the desired source and displays the image with the updated flip setting.

Flip can be performed along two axes: a short axis flip and a long axis flip. Flips along both axes are shown in Figure 5-14 and Figure 5-15. Any combination of these two flip settings is permitted.

Figure 5-14. Short Axis Flip

Figure 5-15. Long Axis Flip

5.2 Batch Command Sets

Batch command sets are groups of commands that are stored in flash. Executing a batch command set that is stored in flash requires one command to be sent from the host: the *Execute Batch Command Set* command. Once the host requests execution of a desired batch index, the DLPC230-Q1 software will loop through all of the instructions stored in that batch command set. During this execution time the *Request In Progress* flag will be set in the *Short Status* to indicate that the batch command set is executing and that no other commands should be sent from the host.

The primary use of batch command sets is to configure the video source to pre-configured parameters stored in flash. For example, one batch command set can configure 1152 x 576 parallel video, and another can configure a solid white test pattern. The available batch command sets and their indexes may change based on flash configuration.

Only main application write commands can be used for batch command sets. The boot application cannot execute batch command sets. Read commands are not possible through batch command sets. The table below describes commands that cannot be executed through batch command sets.

Table 5-2. Commands Excluded from Batch Use

COMMAND	OP CODE
System Reset	00h
Read Pre-Fetch	01h
Read Activate	02h
Execute Batch Command Set	21h
Execute Non-Periodic BIST	28h
Flash Data Type Select	A0h
Flash Erase Data	A1h
Flash Write Data	A2h
Flash Verify Data	A4h
All read commands	Various
All boot application commands	Various

Batch index 0 is reserved for an auto-initialization batch command set that is automatically run by the software main application when it enters Standby Mode during the startup procedure.

The batch command sets that are available in a flash file can be found in the flash header file that is delivered along with the flash data.

5.3 Flash Programming

Flash programming can be performed by the boot application or the main application in Standby mode. Each application has its own procedure, and these are described in the following sections. Reading flash may only be accomplished using the main application in any mode.

5.3.1 Flash Program - Main Application

Flash programming may be performed by the main application in Standby mode. The entire flash may be programmed, or a flash block may be selected for partial flash updates. The main application determines whether each flash block is stored in flash or EEPROM, if it is in use. Therefore the procedure is identical for EEPROM programming.

The following commands are used to erase, program, and verify a flash block:

- *Flash Data Type Select*
- *Flash Erase Data*
- *Flash Write Data*
- *Flash Verify Data*

Figure 5-16 demonstrates the flash programming steps. The steps to write each command must follow the write command handler procedure described in Section 3.4.3.

1. Write *Flash Data Type Select* - Sets the desired flash block or blocks to be erased and re-programmed.
2. Write *Flash Erase Data* - Erases the selected flash block(s). This process may require significant time to complete. The *Write Command Handler* should expect additional time to poll the Short Status to confirm that the command tag has changed to the appropriate value and that the *Request In Progress* bit has been cleared before proceeding to the next step.
3. *Flash Write Data* - Repeatedly write up to a page (256 bytes) of flash data. The data size being written must be a multiple of 4. After each section of data is written, read the short status repeatedly until the write has completed. Continue until there is no remaining flash data.
4. *Flash Verify Data* - Begin verification of the flash. This process may require significant time to complete. The *Write Command Handler* should expect additional time to poll the Short Status to confirm that the command tag has changed to the appropriate value and that the *Request In Progress* bit has been cleared. A flash verification error will flag the *Command Error* bit in the Short Status. This should be detected in the write command handling procedure.

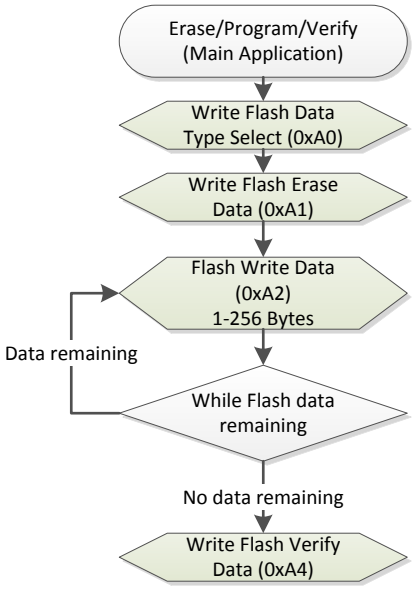


Figure 5-16. Flash Program Main Application

5.3.2 Flash Read - Main Application

Flash read may be performed by the main application in Standby mode. The entire flash may be read, or a flash block may be selected to be read. The main application determines whether each flash block is stored in flash or EEPROM, if it is in use. Therefore the procedure is identical for reading EEPROM.

The following commands are used to read flash data:

- *Flash Data Size*
- *Flash Read Data*
- *Read Pre-fetch*
- *Read Activate*

Figure 5-17 demonstrates the flash programming steps.

1. Read *Flash Data Size* - Read the size of the desired flash data type. This determines how many reads need to be performed.
2. *Flash Read Data* - Read up to a page (256 bytes) of flash data. Note that the data length is 0-based, so 0 = 1 byte. The length in bytes must be a multiple of 4.
3. Data read ≥ Flash Data Size - Once the amount of data that has been read equals the flash data size, reading should terminate.

The read command handler flow described in Section 3.5.4 should be used for each of these read command transactions to ensure that the read data is available prior to a retrieval attempt.

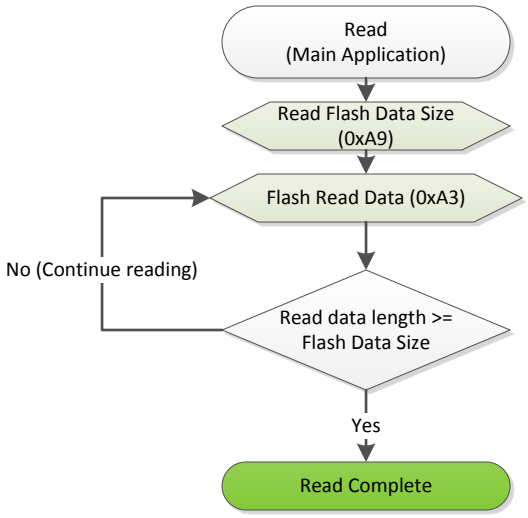


Figure 5-17. Flash Read Main Application

5.3.3 Flash Program - Boot Application

Flash programming may also be performed by the boot application. The boot application can only program the full flash; it is not capable of partial flash updating and it cannot program EEPROM data. The primary usage of boot flash programming is for situations when the system cannot transition to the main application, such as an empty or corrupted flash.

The following commands are used to erase, program, and verify flash:

- *Flash Interface Rate*
- *Flash Full Erase*
- *Flash Write Data*
- *Flash Verify Data*

Figure 5-18 demonstrates the flash programming steps.

1. Write *Flash Interface Rate* - Notifies the Boot Application of the maximum allowable flash interface speed and the read modes that the flash part supports. This is used to maximize flash SPI bandwidth in order to shorten flash programming time based on the selected flash part's capabilities. If this command is not sent prior to flash programming, a default speed of 10 MHz and only Fast Read will be used which may significantly slow programming operations. This default may also cause issues with the flash verification step since it does not meet the DLPC230 minimum flash bandwidth requirement.
2. Write *Flash Full Erase* - Erases the entire flash.
3. *Flash Write Data* - Repeatedly write a page (256 bytes) of flash data. If the desired data is less than 256 bytes, the remaining bytes should be set to 0xFF. After each section of data is written, read the short status repeatedly until the write has completed. Continue until there is no remaining flash data.
4. Write *Flash Interface Rate* - This step is required if a quad read mode is desired. The boot application will disable quad mode in some flash devices during write operations. This step ensures that quad mode is enabled in the flash device prior to flash verification. The same data may be written for this step and step 1.
5. *Flash Verify Data* - Begin verification of the flash data. This process may require significant time to complete. The *Write Command Handler* should expect additional time to poll the Short Status to confirm that the command tag has changed to the appropriate value and that the *Request In Progress* bit has been cleared. A flash verification error will flag the Command Error tag in the Short Status. This should be detected in the write command handling procedure.

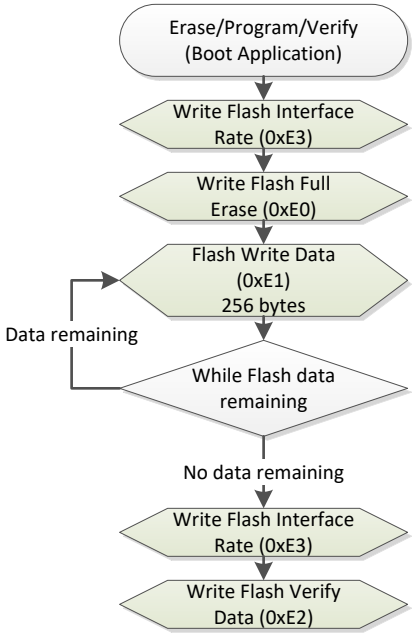


Figure 5-18. Flash Program Boot Application

5.4 Video Frame and Illumination Bin Delay

This section defines the input video delay and command processing delay for illumination bins while displaying external video.

The DLPC230-Q1 has a 1 frame processing delay between external input video data and displayed data.

The DLPC230-Q1 has a 3 frame delay between receiving an illumination bin index and applying the illumination bin. This delay only applies if the *Illumination Transition Rate* is set to 0, otherwise the transition rate will further delay the application of the illumination bin. Additionally, this delay only applies while displaying external video or a built-in test pattern. Displaying a splash image will extend the time taken to apply an illumination bin due to the time required to reload the splash image from flash.

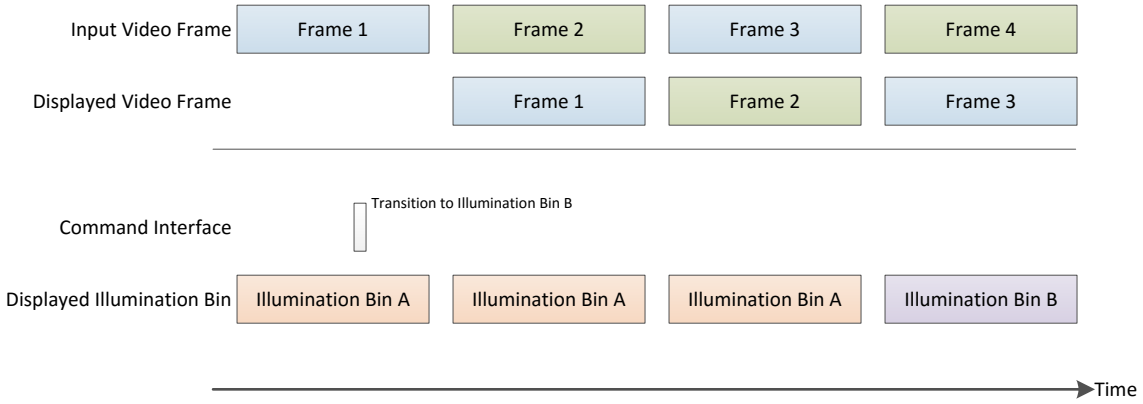


Figure 5-19. External Video Delay and Illumination Bin Delay

Figure 5-19 assumes that the illumination bin command is received during a specific window of time within the video frame. The write illumination bin command transaction must be complete within a specific time frame relative to the VSYNC active edge to ensure a consistent 3 frame delay. Table 5-3 defines these constraints. If the command is received outside of that time frame the illumination bin will still be applied, but the exact frame delay becomes uncertain (2-4 frames).

Table 5-3. Illumination Bin Synchronization Timing Parameters

PARAMETER		MIN	UNIT
t_{sync_open}	Minimum time between VSYNC active edge (polarity-dependent) and illumination bin command transaction completion.	4	ms
t_{sync_close}	Minimum time between illumination bin command transaction completion and next VSYNC active edge (polarity-dependent).	1	ms

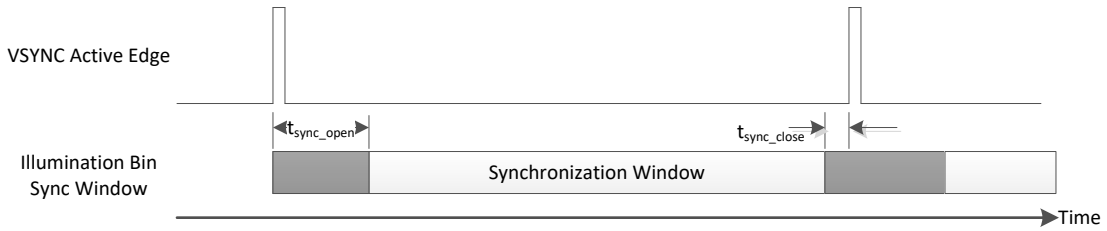


Figure 5-20. Illumination Bin Synchronization Window

5.5 Smooth Illumination Transition

The main application supports smooth transitions between sequences, or illumination bins, using digital contrast adjustments over a series of frames in order to attenuate the display brightness.

This feature prevents instantaneous changes in system brightness caused by large duty cycle differences in illumination bins.Figure 5-21 demonstrates an instantaneous brightness change that would occur without the use of this feature.

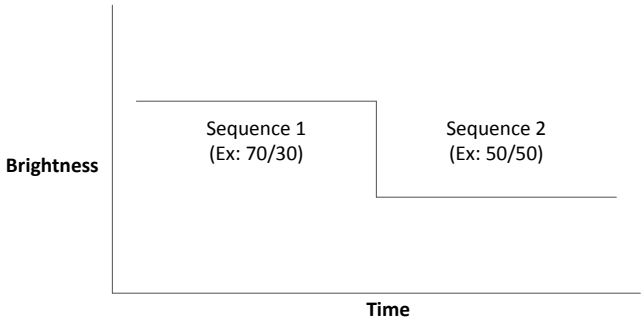


Figure 5-21. Instantaneous Duty Cycle Change

The following commands are used for this feature:

- *Illumination Transition Rate* - Specifies the number of frames over which the transition will occur. This effectively determines the time of transition based on the frame period multiplied by the number of frames. Changes to the brightness are calculated and applied each frame despite the step size allowed by this command.
- *Illumination Bin Select* - Specifies the illumination bin index that will be transitioned to. This command also triggers the start of the transition.

Note that this feature is only available while displaying external video or test patterns. It is not available while displaying splash images. While a splash image is displayed, the transition rate value is ignored and the commanded illumination bin will always be applied in one step.

When an *Illumination Bin Select* command is received, the main application begins transitioning brightness levels from the full digital brightness of the currently displayed illumination bin to the full digital brightness of the selected illumination bin. The main application uses the latest transition rate that was received prior to the start of the transition.

If an *Illumination Bin Select* command is received during an ongoing transition, the main application will terminate the ongoing transition by immediately switching to the last selected illumination bin. It will then start a new transition from the previously selected illumination bin to the newly selected illumination bin.

Figure 5-22 demonstrates the procedure that the main application performs to transition from a high brightness sequence to a low brightness sequence across 8 frames. Initially the system remains in the high brightness sequence and the digital contrast is decreased each frame so that the contrast-adjusted brightness of sequence 1 approaches the full-contrast brightness of sequence 2. On the last transition frame the contrast is returned to 1x and the sequence is transitioned to sequence 2. Figure 5-23 highlights the resulting total brightness of the combination of sequence and contrast brightness.

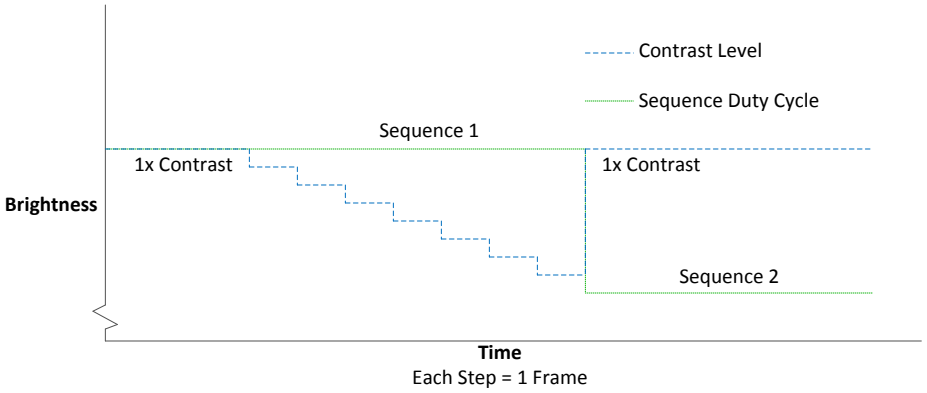


Figure 5-22. Duty Cycle Decrease

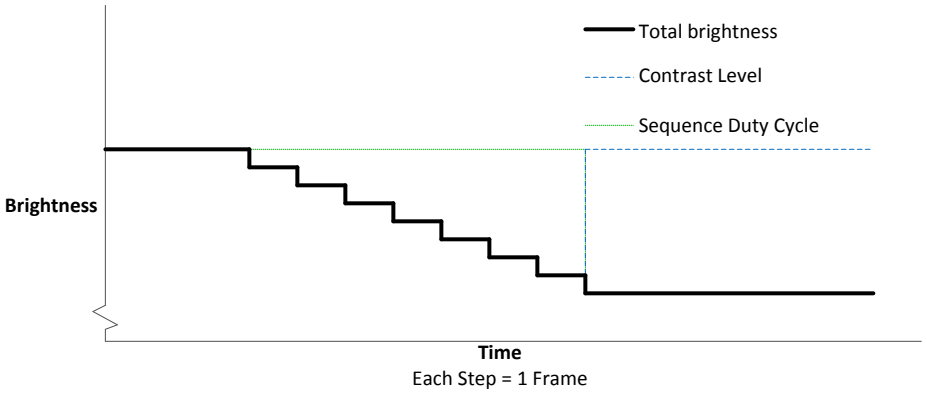


Figure 5-23. Duty Cycle Decrease Total Brightness

Figure 5-24 demonstrates the use of this function to transition from a low brightness sequence to a high brightness sequence across 8 frames. On the first transition frame, the sequence transitions to the high brightness sequence and the contrast is decreased to approximately match the low brightness sequence. Then the contrast is increased across the transition frames until the contrast returns to 1x. Figure 5-25 highlights the resulting total brightness of the combination of sequence and contrast brightness.

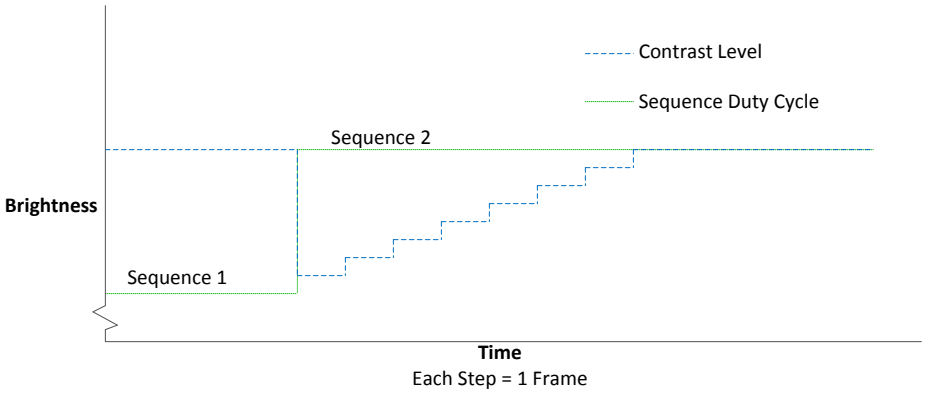


Figure 5-24. Duty Cycle Increase

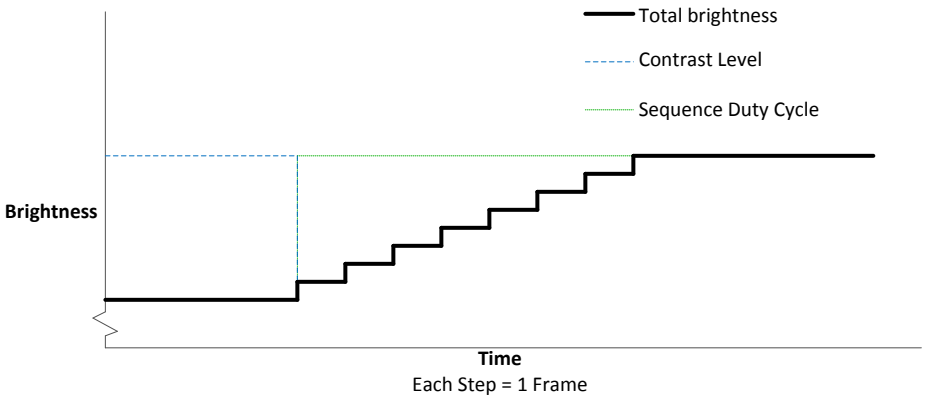


Figure 5-25. Duty Cycle Increase Total brightness

5.6 Temperature Management

The DLPC230-Q1 chipset supports a wide temperature operating range for automotive applications. The following functions are used for temperature management. These include functions to Park and Unpark the DMD if operating temperate is too high or too low. PWM function to control fan speed based on temperature feedback.

5.6.1 Temperature Management for DMD Park/Unpark

The following command is used for temperature management to Park the DMD when the operating temperature is too high or too low, and Unpark the DMD when the temperature is within the recommended operating temperature range.

- *System Temperatures - Read* - Read the DMD and system temperature. This command also notifies the host when the sequence park timer is active.

During system operation, the main application monitors the DMD temperature using I^2C communication to an external TMP411 connected to a temperature sensing diode built into the DMD. This value is filtered using weighted averaging across multiple samples to reduce the impact of signal noise.

The flash header file provided along with the flash data define maximum and minimum park temperature conditions. If either the maximum or minimum park temperature condition is met, the main application will immediately enter Standby mode to park the DMD and disable illumination. The main application will remain in Standby mode until the host requests a transition to any other mode. The host will be unable to switch to Display mode until the DMD temperature has reached the related unpark temperature.

The DMD data sheet specifies the DMD operating temperature range. The maximum and minimum DMD park temperatures that are set in the flash project may not match the DMD operating temperature range if these threshold configurations are modified. The DMD must only operate within its data sheet specified temperature limits regardless of these threshold temperature settings.

The following figure provides an overview of the full DMD temperature range and the actions taken by the main application.

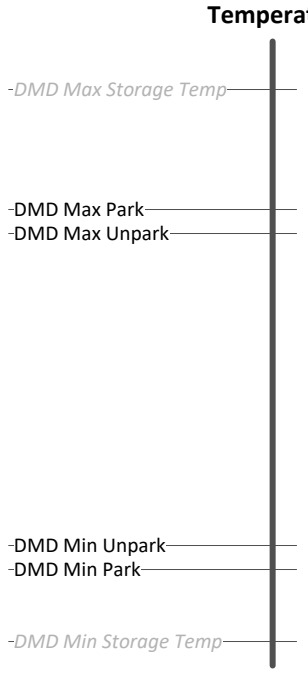


Figure 5-26. Temperature Actions

5.6.2 PWM Temperature Management Function

Some applications have side effects at higher temperatures and require active cooling solutions such as fans. For systems that use a variable-speed fan controlled by a PWM signal, the DLPC230-Q1 offers a PWM Temperature Management Function. It is intended to self-monitor a system temperature level and automatically update the fan speed via PWM duty cycle. This implementation avoids over-design and reduces noise level of a fixed-speed fan. This function also supports hysteresis to avoid switching back and forth between temperature regions and duty cycle levels. This hysteresis is applied only as the temperature decreases. A 2-Dimensional LUT stored inside DLPC230 flash dictates transition temperatures and PWM duty cycle for this automated function. These parameters, which include enable/disable, temperature thresholds, PWM Duty Cycle percent, PWM GPIO pin, and hysteresis, must be specified in the Automotive DLP Composer(TM) tool. Figure 5-27 shows an example use case. See PWM temperature management commands 35h-39h, starting with Section 8.2.31.

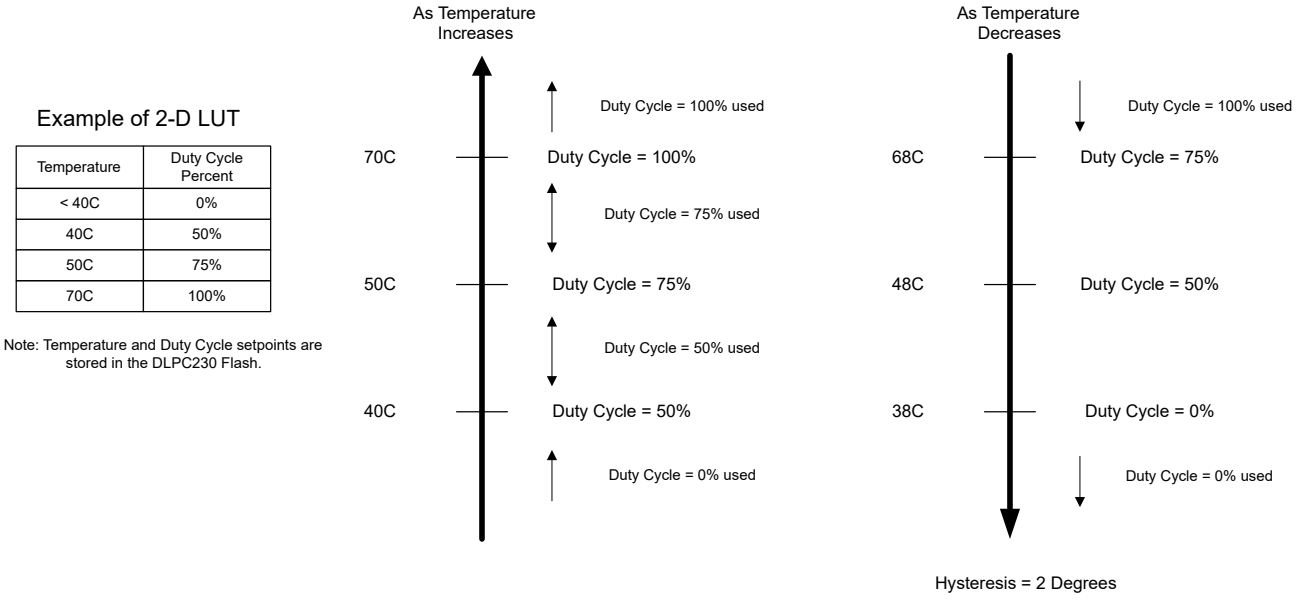


Figure 5-27. PWM Duty Cycle Temperature Management Example

5.7 ADC Measurements

The TPS99000-Q1 includes an analog to digital converter block with a 32:1 input mux. The DLPC230-Q1 controls the timing of ADC measurements from this block using a dedicated high-speed serial control interface in order to capture up to 63 sequence-aligned samples per frame while an image is being displayed. From the command interface, there are two ADC read commands. The usage of these two commands is dependent on whether an image is being displayed or not:

- *ADC Measurements - Read* - Used while an image is displayed. This allows access to the last frame of sequence-aligned ADC measurement capture results. Measurement capture timing is not decided by the command itself, as the results were already captured during a previous video frame.
- *ADC Single Measurement - Read* - Used in Standby mode. This commands the DLPC230-Q1 main application software to read a single ADC measurement after the command has been received and return the result.

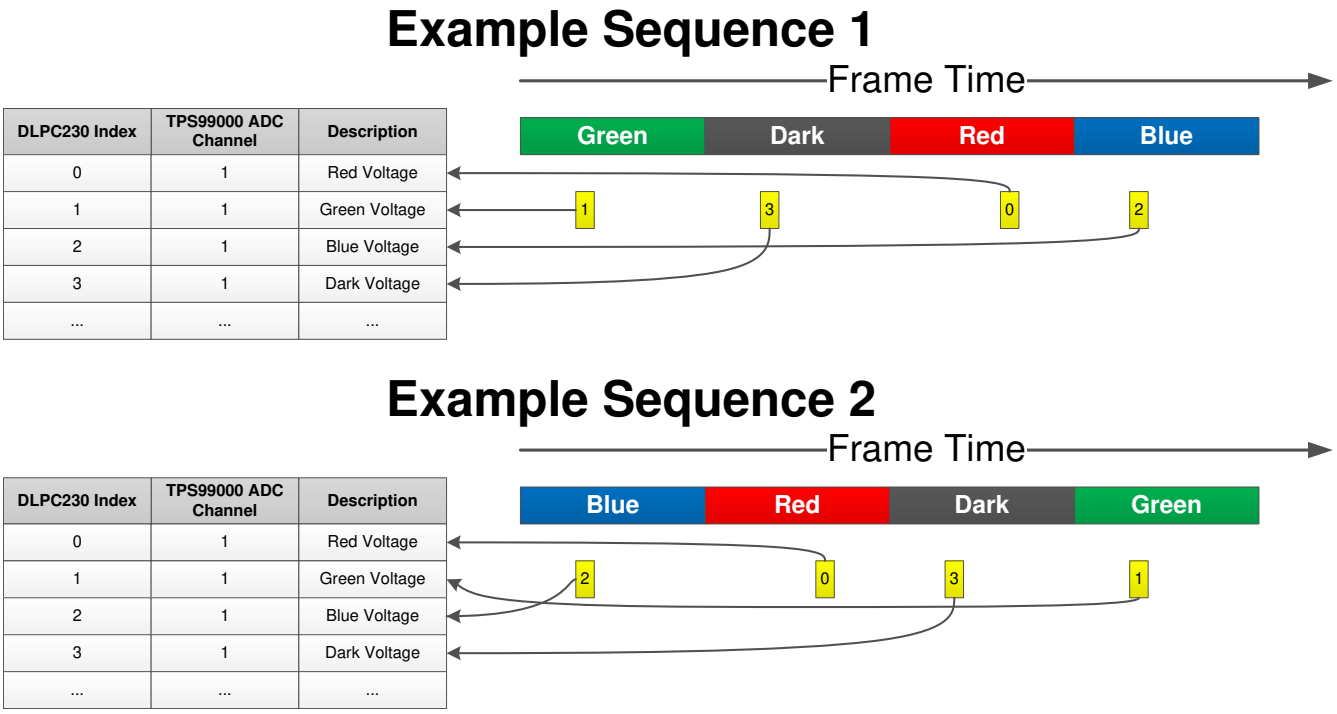
5.7.1 Sequence-Aligned ADC Measurements

The DLPC230-Q1 continuously commands the TPS99000-Q1 to capture specific ADC measurements throughout each video frame. These measurements are aligned to events in the DMD sequence, such as illumination enables or dark time. DLPC230-Q1 hardware stores the ADC measurements in buffered memory. The memory buffer is swapped each video frame so that DLPC230-Q1 software is continuously reading the previous frame's ADC measurement data.

The order in which each ADC measurement is captured is dependent on the order of events in a given sequence. Figure 5-28 demonstrates two example sequences with sequence-aligned ADC measurements to capture various illumination voltage levels. The same measurements are captured during each sequence and the results are placed in the same DLPC230-Q1 memory indexes, but the order in which the measurements are retrieved by the DLPC230-Q1 varies to accomodate the sequence timing.

The command parameters for the *ADC Measurements* command specify the desired measurement index range in DLPC230-Q1 memory from 0 to 63.

The flash header file defines the DLPC230-Q1 indexes and their associated TPS99000-Q1 ADC channels for a specific flash build.



DLPC230 Index	TPS99000 ADC Channel	Description
0	1	Red Voltage
1	1	Green Voltage
2	1	Blue Voltage
3	1	Dark Voltage
...	...	...

DLPC230 Index	TPS99000 ADC Channel	Description
0	1	Red Voltage
1	1	Green Voltage
2	1	Blue Voltage
3	1	Dark Voltage
...	...	...

Figure 5-28. Sequence-Aligned ADC Example

5.7.2 Single ADC Measurements

When the *ADC Single Measurement* command is received by the DLPC230-Q1 main application, it will request an ADC capture from the TPS99000-Q1 for the specific ADC multiplexer index that is specified by the command, from 0 to 31. Note that this index is different than the DLPC230-Q1 memory index which is not relevant for a single ADC measurement.

Chapter 6

Tests and Diagnostics

6.1 Overview

The DLP553X-Q1 chipset includes numerous test and diagnostic features to verify proper system operation such as video interface tests, communication interface tests, and self-tests. The following sections describe available features.

6.2 Emergency Shutdown

The DLPC230-Q1 main application performs an emergency shutdown when a serious system error occurs such as losing communication with a chipset signal interface. The specific errors that caused the shutdown can be retrieved from the *Error History* command unless the error has caused a system reset. If the error has caused a system reset, the *System Information* command can be used to read the cause of the last reset. If the error has not caused a system reset, other errors may occur as a side-effect of the failure mechanism that caused the emergency shutdown. These side-effect errors will also be included in the error history if possible.

The actions that are taken by the DLPC230-Q1 main application for an emergency shutdown are:

- LEDs disabled
- DMD parked and powered down
- Main application mode transitions to Standby Mode
- Errors captured in *Error History*
- *Emergency Shutdown* bit is set in the *Short Status*
- *HOST_IRQ* signal is set to HIGH state

To recover from an emergency shutdown, the system will require a full power cycle using the PROJ_ON signal to the TPS99000-Q1. Details on PROJ_ON signal timing requirements can be found in the DLPC230-Q1 Data Sheet and the TPS99000-Q1 Data Sheet. The host should attempt to obtain the *Error History* from the DLPC230-Q1 prior to this full reset.

Section 6.4 defines tests performed and notes those that may result in an emergency shutdown. During operation, the *Short Status* should be checked periodically for any errors including those that cause emergency shutdown.

6.2.1 Emergency Shutdown Causes

Table 6-1. Emergency Shutdown Causes

INDEX	DESCRIPTION
1	Front End BIST failed
2	Back End BIST failed
3	DMD Memory BIST failed
4	TPS99000-Q1 Interface BIST failed
5	DLPC230-Q1 Memory BIST failed
6	DMD clock out of range
7	Video Tell-Tale Checksum failed
8	TPS99000-Q1 Watchdog 1 error
9	TPS99000-Q1 Watchdog 2 error
10	TPS99000-Q1 Clock Ratio Monitor error

Table 6-1. Emergency Shutdown Causes (continued)

INDEX	DESCRIPTION
11	Frame Memory Buffer Swap Watchdog error
12	Sequencer Instruction Watchdog error
13	DMD Reset Instruction Watchdog error
14	TPS99000-Q1 Register Checksum error
16	TPS99000-Q1 SPI Interface Parity error
17	TPS99000-Q1 ADC Interface Parity error
18	Multi-bit ECC error in DLPC230-Q1 memory
19	Multi-bit ECC error in DLPC230-Q1 memory
20	Temperature sensor remote channel open
21	Temperature sensor read errors
22	Average Picture Level error
23	DMD voltage out of range

6.3 Diagnostic Memory Interface

The diagnostic interface provides access to a 64-byte memory that is updated by the DLPC230-Q1 main application during operation. This diagnostic memory information is intended to summarize operation and to allow the monitoring host to independently check operation of the system and the primary host controller.

The diagnostic memory is accessed through a separate port that is independent of the host command interface. The communication protocol for this interface is either I^2C or SPI, and will always be the protocol that is not selected for use by the host command interface. The protocol selection is determined by the state of the *HOST_IF_SEL* hardware signal at power-up. Communication requests to the diagnostic interface to access and clear the memory contents are handled entirely by DLPC230-Q1 hardware. The values set in the memory are updated by the DLPC230-Q1 main application software. The diagnostic interface commands are described in *Chapter 9*. Read and write protocols are identical to the host command interface.

The following commands are used for the diagnostic memory interface:

- *Diagnostic Interface Status - Read* - Read desired bytes from the diagnostic memory.
- *Diagnostic Interface Status Clear - Write* - Clear desired bytes in the diagnostic memory to 0. Any bytes may be cleared, but it is only critical to clear values if they have a risk of reaching a maximum value, such as a counter. All values will be updated at a specified rate, but clearing a value may also provide more confidence that the value is being updated in case the value does not change after a significant time interval.

The diagnostic memory contents may also be read through the host command interface by requesting the data from the DLPC230-Q1 main application software. Note that the host command interface has no command to clear the diagnostic memory. This memory may only be cleared through the dedicated diagnostic interface.

- *Diagnostic Interface Status - Read* - Read desired bytes from the diagnostic memory.

Table 6-2 defines the diagnostic memory content.

Table 6-2. Diagnostic Memory Definition

BYTE	DESCRIPTION	UPDATE RATE	SHOULD BE CLEARED
0	**Operating Mode** 0x0: Reserved for clear 0x1: Main application - Standby 0x2: Main application - Display	Mode change	No
2:1	**Software Activity Counter** Incrementing counter to confirm that the main application is still active. 0-65535, rolls over to 0.	Approximately 16ms	No
3	**TPS99000-Q1 State** 0 - 7: Off 8: Standby 9 - 11: Powering DMD 12: Display Ready 13: Display On 14 - 17: Parking	State change	No
4	**Last Reset Cause** 0x0: Power Cycle 0x1: PROJ_ON 0x2: TPS99000-Q1 Watchdog Software Error 0x3: TPS99000-Q1 Watchdog Sequence Error 0x4: TPS99000-Q1 Die Temperature Exceeded 0x5: Software commanded power cycle 0x7: Host commanded reset 0x8: Software commanded DLPC230-Q1-only reset 0x9 - 0xFF: *Reserved*	Software initialization	No
10:5	***Reserved***		

Table 6-2. Diagnostic Memory Definition (continued)

BYTE	DESCRIPTION	UPDATE RATE	SHOULD BE CLEARED
11	**Error History Count** Number of errors received since the error history was last cleared through the host command interface. Note that clearing this diagnostic byte will not reset the count. Clearing this diagnostic byte will only set the value to 0 until the value is re-written by the main application.	On error	No
13:12	**Last Error Code** Latest error code that has occurred. Refer to *Section A.2* for error code descriptions. **Bits 15:12** - Unused **Bits 11:0** - Error code.	On error	No
14	**ECC Error Count** Number of single-bit or multi-bit ECC errors received since the last system reset.	On error	No
15	**Emergency Shutdown** 0x0: No emergency shutdown has occurred 0x1: Emergency shutdown has occurred	On error	No
19:16	***Reserved***		
23:20	**External Video Checksum Result** Last calculated external video checksum.	Per frame	No
24	**Average Picture Level Result** Last calculated average picture level.	Per frame	No
25	**Source Type** **Bit 3** • 0x1: External source Lost **Bits 2:0** • 0x0: Not Applicable (Standby mode or source transition) • 0x1: External Video Parallel • 0x2: External Video OpenLDI • 0x3: Test Pattern • 0x4: Splash	Source change	No
27:26	**External Source - Total Pixels Per Line** Set to 0 while splash or test pattern are in use.	Source change	No
29:28	**External Source - Total Lines Per Frame** Set to 0 while splash or test pattern are in use.	Source change	No
31:30	**External Source - Active Pixels Per Line** Set to 0 while splash or test pattern are in use.	Source change	No
33:32	**External Source - Active Lines Per Frame** Set to 0 while splash or test pattern are in use.	Source change	No
35:34	**Sequence Maximum Temperature** Maximum sequence temperature that is currently being enforced. **Bits 15:8** - Two's complement integer portion of temperature in Celsius **Bits 7:4** - 1/16 Celsius steps always added to the integer portion. Note that this fraction is still added even if the integer portion is negative. **Bits 3:0** - Unused	Per frame	No

Table 6-2. Diagnostic Memory Definition (continued)

BYTE	DESCRIPTION	UPDATE RATE	SHOULD BE CLEARED
37:36	***Reserved***		
39:38	**DMD Filtered Temperature** **Bits 15:8** - Two's complement integer portion of temperature in Celsius **Bits 7:4** - 1/16 Celsius steps always added to the integer portion. Note that this fraction is still added even if the integer portion is negative. **Bits 3:0** - Unused	Temperature read Approximately 125ms	No
41:40	**TMP411 Local Filtered Temperature** **Bits 15:8** - Two's complement integer portion of temperature in Celsius **Bits 7:4** - 1/16 Celsius steps always added to the integer portion. Note that this fraction is still added even if the integer portion is negative. **Bits 3:0** - Unused	Temperature read Approximately 125ms	No
43:42	**DMD VBIAS ADC Value** **Bits 15:0** - Two's complement integer format. 1mV scale.	Per frame	No
45:44	**DMD VRESET ADC Value** **Bits 15:0** - Two's complement integer format. 1mV scale.	Per frame	No
47:46	**DMD VOFFSET ADC Value** **Bits 15:0** - Two's complement integer format. 1mV scale.	Per frame	No
53:48	***Reserved***		
55:54	***Reserved***		
64:56	***Reserved***		

6.4 Test Descriptions

The DLP553X-Q1 chipset includes diagnostic features to aid in the detection of potential failures. These features are grouped into two categories:

- Non-Periodic - Run while the system is not displaying an image, typically at power-up or power-down.
- Periodic - Execute continuously while an image is being displayed.

6.4.1 Periodic Tests

These tests can be executed continuously in certain modes. These tests do not corrupt or interfere with normal operation of the system. Some tests may be controlled by the host, but others will run continuously as functions are in use. For example, the frame memory buffer swap watchdog will always be enabled while the frame memory is being used.

Figure 6-1 shows a block diagram with labels on DLPC230-Q1 internal blocks and various system components. These labels are used in Table 6-3 to identify the coverage of each test.

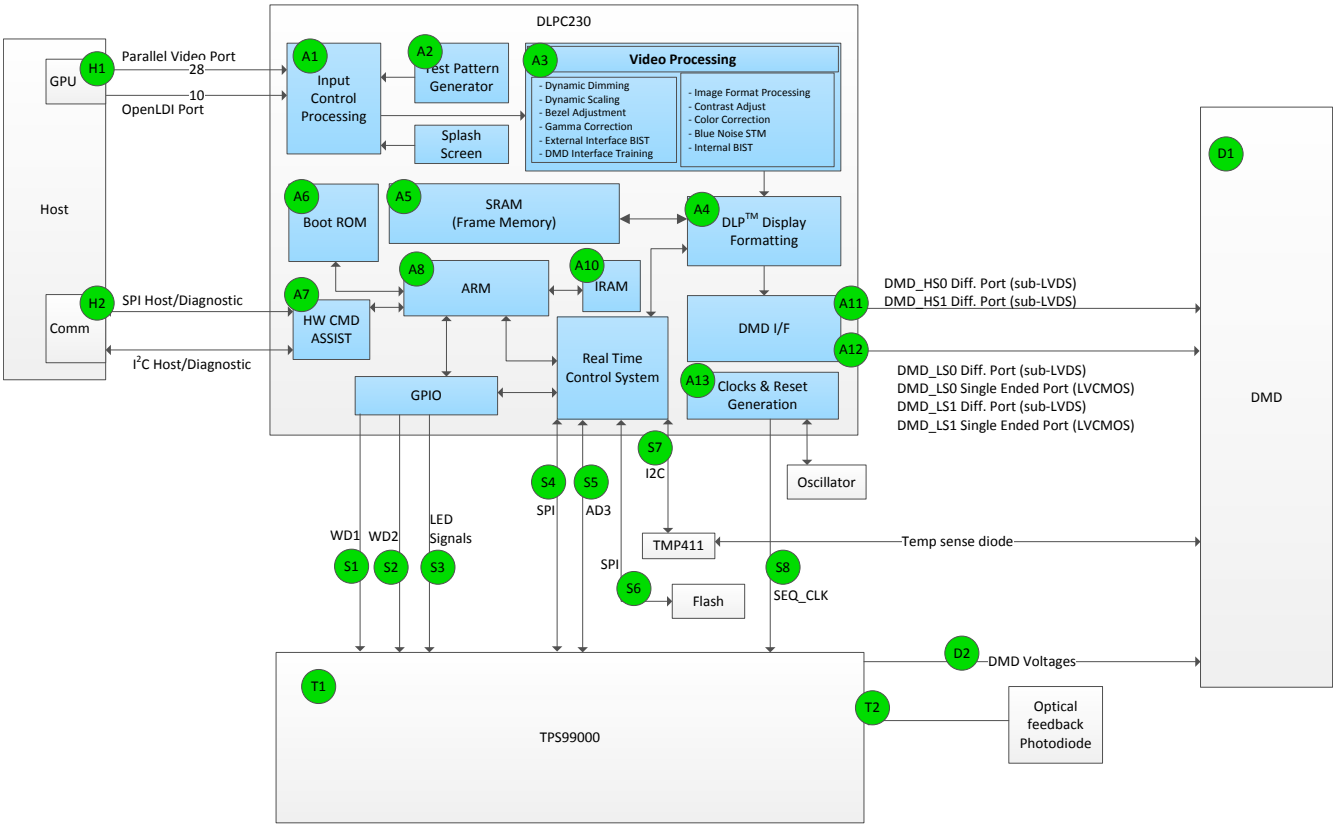


Figure 6-1. Test Coverage System Block Diagram

Table 6-3. Periodic Tests Overview

TEST	COVERAGE	HOST-CONTROLLED	FAIL ACTION	SECTION
Video Source Loss	H1, A1, A3	Continuously run	Alternate Source	Section 6.4.1.1
Video Tell-Tale Checksum	H1, A1, A3	Yes	Log error / Alternate Source / Emergency Shutdown	Section 6.4.1.2
Video Frame Counter Checksum	H1, A1, A3	Yes	Log error / Alternate Source / Emergency Shutdown	Section 6.4.1.2
Average Picture Level	H1, A1, A3	Yes	Log Error / Emergency Shutdown	Section 6.4.1.4
Loss of Ping Command	H2	Yes	Alternate Source	Section 6.4.1.5
DLPC230-Q1 Processor Memory ECC	A10	Continuously run	Emergency Shutdown	Section 6.4.1.6
Flash Table Transport CRC	S6	Continuously run	Log Error	Section 6.4.1.7
Frame Buffer Swap Watchdog	A4, A5	Continuously run	Emergency Shutdown	Section 6.4.1.8
Sequencer Instruction Read Watchdog	A4	Continuously run	Emergency Shutdown	Section 6.4.1.9
DMD Reset Instruction Watchdog	A4, A11	Continuously run	Emergency Shutdown	Section 6.4.1.10
DLPC230-Q1 System Voltage Monitor	Voltage Rails, S5	Continuously run	Log Error	Section 6.4.1.11
DLPC230-Q1 DMD Voltage Monitor	D2, S5	Continuously run	Emergency Shutdown	Section 6.4.1.12
DLPC230-Q1 TPS99000-Q1 Bandgap Monitor	T1, S5	Continuously run	Log Error	Section 6.4.1.13
DMD Temperature Monitor	D1	Continuously run	Standby mode	Section 6.4.1.14
DMD Clock Monitor	A13	Continuously run	Emergency Shutdown	Section 6.4.1.15
DMD High Speed Interface Training	A11, A12, D1	Continuously run	Log Error	Section 6.4.1.16
DMD Low Speed Interface Test	A12, D1	Continuously run	Log Error	Section 6.4.1.17
TPS99000-Q1 DLPC230-Q1 Processor Watchdog (WD1)	A8, S1	Continuously run	System Reset	Section 6.4.1.18
TPS99000-Q1 DLPC230-Q1 Sequencer Watchdog (WD2)	A4, S2	Continuously run	Attempt to correct Emergency Shutdown	Section 6.4.1.19
TPS99000-Q1 Temperature Warning / Error	T1	Continuously run	Log Error System Reset	Section 6.4.1.20
TPS99000-Q1 Clock Ratio Monitor	A13, S8	Continuously run	Log Error	Section 6.4.1.21
TPS99000-Q1 Register Password Lock	T1	Continuously run		Section 6.4.1.22
TPS99000-Q1 Register Checksum	T1	Continuously run	Attempt to correct Emergency Shutdown	Section 6.4.1.23
Software Monitor Thread	A8	Continuously run	Log Error	Section 6.4.1.24

6.4.1.1 Video Source Loss Detection

While external video is being displayed, this test monitors the external source VSYNC, pixel clock, active lines per frame, and active pixels per line every frame to verify that the source signals are stable.

The test will take action if:

- VSYNC is out of range
- Pixel clock is out of range
- Active lines per frame does not match the expected value
- Active pixels per line does not match the expected value

6.4.1.1.1 Configuration

The following configuration options are set in flash. The value of these parameters can be found in the flash header file that is provided along with the flash binary file.

Table 6-4. Video Source Loss Parameters

PARAMETER	DESCRIPTION
Alternate Source On Failure	Any test pattern or splash image to be displayed upon source loss.
Auto-recover	0x0: Once the test fails, alternate source will continue to display even if the external source returns to a valid range. Host action is required to change to external source after failure. 0x1: If the test passes after failing, the system will return to displaying the external source. This does not require host action.
Pixel Clock maximum, minimum	Maximum and minimum allowable pixel clock frequency. This is generally set to ±2% of the nominal pixel clock.
VSYNC maximum, minimum	Maximum and minimum allowable VSYNC frequency. This is set to the VSYNC range specified in the DLPC230-Q1 data sheet.
Active Lines Per Frame	Vertical resolution of the external source.
Active Pixels Per Line	Horizontal resolution of the external source.

6.4.1.1.2 Execution

This test is always executed when an external source is in use.

6.4.1.1.3 Failure Actions

If the source was initially in-range, a failure will trigger the display of an alternate internal video source. This can be configured as any test pattern or splash image stored in flash. If the source is initially out-of-range during configuration or during a transition to display, no image will be displayed.

Table 6-5 describes various source scenarios and the action that will be taken.

Table 6-5. Video Source Loss Failures

CONFIGURATION	EVENT	ACTION
Source valid	Standby to Display transition	Display external source
Source invalid	Standby to Display transition	Stay in standby Log Error
Source valid	Source select command	Display external source
Source invalid	Source select command	Illumination disabled Log error
Source valid	Source valid to invalid	Display alternate image Log error
Source invalid	Source invalid to valid	Display external source IF auto-recover

6.4.1.1.4 Error Codes

The following error codes indicate a failure was detected:

Table 6-6. Video Source Loss Detection Error Codes

ERROR CODE	DESCRIPTION
665	External source measured active lines per frame does not match source definition.
666	External source measured active pixels per line does not match source definition.
671	External source Open LDI port pixel clock frequency is not within the allowable range.
674	External source parallel port pixel clock frequency is not within the allowable range.
676	External source pixel clock above the allowable range.
677	External source pixel clock below the allowable range.
678	External source VSYNC frequency greater than defined maximum.
679	External source VSYNC frequency less than defined minimum.
913	External source has been lost.
914	External source has been lost and attempt to transition to alternate source failed.
946	Expected VSYNC signal did not arrive in the time allocated.

6.4.1.2 Video Tell-Tale Checksum

This test performs a checksum of the pixels within a specified region of video pixel data that is expected to remain static across multiple frames. The host provides an expected checksum for the specified pixel region. DLPC230-Q1 hardware calculates the checksum of the specified region every frame and compares this to the host-specified checksum. If a mismatch is detected between the expected and calculated checksum after any video frame, the test will take action.

6.4.1.2.1 Configuration

External Video Checksum Settings command is used to specify the desired window of video pixels and the expected checksum. This command can also be read to confirm the actual checksum of the current video data.

Figure 6-2 demonstrates the checksum region configuration.

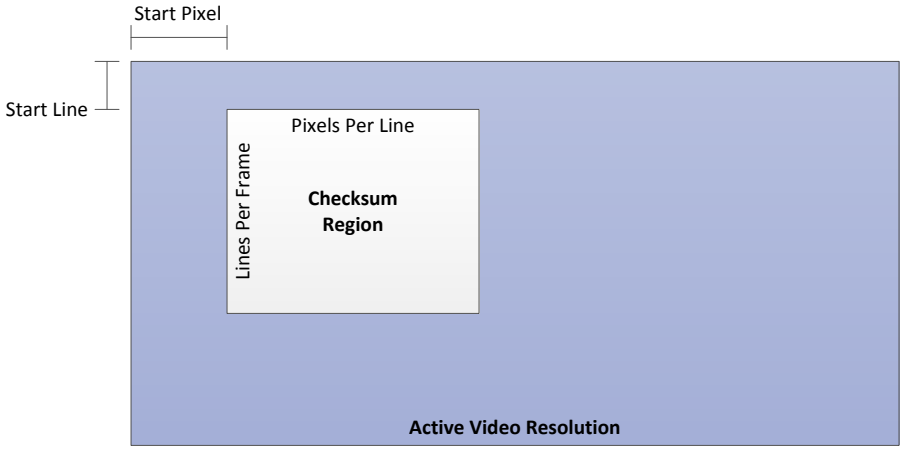


Figure 6-2. Video Tell-Tale Checksum Configuration

The following equation is used to calculate the checksum. For each pixel within the specified window, sum the red, green, and blue level and add it to the checksum. If the value exceeds 32-bits (0xFFFFFFFF), the checksum rolls over to 0.

CS(n) = (CS(n-1) + R(n) + G(n) + B(n)) & 0xFFFFFFFF

Where:

- "n" is the current pixel that is being added to the checksum
- "CS(n)" is the current checksum
- "CS(n-1)" is the previous checksum. The initial value is 0.
- "R(n)", "G(n)", "B(n)" are the red, green, and blue components of the current pixel, each with a value of 0 to 255.

Settings should only be modified while the test is disabled. If changes need to be made to the settings, the test should be disabled and then re-enabled after settings have been applied.

6.4.1.2.2 Execution

External Video Checksum Control command is used to enable or disable the test. Note that video tell-tale and video frame counter cannot be executed at the same time.

6.4.1.2.3 Failure Actions

The failure action is flash-configurable. Software may log an error, switch to the alternate display source, or execute an emergency shutdown.

6.4.1.2.4 Error Codes

The following error codes indicate a failure was detected:

Table 6-7. Video Tell-Tale Checksum Error Codes

ERROR CODE	DESCRIPTION
238	Command error: Attempted to change external video checksum settings, but external video checksum is enabled.
535	External video checksum failed due to checksum mismatch.

6.4.1.3 Video Frame Counter Checksum

This test utilizes the video tell-tale checksum hardware to confirm that the video frame is being updated by the input graphics processor each frame. It cannot be executed at the same time as the video tell-tale checksum since they rely on the same hardware.

A pixel region is specified by host command and the checksum of the pixel data within that region must increment by 1 each frame until the value reaches its maximum value. Once the checksum has reached its maximum value, the value must roll over to 0 and the pattern repeats. If the value does not increment in a frame, the test will take action. The maximum and minimum frame count values are flash settings.

The following equation is used to calculate the checksum. For each pixel within the specified window, sum the red, green, and blue level and add it to the checksum.

CS(n) = (CS(n-1) + R(n) + G(n) + B(n)) & 0xFFFFFFFF

Where:

- "n" is the current pixel that is being added to the checksum
- "CS(n)" is the current checksum
- "CS(n-1)" is the previous checksum. The initial value is 0.
- "R(n)", "G(n)", "B(n)" are the red, green, and blue components of the current pixel, each with a value of 0 to 255.

There are several possible implementations of this frame counter that can be used by a graphics processor to fulfill this test's requirements.

6.4.1.3.1 One-Pixel Frame Counter

This implementation uses one pixel in a corner of the active video. The red and green values of this pixel are set to 0, and the blue value is incremented from 0 to 7 each frame. This pixel remains effectively black across all frames due to the low maximum pixel value (R 0, G 0, B 7). The value of the frame counter checksum is equal to the blue value of this single pixel since it is the only pixel within the checksum region and its red and green values are 0.

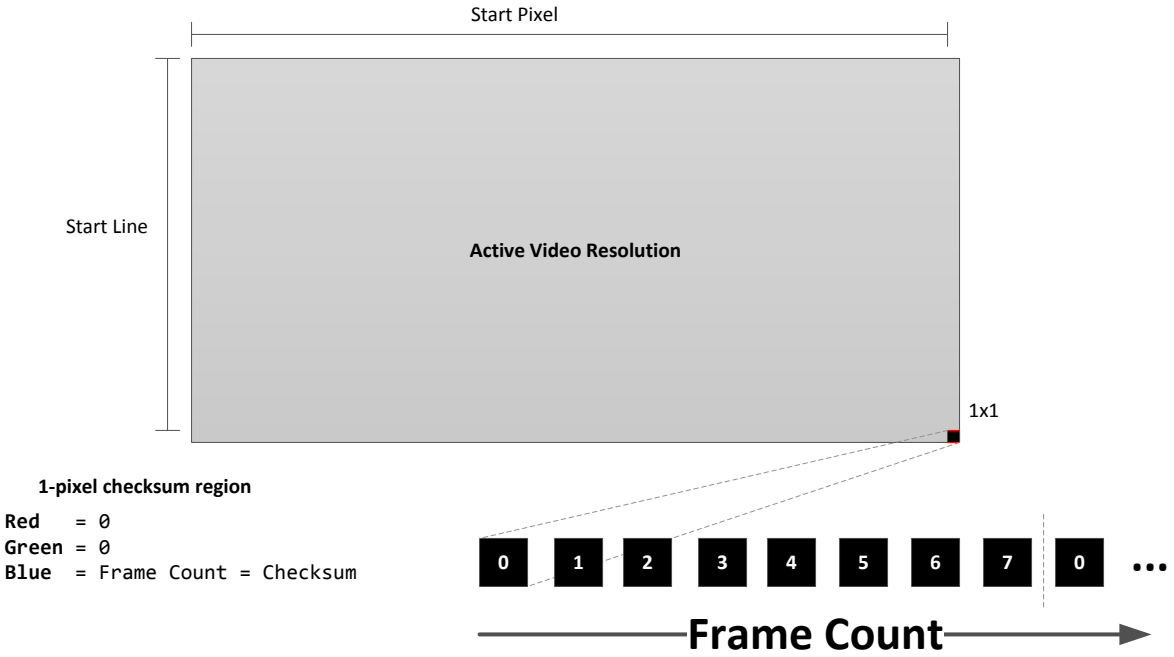


Figure 6-3. One-Pixel Frame Counter

6.4.1.3.2 Seven-Pixel Frame Counter

This implementation uses seven pixels in the corner of the active video. The red and green values of these pixels are set to 0, and the blue values are set to 0 or 1 each frame. Each frame an additional blue pixel within the region is set to 1 to increment the checksum from 0 to 7. The advantage of this implementation is that each pixel's maximum value is (R 0, G 0, B 1). The disadvantage is that it requires a larger checksum region where these pixels will effectively appear black and cannot be used for other video content.

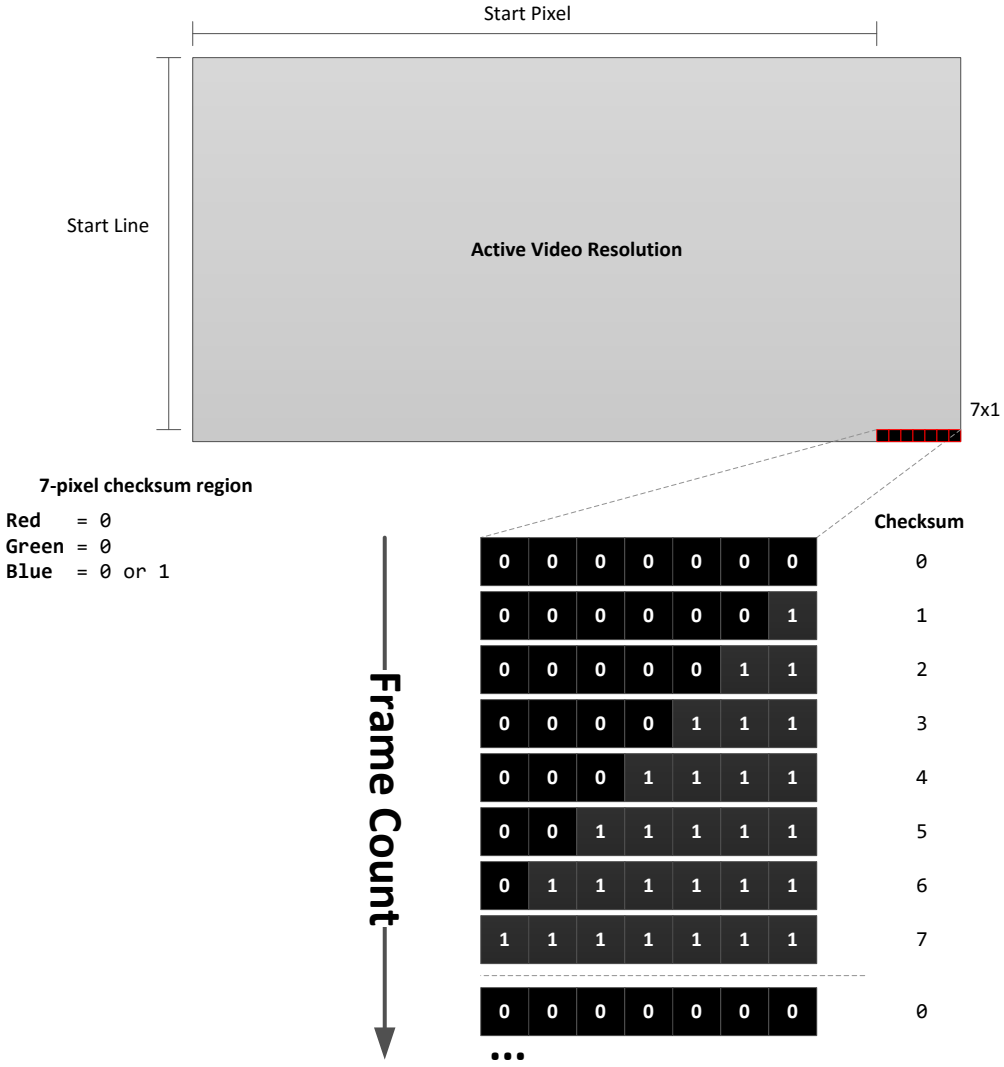


Figure 6-4. Seven-Pixel Frame Counter

6.4.1.3.3 Configuration

External Video Checksum Settings command is used to specify the desired window of video pixels. This command can also be read to confirm the actual checksum of the current video data.

Figure 6-5 demonstrates the checksum region configuration.

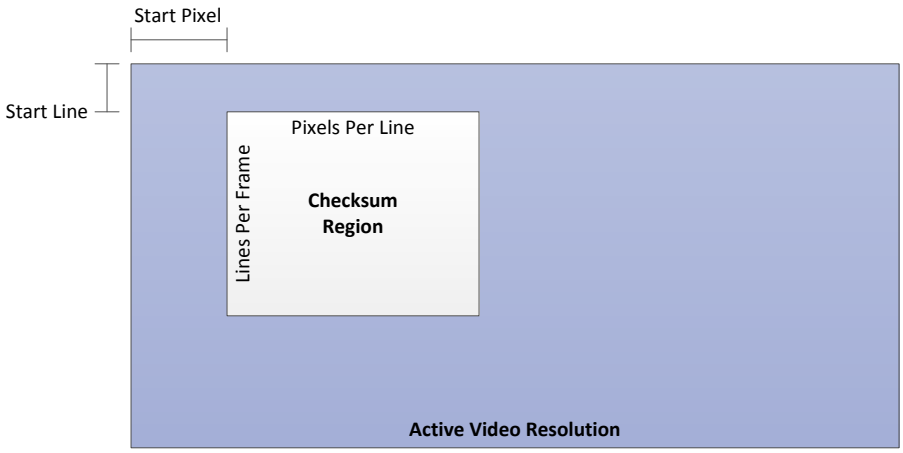


Figure 6-5. Video Frame Counter Checksum Configuration

Settings should only be modified while the test is disabled. If changes need to be made to the settings, the test should be disabled and then re-enabled after settings have been applied.

6.4.1.3.4 Execution

External Video Checksum Control command is used to enable or disable the test. Note that video tell-tale and video frame counter cannot be executed at the same time.

6.4.1.3.5 Failure Actions

The failure action is flash-configurable. Software may log an error, switch to the alternate display source, or execute an emergency shutdown.

6.4.1.3.6 Error Codes

The following error codes indicate a failure was detected:

Table 6-8. Video Frame Counter Checksum Error Codes

ERROR CODE	DESCRIPTION
558	External video frame count out of sequence.
760	Video frame counter failed to lock due to value out of range.

6.4.1.4 Average Picture Level

This test checks the average level of the incoming video pixel data. The host specifies an expected maximum average level for the incoming data. The main application will compare this host-specified maximum to the actual level each frame. If the actual picture level exceeds the host-specified maximum, the test will take action.

6.4.1.4.1 Configuration

Average Picture Level Control command is used to configure this test.

The following procedure is used to calculate the average picture level for all active pixels in a video frame.

- Pixel_Brightness = Round((Red + Green + Blue) / 3)
  - Where Red, Green, Blue are 8-bit values from 0-255
- Line_Brightness = Round(Sum(Pixel_Brightness[0:Horizontal_Resolution]) / Horizontal_Resolution)
- Picture_Brightness = Round (Sum(Line_Brightness[0:Vertical_Resolution]) / Vertical_Resolution)

The resulting Picture_Brightness should be an 8-bit value from 0 to 255.

6.4.1.4.2 Execution

Average Picture Level Control command is used to enable or disable this test.

6.4.1.4.3 Failure Actions

This test can be configured to take two actions upon threshold exeeded:

- Log an error, but continue to display external source
- Execute emergency shutdown

6.4.1.4.4 Error Codes

The following error codes indicate a failure was detected:

Table 6-9. Average Picture Level Error Codes

ERROR CODE	DESCRIPTION
915	Average Picture Level exceeded its specified limit.

6.4.1.5 Loss of Ping Command

This test monitors for ping commands within a specified number of milliseconds. Each time a command is received the timer resets to zero. If the timer reaches the specified maximum time, the test will take action. This acts as a watchdog timeout on host communication.

6.4.1.5.1 Configuration

Loss of Ping Control command is used to specify a time period in which each command must be sent and enable or disable this test.

The time window can also be specified to a default value in flash.

6.4.1.5.2 Execution

Loss of Ping Control command is also used to enable or disable this test. The default enable or disable state can also be specified in flash.

Headlight Ping - Write command is used to periodically satisfy the timeout condition.

6.4.1.5.3 Failure Actions

An alternate source will be displayed on failure. This alternate source is the same source that *Video Source Loss Detection* uses on failure.

6.4.1.5.4 Error Codes

The following error codes indicate a failure was detected:

Table 6-10. Loss of Ping Error Codes

ERROR CODE	DESCRIPTION
222	Loss of ping BIST failed. No dimming / ping command was received within the configured time frame.

6.4.1.6 DLPC230-Q1 Processor Memory ECC

The DLPC230-Q1 ARM processor's memory includes ECC that hardware uses to detect multi-bit errors and correct 1-bit errors. If a multi-bit error occurs, the main application will attempt to take action.

6.4.1.6.1 Configuration

No configuration is required.

6.4.1.6.2 Execution

ECC detection is always run during operation.

6.4.1.6.3 Failure Actions

The main application will attempt to perform an emergency shutdown.

6.4.1.6.4 Error Codes

The following error codes indicate a failure was detected:

Table 6-11. DLPC230-Q1 Processor Memory ECC Error Codes

ERROR CODE	DESCRIPTION
954	Multi-bit ECC memory error detected.
955	Single-bit ECC memory error detected.

6.4.1.7 Flash Table Transport CRC

Look-up tables, such as sequences, are CRC-verified while being transferred from flash. Once the table is written to internal memory, hardware calculates a CRC of the data and compares it to the CRC value that was calculated while the table was being read from flash. If the CRC is incorrect, this test will take action.

6.4.1.7.1 Configuration

No configuration is required.

6.4.1.7.2 Execution

This test is always executed during table data transport.

6.4.1.7.3 Failure Actions

The table will be reloaded and the main application will log the error.

6.4.1.7.4 Error Codes

The following error codes indicate a failure was detected:

Table 6-12. Flash Table Transport CRC Error Codes

ERROR CODE	DESCRIPTION
526	Flash Table Transport CRC failed. CRC error while transferring the table from flash.
653	Flash Table Type 1 CRC failed.
658	Flash Table Type 5 memory access failed.
659	Flash Table Type 5 CRC failed.
660	Flash Table Type 6 CRC failed.
661	Flash Table Type 7 CRC failed.
663	Flash Table Type 8 CRC failed.
664	Flash Table Type 9 access failed.

6.4.1.8 Frame Buffer Swap Watchdog

The DLPC230-Q1 video frame memory is double-buffered. The main application configures a watchdog timer for the buffer swap and each buffer swap event resets the timer. If the timer expires, the buffers did not swap as expected and the main application takes action. The timer is configured to be approximately 9 frame times in length.

6.4.1.8.1 Configuration

No configuration is required.

6.4.1.8.2 Execution

This test is always executed while displaying external source and test pattern. The test is not executed while displaying splash images because of the longer buffer swap time required for splash images.

6.4.1.8.3 Failure Actions

Emergency shutdown will be performed on failure.

6.4.1.8.4 Error Codes

The following error codes indicate a failure was detected:

Table 6-13. Frame Buffer Swap Watchdog Error Codes

ERROR CODE	DESCRIPTION
969	Frame memory buffer swap watchdog error.

6.4.1.9 Sequencer Instruction Read Watchdog

This test monitors video sequencer instruction processing. The main application sets up a hardware timer and each sequencer instructions will reset this timer. The timer is configured to be approximately 9 frame times in length.

6.4.1.9.1 Configuration

No configuration is required.

6.4.1.9.2 Execution

This test is always executed while displaying an image.

6.4.1.9.3 Failure Actions

Emergency shutdown will be performed on failure.

6.4.1.9.4 Error Codes

The following error codes indicate a failure was detected:

Table 6-14. Sequencer Instruction Read Watchdog Error Codes

ERROR CODE	DESCRIPTION
968	Sequencer instruction watchdog error.

6.4.1.10 DMD Reset Instruction Watchdog

This test monitors for DMD bias and reset instructions sent from the DLPC230-Q1 to the DMD while displaying an image. Failure indicates that the DMD mirrors may not be moving as intended since no reset instructions are being sent.

6.4.1.10.1 Configuration

No configuration is required.

6.4.1.10.2 Execution

This test is always executed while displaying an image.

6.4.1.10.3 Failure Actions

Emergency shutdown will be performed on failure.

6.4.1.10.4 Error Codes

The following error codes indicate a failure was detected:

Table 6-15. Sequencer Instruction Read Watchdog Error Codes

ERROR CODE	DESCRIPTION
967	DMD reset instruction watchdog error.

6.4.1.11 DLPC230-Q1 System Voltage Monitor

This test monitors system voltage levels each frame using TPS99000-Q1 ADC measurements. The following voltages are monitored:

- P1P1V
- P1P8V
- P3P3V
- VMAIN
- DVDD
- LDOT_M8
- ADC_VREF
- DRVR_PWR

If any of these voltages are measured outside of a specified maximum and minimum value, the test will take action. The monitor ranges are specified in the flash header file provided along with the flash binary file. The true voltage operating ranges are specified in the DLPC230-Q1 Data Sheet and TPS99000-Q1 Data Sheet. The monitor ranges used for this test are set wider than the true voltage operating ranges to account for measurement tolerances.

6.4.1.11.1 Configuration

No configuration is required.

6.4.1.11.2 Execution

This test is always executed during display mode.

6.4.1.11.3 Failure Actions

An error will be logged on failure.

6.4.1.11.4 Error Codes

The following error codes indicate a failure was detected:

Table 6-16. DLPC230-Q1 System Voltage Monitor Error Codes

ERROR CODE	DESCRIPTION
843	System 1.1V rail out of range.
844	System 1.8V rail out of range.
845	System 3.3V rail out of range.
846	ADC external VREF voltage out of range.
847	DVDD voltage out of range.
848	LDOT_M8 voltage out of range.
849	VMAIN voltage out of range.
1015	Driver power voltage out of range.

6.4.1.12 DLPC230-Q1 DMD Voltage Monitor

This test monitors the DMD reset voltage levels each frame using TPS99000-Q1 ADC measurements. The following voltages are monitored:

- VBIAS
- VRESET
- VOFFSET

If any of these voltages are measured outside of a specified maximum and minimum value, the test will take action. The allowable ranges are specified in the flash header file provided along with the flash binary file.

6.4.1.12.1 Configuration

No configuration is required.

6.4.1.12.2 Execution

This test is always executed during display mode.

6.4.1.12.3 Failure Actions

Emergency shutdown will be performed on failure.

6.4.1.12.4 Error Codes

The following error codes indicate a failure was detected:

Table 6-17. DLPC230-Q1 DMD Voltage Monitor Error Codes

ERROR CODE	DESCRIPTION
949	DMD VRESET voltage reading is out of range.
950	DMD VOFFSET voltage reading is out of range.
951	DMD VBIAS voltage reading is out of range.

6.4.1.13 DLPC230-Q1 TPS99000-Q1 Bandgap Monitor

The TPS99000-Q1 includes two bandgap voltage references: one is used in the ADC measurement system and one is used in the control systems. The ADC measures the scaled voltage of the control bandgap reference. Using the ADC system to measure the control bandgap ensures that the two bandgap references match. The DLPC230-Q1 reads this ADC measurement each frame. If this bandgap reference voltage is measured outside of a specified maximum and minimum value, the test will take action. The monitor ranges are specified in the flash header file provided along with the flash binary file.

6.4.1.13.1 Configuration

No configuration is required.

6.4.1.13.2 Execution

This test is always executed during display mode.

6.4.1.13.3 Failure Actions

An error will be logged on failure.

6.4.1.13.4 Error Codes

The following error codes indicate a failure was detected:

Table 6-18. DLPC230-Q1 TPS99000-Q1 Bandgap Monitor Error Codes

ERROR CODE	DESCRIPTION
1008	External bandgap voltage out of range.

6.4.1.14 DMD Temperature Monitor

The DLPC230-Q1 uses an external TMP411 to periodically read the DMD temperature using a temperature sensing diode built into the DMD. The main application monitors for the following conditions to take action:

- DMD over maximum park temperature
- DMD below minimum park temperature

Section 5.6 further describes temperature management.

6.4.1.14.1 Configuration

The maximum and minimum threshold temperature are set in flash.

6.4.1.14.2 Execution

This test is always executed.

6.4.1.14.3 Failure Actions

The main application will enter Standby mode to park the DMD. The main application will stay in Standby until the host requests a different operating mode. The operating mode command may receive an error if the respective DMD un-park temperature has not yet been reached. This is intended to avoid repeated temperature-related failures that may be caused by DMD temperature variation when illumination is enabled and disabled. If the DMD maximum threshold temperature is reached, the DMD voltages will be turned off in Standby mode.

6.4.1.14.4 Error Codes

The following error codes indicate a failure was detected:

Table 6-19. DMD Temperature Monitor Error Codes

ERROR CODE	DESCRIPTION
895	Temperature is lower than the minimum DMD park temperature.
896	Temperature is higher than the maximum DMD park temperature.

6.4.1.15 DMD Clock Monitor

The DMD clock generator is monitored to ensure that it continually operates within a specified frequency range. The main application sets up a hardware monitor upper and lower frequency limit. Hardware will flag an error if the clock frequency leaves this specified window.

6.4.1.15.1 Configuration

No configuration is required.

6.4.1.15.2 Execution

This test is always executed while the high-speed DMD interface is in use.

6.4.1.15.3 Failure Actions

Emergency shutdown will be performed on failure.

6.4.1.15.4 Error Codes

The following error codes indicate a failure was detected:

Table 6-20. DMD Clock Monitor Error Codes

ERROR CODE	DESCRIPTION
898	DMD clock out of range.

6.4.1.16 DMD High Speed Interface Training

The main application performs training of the high speed sub-LVDS data eye while the DMD high-speed interface is in use. This maintains optimized clock alignment over environmental conditions such as temperature. Every pin on the high-speed and low-speed bus is exercised during this tuning process, so any connection faults should be detected by the training. The training results are read back on the low-speed bus. If no valid training data is found for one or more lanes, the main application will take action.

6.4.1.16.1 Configuration

No configuration is required.

6.4.1.16.2 Execution

This test is always executed during display mode.

6.4.1.16.3 Failure Actions

An error will be logged on failure.

6.4.1.16.4 Error Codes

The following error codes indicate a failure was detected:

Table 6-21. DMD High Speed Interface Training Error Codes

ERROR CODE	DESCRIPTION
556	DMD HS training failed: returned invalid results.
573	DMD HS training failed: real-time test controller HS training error 1.
574	DMD HS training failed: real-time test controller HS training error 2.
575	DMD HS training failed: real-time test controller HS training error 3.
576	DMD HS training failed: real-time test controller HS training processing error.

6.4.1.17 DMD Low Speed Interface Test

The main application writes data to a reserved DMD register and then reads the value from the same register to confirm that the value matches the written value. The main application also periodically reads the DMD configuration reigsters to verify that the DMD register content remains correctly programmed. If an incorrect value is detected in any of these registers, the main application will take action.

6.4.1.17.1 Configuration

No configuration is required.

6.4.1.17.2 Execution

This test is always executed while the DMD low-speed interface is in use (Display mode).

6.4.1.17.3 Failure Actions

The register values will be re-written and the error will be logged.

6.4.1.17.4 Error Codes

The following error codes indicate a failure was detected:

Table 6-22. DMD Low Speed Interface Test Error Codes

ERROR CODE	DESCRIPTION
577	DMD Low Speed validation failed: processing error.

6.4.1.18 TPS99000-Q1 DLPC230-Q1 Processor Watchdog (WD1)

The TPS99000-Q1 includes a watchdog that monitors a periodic pulse generated by the DLPC230-Q1 using a DLPC230-Q1 GPIO pin. This is used to confirm that the DLPC230-Q1 main application is functional. The main

application configures the watchdog window during system initialization. If the pulse is not detected within the specified window, the TPS99000-Q1 will take action.

6.4.1.18.1 Configuration

A DLPC230-Q1 GPIO connects to TPS99000-Q1 WD1 signal for this monitoring. Refer to DLPC230-Q1 Data Sheet GPIO configuration for the specific GPIO.

6.4.1.18.2 Execution

This test is always executed.

6.4.1.18.3 Failure Actions

The TPS99000-Q1 will signal a park of the DMD and reset the chipset. The main application will read the reset cause from the TPS99000-Q1 and assert HOST_IRQ during reset initialization.

6.4.1.18.4 Error Codes

The following error codes indicate a failure was detected:

Table 6-23. TPS99000-Q1 DLPC230-Q1 Processor Watchdog Error Codes

ERROR CODE	DESCRIPTION
1011	TPS99000-Q1 watchdog 1 did not detect a rising edge within the expected trigger window.

6.4.1.19 TPS99000-Q1 DLPC230-Q1 Sequencer Watchdog (WD2)

The TPS99000-Q1 includes a watchdog that monitors DLPC230-Q1 sequencer operation. The main application configures the TPS99000-Q1 watchdog window to approximately 7 frame times in length. If the SEQ_START pulse is not detected within the specified window, the TPS99000-Q1 will interrupt the DLPC230-Q1 main application to take action.

6.4.1.19.1 Configuration

SEQ_START from the DLPC230-Q1 connects to TPS99000-Q1 WD2 signal for this monitoring.

6.4.1.19.2 Execution

This test is always executed while displaying an image.

6.4.1.19.3 Failure Actions

The sequencer will attempt to disable and re-enable the sequencer for 3 consecutive failures. On the fourth failure, an emergency shutdown will be performed. When the sequencer is disabled, the display will be lost momentarily.

6.4.1.19.4 Error Codes

The following error codes indicate a failure was detected:

Table 6-24. TPS99000-Q1 DLPC230-Q1 Sequencer Watchdog Error Codes

ERROR CODE	DESCRIPTION
1012	TPS99000-Q1 watchdog 2 did not detect a rising edge within the expected trigger window.

6.4.1.20 TPS99000-Q1 Temperature Warning / Error

The TPS99000-Q1 includes temperature detection for a warning and error threshold temperature. A temperature warning will set a status register and notify the DLPC230-Q1 main application which will take an initial warning action. If the error threshold is exceeded, the TPS99000-Q1 will take an error action. The TPS99000-Q1 Data Sheet specifies the warning and error threshold temperature levels.

6.4.1.20.1 Configuration

No configuration is required.

6.4.1.20.2 Execution

This test is always executed.

6.4.1.20.3 Failure Actions

Warning: An error will be logged to notify of the TPS99000-Q1 temperature warning.

Error: The TPS99000-Q1 will park the DMD and enter an off state. The TPS99000-Q1 will not exit this state until the temperature returns to a valid operating temperature.

6.4.1.20.4 Error Codes

The following error codes indicate a failure was detected:

Table 6-25. TPS99000-Q1 Temperature Warning/Error Error Codes

ERROR CODE	DESCRIPTION
890	TPS99000-Q1 thermal conditions on chip have reached the warning level. If temperature continues to rise, system will reach die over-temp error temperature and emergency actions will be taken by TPS99000-Q1.

6.4.1.21 TPS99000-Q1 Clock Ratio Monitor

The TPS99000-Q1 calculates a ratio between its internal clock and the external DLPC230-Q1 clock input in order to validate proper frequency operation of the main DLPC230-Q1 clock source. The DLPC230-Q1 main application periodically reads this ratio and takes action if the ratio is outside of the expected range.

6.4.1.21.1 Configuration

No configuration is required.

6.4.1.21.2 Execution

This test is always executed.

6.4.1.21.3 Failure Actions

The error is logged if the ratio is out of range.

6.4.1.21.4 Error Codes

The following error codes indicate a failure was detected:

Table 6-26. TPS99000-Q1 Clock Ratio Monitor Error Codes

ERROR CODE	DESCRIPTION
961	Sequence clock ratio is not within the specified limits.

6.4.1.22 TPS99000-Q1 Register Password Lock

The TPS99000-Q1 includes a password lock on register banks to prevent unexpected register changes. During operation the main application writes register values to the TPS99000-Q1 and then sets the password lock until the next register update must be performed.

6.4.1.22.1 Configuration

No configuration is required.

6.4.1.22.2 Execution

This function is always performed while TPS99000-Q1 registers are written.

6.4.1.22.3 Failure Actions

No failure action.

6.4.1.22.4 Error Codes

No associated error codes.

6.4.1.23 TPS99000-Q1 Register Checksum

The TPS99000-Q1 includes checksum coverage of critical illumination-related registers. During operation the main application writes register values to the TPS99000-Q1 and then updates the checksum values to match the newly written values. If a checksum error is detected and reported by the TPS99000-Q1, the main application will take action.

6.4.1.23.1 Configuration

No configuration is required.

6.4.1.23.2 Execution

This test is always executed while TPS99000-Q1 registers are written.

6.4.1.23.3 Failure Actions

The main application will attempt to re-write all registers 3 times. If the failure persists, the main application will execute an emergency shutdown.

6.4.1.23.4 Error Codes

The following error codes indicate a failure was detected:

Table 6-27. TPS99000-Q1 Register Checksum Error Codes

ERROR CODE	DESCRIPTION
938	TPS99000-Q1 checksum error in group 1 register set.
939	TPS99000-Q1 checksum error in group 2 register set.
940	TPS99000-Q1 checksum error in group 3 register set.

6.4.1.24 Software Monitor Thread

The main application uses a real-time operating system (RTOS), and threads are created to support various aspects of functionality. A monitoring thread is also created with the purpose of checking that other threads are functioning as expected. Each thread must check in with the monitoring thread within a specified amount of time, or failure action will be taken.

6.4.1.24.1 Configuration

No configuration is required.

6.4.1.24.2 Execution

This test is always executed.

6.4.1.24.3 Failure Actions

The main application will log an error.

6.4.1.24.4 Error Codes

The following error codes indicate a failure was detected:

Table 6-28. Software Monitor Thread Error Codes

ERROR CODE	DESCRIPTION
634	Thread is non-responsive within timeout.

6.4.2 Non-Periodic Tests

Non-periodic tests are run while the system is not displaying an image, typically once at power-up or power-down. Non-periodic tests that can be commanded by the host must be executed while the main application is in Standby mode. The coverage of these tests requires full usage of data paths and specific internal configuration that cannot be accomplished while displaying an image. Optional tests can be setup through flash settings or host commands. Non-optional BISTs will always be performed.

Figure 6-6 shows a block diagram with numbering on DLPC230-Q1 internal blocks and various system components. The numbering is used in Table 6-29 to identify the coverage of each test.

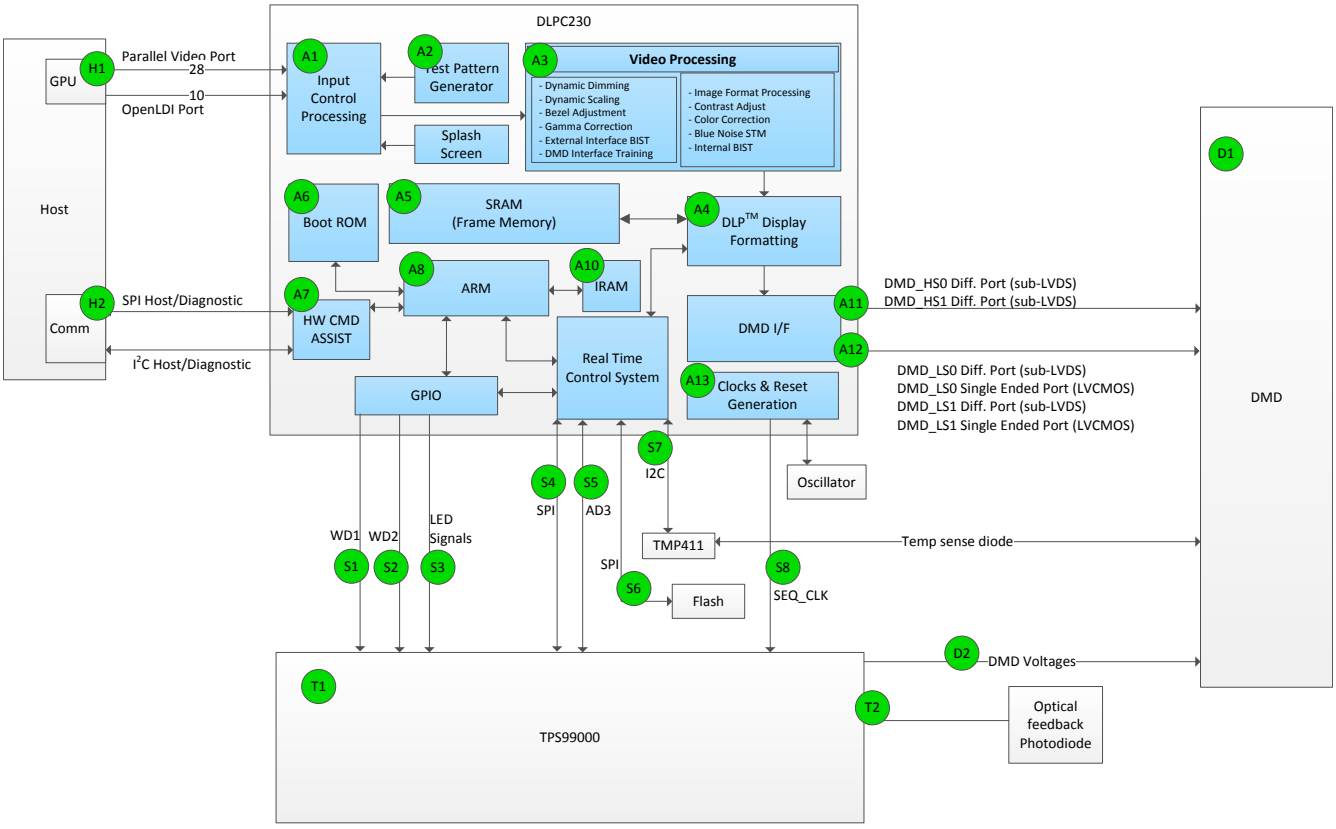


Figure 6-6. Test Coverage System Block Diagram

Table 6-29. Non-Periodic Tests Overview

TEST	COVERAGE	APPLICATION	COMMAND	FLASH OPTION	FAIL ACTION	SECTION
DLPC230-Q1 Front End Functional Test	A1, A2, A3	Main	Yes	Yes	Hold in Standby	Section 6.4.2.2
DLPC230-Q1 Back End Functional Test	A3, A4	Main	Yes	Yes	Hold in Standby	Section 6.4.2.3
DLPC230-Q1 Memory BISTs	A3, A4, A5	Main	Yes	Yes	Hold in Standby	Section 6.4.2.4
TPS99000-Q1 Signal Interface Test	S3	Main	No	Always Executed	Hold in Standby	Section 6.4.2.5
DMD Memory Test	A12, D1	Main	Yes	Yes	Hold in Standby	Section 6.4.2.6
Flash Data Verification	S6	Main / Boot	Yes	Yes	Hold in Standby (Main) Stay in Boot (Boot)	Section 6.4.2.7
DLPC230-Q1 Boot ROM CRC	A6	Boot	Always executed		Stay in Boot	Section 6.4.2.8
DLPC230-Q1 Flash Table CRC	S6	Boot	Always executed		Stay in Boot	Section 6.4.2.9

Table 6-29. Non-Periodic Tests Overview (continued)

TEST	COVERAGE	APPLICATION	COMMAND	FLASH OPTION	FAIL ACTION	SECTION
DLPC230-Q1 Main Application CRC	A10, A6	Boot	Always executed		Stay in Boot	Section 6.4.2.10
DLPC230-Q1 Command and Flash Interface Memory Test	A7	Boot	No	Yes	Stay in Boot	Section 6.4.2.11

6.4.2.1 Execution Time

Optional non-periodic tests will impact power-up time, and therefore the time to display image content. Table 6-30 provides approximate execution times of non-periodic tests for design consideration. These values are not specifications, and actual execution times will vary.

Flash bandwidth will impact test speed since test data is stored in flash. Bandwidth is calculated as number of read lines multiplied by SPI data rate. For example, quad I/O SPI at 49.41MHz has a bandwidth of 197.64 Mbps.

A test setup time must be accounted for once if any number of tests are performed. Refer to Table 6-31. The formula for execution time is $t_{Test\_Setup} + \sum t_{Test}$. Test setup time does not apply to Boot Application tests.

Table 6-30. Non-Periodic Test Execution Time

TEST	TIME (ms)	
	LOW FLASH BANDWIDTH (50.92 Mbps)	HIGH FLASH BANDWIDTH (197.64 Mbps)
DLPC230-Q1 Front End Functional Test	20	18
DLPC230-Q1 Back End Functional Test	10	7
DLPC230-Q1 Memory BISTs	47	28
TPS99000-Q1 Signal Interface Test	3	2
DMD Memory Test	42	38
Flash Data Verification (Boot)		180
DLPC230-Q1 Command and Flash Interface Memory Test (Boot)		1

Table 6-31. Non-Periodic Test Setup Times

TEST	TIME (ms)	
	LOW FLASH BANDWIDTH (50.92 Mbps)	HIGH FLASH BANDWIDTH (197.64 Mbps)
Test Setup	2	1

6.4.2.2 DLPC230-Q1 Front End Functional BIST (Main)

This set of tests generates test patterns and runs them through the entire video processing block. For each frame of data a CRC is generated and compared to the expected value. If any of the CRC values do not match the expected value, the main application will take action.

6.4.2.2.1 Configuration

No configuration is required.

6.4.2.2.2 Execution

A flash option determines whether this test is executed during initialization, prior to displaying an image. The value of this flash option is specified in the flash header file that is provided along with the flash data.

Execute Non-Periodic BIST command can be used to execute this test while in Standby mode.

6.4.2.2.3 Failure Actions

The main application will stay in Standby mode even if it receives a host command to transition to another mode.

6.4.2.2.4 Error Codes

The following error codes indicate a failure was detected:

Table 6-32. DLPC230-Q1 Front End Functional BIST Error Codes

ERROR CODE	DESCRIPTION
524	Front end BIST failed.

6.4.2.3 DLPC230-Q1 Back End Functional BIST (Main)

This set of tests uses the video processing block to output test data through the entire back end formatting block. For each frame of data a CRC is generated and compared to the expected value. If any of the CRC values do not match the expected value, the main application will take action.

6.4.2.3.1 Configuration

No configuration is required.

6.4.2.3.2 Execution

A flash option determines whether this test is executed during initialization, prior to displaying an image. The value of this flash option is specified in the flash header file that is provided along with the flash data.

Execute Non-Periodic BIST command can be used to execute this test while in Standby mode.

6.4.2.3.3 Failure Actions

The main application will stay in Standby mode even if it receives a host command to transition to another mode.

6.4.2.3.4 Error Codes

The following error codes indicate a failure was detected:

Table 6-33. DLPC230-Q1 Back End Functional BIST Error Codes

ERROR CODE	DESCRIPTION
512	Back end BIST failed.

6.4.2.4 DLPC230-Q1 Memory BISTs (Main)

These tests are a series of writes, delays, and reads to most internal memories. If any of the read values do not match the expected result, the main application will take action.

6.4.2.4.1 Configuration

No configuration is required.

6.4.2.4.2 Execution

A flash option determines whether this test is executed during initialization, prior to displaying an image. The value of this flash option is specified in the flash header file that is provided along with the flash data.

Execute Non-Periodic BIST command can be used to execute this test while in Standby mode.

6.4.2.4.3 Failure Actions

The main application will stay in Standby mode even if it receives a host command to transition to another mode.

6.4.2.4.4 Error Codes

The following error codes indicate a failure was detected:

Table 6-34. DLPC230-Q1 Memory BISTs Error Codes

ERROR CODE	DESCRIPTION
519	Memory BIST 17 failed.
520	Memory BIST 18 failed.
521	Memory BIST 19 failed.

Table 6-34. DLPC230-Q1 Memory BISTs Error Codes (continued)

ERROR CODE	DESCRIPTION
522	Memory BIST 20 failed.
523	Memory BIST 21 failed.
530	Memory BIST 1 failed.
531	Memory BIST 2 failed.
533	Memory BIST 22 failed.
536	Memory BIST 23 failed.
537	Memory BIST 24 failed.
538	Memory BIST 26 failed.
539	Memory BIST 25 failed.
540	Memory BIST 3 failed.
541	Memory BIST 12 failed.
542	Memory BIST 13 failed.
543	Memory BIST 14 failed.
544	Memory BIST 15 failed.
545	Memory BIST 16 failed.
546	Memory BIST 4 failed.
547	Memory BIST 5 failed.
548	Memory BIST 6 failed.
549	Memory BIST 7 failed.
550	Memory BIST 8 failed.
551	Memory BIST 9 failed.
552	Memory BIST 10 failed.
553	Memory BIST 11 failed.

6.4.2.5 TPS99000-Q1 Interface Signal Connection Test (Main)

The DLPC230-Q1 main application forces values on each LEDSEL, SEN, and DEN signals to the TPS99000-Q1. It then reads back the signal states from the TPS99000-Q1 using SPI.

The test is considered a pass if the values of the signals that are read back do **not** match, since these signals are not intended to be connected between the DLPC230-Q1 and TPS99000-Q1 for headlight applications. If the signals are read back with the correct values, the test is considered a fail. A fail suggests that the hardware configuration may not match the flash data's expected product configuration.

6.4.2.5.1 Configuration

No configuration is required.

6.4.2.5.2 Execution

This test is always executed during main application initialization.

6.4.2.5.3 Failure Actions

The main application will stay in Standby mode even if it receives a host command to transition to another mode.

6.4.2.5.4 Error Codes

The following error codes indicate a failure was detected:

Table 6-35. TPS99000-Q1 Interface Signal Connection Test Error Codes

ERROR CODE	DESCRIPTION
532	TPS99000-Q1 signal interface BIST failed.

6.4.2.6 DMD Memory Test (Main)

The main application commands the DMD into a testing mode and the DMD writes known values into the memory cells below its pixels. The DMD then reads back the state of each memory cell and drives a signal to the DLPC230-Q1 to indicate pass or fail for each column of the DMD memory. A column is a top-to-bottom, width 1 array of memory cells (referred to as a DMD row by DMD design convention). A column will be reported as a fail if one or more memory cells in that column reads an incorrect value. The DLPC230-Q1 main application then sums the number of failed columns. Note that results cannot be reported at a per-memory-cell level of detail because the results are only transmitted to the DLPC230-Q1 at column-level granularity.

This process is executed four times and the number of failed columns are summed from each execution. The four executions includes two opposite checkerboard patterns on the two halves of the DMD. The use of two checkerboard patterns ensures that every memory cell is tested at both high and low state. Although the results of these four tests are summed, the data from each test is unique meaning that no memory cell fail should be counted more than once.

The pass/fail criteria are shown in Table 6-36.

Table 6-36. DMD Memory Test Pass/Fail Criteria

NUMBER OF COLUMN FAILS	TEST RESULT
0 or 1	Pass
2 or more	Fail

6.4.2.6.1 Configuration

No configuration is required.

6.4.2.6.2 Execution

A flash option determines whether this test is executed during initialization, prior to displaying an image. The value of this flash option is specified in the flash header file that is provided along with the flash data.

Execute Non-Periodic BIST command can be used to execute this test while in Standby mode.

6.4.2.6.3 Failure Actions

The main application will stay in Standby mode even if it receives a host command to transition to another mode.

6.4.2.6.4 Error Codes

The following error codes indicate a failure was detected. If the test fails, both errors will be received on a single execution of the test. One error is informational to specify the actual number of columns that failed.

Table 6-37. DMD Memory Test Error Codes

ERROR CODE	DESCRIPTION
220	DMD Memory BIST columns failed. The information bits of this error indicate the number of columns that failed.
221	DMD Memory BIST failed.

6.4.2.7 Flash Data Verification (Boot/Main)

Both the boot application and main application support flash verification by generating a CRC of flash data blocks and comparing them to the expected values stored with each flash block. This is typically performed immediately after flash programming, but it can also be performed as part of power-up or power-down. Note that this test requires reading the entire flash image so it can add significant time to power-up. The time will increase with the size of the flash image.

6.4.2.7.1 Configuration

No configuration is required.

6.4.2.7.2 Execution

- A flash option determines whether this test is executed during boot application initialization.

- *Flash Verify Data* (Main) command can be used to execute this test while in main application Standby mode.
- *Flash Verify Data* (Boot) command can be used to execute this test while in boot application.

The value of these flash options are specified in the flash header file that is provided along with the flash data.

6.4.2.7.3 Failure Actions

Boot application: The system will stay in boot application execution.

Main application: The system will stay in Standby mode even if it receives a host command to transition to another mode.

6.4.2.7.4 Error Codes

The following error codes indicate a failure was detected:

Table 6-38. Flash Data Verification Error Codes

ERROR CODE	DESCRIPTION
134	A CRC error was detected while verifying the contents of the flash.
762	Flash data verification CRC failed.

6.4.2.8 DLPC230-Q1 Boot ROM CRC (Boot)

The boot application runs a CRC on the boot ROM data and compares it to an expected value stored in the boot ROM memory.

6.4.2.8.1 Configuration

No configuration is required.

6.4.2.8.2 Execution

This test is always executed during boot application initialization prior to exercising any boot application functionality.

6.4.2.8.3 Failure Actions

The system stays in boot application execution if this test fails.

6.4.2.8.4 Error Codes

The following error codes indicate a failure was detected:

Table 6-39. DLPC230-Q1 Boot ROM CRC Error Codes

ERROR CODE	DESCRIPTION
131	A CRC was error was detected while transferring the boot application to RAM.

6.4.2.9 DLPC230-Q1 Flash Table CRC (Boot)

The boot application always runs a CRC of the flash table at the top of flash and compares the calculated CRC to the CRC that is stored with the flash table. This flash table points to other addresses in flash that are used during operation. If the CRC values do not match, the boot application will take action.

6.4.2.9.1 Configuration

No configuration is required.

6.4.2.9.2 Execution

This test is always executed by boot application.

6.4.2.9.3 Failure Actions

The system stays in boot application execution if this test fails.

6.4.2.9.4 Error Codes

The following error codes indicate a failure was detected:

Table 6-40. DLPC230-Q1 Flash Table CRC Error Codes

ERROR CODE	DESCRIPTION
30	A CRC error was detected while verifying the Flash table.

6.4.2.10 DLPC230-Q1 Main Application CRC (Boot)

The DLPC230-Q1 boot application runs a CRC of the main application data by reading it from IRAM after it has been transferred from flash to IRAM. If this test fails, the boot application will take action.

6.4.2.10.1 Configuration

No configuration is required.

6.4.2.10.2 Execution

This test is always executed by boot application.

6.4.2.10.3 Failure Actions

The system stays in boot application execution if this test fails.

6.4.2.10.4 Error Codes

The following error codes indicate a failure was detected:

Table 6-41. DLPC230-Q1 Main Application Transfer CRC Error Codes

ERROR CODE	DESCRIPTION
132	A CRC error was detected while transferring the main application to RAM.
133	A CRC error was detected after transferring the main application to RAM.

6.4.2.11 DLPC230-Q1 Command and Flash Interface Memory Test (Boot)

The DLPC230-Q1 boot application can optionally run a memory test on the command interface memories and flash data access memories. These tests are a series of writes, delays, and reads to memory. If any of these tests fails, the boot application will take action.

6.4.2.11.1 Configuration

No configuration is required.

6.4.2.11.2 Execution

A flash option determines whether this test is executed.

6.4.2.11.3 Failure Actions

The system stays in boot application execution if this test fails.

6.4.2.11.4 Error Codes

The following error codes indicate a failure was detected:

Table 6-42. DLPC230-Q1 Command and Flash Interface Tests Error Codes

ERROR CODE	DESCRIPTION
534	DLPC230-Q1 Command and Flash Interface Memory test failed.

6.4.3 Interface Tests

Interface tests are features built into communication protocols to confirm proper communication between chips.

Figure 6-7 shows a block diagram with numbering on DLPC230-Q1 internal blocks and various system components. The numbering is used in Table 6-43 to identify the coverage of each test.

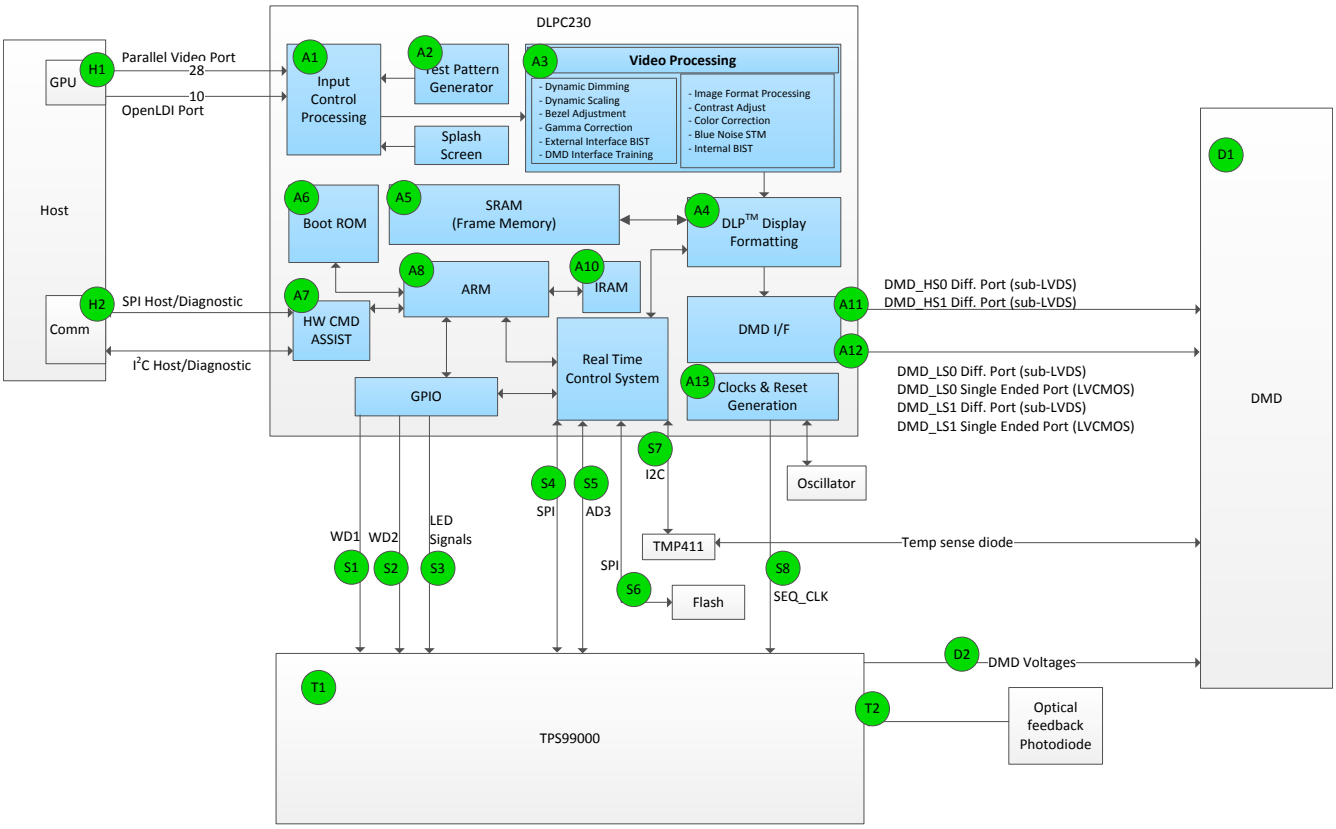


Figure 6-7. Test Coverage System Block Diagram

Table 6-43. Interface Tests Overview

TEST	COVERAGE	FAIL ACTION	SECTION
Temperature Sensor Interface	S7	Emergency Shutdown	Section 6.4.3.1
DLPC230-Q1 to TPS99000-Q1 SPI Interface	S4	DLPC230-Q1 Reset	Section 6.4.3.2
DLPC230-Q1 to TPS99000-Q1 ADC Interface	S5	Emergency Shutdown	Section 6.4.3.3
DMD Socket Connectivity Test	DMD Socket	Log Error	Section 6.4.3.4

6.4.3.1 Temperature Sensor Interface

The DLPC230-Q1 communicates with the TMP411 temperature sensor using I^2C. If a NAK is received during a communication attempt, the command will be re-attempted up to 3 times. If these attempts fail, the main application will take action.

6.4.3.1.1 Failure Actions

The main application will execute emergency shutdown.

6.4.3.2 DLPC230-Q1 to TPS99000-Q1 SPI Interface

The DLPC230-Q1 is the host controller of a SPI communication interface with the TPS99000-Q1. This SPI protocol uses 9-bit bytes with one parity bit per byte. Each write and read transaction returns the parity error status from the TPS99000-Q1. The returned parity error status and the parity of the returned bytes must be correct for the command to be considered successful. If 3 consecutive frames of commands fail, the main application will take action.

6.4.3.2.1 Failure Actions

The main application will stop servicing TPS99000-Q1 processor watchdog (WD1) so that the TPS99000-Q1 resets the DLPC230-Q1.

6.4.3.3 DLPC230-Q1 to TPS99000-Q1 ADC Interface

The DLPC230-Q1 receives ADC data from the TPS99000-Q1 over a dedicated SPI-like interface. Each write transaction on this interface includes repeated command bits and an odd parity bit. The two command packets must match and the parity bit must be correct for the write transaction to be considered valid. The return data includes error bit status for the previous transaction, repeated data and error bits, and an odd parity bit. The two copies of the read data and error bits must match, and the parity bit must be correct for the write transaction to be considered valid. If 3 consecutive frames of commands exhibit failure, the main application will take action.

6.4.3.3.1 Failure Actions

The main application will execute emergency shutdown.

6.4.3.4 DMD Socket Connectivity Test

This test enables testing of the connectivity between the DLPC230-Q1 and the DMD socket. This test cannot be used during operation. Each pin or differential pair for the DMD high speed interface, low speed interface, DEN_ARSTZ, and DMD reset voltages can be set high or low for testing. These signals can be check by test equipment such as multimeters, oscilloscopes, or other probes to ensure connectivity between the DMD socket and DLPC230S-Q1.

6.4.3.4.1 Configuration

Section 8.2.25 can be used to write the state of each pin, pin pair, or voltages for testing purposes. This test can only be executed with no DMD in the system.

6.4.3.4.2 Execution

This test can be executed by the user only when no DMD is attached to the socket

6.4.3.4.3 Failure Actions

An error will be logged on failure

6.4.3.4.4 Error Codes

The following error codes indicate a failure was detected:

ERROR CODE	DESCRIPTION
246	DMD socket connectivity test is able to read DMD ID. A DMD is inserted in the socket.
247	Reset required after empty socket test.

Chapter 7

Commands - Boot Application

7.1 Command Table

Table 7-1 summarizes the boot application commands.

Table 7-1. Command Summary - Boot Application

COMMAND	TYPE	OP CODE	SECTION
System Reset	Write	00h	Section 7.2.1
Read Pre-fetch	Write	01h	Section 7.2.2
Read Activate	Read	02h	Section 7.2.3
System Software Version	Read	B0h	Section 7.2.4
Flash Device ID	Read	B1h	Section 7.2.5
Short Status	Read	C0h	Section 7.2.6
Error History	Read	C1h	Section 7.2.7
Clear Short Status Errors	Write	C2h	Section 7.2.8
Clear Error History	Write	C3h	Section 7.2.9
Flash Full Erase	Write	E0h	Section 7.2.10
Flash Write Data	Write (Bulk)	E1h	Section 7.2.11
Flash Verify Data	Read	E2h	Section 7.2.12
Flash Interface Rate	Write	E3h	Section 7.2.13
Flash Interface Rate	Read	E4h	Section 7.2.14

7.2 Command Definitions

7.2.1 System Reset - Write (00h)

This command is used to reset the DLPC230-Q1. The proper byte signature must be transmitted to initiate the reset in order to prevent unintended system resets.

When the command is received, the DLPC230-Q1 will perform a full reset except for the main PLL. This will not reset the DMD or the TPS99000-Q1. In order to reset the entire chipset, the external *PROJ_ON* signal should be used.

Write Parameters

Table 7-2. System Reset Write Parameters (Boot)

BYTE	BITS	DESCRIPTION
1	7:0	**Signature Byte 1** 0xAA
2	7:0	**Signature Byte 2** 0xBB
3	7:0	**Signature Byte 3** 0xCC
4	7:0	**Signature Byte 4** 0xDD

7.2.2 Read Pre-Fetch - Write (01h)

This command is used to send the desired read command op-code and associated command parameters to initiate a read request from embedded software. The desired read data will not be returned during this command transaction. More information on the read procedure can be found in Section 3.5.

Write Parameters

Table 7-3. Read Pre-Fetch Write Parameters (Boot)

BYTE	BITS	DESCRIPTION
1	7:0	**Read command op-code**
2 ... n		**Read command parameters** Depending on the Read op-code, a specific number of command parameter bytes will be expected. The required byte parameters are documented with each read command.

7.2.3 Read Activate (02h)

This command is used to activate the read operation in order to retrieve the previously requested read data. This generic command is used to retrieve all requested read data with the exception of the *Short Status* command. More information on the read procedure can be found in Section 3.5.

Command Parameters

No command parameters.

Return Parameters

Table 7-4. Read Activate Return Parameters (Boot)

BYTE	BITS	DESCRIPTION
1 ... n		**Data Bytes 1...n** Read data bytes. The number of bytes will vary depending on the command that is currently being read.

7.2.4 System Software Version - Read (B0h)

This command is used to read the software version of the boot application.

Command Parameters

No command parameters.

Return Parameters

Table 7-5. System Software Version Return Parameters (Boot)

BYTE	BITS	DESCRIPTION
2:1	7:0	**DLPC230-Q1 Boot Application Version - Patch** LSByte = 1
3	7:0	**DLPC230-Q1 Boot Application Version - Minor**
4	7:0	**DLPC230-Q1 Boot Application Version - Major**

7.2.5 Flash Device ID - Read (B1h)

This command is used to read flash device information.

Command Parameters

No command parameters.

Return Parameters

Table 7-6. Flash Device ID Return Parameters (Boot)

BYTE	BITS	DESCRIPTION
1	7:0	**Flash Manufacturer ID**
2	7:0	**Flash Memory Type**
3	7:0	**Flash Memory Capacity**

The manufacturer ID and memory type can be found from the selected flash device specification.

The flash memory size can be determined from Table 7-7.

Table 7-7. Flash Memory Size Values

MEMORY CAPACITY (BYTE 3)	ACTUAL SIZE
0x15	16Mb
0x16	32Mb
0x17	64Mb
0x18	128Mb
0x19	256Mb(1)

(1) Only the first 128Mb of address space is useable.

7.2.6 Short Status - Read (C0h)

This command is used to read the short status from hardware. This is the only read command that does not require the use of *Read Pre-Fetch* and *Read Activate* commands. Refer to Section 3.5 for more information on the *Short Status* protocol.

Command Parameters

No command parameters.

Return Parameters

Table 7-8. Short Status Return Parameters (Boot)

BYTE	BITS	DESCRIPTION
1	7:6	**Application / Mode** 0x0: Boot application 0x1: Main application - Standby 0x2: Main application - Display
	5	**Emergency Shutdown** 0x0: Not activated 0x1: Activated
	4	***Reserved***
	3	**Read Data Available** 0x0: No data available 0x1: Data available
	2	**System Busy** 0x0: Not busy 0x1: Busy
	1	**Request in Progress** 0x0: Not in progress 0x1: In progress
	0	**System Initialized** 0x0: Not initialized 0x1: Initialized
2	7:0	**Execution Command Tag**
4:3	15	**BIST Error** 0x0: No error 0x1: Error
	14	**Operational Error** 0x0: No error 0x1: Error
	13	**Command Error** 0x0: No error 0x1: Error
	12	**Communication Error** 0x0: No error 0x1: Error
	11:0	**CMD/COMM Error Code**

A diagram of these short status bits is shown in Figure 7-1.

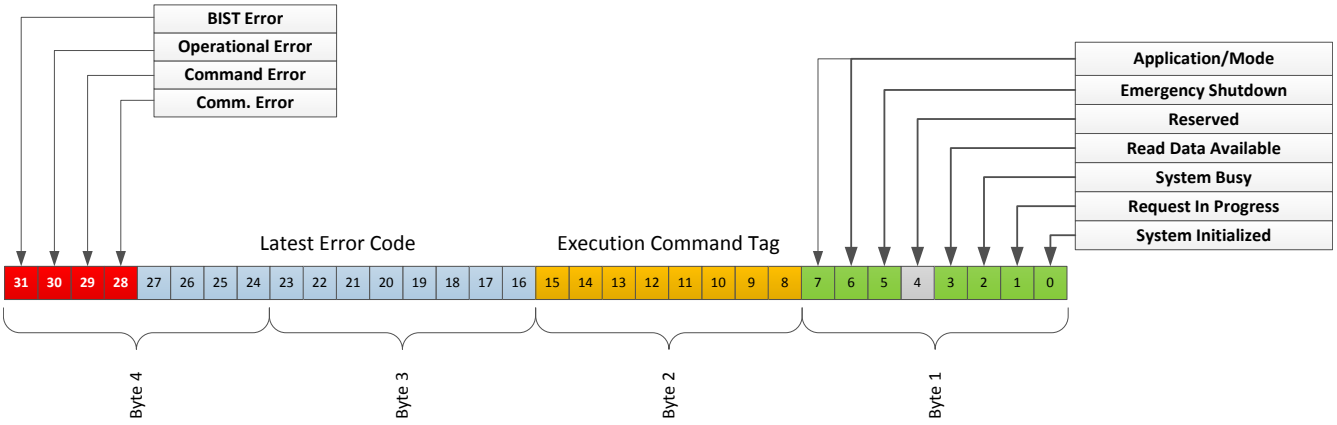


Figure 7-1. Short Status Bit Definition

The bits are described in Table 7-9.

Table 7-9. Short Status Field Descriptions (Boot)

BIT FIELD	DEFINITION
System Initialized	Indicates that the system software is ready to accept commands for processing. Typically this would be set by the Main Application, except when the system is forced to stay in boot.
Request in Progress	This bit is used to inform the host that a commanded task is being performed. When the task is initiated, this bit will be set to 1 and when the operation has completed the bit will be cleared to 0. If the task was a BIST, the results for the requested test will be valid once the bit has been cleared by embedded software. Further requests can be started after this.
System Busy	This bit is used to inform the host that the system's receive FIFO is full. The host should not send any more commands when the system is busy, or the commands and associated data may be lost. The Host is free to send commands when the system is not busy.
Read Data Available	Indicates when read data is available after the host has sent a Read Prefetch command. When data is available, the Host should send the Read Activate command to fetch the requested data. The Host should always fetch requested data using the Read Activate command before sending another Read Prefetch command. If a Read Prefetch command is sent before the data from a previous Read Prefetch command has been fetched, the previous data will be flushed, and the latest requested data will be made available for fetching by Read Activate. There will be no error indication that this has occurred.
Emergency Shutdown	This bit is used to indicate that the system has automatically gone to Standby Mode due to a critical system error. The specifics of the error may be available via the Error History.
Application/Mode	These bits indicate whether the boot application or main application is being executed. During main application execution, the current operating mode is also specified.
Execution Command Tag	This byte contains the command tag for the last write command received that has been executed, whether successful or not. This tag is used along with the Error Code to determine if the command associated with this tag was executed successfully or not. This byte is continually updated as new commands are received and executed. In the Boot Application, this byte is also used for any system errors. For errors where the command tag is valid, the actual tag will be used. For errors where there is no command tag or the command tag may be invalid, a null (0h) command tag will be provide in this field. In both cases, the appropriate error code will be provided in the error code field.
Error Code	The 12-bit error code is used to specify the last error received during system operation. The error code can indicate no error (error code = 0h), or indicate the error code for the most recent error to occur. This error code is continually updated as new errors occur. The *Error History* command can be used to obtain details about previous errors. For any communication errors, the error code for this error will also be provided in this field.

Table 7-9. Short Status Field Descriptions (Boot) (continued)

BIT FIELD	DEFINITION
Communication Error	A flag set to indicate a communication error, which is used to indicate a problem with the transmission/ reception of a command. Some examples are: • RXFIFO overflow • Command transmission terminated early (the host didn't provide enough SPI clock pulses for all requested data). • Command transmission terminated late (the host provided too many SPI clock pulses for requested data).
Command Error	A flag set to indicate a command error, or an error in the action requested by the command. Some examples are: • Command executed in an invalid operating mode • CRC Error in Command Header • CRC Error in Payload (Bulk Command) • Invalid command op-code • Invalid command parameter (for example, out of range) • Incorrect number of command parameters • Non-periodic BIST failure when BIST started by command • Error erasing, writing, or reading flash when commanded • Flash overflow error
Operational Error	A flag set to indicate an operational error, which are any errors that don't fall into one of the other three error categories. Some examples are: • Sequence CRC error • CMT CRC error • Sequence / CMT mismatch error • Periodic BIST failure • Unable to communicate with TPS99000-Q1 • Unable to communicate with temperature sensor
BIST Error	A flag set to indicate a non-periodic or periodic BIST error.

7.2.7 Error History - Read (C1h)

This command is used to read the details of the first 62 errors that the system encounters. Subsequent errors beyond the first 62 will not be stored, but the error count will continue to increase to indicate that more errors occurred. Four bytes are allocated for each error. The error history can be cleared using the *Clear Error History* command. The command will always return 249 bytes regardless of the number of errors currently contained in the error history.

Boot Application error codes are defined in Section A.1.

Command Parameters

No command parameters.

Return Parameters

Table 7-10. Error History Return Parameters (Boot)

BYTE	BITS	DESCRIPTION
1	7:0	**Error Count** Count of errors received since the last power-cycle or clear of the error history. If the error count is greater than 62, some error details are not available since storage is full.
5:2	31:0	**Error 1 Details** LSByte = 2
9:6	31:0	**Error 2 Details** LSByte = 6
...	...	**Error n Details**
249:246	31:0	**Error 62 Details** LSByte = 246

Each set of error details includes a unique error code, category flags, and additional information bits that vary based on the type of error. The bit definition for each error is broken down as follows:

Table 7-11. Error Detail Bit Descriptions (Boot)

BITS	DESCRIPTION
31	**BIST Error** 0x1: This error is categorized as a BIST error
30	**Operational Error** 0x1: Error is categorized as an operational error
29	**Command Error** 0x1: Error is categorized as a command error
28	**Communication Error** 0x1: Error is categorized as a communication error
27	***Reserved*** Always 0
26:16	**Error code** Unique error code to identify the error.
15:0	**Informational bits** These can vary depending on the category and specific type of error.

A diagram of the error detail fields is shown in Figure 7-2:

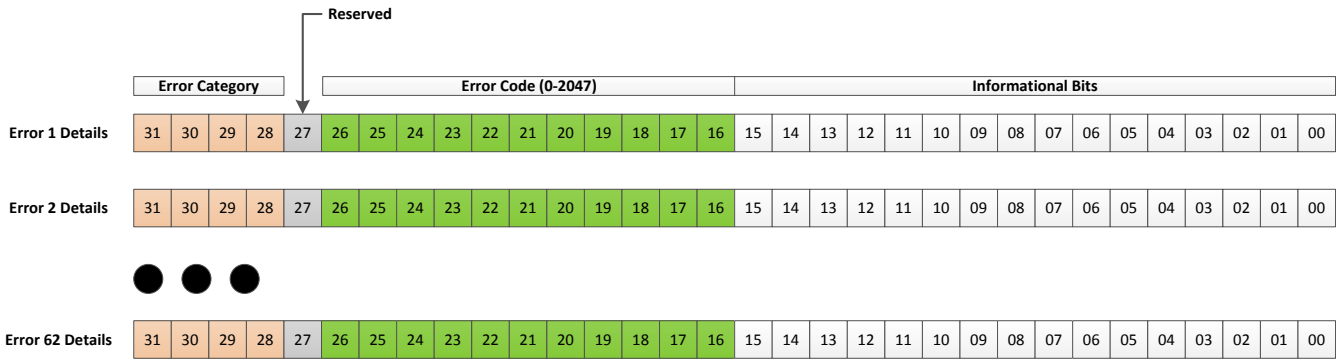


Figure 7-2. Error History Details

7.2.8 Clear Short Status Errors - Write (C2h)

This command is used to clear the Short Status error bits. This includes bytes 3 and 4 of the Short Status.

Write Parameters

Table 7-12. Clear Short Status Errors Write Parameters (Boot)

BYTE	BITS	DESCRIPTION
1	7:0	**Signature Byte 1** 0xAA
2	7:0	**Signature Byte 2** 0xBB
3	7:0	**Signature Byte 3** 0xCC
4	7:0	**Signature Byte 4** 0xDD

7.2.9 Clear Error History - Write (C3h)

This command is used to clear the Error History, the Error Count, and the Short Status error bits.

Write Parameters

Table 7-13. Clear Error History Write Parameters (Boot)

BYTE	BITS	DESCRIPTION
1	7:0	**Signature Byte 1** 0xAA
2	7:0	**Signature Byte 2** 0xBB
3	7:0	**Signature Byte 3** 0xCC
4	7:0	**Signature Byte 4** 0xDD

7.2.10 Flash Full Erase - Write (E0h)

This command directs the boot application to erase the entire contents of the flash device.

The *Request in Progress* bit of the Short Status will be set at the start of the flash erase process and will be cleared when the erase process is complete.

The error bits in the Short Status can be used to determine whether an error occurred during the erase process.

More information on flash programming can be found in Section 5.3.3.

Write Parameters

Table 7-14. Flash Full Erase Write Parameters (Boot)

BYTE	BITS	DESCRIPTION
1	7:0	**Signature Byte 1** 0xAA
2	7:0	**Signature Byte 2** 0xBB
3	7:0	**Signature Byte 3** 0xCC
4	7:0	**Signature Byte 4** 0xDD

7.2.11 Flash Write Data - Write (E1h)

This is a bulk write command that directs the boot application to write the included 256 bytes of payload data to the flash device. The first write transaction after a flash erase will start at the first flash address (0x0). Each subsequent command will write a page of flash data and the flash address will be incremented to the next page. If the amount of data is less than 256 bytes, the host should fill the remaining bytes with 0xFF to pad.

The *Request in Progress* bit of the Short Status will be set at the start of the flash write process and will be cleared when each write transaction is complete. The next flash write should not begin until the *Request in Progress* bit is cleared.

More information on flash programming can be found in Section 5.3.3.

Write Parameters

Table 7-15. Flash Write Data Write Parameters (Boot)

BYTE	BITS	DESCRIPTION
1	7:0	**Data Byte 1**
2	7:0	**Data Byte 2**
...	...	**Data Byte ...**
256	7:0	**Data Byte 256**

7.2.12 Flash Verify Data - Write (E2h)

This command tells the boot application to verify the contents of the flash device.

The *Request in Progress* bit of the Short Status will be set at the start of the flash verify process and will be cleared when the verify process is complete.

Once the process is completed, the error bits in the Short Status can be used to determine whether an error occurred during the verification process.

More information on flash programming can be found in Section 5.3.3.

Write Parameters

No write parameters.

7.2.13 Flash Interface Rate - Write (E3h)

This command is used to specify the maximum supported flash clock rate and to indicate the supported memory read commands. Typically, this information would be stored in the flash information, but in situations when the flash is corrupt or empty, the values are not accessible from flash data. In these situations, this command can significantly reduce flash programming and verification time.

There is a minimum interface bandwidth requirement that is specified in the DLPC230-Q1 Data Sheet. Developers should ensure that the flash device selected can meet these minimum requirements.

Write Parameters

Table 7-16. Flash Interface Rate Write Parameters (Boot)

BYTE	BITS	DESCRIPTION
1	7:5	**Reserved** Set to 0x0
	4	**Quad Input / Output Read** 0x0: Not supported 0x1: Supported
	3	**Quad Output Read** 0x0: Not supported 0x1: Supported
	2	**Dual Input / Output Read** 0x0: Not supported 0x1: Supported
	1	**Dual Output Read** 0x0: Not supported 0x: Supported
	0	**Fast Read** 0x0: Not supported 0x1: Supported
3:2	15:0	**Maximum Flash Clock Rate** LSByte = Byte 2 The MHz frequency with two decimals of precision should be multiplied by 100. For example, 50.25 MHz * 100 = 5025 = 0x13A1.

Because flash instruction names vary between manufacturers, Table 7-17 defines the corresponding flash op-codes for each read instruction.

Table 7-17. Flash Read Instruction Op-Codes

FLASH INSTRUCTION NAME	FLASH OP-CODE
Fast Read	0x0B
Dual Output Read	0x3B
Dual Input / Output Read	0xBB
Quad Output Read	0x6B
Quad Input / Output Read	0xEB

7.2.14 Flash Interface Rate - Read (E4h)

This command is used to read the flash interface rate settings.

Command Parameters

No command parameters.

Return Parameters

Table 7-18. Flash Interface Rate Return Parameters (Boot)

BYTE	BITS	DESCRIPTION
1	7:5	**Reserved** Always 0x0
	4	**Quad Input / Output Read** 0x0: Not supported 0x1: Supported
	3	**Quad Output Read** 0x0: Not supported 0x1: Supported
	2	**Dual Input / Output Read** 0x0: Not supported 0x1: Supported
	1	**Dual Output Read** 0x0: Not supported 0x: Supported
	0	**Fast Read** 0x0: Not supported 0x1: Supported
3:2	15:0	**Maximum Flash Clock Rate** LSByte = Byte 2 This value should be divided by 100 to read the MHz value. For example, 0x13A1 = 5025 / 100 = 50.25 MHz.

Chapter 8

Commands - Main Application

8.1 Mode Availability

Table 8-1 summarizes the main application commands and indicates the operating modes in which they are available. Embedded software will flag an error if a command is used outside of the proper operating mode.

Mode key:

- **Yes** - Command is available and values will be applied or retrieved as soon as possible
- **No** - Command is not available and an error will be provided if the command is used
- **Hold Mode** - Command values will be held, but not applied until the operating mode is changed to a mode that says "Yes". This can be used to configure certain system parameters before switching modes and applying them.
- **Hold Source** - Command values will be held until the source is changed. This is done to coordinate the application of all source parameters.

Read commands always return the last values that were written, or the values stored in flash if no values have been written since startup. Because of data holds, this value may be different than what is actually being displayed. For example if a new flip value is written to *Display Image Orientation*, but no source change has been commanded, then the value that is read back will not match what is currently being applied. The *Current Source Information* and *Current Display Information* commands allow the host to read back the settings that are currently applied to the displayed image.

Table 8-1. Command Summary - Main Application

COMMAND	TYPE	OP CODE	STANDBY	DISPLAY	CALIBRATION (Not applicable for headlight)	SECTION
System Reset	Write	00h	Yes	Yes	Yes	Section 8.2.1
Read Pre-fetch	Write	01h	Yes	Yes	Yes	Section 8.2.2
Read Activate	Read	02h	Yes	Yes	Yes	Section 8.2.3
Operating Mode	Write	03h	Yes	Yes	Yes	Section 8.2.4
Operating Mode	Read	04h	Yes	Yes	Yes	Section 8.2.5
Source Select	Write	05h	Hold Mode	Yes	Yes	Section 8.2.6
Source Select	Read	06h	Yes	Yes	Yes	Section 8.2.7
Prepare for Source Change	Write	07h	No	Yes	Yes	Section 8.2.8
Display Image Orientation	Write	18h	Hold Mode	Hold Source	Hold Source	Section 8.2.9
Display Image Orientation	Read	19h	Yes	Yes	Yes	Section 8.2.10
System Mode Select	Write	1Ch	Hold Mode	Hold Source	Hold Source	Section 8.2.11
System Mode Select	Read	1Dh	Yes	Yes	Yes	Section 8.2.12
Execute Batch Command Set	Write	21h	Yes	Yes	Yes	Section 8.2.13
GPIO Configure	Write	23h	Yes	Yes	Yes	Section 8.2.15
GPIO Configure	Read	24h	Yes	Yes	Yes	Section 8.2.16
GPIO Outputs	Write	25h	Yes	Yes	Yes	Section 8.2.17
GPIO Outputs	Read	26h	Yes	Yes	Yes	Section 8.2.18
GPIO Reserved	Read	27h	Yes	Yes	Yes	Section 8.2.19

Table 8-1. Command Summary - Main Application (continued)

COMMAND	TYPE	OP CODE	STANDBY	DISPLAY	CALIBRATION (Not applicable for headlight)	SECTION
Execute Non-Periodic BIST	Write	28h	Yes	No	No	Section 8.2.20
External Video Checksum Control	Write	29h	Hold Mode	Yes	Yes	Section 8.2.21
External Video Checksum Control	Read	2Ah	Yes	Yes	Yes	Section 8.2.22
External Video Checksum Settings	Write	2Bh	Hold Mode	Yes	Yes	Section 8.2.23
External Video Checksum Settings	Read	2Ch	Yes	Yes	Yes	Section 8.2.24
DMD Socket Connectivity Test	Write	2Dh	Yes	No	No	Section 8.2.25
DMD Socket Connectivity Test	Read	2Eh	Yes	No	No	Section 8.2.25
Average Picture Level Control	Write	2Fh	Hold Mode	Yes	Yes	Section 8.2.27
Loss of /Ping Control	Write	33h	Hold Mode Yes	Yes	Yes	Section 8.2.29
PWM Temperature Management Enable	Write	35h	Yes	Yes	Yes	Section 8.2.31
PWM Temperature Management Enable	Read	36h	Yes	Yes	Yes	Section 8.2.32
PWM Temperature Management Source	Write	37h	Yes	Yes	Yes	Section 8.2.33
PWM Temperature Management Source	Read	38h	Yes	Yes	Yes	Section 8.2.34
PWM Temperature Management Duty Cycle	Read	39h	Yes	Yes	Yes	Section 8.2.35
Headlight Ping	Write	46h	Yes	Yes	Yes	Section 8.2.36
PWM Control	Write	47h	Yes	Yes	Yes	Section 8.2.37
PWM Control	Read	48h	Yes	Yes	Yes	Section 8.2.38
Illumination Transition Rate	Write	49h	Hold Mode	Yes	Yes	Section 8.2.39
Contrast	Read	53h	Yes	Yes	Yes	
De-gamma Select	Write	54h	Hold Mode	Yes	Yes	Section 8.2.41
Compensation Temp Host Override	Read	5Bh	Yes	Yes	Yes	
Unused		*5F - 62h*				
ADC Single Measurement	Read	63h	Yes	No	No	Section 8.2.44
Illumination Bin Select	Write	70h	Hold Mode	Yes	Yes	Section 8.2.45
TPS99000-Q1 RGB Blanking Levels	Read	85h	Yes	Yes	Yes	
TPS99000-Q1 TIA1 Trims	Write	86h	No	Yes	Yes	Section 8.2.47
TPS99000-Q1 TIA1 Trims	Read	87h	Yes	Yes	Yes	Section 8.2.48
TPS99000-Q1 TIA1 Gain	Write	88h	No	Yes	Yes	Section 8.2.49
TPS99000-Q1 TIA1 Gain	Read	89h	Yes	Yes	Yes	Section 8.2.50
TPS99000-Q1 TIA1 Capacitance	Write	8Ah	No	Yes	Yes	Section 8.2.51
TPS99000-Q1 TIA1 Capacitance	Read	8Bh	Yes	Yes	Yes	Section 8.2.52
TPS99000-Q1 TIA1 Dark Offsets	Write	8Ch	No	Yes	Yes	Section 8.2.53
TPS99000-Q1 TIA1 Dark Offsets	Read	8Dh	Yes	Yes	Yes	Section 8.2.54
TPS99000-Q1 TIA1 Input Offsets	Write	8Eh	No	Yes	Yes	Section 8.2.55
TPS99000-Q1 DM Pulse Widths	Read	91h	Yes	Yes	Yes	
TPS99000-Q1 Drive Mode	Write	92h	No	Yes	Yes	

Table 8-1. Command Summary - Main Application (continued)

COMMAND	TYPE	OP CODE	STANDBY	DISPLAY	CALIBRATION (Not applicable for headlight)	SECTION
TPS99000-Q1 Drive Mode	Read	93h	Yes	Yes	Yes	Section 8.2.57
TPS99000-Q1 ADC Configuration	Write	94h	No	Yes	Yes	Section 8.2.58
TPS99000-Q1 ADC Configuration	Read	95h	Yes	Yes	Yes	Section 8.2.59
TPS99000-Q1 Illumination Sync Control	Write	96h	No	Yes	Yes	Section 8.2.60
TPS99000-Q1 Illumination Sync Control	Read	97h	Yes	Yes	Yes	Section 8.2.61
TPS99000-Q1 TIA2 Control	Write	98h	Yes	Yes	Yes	Section 8.2.62
TPS99000-Q1 TIA2 Control	Read	99h	Yes	Yes	Yes	Section 8.2.63
LED Drive Errors	Read	9Ah	No	Yes	Yes	Section 8.2.64
LED Drive Errors Clear	Write	9Bh	No	Yes	Yes	Section 8.2.65
TPS99000-Q1 Test Mux Select	Write	9Ch	Yes	Yes	Yes	Section 8.2.66
TPS99000-Q1 TIA1 Offsets	Read	9Fh	Yes	Yes	Yes	
Flash Data Type Select	Write	A0h	Yes	No	No	Section 8.2.68
Flash Erase Data	Write	A1h	Yes	No	No	Section 8.2.69
Flash Write Data	Write (Bulk)	A2h	Yes	No	No	Section 8.2.70
Flash Read Data	Read	A3h	Yes	Yes	Yes	Section 8.2.71
Flash Verify Data	Read	A4h	Yes	No	No	Section 8.2.72
Flash Block Count	Read	A5h	Yes	Yes	Yes	Section 8.2.73
Flash Block CRCs	Read	A6h	Yes	Yes	Yes	Section 8.2.74
Unused		*AA - AFh*				
System Software Version	Read	B0h	Yes	Yes	Yes	Section 8.2.77
Flash Device ID	Read	B1h	Yes	Yes	Yes	Section 8.2.78
DLPC230-Q1 Device ID	Read	B2h	Yes	Yes	Yes	Section 8.2.79
DMD Device ID	Read	B3h	Yes	Yes	Yes	Section 8.2.80
TPS99000-Q1 Device ID	Read	B4h	Yes	Yes	Yes	Section 8.2.81
System Temperatures	Read	B5h	Yes	Yes	Yes	Section 8.2.82
System Information	Read	BAh	Yes	Yes	Yes	Section 8.2.85
Unused		*BC - BFh*				
Short Status	Read	C0h	Yes	Yes	Yes	Section 8.2.87
Error History	Read	C1h	Yes	Yes	Yes	Section 8.2.88
Clear Short Status Errors	Write	C2h	Yes	Yes	Yes	Section 8.2.89
Unused		*F4 - FFh*				

8.2 Command Definitions

8.2.1 System Reset - Write (00h)

This command is used to reset the DLPC230-Q1. The proper byte signature must be transmitted to initiate the reset in order to prevent unintended system resets.

When the command is received, the DLPC230-Q1 will perform a full reset except for the main PLL and the diagnostic port memory. This will not reset the DMD or the TPS99000-Q1. In order to reset the entire chipset, the external *PROJ_ON* signal should be used.

Write Parameters

Table 8-2. System Reset Write Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**Signature Byte 1** 0xAA
2	7:0	**Signature Byte 2** 0xBB
3	7:0	**Signature Byte 3** 0xCC
4	7:0	**Signature Byte 4** 0xDD

8.2.2 Read Pre-Fetch - Write (01h)

This command is used to send the desired read command op-code and associated command parameters to initiate a read request from embedded software. The desired read data will not be returned during this command transaction. More information on the read procedure can be found in Section 3.5.

Write Parameters

Table 8-3. Read Pre-Fetch Write Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**Read command op-code**
2 ... n		**Read command parameters** Depending on the Read op-code, a specific number of command parameter bytes will be expected. The required byte parameters are documented with each read command.

8.2.3 Read Activate (02h)

This command is used to activate the read operation in order to retrieve the previously requested read data. This generic command is used to retrieve all requested read data with the exception of the *Short Status* command. More information on the read procedure can be found in Section 3.5.

Command Parameters

No command parameters.

Return Parameters

Table 8-4. Read Activate Return Parameters

BYTE	BITS	DESCRIPTION
1 ... n		**Data Bytes 1...n** Read data bytes. The number of bytes will vary depending on the command that is currently being read.

8.2.4 Operating Mode - Write (03h)

This command is used to set the embedded software operating mode. Refer to Section 4.1 for descriptions of operating modes.

Write Parameters

Table 8-5. Operating Mode Write Parameters

BYTE	BITS	DESCRIPTION
1	7:2	***Reserved***
	1:0	**Operating Mode** 0x0: Standby 0x1: Display 0x3: *Reserved*

8.2.5 Operating Mode - Read (04h)

This command is used to read the embedded software operating mode. Refer to Section 4.1 for descriptions of operating modes.

The *Short Status* command can also be used to read the current operating mode. Using the Short Status is faster than using this command since the Short Status can be read in one transaction.

Command Parameters

No command parameters.

Return Parameters

Table 8-6. Operating Mode Return Parameters

BYTE	BITS	DESCRIPTION
1	7:2	***Reserved***
	1:0	**Operating Mode** 0x0: Standby 0x1: Display 0x3: *Reserved*

8.2.6 Source Select - Write (05h)

This command is used to select the display source. Refer to Section 5.1 for instructions on source configuration.

Write Parameters

Table 8-7. Source Select Write Parameters

BYTE	BITS	DESCRIPTION
1	7:2	***Reserved***
	1:0	**Display Source Type** 0x0: External Video 0x1: Test Pattern Generator 0x2: Splash Screen 0x3: *Reserved*

8.2.7 Source Select - Read (06h)

This command is used to read the host-specified value for the display source type.

Command Parameters

No command parameters.

Return Parameters

Table 8-8. Source Select Return Parameters

BYTE	BITS	DESCRIPTION
1	7:2	***Reserved***
	1:0	**Display Source Type** 0x0: External Video 0x1: Test Pattern Generator 0x2: Splash Screen 0x3: *Reserved*

8.2.8 Prepare for Source Change - Write (07h)

This command starts the source change process. It will disable illumination and disable source error-checking until the *Source Select* command is sent. Refer to Section 5.1 for instructions on source configuration.

Write Parameters

No write parameters.

8.2.9 Display Image Orientation - Write (18h)

This command sets the image flip orientation of the displayed image. This command is not applied until a source change is requested. Refer to Section 5.1.6 for more information on performing an image flip.

Write Parameters

Table 8-9. Display Image Orientation Write Parameters

BYTE	BITS	DESCRIPTION
1	7:2	***Reserved***
	1	**Short Axis Image Flip** 0x0: Image not flipped 0x1: Image flipped
	0	**Long Axis Image Flip** 0x0: Image not flipped 0x1: Image flipped

8.2.10 Display Image Orientation - Read (19h)

This command is used to read the host-specified values for the display image flip orientation. The default image flip settings are specified in the flash header file that is included along with the flash data.

Command Parameters

No command parameters.

Return Parameters

Table 8-10. Display Image Orientation Return Parameters

BYTE	BITS	DESCRIPTION
1	7:2	***Reserved***
	1	**Short Axis Image Flip** 0x0: Image not flipped 0x1: Image flipped
	0	**Long Axis Image Flip** 0x0: Image not flipped 0x1: Image flipped

8.2.11 System Mode Select - Write (1Ch)

This command is used to specify the System Mode index. The available System Modes are specified in the flash header file that is included along with the flash data. This command is not applied until a source change is requested. Section 10.2 provides an overview of system modes.

Write Parameters

Table 8-11. System Mode Select Write Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**System Mode Index** First index = 0

8.2.12 System Mode Select - Read (1Dh)

This command is used to read back the host-specified System Mode index.

Command Parameters

No command parameters.

Return Parameters

Table 8-12. System Mode Select Return Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**System Mode Index** First index = 0

8.2.13 Execute Batch Command Set - Write (21h)

This command is used to execute a batch command set that is stored in flash. More information of batch command sets can be found in Section 5.2.

Write Parameters

Table 8-13. Execute Batch Command Set Write Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**Batch index** Index of the batch command set stored in flash (0-based).

8.2.14 Execution Delay - Write (22h)

This command is used to specify a minimum time delay in batch command set execution or on the SPI or I2C host communication interface. This is only a minimum delay time and should not be considered an exact delay due to other ongoing software processes and the time taken to process the command.

Write Parameters

Table 8-14. Execution Delay Write Parameters

BYTE	BITS	DESCRIPTION
2:1	15:0	**Delay Time** 0x0 - 0xFFFE: Time in milliseconds (500 ms = 0x1F4) 0xFFFF: Invalid

8.2.15 GPIO Configure - Write (23h)

This command is used to configure non-reserved GPIO pins as inputs or outputs. The default GPIO configurations are stored in flash and are set by the DLPC230-Q1 main application at startup. The default GPIO configurations can be found in the flash header file that is delivered along with the flash data or by reading the GPIO Configure command during system operation.

"No change (0x0)" should be specified for all reserved GPIOs. Any other configuration for reserved GPIO will cause a command error and the values will not be set.

Refer to Section 2.2.2 for more details on GPIO usage.

Write Parameters

Table 8-15 indicates the GPIO index for each pair of configuration bits. Table 8-16 shows the configuration options available for each GPIO.

Table 8-15. GPIO Configure Write Parameters

BYTE	BIT(7:6)	BIT(5:4)	BIT(3:2)	BIT(1:0)
1	03	02	01	00
2	07	06	05	04
3	11	10	09	08
4	15	14	13	12
5	19	18	17	16
6	23	22	21	20
7	27	26	25	24
8	31	30	29	28

Table 8-16 shows the configuration options available for each non-reserved GPIO pin.

Table 8-16. GPIO Configuration Write Values

GPIO CONFIGURATION VALUES	DESCRIPTION
0x0	No change. The previous configuration of the GPIO will be retained.
0x1	Input
0x2	Output (standard)
0x3	Output (open drain)

8.2.16 GPIO Configure - Read (24h)

This command is used to read back the current drive state of the GPIO pins. The default GPIO configurations are stored in flash and are set by the DLPC230-Q1 main application at startup.

Command Parameters

No command parameters.

Return Parameters

Table 8-17 indicates the GPIO index for each pair of configuration bits. Table 8-18 shows the configuration options available for each GPIO.

Table 8-17. GPIO Configure Return Parameters

BYTE	BIT(7:6)	BIT(5:4)	BIT(3:2)	BIT(1:0)
1	03	02	01	00
2	07	06	05	04
3	11	10	09	08
4	15	14	13	12
5	19	18	17	16
6	23	22	21	20
7	27	26	25	24
8	31	30	29	28

Table 8-18. GPIO Configuration Return Values

GPIO CONFIGURATION VALUES	DESCRIPTION
0x0	Alternative function. No GPIO configuration may be applied.
0x1	Input
0x2	Output (standard)
0x3	Output (open drain)

8.2.17 GPIO Outputs - Write (25h)

This command is used to set the output values for non-reserved GPIO.

Each of the 32 GPIO pins has a mask bit (byte 4:1) and a value bit (byte 8:5). If the mask bit is set to 0x1, then the value bit will be applied. If the mask bit is not set, then the value will be ignored.

The mask should be set to 0x0 for all reserved GPIOs. Any other mask configuration for reserved GPIO will cause a command error and the values will not be set.

If a value is specified for a GPIO configured as an input, the value will not be applied until the GPIO is configured as an output.

Refer to Section 2.2.2 for more details on GPIO usage.

Write Parameters

Table 8-19 indicates the GPIO index for each write bit. Each byte is noted as either a mask byte or a value byte.

Table 8-19. GPIO Outputs Write Parameters

BYTE	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1 (Mask)	07	06	05	04	03	02	01	00
2 (Mask)	15	14	13	12	11	10	09	08
3 (Mask)	23	22	21	20	19	18	17	16
4 (Mask)	31	30	29	28	27	26	25	24
5 (Value)	07	06	05	04	03	02	01	00
6 (Value)	15	14	13	12	11	10	09	08
7 (Value)	23	22	21	20	19	18	17	16
8 (Value)	31	30	29	28	27	26	25	24

Table 8-20 defines the mask bit values.

Table 8-20. GPIO Mask Values

GPIO MASK VALUES	DESCRIPTION
0x0	Not selected - The value written will be ignored.
0x1	Selected - The value written will be used.

Table 8-21 defines the GPIO values.

Table 8-21. GPIO Output Values

GPIO VALUES	DESCRIPTION
0x0	Set low.
0x1	Set high.

8.2.18 GPIO Outputs - Read (26h)

This command is used to read back the current values of the GPIO pins.

Command Parameters

No command parameters.

Return Parameters

Table 8-22 indicates the GPIO index for each returned bit.

Table 8-22. GPIO Outputs Return Parameters

BYTE	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1 (Value)	07	06	05	04	03	02	01	00
2 (Value)	15	14	13	12	11	10	09	08
3 (Value)	23	22	21	20	19	18	17	16
4 (Value)	31	30	29	28	27	26	25	24

Table 8-23 defines the GPIO values.

Table 8-23. GPIO Values

GPIO VALUES	DESCRIPTION
0x0	Set low.
0x1	Set high.

8.2.19 GPIO Reserved - Read (27h)

This command is used to read which of the 32 GPIO pins are reserved and which are available for host use.

Refer to Section 2.2.2 for more details on GPIO usage.

Command Parameters

No command parameters.

Return Parameters

Table 8-24 indicates the GPIO index for each return bit.

Table 8-24. GPIO Reserved Return Parameters

BYTE	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	07	06	05	04	03	02	01	00
2	15	14	13	12	11	10	09	08
3	23	22	21	20	19	18	17	16
4	31	30	29	28	27	26	25	24

Table 8-25 shows the GPIO reserved state values.

Table 8-25. GPIO Reserved State Values

GPIO RESERVED STATE	DESCRIPTION
0x0	Not Reserved
0x1	Reserved

8.2.20 Execute Non-Periodic BIST - Write (28h)

This command is used to execute a selected non-periodic BIST.

Write Parameters

Table 8-26. Execute Non-Periodic BIST Write Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**Non-Periodic BIST Select** Value of the BIST to execute

Table 8-27 specifies the non-periodic BIST values.

Table 8-27. Non-Periodic BIST Values

BYTE 1 (HEX)	TEST NAME	TEST DESCRIPTION
28	DMD MBIST	DMD Memory BIST **Not supported on 0.55" ES1 DMD samples.**
40	DLPC230-Q1 Memory BISTs	Tests the DLPC230-Q1 internal memories.
41	Front End Functional BIST	Tests the DLPC230-Q1 video processing block.
42	Back End Functional BIST	Tests the DLPC230-Q1 formatting block.

8.2.21 External Video Checksum Control - Write (29h)

This command is used to enable or disable the external video checksum functions: video tell-tale checksum and video frame checksum. These features should only be enabled while external video is being displayed. They should be disabled while displaying internal test patterns or splash images. Additional details on these tests can be found in Section 6.4.1.2 and Section 6.4.1.3.

Write Parameters

Table 8-28. Execute Video Checksum Control Write Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**Enable/Disable** 0x0: Disable 0x1: Enable video tell-tale checksum 0x2: Enable video frame counter checksum 0x3: Checksum information only - Updates actual checksum value each frame, but software will never log an error while this mode is selected.

8.2.22 External Video Checksum Control - Read (2Ah)

This command is used to read back the host-specified state of the external video checksum function.

Command Parameters

No command parameters.

Return Parameters

Table 8-29. Execute Video Checksum Control Return Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**Enable/Disable** 0x0: Disabled 0x1: Video tell-tale checksum enabled 0x2: Video frame counter checksum enabled 0x3: Checksum information only

8.2.23 External Video Checksum Settings - Write (2Bh)

This command is used to specify the settings for the external video checksum functions: video tell-tale checksum and video frame counter checksum. Additional details on these tests can be found in Section 6.4.1.2 and Section 6.4.1.3.

Write Parameters

Table 8-30. External Video Checksum Settings Write Parameters

BYTE	BITS	DESCRIPTION
2:1	15:0	**Checksum Start Pixel** Starting pixel coordinate in the active video data to checksum. This is 0-based. LSByte = Byte 1
4:3	15:0	**Checksum Start Line** Starting line coordinate in the active video data to checksum. This is 0-based. LSByte = Byte 3
6:5	15:0	**Checksum Total Pixels Per Line** Number of pixels in each active line to checksum. This is 1-based. LSByte = Byte 5
8:7	15:0	**Checksum Total Lines Per Frame** Number of active lines in each frame to checksum. This is 1-based. LSByte = Byte 7
12:9	31:0	**Expected Checksum** Expected checksum value for all pixels in the specified range. This value is only used for video tell-tale checksum. It may be set to 0 for video frame counter checksum. LSByte = Byte 9

8.2.24 External Video Checksum Settings - Read (2Ch)

This command is used to read back the host-specified values for the external video checksum settings.

Command Parameters

No command parameters.

Return Parameters

Table 8-31. External Video Checksum Settings Return Parameters

BYTE	BITS	DESCRIPTION
2:1	15:0	**Checksum Start Pixel** Starting pixel coordinate in the active video data to checksum. LSByte = Byte 1
4:3	15:0	**Checksum Start Line** Starting line coordinate in the active video data to checksum. LSByte = Byte 3
6:5	15:0	**Checksum Total Pixels Per Line** Number of pixels in each active line to checksum. LSByte = Byte 5
8:7	15:0	**Checksum Total Lines Per Frame** Number of active lines in each frame to checksum. LSByte = Byte 7
12:9	31:0	**Expected Checksum** Expected checksum value for all pixels in the specified range. LSByte = Byte 9

Table 8-31. External Video Checksum Settings Return Parameters (continued)

BYTE	BITS	DESCRIPTION
16:13	31:0	**Actual Checksum** Latest measured checksum of input video data. This can be used to compare against the input expected checksum. LSByte = Byte 13

8.2.25 DMD Socket Connectivity Test - Write (2Dh)

This command can enable or disable each pin or differential pin pairs for the high speed interfaces, low speed interface, DEN_ARSTZ, and DMD reset voltages.

Write Parameters

Table 8-32. Socket Connectivity Test Write Parameters

BYTE	BITS	DESCRIPTION
3	7:5	Reserved
	4	**RST Voltages** Enable or disable DMD reset voltages 0x0: Disable 0x1: Enable
	3	Reserved
	2	**DEN_ARSTZ** Enable or disable DEN_ARSTZ 0x0: Disable 0x1: Enable
	1	**LS TX Pair** Sets LS TX Pair High (Enable) or Low (Disable) 0x0: Disable 0x1: Enable
	0	**LS CLK** Sets LS CLK High (Enable) or Low (Disable) 0x0: Disable 0x1: Enable

Table 8-32. Socket Connectivity Test Write Parameters (continued)

BYTE	BITS	DESCRIPTION
2	7:5	Reserved
	4	**CH1 CLK** Sets CH1 CLK High (Enable) or Low (Disable) 0x0: Disable 0x1: Enable
	3	**CH1 Pair 7** Sets CH1 Pair 7 High (Enable) or Low (Disable) 0x0: Disable 0x1: Enable
	2	**CH1 Pair 6** Sets CH1 Pair 6 High (Enable) or Low (Disable) 0x0: Disable 0x1: Enable
	1	**CH1 Pair 5** Sets CH1 Pair 5 High (Enable) or Low (Disable) 0x0: Disable 0x1: Enable
	0	**CH1 Pair 4** Sets CH1 Pair 4 High (Enable) or Low (Disable) 0x0: Disable 0x1: Enable
1	7	**CH1 Pair 3** Sets CH1 Pair 3 High (Enable) or Low (Disable) 0x0: Disable 0x1: Enable
	6	**CH1 Pair 2** Sets CH1 Pair 2 High (Enable) or Low (Disable) 0x0: Disable 0x1: Enable
	5	**CH1 Pair 1** Sets CH1 Pair 1 High (Enable) or Low (Disable) 0x0: Disable 0x1: Enable
	4	**CH1 Pair 0** Sets CH1 Pair 0 High (Enable) or Low (Disable) 0x0: Disable 0x1: Enable
	3:1	Reserved
	0	**CH0 CLK** Sets CH0 CLK High (Enable) or Low (Disable) 0x0: Disable 0x1: Enable

Table 8-32. Socket Connectivity Test Write Parameters (continued)

BYTE	BITS	DESCRIPTION
0	7	**CH0 Pair 7** Sets CH0 Pair 7 High (Enable) or Low (Disable) 0x0: Disable 0x1: Enable
	6	**CH0 Pair 6** Sets CH0 Pair 6 High (Enable) or Low (Disable) 0x0: Disable 0x1: Enable
	5	**CH0 Pair 5** Sets CH0 Pair 5 High (Enable) or Low (Disable) 0x0: Disable 0x1: Enable
	4	**CH0 Pair 4** Sets CH0 Pair 4 High (Enable) or Low (Disable) 0x0: Disable 0x1: Enable
	3	**CH0 Pair 3** Sets CH0 Pair 3 High (Enable) or Low (Disable) 0x0: Disable 0x1: Enable
	2	**CH0 Pair 2** Sets CH0 Pair 2 High (Enable) or Low (Disable) 0x0: Disable 0x1: Enable
	1	**CH0 Pair 1** Sets CH0 Pair 1 High (Enable) or Low (Disable) 0x0: Disable 0x1: Enable
	0	**CH0 Pair 0** Sets CH0 Pair 0 High (Enable) or Low (Disable) 0x0: Disable 0x1: Enable

8.2.26 DMD Socket Connectivity Test - Read (2Eh)

This command can is used to read the state of each pin, differential pin pair, DEN_ARSTZ, and DMD reset voltages set by the Socket Connectivity Test Write command.

Command Parameters

No command parameters

Return Parameters

Table 8-33. DMD Socket Connectivity Test Return Parameters

BYTE	BITS	DESCRIPTION
3	7:5	Reserved
	4	**RST Voltages** DMD reset voltages state 0x0: Disabled 0x1: Enabled
	3	Reserved
	2	**DEN_ARSTZ** DEN_ARSTZ state 0x0: Disabled 0x1: Enabled
	1	**LS TX Pair** LS TX Pair state 0x0: Disabled (Low) 0x1: Enabled (High)
	0	**LS CLK** LS CLK state 0x0: Disabled (Low) 0x1: Enabled (High)
2	7:5	Reserved
	4	**CH1 CLK** CH1 CLK state 0x0: Disabled (Low) 0x1: Enabled (High)
	3	**CH1 Pair 7** CH1 Pair 7 state 0x0: Disabled (Low) 0x1: Enabled (High)
	2	**CH1 Pair 6** CH1 Pair 6 state 0x0: Disabled (Low) 0x1: Enabled (High)
	1	**CH1 Pair 5** CH1 Pair 5 state 0x0: Disabled (Low) 0x1: Enabled (High)
	0	**CH1 Pair 4** CH1 Pair 4 state 0x0: Disabled (Low) 0x1: Enabled (High)

Table 8-33. DMD Socket Connectivity Test Return Parameters (continued)

BYTE	BITS	DESCRIPTION
1	7	**CH1 Pair 3** CH1 Pair 2 state 0x0: Disabled (Low) 0x1: Enabled (High)
	6	**CH1 Pair 2** CH1 Pair 2 state 0x0: Disabled (Low) 0x1: Enabled (High)
	5	**CH1 Pair 1** CH1 Pair 1 state 0x0: Disabled (Low) 0x1: Enabled (High)
	4	**CH1 Pair 0** CH1 Pair 0 state 0x0: Disabled (Low) 0x1: Enabled (High)
	3:1	Reserved
	0	**CH0 CLK** CH0 CLK state 0x0: Disabled (Low) 0x1: Enabled (High)

Table 8-33. DMD Socket Connectivity Test Return Parameters (continued)

BYTE	BITS	DESCRIPTION
0	7	**CH0 Pair 7** CH0 Pair 7 state 0x0: Disabled (Low) 0x1: Enabled (High)
	6	**CH0 Pair 6** CH0 Pair 6 state 0x0: Disabled (Low) 0x1: Enabled (High)
	5	**CH0 Pair 5** CH0 Pair 5 state 0x0: Disabled (Low) 0x1: Enabled (High)
	4	**CH0 Pair 4** CH0 Pair 4 state 0x0: Disabled (Low) 0x1: Enabled (High)
	3	**CH0 Pair 3** CH0 Pair 3 state 0x0: Disabled (Low) 0x1: Enabled (High)
	2	**CH0 Pair 2** CH0 Pair 2 state 0x0: Disabled (Low) 0x1: Enabled (High)
	1	**CH0 Pair 1** CH0 Pair 0 state 0x0: Disabled (Low) 0x1: Enabled (High)
	0	**CH0 Pair 0** CH0 Pair 0 state 0x0: Disabled (Low) 0x1: Enabled (High)

8.2.27 Average Picture Level Control - Write (2Fh)

This command is used to control the average picture level function. Additional details on this test can be found in Section 6.4.1.4

Write Parameters

Table 8-34. Average Picture Level Control Write Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**APL Control** 0x0: Disable 0x1: Enable - Log error if threshold is exceeded 0x2: Enable - Log error and execute emergency shutdown if threshold is exceeded
2	7:0	**APL Threshold** Threshold picture level. If the incoming video exceeds this threshold, a failure action will be taken as specified in the control bits.

8.2.28 Average Picture Level Control - Read (30h)

This command is used to read back the host-specified values for the average picture level function. It also returns the most recent average picture level value from the function.

Command Parameters

No command parameters.

Return Parameters

Table 8-35. Average Picture Level Control Return Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**APL Control** 0x0: Disable 0x1: Enable - Log error if threshold is exceeded 0x2: Enable - Log error and execute emergency shutdown if threshold is exceeded
2	7:0	**APL Threshold** Host-specified threshold picture level.
3	7:0	**APL Value** Most recent average picture level value.

8.2.29 Loss Of Ping Control - Write (33h)

This command is used to enable the loss-of-ping-command test and to specify the maximum allowable time between ping commands. The default value for this function is specified in flash data.

Section 6.4.1.5 describes the loss-of-ping test in more detail.

Write Parameters

Table 8-36. Loss of Ping Control Write Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**Timeout length** Maximum allowed time in milliseconds between ping command transmissions. 0 = Disable For example, 16 = host must continuously send a command within 16-ms time frame.

8.2.30 Loss Of Ping Control - Read (34h)

This command is used to read back the host-specified values for the loss-of-ping-command function.

Command Parameters

No command parameters.

Return Parameters

Table 8-37. Loss of Ping Control Return Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**Timeout length** Maximum allowed time in milliseconds between ping command transmissions. 0 = Disable

8.2.31 PWM Temperature Management Enable - Write (35h)

This command enables or disables the PWM temperature management function.

See section Section 5.6.2 for use case information.

Write Parameters

Table 8-38. PWM Temperature Management Enable

BYTE	BITS	DESCRIPTION
1	7:0	0x0: Disable 0x1: Enable

8.2.32 PWM Temperature Management Enable - Read (36h)

This command reads the status of PWM temperature management enable.

Command Parameters

No command parameters.

Return Parameters

Table 8-39. PWM Temperature Management Enable

BYTE	BITS	DESCRIPTION
1	7:0	0x0: Disabled 0x1: Enabled

8.2.33 PWM Temperature Management Source - Write (37h)

This command selects the source of the PWM temperature management function.

Write Parameters

Table 8-40. PWM Temperature Management Source

BYTE	BITS	DESCRIPTION
1	7:0	0x0: Sets source from the raw DMD temperature 0x1: Sets source from the filtered DMD temperature 0x2: Sets source from the raw local temperature 0x3: Sets source from the filtered local temperature

8.2.34 PWM Temperature Management Source - Read (38h)

This command reads the PWM temperature management source selected.

Command Parameters

No command parameters.

Return Parameters

Table 8-41. PWM Temperature Management Source

BYTE	BITS	DESCRIPTION
1	7:0	0x0: source from the raw DMD temperature 0x1: source from the filtered DMD temperature 0x2: source from the raw local temperature 0x3: source from the filtered local temperature

8.2.35 PWM Temperature Management Duty Cycle - Read (39h)

This command returns current duty cycle of the PWM signal used for the PWM temperature management function.

Command Parameters

No command parameters.

Return Parameters

Table 8-42. PWM Temperature Management Duty Cycle

BYTE	BITS	DESCRIPTION
1	7:0	Returns PWM temperature management duty cycle. Value returned is 0-100 for 0% to 100%. For example, 0x1 is 1% and 0x63 is 99%.

8.2.36 Headlight Ping - Write (46h)

This command is used to fulfill the loss-of-ping timeout requirement within the specified timeout window. It does not have any functional purpose other than notifying the DLPC230-Q1 Main Application that the host is still active and able to communicate.

Write Parameters

No write parameters.

8.2.37 PWM Control - Write (47h)

This command is used to control DLPC230-Q1 PWM duty cycles. The default values are specified in flash data. Section 2.2.3 describes the PWM signals. Note, enabling spread spectrum will affect the PWM frequency, which might affect the PWM duty cycle.

Write Parameters

Table 8-43. PWM Control Write Parameters

BYTE	BITS	DESCRIPTION
2:1	15:0	**PWM 0 Duty Cycle** LSByte = Byte 1 0 = 0% 1024 = 100%
4:3	15:0	**PWM 1 Duty Cycle** LSByte = Byte 3 0 = 0% 1024 = 100%
6:5	15:0	**PWM 2 Duty Cycle** LSByte = Byte 5 0 = 0% 1024 = 100%

8.2.38 PWM Control - Read (48h)

This command is used to read back the host-specified PWM duty cycles.

Command Parameters

No command parameters.

Return Parameters

Table 8-44. PWM Control Read Parameters

BYTE	BITS	DESCRIPTION
2:1	15:0	**PWM 0 Duty Cycle** LSByte = Byte 1 0 = 0% 1024 = 100%
4:3	15:0	**PWM 1 Duty Cycle** LSByte = Byte 3 0 = 0% 1024 = 100%
6:5	15:0	**PWM 2 Duty Cycle** LSByte = Byte 5 0 = 0% 1024 = 100%

8.2.39 Illumination Transition Rate - Write (49h)

This command is used to specify a time in which to smoothly transition between illumination bins. This function is described in Section 5.5.

Write Parameters

Table 8-45. Illumination Transition Rate Write Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**Transition time** Number of frames = Value * 8 0x0 = Immediate transition 0x1 = Transition over 8 frames 0xFF = Transition over 2040 frames

8.2.40 Illumination Transition Rate - Read (4Ah)

This command is used to specify a time in which to smoothly transition between illumination bins.

Command Parameters

No command parameters.

Return Parameters

Table 8-46. Illumination Transition Rate Return Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**Transition time** Number of frames = Value * 8

8.2.41 De-gamma Select - Write (54h)

This command is used to specify the de-gamma curve index that is applied. De-gamma curves are configurable in flash. The available de-gamma curves are specified in the flash header file that is included along with the flash data.

Write Parameters

Table 8-47. De-gamma Select Write Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**De-gamma curve Index** First index = 0

8.2.42 De-gamma Select - Read (55h)

This command is used to read back the host-specified de-gamma curve index. De-gamma curves are configurable in flash. The available de-gamma curves are specified in the flash header file that is included along with the flash data.

Command Parameters

No command parameters.

Return Parameters

Table 8-48. De-gamma Select Return Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**De-gamma curve Index** First index = 0

8.2.43 ADC Measurements - Read (5Ch)

This command is used to read the latest ADC measurements captured by the TPS99000-Q1 ADC block and reported to the DLPC230-Q1. There are a total of 63 ADC measurement indexes that can be captured each frame. The definition of each ADC index is defined by flash data and this information can be found in the flash header file.

The returned ADC measurements are converted and scaled voltage values in two's complement integer format. The measurement values from all TPS99000-Q1 channels except for channel 0 are represented on a 1-mV scale. TPS99000-Q1 channel 0 is represented on a 10-µV scale due to its higher gain.

Note: TPS99000-Q1 channel 0 is not the same as index 0 for this command. The mapping of these two values can be found in the flash header file. Refer to *Section 5.7.1* for an example of this mapping.

Command Parameters

Table 8-49. ADC Measurements Command Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**Start index** Index of first ADC measurement to be returned. 0 - 62
2	7:0	**Measurement Count** Number of measurements to be returned 1 - 63 Start Index + Measurement Count must be less than or equal to 63.

Return Parameters

Table 8-50. ADC Measurements Return Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**Frame Count** Incrementing count to allow the host to check the update rate.
3:2	15:0	**ADC Measurement 1** First returned ADC measurement. The index of this measurement is specified by the Start Index command parameter. Two's complement integer format. All TPS99000-Q1 ADC channels will be reported on a 1-mV scale, except for channel 0 which is reported on a 10-µV scale. LSByte = 2

Table 8-50. ADC Measurements Return Parameters (continued)

BYTE	BITS	DESCRIPTION
4	7:5	***Reserved***
	4	**Measurement Index Not Used** 0x1: ADC measurement index not used. Measurement is not assigned to an ADC channel. Other fields for this measurement, including value and error bits, should be ignored. 0x0: ADC measurement index is used.
	3:0	**ADC Measurement 1 Error Bits** b1xxx = Parity error for this measurement. All other bits and measurement value should be considered invalid. b0001 = ADC value saturated at maximum value b0010 = ADC value underflow at minimum value b0011 = ADC timing error (measurements were triggered too close together) b0100 = Previous measurement mismatch error b0101 = Previous measurement parity error b0110 = Previous measurement stop bit error b0111 = Input ADC received mismatch error
...	...	**ADC Measurement ...**
n*3 : n*3 - 1	15:0	**ADC Measurement n** Final ADC measurement. "n" is specified by Measurement Count command parameter.
n*3 + 1	7:4	*Reserved*
	3:0	**ADC Measurement n Error Bits** Final ADC measurement error bits.

8.2.44 ADC Single Measurement - Read (63h)

This command is used to read a single channel of the TPS99000-Q1 ADC. This cannot be performed while video is displaying. While video is displaying, ADC capture is specified by a sequence-aligned look-up table each frame.

The returned ADC measurement is a converted and scaled voltage value in two's complement integer format. The measurement values from all TPS99000-Q1 channels except for channel 0 are represented on a 1mV scale. Channel 0 is represented on a 10µV scale due to its higher gain.

Command Parameters

Table 8-51. ADC Single Measurement Command Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**TPS99000-Q1 ADC Channel** 0 - 31

Return Parameters

Table 8-52. ADC Single Measurement Return Parameters

BYTE	BITS	DESCRIPTION
2:1	15:0	**ADC Measurement** LSByte = Byte 1 Two's complement integer format. All TPS99000-Q1 ADC channels will be reported on a 1-mV scale, except for channel 0 which is reported on a 10-µV scale.
3	7:4	***Reserved***
	3:0	**ADC Measurement Error Bits** b1xxx = Parity error for this measurement. All other bits and measurement value should be considered invalid. b0001 = ADC value saturated at maximum value b0010 = ADC value underflow at minimum value b0011 = ADC timing error (measurements were triggered too close together) b0100 = Previous measurement mismatch error b0101 = Previous measurement parity error b0110 = Previous measurement stop bit error b0111 = Input ADC received mismatch error

8.2.45 Illumination Bin Select - Write (70h)

This command is used to specify the desired illumination bin. Section 10.2 describes the function of illumination bins.

Write Parameters

Table 8-53. Illumination Bin Select Write Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**Illumination Bin Index**

8.2.46 Illumination Bin Select - Read (71h)

This command is used to read back the last applied illumination bin.

Command Parameters

No command parameters.

Return Parameters

Table 8-54. Illumination Bin Select Return Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**Illumination Bin Index**

8.2.47 TPS99000-Q1 TIA1 Trims - Write (86h)

This command is used to set TPS99000-Q1 TIA1 RGB Trim gain multipliers. The trim values range from 1.0x to 0.2x linearly. These trims may also be auto-calculated by the DLPC230-Q1 main application.

For single-color applications, all trims should be set to the same value.

Write Parameters

Table 8-55. TPS99000-Q1 TIA1 Trims Write Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**Red Trim** Trim gain = 1 – N × (0.8 / 255) 0x0: 1.0x 0xFF: 0.2x
2	7:0	**Green Trim** Trim gain = 1 – N × (0.8 / 255) 0x0: 1.0x 0xFF: 0.2x
3	7:0	**Blue Trim** Trim gain = 1 – N × (0.8 / 255) 0x0: 1.0x 0xFF: 0.2x
4	0	**Auto-Calculate Enable** 0x0: Directly apply trim values specified in this command. 0x1: DLPC230-Q1 software will auto-calculate trim values. Other fields in this command will be ignored.

8.2.48 TPS99000-Q1 TIA1 Trims - Read (87h)

This command is used to read back the last applied TPS99000-Q1 TIA1 trim gains.

Command Parameters

No command parameters.

Return Parameters

Table 8-56. TPS99000-Q1 TIA1 Trims Return Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**Red Trim**
2	7:0	**Green Trim**
3	7:0	**Blue Trim**
4	0	**Auto-Calculate Enable**

8.2.49 TPS99000-Q1 TIA1 Gain - Write (88h)

This command is used to set the TPS99000-Q1 TIA1 gain level.

Write Parameters

Table 8-57. TPS99000-Q1 TIA1 Gain Write Parameters

BYTE	BITS	DESCRIPTION
1	3:0	**Gain Index** *Table 8-58* defines the gain level settings.

Refer to the TPS99000-Q1 Data Sheet for detail regarding gain specifications.

Table 8-58. TIA1 Gain Values

GAIN INDEX	GAIN (kV/A)(1)
0	0.75
1	1.5
2	3
3	6
4	9
5	12
6	18
7	24
8	36
9	48
10	72
11	96
12	144
13	288

(1) Trim set to 1.0.

8.2.50 TPS99000-Q1 TIA1 Gain - Read (89h)

This command is used to read back the last applied TPS99000-Q1 TIA1 gain level.

Write Parameters

Table 8-59. TPS99000-Q1 TIA1 Gain Return Parameters

BYTE	BITS	DESCRIPTION
1	3:0	**Gain Index** *Table 8-58* defines the gain level settings.

8.2.51 TPS99000-Q1 TIA1 Capacitance - Write (8Ah)

This command is used to set TPS99000-Q1 TIA1 capacitance for photodiode capacitance compensation.

Write Parameters

Table 8-60. TPS99000-Q1 TIA1 Capacitance Write Parameters

BYTE	BITS	DESCRIPTION
1	5:0	**TIA1 Capacitance** Capacitance = N × 0.5 pF 0x0: 0.0 pF 0x3F: 31.5 pF

8.2.52 TPS99000-Q1 TIA1 Capacitance - Read (8Bh)

This command is used to read back the last applied TPS99000-Q1 TIA1 capacitance.

Command Parameters

No command parameters.

Return Parameters

Table 8-61. TPS99000-Q1 TIA1 Capacitance Return Parameters

BYTE	BITS	DESCRIPTION
1	5:0	**TIA1 Capacitance**

8.2.53 TPS99000-Q1 TIA1 Dark Offsets - Write (8Ch)

This command is used to set TPS99000-Q1 TIA1 RGB offsets to compensate for photodiode dark current.

For single-color applications, all dark offsets should be set to the same value.

Write Parameters

Table 8-62. TPS99000-Q1 TIA1 Dark Offsets Write Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**Red Dark Offset** Offset = (N+1) × 1.5mV
2	7:0	**Green Dark Offset** Offset = (N+1) × 1.5mV
3	7:0	**Blue Dark Offset** Offset = (N+1) × 1.5mV

8.2.54 TPS99000-Q1 TIA1 Dark Offsets - Read (8Dh)

This command is used to read back the last applied TPS99000-Q1 TIA1 RGB dark offsets to compensate for photodiode dark current.

Command Parameters

No command parameters.

Return Parameters

Table 8-63. TPS99000-Q1 TIA1 Dark Offsets Return Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**Red Dark Offset**
2	7:0	**Green Dark Offset**
3	7:0	**Blue Dark Offset**

8.2.55 TPS99000-Q1 TIA1 Input Offsets - Write (8Eh)

This command is used to set TPS99000-Q1 TIA1 RGB input offsets.

For single-color applications, all input offsets should be set to the same value.

Write Parameters

Table 8-64. TPS99000-Q1 TIA1 Input Offsets Write Parameters

BYTE	BITS	DESCRIPTION
1	2:0	**Red Input Offset** Offset = (N+1) × 20mV 0x7 = 0mV. This value is an exclusion from the formula above and should not be used because the amplifier requires offset.
2	2:0	**Green Input Offset** Offset = (N+1) × 20mV 0x7 = 0mV. This value is an exclusion from the formula above and should not be used because the amplifier requires offset.
3	2:0	**Blue Input Offset** Offset = (N+1) × 20mV 0x7 = 0mV. This value is an exclusion from the formula above and should not be used because the amplifier requires offset.
4	2:0	**Default Input Offset** This default is used when no illumination enable is active during the sequence time. Offset = (N+1) × 20mV 0x7 = 0mV. This value is an exclusion from the formula above and should not be used because the amplifier requires offset.

8.2.56 TPS99000-Q1 TIA1 Input Offsets - Read (8Fh)

This command is used to read the last applied TPS99000-Q1 TIA1 RGB input offsets.

Command Parameters

No command parameters.

Return Parameters

Table 8-65. TPS99000-Q1 TIA1 Input Offsets Return Parameters

BYTE	BITS	DESCRIPTION
1	2:0	**Red Input Offset**
2	2:0	**Green Input Offset**
3	2:0	**Blue Input Offset**
4	2:0	**Default Input Offset**

8.2.57 TPS99000-Q1 Drive Mode - Read (93h)

This command is used to read back the last applied value of photo feedback mode, minimum COMPOUT low time, and CMODE setting.

Command Parameters

No command parameters.

Return Parameters

Table 8-66. TPS99000-Q1 Drive Mode Return Parameters

BYTE	BITS	DESCRIPTION
1	0	**Photo Feedback Mode Enabled**
2	7:0	**COMPOUT Low**
3	0	**CMODE Enable**

8.2.58 TPS99000-Q1 ADC Configuration - Write (94h)

This command is used to configure ADC selections for the TIA1 Stage Monitor and low-side current sense gain in the TPS99000-Q1.

Write Parameters

Table 8-67. TPS99000-Q1 ADC Configuration Write Parameters

BYTE	BITS	DESCRIPTION
1	7:4	***Reserved***
	3:2	**Low-side current sense gain** Gain for ADC0 input. 0x0: 24x 0x1: 12x 0x2: 9x 0x3: 24x
	1:0	**TIA1 Stage Monitor Select** Voltage monitor point within TIA1. Input to ADC30. 0x0: Stage A (Input offset output, Trim input) 0x1: Stage B (Trim output, Dark offset input) 0x2: Stage C (Gain #2 output, Gain #3 input) 0x3: Stage D (Gain #3 output, TIA output)

8.2.59 TPS99000-Q1 ADC Configuration - Read (95h)

This command is used to read back last applied ADC configuration settings.

Command Parameters

No command parameters.

Return Parameters

Table 8-68. TPS99000-Q1 ADC Configuration Return Parameters

BYTE	BITS	DESCRIPTION
1	7:4	***Reserved***
	3:2	**Low-side current sense gain**
	1:0	**TIA1 Stage Monitor Select**

8.2.60 TPS99000-Q1 Illumination Sync Control - Write (96h)

This command is used to configure TPS99000-Q1 illumination synchronization.

Write Parameters

Table 8-69. TPS99000-Q1 Illumination Sync Control Write Parameters

BYTE	BITS	DESCRIPTION
1	3:0	**DRV_EN Low Time** DRV_EN low time = (N+1) * (SEQ_CLK Period) 0x0: 1 SEQ_CLK Period 0x15: 16 SEQ_CLK Period
2	3:0	**SYNC Pulse Width** SYNC clock pulse high width SYNC Pulse Width = N * (SEQ_CLK Period) 0x0: SYNC pin low 0xF: 15 * SEQ_CLK Period
3	7:0	**SYNC Period** SYNC clock period 0x0: SYNC pin low 0x1 - 0x8: **Invalid, do not use.** 0x9 - 0xFF: SYNC Period = (N+1) * (SEQ_CLK Period)

8.2.61 TPS99000-Q1 Illumination Sync Control - Read (97h)

This command is used to read back the last applied values for TPS99000-Q1 illumination synchronization.

Write Parameters

Table 8-70. TPS99000-Q1 Illumination Sync Control Read Parameters

BYTE	BITS	DESCRIPTION
1	3:0	**DRV_EN Low Time**
2	3:0	**Sync Pulse Width**
3	7:0	**Sync Period**

8.2.62 TPS99000-Q1 TIA2 Control - Write (98h)

This command is used to configure TPS99000-Q1 transimpedance amplifier 2 (TIA2).

Write Parameters

Table 8-71. TPS99000-Q1 TIA2 Control Write Parameters

BYTE	BITS	DESCRIPTION
1	0	**Enable** Enables TIA2. When disabled, TIA2 enters a low power state. 0x0: Disable 0x1: Enable
2	3:0	**Gain Index** *Table 8-72* defines the gain level settings.
3	7:0	**Trim** Trim gain = 1 – N * (0.8 / 255) 0x0: 1.0x 0xFF: 0.2x
4	2:0	**Input Offset** Offset = (N+1) * 20mV 0x7 = 0mV. This value is an exclusion from the formula above and should not be used because the amplifier requires offset.
5	7:0	**Dark Offset** Offset = N * 1.5mV
6	5:0	**Capacitance** Capacitance = N * 0.5pF 0x0: 0.0 pF 0x3F: 31.5 pF
7	1:0	**Stage Monitor Select** Voltage monitor point within TIA2. Input to ADC31. 0x0: Stage A (Input offset output, Trim input) 0x1: Stage B (Trim output, Dark offset input) 0x2: Stage C (Gain #2 output, Gain #3 input) 0x3: Stage D (Gain #3 output, TIA output)

Table 8-72. TIA2 Gain Values

GAIN INDEX	GAIN (kV/A)(1)
0	0.75
1	1.5
2	3
3	6
4	9
5	12
6	18
7	24
8	36
9	48
10	72
11	96
12	144
13	288

(1) Trim set to 1.0.

8.2.63 TPS99000-Q1 TIA2 Control - Read (99h)

This command is used to read back the last applied configuration of TPS99000-Q1 transimpedance amplifier 2 (TIA2).

Command Parameters

No command parameters.

Return Parameters

Table 8-73. TPS99000-Q1 TIA2 Control Return Parameters

BYTE	BITS	DESCRIPTION
1	0	**Enable**
2	3:0	**Gain Index**
3	7:0	**Trim**
4	2:0	**Input Offset**
5	7:0	**Dark Offset**
6	5:0	**Capacitance**
7	1:0	**Stage Monitor Select**

8.2.64 LED Drive Errors - Read (9Ah)

This command is used to read back the current illumination pulse error information while operating in discontinuous mode.

Command Parameters

No command parameters.

Return Parameters

Table 8-74. LED Drive Errors Return Parameters

BYTE	BITS	DESCRIPTION
1	7:3	***Reserved***
	2	**Pulse Width Error Flag** Indicates that the SEN signal was held low longer than the maximum allowable time for a single pulse and was therefore forced high. This typically indicates that a light pulse was unable to achieve the desired photo feedback level within the allocated pulse time. 0x0: No error 0x1: Error
	1	**Pulse Count Error Flag** Indicates that too few pulses were generated within a bit slice. 0x0: No error 0x1: Error
	0	**No Compare Error Flag** Indicates that COMPOUT never went low within a bit slice. 0x0: No error 0x1: Error

8.2.65 LED Drive Errors Clear - Write (9Bh)

This command is used to clear illumination pulse error flags. Writing a 1 to any bit will clear the specified flag.

Write Parameters

Table 8-75. LED Drive Errors Clear Write Parameters

BYTE	BITS	DESCRIPTION
1	7:3	***Reserved***
	2	**Pulse Width Error Clear Mask** 0x0: Do not clear 0x1: Clear error flag to 0
	1	**Pulse Count Error Clear Mask** 0x0: Do not clear 0x1: Clear error flag to 0
	0	**No Compare Error Clear Mask** 0x0: Do not clear 0x1: Clear error flag to 0

8.2.66 TPS99000-Q1 Test Mux Select - Write (9Ch)

This command is used to select TPS99000-Q1 test mux output signals.

Write Parameters

Table 8-76. TPS99000-Q1 Test Mux Select Write Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**AMUX0 Select** 0x0: TIA1 Output Voltage 0x1: TIA1 Filtered Output Voltage 0x2: TIA2 Output Voltage 0x3: TIA2 Filtered Output Voltage 0x1B: Current sense amplifier output
2	7:0	**AMUX1 Select** 0x0: Photo Feedback DAC 0x1: Current Feedback DAC 0x3: TIA1 Input Offset 0x4: TIA2 Input Offset 0xC: VMAIN
3	7:0	**DMUX0 Select** 0x0: DLPC230-Q1 to TPS99000-Q1 SPI Parity Error 0x1C: 3.3V Monitor Power Good 0x1E: 1.8V Monitor Power Good 0x20: 1.1V Monitor Power Good 0x40: Over-brightness Fault 0x79: 2MHz Internal Oscillator 0x7A: 15MHz State Machine Clock 0x7B: 2.5MHz ADC Clock 0x7C: SPI1 Clock 0x7D: SPI2 Clock 0x7E: 30MHz Sequencer Clock 0x84: Watchdog 1 Error 0x86: Watchdog 2 Error 0x9C: ADC Interface Parity Error 0xA2: ADC Channel 1 Saturated 0xA3: ADC Channel 1 Underflow 0xAD: ADC End Of Conversion 0xB3: State Machine Change 0xD0: State Machine Index Bit 0 0xD1: State Machine Index Bit 1 0xD2: State Machine Index Bit 2 0xD3: State Machine Index Bit 3 0xD4: State Machine Index Bit 4

Table 8-76. TPS99000-Q1 Test Mux Select Write Parameters (continued)

BYTE	BITS	DESCRIPTION
4	7:0	**DMUX1 Select** 0x0: DLPC230-Q1 to TPS99000-Q1 SPI Parity Error 0x1C: 3.3V Monitor Power Good 0x1E: 1.8V Monitor Power Good 0x20: 1.1V Monitor Power Good 0x40: Over-brightness Fault 0x79: 2MHz Internal Oscillator 0x7A: 15MHz State Machine Clock 0x7B: 2.5MHz ADC Clock 0x7C: SPI1 Clock 0x7D: SPI2 Clock 0x7E: 30MHz Sequencer Clock 0x84: Watchdog 1 Error 0x86: Watchdog 2 Error 0x9C: ADC Interface Parity Error 0xA2: ADC Channel 1 Saturated 0xA3: ADC Channel 1 Underflow 0xAD: ADC End Of Conversion 0xB3: State Machine Change 0xD0: State Machine Index Bit 0 0xD1: State Machine Index Bit 1 0xD2: State Machine Index Bit 2 0xD3: State Machine Index Bit 3 0xD4: State Machine Index Bit 4

8.2.67 TPS99000-Q1 Test Mux Select - Read (9Dh)

This command is used to read back the last applied TPS99000-Q1 test mux indexes.

Command Parameters

No command parameters.

Return Parameters

Table 8-77. TPS99000-Q1 Test Mux Select Return Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**AMUX0 Select**
2	7:0	**AMUX1 Select**
3	7:0	**DMUX0 Select**
4	7:0	**DMUX1 Select**

8.2.68 Flash Data Type Select - Write (A0h)

This command is used to specify the type of data that will be written to flash memory. More information on flash programming can be found in Section 5.3.

Write Parameters

Table 8-78. Flash Data Type Select Write Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**Flash Data Type** Refer to Table 8-79 for options.
5:2	31:0	**Flash Package Size** The number of bytes in the binary file that is to be written to flash.
21:6	127:0	**Data Header** The first 16 bytes of the flash package that is being written. These header bytes allow embedded software to identify and verify the data stored in the binary.

The flash data types are described in Table 8-79.

Table 8-79. Flash Data Type Values

FLASH DATA TYPE VALUE	DESCRIPTION
0x00	Entire flash
0x01	Entire flash except scratchpad data
0x30	Main Application
0x40	Configuration Data 0
0x50	Configuration Data 1
0x60	Batch command sets
0x90	Splash Data 0
0x91	Splash Data 1
0x92	Splash Data 2
0x93	Splash Data 3
0x94	Splash Data 4
0x95	Splash Data 5
0x96	Splash Data 6
0x97	Splash Data 7
0xA0	Scratchpad Data 0
0xA1	Scratchpad Data 1

Table 8-79. Flash Data Type Values (continued)

FLASH DATA TYPE VALUE	DESCRIPTION
0xA2	Scratchpad Data 2
0xA3	Scratchpad Data 3

8.2.69 Flash Erase Data - Write (A1h)

This command directs software to erase the flash data that was selected using the *Flash Data Type Select* command.

The *Request in Progress* bit of the Short Status will be set at the start of the flash erase process and will be cleared when the erase process is complete.

The *Operational Error* bit in the Short Status can be used to determine whether an error occurred during the erase process.

More information on flash programming can be found in Section 5.3.

Write Parameters

Table 8-80. Flash Erase Write Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**Signature Byte 1** 0xAA
2	7:0	**Signature Byte 2** 0xBB
3	7:0	**Signature Byte 3** 0xCC
4	7:0	**Signature Byte 4** 0xDD

8.2.70 Flash Write Data - Write (A2h)

This is a bulk write command that directs the system to write up to 256 bytes of payload data to the flash device. The number of bytes must be a multiple of 4. The first write transaction after a *Flash Erase Data* command will start at the first address of the flash data type that was selected using the *Flash Data Type Select* command. Each subsequent command will write a page of flash data and the flash address will be incremented to the next page.

Note that this command is not the same as the boot application flash write command because this command allows a variable number of bytes to be sent.

The *Request in Progress* bit of the Short Status will be set at the start of the flash write process and will be cleared when each write transaction is complete. The next flash write should not begin until the *Request in Progress* bit is cleared.

More information on flash programming can be found in Section 5.3.

Write Parameters

Table 8-81. Flash Write Data Write Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**Data Byte 1**
2	7:0	**Data Byte 2**
...	...	**Data Byte ...**
n	7:0	**Data Byte n** The number of bytes, n, must be a multiple of 4.

8.2.71 Flash Read Data - Read (A3h)

This command is used to read data from the flash device. A flash data block is selected for reading and the read will occur relative to that block's location in flash.

For most flash data types, the start address has two possible values: 0x0000 indicates start at the beginning of the selected block type. 0xFFFF indicates start where the last read operation ended for continuous reading.

For scratchpad data types, the start address can be any 32-bit aligned address (0x0, 0x4, 0x8, ...). This unique case allows for partial reading of specific regions of scratchpad data.

Command Parameters

Table 8-82. Flash Read Data Command Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**Flash Data Type** Refer to Table 8-83 for options.
3:2	15:0	**Start Address** First byte of flash memory to read Most flash data types: • 0x0000: Start at beginning of flash data type • 0x0001 - 0xFFFE: *Reserved* • 0xFFFF: Start where last read operation ended (continue reading) Scratchpad data types: Relative 32-bit aligned address (0x0, 0x4, 0x8, ...)
4	15:0	**Read Data Length** 0-based number of bytes to read (0x0 = 1 byte) Must be a multiple of 4 and the maximum is 256 bytes (0xFF)

The flash data types are described in Table 8-83.

Table 8-83. Flash Data Type Values

FLASH DATA TYPE VALUE	DESCRIPTION
0x00	Entire flash
0x01	Entire flash except scratchpad data
0x30	Main Application
0x40	Configuration Data 0
0x48	Configuration Data 0 Flash Information Field
0x50	Configuration Data 1
0x58	Configuration Data 1 Flash Information Field
0x60	Batch Command Sets
0x68	Batch Command Sets Flash Information Field
0x78	System Data Flash Information Field
0x90	Splash Data 0
0x91	Splash Data 1
0x92	Splash Data 2
0x93	Splash Data 3
0x94	Splash Data 4
0x95	Splash Data 5
0x96	Splash Data 6
0x97	Splash Data 7
0x98	Splash Data 0 Flash Information Field
0x99	Splash Data 1 Flash Information Field

Table 8-83. Flash Data Type Values (continued)

FLASH DATA TYPE VALUE	DESCRIPTION
0x9A	Splash Data 2 Flash Information Field
0x9B	Splash Data 3 Flash Information Field
0x9C	Splash Data 4 Flash Information Field
0x9D	Splash Data 5 Flash Information Field
0x9E	Splash Data 6 Flash Information Field
0x9F	Splash Data 7 Flash Information Field
0xA0	Scratchpad Data 0
0xA1	Scratchpad Data 1
0xA2	Scratchpad Data 2
0xA3	Scratchpad Data 3
0xB8	Calibration Data Flash Information Field

Return Parameters

Table 8-84. Flash Read Data Return Parameters

BYTE	BITS	DESCRIPTION
1 ... n		**Read Data Bytes** Data bytes read from flash beginning at the commanded start address. The number of bytes, n, is determined by the commanded read data length.

8.2.72 Flash Verify Data - Write (A4h)

This command is used to verify the contents of the flash device.

The *Request in Progress* bit of the Short Status will be set at the start of the flash verify process and will be cleared when the verify process is complete. The error bits of the Short Status can be read to determine if an error occurred during the flash verify.

More information on flash programming can be found in Section 5.3.

Write Parameters

Table 8-85. Flash Verify Data Write Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**Flash Data Type** Refer to Table 8-86 for options.

The flash data types are described in Table 8-86.

Table 8-86. Flash Data Type Values

FLASH DATA TYPE VALUE	DESCRIPTION
0x00	Entire flash
0x01	Entire flash except scratchpad data
0x30	Main Application
0x40	Configuration Data 0
0x50	Configuration Data 1
0x60	Batch command sets
0x90	Splash Data 0
0x91	Splash Data 1
0x92	Splash Data 2
0x93	Splash Data 3
0x94	Splash Data 4
0x95	Splash Data 5
0x96	Splash Data 6
0x97	Splash Data 7
0xA0	Scratchpad Data 0
0xA1	Scratchpad Data 1
0xA2	Scratchpad Data 2
0xA3	Scratchpad Data 3

8.2.73 Flash Block Count - Read (A5h)

This command is used to read the number of flash blocks stored in flash memory and EEPROM memory.

Command Parameters

No command parameters.

Return Parameters

Table 8-87. Flash Block Count Return Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**Block count** Number of flash blocks.

8.2.74 Flash Block CRCs - Read (A6h)

This command is used to read the CRC values for all of the flash blocks stored in flash memory. These CRC values are used by the DLPC230-Q1 software to verify the contents of the flash blocks, but they can also be used as a tracking identifier for flash content. The expected CRC values for the flash data can be found in the flash header file that is delivered along with the flash data.

The return data length is variable based on the number of flash blocks. For each flash block, the command will return eight bytes of data. The number of flash blocks can be determined using the *Flash Block Count* command.

Command Parameters

No command parameters.

Return Parameters

Table 8-88. Flash Block CRCs Return Parameters

BYTE	BITS	DESCRIPTION
4:1	31:0	**Flash Block 1 Identifier** 7-bit ASCII code identifier of the flash block LSByte = Byte 1
8:5	31:0	**Flash Block 1 CRC** CRC value of the flash block LSByte = Byte 5
9:n		**Flash Block Identifier and CRC for remaining blocks** Each block receives 8 bytes for identifier and CRC. These bytes are in the same format as bytes 8:1 for Flash Block 1.

8.2.75 Flash Structure Version - Read (A7h)

This command is used to read the flash structure version of the current flash build. Each main application expects a certain flash structure version. Flash programming will fail if the flash structure version of the incoming data does not match the version that the main application expects. Note that this is different than data-dependent versioning. Data-dependent versioning is performed with flash block CRCs.

Command Parameters

No command parameters.

Return Parameters

Table 8-89. Flash Structure Version Return Parameters

BYTE	BITS	DESCRIPTION
2:1	15:0	**Flash Structure Version - Patch** LSByte = Byte 1
3	7:0	**Flash Structure Version - Minor**
4	7:0	**Flash Structure Version - Major**

8.2.76 Flash Data Size - Read (A9h)

This command is used to read the size of the specified flash block. This size returned is the maximum size available for the specified block. This can be used during partial flash programming to confirm whether the new flash block can fit within the previously allocated size. Refer to Section 5.3 for more information on flash programming.

Command Parameters

Table 8-90. Flash Data Size Command Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**Flash Data Type** Refer to Table 8-91 for options.

The flash data types are described in Table 8-91.

Table 8-91. Flash Data Type Values

FLASH DATA TYPE VALUE	DESCRIPTION
0x00	Entire flash
0x01	Entire flash except scratchpad data
0x30	Main Application (Build 0)
0x40	Configuration Data 0
0x48	Configuration Data 0 Flash Information Field
0x50	Configuration Data 1
0x58	Configuration Data 1 Flash Information Field
0x60	Batch Command Sets
0x90	Splash Data 0
0x91	Splash Data 1
0x92	Splash Data 2
0x93	Splash Data 3
0x94	Splash Data 4
0x95	Splash Data 5
0x96	Splash Data 6
0x97	Splash Data 7
0xA0	Scratchpad Data 0
0xA1	Scratchpad Data 1
0xA2	Scratchpad Data 2
0xA3	Scratchpad Data 3

Return Parameters

Table 8-92. Flash Data Size Return Parameters

BYTE	BITS	DESCRIPTION
4:1	31:0	**Block Size** Number of bytes in the selected flash block. LSByte = Byte 1

8.2.77 System Software Version - Read (B0h)

This command is used to read the software version of the main application. This command is identical to the boot application command, but the boot application and main application will not have the same version number.

Command Parameters

No command parameters.

Return Parameters

Table 8-93. System Software Version Return Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**DLPC230-Q1 Main Application Version - Patch**
2	7:0	**DLPC230-Q1 Main Application Version - Branch**
3	7:0	**DLPC230-Q1 Main Application Version - Minor**
4	7:0	**DLPC230-Q1 Main Application Version - Major**

8.2.78 Flash Device ID - Read (B1h)

This command is used to read flash device information.

Command Parameters

No command parameters.

Return Parameters

Table 8-94. Flash Device ID Return Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**Flash Manufacturer ID**
2	7:0	**Flash Memory Type**
3	7:0	**Flash Memory Capacity**

The manufacturer ID and memory type can be found from the selected flash device specification.

The flash memory size can be determined from Table 8-95.

Table 8-95. Flash Memory Size Values

MEMORY CAPACITY (BYTE 3)	ACTUAL SIZE
0x15	16Mb
0x16	32Mb
0x17	64Mb
0x18	128Mb
0x19	256Mb(1)

(1) Only the first 128Mb of address space is useable.

8.2.79 DLPC230-Q1 Device ID - Read (B2h)

This command is used to read the DLPC230-Q1 device ID. This command can return two sets of bytes depending on the byte parameter that is sent with the command.

Command Parameters

Table 8-96. DLPC230-Q1 Device ID Command Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**DLPC230-Q1 Data Selection** 0x0: Read DLPC230-Q1 device ID and Product Configuration ID 0x1: Read DLPC230-Q1 fuse data 0x02 - 0xFF: *Reserved*

Return Parameters (0x0)

The return parameters in Table 8-97 are returned if the command parameter is 0x0.

Table 8-97. DLPC230-Q1 Device ID Return Parameters 0x0

BYTE	BITS	DESCRIPTION
1	7:0	**DLPC230-Q1 Device ID**
2	7:4	**Reserved**
	3:0	**DLPC230-Q1 Product Configuration ID**

Table 8-98 shows the possible DLPC230-Q1 Device ID values.

Table 8-98. DLPC230-Q1 Device IDs

Device ID Value	DLPC230-Q1 Architecture	DLPC230-Q1 Turn	DLPC230-Q1 Pass
0x98	b10011	b00	b0

Return Parameters (0x1)

The return parameters in Table 8-99 are returned if the command parameter is 0x1.

Table 8-99. DLPC230-Q1 Device ID Return Parameters 0x1

BYTE	BITS	DESCRIPTION
8:1	63:0	**DLPC230-Q1 Fuse ID**

Table 8-100. DLPC230-Q1 Fuse ID Definition

BITS	LENGTH (BITS)	DESCRIPTION
0:29	30	Product Lot ID Device manufacturer tracking data
30:31	2	TI Device ID
32:36	5	Wafer ID Device manufacturer tracking data
37:50	14	Die X/Y Device manufacturer tracking data
51	1	Fab ID Device manufacturer tracking data
52:55	4	TI Product Cfg ID
56:63	8	CRC8(1)

(1) CRC uses CRC-8 CCITT (x^8+x^2+x+1) on each byte of data beginning with the least significant byte. The initial value is 0xFF.

8.2.80 DMD Device ID - Read (B3h)

This command is used to read the DMD device ID. This command can return two sets of bytes depending on the byte parameter that is sent with the command.

Command Parameters

Table 8-101. DMD Device ID Command Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**DMD Data Selection** 0x0: Read DMD Device ID *0x1 - 0xFF: Reserved*

Return Parameters

The return parameters below are returned if the command parameter is 0x0.

Table 8-102. DMD Device ID Return Parameters

BYTE	BITS	DESCRIPTION
4:1	31:0	**DMD Device ID**

Table 8-103. DMD Device IDs

DEVICE DESCRIPTION	DEVICE ID (HEX)
0.55" ES1	60 0D 00 7E
0.55" ES1.1	60 0D 00 82
0.55" ES2	60 0D 00 81
DLP5530A-Q1 / DLP5530S-Q1	60 0D 00 97
DLP462XX-Q1 (0.46")	60 0D 00 A1

8.2.81 TPS99000-Q1 Device ID - Read (B4h)

This command is used to read the TPS99000-Q1 device information.

Command Parameters

No command parameters.

Return Parameters

Table 8-104. TPS99000-Q1 Device ID Return Parameters

BYTE	BITS	DESCRIPTION
1	7:4	**TPS99000-Q1 Device ID** Major
	3:0	**TPS99000-Q1 Device ID** Minor

8.2.82 System Temperatures - Read (B5h)

This command is used to read system temperatures. The reporting range is -40°C to 128°C. The reported temperature includes an integer component and a fractional component. The fractional component is always added to the integer component. Temperature filtering averages multiple raw measurements in order to reduce noise from the temperature measurements.

Command Parameters

Table 8-105. System Temperatures Command Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**Temperature Source Selection** 0x0: DMD raw temperature 0x1: DMD filtered temperature 0x2: Temperature sensor local raw temperature 0x3: Temperature sensor local filtered temperature 0x4 - 0xFF: *Reserved*

Return Parameters

Table 8-106. System Temperatures Return Parameters

BYTE	BITS	DESCRIPTION
2:1	15:8	**Temperature Integer Celsius** Two's complement integer portion of temperature 127 = 127 Celsius 255 = –1 Celsius
	7:4	**Temperature Fraction Celsius** 1/16 Celsius steps always added to the integer portion. Note that this fraction is still added even if the integer portion is negative. 1 = 0.0625 Celsius 2 = 0.125 Celsius
	3:0	***Reserved***
3	7:0	***Reserved***

8.2.83 Current Source Information - Read (B6h)

This command is used to read information about the currently active image source. The first byte specifies the type of source and subsequent bytes describe the settings for the source type. Each source type has different description parameters, but the length of the return bytes from this command is always 15 bytes.

Command Parameters

No command parameters.

Return Parameters (External Video)

Table 8-107. Current Source Information Return Parameters (External Video)

BYTE	BITS	DESCRIPTION
1	7:0	**Input Source Type** 0x0: External Video 0x1: Test Pattern 0x2: Splash Screen 0x3: System in Standby or source is invalid
3:2	15:0	**External VSync Rate** LSByte = Byte 2 Specified in Hz with a resolution of 0.125 Hz. For example, 0x1E5 = 60.625 Hz.
4	7:0	**External Video Format** 0x00: OpenLDI 0x40: Parallel
5	7:4	***Reserved***
	3	**External Data Enable Polarity** Polarity of the parallel video DATEN signal. This parameter is unused for OpenLDI. 0x0: Active Low 0x1: Active High
	2	**External Clock Polarity** Polarity of the parallel video PCLK signal. This parameter is unused for OpenLDI. 0x0: Falling Edge Sample 0x1: Rising Edge Sample
	1	**External HSync Polarity** 0x0: Falling Edge Active (Negative Pulse) 0x1: Rising Edge Active (Positive Pulse)
	0	**External VSync Polarity** 0x0: Falling Edge Active (Negative Pulse) 0x1: Rising Edge Active (Positive Pulse)
7:6	15:0	**External Pixel Clock Rate** LSByte = Byte 6 Specified in MHz with a resolution of 0.125 MHz. For example, 0x1E1 = 60.125 MHz.
9:8	15:0	**Input Total Pixels Per Line** LSByte = Byte 8
11:10	15:0	**Input Total Lines Per Frame** LSByte = Byte 10
13:12	15:0	**Input Active Pixels Per Line** LSByte = Byte 12
15:14	15:0	**Input Active Lines Per Frame** LSByte = Byte 14

Return Parameters (Test Pattern)

Table 8-108. Current Source Information Return Parameters (Test Pattern)

BYTE	BITS	DESCRIPTION
1	7:0	**Input Source Type** 0x0: External Video 0x1: Test Pattern 0x2: Splash Screen 0x3: Not applicable (System in Standby)
3:2	15:0	**Internal VSync Rate** LSByte = Byte 2 Specified in Hz with a resolution of 0.125 Hz. For example, 0x1E5 = 60.625 Hz.
4	7:0	**Test Pattern Type** 0x00: Solid 0x01: Horizontal Ramp (Fixed step) 0x02: Vertical Ramp (Fixed step) 0x03: Horizontal Lines 0x04: Diagonal Lines 0x05: Vertical Lines 0x06: Grid Lines (Horizontal and Vertical) 0x07: Checkerboard 0x08 Color Bars 0x09 - 0x0F: *Reserved*
5	7:4	**Foreground Color** Foreground color is not used when certain test pattern types are displayed. Refer to Table 8-109. This value should be ignored if the test pattern type does not use this value. 0x0: Black 0x1: Red 0x2: Green 0x3: Blue 0x4: Cyan 0x5: Magenta 0x6: Yellow 0x7: White 0x8 - 0xF: *Reserved*
	3:0	**Background Color** Background color is not used when certain test pattern types are displayed. Refer to Table 8-109. This value should be ignored if the test pattern type does not use this value. 0x0: Black 0x1: Red 0x2: Green 0x3: Blue 0x4: Cyan 0x5: Magenta 0x6: Yellow 0x7: White 0x8 - 0xF: *Reserved*
9:6	39:0	**Test Pattern Parameters** *Reserved*
15:10	47:0	**Pad Bytes** Extra bytes to keep the command's return length 15 bytes for any source type Value = 0x0

Table 8-109. Test Pattern Foreground / Background Color Usage

PATTERN	FOREGROUND COLOR USED	BACKGROUND COLOR USED
Solid	Yes	No
Horizontal Ramp	Yes	No
Vertical Ramp	Yes	No
Horizontal Lines	Yes	Yes
Vertical Lines	Yes	Yes
Diagonal Lines	Yes	Yes
Grid Lines	Yes	Yes
Checkerboard	Yes	Yes
Color Bars	No	No

Return Parameters (Splash)

Table 8-110. Current Source Information Return Parameters (Splash)

BYTE	BITS	DESCRIPTION
1	7:0	**Input Source Type** 0x0: External Video 0x1: Test Pattern 0x2: Splash Screen 0x3: Not applicable (System in Standby)
3:2	15:0	**Internal VSync Rate** LSByte = Byte 2 Specified in Hz with a resolution of 0.125 Hz. For example, 0x1E5 = 60.625 Hz.
4	7:0	**Splash Screen Index** Reference number of the splash screen stored in flash (0-7).
15:5	87:0	**Pad Bytes** Extra bytes to keep the command's return length 15 bytes for any source type Value = 0x0

8.2.84 Current Display Information - Read (B8h)

This command is used to read information about the currently active display settings.

Command Parameters

No command parameters.

Return Parameters

Table 8-111. Current Display Information Return Parameters

BYTE	BITS	DESCRIPTION
1	7:1	***Reserved***
	0	**LED Status** 0x0: LEDs disabled 0x1: LEDs enabled
2	7:2	***Reserved***
	1	**Short Axis Image Flip** 0x0: Image not flipped 0x1: Image flipped
	0	**Long Axis Image Flip** 0x0: Image not flipped 0x1: Image flipped
8:3	47:0	***Reserved***
9	7:0	**System Mode Index**
10	7:0	**Illumination Bin Index**
11	7:0	**De-gamma LUT Index**
12	7:0	**Sequence Index**

8.2.85 System Information - Read (BAh)

This command is used to read the last reset cause and the product configuration.

Command Parameters

No command parameters.

Return Parameters

Table 8-112. System Information Return Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**Reset Cause** 0x00: ASIC Power-Up 0x01: PROJ_ON 0x02: TPS99000-Q1 Watchdog Software Error 0x03: TPS99000-Q1 Watchdog Sequence Error 0x02 - 0x03: *Reserved* 0x04: TPS99000-Q1 Die Temperature Exceeded 0x05: Software Commanded Power Cycle 0x06: *Reserved* 0x07: Host Commanded Reset 0x08: Software commanded DLPC230-Q1-Only Reset 0x09 - 0xFF: *Reserved*

Table 8-112. System Information Return Parameters (continued)

BYTE	BITS	DESCRIPTION
2	7:0	**Product Configuration** 0x00: HUD 0x01: Headlight 0x02 - 0xFF: *Reserved*

8.2.86 Flash Interface Rate - Read (BBh)

This command is used to read the flash interface rate settings.

Command Parameters

No command parameters.

Return Parameters

Table 8-113. Flash Interface Rate Return Parameters

BYTE	BITS	DESCRIPTION
1	7:5	**Reserved** Always 0x0
	4	**Quad Input / Output Read** 0x0: Not supported 0x1: Supported
	3	**Quad Output Read** 0x0: Not supported 0x1: Supported
	2	**Dual Input / Output Read** 0x0: Not supported 0x1: Supported
	1	**Dual Output Read** 0x0: Not supported 0x1: Supported
	0	**Fast Read** 0x0: Not supported 0x1: Supported
3:2	15:0	**Maximum Flash Clock Rate** LSByte = Byte 2 This value should be divided by 100 to read the MHz value. For example, 0x134D = 4941 / 100 = 49.41 MHz.

Table 8-114. Flash Read Instruction Op-Codes

FLASH INSTRUCTION NAME	FLASH OP-CODE
Fast Read	0x0B
Dual Output Read	0x3B
Dual Input / Output Read	0xBB
Quad Output Read	0x6B
Quad Input / Output Read	0xEB

8.2.87 Short Status - Read (C0h)

This command is used to read the short status from hardware. This is the only read command that does not require the use of *Read Pre-Fetch* and *Read Activate* commands. Refer to the *Communication Protocol* section for more information on the Short Status protocol.

Command Parameters

No command parameters.

Return Parameters

Table 8-115. Short Status Return Parameters

BYTE	BITS	DESCRIPTION
1	7:6	**Application / Mode** 0x0: Boot application 0x1: Main application - Standby 0x2: Main application - Display
	5	**Emergency Shutdown** 0x0: Not activated 0x1: Activated
	4	***Reserved***
	3	**Read Data Available** 0x0: No data available 0x1: Data available
	2	**System Busy** 0x0: Not busy 0x1: Busy
	1	**Request in Progress** 0x0: Not in progress 0x1: In progress
	0	**System Initialized** 0x0: Not initialized 0x1: Initialized
2	7:0	**Execution Command Tag**
4:3	15	**BIST Error** 0x0: No error 0x1: Error
	14	**Operational Error** 0x0: No error 0x1: Error
	13	**Command Error** 0x0: No error 0x1: Error
	12	**Communication Error** 0x0: No error 0x1: Error
	11:0	**Error Code**

A diagram of these short status bits is shown in Figure 8-1.

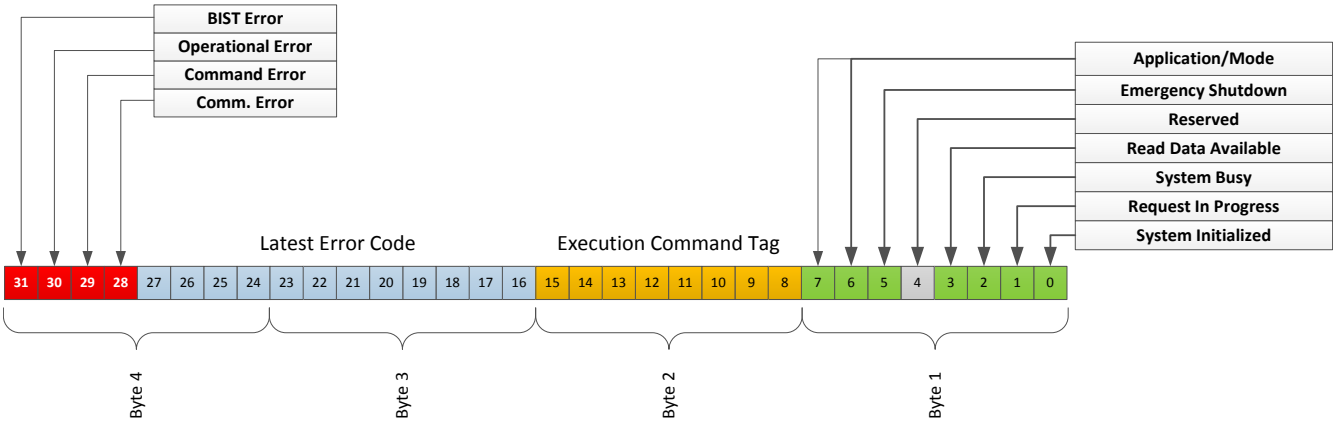


Figure 8-1. Short Status Bit Definition

The bits are described in Table 8-116.

Table 8-116. Short Status Field Descriptions

BIT FIELD	DEFINITION
System Initialized	Indicates that the system software is ready to accept commands for processing. Typically this would be set by the Main Application, except when the system is forced to stay in boot.
Request in Progress	This bit is used to inform the host that a commanded task is being performed. When the task is initiated, this bit will be set to “Yes” and when the operation has completed the bit will be cleared to “No”. If the task was a BIST, the results for the requested test will be valid once the bit has been cleared by embedded software. Further requests can be started after this.
System Busy	This bit is used to inform the host that the system's receive FIFO is full. The host should not send any more commands when the system is busy, or the commands and associated data may be lost. The Host is free to send commands when the system is not busy.
Read Data Available	Indicates when read data is available after the host has sent a Read Prefetch command. When data is available, the host should send the Read Activate command to fetch the requested data. The host should always fetch requested data using the Read Activate command before sending another Read Prefetch command. If a Read Prefetch command is sent before the data from a previous Read Prefetch command has been fetched, the previous data will be flushed, and the latest requested data will be made available for fetching by Read Activate. There will be no error indication that this has occurred.
Emergency Shutdown	This bit is used to indicate that the system has automatically gone to Standby Mode due to a critical system error. More information on emergency shutdown can be found in Section 6.2.
Application/Mode	These bits indicate which software application is currently running, and when in the Main Application, they indicate the current operational mode of the system.
Execution Command Tag	The command tag for the last write command that has completed software execution, whether successful or not. This byte is continually updated as new commands are received and executed.
Error Code	The 12-bit error code is used to specify the last communication or command error received during system operation. The error code can indicate no error (error code = 0h), or indicate a specific code for the most recent error to occur. The *Error History* command can be used to obtain details about previous errors.
Communication Error	A flag set to indicate the occurrence of any communication error, which is used to indicate a problem with the transmission/reception of a command. Some examples are: • RXFIFO overflow • Command transmission terminated early (the host didn't provide enough SPI clock pulses for all requested data). • Command transmission terminated late (the host provided too many SPI clock pulses for requested data).

Table 8-116. Short Status Field Descriptions (continued)

BIT FIELD	DEFINITION
Command Error	A flag set to indicate a command error, or an error in the action requested by the command. Some examples are: • Command executed in an invalid operating mode • CRC Error in Command Header • CRC Error in Payload (Bulk Command) • Invalid command op-code • Invalid command parameter (for example, out of range) • Incorrect number of command parameters • Non-periodic BIST failure when BIST started by command • Error erasing, writing, or reading flash when commanded • Flash overflow error
Operational Error	A flag set to indicate an operational error, which are any errors that don't fall into one of the other three error categories. Some examples are: • Sequence CRC error • CMT CRC error • Sequence / CMT mismatch error • Periodic BIST failure • Unable to communicate with TPS99000-Q1 • Unable to communicate with temperature sensor
BIST Error	A flag set to indicate a non-periodic or periodic BIST error.

8.2.88 Error History - Read (C1h)

This command is used to read the details of the first 62 errors that the system encounters. Subsequent errors beyond the first 62 will not be stored, but the error count will continue to increase to indicate that more errors occurred. Four bytes are allocated for each error. The *Clear Error History* command should be used periodically after reading the error history in order to clear errors and to avoid losing any future error debug information. The command will always return 249 bytes regardless of the number of errors currently contained in the error history.

Command Parameters

No command parameters.

Return Parameters

Table 8-117. Error History Return Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**Error Count** Count of errors received since the last power-cycle or clear of the error history. If the error count is greater than 62, some error details are not available since storage is full.
5:2	31:0	**Error 1 Details** LSByte = 2
9:6	31:0	**Error 2 Details** LSByte = 6
...	...	**Error n Details**
249:246	31:0	**Error 62 Details** LSByte = 246

Each set of error bits includes a unique error code, category flags, and additional information bits that vary based on the type of error. The bit definition for each error is broken down as follows:

Table 8-118. Error Detail Bit Descriptions

BITS	DESCRIPTION
31	**BIST Error** 0x1: This error is categorized as a BIST error
30	**Operational Error** 0x1: Error is categorized as an operational error
29	**Command Error** 0x1: Error is categorized as a command error
28	**Communication Error** 0x1: Error is categorized as a communication error
27	***Reserved*** Always 0
26:16	**Error code** Unique error code to identify the error.
15:0	**Informational bits** These can vary depending on the category and specific type of error.

A diagram of the error detail fields is shown in Figure 8-2:

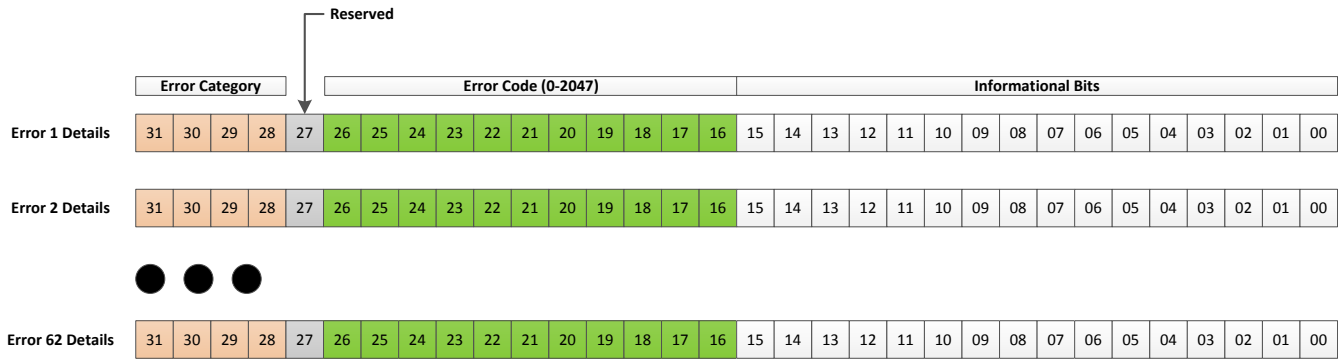


Figure 8-2. Error History Details

8.2.88.1 Information Bits - Command or Communication

If the Command Error or Communication Error bits are set in the error, then the information bits will provide the following information.

Table 8-119. Information Bit Definition Command or Communication

BITS	DESCRIPTION
15:8	**Command tag** Tag of the command that received the error.
7:0	**Command Op-Code** Op-code of the command that received the error.

8.2.88.2 Information Bits - System Voltage

If the error is a system voltage error, then the information bits will provide the value of the invalid voltage that was detected.

These errors include 843, 844, 845, 846, 847, 848, 849, 1008, 1015.

Table 8-120. Information Bit Definition System Voltage

BITS	DESCRIPTION
15:0	**Error voltage level** Absolute value of the measured floating point value converted to u8.8 format.

8.2.88.3 Information Bits - DMD High Speed Interface Training

If the error is related to DMD high speed interface training, then the information bits will provide the channel and pin that received the error.

These errors include 555, 556, 559.

Table 8-121. Information Bit Definition DMD HS Interface Training

BITS	DESCRIPTION
15:8	**Training Channel** From the DLPC230-Q1 hardware perspective. Due to channel swapping options, Channel 0 on the DLPC230-Q1 can sometimes connect to Channel 1 on the DMD. 0x0: Channel 0 0x1: Channel 1 0x2: Both channels
7:0	**Signal number** From the DLPC230-Q1 hardware perspective. Due to pin swapping options, a pin on the DLPC230-Q1 may be connected to a different pin on the DMD. 0 - 7 corresponding to the 8 differential signals on each channel.

8.2.88.4 Information Bits - DMD Memory Test

Error 220 includes the number of CMOS columns that failed the DMD memory test. Refer to Section 6.4.2.6 for more information on the DMD memory test.

BITS	DESCRIPTION
15:0	**Number of failed columns**

8.2.89 Clear Short Status Errors - Write (C2h)

This command is used to clear the Short Status error bits. This includes bytes 3 and 4 of the Short Status.

Write Parameters

Table 8-122. Clear Short Status Errors Write Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**Signature Byte 1** 0xAA
2	7:0	**Signature Byte 2** 0xBB
3	7:0	**Signature Byte 3** 0xCC
4	7:0	**Signature Byte 4** 0xDD

8.2.90 Clear Error History - Write (C3h)

This command is used to clear the Error History, the Error Count, and the Short Status error bits.

Write Parameters

Table 8-123. Clear Error History Write Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**Signature Byte 1** 0xAA
2	7:0	**Signature Byte 2** 0xBB
3	7:0	**Signature Byte 3** 0xCC
4	7:0	**Signature Byte 4** 0xDD

9.1 Diagnostic Command Read Procedure

The diagnostic interface is hardware controlled, meaning that there is no delay time to fetch requested data. The read pre-fetch and read-activate command flow is still used, but there is no short status to determine when read data is available. Read activate can occur immediately following read pre-fetch.

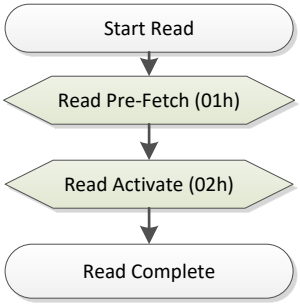


Figure 9-1. Read Procedure for Diagnostic Interface

9.2 Command Table

Table 9-1 summarizes the diagnostic interface commands.

Table 9-1. Command Summary - Diagnostic Interface

COMMAND	TYPE	OP CODE	SECTION
Read Pre-fetch	Write	01h	Section 9.3.1
Read Activate	Read	02h	Section 9.3.2
Diagnostic Interface Status	Read	F0h	Section 9.3.3
Diagnostic Interface Status Clear	Write	F1h	Section 9.3.4

9.3 Command Definitions

9.3.1 Read Pre-Fetch - Write (01h)

This command is used to send the desired read command op-code and associated command parameters to initiate a read request. The desired read data will not be returned during this command transaction. More information on the read procedure can be found in Section 3.5.

Write Parameters

Table 9-2. Read Pre-Fetch Write Parameters

BYTE	BITS	DESCRIPTION
1	7:0	**Read command op-code**
2 ... n		**Read command parameters** Depending on the Read op-code, a specific number of command parameter bytes will be expected. The required byte parameters are documented with each read command.

9.3.2 Read Activate (02h)

This command is used to activate the read operation in order to retrieve the previously requested read data. This generic command is used to retrieve all requested read data with the exception of the *Short Status* command. More information on the read procedure can be found in Section 3.5.

Command Parameters

No command parameters.

Return Parameters

Table 9-3. Read Activate Return Parameters

BYTE	BITS	DESCRIPTION
1 ... n		**Data Bytes 1...n** Read data bytes. The number of bytes will vary depending on the command that is currently being read.

9.3.3 Diagnostic Interface Status - Read (F0h)

This command is used to read status information from the diagnostic interface.

Command Parameters

Table 9-4. Diagnostic Interface Status Command Parameters

BYTE	BITS	DESCRIPTION
2:1	15:0	**Start Index** 0-based index of the first desired byte value to be returned. 0 - 63 LSByte = Byte 1
4:3	15:0	**Number of Bytes** 1-based number of data bytes to be returned. 1 - 64 Start Index + Number of Bytes must be less than or equal to 64. LSByte = Byte 3

Return Parameters

Table 9-5. Diagnostic Interface Status Return Parameters

BYTE	BITS	DESCRIPTION
1 ... n		**Data Bytes** Data values to return. The number of bytes is determined by the "Number of Bytes" field specified in the command parameters.

9.3.4 Diagnostic Interface Status Clear - Write (F1h)

This command is used to clear the diagnostic interface status data. There is a mask bit for each byte to indicate which bytes will be cleared by the command.

Write Parameters

Table 9-6 indicates the byte index for each write bit. *Table 9-7* describes the value options for each bit.

Table 9-6. Diagnostic Interface Status Clear Write Parameters

BYTE	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	07	06	05	04	03	02	01	00
2	15	14	13	12	11	10	09	08
3	23	22	21	20	19	18	17	16
4	31	30	29	28	27	26	25	24
5	39	38	37	36	35	34	33	32
6	47	46	45	44	43	42	41	40
7	55	54	53	52	51	50	49	48
8	63	62	61	60	59	58	57	56

Table 9-7. Diagnostic Interface Clear Values

DIAGNOSTIC CLEAR VALUES	DESCRIPTION
0x0	Not selected - The byte will not be cleared.
0x1	Selected - The byte will be cleared.

Chapter 10
Flash Configuration

10.1 Overview

Flash memory is used to store the main application and operational data that is used by the main application. The flash data blocks consist of:

- **Main Application** - The embedded software binary that is executed by the DLPC230-Q1 ARM processor.
- **Configuration Data** - Data read by the main application and boot application to configure system functionality.
- **Sequence Data** - Includes sequencing information that dictates how the DMD and illumination are driven during a frame.
- **Batch Command Sets** - Contains all of the available batch command sets. More detail about batch command sets can be found in Section 5.2.
- **Splash Image Data** - Stores all of the available splash image binaries. More detail about splash images can be found in Section 5.1.5.
- **Scratchpad Data** - Stores any miscellaneous data that the end-user requires. Embedded software does not use this data for any reason. It can be read back during system operation using *Flash Read* commands.

10.2 System Mode Overview

The following terms are commonly used when referring to illumination and display settings:

- **System Mode** - A flash structure that maps several types of look-up tables that are intended to be used together in order to achieve a desired image appearance. The tables within a System Mode are all designed for a specific:
  - Frame Rate - The sequences are designed to match a specific input video frame rate
  - RGB Duty Cycle - Percentage of sequence time allocated for red, green, and blue illumination
- **Sequence** - A table that controls the synchronization of DMD timing and illumination enable signals throughout a video frame time. Every sequence within a System Mode is designed for the same frame rate and RGB duty cycle. Each sequence is characterized by a specific DMD duty cycle. For example, one sequence may use a 70% (70/30) DMD duty cycle and another may use 50%, 50% (50/50) DMD duty cycle. This percentage is related to the maximum light output that the system can output.
- **Illumination Bin** - A table that controls the amount of time that illumination drive signals are active for a given sequence. This is used to provide coarse step control over the amount of light output in dimming use-cases. For applications that do not use dimming, illumination bins are designed to provide maximum sequence intensity.
- **De-gamma Table** - A de-gamma curve that controls the mapping of input pixel levels to output pixel brightness levels. These are generally used to compensate for gamma curves that are inherent to video content or to accentuate specific regions of pixel levels. The same de-gamma curves are used for each sequence so that every system brightness can use the same de-gamma curves.

The available sequences, illumination bins, and de-gamma tables are specified in the flash header file provided along with the flash binary file.

The host specifies a System Mode index, an Illumination Bin index, and a De-gamma index. The host never directly specifies a sequence index. This is explained by the example below.

Figure 10-1 uses an example indexing to show the relationship between these tables. A System Mode contains a set of Illumination Bins. The host specifies the system brightness by selecting the desired illumination bin

index within a System Mode. Each Illumination Bin is designed for a specific sequence. When the host selects an illumination bin, the main application will automatically select the one sequence that the illumination bin is mapped to. The host never directly sets the sequence. Different Illumination Bins can be mapped to the same sequence. This indicates the use of different illumination percentages with the same DMD duty cycle to achieve a different total brightness level. Each sequence maps to one or more de-gamma curves. The same set of de-gamma curves is applied to all sequences. For example, de-gamma curve 0 could be a simple linear mapping from input pixel to output pixel level (128 in = 128 out). In that example, de-gamma 0 will be linear for all sequences.

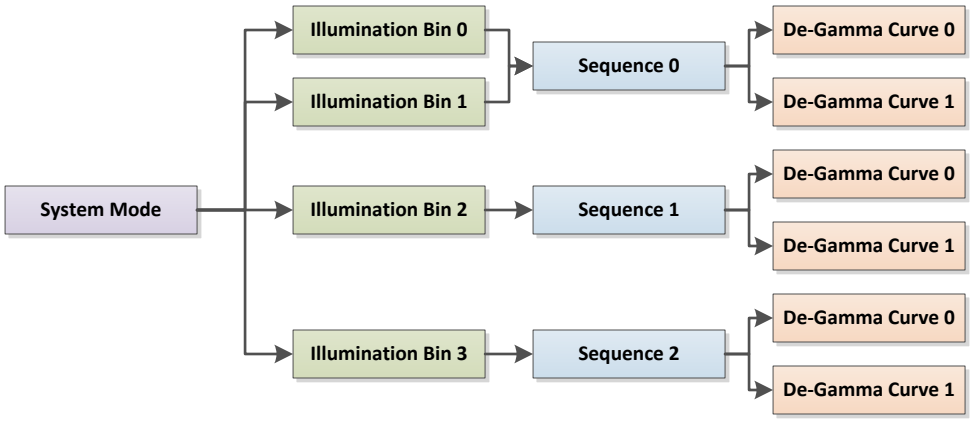


Figure 10-1. System Mode Example

10.3 Scratchpad Data

There are four flash scratchpad blocks that may be used to store any desired data. The data within these blocks are not used for DLPC230-Q1 software configuration in any way. The data structure for each of these individual blocks is shown in Table 10-1.

Multi-byte values use little endian format (LSB first).

The size of each block is one flash sector (4096 bytes) and the block data must fill the flash sector. Padding bytes should be 0xFF.

Table 10-1. Scratchpad Data

BYTE ADDRESS (HEX)	0	1	2	3
0000	$'O'_C$	$'E'_C$	$'M'_C$	Block $Number_{U8}$
0004	CRC_{U32}			
0008	$SIZE_{U32}$ = 4084			
000C	$Major_{U8}$	$Minor_{U8}$	$Patch_{U16}$	
0010	Custom Data			
...				
0FFC				

10.3.1 CRC

The 32-bit CRC is implemented as follows:

- Polynomial $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$ (normal polynomial representation = 0x04C11DB7)
- Initial value is 0xFFFFFFFF

The bytes within each word (4 bytes) of data must be reversed prior to the CRC calculation. For example, using the data word 0xAABBCCDD, the value to be used in the CRC is 0xDDCCBBAA.

The CRC must include all words in the scratchpad block starting after the "Size" field and including all padding bytes up to the flash sector size.

10.3.2 Block Number

The block number distinguishes between the four scratchpad blocks for partial flash programming and partial flash reading. Valid values are 0, 1, 2, 3.

10.3.3 Version (Major, Minor, Patch)

The version fields must match the expected flash structure version. This value may change with new software releases, but can be read back using the *Flash Structure Version - Read* command.

10.3.4 Custom Data

This field may include any desired content. It will not be used by DLPC230-Q1 software.

Appendix A
Error Codes

A.1 Boot Application

The following table describes the error codes that may be received during boot application execution.

Table A-1. Boot Application Error Codes

CODE	NAME	DESCRIPTION
03	Flash Out Of Range	Set during Flash Write command if the flash image is larger than 128Mb.
04	Flash Read Timeout	Set during startup if the flash device ID cannot be read.
05	Flash Busy	Set during any flash read, erase, or write operation if another operation is already accessing flash memory.
06 - 09	Internal Error	Internal system error. Contact TI for more information.
10	Invalid Command Type	Set if the next byte from the command interface is not the correct type. An example of this is if a data byte is read when an op-code byte is expected. This can occur if the host sends too many or too few bytes for a particular command.
11	Invalid Command Format	Set if the op-code sent using a Read Pre-fetch command is actually a "write" op-code, or the Read Pre-fetch command did not contain an op-code parameter. It can also be set during a Flash Write if the data payload length was not provided.
13	Invalid Flash Write Sequence	Set during the flash write command if the flash has not been erased.
14	Internal Error	Internal system error. Contact TI for more information.
15	Invalid Command Op-Code	The requested command op-code does not exist.
16	Invalid Command Signature	Set if the signature bytes are invalid for the following commands: System Reset, Flash Erase, Clear Error History, or Clear Short Status.
17	Internal Error	Internal system error. Contact TI for more information.
18	Mismatch Command CRC / Checksum	A CRC or checksum in the command is invalid.
19	Mismatch Command Payload Size	The command payload size didn't match the number of bytes transmitted.
20	Internal Error	Internal system error. Contact TI for more information.
22	Software FIFO Full	The software command FIFO is full and cannot accept more commands. This error can be prevented by waiting for the *System Busy* Short Status bit to be set low between command transactions so that the FIFO does not overrun.
24	Flash Application Block ID Error	Set during startup if the flash image detects an error in the application binary data.
25	Internal Error	Internal system error. Contact TI for more information.
26	Flash Invalid Device	Set if the flash device is not found during startup, an unknown flash device is found during startup, a flash device is not found and a Flash Erase is requested, or a flash device is not found and a Flash Write command is requested.
28 - 29	Internal Error	Internal system error. Contact TI for more information.
30	Flash Table CRC Error	A CRC error was detected while verifying the Flash table.
31 - 33	Internal Error	Internal system error. Contact TI for more information.
34	Flash Table ID Error	Flash table signature is not located at the correct address. If the first 4 bytes of flash are erased or corrupt, this error will occur when a flash verify or transition to main application are attempted.

Table A-1. Boot Application Error Codes (continued)

CODE	NAME	DESCRIPTION
35 - 58	Internal Error	Internal system error. Contact TI for more information.
70	Incorrect Command Header Format	The host command header format is invalid.
71	Read Activate Invalid CRC / Checksum	The host command had an invalid CRC or checksum during a Read Activate command.
72	Read Activate Header Early Termination	The host terminated a Read Activate transaction before the entire header data was sent.
73	Read Activate Response Early Termination	The host terminated a Read Activate transaction before the entire read payload was sent.
74	Read Activate Late Termination	The host terminated a Read Activate transaction later than expected based on the read payload length.
75	Read Activate Data Unavailable	Set if there is no read data available when the host sends a Read Activate command.
76	Host RX FIFO Overflow	A received host command caused the read FIFO to overflow, resulting in the loss of data from the previous read command.
77	Internal Error	Internal system error. Contact TI for more information.
78	Short Status Invalid CRC / Checksum	The host command had an invalid CRC or checksum during a Short Status command.
79	Short Status Header Early Termination	The host terminated a short status transaction before the entire short status header was sent.
80	Short Status Response Early Termination	The host terminated a short status transaction before the entire short status response was sent.
81	Short Status Late Termination	The host terminated a short status transaction later than expected.
82	Host TX FIFO Overflow	Software attempted to write to a full host interface FIFO.
85 - 121	Internal Error	Internal system error. Contact TI for more information.
123	Flash FIFO Timeout	Set during Flash Write command if software times out waiting for the serial flash controller. Also possibly set during startup if the serial flash controller times out while retrieving the flash device ID.
124 - 128	Internal Error	Internal system error. Contact TI for more information.
129	Invalid Read Prefetch Op Code	An invalid read op-code was sent using a Read Pre-Fetch command.
130	Internal Error	Internal system error. Contact TI for more information.
131	Boot Application Load CRC Error	A CRC error was detected while transferring the boot application to RAM.
132	Main Application Load CRC Error	A CRC error was detected while transferring the main application to RAM.
133	Main Application CRC Error in RAM	A CRC error was detected after transferring the main application to RAM.
134	Flash Block Verify CRC Error	A CRC error was detected while verifying the contents of the flash.
136	Internal Error	Internal system error. Contact TI for more information.
137	Internal Error	Internal system error. Contact TI for more information.

A.2 Main Application

The following table describes the error codes that may be received during main application execution. This preliminary list does not define every error code and will be updated during main application development.

Table A-2. Main Application Error Codes

CODE	NAME	DESCRIPTION
03	Flash Out Of Range	Set during Flash Write command if the flash image is larger than 128Mb.
04	Flash Read Timeout	Set during startup if the flash device ID cannot be read.
05	Flash Busy	Set during any flash read, erase, or write operation if another operation is already accessing flash memory.
06 - 09	Internal Error	Internal system error. Contact TI for more information.
10	Invalid Command Type	Set if the next byte from the command interface is not the correct type. For example, if a data byte is read when an op-code byte is expected. This can occur if the host sends too many or too few bytes for a particular command.
13	Invalid Flash Write Sequence	Set during the flash write command if the flash has not been erased.
15	Invalid Command Op-Code	The requested command op-code does not exist.
16	Invalid Command Signature	Set if the signature bytes are invalid for the following commands: System Reset, Flash Erase, Clear Error History, or Clear Short Status.
17	Internal Error	Internal system error. Contact TI for more information.
18	Mismatch Command CRC / Checksum	A CRC or checksum in the command is invalid.
19	Mismatch Command Payload Size	The command payload size didn't match the number of bytes transmitted.
20	Internal Error	Internal system error. Contact TI for more information.
22	Software FIFO Full	The software command FIFO is full and cannot accept more commands. This error can be prevented by waiting for the *System Busy* Short Status bit to be set low between command transactions so that the FIFO does not overrun.
26	Flash Invalid Device	Set if the flash device is not found during startup, an unknown flash device is found during startup, a flash device is not found and a Flash Erase is requested,or a flash device is not found and a Flash Write command is requested.
33 - 46	Internal Error	Internal system error. Contact TI for more information.
70	Incorrect Command Header Format	The host command header format is invalid.
71	Read Activate Invalid CRC / Checksum	The host command had an invalid CRC or checksum during a Read Activate command.
72	Read Activate Header Early Termination	The host terminated a Read Activate transaction before the entire header data was sent.
73	Read Activate Response Early Termination	The host terminated a Read Activate transaction before the entire read payload was sent.
74	Read Activate Late Termination	The host terminated a Read Activate transaction later than expected based on the read payload length.
75	Read Activate Data Unavailable	Set if there is no read data available when the host sends a Read Activate command.
76	Host RX FIFO Overflow	A received host command caused the read FIFO to overflow, resulting in the loss of data from the previous read command.
77	Internal Error	Internal system error. Contact TI for more information.
78	Short Status Invalid CRC / Checksum	The host command had an invalid CRC or checksum during a Short Status command.
79	Short Status Header Early Termination	The host terminated a short status transaction before the entire short status header was sent.
80	Short Status Response Early Termination	The host terminated a short status transaction before the entire short status response was sent.
81	Short Status Late Termination	The host terminated a short status transaction later than expected.
82	Host TX FIFO Overflow	Software attempted to write to a full host interface FIFO.
94 - 121	Internal Error	Internal system error. Contact TI for more information.

Table A-2. Main Application Error Codes (continued)

CODE	NAME	DESCRIPTION
123	Flash FIFO Timeout	Set during Flash Write command if software times out waiting for the serial flash controller. Also possibly set during startup if the serial flash controller times out while retrieving the flash device ID.
125 - 128	Internal Error	Internal system error. Contact TI for more information.
129	Invalid Read Prefetch Op Code	An invalid read op-code was sent using a Read Pre-Fetch command.
139	Command Error	Bezel horizontal offset exceeds -10% limit.
140	Command Error	Bezel horizontal offset exceeds +10% limit.
141	Command Error	Bezel vertical offset exceeds -50% limit.
142	Command Error	Bezel vertical offset exceeds +50% limit.
143 - 147	Internal Error	Internal system error. Contact TI for more information.
148	Command Error	Bezel horizontal offset is not a multiple of 2.
149	Command Error	Image orientation (flip) is invalid.
150 - 153	Internal Error	Internal system error. Contact TI for more information.
154	Fuse ID Error code	Number of lines * number of pixels exceeds the maximum resolution for this product
157 - 161	Internal Error	Internal system error. Contact TI for more information.
162	Command Error	Bezel vertical offset is not a multiple of 4.
163	Command Error	Execution delay error.
164	Command Error	GPIO invalid owner.
165	Command Error	GPIO wrong direction.
166	Command Error	Illumination Bin not found.
167	Command Error	System Mode not found.
168	Command Error	Contrast Parameter error.
169	Internal Error	Internal system error. Contact TI for more information.
170	Command Error	Flash invalid block type.
171	Command Error	Flash invalid block size.
172	Command Error	Flash invalid block version.
173	Command Error	Flash invalid data type.
174	Command Error	Invalid batch command.
175	Command Error	Batch command set invalid index.
176	Command Error	Flash invalid command sequence.
177	Command Error	Invalid payload length.
179	Command Error	ADC Measurement start or length parameter out of range.
180	Command Error	Command is not allowed in current operating mode.
181	Command Error	Temperature compensation source selection invalid.
182	Command Error	Reset is required after flash update.
183	Command Error	PWM duty cycle command parameter is greater than maximum allowed.
184	Command Error	De-gamma selection cannot be changed while illumination transition is in progress.
185 - 200	Internal Error	Internal system error. Contact TI for more information.
201	Command Error	Invalid source type selected.
202 - 205	Internal Error	Internal system error. Contact TI for more information.
206	Command Error	No splash screen at requested index.
207 - 213	Internal Error	Internal system error. Contact TI for more information.
214	Command Error	ASIC Device ID not yet available.
215	Command Error	System must stay in Standby mode until reset has occurred.

Table A-2. Main Application Error Codes (continued)

CODE	NAME	DESCRIPTION
216	Command Error	ASIC Device ID invalid selection.
217	Command Error	Operating Mode Select invalid mode.
218	Command Error	Temperature compensation parameter invalid.
219	Internal Error	Internal system error. Contact TI for more information.
220	Command Error	DMD Memory BIST columns failed.
221	Command Error	DMD Memory BIST failed.
222	BIST Error	Loss of dimming / ping BIST failed. No dimming / ping command was received within the configured time frame.
223	Command Error	Start index for diagnostic memory is out of range.
224	Command Error	Number of bytes for diagnostic memory is out of range.
229	Command Error	External video checksum start column invalid.
230	Command Error	External video checksum number of pixels per line invalid.
231	Command Error	External video checksum start row invalid.
232	Command Error	External video checksum number of lines per frame invalid.
233	Command Error	No settings provided for external video checksum. Write External Video Checksum Settings command prior to enabling the test.
234	Internal Error	Internal system error. Contact TI for more information.
237	Internal Error	Internal system error. Contact TI for more information.
238	Command Error	Attempted to change external video checksum settings, but external video checksum is enabled.
239	Command Error	Average Picture Level command parameter is out of range.
240	Command Error	Command not available for this product type.
241	Command Error	TPS99000-Q1 RGB Limits parameter is out of range.
242	Command Error	TPS99000-Q1 TIA2 parameter error.
244	Command Error	Commanded periodic BIST is disabled by flash setting and cannot be enabled by host command.
245	Command Error	CMODE value out of range.
246	Command Error	Socket connectivity test is able to read DMD ID
247	Command Error	Reset required after empty socket test
512	BIST Error	Back End BIST 1 failed.
513-516	Internal Error	Internal system error. Contact TI for more information.
517	BIST Error	Diagnostics emergency shutdown.
518	Internal Error	Internal system error. Contact TI for more information.
519	BIST Error	Memory BIST 17 failed.
520	BIST Error	Memory BIST 18 failed.
521	BIST Error	Memory BIST 19 failed.
522	BIST Error	Memory BIST 20 failed.
523	BIST Error	Memory BIST 21 failed.
524	BIST Error	Front End BIST 1 failed.
525	BIST Error	Internal system error. Contact TI for more information.
526	BIST Error	Flash Table Transport CRC failed.
527	BIST Error	Internal system error. Contact TI for more information.
528	BIST Error	External video checksum control invalid.
529	BIST Error	Non-Periodic test invalid.
530	BIST Error	Memory BIST 1 failed.
531	BIST Error	Memory BIST 2 failed.

Table A-2. Main Application Error Codes (continued)

CODE	NAME	DESCRIPTION
532	BIST Error	TPS99000-Q1 signal interface BIST failed.
533	BIST Error	Memory BIST 22 failed.
534	BIST Error	DLPC230-Q1 Command and Flash Interface Memory test failed.
535	BIST Error	External Video Checksum failed due to checksum mismatch.
536	BIST Error	Memory BIST 23 failed.
537	BIST Error	Memory BIST 24 failed.
538	BIST Error	Memory BIST 26 failed.
539	BIST Error	Memory BIST 25 failed.
540	BIST Error	Memory BIST 3 failed.
541	BIST Error	Memory BIST 12 failed.
542	BIST Error	Memory BIST 13 failed.
543	BIST Error	Memory BIST 14 failed.
544	BIST Error	Memory BIST 15 failed.
545	BIST Error	Memory BIST 16 failed.
546	BIST Error	Memory BIST 4 failed.
547	BIST Error	Memory BIST 5 failed.
548	BIST Error	Memory BIST 6 failed.
549	BIST Error	Memory BIST 7 failed.
550	BIST Error	Memory BIST 8 failed.
551	BIST Error	Memory BIST 9 failed.
552	BIST Error	Memory BIST 10 failed.
553	BIST Error	Memory BIST 11 failed.
554 - 555	Internal Error	Internal system error. Contact TI for more information.
556	DMD Training Error	DMD HS training failed: returned invalid results.
557	Internal Error	Internal system error. Contact TI for more information.
558	BIST Error	External video frame count out of sequence.
559	DMD Training Error	DMD HS Training failed: Insufficient DLL window.
560 - 563	Internal Error	Internal system error. Contact TI for more information.
564	DMD Compatibility Error	TPS99000-Q1 pixel type configuration is not compatible with the DMD used in the system.
565	DMD LS Error	Unexpected data was received during a transaction on the DMD low speed interface.
566	Flash Configuration Error	Invalid external video checksum fail action.
567 - 572	Internal Error	Internal system error. Contact TI for more information.
573	DMD Training Error	DMD HS training failed: real-time test controller HS training error 1.
574	DMD Training Error	DMD HS training failed: real-time test controller HS training error 2.
575	DMD Training Error	DMD HS training failed: real-time test controller HS training error 3.
576	DMD Training Error	DMD HS training failed: real-time test controller HS training processing error.
577	DMD LS Error	DMD Low Speed validation failed: processing error.
578 - 586	Internal Error	Internal system error. Contact TI for more information.
587	Flash Error	Serial flash device is locked.
588	System Mode Error	The currently selected system mode is not compatible with the source frame rate.
589	De-gamma Error	The requested de-gamma table was not found in the flash data.
590	Internal Error	Internal system error. Contact TI for more information.

Table A-2. Main Application Error Codes (continued)

CODE	NAME	DESCRIPTION
591	ADC Error	An attempt to manually read an ADC channel measurement requested an invalid channel (-1).
592 - 595	Internal Error	Internal system error. Contact TI for more information.
596	ADC Error	ADC measurement error: retry count exceeded.
597	ADC Error	ADC interface communication error. Data, parity, or stop-bit error occurred during transaction.
598 - 600	Internal Error	Internal system error. Contact TI for more information.
601	ADC Error	Attempted to manually read ADC measurement, but channel selection was invalid (>=64).
602	ADC Error	ADC measurement error: value saturated at minimum or maximum.
603 - 625	Internal Error	Internal system error. Contact TI for more information.
626	TPS99000-Q1 Interface Error	TPS99000-Q1 parity error during read.
627	TPS99000-Q1 Interface Error	TPS99000-Q1 parity error during write.
628 - 632	Internal Error	Internal system error. Contact TI for more information.
634	Thread Monitor Error	Thread is non-responsive within timeout.
636	DMD Access Error	DMD low speed interface communication failed. The ASIC did not receive an ACK from the DMD. This can occur if there is a DMD connection issue, or if the flash low speed port configuration does not match hardware connections.
637 - 652	Internal Error	Internal system error. Contact TI for more information.
653	Flash Table Error	Flash Table Type 1 CRC failed.
654	Flash Table Error	Flash Table Type 2 CRC failed.
655	Flash Table Error	Flash Table Type 3 transport CRC failed.
656	Flash Table Error	Flash Table Type 3 CRC failed.
657	Flash Table Error	Flash Table Type 4 CRC failed.
658	Flash Table Error	Flash Table Type 5 memory access failed.
659	Flash Table Error	Flash Table Type 5 CRC failed.
660	Flash Table Error	Flash Table Type 6 memory CRC failed.
661	Flash Table Error	Flash Table Type 7 memory CRC failed.
662	Internal Error	Internal error. Contact TI for more information.
663	Flash Table Error	Flash Table Type 8 memory CRC failed.
664	Flash Table Error	Flash Table Type 9 memory access failed.
665	Source Error	External source measured active lines per frame does not match source definition.
666	Source Error	External source measured active pixels per line does not match source definition.
667 - 668	Internal Error	Internal system error. Contact TI for more information.
669	Source Error	External source Open LDI DLL could not lock to pixel clock.
670	Internal Error	Internal system error. Contact TI for more information.
671	Source Error	External source Open LDI port pixel clock frequency is not within the allowable range.
672	Source Error	External source Open LDI total pixels per line are unstable.
673	Internal Error	Internal system error. Contact TI for more information.
674	Source Error	External source parallel port pixel clock frequency is not within the allowable range.
675	Source Error	External source parallel port total pixels per line are unstable.
676	Source Error	External source pixel clock above the allowable range.
677	Source Error	External source pixel clock below the allowable range.
678	Source Error	External source VSYNC frequency greater than defined maximum.

Table A-2. Main Application Error Codes (continued)

CODE	NAME	DESCRIPTION
679	Source Error	External source VSYNC frequency less than defined minimum.
680 - 687	Internal Error	Internal system error. Contact TI for more information.
688	Source Error	Source selected is invalid (not external source, test pattern, or splash screen).
689 - 712	Internal Error	Internal system error. Contact TI for more information.
713	Clock Configuration Error	Attempted to enable spread spectrum on the DMD interface, but the spread spectrum enable signal is configured to keep it disabled.
714 - 733	Internal Error	Internal system error. Contact TI for more information.
734	Clock Configuration Error	Attempted to enable spread spectrum on the clock generator, but the spread spectrum enable signal is configured to keep it disabled.
735 - 742	Internal Error	Internal system error. Contact TI for more information.
743	Flash Configuration Error	Flash structure version does not match the structure version required by the main application.
744	Flash Configuration Error	Flash block type is not supported.
745	Flash Configuration Error	Flash table address is outside of valid range.
746	Flash Configuration Error	Flash block does not exist.
747 - 758	Internal Error	Internal system error. Contact TI for more information.
760	BIST Error	Video frame counter failed to lock due to value out of range.
761	Internal Error	Internal system error. Contact TI for more information.
762	Flash Error	Flash data verification CRC failed.
763 - 777	Internal Error	Internal system error. Contact TI for more information.
778	GPIO Error	Invalid pin access.
779	GPIO Error	Invalid pin direction.
780	GPIO Error	Invalid polarity type.
781 - 836	Internal Error	Internal system error. Contact TI for more information.
843	System Voltage Error	System 1.1V rail out of range.
844	System Voltage Error	System 1.8V rail out of range.
845	System Voltage Error	System 3.3V rail out of range.
846	System Voltage Error	ADC external VREF voltage out of range.
847	System Voltage Error	DVDD voltage out of range.
848	System Voltage Error	LDOT_M8 voltage out of range.
849	System Voltage Error	VMAIN voltage out of range.
850 - 868	Internal Error	Internal system error. Contact TI for more information.
869	Temperature Error	The temperature sensor remote channel is open. This is determined by a bit read from the temperature sensor status. This could be caused by a faulty connection between the DMD temperature sensing diode and the temperature sensor.
870-871	Internal Error	Internal system error. Contact TI for more information.
872	Temperature Error	Temperature sensor not present.
873 - 874	Internal Error	Internal system error. Contact TI for more information.
875	Temperature Error	Errors occurred during temperature read.
876	Temperature Error	Unknown temperature sensor device.
877 - 881	Internal Error	Internal system error. Contact TI for more information.
887	ADC Error	TPS99000-Q1 ADC interface measurement read error: A TPS99000-Q1 Data Reception error, or an ADC command control timeline violation error, or a TPS99000-Q1 input all HIGH, or TPS99000-Q1 disabled error detected.
888	DMD Interface Error	Attempt to park the DMD failed.
889	DMD Interface Error	Attempt to unpark the DMD failed.

Table A-2. Main Application Error Codes (continued)

CODE	NAME	DESCRIPTION
890	BIST Error	TPS99000-Q1 thermal conditions on chip have reached the warning level. If temperature continues to rise, system will reach die over-temp error temperature and emergency actions will be taken by TPS99000-Q1.
891	TPS99000-Q1 Interface Error	TPS99000-Q1 parity error on a main SPI port transaction occurred (command or write data) on previous command.
892	TPS99000-Q1 Error	TPS99000-Q1 top level state machine has changed states unexpectedly. Can be used to indicate to processor that TPS99000-Q1 has exited DISPLAY state due to a fault.
893	ADC Error	TPS99000-Q1 ADC error bit was set. This can occur due to ADC measurement saturation, measurement underflow, or parity errors on the ADC interface.
894	DMD Device ID Error	DMD device ID does not match flash project setting.
895	Temperature Error	Temperature is lower than the minimum DMD park temperature.
896	Temperature Error	Temperature is higher than the maximum DMD park temperature.
897	Internal Error	Internal system error. Contact TI for more information.
898	BIST Error	DMD clock out of range.
899	Internal Error	Internal system error. Contact TI for more information.
900	Temperature Error	DMD is parked and temperature function indicates that DMD is not allowed to unpark.
901	Flash Configuration Error	TPS99000-Q1 product ID does not match value in flash. Note that the product ID is different than the TPS99000-Q1 version number.
908 - 910	Internal Error	Internal system error. Contact TI for more information.
911	Flash Read Error	Attempted to read flash data, but the flash block has been erased.
912	Internal Error	Internal system error. Contact TI for more information.
913	Source Error	External source has been lost. This may occur as a result of source-lost monitoring, or may be triggered due to other external source test failures such as video checksums.
914	Source Error	External source has been lost and attempt to transition to alternate source failed.
915	BIST Error	Average Picture Level exceeded its specified limit.
916	Internal Error	Internal system error. Contact TI for more information.
917	BIST Error	Excessive brightness detector indicates an over-bright fault condition.
918-919	Internal Error	Internal system error. Contact TI for more information.
920	Command Error	TPS99000-Q1 illumination sync control setting is out of range.
934	Command Error	TPS99000-Q1 drive mode setting write parameter out of range.
935	Command Error	TPS99000-Q1 ADC configuration setting write parameter out of range.
936	Command Error	TPS99000-Q1 illumination sync control setting write parameter out of range.
938	BIST Error	TPS99000-Q1 checksum error in group 1 register set.
939	BIST Error	TPS99000-Q1 checksum error in group 2 register set.
940	BIST Error	TPS99000-Q1 checksum error in group 3 register set.
941	BIST Error	Either no ACK was received from DMD after sending requested transfer for DMD low-speed interface validation or a DMD register compare error was detected.
942 - 945	Internal Error	Internal system error. Contact TI for more information.
946	Source Error	Expected VSYNC signal did not arrive in the time allocated.
947-948	Internal Error	Internal system error. Contact TI for more information.
949	BIST Error	DMD VRESET voltage reading is out of range.
950	BIST Error	DMD VOFFSET voltage reading is out of range.
951	BIST Error	DMD VBIAS voltage reading is out of range.

Table A-2. Main Application Error Codes (continued)

CODE	NAME	DESCRIPTION
952	ADC Error	Sequence aborted so the ADC measurement data could not be read.
953	Source Error	The flash data defining the source to be displayed when the external source is lost contains invalid settings (test pattern, splash, system mode, or VSYNC frequency).
954	BIST Error	Multi-bit ECC memory error detected.
955	BIST Error	Single-bit ECC memory error detected.
956 - 960	Internal Error	Internal system error. Contact TI for more information.
961	BIST Error	Sequence clock ratio is not within the specified limits.
962 - 966	Internal Error	Internal system error. Contact TI for more information.
967	BIST Error	DMD reset instruction watchdog error.
968	BIST Error	Sequencer instruction watchdog error.
969	BIST Error	Frame memory buffer swap watchdog error.
970 - 971	Internal Error	Internal system error. Contact TI for more information.
972	BIST Error	Register CRC mismatched.
973 - 974	Internal Error	Internal system error. Contact TI for more information.
983 - 999	Internal Error	Internal system error. Contact TI for more information.
1000	Source Error	Vertical blanking error. Most likely caused by insufficient vertical front porch.
1001 - 1006	Internal Error	Internal system error. Contact TI for more information.
1007	BIST Error	DMD low speed interface parity error.
1008	System Voltage Error	ADC external bandgap voltage out of range.
1009 - 1010	Internal Error	Internal system error. Contact TI for more information.
1011	BIST Error	TPS99000-Q1 watchdog 1 did not detect a rising edge within the expected trigger window.
1012	BIST Error	TPS99000-Q1 watchdog 2 did not detect a rising edge within the expected trigger window.
1013 - 1014	Internal Error	Internal system error. Contact TI for more information.
1015	System Voltage Error	Driver power voltage is out of range.
1016	Internal Error	Internal system error. Contact TI for more information.
1017 - 1022	Internal Error	Internal system error. Contact TI for more information.
1023	Internal Error	Splash thread has terminated on an error.
1024 - 1025	Internal Error	Internal system error. Contact TI for more information.
1026	Flash Configuration Error	TPS99000-Q1 version does not match value in flash.
1027 - 1067	Internal Error	Internal system error. Contact TI for more information.
1068	Command Error	Diagnostic memory interface received an invalid command
1069	Command Error	Excessive brightness BIST cannot be command because the functionality is set to internal mode in flash
1070	BIST Error	DAC ADC Loop BIST: Current Control DAC error
1071	BIST Error	DAC ADC Loop BIST: Photo Feedback DAC error
1072	BIST Error	DAC ADC Loop BIST: Overbright DAC error
1073	Internal Error	Internal system error. Contact TI for more information.
1077 - 1078	Internal Error	Internal system error. Contact TI for more information.
1079	BIST Error	The provided current limit percentage threshold was not within 0-100.
1080	BIST Error	Photo feedback BIST failed.
1081	TPS99000-Q1 Interface Error	TPS99000-Q1 SPI transaction detected a read of swapped bytes.
1082	Internal Error	Internal system error. Contact TI for more information.
1083	PWM Temperature management error	DMD and PWM Temp management are both enabled and are using the same PWM GPIO

Table A-2. Main Application Error Codes (continued)

CODE	NAME	DESCRIPTION
1084	PWM Temperature management error	PWM Temperature management duty cycle setting is outside range of 0-100%. Verify LUT in the flash is correct.
1085	PWM Temperature management error	PWM Temperature mangement output selection invalid.
1086	PWM Temperature management error	PWM Temperature management temperature source invalid. Please refer to Section 8.2.33
1087	PWM Temperature management error	PWM Temperature failed to set PWM output.
2011 - 2043	Internal Error	Internal system error. Contact TI for more information.

Revision History

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

Changes from Revision D (March 2019) to Revision E (July 2024)	Page
• Updated to include DLPC231 DLP462x-Q1 family of chips; Updated for inclusive language	15
• Updated headlight system block diagram	16
• Updated for inclusive terminology	22
• Added in a new section explaining the different types of processing modes supported by the DLPC23x-Q1 when combined with a diamond configuration.	43
• Added important note for resolution compatibility for DLP462xx-Q1 DMDs	46
• Added resolutions supported for .46" DMDs	50
• Added PWM Temperature Management Function	61
• Added commands 35h-39h	114
• Added PWM Temperature Management Enable - Write (35h)	135
• Added PWM Temperature Management Enable - Read (36h)	136
• Added PWM Temperature Management Source - Write (37h)	136
• Added PWM Temperature Management Source - Read (38h)	136
• Added PWM Temperature Management Duty Cycle - Read (39h)	136
• Added Device ID for DLP462XX-Q1 (0.46")	167
• Added Fuse ID error code 154; Added error codes 1085-1087	191

IMPORTANT NOTICE AND DISCLAIMER

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

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

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

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

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

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