User's Guide

DLP® LightCrafter™ Display 230NP EVM

ABSTRACT

This user's guide presents an overview of the DLP® LightCrafter™ Display 230NP evaluation module (EVM) and a general description of the main features and functions. It explains the first steps to get started, and shows a detailed description of onboard LEDs, connectors, and overall EVM assembly.

The EVM can be used by itself to display a DLP LightCrafter Display test image or in combination with a low-end processor including the Raspberry PI 4B.

This guide provides the user with further information on how to successfully connect the EVM to the Raspberry PI 4B and get the Raspberry PI content projected on the wall.

Table of Contents

List of Figures

List of Tables

Trademarks

LightCrafter™, TI E2E™, are trademarks of Texas Instruments.
DLP® are registered trademarks of Texas Instruments.
Microsoft® and Windows® are registered trademarks of Microsoft Corporation.
All trademarks are the property of their respective owners.

1 DLP® LightCrafter™ Display 230NP EVM Overview

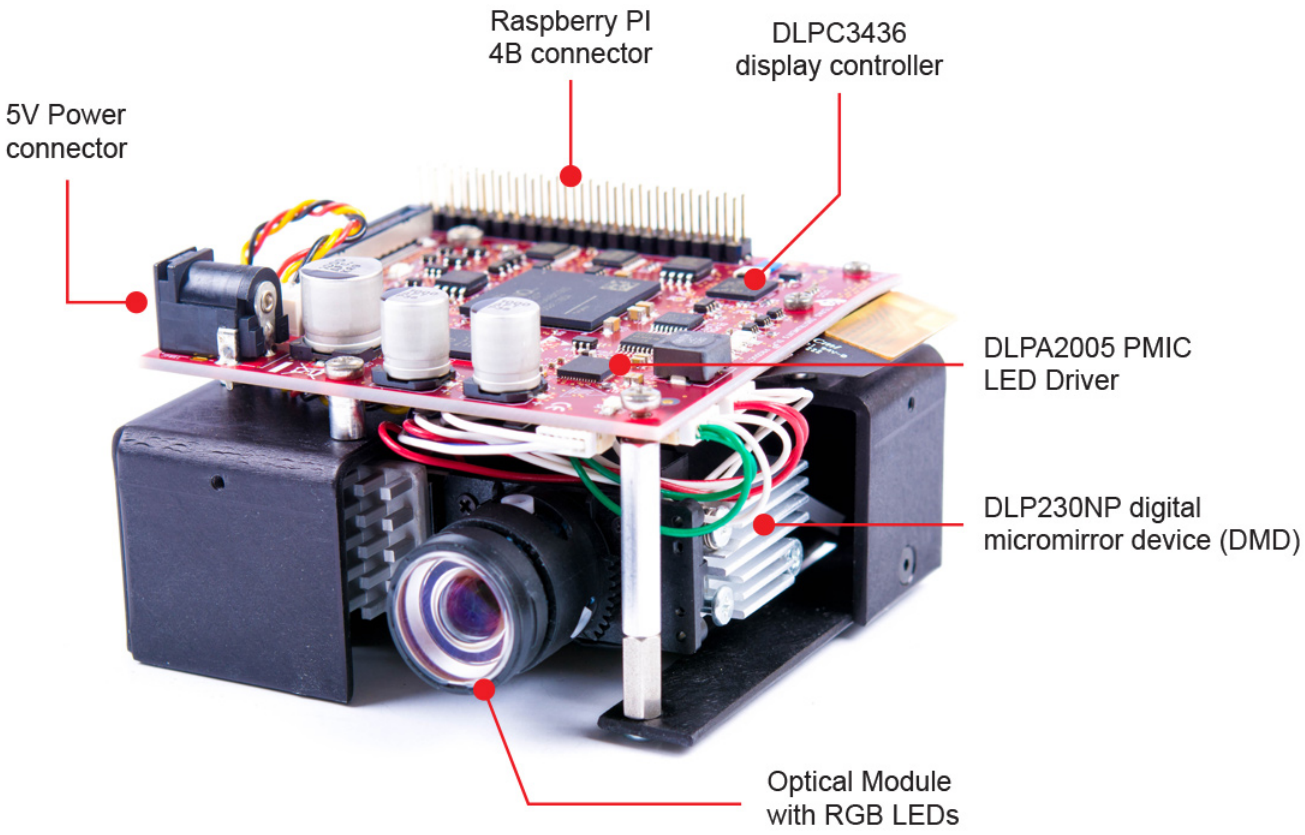

Figure 1-1. DLP® LightCrafter™ Display 230NP EVM

In addition to this document, use the documents shown in Section 3.

2 Safety Instructions

CAUTION

Caution hot surface. Contact may cause burns. Do not touch.

WARNING

Possible hazardous optical radiation emitted from this product. Do not stare at the operating lamp. May be harmful to the eye.

WARNING

Observe handling precautions. Electrostatic sensitive devices.

WARNING

Always ensure both fans are running during operation to avoid overheating and ensure reliable operation.

3 Applicable Documents

The following documents are applicable to the DLP LightCrafter Display 230NP EVM and are available at TI.com (www.ti.com).

- *DLP230NP (0.23 1080p) Digital Micromirror Device (DMD) Data Sheet*
- *DLPA2005 Power Management and LED/Lamp Driver IC Data Sheet*
- *DLPC3436 Display Controller Data Sheet*
- *DLPC3436 Software Programmer's Guide*

For further assistance, see the *DLP Products and MEMS TI E2E™ community support forums.*

4 What is in the DLP® LightCrafter™ Display 230NP EVM?

The DLP LightCrafter Display module consists of two subsystems:

- Light engine – includes the optics, red, green, and blue LEDs, and a 960 × 540 DMD which displays 1920 × 1080 pixels on the screen.
- Driver board – includes the DLP chipset comprised of the DLPC3436 Controller and DLPA2005 PMIC/LED driver.
- The DLP Light Crafter Display 230NP EVM is designed for a max. ambient temperature of 35°C.

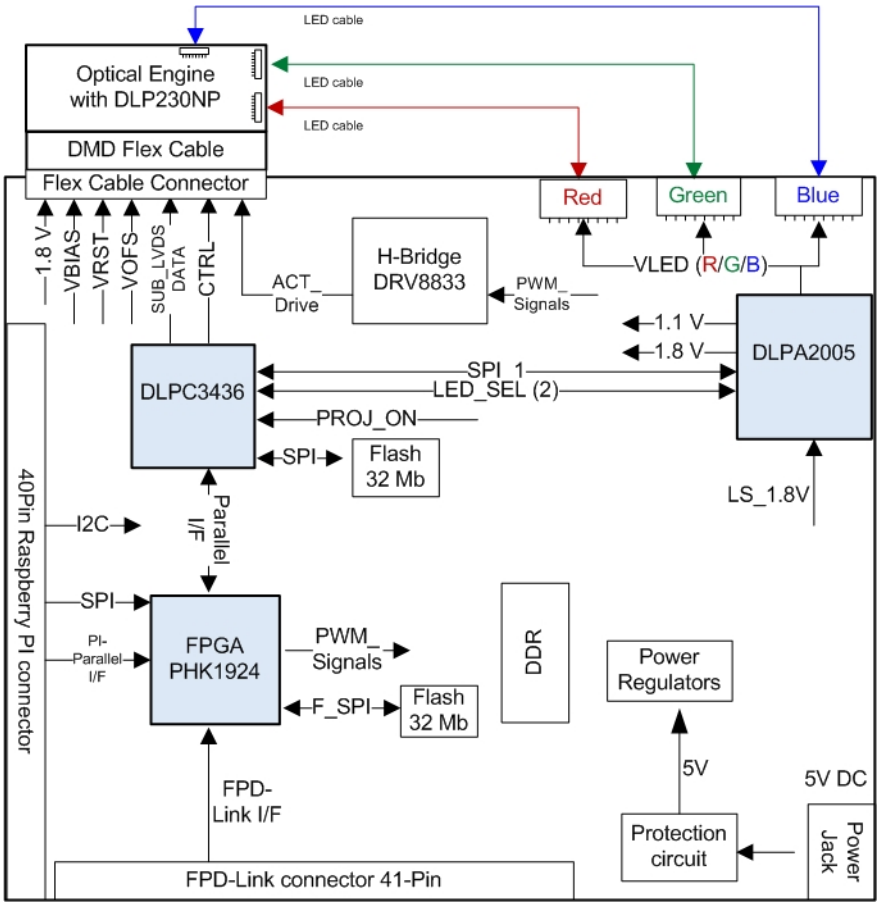

Figure 4-1. DLP® LightCrafter™ Display EVM Block Diagram

5 Light Engine

The optical engine in the EVM is developed by Young Optics and is production ready. The light engine consists of the following components:

- DLP230NP (0.23-inch 1080p DMD)
- OSRAM red (KR CSLNM1.23_EN), green (KP CSLNM1.F1_EN), and blue (KB CSLNM1.14_EN) LED
- This light engine interfaces with the EVM using DMD pin mapping **Option 1**. For more information about the DMD interface, see the DLPC3436 data sheet.

For more information on the LEDs used in this optical engine, contact OSRAM through email using the following mailing list: dlp-pico-osram@list.ti.com

Table 5-1. Optical Engine Specifications

Parameter	MIN	TYP	MAX	Unit
Brightness at Red 2-A, Green 2-A, Blue 2-A LED Current (Wx = 0.29, Wy-0.33)		100		Lm
Red / Green / Blue LED Current		2		A
Brightness Uniformity (JBMA specification)	85%			
Throw Ratio (at a 70" screen)		1.2		
Offset		100%		
Focus Range (Optimized at 70")	30		120	inch
Image Diagonal Size	30		120	inch

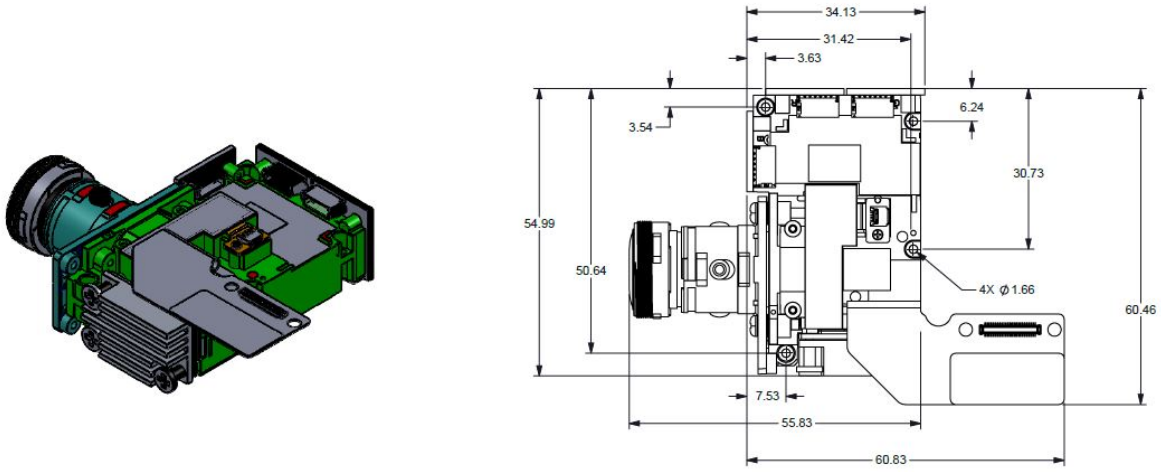

Figure 5-1. Optical Engine

6 Quick-Start Procedure

This quick-start assumes the default EVM condition as shipped.

1. In case a Raspberry PI is available, proceed with connecting the DLPDLCR230NPEVM to the Raspberry PI. Ensure that the Rasperry Pi has been setup as described in Section 9. If the Raspberry PI is currently not needed, proceed with step 2.

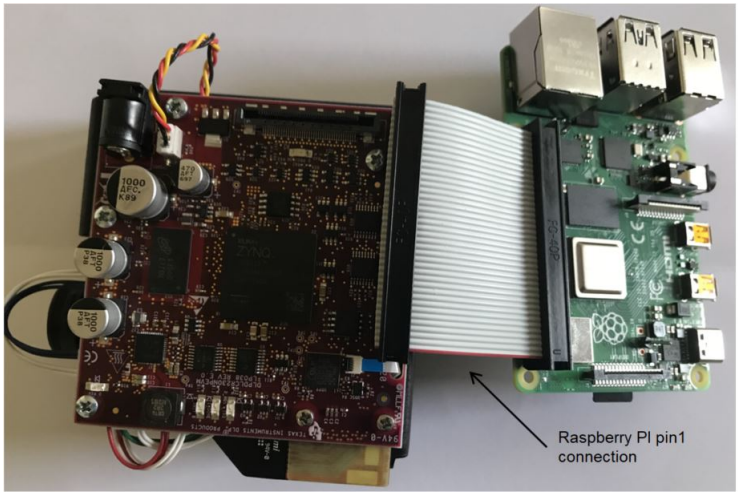

Figure 6-1. DLPDLCR230NPEVM With Raspberry PI

2. Power up the DLP LightCrafter Display 230NP EVM by applying an external DC power supply (5 V DC) to the JPWR1 connector.The D_P5V LED will indicate that the 5-V DC power supply has been applied to the system.
   External Power Supply Requirements:
   - Nominal Output Voltage: 5 VDC
   - Minimum Output Current: 3 A; Maximum Output Current: 4 A
   - Efficiency Level: VI

   NOTE: TI recommends using an external power supply that complies with applicable regional safety standards such as UL, CSA, VDE, CCC, PSE, and so forth.
3. The system will automatically drive Proj-ON high when the 5-V DC power supply is applied and the fan is plugged in. If the fan is not plugged in or a loose connection is present Proj-ON will not be driven high. The DLPDLCR230NPEVM includes a input voltage monitor which drives Proj-ON low if the input voltage drops below approximately 4.65 V for any reason during operation.
4. After the DLP LightCrafter Display 230NP EVM is turned on; the projector will default to displaying a DLP logo followed by a LightCrafter Display test pattern image. The D_HOST_IRQ will be OFF which indicates a succesfull boot of the DLPC3436. The D_DONE and D_INIT_B will be ON to indicate a successful boot-up of the FPGA.

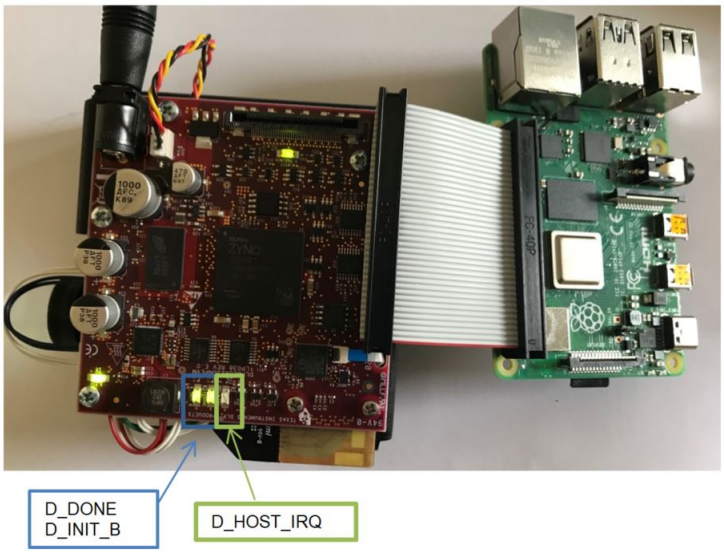

Figure 6-2. DLPDLCR230NPEM LED Indications

5. The focus of the image can be adjusted manually on the optical engine.

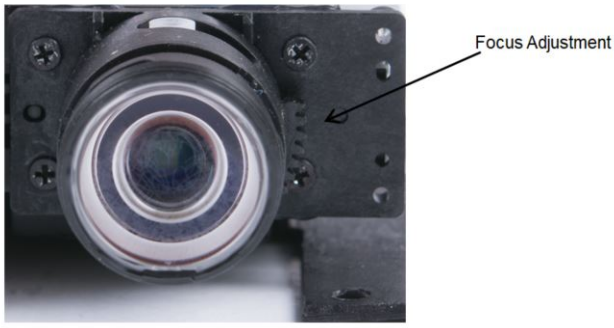

Figure 6-3. Optical Engine With Focus Adjustment

6. Ensure that Raspberry has been configured correctly as described in Section 9 before applying power to the Raspberry PI via the USB Type C cable.
7. Power down with the Raspberry PI connected:
   - Drive PROJ_ON LOW via Raspberry PI GPIO 25 (refer to Section 9)
   - Shutdown the Raspberry PI (refer to Section 9)
   - Remove power from the Raspberry PI by removing the USB Type C cable
   - Remove the power adapter to the DLPDLCR230NPEVM
8. Power down without Raspberry PI connected:
   The projector can be turned off by just removing the power cable. A voltage monitor on the EVM will detect the input voltage go below approximately 4.65 V and will drive Proj-ON low automatically.

There are five indicator LEDs on the DLP LightCrafter Display 230NP EVM, and they are defined in Table 6-1:

Table 6-1. LEDs on the DLP® LightCrafter™ Display 230NP EVM

LED Reference	Signal Indication	Description
D_HOST_IRQ	HOST_IRQ	ON during DLPC3436 boot, OFF when projector is running. Indication of DLPC3436 boot-up completed and ready to receive commands
D_PROJ_ON	PROJ_ON	PROJ_ON signal is HIGH
D_INIT_B	INT_B	ON when FPGA initialization is completed. OFF indicates that the FPGA is in RESET or a configuration error occured.
D_DONE	DONE	ON when FPGA configuration is completed.
D_P5V	P5V	Input voltage 5 V applied

7 Connectors on Formatter Board

Table 7-1. Installed Connectors on Formatter Board

Installed Connectors and Headers	Description
JPWR1	Connector for 5-V external power supply interface
J1	Connector for optical engine flex cable.
J2	Connector (40 pin) for Raspberry PI cable
J3	Connector (41 pin) for FPD Link interface (not installed by default)
J4	Connector for 5-V cooling fan
J500	Connector for DMD interface flex cable
J501	Connector for Green LED cable
J502	Connector for Red LED cable
J503	Connector for Blue LED cable

8 EVM Setup

The DLP LightCrafter Display 230NP EVM is composed of two main parts:

- DLPDLCR230NPEVM formatter board
- Engine with LED connection, flex cable, and mechanical setup

The formatter board contains the connector for the power supply, the connector for the fan, and the Raspberry PI ribbon cable connector. It also contains a protection circuit which prevents the power up of the DLP chipset when no fan is connected to the board. This also applies if the connection is loose or in case a fan wire is broken. Figure 8-1 shows the DLPDLCR230NPEVM formatter board with the main connections:

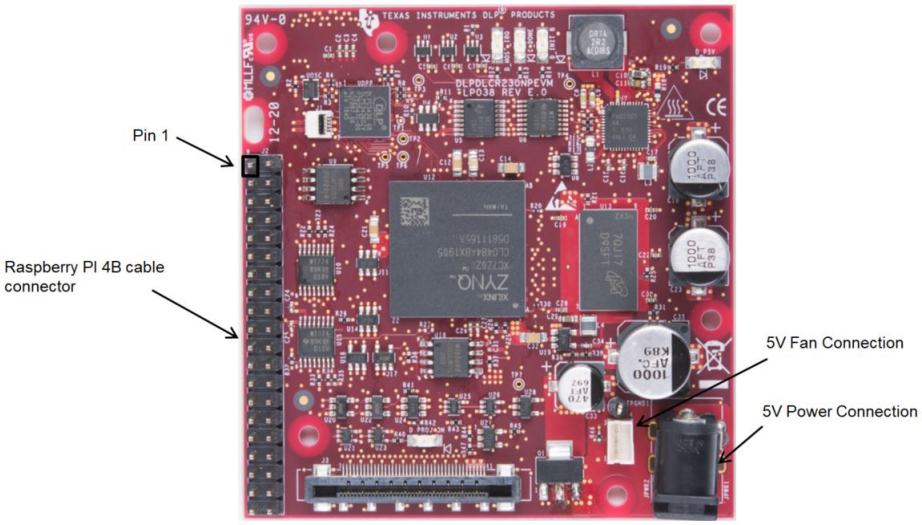

Figure 8-1. DLP® LightCrafter™ 230NP EVM Formatter Board

The light engines contain the LED connectors and the flex cable which connects to the formatter board on the bottom via J500. Figure 8-2 shows the optical engine connections.

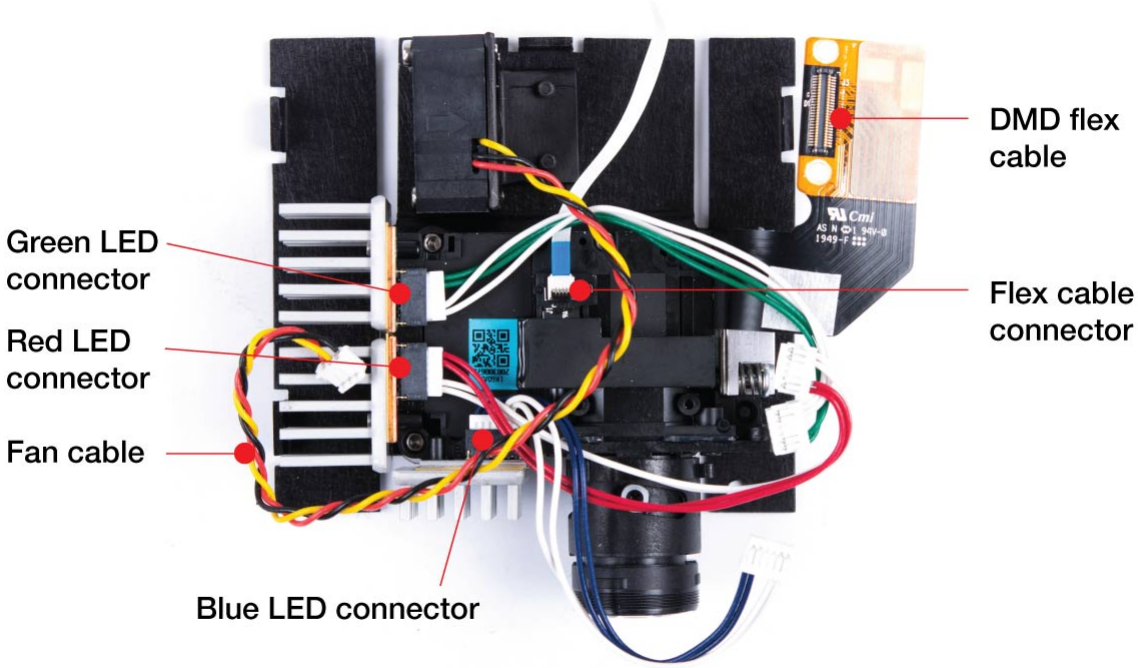

Figure 8-2. DLPDLCR230NPEVM Optical Engine Connection

The DLPDLCR230NPEVM formatter board is mounted on top of the mechanical base. The fan cable has to be connected all the way to ensure proper connection of the fan wires to the PCB. Figure 8-3 shows the formatter board mounted on the mechanical assembly with all cables connected.

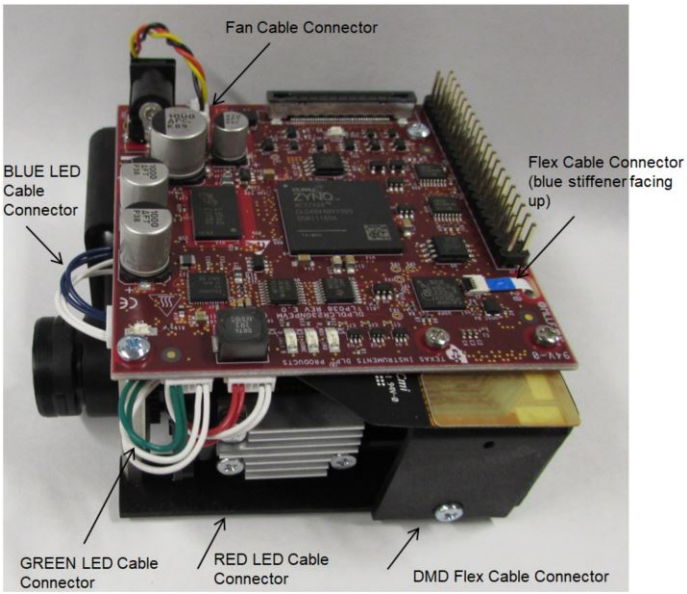

Figure 8-3. DLP® LightCrafter™ Display 230NPEVM

Ensure a good connection of the flex cable, fan and LED cables to the DLP LightCrafter Display 230NPEVM formatter board before turning it on.

9 Raspberry Pi Guide

This section provides instructions for bringing up the DLPDLCR230NPEVM using the Raspberry Pi. It is limited to the basics of installation and communication with the system. Basics of SD card setup for the Raspberry Pi are covered, along with use of the Python API library for the EVM. The function and utility of each sample script in the Python library is also discussed.

9.1 Raspberry Pi General Configuration

Before starting, ensure that all of the following hardware materials are available:

- DLPDLCR230NPEVM (1 ×)
- DLPDLCR230NPEVM Ribbon Cable (1 ×, included with system)
- Raspberry Pi 4B (or compatible)
- microSD card (16GB, or any size compatible with Raspberry Pi 4B)
- microSD card Adapter (for use with PC)
- PC with SD card slot (for writing operating system image)

To begin, the microSD card should be programmed with an installation of Raspbian. An image can be downloaded from the Raspberry Pi website (https://www.raspberrypi.org/downloads/raspberry-pi-os/).

Please consult the Raspberry Pi documentation for information on how to install an operating system image (https://www.raspberrypi.org/documentation/installation/installing-images/README.md) [1]

[1] raspberrypi.org is third party content (“Third Party Content”). Third Party Content is not under TI’s control and TI is not responsible for such content, or any changes or updates to such content. TI provides links and access to Third Party Content to you only as a convenience and TI does not endorse such content. Use of third party websites, features, and tools is governed by the applicable terms of use and privacy practices of such websites and services. You agree to review and accept applicable terms in respect of Third Party Content.

Once Raspbian is installed to your SD card, you can log into the device over SSH (https://www.raspberrypi.org/documentation/remote-access/ssh/). TI recommends updating the internal software of Raspberry PI to the latest version before proceeding.

The software support package is provided on TI.com (https://www.ti.com/product/DLP230NP), and contains a number of important components which are used to configure and operate the Raspberry Pi. These components are as follows:

- Python 3 API Library (containing EVM compatible host I2C commands, as well as Python-compatible I2C drivers)
- Python 3 Example Scripts (implements above API library in example applications such as test pattern cycle, LED test, and more)
- "config.txt" Example File (provides TI-tested video timing configurations for use with the DLPDLCR230NPEVM)

TI recommends configuring the "config.txt" file used by the Raspberry Pi to define video timings first, before proceeding. This file can either be accessed within the Raspberry Pi terminal (with root level access at /boot/config.txt), or by shutting down the Raspberry Pi, removing the SD card, and accessing the SD card contents from a PC. For a quick setup, the "config.txt" provided can be used to overwrite the existing configuration. The elements of the "config.txt" file modified are as follows:

- Initialization of Raspberry Pi 4B GPIO pins (BCM 0-27). These pins are configured to function as inputs on boot, and are reconfigured as outputs once the Python initialization script is executed.
- Initialization of supported function overlays. Functions supported include 18-bit DPI (RGB666), I2C (software-based), SPI (to write to EVM flash device). DPI (video output) must be disabled while using the SPI function, as they share access to common GPIO lines (BCM 8-11).
- Configuration of 1920 × 1080, 60-Hz video output via 18-bit DPI lines (RGB666). This configuration can be modified to tune the input resolution and framerate, within acceptable limits (defined in Section 9.2).

The Raspberry Pi configuration file (located at /boot/config.txt) can be modified either directly with root access, or via the raspi-config utility. Because the configuration settings used by the DLPDLCR230NPEVM do not fit within the typical use-case settings offered by this graphical user interface, it is recommended that the configuration be manually edited for this system. A copy of the example configuration information from "config.txt" is located in the following script. Only the sections of the configuration which are relevant to the DLPDLCR230NPEVM are shown:

```
################################################################################
# Initialization of supported function overlays.
# Functions supported include 18-bit DPI (RGB666), I2C (software-based), SPI.
# Uncomment some or all of these to enable the optional hardware interfaces
#dtparam=i2c_arm=on
#dtparam=i2s=on dtparam=spi=on
# Configure Raspberry PI for SSH over USB
dtoverlay=dwc2
# Configure I2C on GPIO Pins #22 and #23
dtoverlay=i2c-gpio,i2c-gpio_sda=23,i2c_gpio_scl=22,i2c_gpio_delay_us=2
################################################################################
# Initialization of Raspberry Pi 4B GPIO pins (BCM 0-27).
# Configure DPI on GPIO Pins #0 through #21
gpio=0=op
gpio=0=pn
gpio=1-27=ip
gpio=1-27=pn
# Enable DPI18 Overlay
enable_dpi_lcd=1
display_default_lcd=1
dpi_group=2
dpi_mode=87
################################################################################
# Configuration of 1920 x 1080, 58-60 Hz video output via 18-bit DPI lines (RGB666).
# RGB 666 CFG 1 (MODE 5)
dpi_output_format=458773
# 58 Hz Timings, Works at GPIO Drive Strength 5 to 7
hdmi_timings=1920 0 20 10 10 1080 0 10 10 10 0 0 0 58 0 125000000 3
################################################################################
```

The full "config.txt" file can be accessed from the SD card using a Microsoft® Windows® or OS X computer. See https://www.raspberrypi.org/documentation/configuration/config-txt/ for more information on the configuration file /boot/config.txt (and other Raspberry Pi configuration information). For starters, it is recommended to copy from the provided example "config.txt" file provided in the software support package on TI.com as an initial configuration. Modifications to this configuration can be made to support a particular hardware setup of Raspberry Pi.

9.2 Video Timing Configuration

The Raspberry Pi supports customizable video timing configurations over its DPI video output, as the configuration file (config.txt) in the "hdmi_timings" setting shows. To achieve the highest quality video output via this interface, TI recommends modifying the video timings provided based on the particular system use case for the Raspberry Pi, and the GPIO drive strengths associated with the GPIO bank of the Raspberry Pi.

The video timing configuration of the Raspberry Pi is documented on the Raspberry Pi website at (https://www.raspberrypi.org/documentation/hardware/raspberrypi/dpi/README.md). The syntax required to input a desired video timing is provided in Section 9.1. This includes both the video output timings (such as resolution and framerate) as well as the output format settings. The GPIO drive strength on the Raspberry Pi is configurable from 0 (2 mA) to 7 (16 mA) per pin. TI recommends using the minimum drive strength that still provides enough current capacity to meet the desired video timings. More information on the GPIO pads control is found at (https://www.raspberrypi.org/documentation/hardware/raspberrypi/gpio/gpio_pads_control.md).

Using the 2" ribbon cable provided with the DLPDLCR230NPEVM, a video output of 1920 × 1080 resolution from 58 to 61 Hz can be achieved. Video output configurations are limited by the maximum pixel clock (PCLK) which can be supported by the Raspberry Pi 4B over its GPIO interface. Table 9-1 lists the video timing configurations (minimum and maximum frame rate) evaluated by TI:

Table 9-1. Raspberry Pi Video Timing Settings for DLPDLCR230NPEVM (With Provided Ribbon Cable)

	Configuration Field	Minimum Framerate Timing	Maximum Framerate Timing
Output Format Settings	Output Format	5 (18-Bit RGB666, CFG 1)	5 (18-Bit RGB666, CFG 1)
	RGB Order	1 (R-G-B)	1 (R-G-B)
	Output Enable Mode	1	1
	HSYNC Disable	0	0
	VSYNC Disable	0	0
	Output Enable Disable	0	0
	HSYNC Polarity	0	0
	VSYNC Polarity	0	0
	Output Enable Polarity	0	0
	HSYNC Phase	1	1
	VSYNC Phase	1	1
	Output Enable Phase	1	1
Hardware Settings	GPIO Drive Strength	5	7
Horizontal Settings	Active Pixels	1920	1920
	Sync Polarity	0	0
	Front Porch	20	20
	Sync Pulse	10	10
	Back Porch	10	10
Vertical Settings	Active Lines	1080	1080
	Sync Polarity	0	0
	Front Porch	10	10
	Sync Pulse	10	10
	Back Porch	10	10
	Sync Offset A	0	0
	Sync Offset B	0	0
Other Settings	Pixel Rep	0	0
	Frame Rate	58 Hz	61 Hz
	Interlaced	0	0
	Pixel Frequency	125 MHz	132 MHz
	Aspect Ratio Setting	3	3

9.3 Python Support Software

The Python software package contains an API, as well as example scripts implementing basic features of the DLPDLCR230NPEVM. The directory structure of this package is as follows:

- **/api/dlpc343x_xpr4.py** (Contains I2C host commands)
- **/api/dlpc343x_xpr4_evm.py** (Contains EVM-specific commands such as GPIO initialization)
- **/api/ScriptAPIDoc.html** (Contains documentation on all I2C host commands available in dlpc343x_xpr4.py API library)
- **/flash_write_controller.py** (Switches GPIO to SPI mode and writes to DLPC3436 flash device)
- **/flash_write_fpga.py** (Switches GPIO to SPI mode and writes to DLPC3436 FPGA flash device)
- **/i2c.py** (Driver code for I2C via Python)
- **/init_fpdlink_mode.py** (Initializes video output for FPD-Link)
- **/init_parallel_mode.py** (Initializes video output for 18-bit DPI via RGB666 from Raspberry Pi
- **/linuxi2c.py** (Driver code for I2C via Python)
- **/sample00_template.py** (Blank script, provides base for user development of custom routines)
- **/sample01_tpg.py** (Sample script, cycles internal test patterns available for the DLPDLCR230NPEVM)
- **/sample02_splash.py** (Sample script, cycles splash screens available for the DLPDLCR230NPEVM)
- **/sample03_display.py** (Sample script, tests display configuration; rotation, keystone, image flip)
- **/sample04_looks.py** (Sample script, cycles "Looks" (LED duty cycle configurations) available)
- **/sample05_led.py** (Sample script, iterates RGB LED current values from minimum to maximum)
- **/sample06_status.py** (Sample script, reads and prints status registers available)

To execute these scripts, Python version 3 or higher is required. Once the "config.txt" file on the Raspberry Pi is set up and the above Python software package is installed, you are ready to interface your Raspberry Pi with the DLPDLCR230NPEVM. Each of the above scripts includes docstrings describing code functionality within. TI recommends reviewing these docstrings to take full advantage of the provided system functionality. The included sample scripts are portable and can be executed on their own or incorporated into a larger codebase, provided the API libraries are also present or included. Simply call the script by invoking Python 3 on command line with the script being evaluated.

The "flash_write_controller.py" and "flash_write_fpga.py" scripts are unique in that they operate the outside of its standard operating mode. In these scripts, the GPIOs on the Raspberry Pi are reconfigured to establish a SPI connection between the Raspberry Pi and one of the DLPDLCR230NPEVM's onboard flash devices (DLPC3436 controller flash device or FPGA flash device). A flash image must be provided as command line arguments with these scripts, at which point the onboard GPIOs are set to enable writing to the directed flash device. A GPL software program **flashrom** is required to perform this write operation, and is installed by default on Raspberry Pi 4B systems. Learn more at (https://www.flashrom.org/Flashrom [2]).

Note

When executing any script which performs flash device memory access (including "Splash Screen Select" and "Look Select") it is necessary to enable I2C command delays of with a minimum duration of 0.6 seconds to ensure that no write or readback commands fail during code execution. See / sample02_splash.py for an example implementation of the I2C command delay. Within the provided Python sample scripts, this delay has been defined as "I2c_time_delay".

9.4 Operating Modes

Table 9-2 provides a list of the assigned GPIO configurations for each operating mode on the DLPDLCR230NPEVM. The "Controller Flash Write" and "FPGA Flash Write" configurations are employed when

[2] flashrom is third party content (“Third Party Content”). Third Party Content is not under TI’s control and TI is not responsible for such content, or any changes or updates to such content. TI provides links and access to Third Party Content to you only as a convenience and TI does not endorse such content. Use of third party websites, features, and tools is governed by the applicable terms of use and privacy practices of such websites and services. You agree to review and accept applicable terms in respect of Third Party Content.

using the "flash_write_controller.py" and "flash_write_fpga.py" scripts, respectively. All other scripts can and should use the default "DPI" configuration mode. In the cases where the GPIO pin should be driven by the Raspberry Pi, the particular pin is marked as (HIGH) or (LOW) in the "Pin Mode" column.

Table 9-2. Raspberry Pi GPIO Configurations

	DPI Configuration (RGB666 Only)		Controller Flash Write Configuration		FPGA Flash Write Configuration	
Broadcom Pin Number (BCM)	**Pin Mode**	**Pin Function**	**Pin Mode**	**Pin Function**	**Pin Mode**	**Pin Function**
0	ALT2	PCLK	OUT	N/A	OUT	N/A
1	ALT2	DATAEN	IN	N/A	IN	N/A
2	ALT2	VSYNC	IN	N/A	IN	N/A
3	ALT2	HSYNC	IN	N/A	IN	N/A
4	ALT2	B2	IN	N/A	IN	N/A
5	ALT2	B3	IN	N/A	IN	N/A
6	ALT2	B4	IN	N/A	IN	N/A
7	ALT2	B5	IN	N/A	IN	N/A
8	ALT2	B6	OUT	SPI_CE0	OUT	SPI_CE0
9	ALT2	B7	ALT0	SPI_MISO	ALT0	SPI_MISO
10	ALT2	G2	ALT0	SPI_MOSI	ALT0	SPI_MOSI
11	ALT2	G3	ALT0	SPI_SCLK	ALT0	SPI_SCLK
12	ALT2	G4	IN	N/A	IN	N/A
13	ALT2	G5	IN	N/A	IN	N/A
14	ALT2	G6	IN	N/A	IN	N/A
15	ALT2	G7	IN	N/A	IN	N/A
16	ALT2	R2	IN	N/A	IN	N/A
17	ALT2	R3	IN	N/A	IN	N/A
18	ALT2	R4	IN	N/A	IN	N/A
19	ALT2	R5	IN	N/A	IN	N/A
20	ALT2	R6	IN	N/A	IN	N/A
21	ALT2	R7	IN	N/A	IN	N/A
22	OUT	I2C-SCL	IN	N/A	IN	N/A
23	OUT	I2C-SDA	IN	N/A	IN	N/A
24	IN	SPI_SEL_ASIC	OUT (HIGH)	SPI_SEL_ASIC	IN	SPI_SEL_ASIC
25	OUT (HIGH)	MODE_SEL	IN	MODE_SEL	IN	MODE_SEL
26	IN	PROJ_ON	OUT (HIGH)	PROJ_ON	OUT (HIGH)	PROJ_ON
27	IN	SPI_SEL_FPGA	IN	SPI_SEL_FPGA	OUT (HIGH)	SPI_SEL_FPGA

A brief summary of pin functions used follows:

- **PCLK, DATAEN, VSYNC, HSYNC** - DPI video control signals.
- **R2-R7, G2-G7, B2-B7** - DPI video data signals. Bits 0 and 1 are unused due to video format being limited to **RGB666**.
- **I2C-SCL, I2C-SDA** - Software-based I2C bus, allows Raspberry Pi host to send commands to the DLPDLCR230NPEVM (current settings, test pattern, video modes, and so forth.)
- **SPI_SEL_ASIC, SPI_SEL_FPGA** - Active-high select line when writing to FPGA or controller flash device. **Only one of these should be active at a time**.
- **MODE_SEL** - Active-high DPI buffer enable signal. **Must be driven high when outputting video signal**. During flash write, leave floating (input)
- **PROJ_ON** - Active-low system power signal. Drive high to turn off the DLPDLCR230NPEVM without needing to remove power supply connection. **Drive low or leave floating (input) to keep system powered on.**
- **SPI_CE0, SPI_MISO, SPI_MOSI, SPI_SCLK** - Raspberry Pi SPI bus signals. **Used only for writing to FPGA and controller flash devices.**

CAUTION

Do not attempt to enable SPI functionality or drive SPI_SEL_ASIC or SPI_SEL_FPGA enable pins while DPI video output is active. Enabling access to the SPI bus (and thereby controller/FPGA flash devices) while DPI video output is active may inadvertently erase or corrupt the onboard flash memory of the DLPDLCR230NPEVM.

CAUTION

When writing to the flash device, only ONE of either the "SPI_SEL_ASIC" or "SPI_SEL_FPGA" lines should be active, or driven high at a time. Attempting to drive both SPI select lines can lead to hardware failure during the flash write operation.

9.5 Example Applications

9.5.1 Initialize Communication Between Raspberry Pi and EVM

Start the system from an untouched boot state. Plug in the DLPDLCR230NPEVM, followed by the Raspberry Pi via USB to the laptop. After a few moments the USB connection should initialize to the laptop and communication via remote SSH may begin. The Raspberry Pi starts out at its home directory. For reference, this is located at:

/home/pi/

Navigate to the directory which includes the downloaded Python scripts. In this example, the following directory is assumed:

/home/pi/Documents/dlp

Execute the following to navigate down each directory:

$ cd /home/pi/Documents/dlp

Back up navigation by one directory can be achieved with following command:

$ cd ..

Executing the “ls -l” command to ask the Pi to print out the contents of the current directory:

$ ls -l

To initialize the default Raspberry Pi to EVM connection, execute the initialization script via the following:

$ python3 init_parallel_mode.py

Dialogue will print in the terminal while the script is working. Upon completion of this script the Raspberry Pi desktop should be visible on the EVM’s projector output.

9.5.2 Play Video Content From Raspberry Pi

To begin, navigate to the base directory of the DLPDLCR230NPEVM python scripts and initialize the parallel video mode. Assuming that the python scripts have been loaded to:

/home/pi/Documents/dlp

Execute:

$ cd /home/pi/Documents/dlp

$ python3 init_parallel_mode.py

From here, display of any video content or images from the Raspberry Pi is possible. For example, at:

/home/pi/Documents/dlp/media

Navigate to the desired directory. It is recommended to use the list command to see what content is available:

$ ls -l

This folder is empty by default. To display video content, simply execute the following command (replace the file name with the desired video file):

$ omxplayer <filename>.mp4

The video will play in a letterbox by default. To stretch the video to play in fullscreen (regardless of the resolution) use the following command instead:

$ omxplayer --aspect-mode stretch <filename>.mp4

Press CTRL+C to exit the video playback.

9.5.3 Execute Sample Scripts Using Raspberry Pi

Various sample scripts are provided with the Python API library for use with the DLPDLCR230NPEVM. Assuming that the python scripts have been loaded to:

/home/pi/Documents/dlp

Execute:

$ cd /home/pi/Documents/dlp

It is recommended to use the list command to see what scripts are available:

$ ls -l

To run a given script, execute the following command:

$python3 sample01_tpg.py

or

$python3 <script name>

Only the flash download scripts require the user to provide arguments besides the name of the script itself (in which case the name of the firmware image being downloaded must be provided). In all other cases the script may be executed without any arguments. Many scripts will cycle through various operational modes. In some scripts, the terminal will hold and wait for user input before proceeding to the next step. If such a prompt appears, simply press ENTER on your terminal keyboard to proceed.

There are some situations where the user may want to modify small aspects of the performance or behavior of a script. To perform these modifications, the given script can be opened in an editor by executing the following command:

$ sudo nano sample01_tpg.py

or

$ sudo nano <script name>

The arrowkeys can be used to navigate through a document. Upon completion of edits, use the “CTRL+X” command followed by “Y” then “ENTER” to save your changes and exit nano. For a more thorough description of the many functions of the nano consult the relevant documentation:

(https://www.nano-editor.org/docs.php) [3]

[3] nano is third party content (“Third Party Content”). Third Party Content is not under TI’s control and TI is not responsible for such content, or any changes or updates to such content. TI provides links and access to Third Party Content to you only as a convenience and TI does not endorse such content. Use of third party websites, features, and tools is governed by the applicable terms of use and privacy practices of such websites and services. You agree to review and accept applicable terms in respect of Third Party Content.

9.5.4 Rewrite Controller or FPGA Flash Device Using Raspberry Pi

It is possible to write to the Flash device of the controller (or the FPGA) without turning off the system. Note that the following can be executed even after the system has already been initialized to external video mode. Each of the three scripts mentioned in this tutorial always revert the Raspberry Pi GPIO pins to inputs before any action is taken to prevent damage to the EVM hardware. Assuming that the python scripts have been loaded to:

/home/pi/Documents/dlp

Execute:

$ cd /home/pi/Documents/dlp

To upload a flash image to the EVM, it should be available on the Raspberry Pi in an accessible directory. From here, you should execute the following:

$ python3 flash_write_controller.py dlpdlcr230np_controller_7p3p9.img

or

$ python3 flash_write_controller.py <Flash Image Name>

This will write the existing image file (dlpdlcr230np_controller_7p3p9.img) to the EVM's controller flash device. This name can be substituted for any file which has the ***.img** file type extension.

```
pi@raspberrypi:~/Documents/Pi $ python3 flash_write_controller.py dlpdlcr230np_7p3p8.img
set slave address: 27
Initializing Raspberry Pi Default Settings for DLPC3436...
DO NOT disconnect EVM or Raspberry Pi during flash write...
Resetting GPIO control pins...
Setting GPIO drive strength to 5 (0-7, 3 default)...
Putting DLPC3436 to sleep...
Initializing SPI bus...
Connecting to DLPC3436 flash device...
Writing image to DLPC3436 flash device... (DO NOT DISCONNECT EVM OR POWER DOWN SYSTEM)
flashrom  on Linux 4.19.97-v7l+ (armv7l)
flashrom is free software, get the source code at https://flashrom.org

Using clock_gettime for delay loops (clk_id: 1, resolution: 1ns).
Found Macronix flash chip "MX25U3235E/F" (4096 kB, SPI) on linux_spi.
Reading old flash chip contents... done.
Erasing and writing flash chip...
Warning: Chip content is identical to the requested image.
Erase/write done.
Resetting GPIO control pins...
```

Figure 9-1. DLPDLCR230NPEVM Flash Write Procedure

Note that in the case above, the image being used matches what is already on the flash device (which is why the warning is issued). In this case no write occurs. To write to the FPGA, execute the following instead:

$ python3 flash_write_fpga.py dlpdlcr230np_fpga_v1p1.bin

or

$ python3 flash_write_fpga.py <FPGA Binary Name>

To perform an FPGA flash write, the given binary must be in the ***.bin** file format.

10 Troubleshooting

In case any issues are being encountered while using the DLP LightCrafter Display 230NP EVM, the following frequently ask questions may help resolving the issue.

- **D_PROJ_ON LED is not coming ON after applying power to the DLP LightCrafter Display 230NP EVM**. Check the FAN connection on J4 and ensure all wires have good contact to the formatter board. In case the FAN is not running or a bad connection of the FAN to the formatter board is present the PROJ_ON signal will not be driving HIGH.
- **Video output in external video mode has jitter or is otherwise noisy**. Check the GPIO drive strength and video timing configuration settings on the Raspberry Pi. If the GPIO drive strength is too low for a given video timing, or the pixel clock implemented is too high, then the video output will not look good when sent by the Raspberry Pi.
- **Flash device has been accidentally rewritten with an incorrect firmware image**. The EVM will not boot in this case, but may be recovered. execute the "flash_write_controller.py" and "flash_write_fpga.py" scripts with the default firmware images to reprogram the system with factory default settings.
- **System is unresponsive after switching to splash screen**. Many commands, including "Splash Screen Select" and "Look Select" commands, cause a memory access to occur between the controller (DLPC3436) and flash memory. In the case that I2C commands are sent when this memory access has not completed, I2C bus hangup may occur. When executing any script which performs flash device memory access, it is necessary to enable I2C command delays of with a minimum duration of 0.6 seconds to ensure that no write or readback commands fail during code execution.
- **Raspberry PI video output is black after being idle**. The Raspberry Pi has a black screensaver enabled by default. If this is the case in the particular version being used with the system, it is recommended to disable this screensaver in the Raspberry Pi user settings if desired.

11 Support Resources

TI E2E™ support forums are an engineer's go-to source for fast, verified answers and design help — straight from the experts. Search existing answers or ask your own question to get the quick design help you need.

Linked content is provided "AS IS" by the respective contributors. They do not constitute TI specifications and do not necessarily reflect TI's views; see TI's Terms of Use.

12 Revision History

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

Changes from Revision * (October 2020) to Revision A (March 2021)	Page
• Added contact information for OSRAM	2

IMPORTANT NOTICE AND DISCLAIMER

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

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

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

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