DLP Technology Boot Diagnostic Tool Guide

Sept. 26, 2023

Table of Contents

Introduction

This tool is designed to allow a user to self-diagnose issues that may occur with the DLPC34XX chipset during the bootup sequence. Currently users have no visibility into the bootup sequence and must engage with DLP technology engineers over E2E, Jira, or email to diagnose any problems with the system booting up. This tool would grant more visibility into the process and enable the user to more easily diagnose bootup issues in a more streamlined process.

Diagnostic Tool Specifications

- Supported DLP chip controllers
  - DLPC3430, DLPC3432, DLPC3433, DLPC3434, DLPC3435, DLPC3436, DLPC3426, DLPC3437, DLPC3438, DLPC3439, DLPC3440, DLPC3470, DLPC3478, DLPC3479, DLPC3420, DLPC3421
- Supported Interfaces
  - UART 1.8V, UART 2.5V

Setting up the Diagnostic Tool

Flashing the Image

The included diagnostic tool image file ("DLPC34XX_BootDiag") must be flashed onto the SPI flash device for the controller. On a dual controller system, the tool image can be flashed onto both controllers, or just one of the controllers. Note: The boot diagnostic image must be flashed with an alternate tool, the flash device cannot be written to with DLP technology software until the system has successfully booted up. A SPI host adapter such as the Cheetah adapter from Total Phase is recommended to program the flash memory.

Selecting an Output Pin

Prior to powering on the system, an output pin for UART communication must be selected on each controller by pulling GPIOs 1 and 2 to 1.8V or ground. Since these GPIOs are also used as SPI pins to communicate with the PMIC, they should be pulled up or down with a resistor greater than 5-k Ω. The different output pins are shown below.

GPIO 2	GPIO 1	Output Pin
0 V	0 V	GPIO 6
0 V	1.8 V	GPIO 9
1.8 V	0 V	GPIO 19
1.8 V	1.8 V	Host IRQ

Note: In a dual controller system, the GPIO pins and Host IRQ pin are unique to each controller, requiring you to select an output pin for each controller if diagnosing each. Additionally, the output HIGH levels are 1.8V for the GPIO pins and 2.5V for Host IRQ.

Reading the Diagnostic Tool Output

Before running the diagnostic tool, the controller(s) must be powered off and a method to read UART must be connected to the output pin. One method of doing so is to use the TTL-232RG-VREG1V8 cable (USB driver required) from FTDI Chip in combination with a serial port terminal such as Tera Term. With this configuration, connect GND and RXD on the cable to ground and the selected output pin on the

system. On Tera Term or a similar serial port terminal, open the COM port associated with the cable and set the speed to 115200 bits per second, data to 8 bit, parity to none, and stop bit to none.

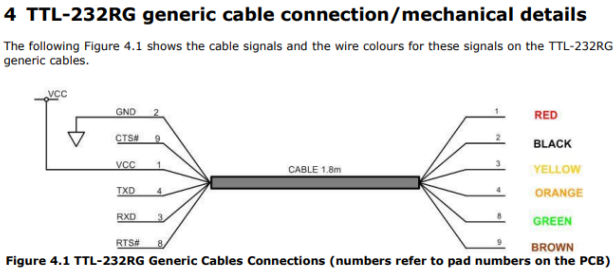
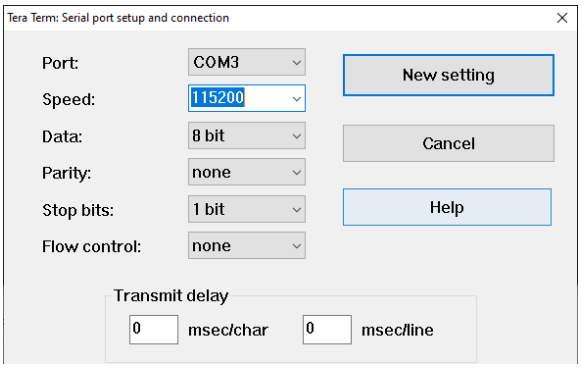


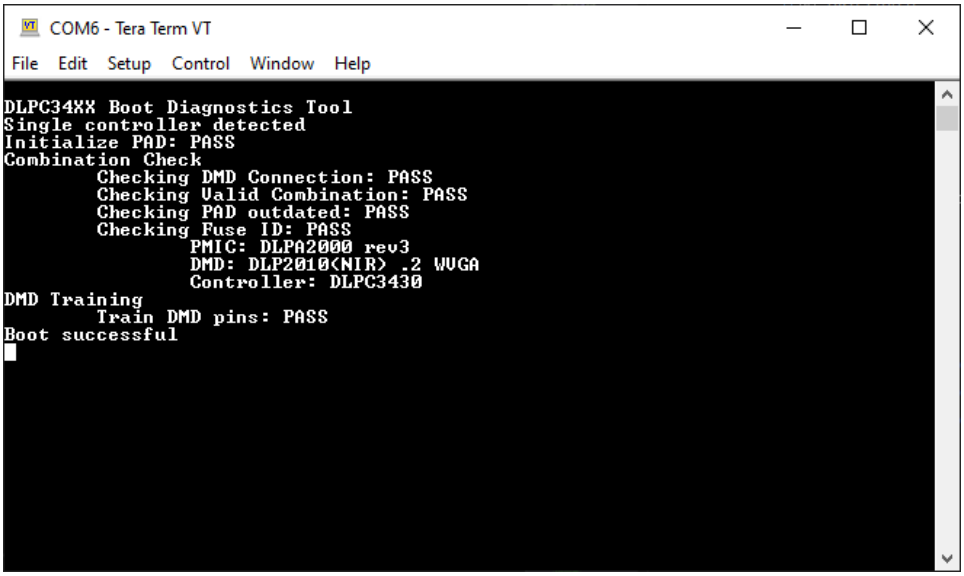
Figure 4.1 TTL-232RG Generic Cables Connections (numbers refer to pad numbers on the PCB)



With the tool connected, the system can be powered on. Depending on the controller being used, the tool will run different checks. A check fails if it explicitly says “FAIL” or if the tool does not produce an output for the check.

Single Controller System (Non XPR)

Single controller non-XPR systems include the DLPC3430, DLPC3432, DLPC3433, DLPC3435, DLPC3438, DLPC3470, DLPC3478, DLPC3420, and DLPC3421. The checks done by the tool are shown below.

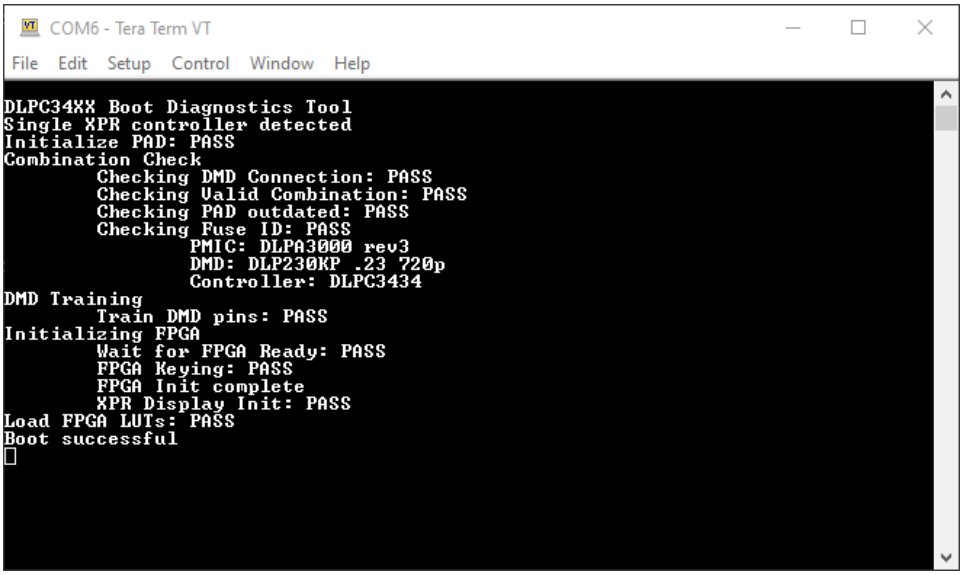


Check	Description
Initialize PMIC	The controller communicates with the PMIC and writes values into the registers on the PMIC. A failure at this point could indicate a poor connection in the communication lines between the controller and the PMIC.

Checking DMD Connection	The controller attempts to read the DMD ID. A failure at this point could indicate either a poor connection with the DMD I2C bus or a lack of power in the DMD.
Checking Valid Combination	The combination of the controller, DMD, and PMIC are checked. A failure indicates that the combination is invalid. Note: The combination check only checks the *size* of the DMD and not the specific DMD
Checking PMIC outdated	The revision of the PMIC is checked to ensure it is not outdated. A failure at this point means a newer revision of the PMIC must be used.
Checking Fuse ID	The specific DMD is checked against the controller and the PMIC. A failure at this point indicates an invalid combination of DMD, PMIC, and controller.
Train DMD pins	The DMD pins are trained to accommodate a difference in speed due to the flex cable.

Single Controller System (XPR)

Single controller XPR systems include the DLPC3434 and DLPC3436. The checks done by the tool are shown below.



Check	Description
Initialize PMIC	The controller communicates with the PMIC and writes values into the registers on the PMIC. A failure at this point could indicate a poor connection in the communication lines between the controller and the PMIC.
Checking DMD Connection	The controller attempts to read the DMD ID. A failure at this point could indicate either a poor connection with the DMD I2C bus or a lack of power in the DMD.
Checking Valid Combination	The combination of the controller, DMD, and PMIC are checked. A failure indicates that the combination is invalid. Note: The

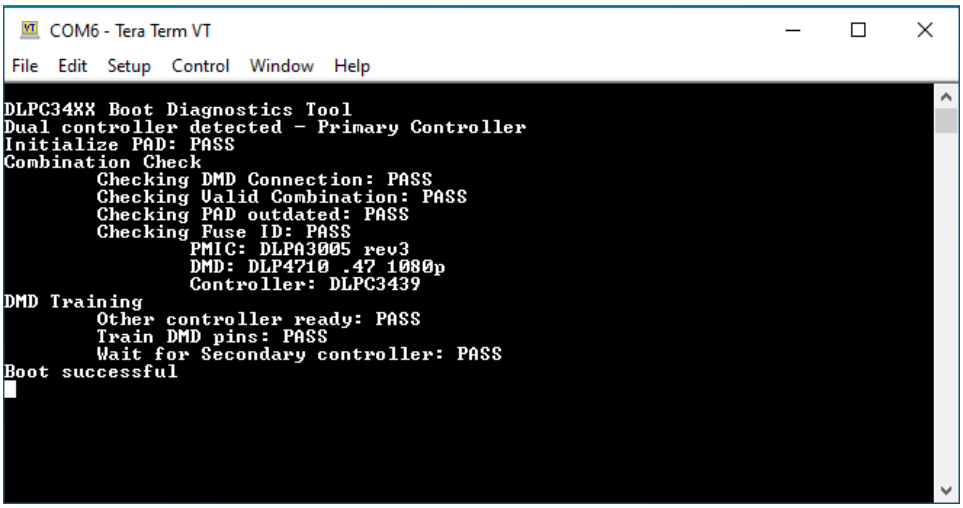
	combination check only checks the *size* of the DMD and not the specific DMD
Checking PMIC outdated	The revision of the PMIC is checked to ensure it is not outdated. A failure at this point means a newer revision of the PMIC must be used.
Checking Fuse ID	The specific DMD is checked against the controller and the PMIC. A failure at this point indicates an invalid combination of DMD, PMIC, and controller.
Train DMD pins	The DMD pins are trained to accommodate a difference in speed due to the flex cable.
Wait for FPGA Ready	The controller waits for the FPGA Ready signal (GPIO 14) to go high. A failure at this point means either a poor connection into GPIO 14 or a failure in the FPGA bootup.
FPGA Keying	The controller attempts to unlock the FPGA with a key. Note: a failure at this point does NOT cause the bootup sequence to fail.
XPR Display Init	The controller writes to the FPGA.
Load FPGA LUTs	The controller writes to the FPGA.

Dual Controller System (Non XPR)

Dual controller non-XPR systems include the DLPC3439, DLPC3440, and DLPC3479. Since this is a dual controller system, the checks differ between the primary and the secondary controller.

Primary Controller

The checks for the primary controller on a dual controller non-XPR system are shown below.

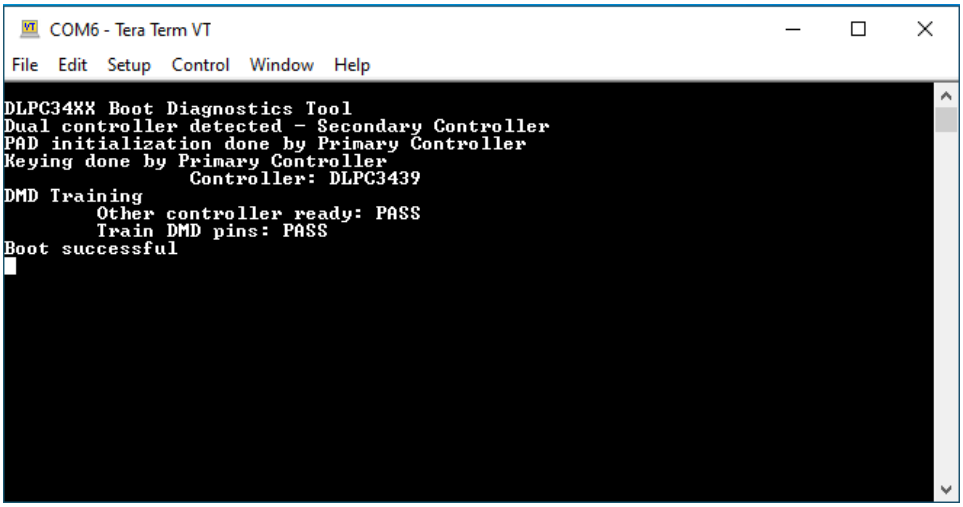


Check	Description
Initialize PMIC	The controller communicates with the PMIC and writes values into the registers on the PMIC. A failure at this point could indicate a poor connection in the communication lines between the controller and the PMIC.

Checking DMD Connection	The controller attempts to read the DMD ID. A failure at this point could indicate either a poor connection with the DMD I2C bus or a lack of power in the DMD.
Checking Valid Combination	The combination of the controller, DMD, and PMIC are checked. A failure indicates that the combination is invalid. Note: The combination check only checks the *size* of the DMD and not the specific DMD
Checking PMIC outdated	The revision of the PMIC is checked to ensure it is not outdated. A failure at this point means a newer revision of the PMIC must be used.
Checking Fuse ID	The specific DMD is checked against the controller and the PMIC. A failure at this point indicates an invalid combination of DMD, PMIC, and controller.
Secondary controller ready	Waits until Secondary controller sync pin goes low (GPIO 16 on secondary controller and GPIO 18 on primary controller). Note: this may pass even with a failure in the secondary controller.
Train DMD pins	The DMD pins are trained to accommodate a difference in speed due to the flex cable. The controllers also synchronize throughout the training. A failure at this point may indicate a failure in the secondary controller.
Wait for Secondary controller	The controller waits for the secondary controller to complete its DMD training. Failure could indicate secondary controller has not passed DMD training or a faulty sync signal between the controllers

Secondary Controller

The checks for the secondary controller on a dual controller non-XPR system are shown below.



Check	Description
Controller Sync	The primary controller attempts to sync with the secondary controller. If there is a failure, the secondary controller could be missing, have a poor connection, have a corrupt flash, or have GPIO 4 designate both controllers as the primary controller. GPIO 4 must be pulled up on the primary controller and pulled down on the secondary controller

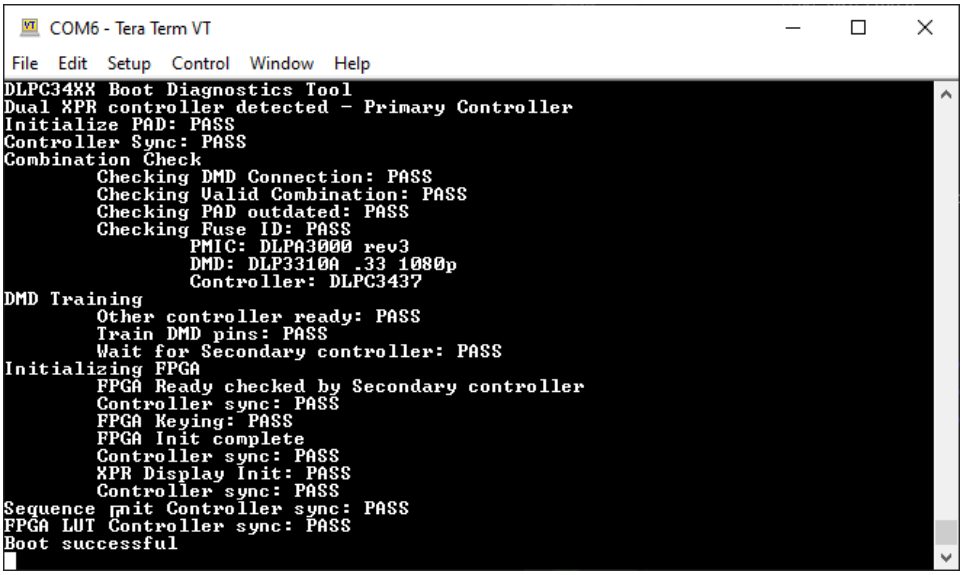
Secondary controller ready	Waits until Primary controller sync pin goes low (GPIO 16 on primary controller and GPIO 18 on secondary controller). Note: this may pass even with a failure in the primary controller.
Train DMD pins	The DMD pins are trained to accommodate a difference in speed due to the flex cable. The controllers also synchronize throughout the training. A failure at this point may indicate a failure in the secondary controller.

Dual Controller System (XPR)

Dual controller XPR systems include the DLPC3437. Since this is a dual controller system, the checks differ between the primary and the secondary controller.

Primary Controller

The checks for the primary controller on a dual controller XPR system are shown below.

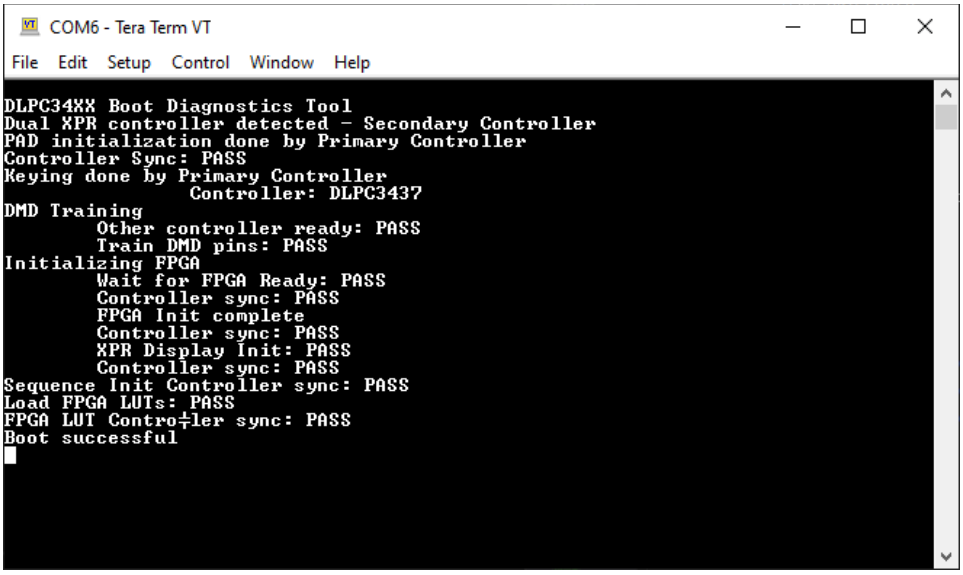


Check	Description
Initialize PMIC	The primary controller communicates with the PMIC and writes values into the registers on the PMIC. A failure at this point could indicate a poor connection in the communication lines between the controller and the PMIC.
Controller Sync	The primary controller attempts to sync with the secondary controller. If there is a failure, the secondary controller could be missing, have a poor connection, have a corrupt flash, or have GPIO 4 designate both controllers as the primary controller.
Checking DMD Connection	The controller attempts to read the DMD ID. A failure at this point could indicate either a poor connection with the DMD I2C bus or a lack of power in the DMD.
Checking Valid Combination	The combination of the controller, DMD, and PMIC are checked. A failure indicates that the combination is invalid. Note: The

	combination check only checks the *size* of the DMD and not the specific DMD
Checking PMIC outdated	The revision of the PMIC is checked to ensure it is not outdated. A failure at this point means a newer revision of the PMIC must be used.
Checking Fuse ID	The specific DMD is checked against the controller and the PMIC. A failure at this point indicates an invalid combination of DMD, PMIC, and controller.
Secondary controller ready	Waits until Secondary controller sync pin goes low (GPIO 16 on secondary controller into GPIO 18 on primary controller). Failure could indicate the controller hasn't finished initializing or signal error.
Train DMD pins	The DMD pins are trained to accommodate a difference in speed due to the flex cable.
Wait for Secondary controller	The controller waits for the secondary controller to complete its DMD training.
Controller sync	This controller sync is meant to make the primary controller wait until the secondary controller receives the FPGA Ready signal. If it fails at this point, the FPGA may have failed to initialize or failed to signal the secondary controller.
FPGA Keying	The controller attempts to unlock the FPGA with a key. Note: a failure at this point does NOT cause the bootup sequence to fail. Ensure fpga is compatible.
Controller sync	The controller notifies the secondary controller that the FPGA Keying is complete.
XPR Display Init	The controller writes to the FPGA.
Controller sync	The controllers sync after completing the XPR Display initialization
Sequence Init Controller sync	The controllers sync after completing the sequence initialization. The tool image doesn't actually initialize any sequences, this sync is only to allow the other controller to proceed. If it fails at this point, the secondary controller failed the sequence initialization.
FPGA LUT Controller sync	The controller waits for the secondary controller to write to the FPGA. A failure at this point indicates the secondary controller failed to write the LUTs.

Secondary Controller

The checks for the secondary controller on a dual controller XPR system are shown below.



Check	Description
Controller Sync	The controller attempts to sync with the secondary controller. If there is a failure, the secondary controller could be missing, have a poor connection, have a corrupt flash, or have GPIO 4 designate both controllers as the primary controller.
Secondary controller ready	Waits until Primary controller sync pin goes low (GPIO 16 on primary controller into GPIO 18 on secondary controller)
Train DMD pins	The DMD pins are trained to accommodate a difference in speed due to the flex cable.
Wait for FPGA Ready	The secondary controller waits for the FPGA ready signal (GPIO 12) to go high. A failure at this point means either a poor connection into GPIO 12 or a failure in the FPGA bootup.
Controller sync	This controller sync is meant to make the primary controller wait until the secondary controller receives the FPGA Ready signal.
FPGA Keying	The controller attempts to unlock the FPGA with a key. Note: a failure at this point does NOT cause the bootup sequence to fail.
Controller sync	The controller waits for the primary controller to perform FPGA keying.
XPR Display Init	The controller writes to the FPGA.
Controller sync	The controllers sync after completing the XPR Display initialization.
Sequence Init Controller sync	The controllers sync after completing the sequence initialization. The tool image doesn't actually initialize any sequences, this sync is only to allow the other controller to proceed. If it fails at this point, the primary controller failed the sequence initialization.
Load FPGA LUTs	The secondary controller writes to the FPGA
FPGA LUT Controller sync	The controller notifies the primary controller that it completed writing the FPGA LUTs

Additional Information

The tool will output additional information that may be helpful. All controller types will output the type of system the controller is intended for and dual controllers will output whether it has been configured as the primary or secondary controller. Single controller systems and the primary controller in a dual controller system will output the detected controller, DMD, and PMIC connected. If the system uses the DLPC3436 or DLPC3426, it will output whether the actuator mode uses PWM or DAC. Finally, the tool will print the phrase “Boot successful” at the end if the bootup sequence is complete.

Further Help

If the tool successfully completes the boot sequence but issues are still present with the full image, possible issues and/or solutions are listed below:

1) Ensure the image file being used matches the detected controller, DMD, and PMIC combination
2) The tool does not account for errors in the PMIC that shut down and/or reset the system. Try scoping the power rail outputs from the PMIC and ensure they are at the intended levels.

If issues are still present, contact a DLP technology engineer using E2E:

1) Please provide the following information in your post/support request.
    a. A screenshot of the boot diagnostic tool output
    b. Any noticeable behavior from the system running the original image, i.e. flickering
    c. Any signals scoped on the system running the original image. Useful signals to scope:
        i. HostIRQ out of the controller
        ii. INTZ out of the PMIC (PARKZ into the controller)
        iii. RESETZ out of the PMIC/into the controller

IMPORTANT NOTICE AND DISCLAIMER

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

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

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