

# TI Designs

## Camera Reference Design for ARM® Cortex®-M Microcontrollers (MCUs)



### Description

This design implements a network camera with a quarter video graphics array (QVGA) display panel and an embedded web server for remote monitoring. With a memory footprint of 250 KB for the web server, there is ample memory available on the TM4C129x microcontroller (MCU) for additional customizations.

### Resources

<a href="#">TIDM-TM4C129CAMERA</a>	Design Folder
<a href="#">TM4C129XNCZAD</a>	Product Folder
<a href="#">SN74ALVC244</a>	Product Folder
<a href="#">SN74LVC2G00</a>	Product Folder
<a href="#">SN74LVC1G32</a>	Product Folder
<a href="#">TPS61042</a>	Product Folder
<a href="#">TPS62177</a>	Product Folder
<a href="#">TPS2051B</a>	Product Folder
<a href="#">LM4819</a>	Product Folder
<a href="#">TMP100</a>	Product Folder
<a href="#">REF3230</a>	Product Folder
<a href="#">DK-TM4C129X</a>	Tools Folder



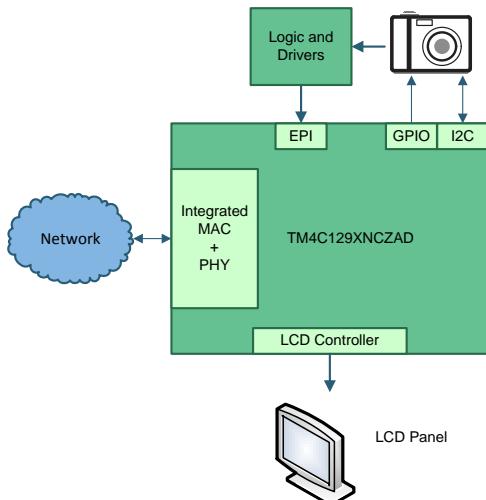
[ASK Our E2E Experts](#)

### Features

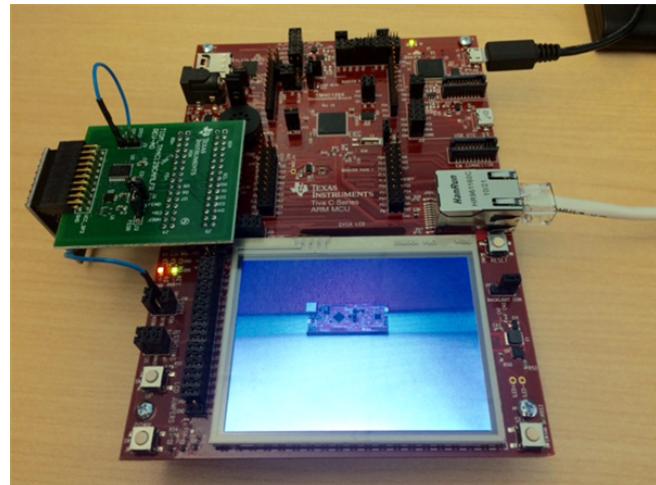
- Supports QVGA Size Image Capture Using External Peripheral Interface (EPI) and Direct Memory Access (DMA) Transfers to Reduce CPU Overhead for Data Transfer
- Embedded Web Server for Remote Monitoring of Image Capture With Periodic Self Reload of Web Page
- Camera Interface Supports up to 30 Frames Per Second for Display on LCD Panel
- Uses TivaWare™ Software Graphics Library for Graphics-Rich User Interface and Experience

### Applications

- Home Security Solutions
- Video Doorbells
- Network Cameras
- Building Automation and Surveillance



Copyright © 2016, Texas Instruments Incorporated





An IMPORTANT NOTICE at the end of this TI reference design addresses authorized use, intellectual property matters and other important disclaimers and information.

## 1 System Description

The TM4C129x family of MCUs from Texas Instruments feature an integrated Ethernet PHY and MAC with cryptographic modules and a large number of serial and parallel interfaces for control and sensor data acquisition. Integrating a camera would require a dedicated camera interface or an external programmable logic like field programmable gate array (FPGA) and complex programmable logic device (CPLD) to interface a MCU to the camera. This design uses the EPI peripheral with a logic gate for protocol conversion without needing an external FPGA or CPLD. Using EPI with the logic gate significantly reduces the complexity of interfacing the camera to the TM4C129x device family while providing the advantage of integrated PHY that allows network connectivity for a host of video monitoring applications. The design files include schematics, bill of materials (BOM), layer plot, Altium files, Gerber files, and embedded firmware.

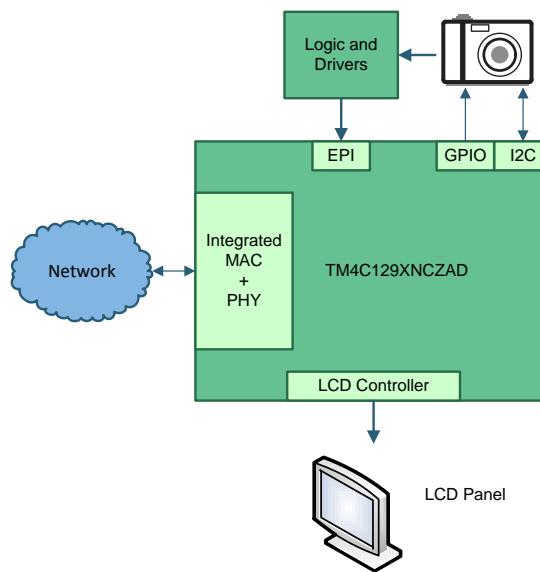
### 1.1 Key System Specifications

**Table 1. Key System Specifications**

PARAMETER	SPECIFICATION
Image capture width	320 pixels
Image capture depth	240 pixels
Image capture format	16 BPP (RGB565 format)
Raw camera clock	20 MHz
Display panel width	320 pixels
Display panel depth	240 pixels
Image frame rate	30 frames per second

## 2 System Overview

### 2.1 Block Diagram



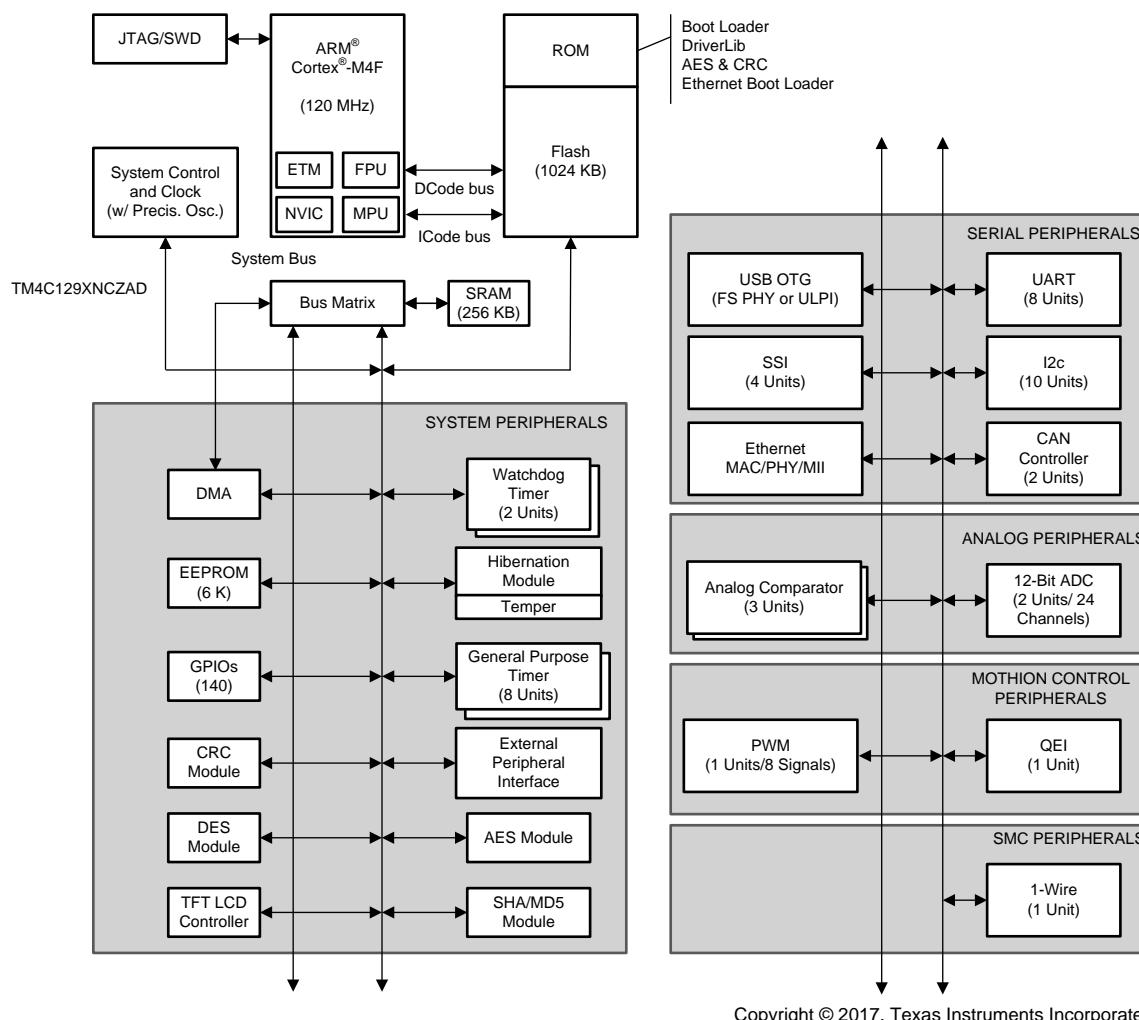
Copyright © 2016, Texas Instruments Incorporated

**Figure 1. Camera Reference Design With TM4C129x Block Diagram**

## 2.2 Highlighted Products

### 2.2.1 TM4C129XNCZAD

TM4C129XNCZAD is a 120-MHz high-performance MCU with 1-MB on-chip flash, 256-KB on-chip SRAM integrated Ethernet MAC PHY for connected applications and cryptographic modules such as advanced encryption standard (AES), data encryption standard (DES) and secure hash algorithm (SHA) for encryption, decryption, data validation, and authentication. The device has high-bandwidth interfaces like a memory controller (EPI) and a high-speed USB 2.0 digital interface. With integration of a number of serial communication peripherals, a 12-bit analog-to-digital converter (ADC) capable of up to 4 MSPS and motion control peripherals. The device provides a unique solution for a variety of applications ranging from industrial communication equipment to smart energy and smart grid applications.



Copyright © 2017, Texas Instruments Incorporated

**Figure 2. TM4C129XNCZAD MCU High-Level Block Diagram**

### 2.2.2 SN74ALVC244

The SN74ALVC244 is organized as two four-bit line drivers with separate output-enable (OEn) inputs. When OEn is low, the device passes data from the A inputs to the Y outputs. When OEn is high, the outputs are in the high-impedance state. This device provides extremely low-propagation delay of 2.8 ns (maximum) when operating at 3.3 V.

### 2.2.3 SN74LVC2G00

The SN74LVC2G00 is a dual two-input positive NAND gate designed for low-propagation delay over a wide supply range of 1.65 to 5 V.

## 3 System Design Theory

This section will elaborate how the TM4C129x MCU is interfaced to the external camera module using the EPI module.

### 3.1 Camera Module Interface

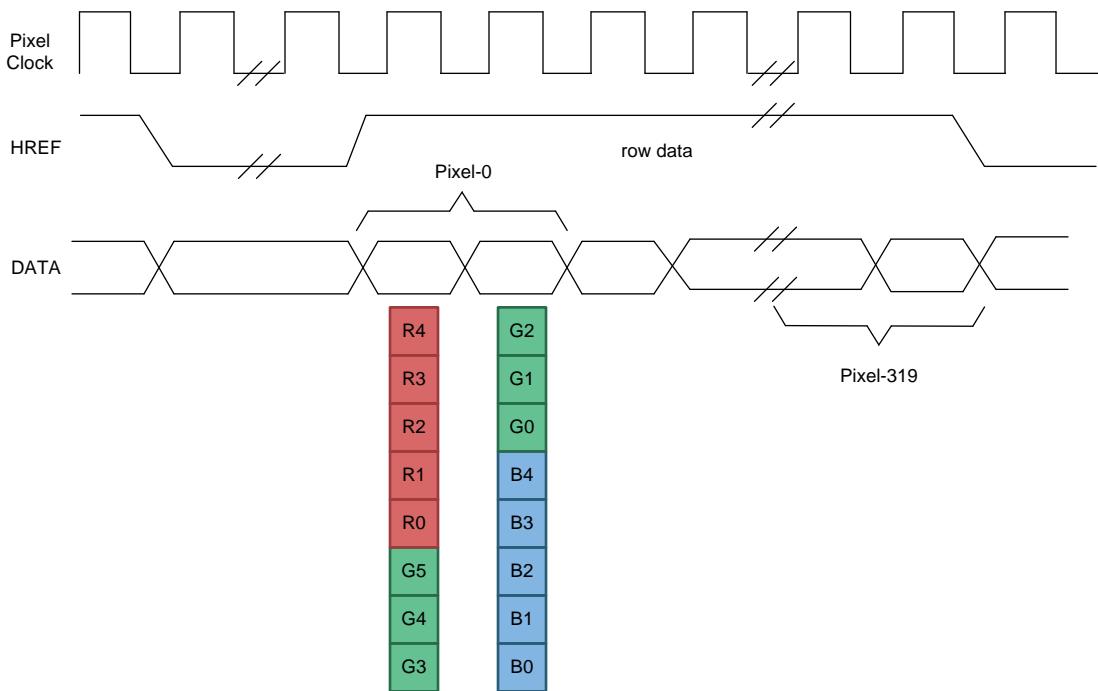
The camera module interface used for this reference design is the OV9655; however, other camera modules with similar data, configuration, and control interfaces may be used.

#### 3.1.1 Camera Data Interface

The OV9655 has an eight-bit data interface, two additional control signals, and a pixel clock that are provided by the camera. The data and the control signals are timing referenced with respect to the pixel clock. The camera provides the following signals.

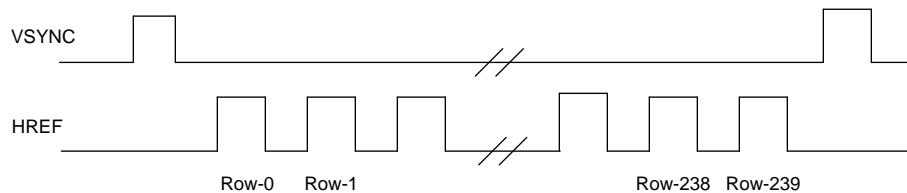
1. Pixel clock output
2. Vertical sync output to indicate end of the frame
3. Horizontal reference output to indicate valid data for pixels in a row
4. Eight-bit data output

The following images illustrate the data sent by the camera module for every line (see [Figure 3](#)) for a 16-BPP QVGA resolution image. The camera data and HREF signal are launched on the falling edge of the pixel clock. The data is valid on every rising edge for sampling by the MCU only when the HREF is logic high. Pixel position R4 and G2 correspond to MSB and pixel position G3 and B0 correspond to LSB.



**Figure 3. Line Information From Camera**

**Figure 4** shows the HREF and VSYNC for every frame for a 16-BPP QVGA resolution image. The HREF signal is toggled high at 240 times a frame. The time between two valid HREF is referred to as the *horizontal blank period*. The VSYNC is held high only after 240 lines of output from the camera module for a duration referred to as *vertical blank period*.



**Figure 4. Frame Information From Camera**

### 3.1.2 Camera Configuration Interface

The configuration interface on this camera module is called the serial camera control bus (SCCB). This interface has two signals. The SIO\_C is a clock input signal, and the SIO\_D is a bidirectional data signal. From a protocol perspective the SCCB is compatible with I<sup>2</sup>C. The configuration performed by the TM4C129x MCU is specific to the camera and the user must refer the manufacturer's data sheet.

### 3.1.3 Control Interface

Along with the data and configuration interface, the camera module provides three control signals. The camera includes an active-low reset signal, an active-high power down signal, and a clock, which generates timing signals and for its internal use.

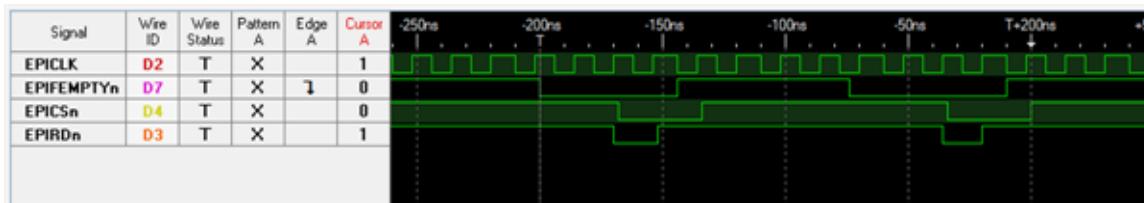
## 3.2 TM4C129x EPI Module

The TM4C129x EPI module provides a high-speed parallel interface for accessing memory devices (SDRAM, SRAM, flash) or programmable devices (FPGA, CPLD). The EPI has multiple modes of operations. To interface the camera module, the host bus 8-bit and 16-bit (HB8 and HB16) mode are used. One of the submodes of the HB8 and HB16 is the XFIFO, which is traditionally used for accessing external FIFO memory and has been leveraged in this reference design for camera access. An additional feature provided by the EPI module is the non-blocking read (NBR) operation which is capable of generating DMA requests through a FIFO. Both the XFIFO and NBRFIFO-DMA features are explained further in [Section 3.2.1](#) and [Section 3.2.2](#).

### 3.2.1 XFIFO Mode Description

The XFIFO mode uses 8- or 16-bits of data, removes address latch enable (ALE), addresses pins, and optionally adds external XFIFO, FFULL, and FEMPTY flag inputs. This submode provides the data side of the normal host-bus interface but is paced by the FIFO control signals. An important consideration is that the XFIFO, FFULL, and FEMPTY control signals may stall the interface and could have an impact on blocking read latency from the processor or μDMA.

For this design the read path for acquiring data from the camera is used, and the XFIFO and FEMPTY signals are used for indicating the EPI when valid data from the camera is available. As shown in [Figure 5](#), the read of the data is triggered only when the FEMPTY signal is asserted low. Once the EPI samples the FEMPTY low, a read can be initiated by the MCU (CPU or DMA) to read the data.



**Figure 5. XFIFO Read Timing Diagram**

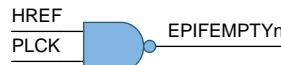
### 3.2.2 NBRFIFO-DMA Description

The EPI controller supports a special kind of read called a *non-blocking read*. While a normal read stalls the processor or  $\mu$ DMA until the data is returned, a non-blocking read is performed in the background.

A non-blocking read is configured by writing the start address into the EPIRADDR register, the size (data width) of transaction into EPIRSIZE register, and the count of operations into EPIRPSTD register. After each read is complete, the result is written into the NBRFIFO, and the EPIRADDR register is incremented by the size (one, two, or four). If the NBRFIFO is filled, the reads pause until space is made available. The NBRFIFO can be configured to interrupt the processor or trigger the DMA based on fullness using the EPIFIFOLVL register. By using the trigger and interrupt method, the DMA can keep space available in the NBRFIFO and allow the reads to continue unimpeded. Another benefit of the mechanism is the low latency of the DMA to transfer the data between the EPI and SRAM.

### 3.3 Logic Block Function

The external logic block is a glue logic that is used to ensure that the signals from the camera and EPI when the two are connected together do not result in bus contention. The logic block ensures that control signals from the camera interface can be converted to a format that the EPI can use. The logic block consists of a NAND gate for creating the FEMPTY signal and an octal driver for controlling the sampling of the data.



**Figure 6. Logic Block Gate**

#### 3.3.1 Generating the FEMPTY Signal

The XFIFO mode works by reading data only when the FEMPTY signal is de-asserted to indicate to the EPI that there is data available. To generate the FEMPTY signal, the HREF control signal and pixel clock from the camera are passed through a NAND. [Table 2](#) shows the truth table for the logic and its interpretation by the EPI.

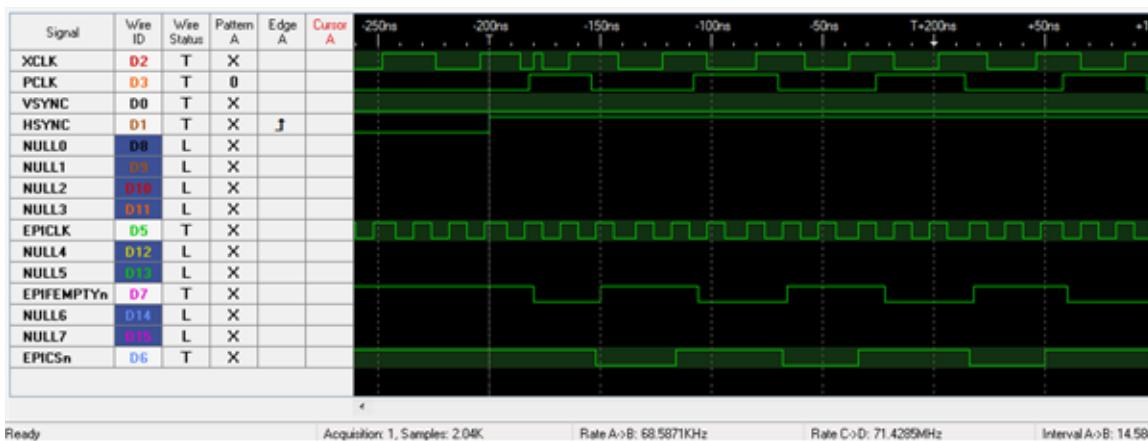
**Table 2. FEMPTY Signal Truth Table**

PCLK	HREF	FEMPTY	INTERPRETATION
L	L	H	Data cannot be sampled
L	H	H	Data cannot be sampled
H	L	H	Data cannot be sampled
H	H	L	Data can be sampled

Because the FEMPTY signal is synchronized by the EPI using its internal clock, the actual FEMPTY used by EPI shall always be a stable signal. Once sampled, FEMPTY can be used to trigger a read request by EPI using the combination of NBRFIFO and DMA allowing a fast response from the MCU to read the data.

### 3.3.2 Data Capture by EPI

The pixel clock is generally a clock in the MHz range. Because there is no frame buffer or FIFO in the data path, every data byte must be sampled correctly. To be able to perform such a high speed capture, the TM4C129x MCU is configured for a system clock of 120 MHz and the EPI configured for a 60-MHz clock. To overcome the absence of a frame buffer, the clock for the camera module is generated by the TM4C129x MCU using the system control clock output. The use of system control clock output provides a synchronous reference (see Figure 7) for the EPI to capture the data from the camera. The phase relation can be seen between the XCLK from TM4C129x to the camera and the PCLK from the camera to the TM4C129x.



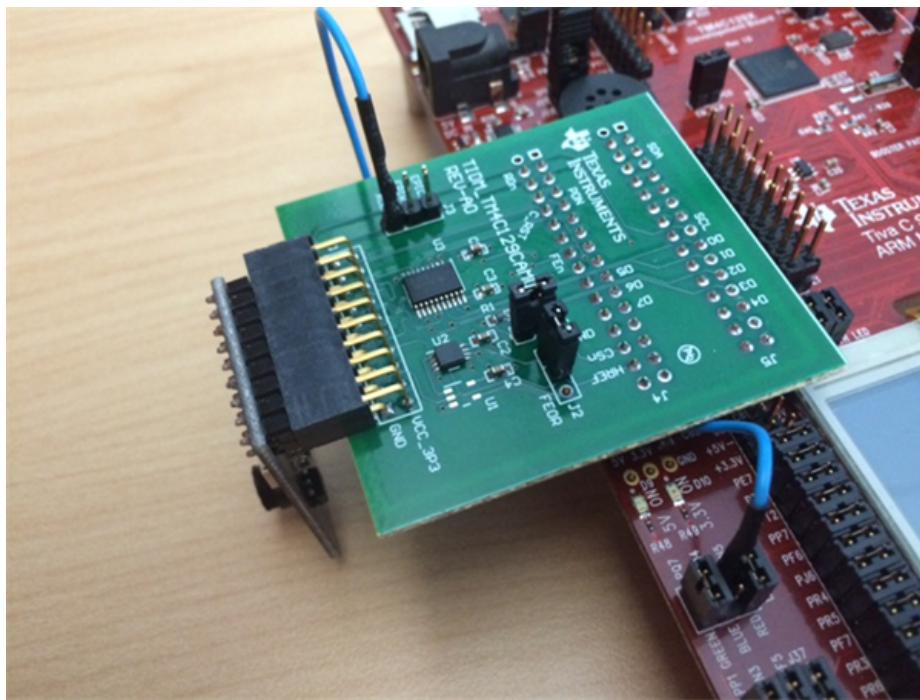
**Figure 7. EPI Data Capture Timing Diagram**

The EPI is configured with a non-blocking read with a read count equivalent to the number of pixels that must be captured for every line based on the HB8 or HB16 mode. As seen in the waveform, the HREF and PCLK from the camera, when combined with the NAND gate, forms the FEMPTYn signal for EPI. When this FEMPTY is sampled by the EPI as low, the EPI controller initiates the access to the octal transceiver by asserting the EPICSn and EPIRDn low. The EPICSn allows the data bus from the camera to be read by the EPI data pins.

## 4 Getting Started Hardware and Software

### 4.1 Hardware

To create the solution, the DK-TM4C129x development board is used as the base board to which the camera board can be connected. The USB is used to provide both power and debug for the reference design. To interface the camera board to the DK-TM4C129x, the camera board is connected to the headers J27 and J28 as shown in [Figure 8](#).



**Figure 8. TIDM-TM4C129CAMERA Assembly**

To provide the clock to the camera module, remove the jumper J26 (PQ4) on DK-TM4C129x and connect a wire from the PQ4 marking on the DK-TM4C129x to the XCLK on the J3 header of the camera board. Ensure that the header J2 is connected between pin positions 1 and 2, and the J6 header is mounted on the camera board.

### 4.2 Software

The software for this reference design comes as an installer that the user must install on their PC. It normally takes about a minute for the installer to execute. The following tools are required for rebuilding the project firmware.

1. Download and install TivaWare 2.1.3.
2. Download and install [Code Composer Studio™ \(CCS\)](#) 6.1.1.00022 with ARM® compiler tool chain version 5.2.7.
3. Download and install [LMFlashProgrammer](#) build 1613.
4. Download and install TIDM-TM4C129CAMERA-Firmware.exe.

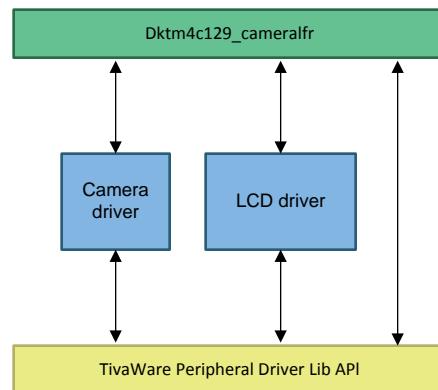
When the installer is executed with the default settings, the software will be installed under C:\Program Files (x86)\Texas Instruments\TM4C\TM4C129CAMERA-1.0. Three directories will be created by the installer in the above path.

#### 4.2.1 Common Drivers for the Embedded Application

The folder appdrivers contains the driver files for the example applications.

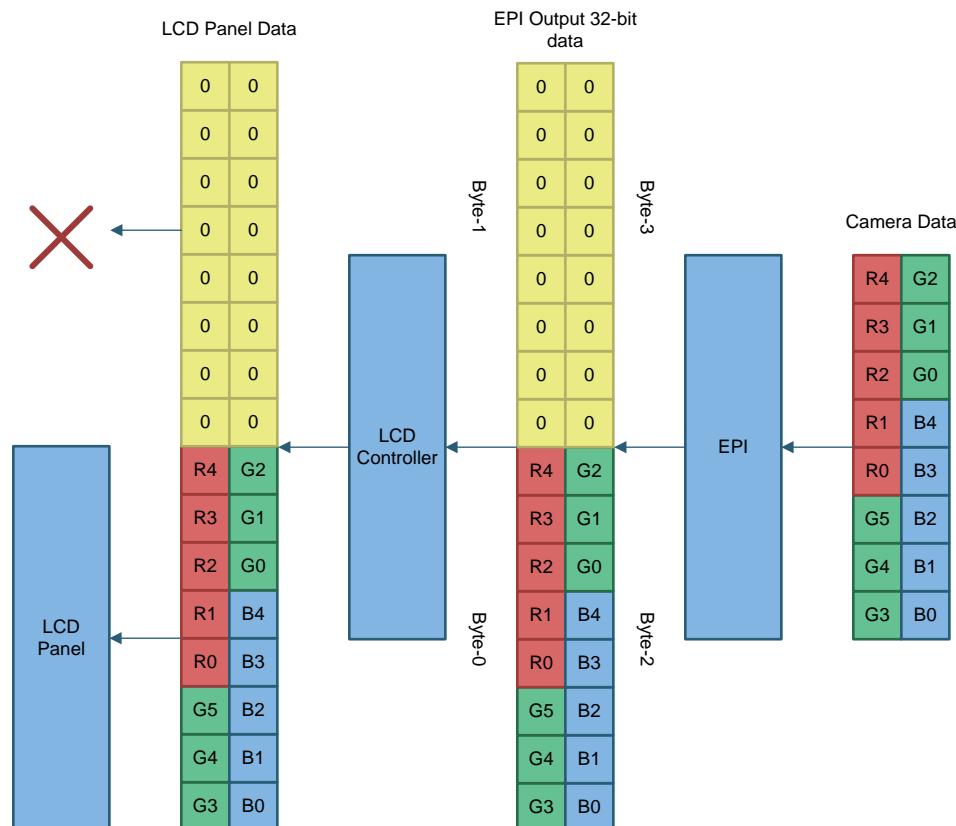
#### 4.2.2 Camera With LCD Project

The folder `dktm4c129_cameralf` contains the CCS project for the camera image capture and displays it on the onboard LCD panel. The software block diagram for the example is shown in [Figure 9](#).



**Figure 9. Software Block Diagram for dktm4c129\_cameralf**

[Figure 10](#) shows the data flow diagram for the project. The camera sends the data for every pixel in two bytes. The EPI is configured in HB16 mode, but because the data is received in the lower 8 bits, the EPI controller pads 0 in the upper byte of every 16-bit word. The padding of the upper eight bits has been done because the LCD panel on the DK-TM4C129x is configured to work in LIDD-8 mode. If the byte padding is not performed then the CPU must write the correct byte to the LCD controller, which would reduce CPU bandwidth for any other operation. By performing the automatic padding of data, the CPU can utilize the LCD controller's internal DMA to read the data and send it to the panel. The LCD DMA controller is configured to transfer the data in LIDD-16 mode, which causes the controller to discard the upper byte and only transmit the lower byte and, thus, retain the image information.

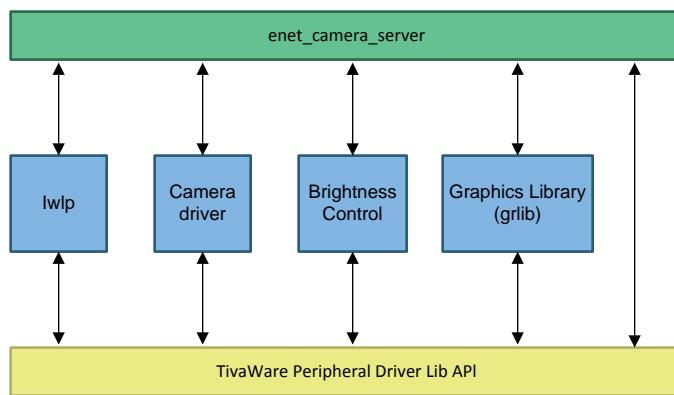


**Figure 10. Pixel Translation During Data Flow**

Since the pixel data captured per line is buffered by the LCD controller's internal FIFO, the SRAM requirement is limited to only one line of data capture, that is, 320 pixels × (2 bytes of pixel information + 2 bytes of padded data) or 1280 bytes.

#### 4.2.3 Ethernet Web Server With Camera Project

The folder `enet_camera_server` contains the CCS project for the camera image capture and viewing it through a web browser. For this application the TM4C129x MCU hosts a web page, which can be opened in a web browser and auto-refreshes every three seconds. The software block diagram for the example is shown in [Figure 11](#).



**Figure 11. Software Block Diagram for `enet_camera_server`**

For this application the EPI is configured in HB8 mode. The camera sends the data for every pixel in two bytes. The EPI controller now concatenates the bytes in a 32-bit word. Because the web server reads the data as is before sending it to the browser, there is no requirement for the EPI to perform padding operation.

The TM4C129x device hosts the web page from the flash but the image is kept in SRAM. If the user wants to change the web page then there are some steps that must be followed. These steps are given in the `ReadMe.txt` of the example code.

## 5 Testing and Results

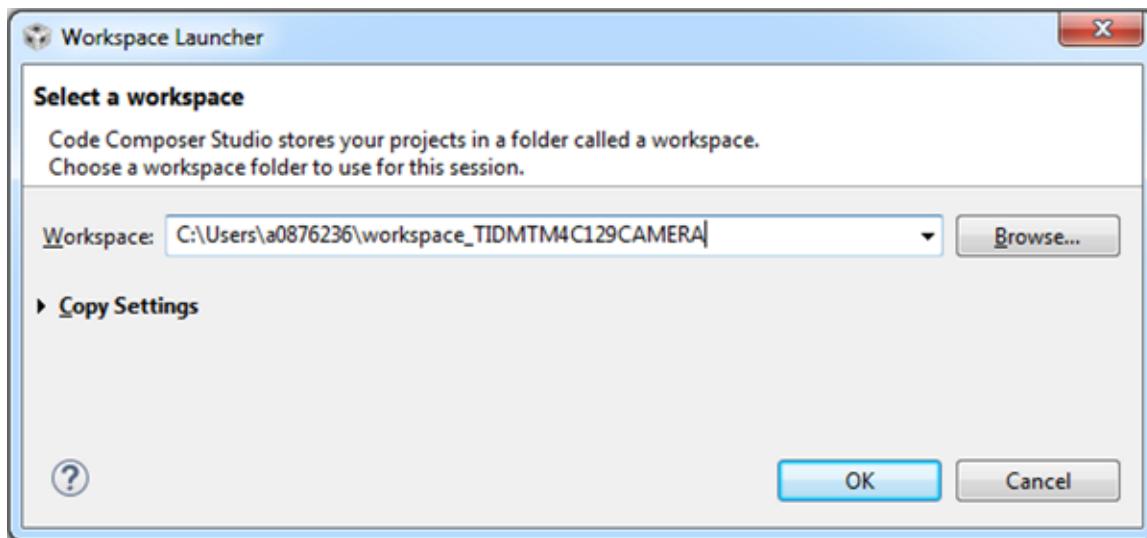
### 5.1 Test Setup

Now that the board has been setup for downloading the application firmware and the user has installed the application software, this section will elaborate on how to import, build, and run the application firmware for voice call.

#### 5.1.1 Create a New Workspace in CCS™

Before building the examples the user must create a new workspace to ensure a clean build of the project.

1. Start CCS.
2. Click on *File* then navigate to *Switch Workspace → Others* in the drop down menu.
3. In the pop-up box, provide the name of the new workspace. The created example workspace is called *workspace\_TIDMTM4C129CAMERA* (see [Figure 12](#)). Click *OK*.



**Figure 12. Create a New Workspace**

### 5.1.2 Import and Build Projects in CCS™

To compile the example code, the opus audio codec library must first be built. The example code uses the precompiled output of the opus audio codec library during the linker phase.

1. In the workspace created (see [Section 5.1.1](#)), select *File → Import*. The import window will display.
2. Expand CCS for more options. Click *CCS Projects*. Click on *Next* (see [Figure 13](#)).

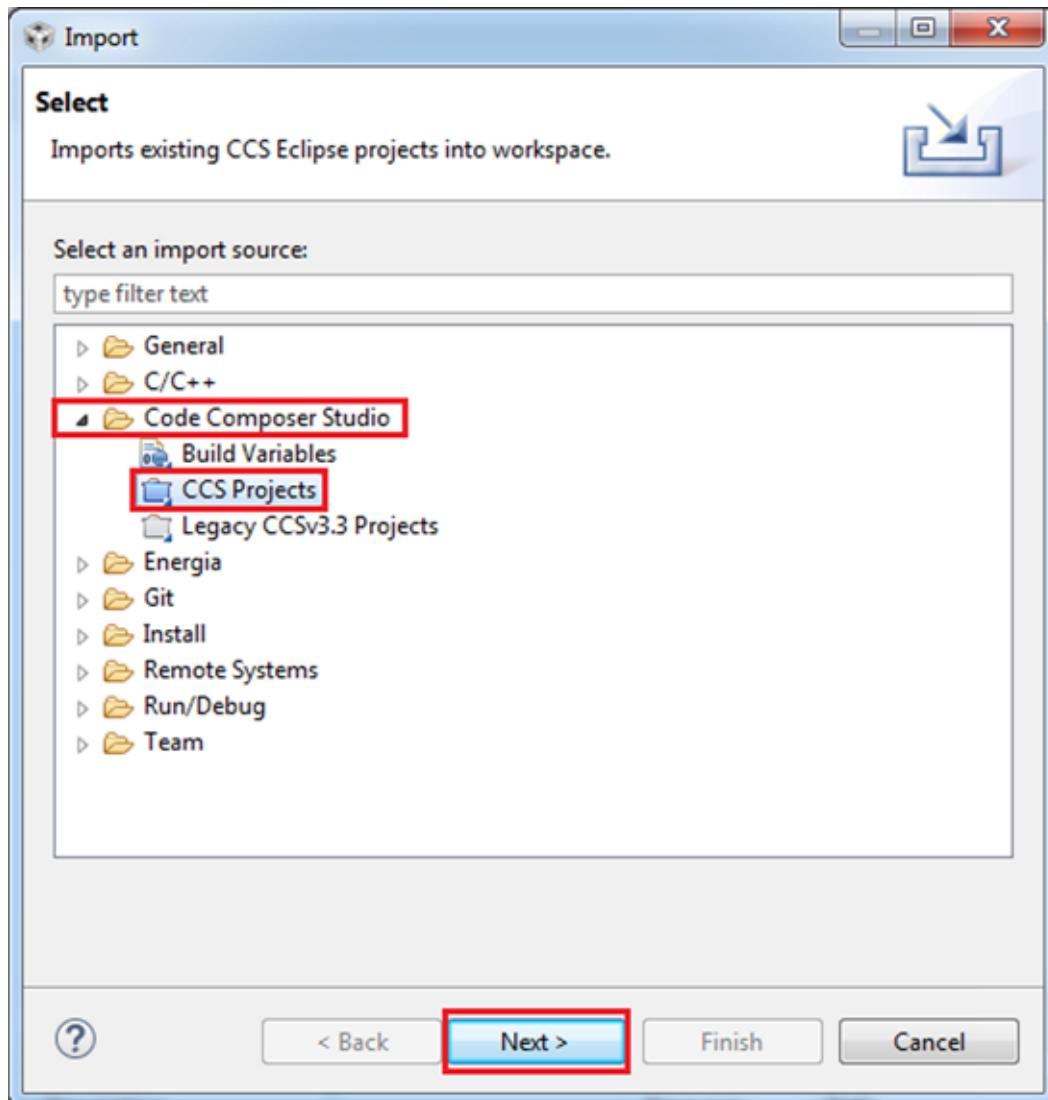
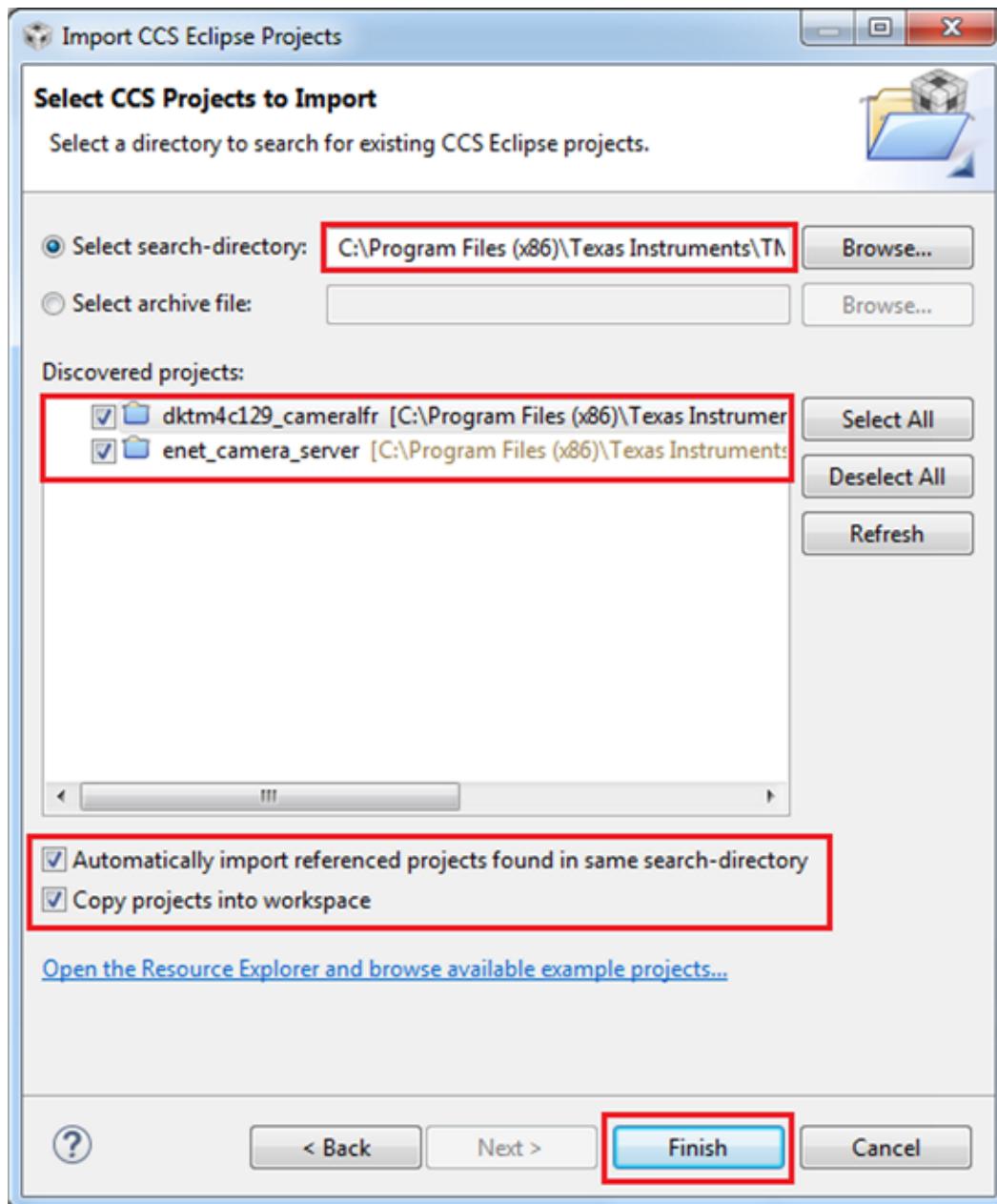


Figure 13. Import Selection View

3. Click the *Browse* button in front of *Select search-directory* option and navigate to the directory where the project collateral has been installed (see [Section 4.2](#)). Select all the projects that are listed in the *Discovered projects: pane*. Make sure that the boxes in front of *Automatically import referenced projects found in the same search-directory* and *Copy projects into workspace* are checked (see [Figure 14](#)). Now click on *Finish* button. The examples will import into CCS.



**Figure 14. Import the Projects**

4. Before building the projects, make sure that the variable `ORIGINAL_PROJECT_ROOT` points to the correct directory where the TivaWare software has been installed and that `TID_ROOT` points to the correct directory where the project collaterals have been installed. To view or change the variable, right-click on the project and click on *Show Build Settings*. In the pop-up window, click on *Build*. Click on the tab *Variables* to view the value of `ORIGINAL_PROJECT_ROOT` or `TID_ROOT`. If the path shown is different compared to the path on the machine where the opus source code has been extracted then select the variable and click on *Edit...* to change the path (see Figure 15).

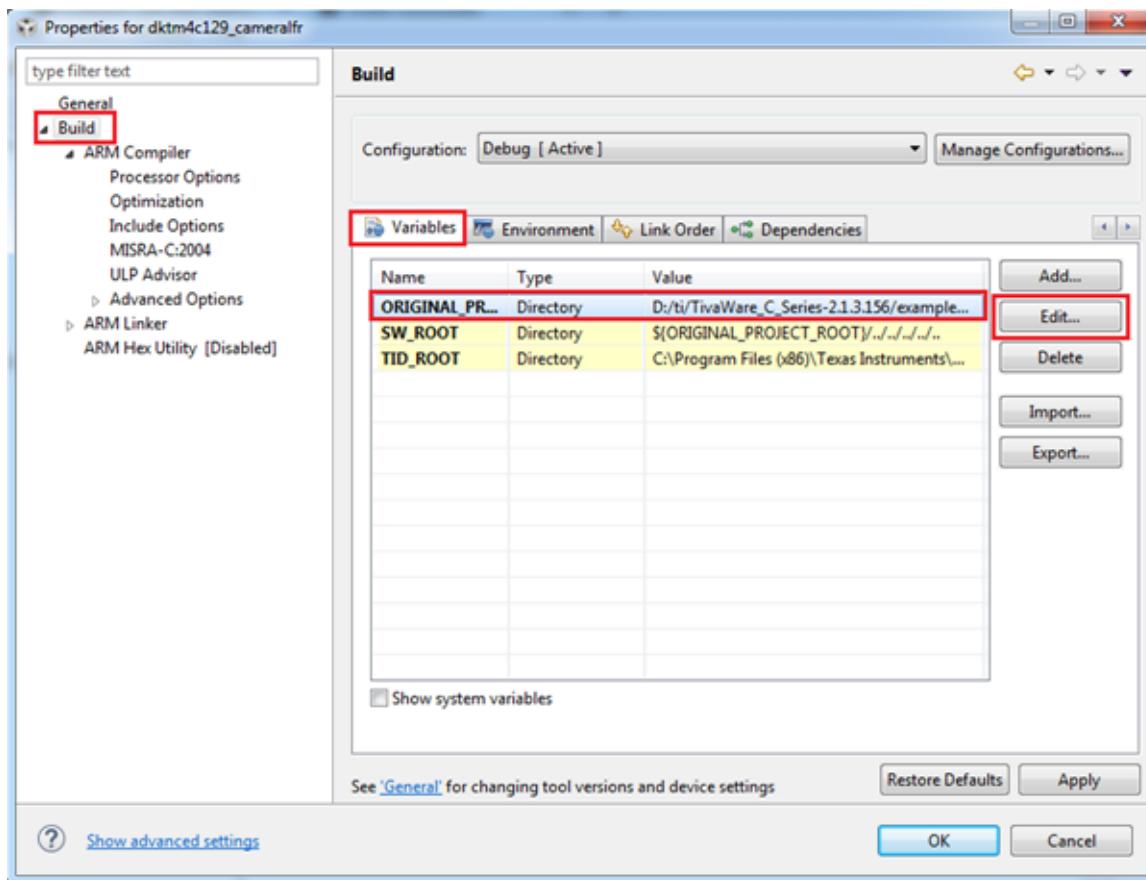
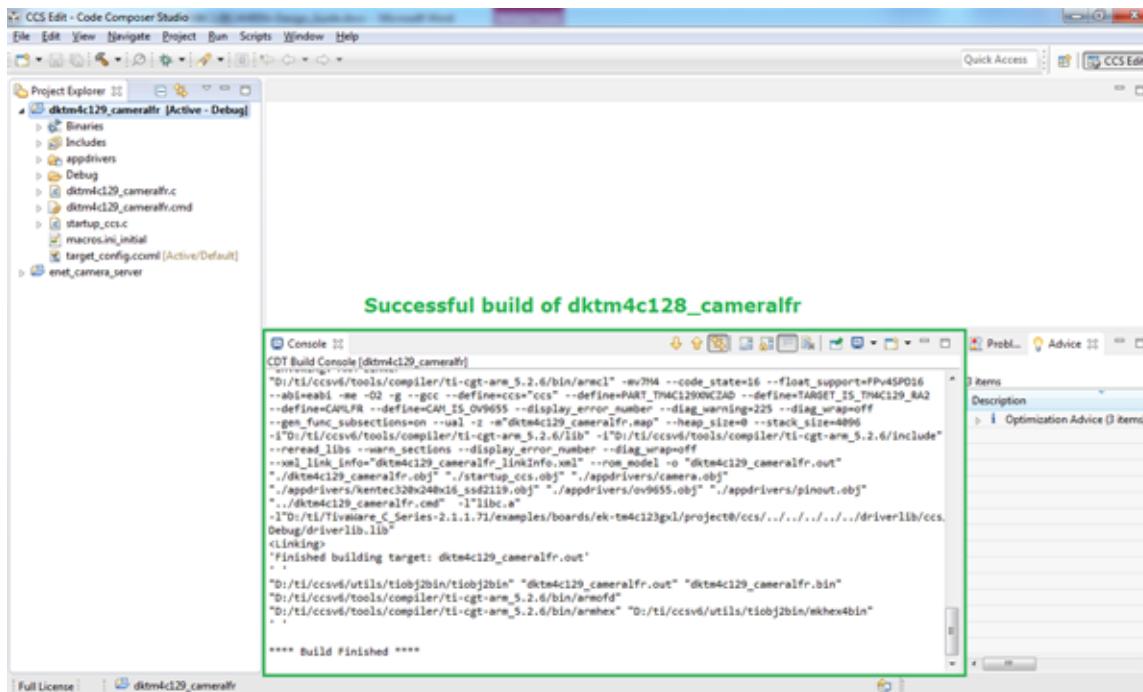
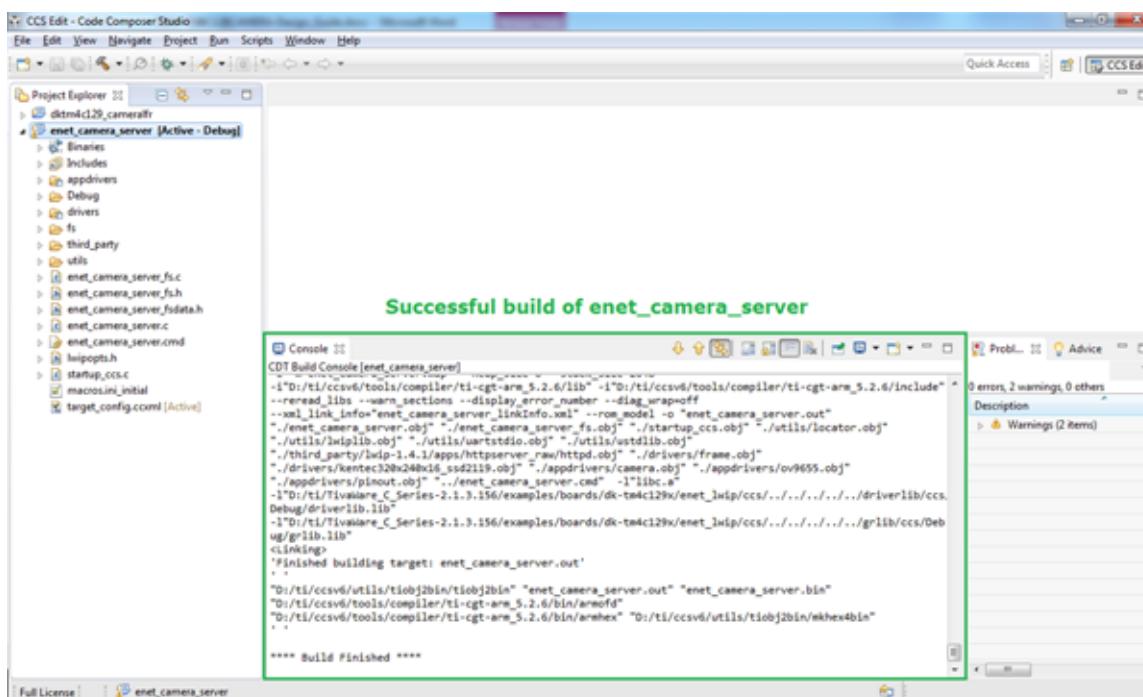


Figure 15. Update the Environment Variables

5. Right-click the project in *Project Explorer* and click *Build Project*. If building the library for the first time, it may take a few minutes. After the compilation is successful, the CCS console must display the message in [Figure 16](#) and [Figure 17](#).



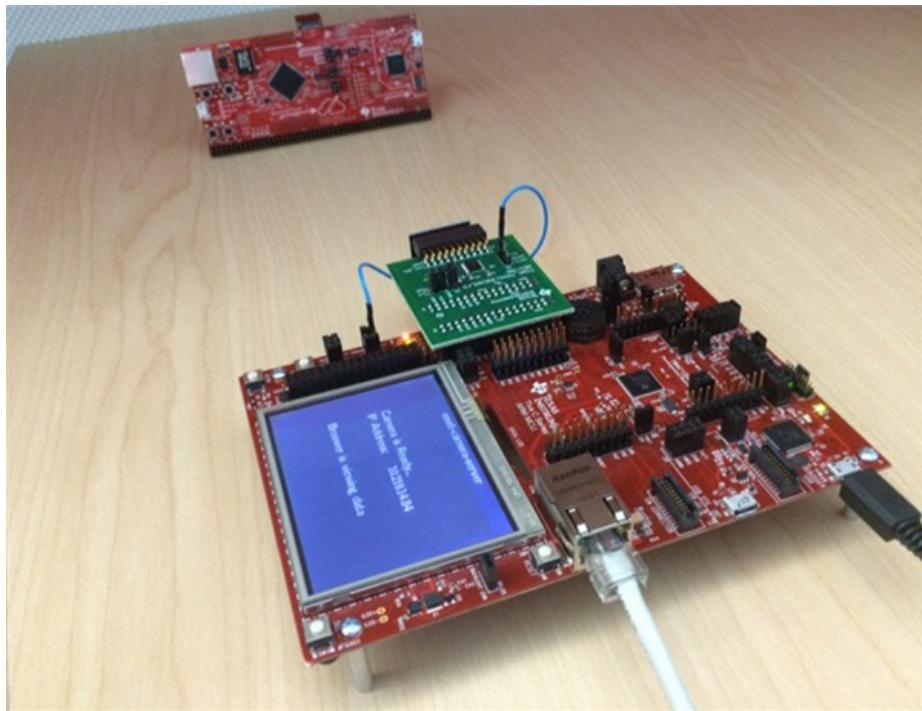
**Figure 16. Compiling dktm4c129\_camerafr Project**



**Figure 17. Compiling enet\_camera\_server Project**

### 5.1.3 Test Rig

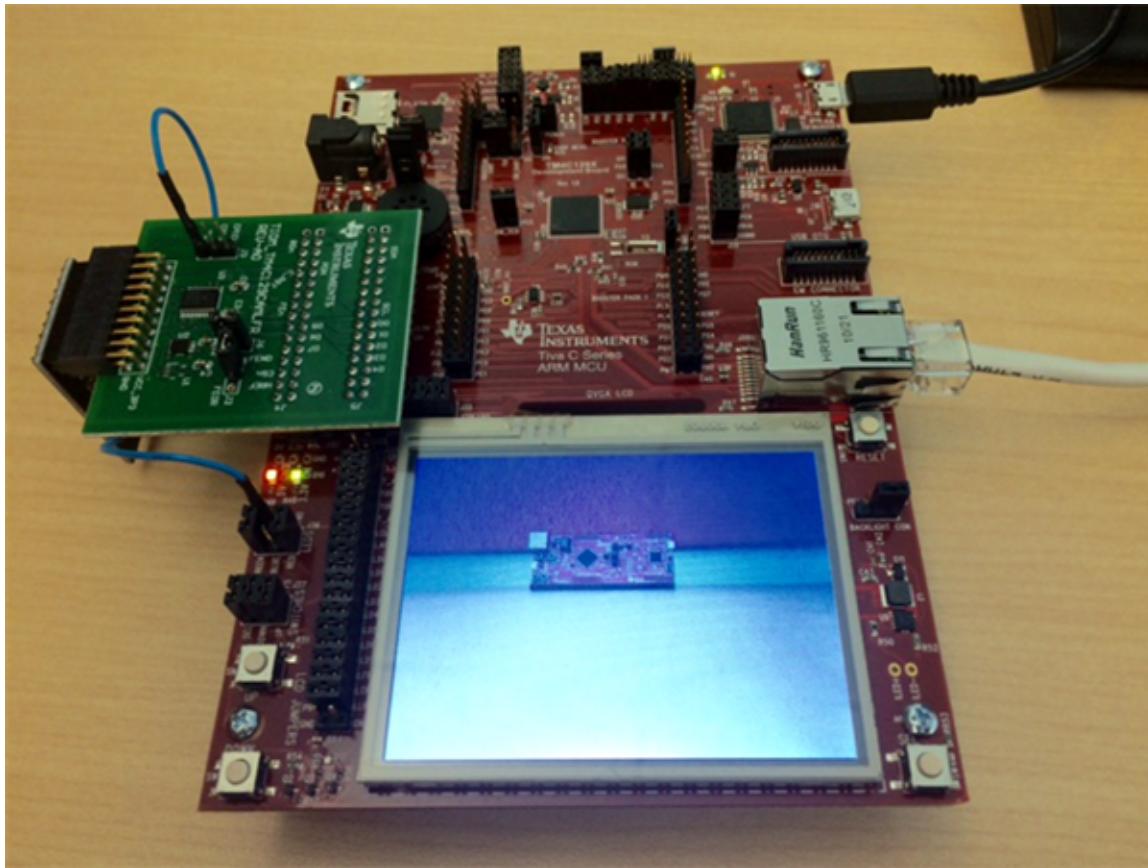
To display an image captured by the camera either on the LCD or hosted by the web server application, a test rig is set up with an EK-TM4C1294XL connected launchpad shown in [Figure 18](#).



**Figure 18. Test Rig**

### 5.1.4 Executing Camera With LCD

Now that the CCS project *dktm4c129\_camerafr* is built, download the binary image using LMFlashProgrammer or use the debug button in CCS to download the image to the DK-TM4C129x. On running the application, the TM4C129x will start capturing data from the camera and display it on the LCD panel (see [Figure 19](#)).



**Figure 19. Board View on Executing *dktm4c129\_camerafr***

### 5.1.5 Executing Camera With Web Server

Now that the CCS project `enet_camera_server` is built, download the binary image using LMFlashProgrammer or use the debug button in CCS to download the image to the DK-TM4C129x.

---

**NOTE:** The web page hosted by the DK-TM4C129x is configured to auto-refresh the web browser every three seconds using the HTML META tag for reload, which eliminates the requirement for the user to manually reload the page or have a plugin to reload the page automatically.

---

On running the application, the TM4C129x will first acquire the link and then an IP address (see Figure 20).

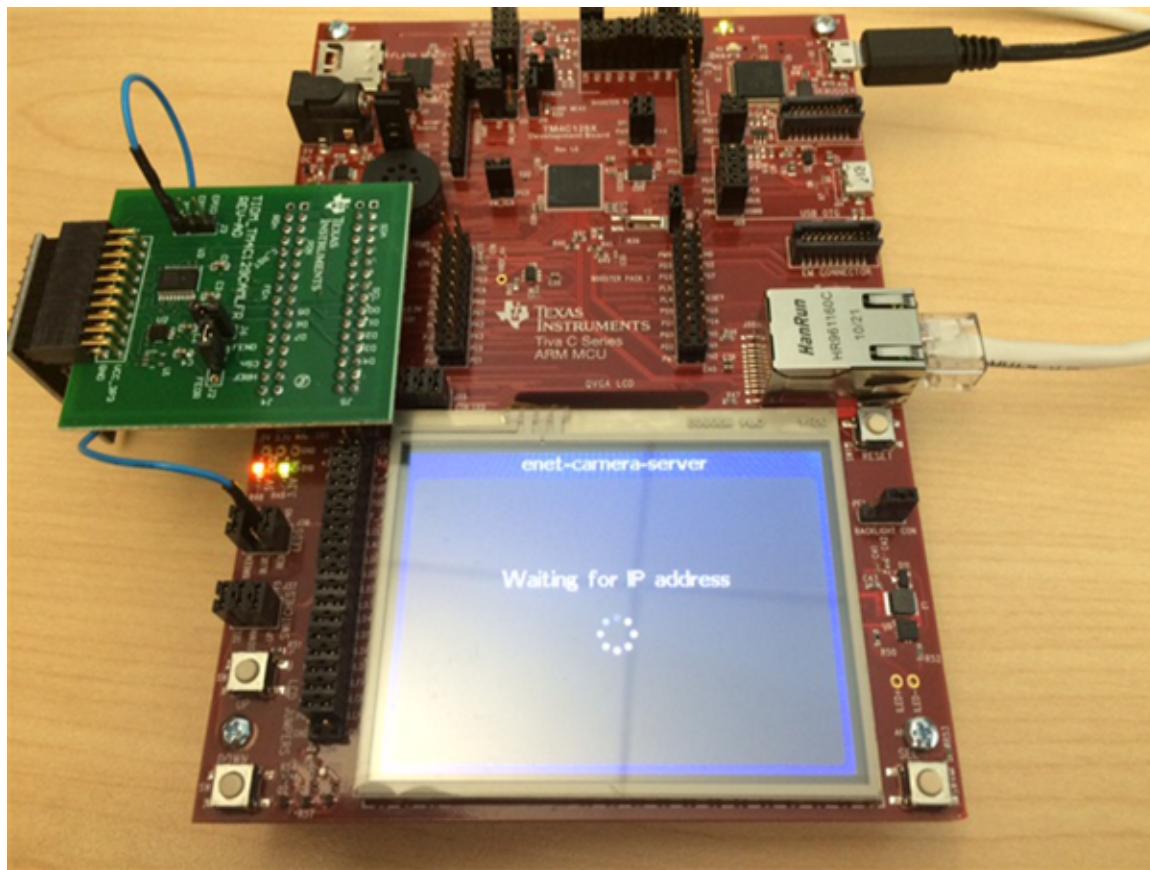
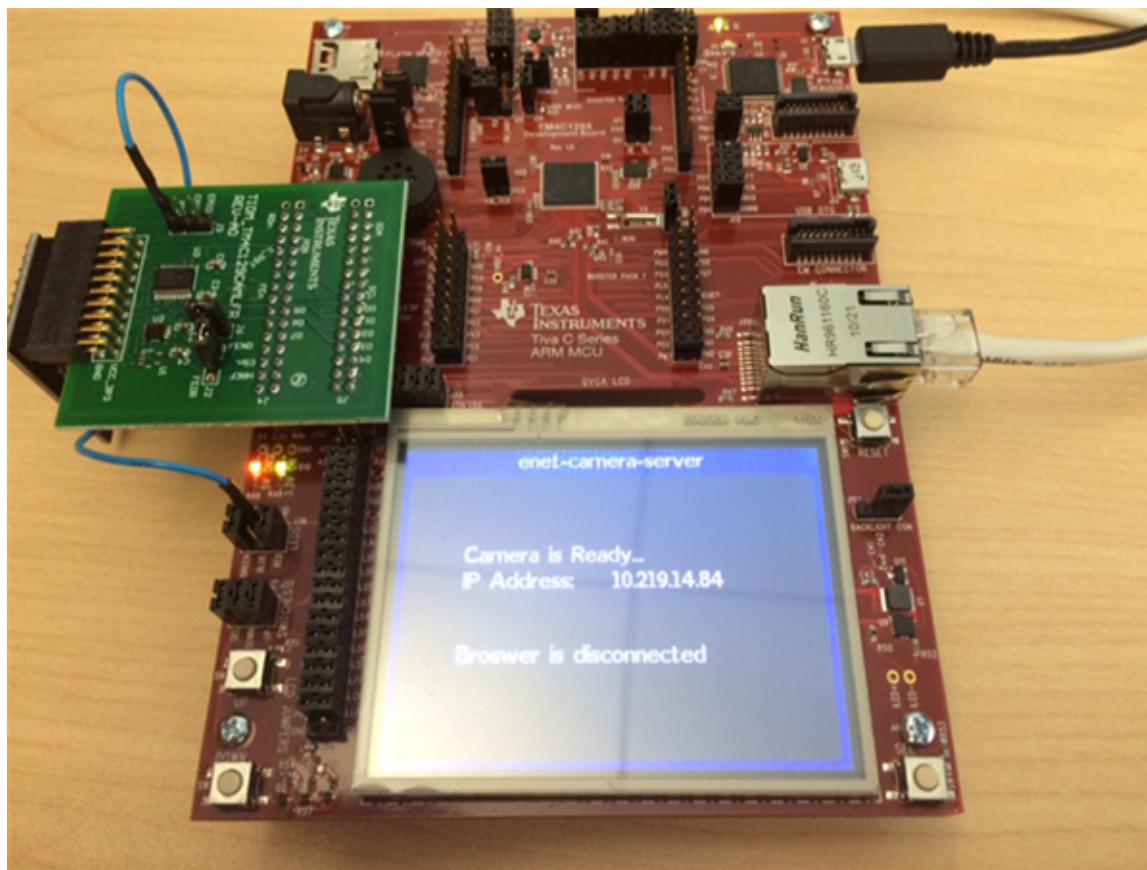


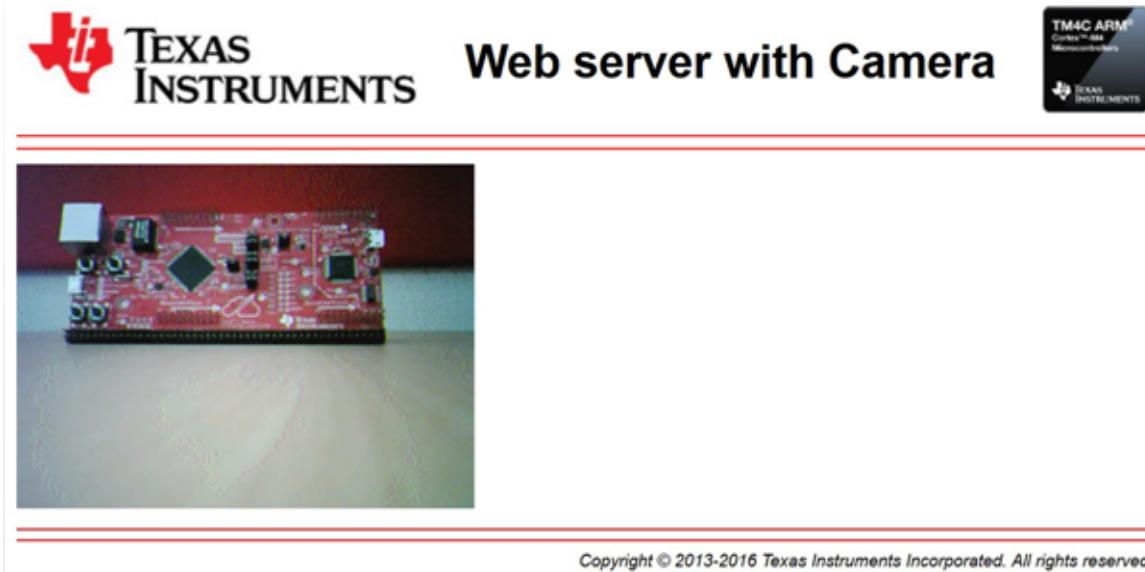
Figure 20. Board View When Acquiring IP Address

Once the IP address is acquired, the application informs the user through the LCD panel that the camera is ready and displays the IP address and connection status of a web browser (see [Figure 21](#)).



**Figure 21. Board View When Application is Ready**

The user can now open a web browser and type in the IP address. When entering in the IP address, the DK-TM4C129x will send the image captured by the camera to the web browser (see [Figure 22](#)). At the same time, the status message on the LCD panel changes to indicate that a web browser is connected to the MCU hosting the camera service (see [Figure 23](#)). If the web browser is closed, the status updates to indicate that the browser is no longer requesting camera data.



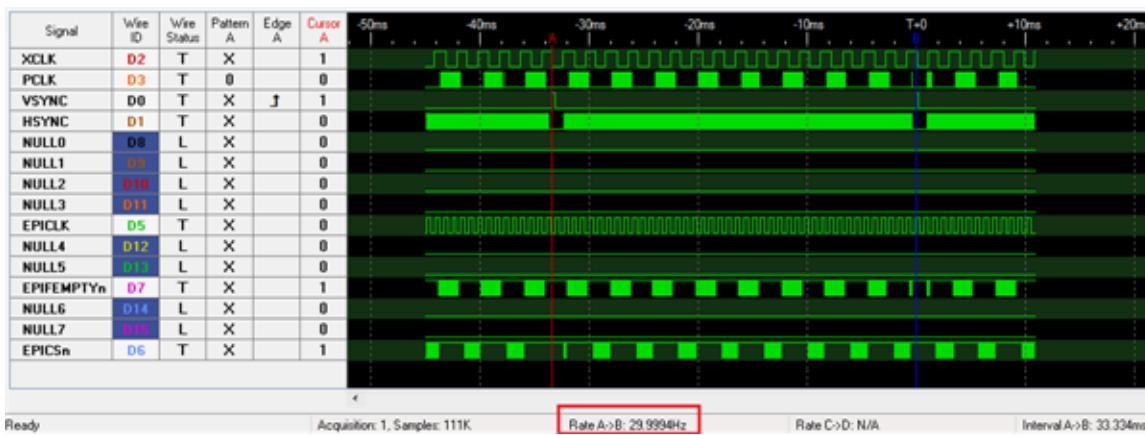
**Figure 22. Web Browser View of Hosted Web Page**



**Figure 23. Board View When Web Browser is Viewing Web Page**

## 5.2 Test Data

Figure 24 shows the frame and line rate generated by the camera and its translation to EPI access. The measurements shown for XCLK and EPICLK must be ignored as a result of aliasing at a lower sampling rate to capture 33.33 ms signal with a limited logic analyzer buffer.



**Figure 24. Frame and Line Rate for Data Capture**

## 6 Design Files

### 6.1 Schematics

To download the schematics, see the design files at [TIDM-TM4C129CAMERA](#).

### 6.2 Bill of Materials

To download the bill of materials (BOM), see the design files at [TIDM-TM4C129CAMERA](#).

### 6.3 PCB Layout Recommendations

#### 6.3.1 Layout Prints

To download the layer plots, see the design files at [TIDM-TM4C129CAMERA](#).

### 6.4 Altium Project

To download the Altium project files, see the design files at [TIDM-TM4C129CAMERA](#).

### 6.5 Gerber Files

To download the Gerber files, see the design files at [TIDM-TM4C129CAMERA](#).

### 6.6 Assembly Drawings

To download the assembly drawings, see the design files at [TIDM-TM4C129CAMERA](#).

## 7 Software Files

To download the software files, see the design files at [TIDM-TM4C129CAMERA](#).

## 8 References

1. Wave Share, [Ov9655 Camera Board](#).

### 8.1 Trademarks

All trademarks are the property of their respective owners.

## 9 About the Author

**AMIT ASHARA** is an application engineer and Member Group Technical staff member at Texas Instruments. He works on developing applications for the TM4C12x family of high-performance MCUs. Amit brings extensive experience in high-speed digital and MCU system-level design expertise to this role. Amit earned his Bachelor of Engineering (BE) from the University of Pune in India.

## Revision A History

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

Changes from Original (September 2016) to A Revision	Page
• Changed location of <a href="#">Section 1</a> (previously Section 1.1).....	2
• Changed location of <a href="#">Section 1.1</a> (previously Section 1.2) .....	2
• Changed <a href="#">Figure 2</a> .....	3

---

## **IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES**

Texas Instruments Incorporated ("TI") technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

TI RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI's standard terms for semiconductor products (<http://www.ti.com/sc/docs/stdterms.htm>), evaluation modules, and samples (<http://www.ti.com/sc/docs/samptersms.htm>).

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2017, Texas Instruments Incorporated