

TI Designs: TIDM-TM4C129SDRAMNVM

Execute from SDRAM with code storage in NVM for high-performance MCU reference design



Description

The memory space of the TM4C129x microcontrollers (MCUs) can be extended when the internal memory is insufficient. This reference design describes the hardware-interfacing requirement and example software for a TM4C129x MCU.

Resources

TIDM-TM4C129SDRAMNVM	Design Folder
EK-TM4C1294XL	Tool Folder
TM4C1294NCPDT	Product Folder
TM4C123GH6PM	Product Folder
TPD4S012	Product Folder
TPS2052B	Product Folder
TPS62177	Product Folder
TPS73733-Q1	Product Folder

Features

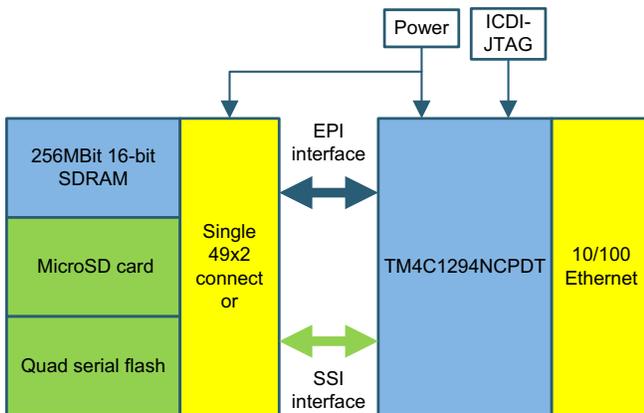
- Extends the useable memory space to 512-Mb 16-bit SDRAM with 60-MHz external peripheral interface (EPI) for high-memory throughput and footprint applications
- Designed for the [Arm® Cortex®-M4F-based MCU TM4C1294 Connected LaunchPad™ evaluation kit](#)
- Implements serial interface bootloaders for SD card or quad serial flash nonvolatile memory
- Offers additional support for quad-serial flash mode command for custom implementation of bare metal external memory loggers
- Offers source code that contains project examples for Code Composer Studio™ IDE

Applications

- Interactive human-machine interfaces
- Industrial automation
- Internet of things (IoT) solutions
- Test and measurement



[ASK Our E2E™ Experts](#)



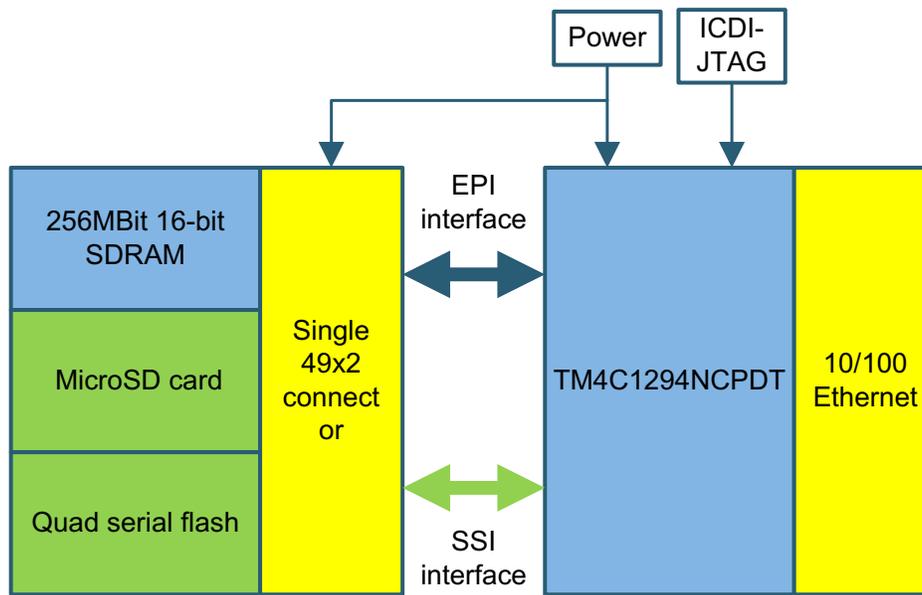
An IMPORTANT NOTICE at the end of this TI reference design addresses authorized use, intellectual property matters and other important disclaimers and information.

1 System Description

The external peripheral interface (EPI) of the TM4C129x MCUs can be used to extend the executable memory region to 16-bit 512Mb of SDRAM. The QSSI interface at 60 MHz can extend storage of NVM code. This capability lets applications use microSD cards or QSSI flash memory greater than 512Mb. The design files include schematics, BOM, Gerber files, and reference example code for an easy-to-use SDRAM, an SD card boot, and an QSSI boot with a [TM4C1294NCPDT Connected LaunchPad development kit](#).

2 System Overview

2.1 Block Diagram



SDRAM-NVM Memory Extender Block Diagram

2.2 TM4C1294NCPDT Microcontroller

The TM4C1294NCPDT is a 120-MHz high-performance microcontroller with 1MB of on-chip flash and 256KB of on-chip SRAM. The device features an integrated Ethernet MAC and PHY for connected applications. The device has high-bandwidth interfaces like a memory controller and a high-speed USB 2.0 digital interface. Integrating low- to mid-speed serials, up to 4-Msps 12-bit ADC, and motion control peripherals, this design is a unique solution for a variety of applications from industrial communication equipment to smart energy and smart grid applications.

Figure 1 shows a high-level block diagram of the TM4C1294NCPDT microcontroller.

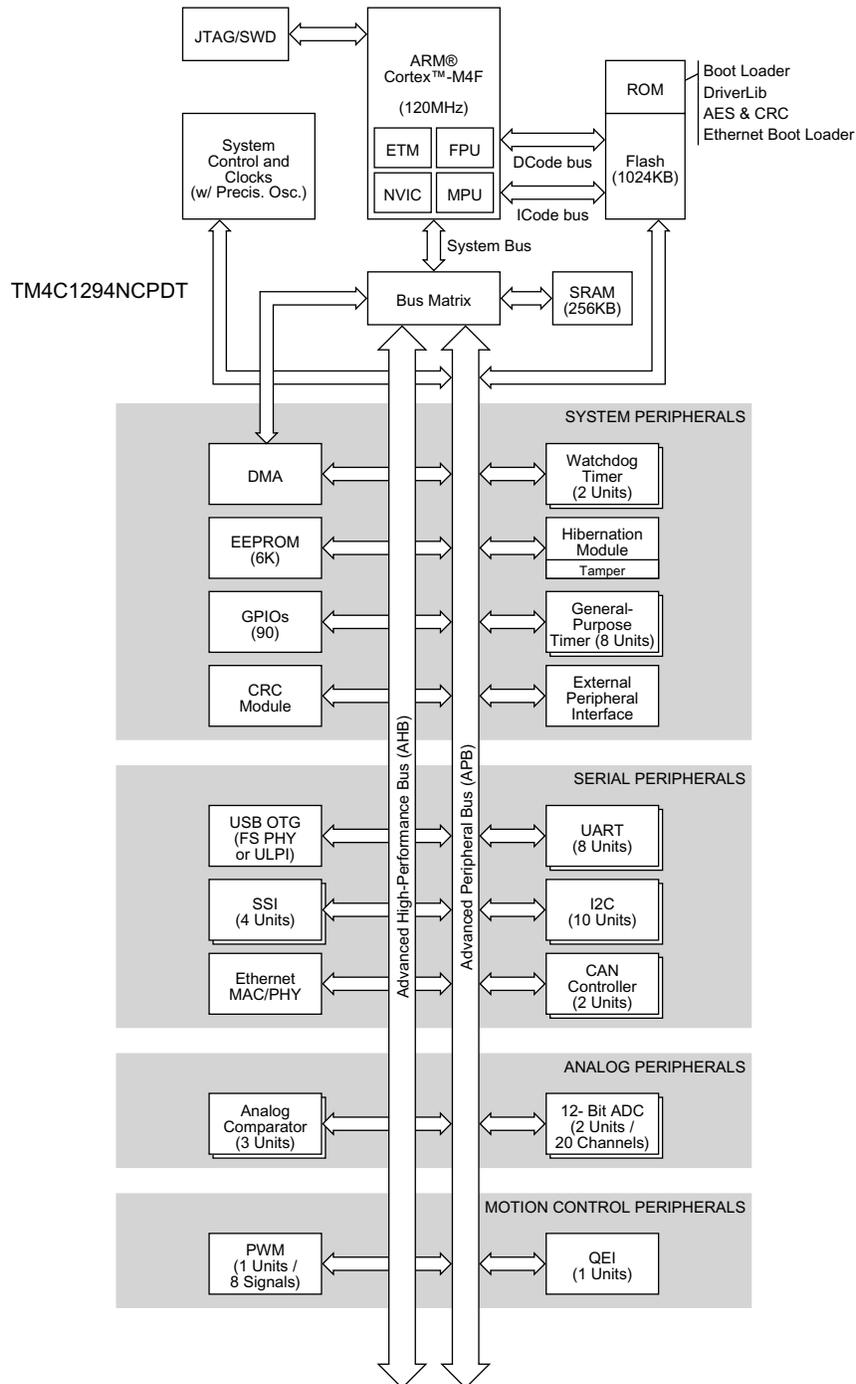


Figure 1. TM4C1294NCPDT Microcontroller High-Level Block Diagram

3 Hardware, Software, Testing Requirements, and Test Results

3.1 Getting Started Hardware

Interfacing the SDRAM-NVM memory to the TM4C1294NCPDT device on an EK-TM4C1294XL Connected LaunchPad kit requires a daughterboard that can connect to the breadboard connector X11.

3.1.1 SDRAM-NVM Daughtercard

The SDRAM-NVM daughtercard interfaces to the EK-TM4C1294XL Connected LaunchPad kit using the 49x2 breadboard connectors. The daughtercard has one jumper (J1) that can be used to select between the microSD card or the QSSI serial flash. The microSD card uses the legacy SPI mode of the QSSI module to interface with the microcontroller while the QSSI serial flash uses the advanced mode of the QSSI module to interface with the microcontroller. [Figure 2](#) shows an overview of the connector mounting for the daughtercard.

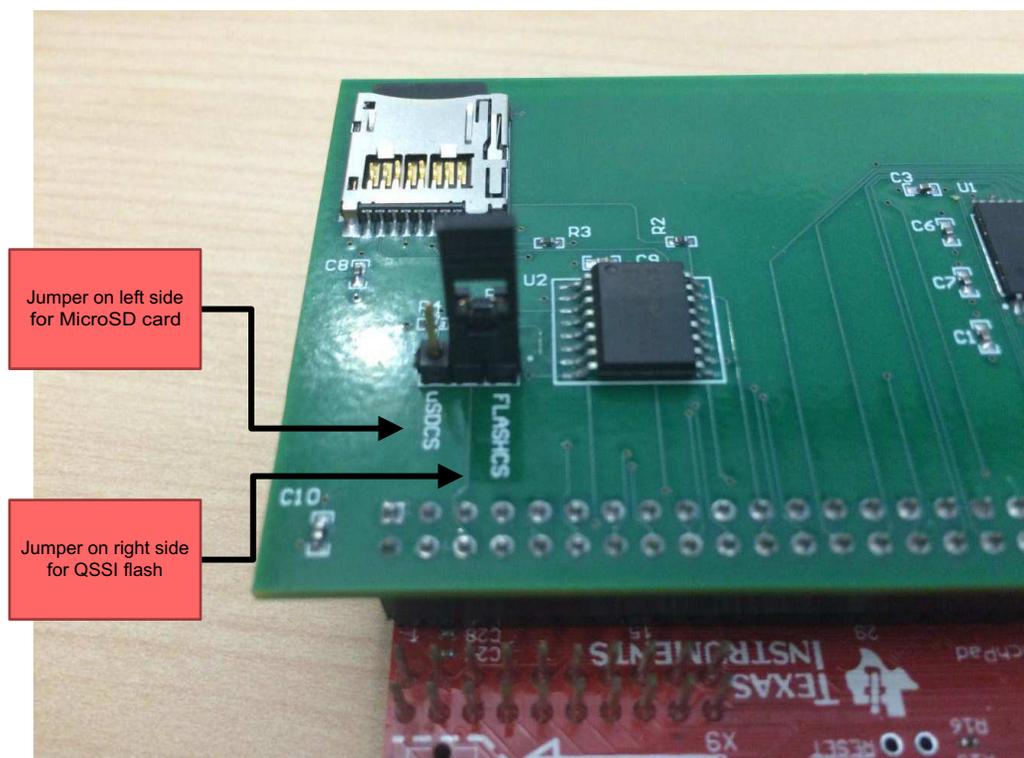


Figure 2. SDRAM-NVM Connector Mounting

3.2 Getting Started Software

The software for this reference design comes with three codes that you can import in Code Composer Studio IDE and use as a starting point for your application (for download information, see [Section 5](#)).

3.2.1 MicroSD Card Boot With SDRAM Code Execution

The MicroSD card boot with SDRAM code uses the internal flash of the TM4C1294NCPDT to hold the FAT file system and bootloader. The bootloader configures the QSSI modules to run the FAT file system and EPI to interface to a 512-Mb SDRAM at interface frequency of 60 MHz. You can have multiple images on an microSD card configured during compile time to execute from an EPI address space of 0x6000 0000. You can select one of the image files that the bootloader copies to the EPI peripheral-connected SDRAM. After the image is copied, the Cortex-M4 disables the interrupts, updates the NVIC_VTABLE register to map to the external address map, and jumps to the external address space of 0x6000 0000. All subsequent code execution occurs in the external address space until the next board reset. You must use a PC to copy the images to the microSD Card.

3.2.2 QSSI Serial Flash Boot With SDRAM Code Execution

The QSSI serial flash boot with SDRAM code uses the internal flash of the TM4C1294NCPDT to hold a custom bootloader. The bootloader configures the QSSI modules to read a QSSI flash memory and the EPI to interface to a 512-Mb SDRAM at an interface frequency of 60 MHz for executing code.

The lowest sector (Sector-0) of QSSI flash memory holds a table indicating the start address, size, and validity of an image. The bootloader updates this location when you download the binary file to the external QSSI flash memory through UART0. Figure 3 shows the structure of the information held in Sector-0 pertaining to an actual application image.

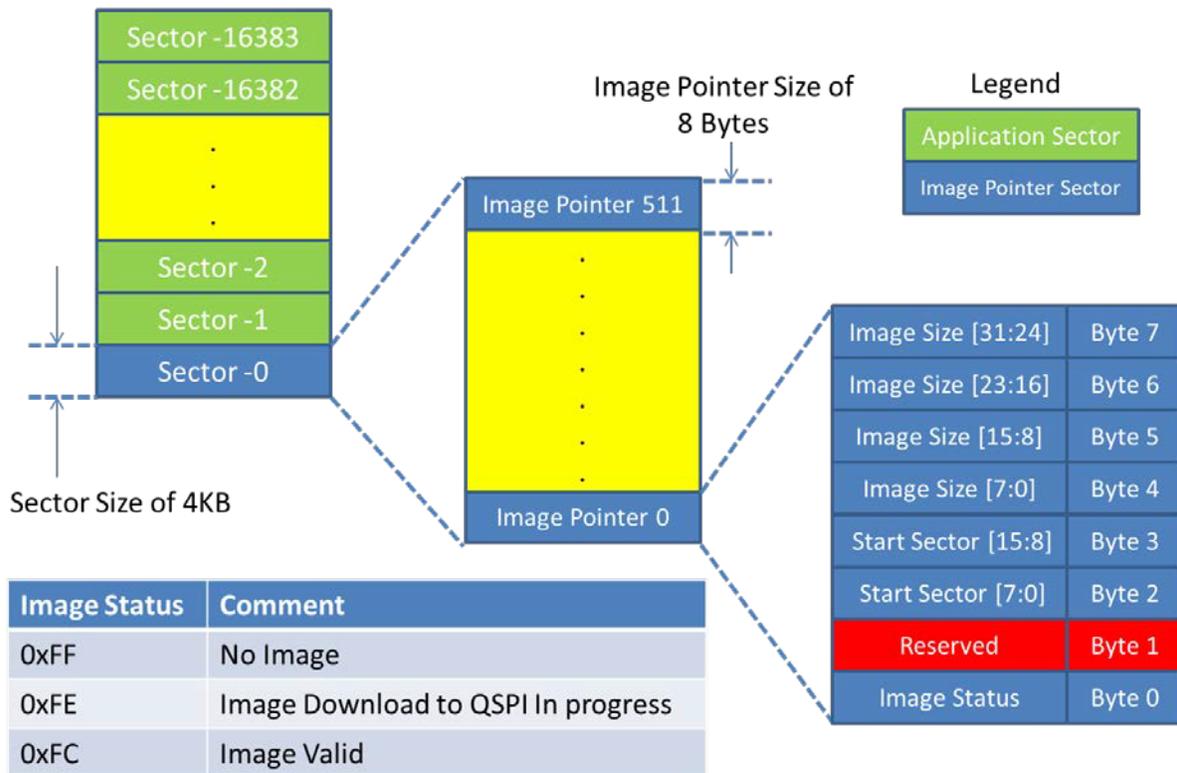


Figure 3. Sector-0 Information Block Structure

You can download an image to the QSSI flash memory that has been configured at compile time to execute from the EPI address space 0x6000 0000. The image is downloaded from a PC or other controller. To load a new image to the QSSI flash, press USR_SW1 when powering up or resetting the LaunchPad kit. This causes the bootloader to enter download mode. If the USR_SW1 is not pressed, the bootloader reads Sector-0 for a valid image pointer and executes the last image available on QSSI flash. If no valid image pointer is found, the bootloader enters download mode and waits for a new image. During

the execution phase, the bootloader copies the image to the SDRAM memory connected to the EPI peripheral. When the image finishes copying, the Cortex-M4 disables the interrupts, updates the NVIC_VTABLE register to map to the external address map, and jumps to the external address space 0x6000 0000. All subsequent code execution occurs in the external address space until the next board reset. Figure 4 shows a flow chart of how the code operates.

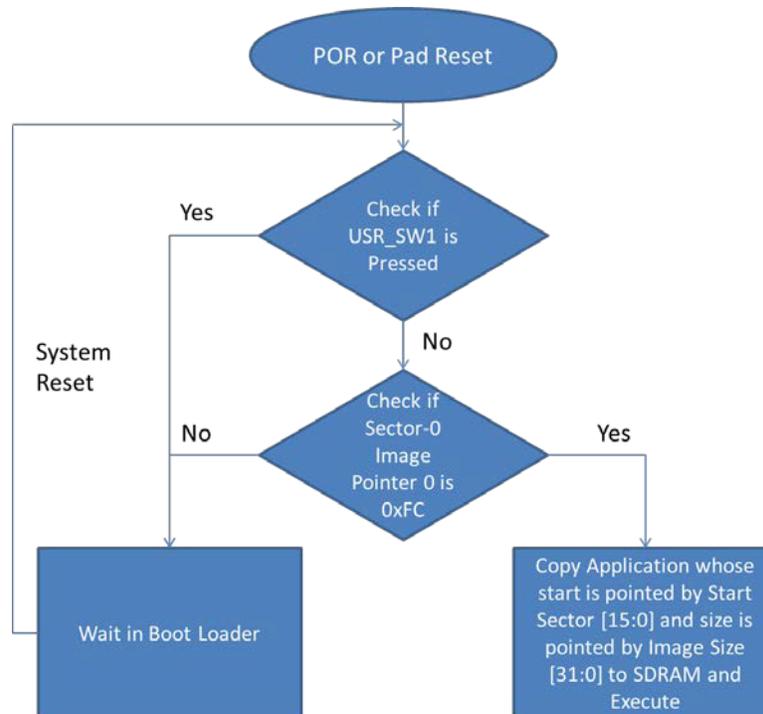


Figure 4. Example Code Program Flow

3.2.3 QSSI Bare Metal Code

The QSSI bare metal code configures the QSSI module of the TM4C1294NCPDT to perform advanced and quad mode programming for write operations and advanced-, bi-, and quad-mode for read operations. For this example, the QSSI serial interface operates at 60 MHz, which demonstrates the maximum achievable throughput.

3.3 Test Setup

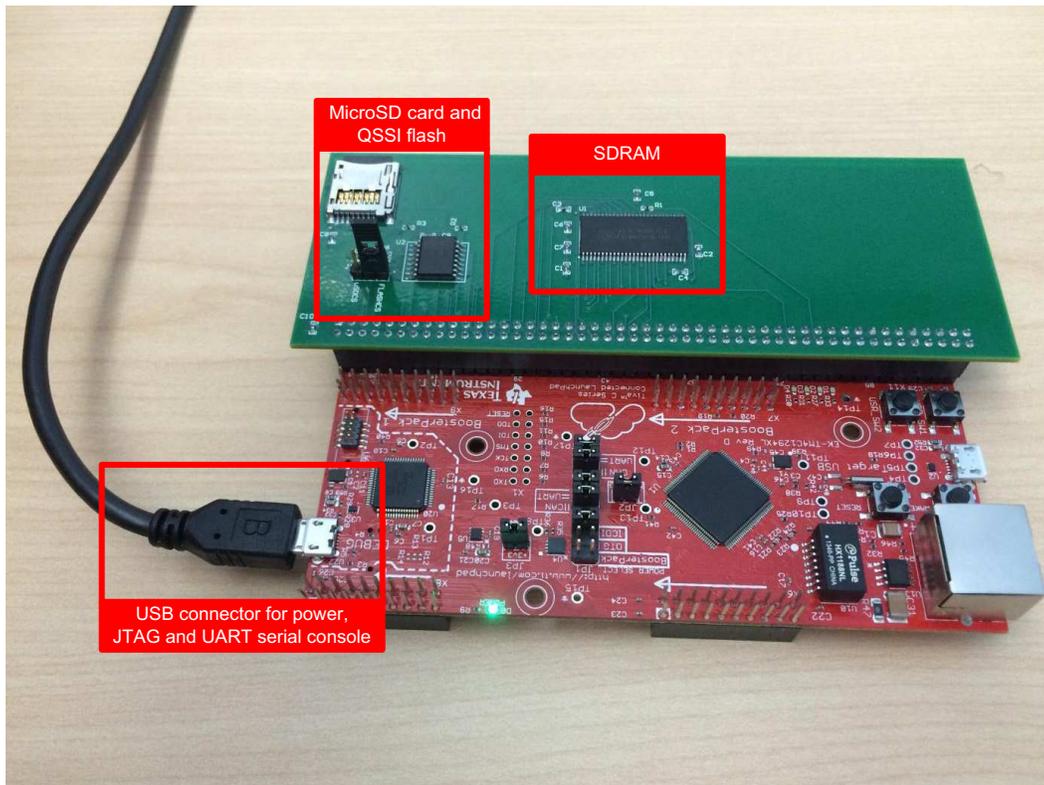
The test setup is:

1. Import the test project into CCS.
2. Build and compile the project.
3. Download the executable to the EK-TM4C1294XL LaunchPad lit.
4. Execute the test code on the target.

During execution, the test code first erases a 4KB sector and checks to ensure the erase was successful. If the erase was successful, the test code then performs a program and read operation on the 4KB sector. After performing the read operation, the test concludes with an erase and erase confirmation of the 4KB test sector.

3.3.1 Hardware Setup

Figure 5 shows an overview of the hardware setup. The USB cable on the left side of the EK-TM4C1294XL Connected LaunchPad kit provides power, connects to JTAG, and connects the UART for communication between a PC terminal window application such as Putty or Tera Term and the hardware. The SDRAM-NVM memory extender uses header X11 to connect to the LaunchPad kit.



Download and install a serial console application (for example, PuTTY, TeraTerm, and so forth), Code Composer Studio™ IDE v6.0.1 or later, and TivaWare™ for C Series v2.1.0-12573 or later to use this example.

Figure 5. Full Test Assembly

3.3.2 Software Setup (microSD Card Boot With SDRAM Code Execution)

1. Download the example software package from [TIDM-TM4C129SDRAMNVM](#). Unzip the software package.
2. Launch Code Composer Studio IDE v6.0.1 or later → Click Import→ Click CCS Projects→ Click Next. Browse to the directory with the software examples. Select "ektm4c129_sdcard_bootloader", "ektm4c129_sdcard_boot_demo1", and "ektm4c129_sdcard_boot_demo2". Click Finish.

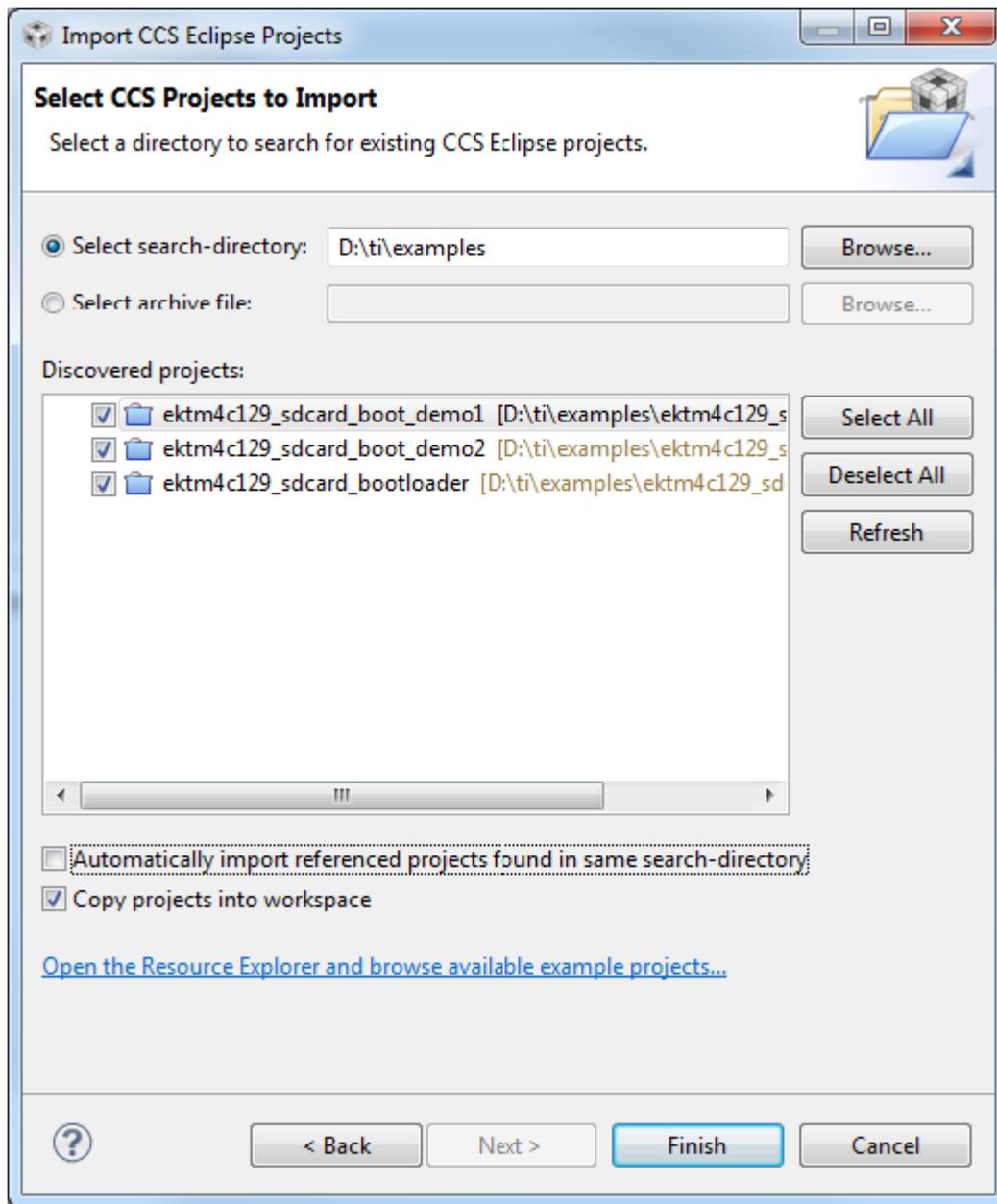


Figure 6. SD Card Boot Project Import

- Build each project. To build a project, right-click on a project. Click Rebuild Project. Ensure the projects compile free of errors.

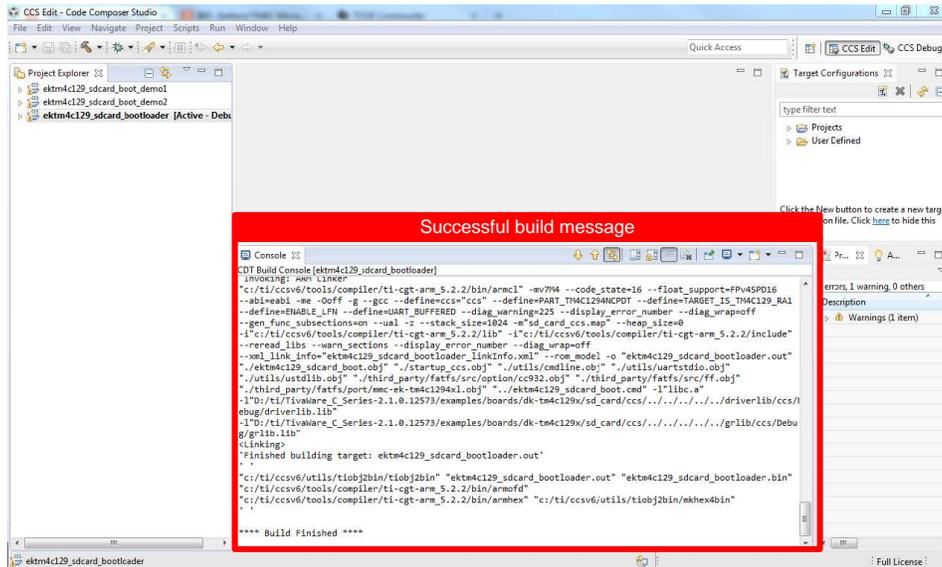
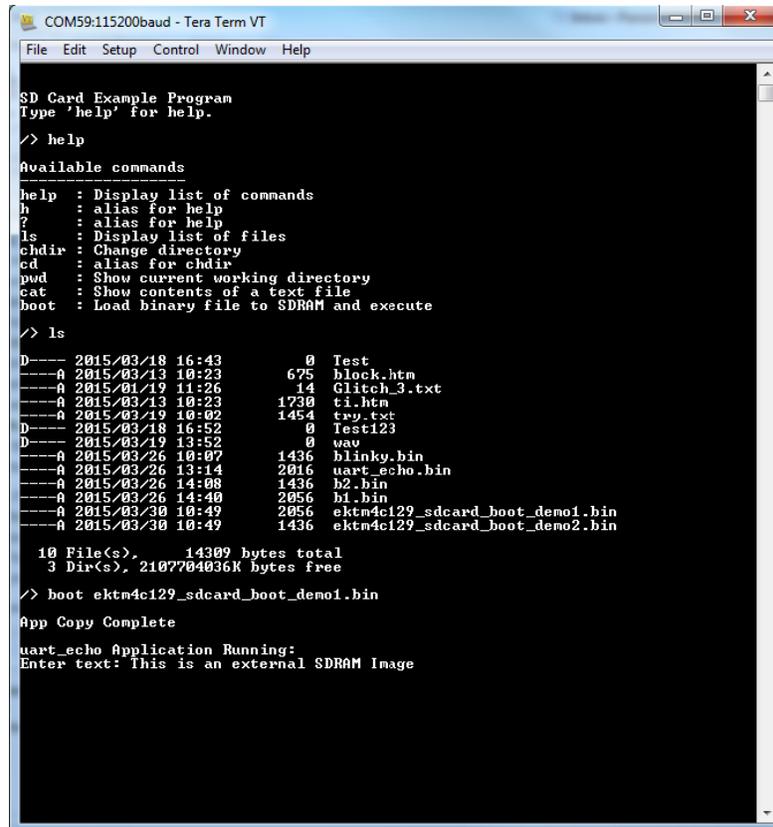


Figure 7. Compiling SD Card Boot Software

- Copy the bin files from "ektm4c129_sdcard_boot_demo1" and "ektm4c129_sdcard_boot_demo2" to a microSD card connected to a PC. Insert the microSD card with the copied files into the slot on the SDRAM-NVM memory extender. Place jumper J1 on the uSDCS side to assert the microSD chip select.

- Press Debug to run the main bootloader, "ektm4c129_sdcard_bootloader", in the TM4C1294NCPDT flash. Press Play when the code loads. When you will see the prompt for the microSD card on a serial console, type "help" to see the options. Type "ls" to see the list of files. To select a binary image, type "boot ektm4c129_sdcard_boot_demo1". The uart_echo application will copy and execute from the SDRAM.



```

COM59:115200baud - Tera Term VT
File Edit Setup Control Window Help

SD Card Example Program
Type 'help' for help.

/> help

Available commands

help : Display list of commands
h    : alias for help
?    : alias for help
ls   : Display list of files
chdir: Change directory
cd   : alias for chdir
pwd  : Show current working directory
cat  : Show contents of a text file
boot : Load binary file to SDRAM and execute

/> ls

D---- 2015/03/18 16:43      0 Test
----A 2015/03/13 10:23    675 block.htm
----A 2015/01/19 11:26     14 Glitch_3.txt
----A 2015/03/13 10:23   1730 ti.htm
----A 2015/03/19 10:02   1454 try.txt
D---- 2015/03/18 16:52      0 Test123
D---- 2015/03/19 13:52      0 wav
----A 2015/03/26 10:07   1436 blinky.bin
----A 2015/03/26 13:14   2016 uart_echo.bin
----A 2015/03/26 14:08   1436 b2.bin
----A 2015/03/26 14:40   2056 b1.bin
----A 2015/03/30 10:49   2056 ektm4c129_sdcard_boot_demo1.bin
----A 2015/03/30 10:49   1436 ektm4c129_sdcard_boot_demo2.bin

 10 File(s),  14309 bytes total
   3 Dir(s), 2107704036K bytes free

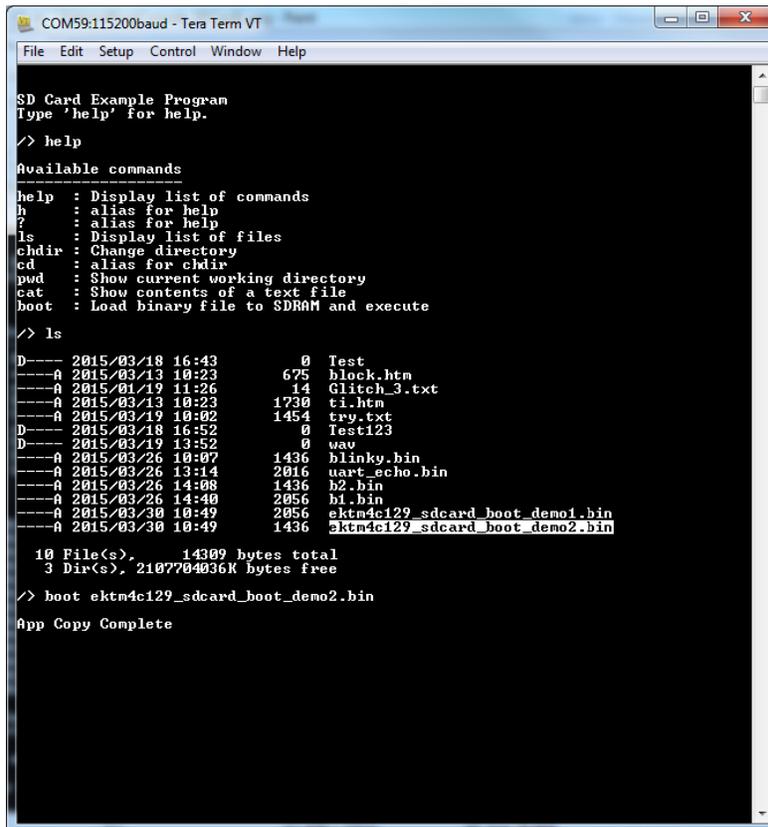
/> boot ektm4c129_sdcard_boot_demo1.bin

App Copy Complete

uart_echo Application Running:
Enter text: This is an external SDRAM Image
    
```

Figure 8. Serial Console Output For ektm4c129_sdcard_boot_demo1.bin

Type "boot ektn4c129_sdcard_boot_demo2" to copy and execute the D2 LED blinky application from the SDRAM.



```

COM59:115200baud - Tera Term VT
File Edit Setup Control Window Help

SD Card Example Program
Type 'help' for help.

/> help

Available commands
-----
help : Display list of commands
h    : alias for help
?    : alias for help
ls   : Display list of files
chdir: Change directory
cd   : alias for chdir
pwd  : Show current working directory
cat  : Show contents of a text file
boot : Load binary file to SDRAM and execute

/> ls

D----- 2015/03/18 16:43      0 Test
D----- 2015/03/13 10:23    675 block.htm
----- 2015/01/19 11:26      14 Glitch 3.txt
----- 2015/03/13 10:23   1730 ti.htm
----- 2015/03/19 10:02   1454 try.txt
D----- 2015/03/18 16:52      0 Test123
D----- 2015/03/19 13:52      0 wav
----- 2015/03/26 10:07   1436 blinky.bin
----- 2015/03/26 13:14   2016 uart_echo.bin
----- 2015/03/26 14:08   1436 b2.bin
----- 2015/03/26 14:40   2056 b1.bin
----- 2015/03/30 10:49   2056 ektn4c129_sdcard_boot_demo1.bin
----- 2015/03/30 10:49   1436 ektn4c129_sdcard_boot_demo2.bin

 10 File(s), 14309 bytes total
  3 Dir(s), 2107704036K bytes free

/> boot ektn4c129_sdcard_boot_demo2.bin

App Copy Complete
    
```

Figure 9. Serial Console Output For ektn4c129_sdcard_boot_demo2.bin

6. To restart the microSD card boot, press reset. The current application does not jump to the microSD card prompt.

3.3.3 Software Setup (QSSI Serial Flash Boot with SDRAM Code Execution)

1. Download the example software package from [TIDM-TM4C129SDRAMNVM](#). Unzip the software package.
2. Launch Code Composer Studio v6.0.1 or later → Click Import → Click CCS Projects → Click Next. Browse to the directory with the software examples. Select "ektm4c129_qssi_bootloader", "ektm4c129_qssi_boot_demo1", and "ektm4c129_sdcard_boot_demo2". Click Finish.

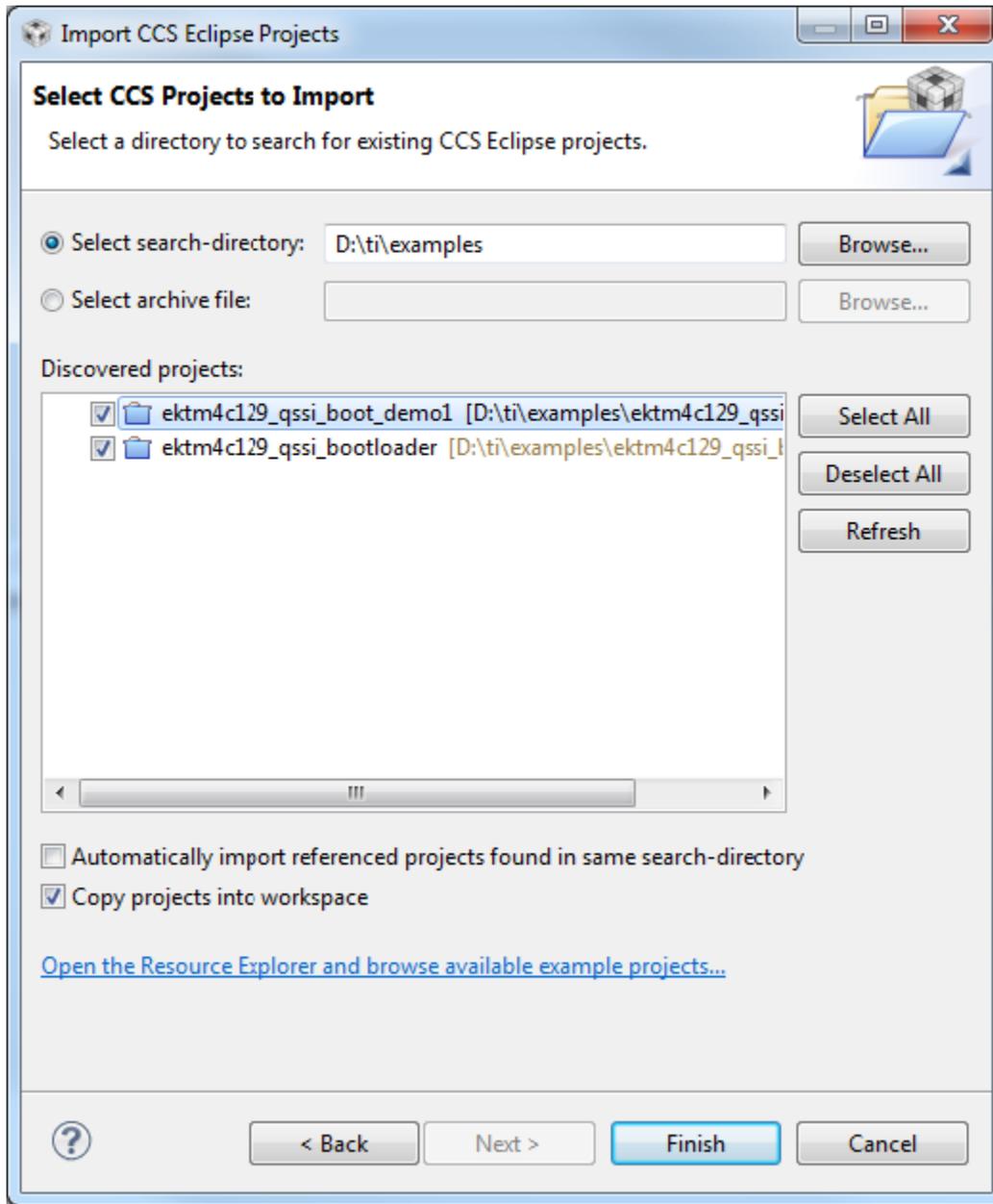


Figure 10. QSSI Example Project Import

- Build each project. To build a project, right-click on a project. Click Rebuild Project. Ensure the projects compile free of errors.

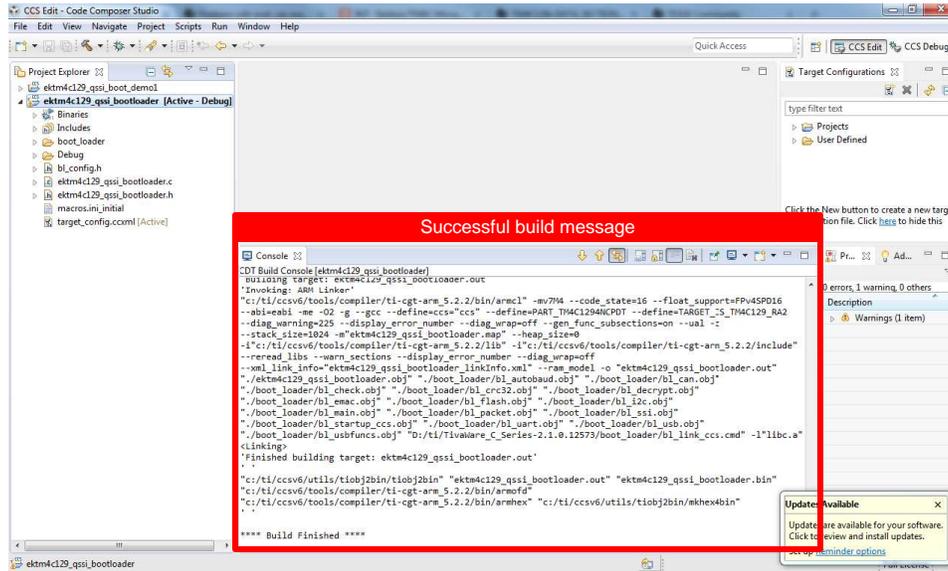


Figure 11. Compiling the QSSI Software Example

- Place jumper J1 on the FLASHCS side. Use the LM Flash programmer to download ektm4c129_qssi_bootloader to an TM4C1294NCPDT Connected LaunchPad kit that has been erased. After performing checks for QSSI and SDRAM memory, the bootloader activates UART0 to download an image to the external QSSI flash. For the setting of the LM flash programmer, see [Figure 12](#) and [Figure 13.](#))

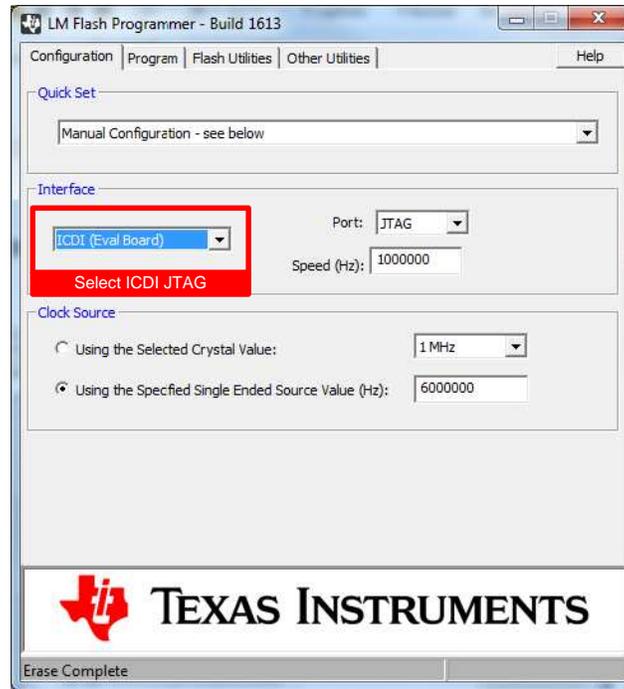


Figure 12. LM Flash Programmer Main Bootloader Programming - Interface Setting (JTAG)

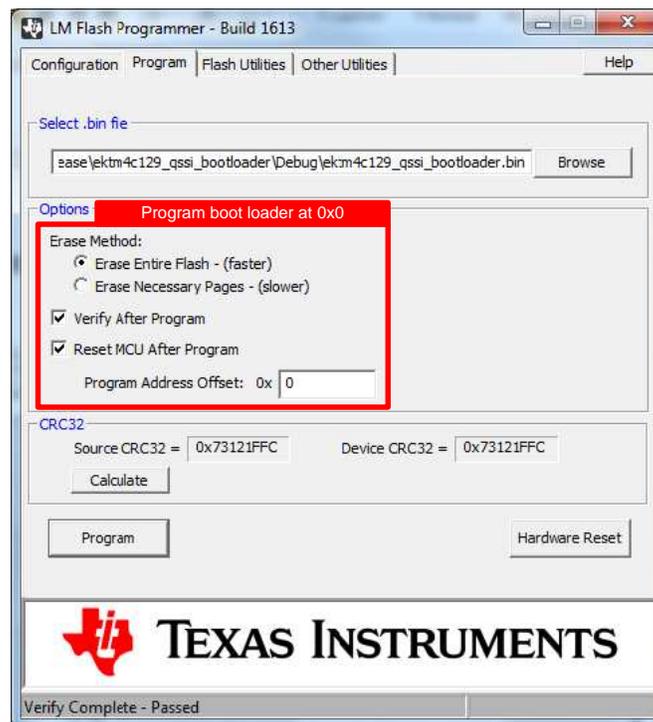


Figure 13. LM Flash Programmer Main Bootloader Programming - Options Settings

- Using the LM Flash Programmer in serial mode, download ektm4c129_qssi_boot_demo1. Ensure Disable Auto Baud Support is checked. Select the correct COM port. Ensure the Transfer Size is 64 bytes or less. On the Program Tab, select the Program Address Offset as the start of a sector of QSSI Flash other than Sector-0. (For the setting of the LM Flash Programmer for downloading the demo code, see Figure 14 and Figure 15. When the demo code downloads, the LED D3 will start blinking.)

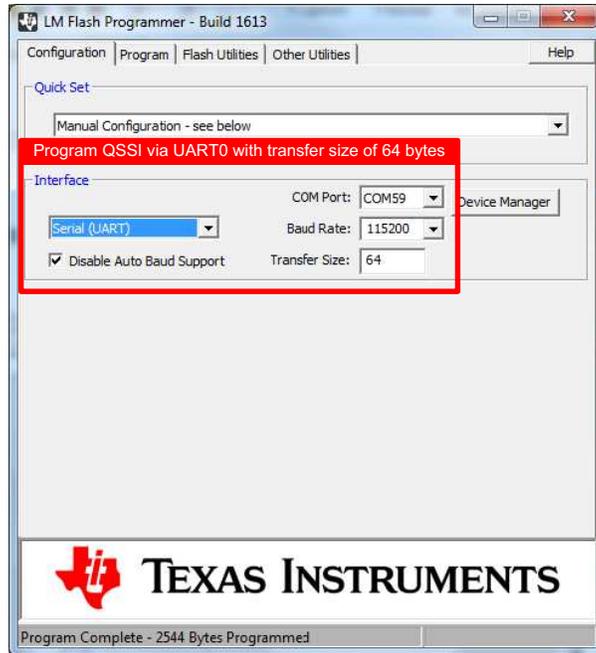


Figure 14. LM Flash Programmer QSSI Boot Demo1 Programming - Interface Settings (UART)

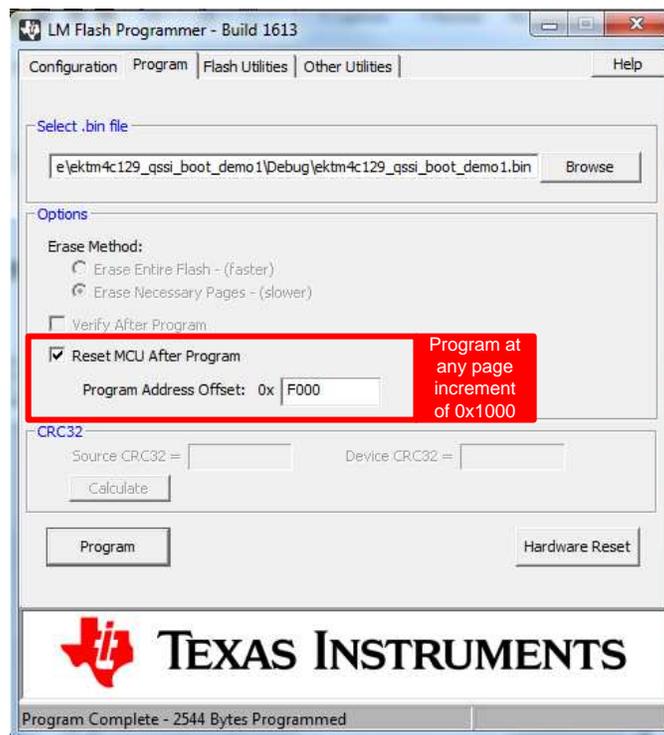


Figure 15. LM Flash Programmer QSSI Boot Demo 1 Programming - Options (Address Offset)

- Use USR_SW2 to accelerate the blinking rate and USR_SW1 to reduce the blinking rate.

3.3.4 Software Setup (QSSI Bare Metal Example)

1. Download the example software package from [TIDM-TM4C129SDRAMNVM](#). Unzip the software package.
2. Launch Code Composer Studio v6.0.1 or later→ Click Import→ Click CCS Projects→ Click Next. Browse to the directory with the software examples. Select " ektm4c129_qssi_example". Click Finish.

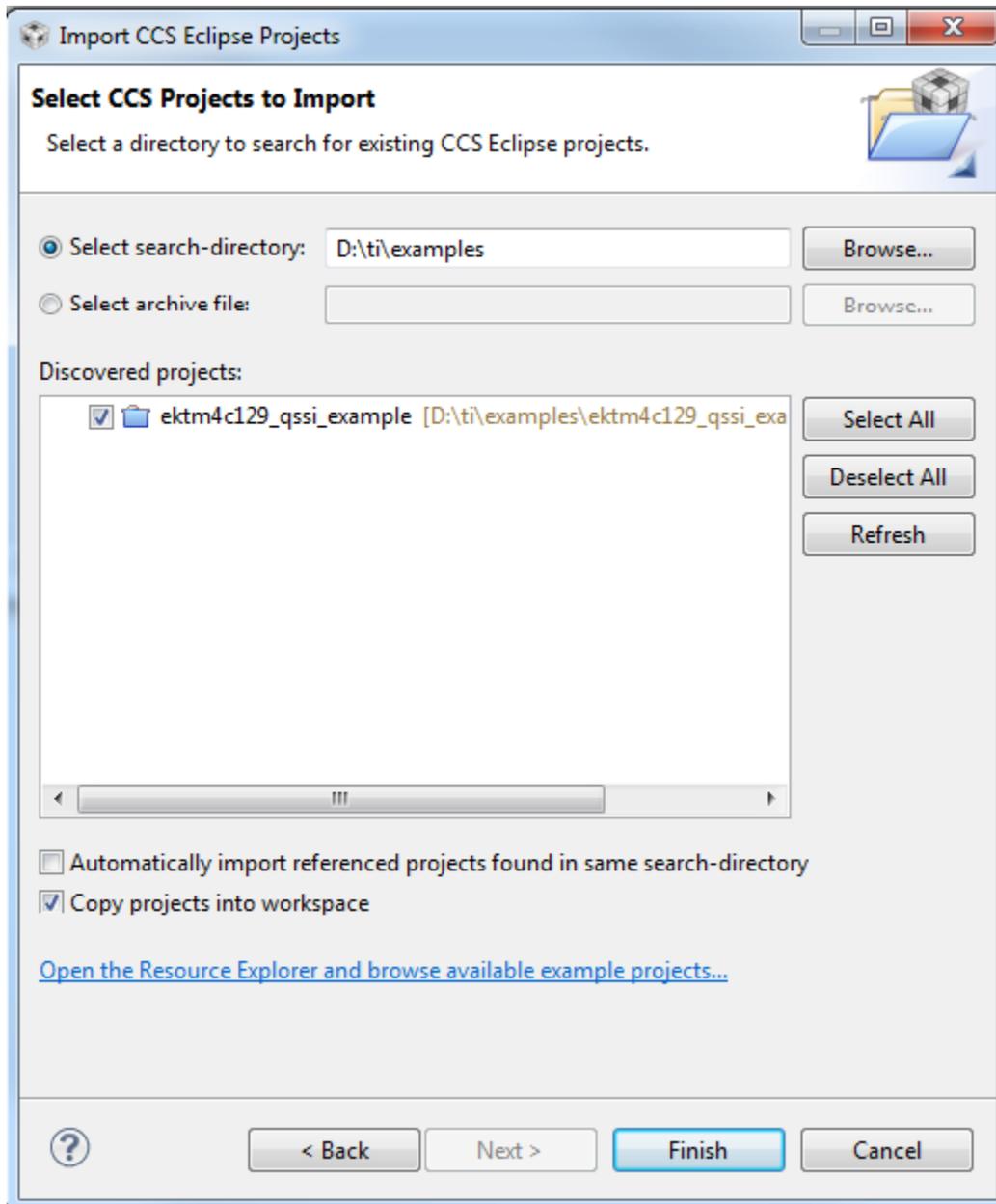


Figure 16. QSSI Bare Metal Project Import

- Build each project. To build a project, right-click on a project. Click Rebuild Project. Ensure the projects compile free of errors.

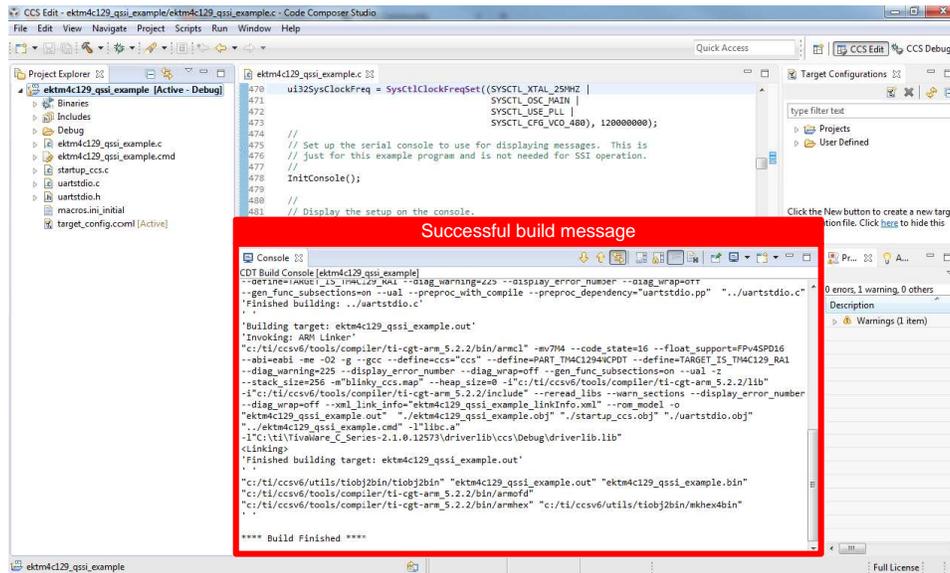


Figure 17. QSSI Bare Metal Compile

- Ensure the J1 jumper is connected to the FLASHCS side. Press Debug to download "ektm4c129_qssi_example" and load the code into the TM4C1294NCPDT flash. Press Play when the code has loaded. On the serial console, ensure you see the log file generated for erase, advanced mode program, quad mode program, advanced mode read, bi-mode read, and quad mode read.

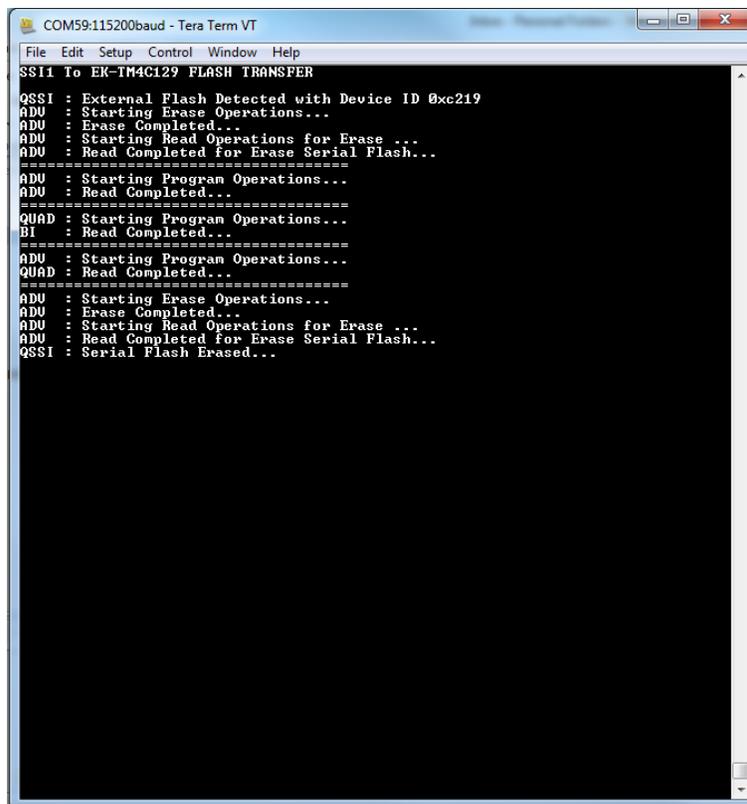


Figure 18. QSSI Bare Metal Example Serial Console Output

4 Design Files

4.1 Schematics

To download the schematics for the board, see the design files at [TIDM-TM4C129SDRAMNVM](#).

4.2 Bill of Materials

To download the bill of materials (BOM), see the design files at [TIDM-TM4C129SDRAMNVM](#).

4.3 PCB Layout Recommendations

When performing the layout, make sure that the EPI0S31 (the SDRAM clock pin) has the shortest trace. To minimize reflections from the shared data and address pins, use a single route from the connector pin to the address pin or data pin without creating a stub.

4.4 Altium Project

Altium Designer® project files are not available for this reference design.

4.5 Gerber Files

To download the Gerber files, see the design files at [TIDM-TM4C129SDRAMNVM](#).

4.6 Assembly Drawings

Assembly drawings are not available for this reference design.

5 Software Files

To download the software files, see the design files at [TIDM-TM4C129SDRAMNVM](#).

6 Related Documentation

1. [IS42/45R86400D/16320D/32160D 16Mx32, 32Mx16, 64Mx8 512Mb SDRAM data sheet](#)
2. [MX66L51235F 3V, 512M-BIT \[x 1/x 2/x 4\] CMOS MXSMIO® \(SERIAL MULTI I/O\) FLASH MEMORY data sheet](#)

6.1 Trademarks

E2E, LaunchPad, Code Composer Studio, TivaWare are trademarks of Texas Instruments. Altium Designer is a registered trademark of Altium LLC or its affiliated companies. Arm, Cortex are registered trademarks of Arm Limited. All other trademarks are the property of their respective owners.

6.2 Third-Party Products Disclaimer

TI'S PUBLICATION OF INFORMATION REGARDING THIRD-PARTY PRODUCTS OR SERVICES DOES NOT CONSTITUTE AN ENDORSEMENT REGARDING THE SUITABILITY OF SUCH PRODUCTS OR SERVICES OR A WARRANTY, REPRESENTATION OR ENDORSEMENT OF SUCH PRODUCTS OR SERVICES, EITHER ALONE OR IN COMBINATION WITH ANY TI PRODUCT OR SERVICE.

7 About the Author

AMIT ASHARA is an application engineer at TI, where he develops applications for the TM4C12x family of high-performance microcontrollers. Amit brings to this role his extensive experience and expertise in high-speed digital and microcontroller system-level design. Amit earned his Bachelor of Engineering (BE) from the University of Pune in India.

Revision History

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

Changes from April 7, 2015 to February 1, 2019	Page
• Editorial, format, and layout changes throughout document	1
• Removed link to layout prints (not available).....	18
• Removed link to Altium project (not available).....	18
• Removed link to assembly drawings (not available)	18
• Updated related documentation links	18

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
Copyright © 2022, Texas Instruments Incorporated