参考資料

TI Designs: TIDM-TM4C129SDRAMNVM
高性能 MCU における NVM にコードを格納した SDRAM からの実行に関するリファレンス・デザイン

TEXAS INSTRUMENTS

概要

内蔵メモリが十分でない場合に、TM4C129x マイクロコントローラ (MCU) のメモリ空間を拡張できます。このリファレンス・デザインでは、TM4C129x MCU のハードウェア・インターフェイス要件とサンプル・ソフトウェアについて説明します。

リソース

TIDM-TM4C129SDRAMNVM	デザイン・フォルダ
EK-TM4C1294XL	ツール・フォルダ
TM4C1294NCPDT	プロダクト・フォルダ
TM4C123GH6PM	プロダクト・フォルダ
TPD4S012	プロダクト・フォルダ
TPS2052B	プロダクト・フォルダ
TPS62177	プロダクト・フォルダ
TPS73733-Q1	プロダクト・フォルダ

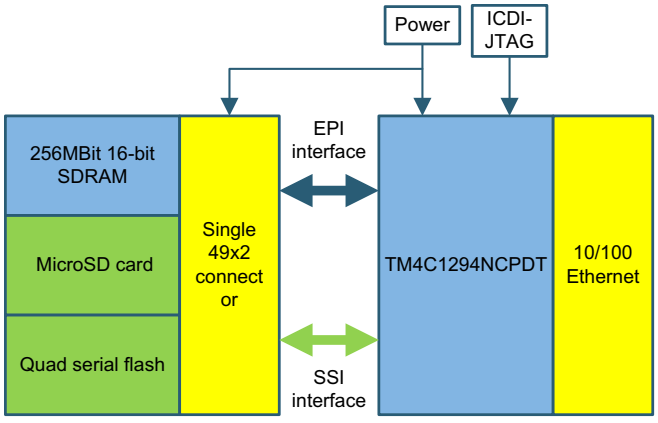
TI E2E™ Community　E2E™エキスパートに質問

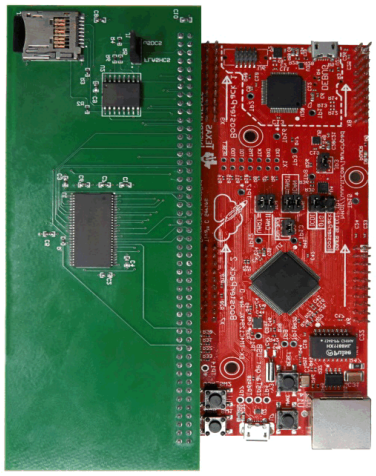
特長

- メモリ・スループットが高く占有面積が大きい用途向けに、60MHz の外部ペリフェラル・インターフェイス (EPI) により使用可能なメモリ空間を 512Mb の 16 ビット SDRAM へと拡張
- Arm® Cortex®-M4F ベース MCU TM4C1294 Connected LaunchPad™ 評価キット用に設計
- SD カードまたはクワッド・シリアル・フラッシュ不揮発性メモリ用のシリアル・インターフェイス・ブートローダを実装
- ベアメタル外部メモリ・ロガーのカスタム実装用にクワッド・シリアル・フラッシュ・モード・コマンドを追加サポート
- Code Composer Studio™ IDE に対応したプロジェクト例を含むソース・コードを提供

アプリケーション

- 対話型ヒューマン・マシン・インタフェース
- 産業用オートメーション
- モノのインターネット (IoT) ソリューション
- 試験 / 測定



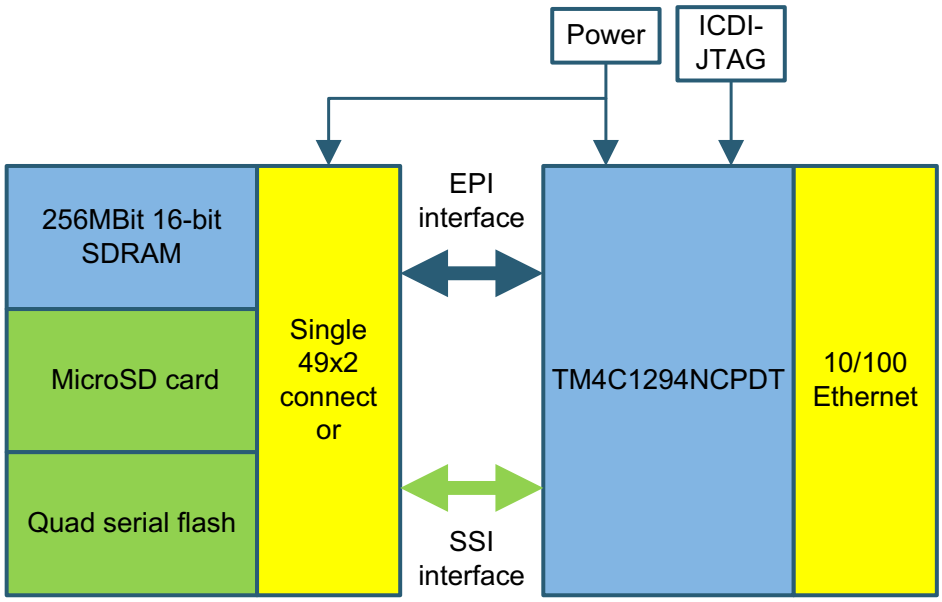


1 System Description

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.

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

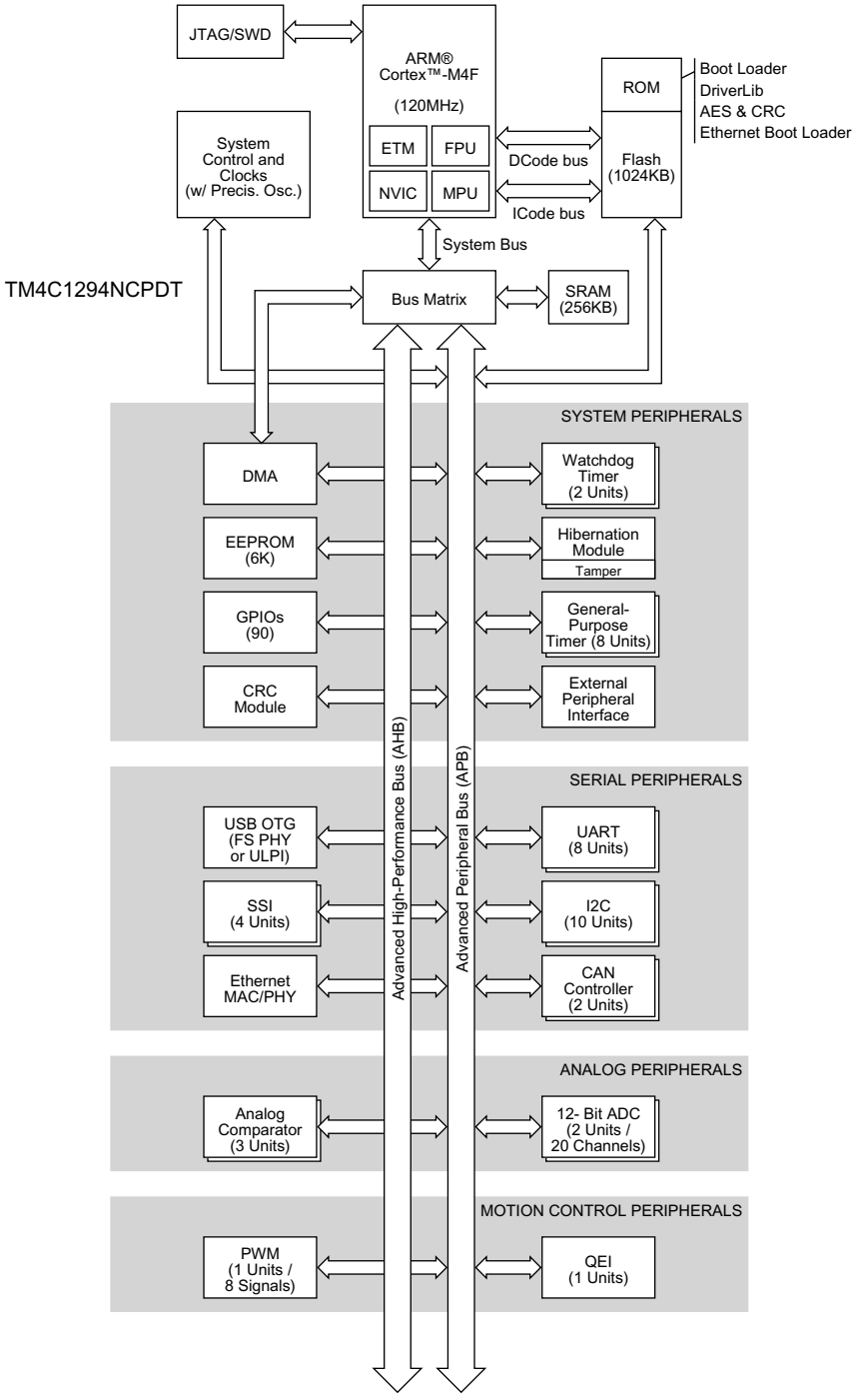


図 1. TM4C1294NCPDT Microcontroller High-Level Block Diagram

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. 図 2 shows an overview of the connector mounting for the daughtercard.

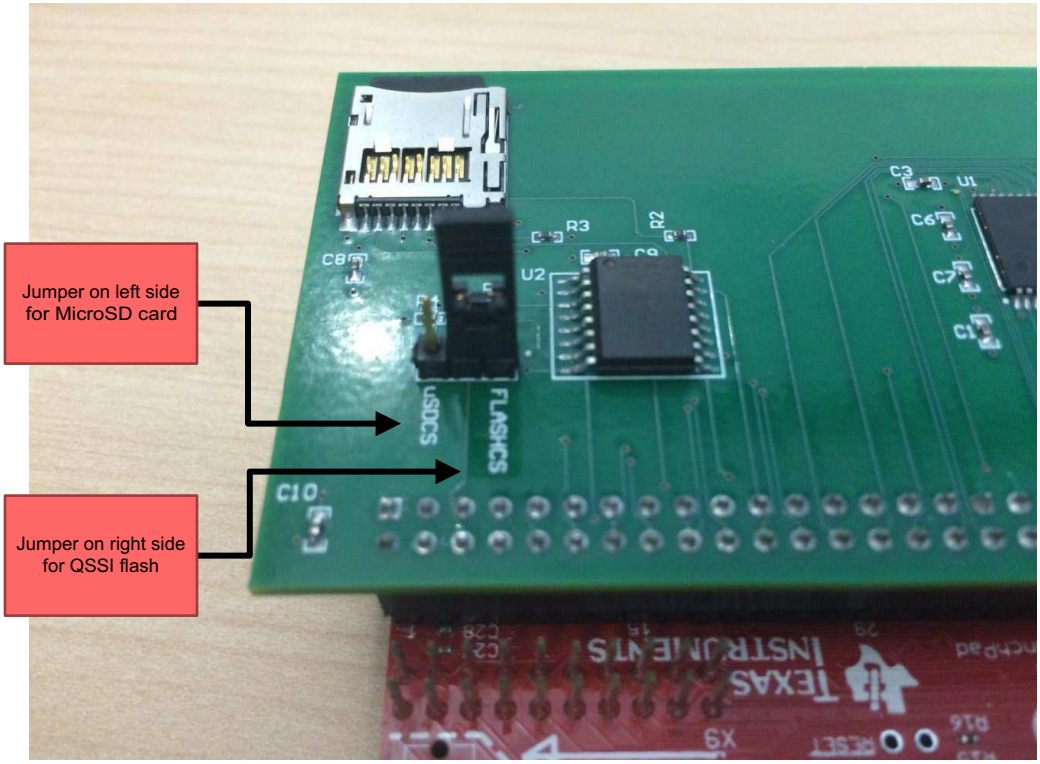


図 2. SDRAM-NVM Connector Mounting

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 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. 図 3 shows the structure of the information held in Sector-0 pertaining to an actual application image.

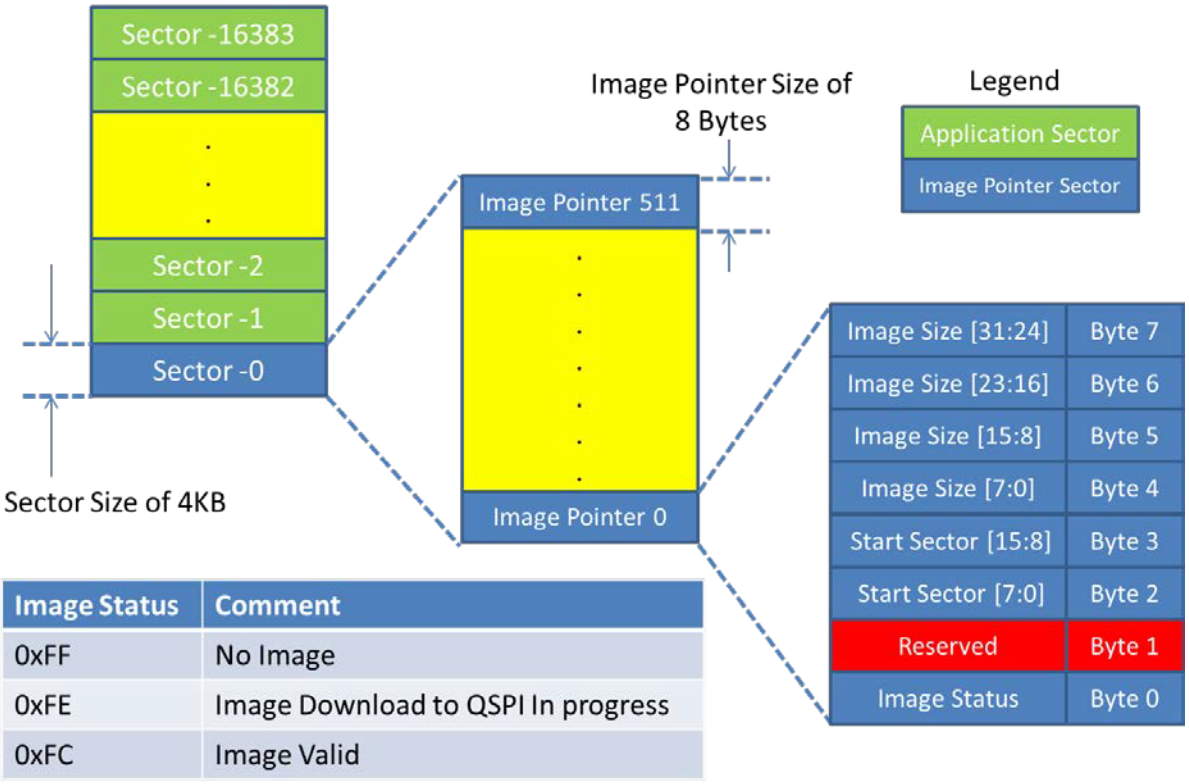


Image Status	Comment
0xFF	No Image
0xFE	Image Download to QSPI In progress
0xFC	Image Valid

図 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. 図 4 shows a flow chart of how the code operates.

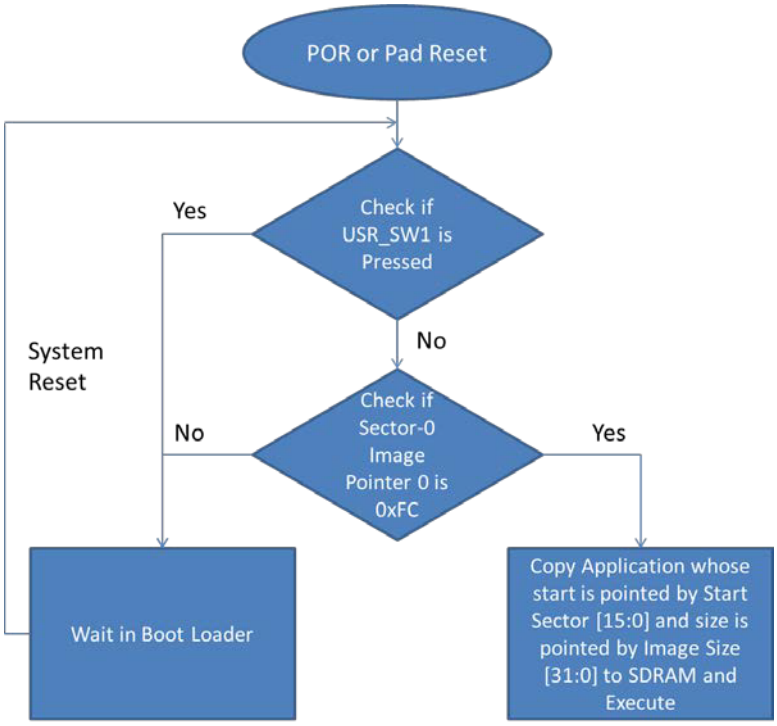


図 4. Example Code Program Flow

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 programing 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

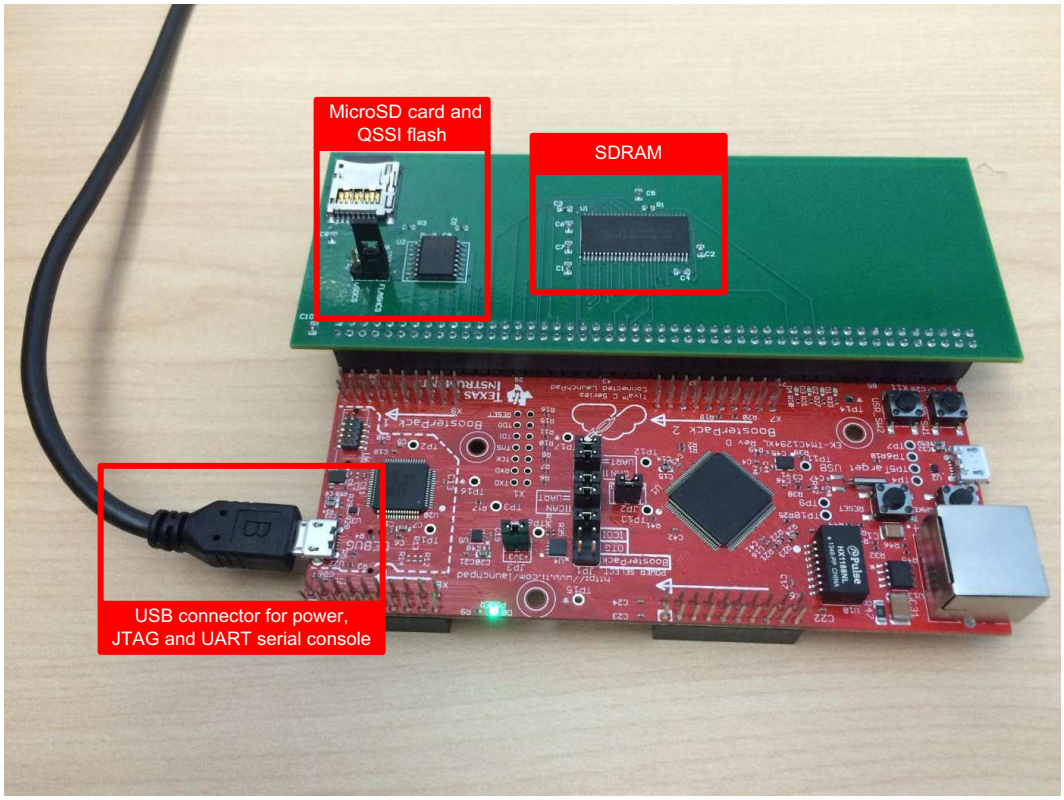
The test setup is:

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

図 5 shows an overview of the hardware setup. The USB cable on the left side of the EK-TM4C129XL 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.

図 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.

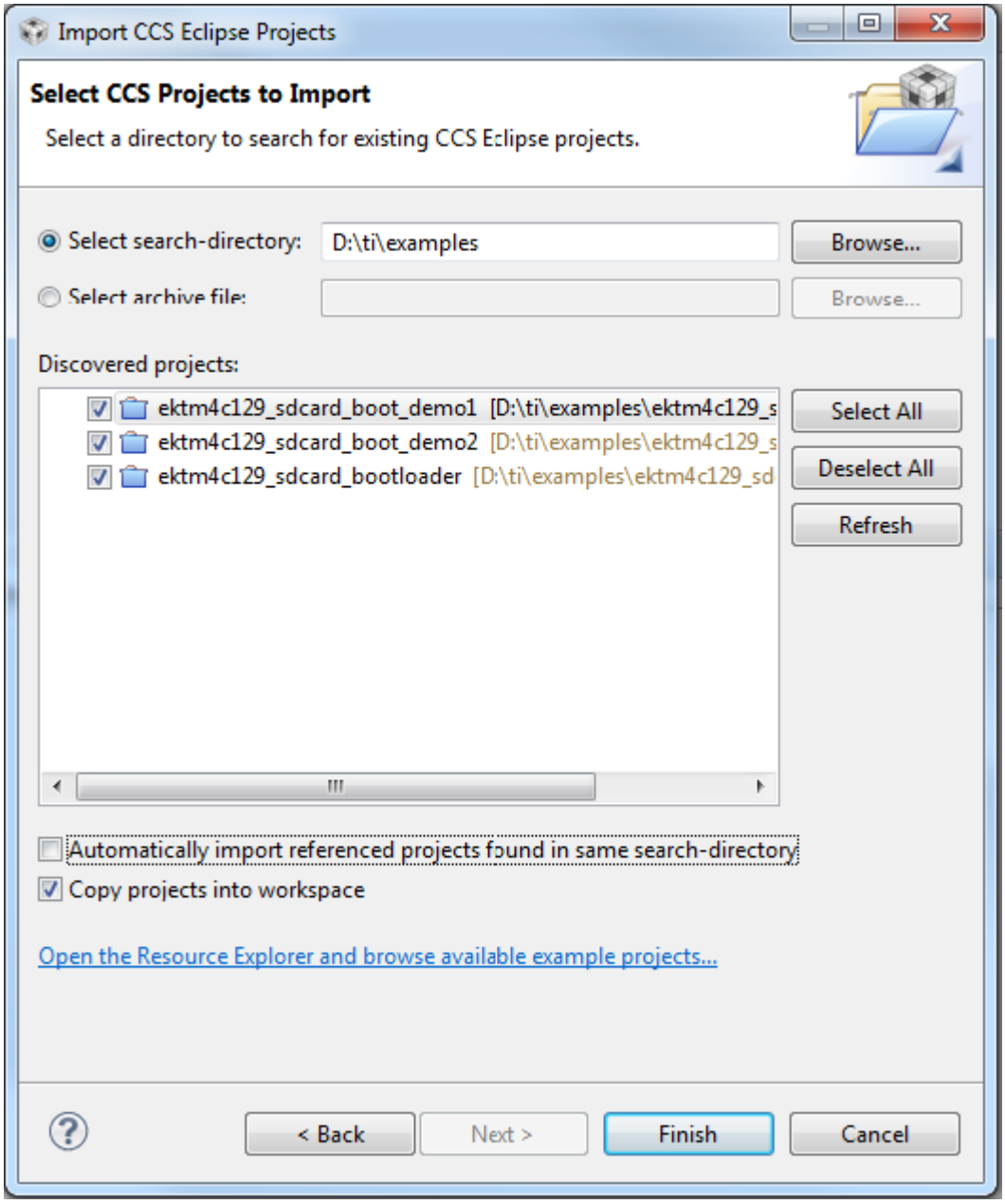


図 6. SD Card Boot Project Import

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

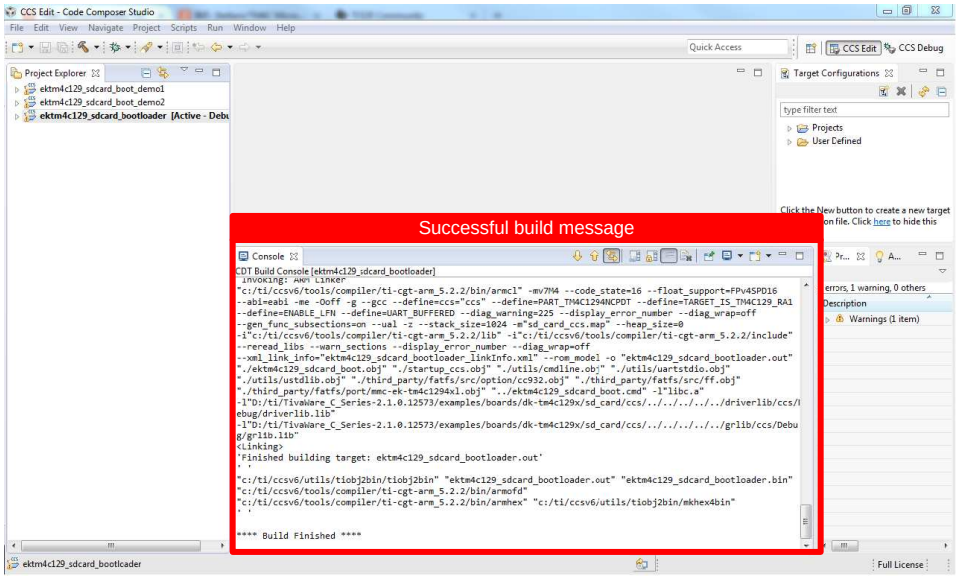


図 7. Compiling SD Card Boot Software

4. 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.

5. 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.

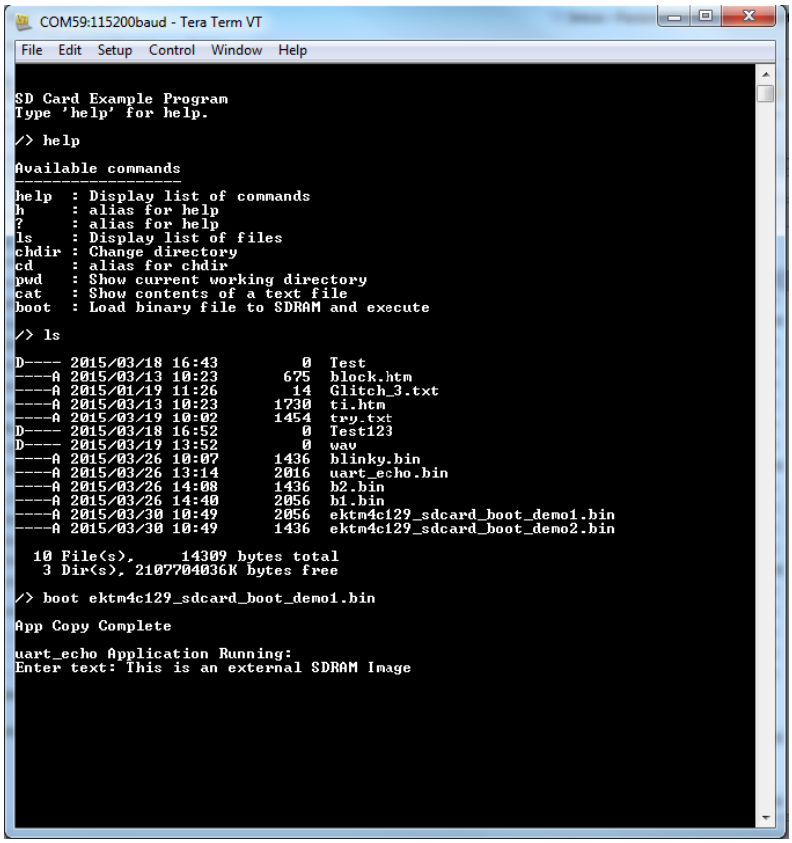


図 8. Serial Console Output For ektm4c129_sdcard_boot_demo1.bin

Type "boot ektm4c129_sdcard_boot_demo2" to copy and execute the D2 LED blinky application from the SDRAM.

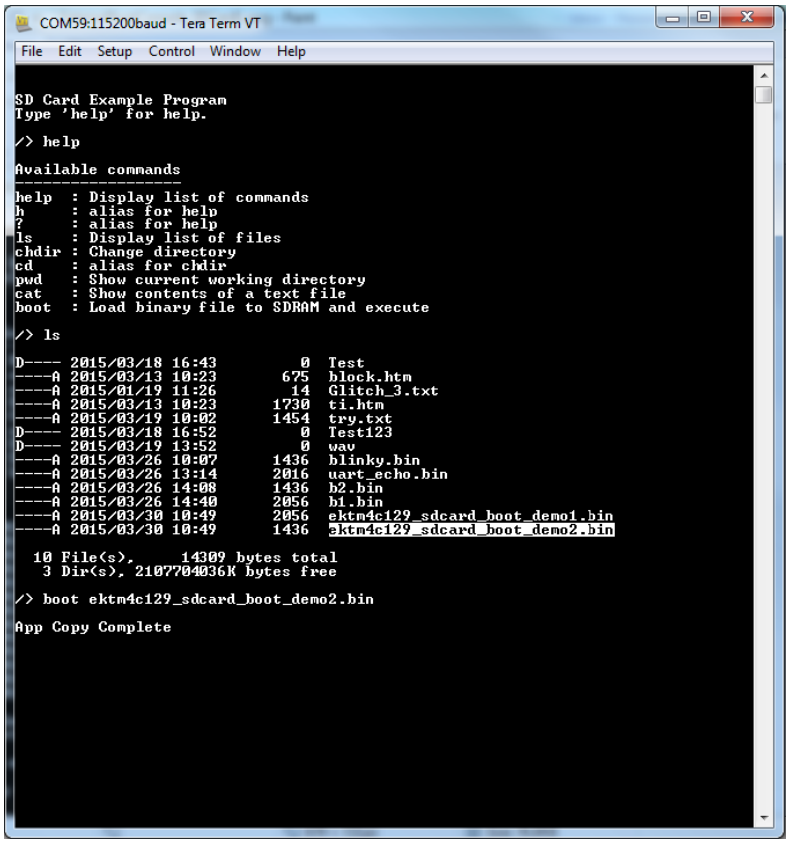


図 9. Serial Console Output For ektm4c129_sdcard_boot_demo2.bin

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.

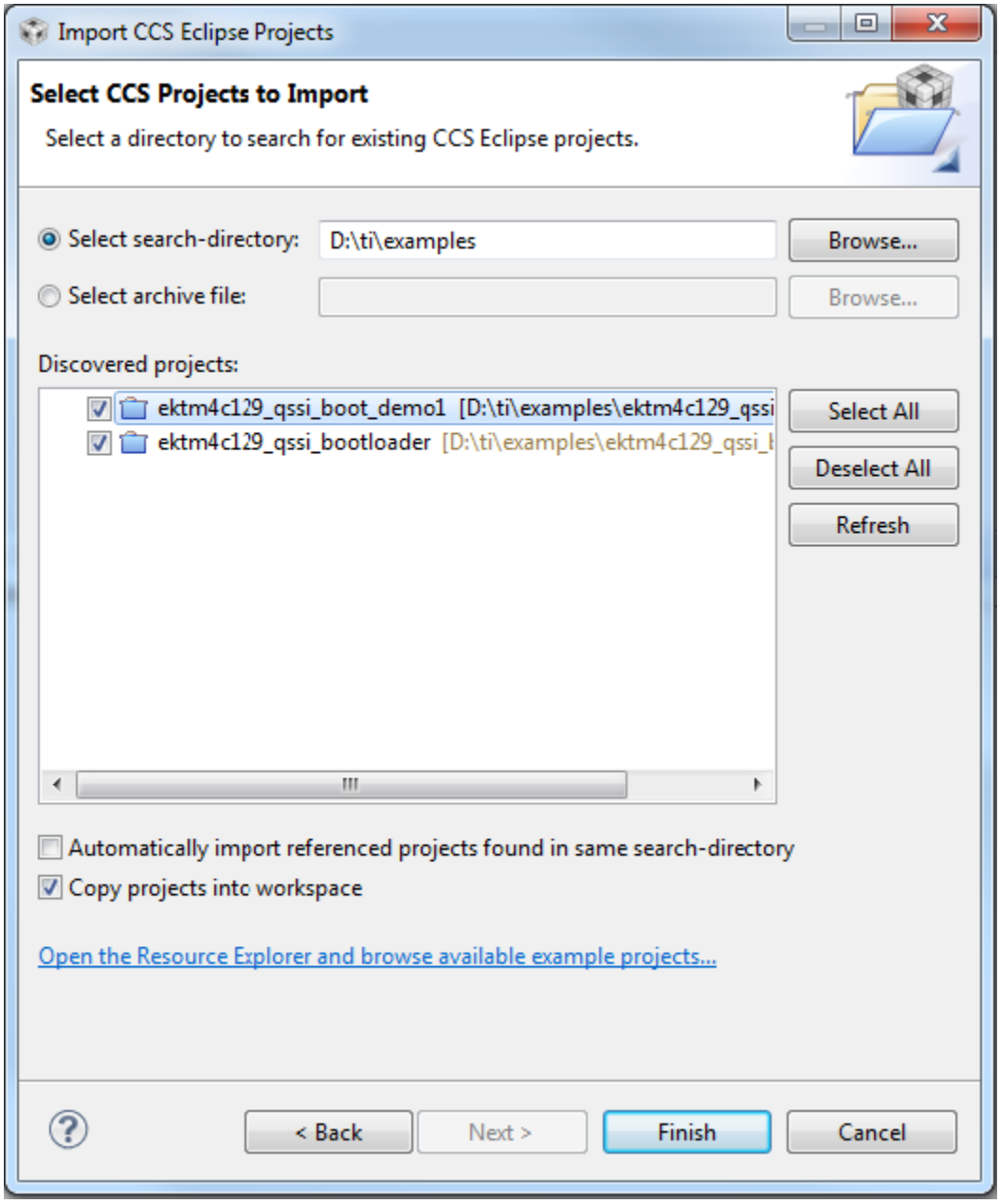


図 10. QSSI Example Project Import

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

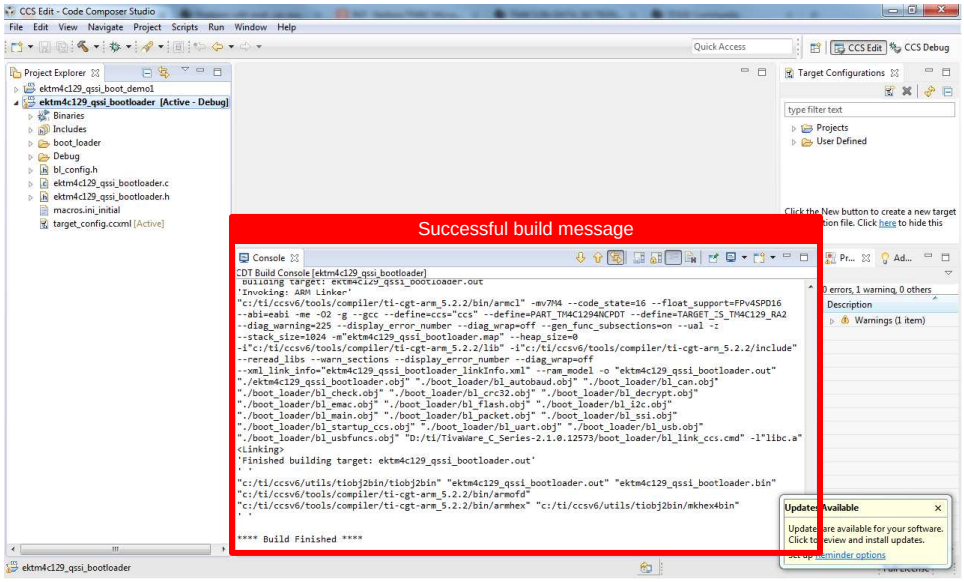


図 11. Compiling the QSSI Software Example

4. 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 図 12 and 図 13.)

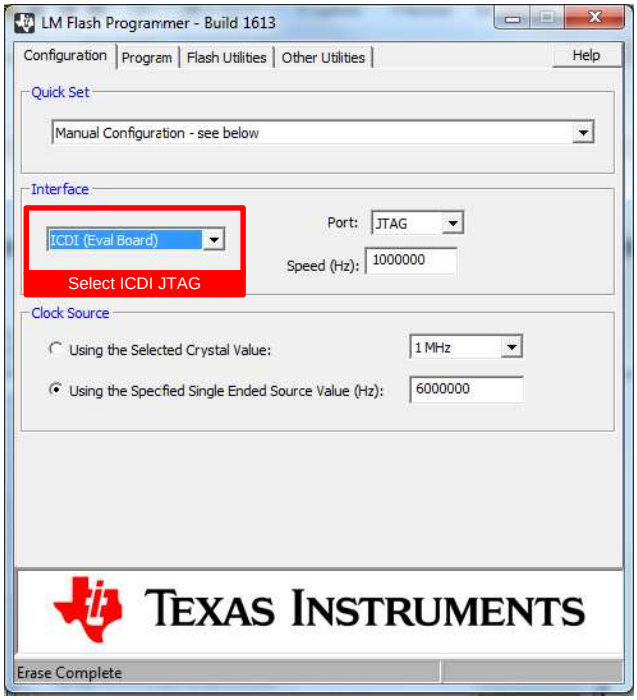


図 12. LM Flash Programmer Main Bootloader Programming - Interface Setting (JTAG)

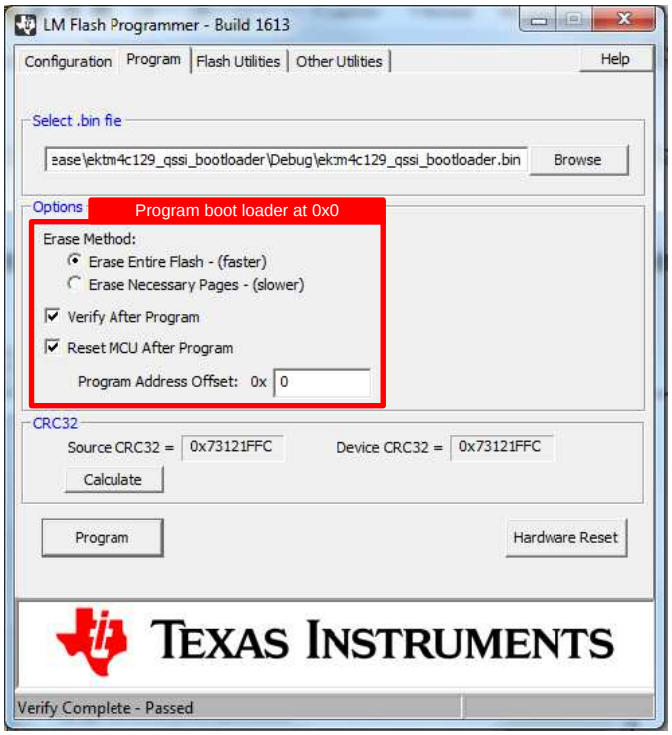


図 13. LM Flash Programmer Main Bootloader Programming - Options Settings

5. 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 図 14 and 図 15. When the demo code downloads, the LED D3 will start blinking.)

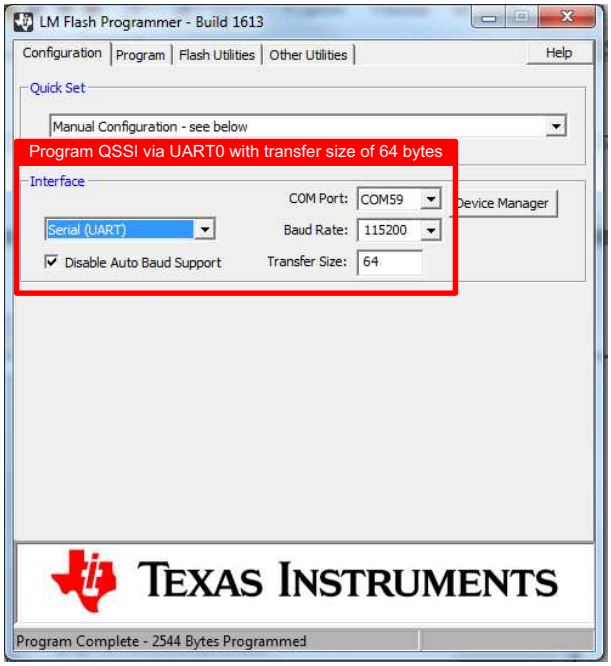


図 14. LM Flash Programmer QSSI Boot Demo1 Programming - Interface Settings (UART)

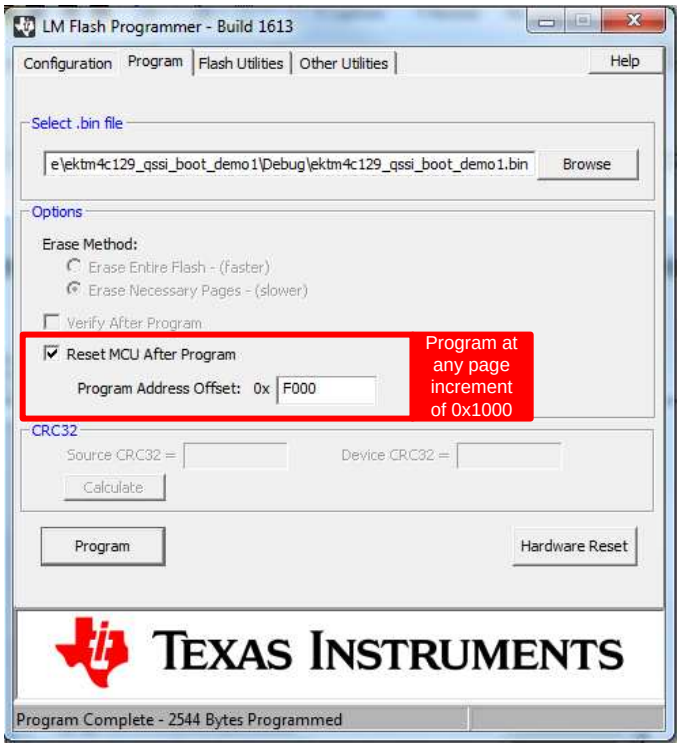


図 15. LM Flash Programmer QSSI Boot Demo 1 Programming - Options (Address Offset)

6. 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.

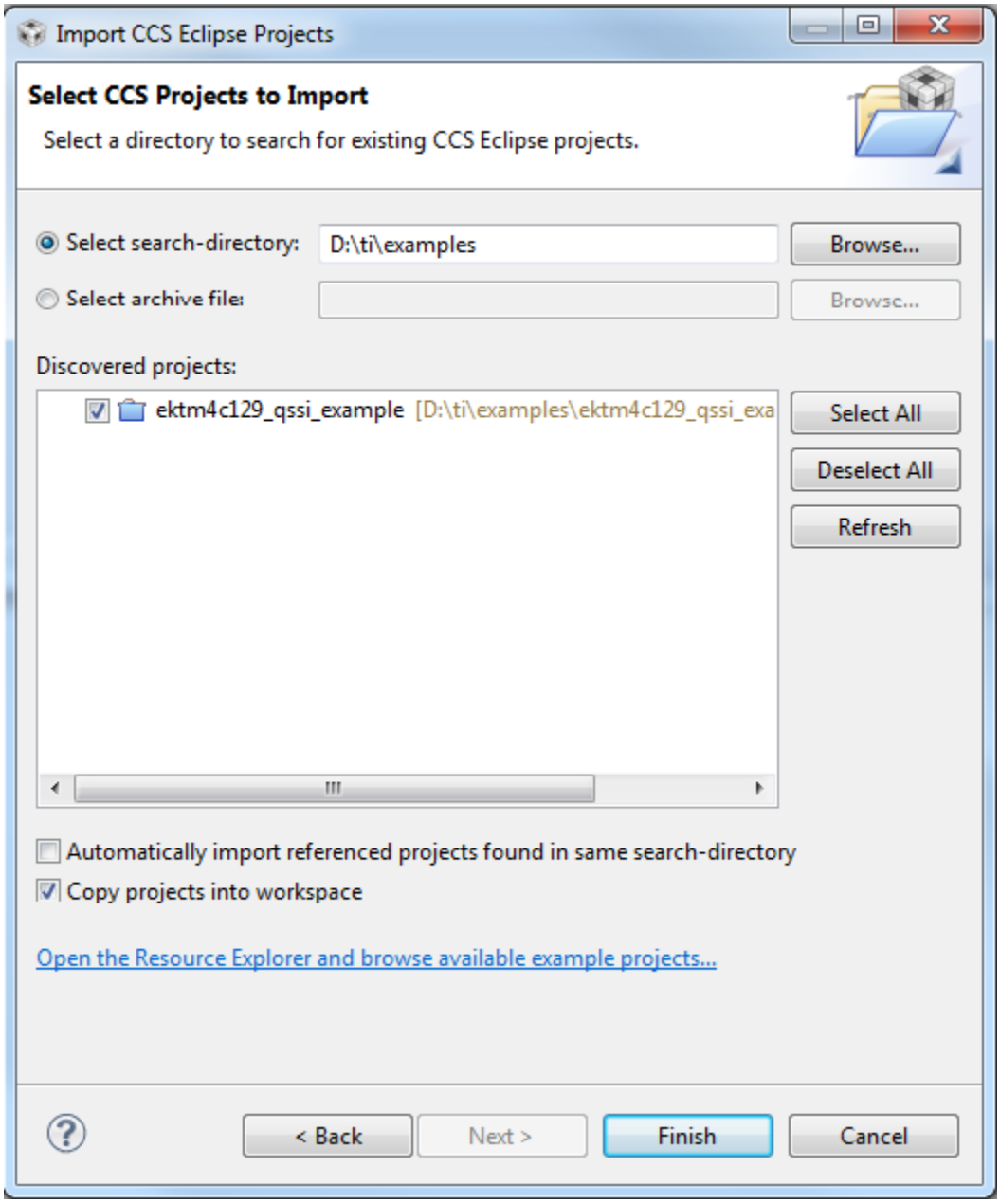


図 16. QSSI Bare Metal Project Import

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

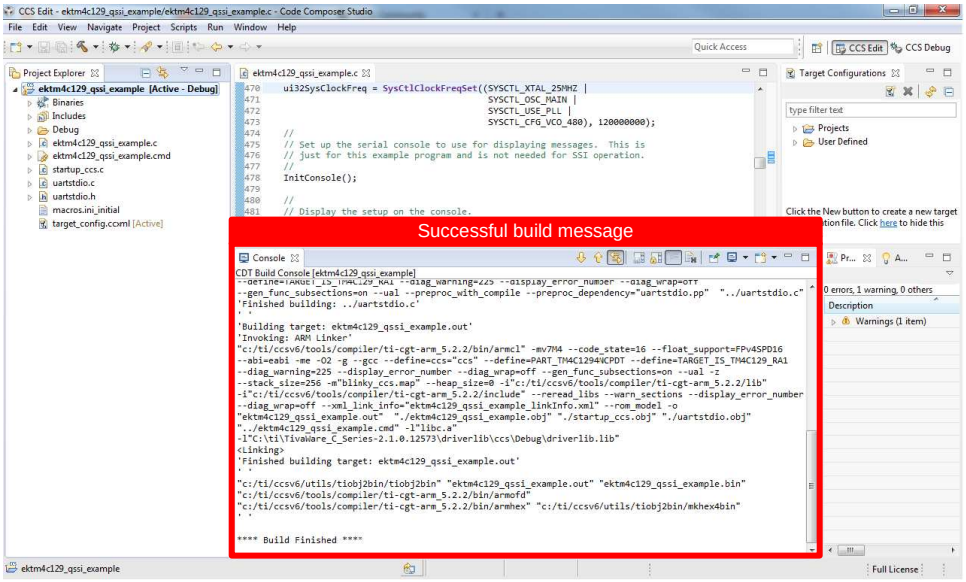


図 17. QSSI Bare Metal Compile

4. 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.

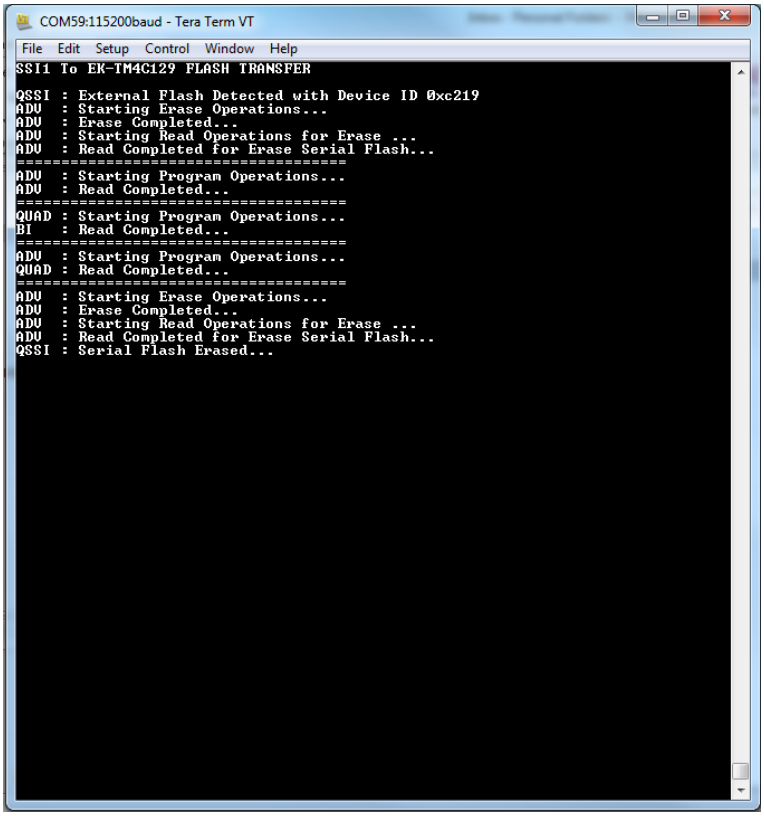


図 18. QSSI Bare Metal Example Serial Console Output

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 商標

6.2 Third-Party Products Disclaimer

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.

改訂履歴

資料番号末尾の英字は改訂を表しています。その改訂履歴は英語版に準じています。

2015年4月7日発行分から2019年2月1日発行分への変更 **Page**

- ドキュメント全体にわたる編集、フォーマット、レイアウトの変更 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

重要なお知らせと免責事項

TI は、技術データと信頼性データ (データシートを含みます)、設計リソース (リファレンス· デザインを含みます)、アプリケーションや設計に関する各種アドバイス、Web ツール、安全性情報、その他のリソースを、欠陥が存在する可能性のある「現状のまま」提供しており、商品性および特定目的に対する適合性の黙示保証、第三者の知的財産権の非侵害保証を含むいかなる保証も、明示的または黙示的にかかわらず拒否します。

これらのリソースは、TI 製品を使用する設計の経験を積んだ開発者への提供を意図したものです。(1) お客様のアプリケーションに適した TI 製品の選定、(2) お客様のアプリケーションの設計、検証、試験、(3) お客様のアプリケーションに該当する各種規格や、その他のあらゆる安全性、セキュリティ、規制、または他の要件への確実な適合に関する責任を、お客様のみが単独で負うものとします。

上記の各種リソースは、予告なく変更される可能性があります。これらのリソースは、リソースで説明されている TI 製品を使用するアプリケーションの開発の目的でのみ、TI はその使用をお客様に許諾します。これらのリソースに関して、他の目的で複製することや掲載することは禁止されています。TI や第三者の知的財産権のライセンスが付与されている訳ではありません。お客様は、これらのリソースを自身で使用した結果発生するあらゆる申し立て、損害、費用、損失、責任について、TI およびその代理人を完全に補償するものとし、TI は一切の責任を拒否します。

TI の製品は、TI の販売条件、または ti.com やかかる TI 製品の関連資料などのいずれかを通じて提供する適用可能な条項の下で提供されています。TI がこれらのリソースを提供することは、適用される TI の保証または他の保証の放棄の拡大や変更を意味するものではありません。

お客様がいかなる追加条項または代替条項を提案した場合でも、TI はそれらに異議を唱え、拒否します。

郵送先住所：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265