CC3120 SimpleLink™ Wi-Fi® Internet-on-a chip™ Solution SDK Getting Started Guide

The CC3120 device is part of the SimpleLink™ microcontroller (MCU) platform, which consists of Wi-Fi®, Bluetooth® low energy, Sub-1 GHz, and host MCUs. All share a common, easy-to-use development environment with a single core software development kit (SDK) and rich tool set. A one-time integration of the SimpleLink platform lets you add any combination of devices from the portfolio into your design. The ultimate goal of the SimpleLink platform is to achieve 100 percent code reuse when your design requirements change. For more information, visit www.ti.com/simplelink.

For more information, visit www.ti.com/simplelink.

This guide is intended to help users in the initial setup and demonstration of the different demos in the CC3120 SDK. The guide lists the software and hardware components required to get started, and explains how to install the supported integrated development environment (IDE), SimpleLink CC3120 SDK, and the various other tools required.

Contents

1 Introduction 2
2 Prerequisites 2
2.1 Hardware 2
2.2 Software Installation 2
3 Board Configuration 3
3.1 Board Setup For Programming 3
3.2 Board Setup With An External MCU 7
4 Programming the CC3120 BoosterPack™ 9
5 Getting Started With the MSP-EXP432P401R LaunchPad™ 13
5.1 Connecting the CC3120BOOST or BOOSTXL-CC3120MOD Board With the MSP-EXP432P401R Board 13
5.2 Terminal Setting 13
5.3 SDK Installation 13
5.4 Setting Up Code Composer Studio™ 14
5.5 Setting Up IAR Embedded Workbench® for ARM 20
6 Getting Started With SimpleLink™ Studio 28
6.1 Visual Studio Installation 28
6.2 Importing network_terminal Example in Visual Studio 30
6.3 Compiling and Debugging the Project in Visual Studio 30

Trademarks

SimpleLink, LaunchPad, Code Composer Studio, BoosterPack, MSP432 are trademarks of Texas Instruments.
Bluetooth is a registered trademark of Bluetooth SIG Inc.
IAR Embedded Workbench is a registered trademark of IAR Systems AB.
Microsoft, Windows are registered trademarks of Microsoft Corporation.
Wi-Fi is a registered trademark of Wi-Fi Alliance.
All other trademarks are the property of their respective owners.

Copyright © 2017–2018, Texas Instruments Incorporated

1 Introduction

The goal of this tutorial is to show users how to set up the development environment, and execute a SimpleLink Wi-Fi CC3120 project, on both the MSP-EXP432P401R LaunchPad™ and SimpleLink Studio, using several IDEs:

- MSP-EXP432P401R LaunchPad
  - Code Composer Studio™ (CCS)
  - IAR Embedded Workbench®
- SimpleLink Studio
  - Visual Studio 2015

2 Prerequisites

2.1 Hardware

The user is expected to have the following:

- SimpleLink CC3120 BoosterPack™ Evaluation Board (CC3120BOOST or BOOSTXL-CC3120MOD)
- SimpleLink CC31xx Emulation BoosterPack Debug Board (CC31XXEMUBOOST)
- MSP-EXP432P401R LaunchPad Development Kit (MSP-EXP432P401R)
- Micro-USB cable
- 802.11 b/g/n wireless router with Internet access
- PC running at least a Microsoft® Windows® 7 operating system with Internet access

2.2 Software Installation

The user is expected to have the following:

- SimpleLink MSP432™ SDK
- SimpleLink CC3120 SDK (also referred to as the MSP432 Wi-Fi plug-in)
- SimpleLink CC3120 ServicePack. It is part of the SDK under <SDK install dir>/ tools/cc32xx_tools/servicepack-cc3x20
- UniFlash v4.x for flashing the device
- CCS Version 7.0.0 or later
- IAR Embedded Workbench Version 7.80.1 or later, for ARM
- Visual Studio 2015, download the Visual Studio Community package.
- FreeRTOSv9.0.0 – the SDK does not include FreeRTOS sources. The user is expected to download the FreeRTOS sources.

This tutorial assumes the packages are installed at the default location. The paths for these installations may be different, depending on the version of the software you downloaded. Regardless, users can manually define these paths without affecting the overall experience.

If FTDI drivers are not installed on your Windows PC, install the drivers when prompted by the SDK installer.

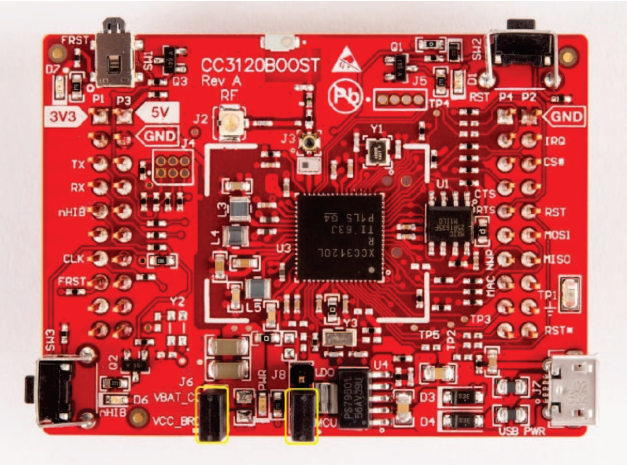
Copyright © 2017–2018, Texas Instruments Incorporated

3 Board Configuration

3.1 Board Setup For Programming

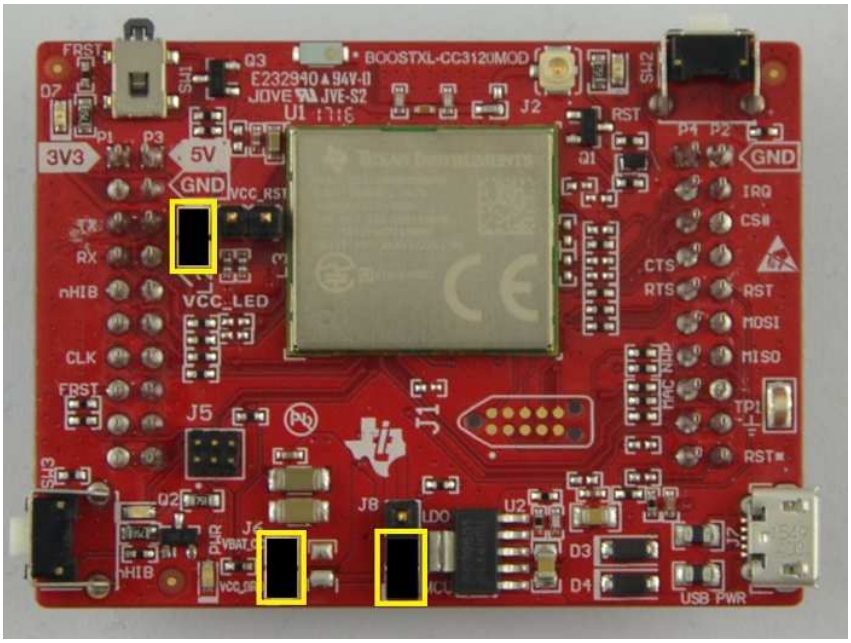
To program the CC3120BOOST or BOOSTXL-CC3120MOD board, the CC31XXEMUBOOST debug board is required. The CC31XXEMUBOOST board uses an FTDI chipset, which connects to the CC3120 through UART for programming purposes, and also connects to the reset line for automatic reset. The CC31XXEMUBOOST debug board is required for programming only.

Figure 1, Figure 2, and Figure 3 show how to properly configure the jumpers on the CC3120BOOST or BOOSTXL-CC3120MOD and CC31XXEMUBOOST boards.



Copyright © 2017, Texas Instruments Incorporated

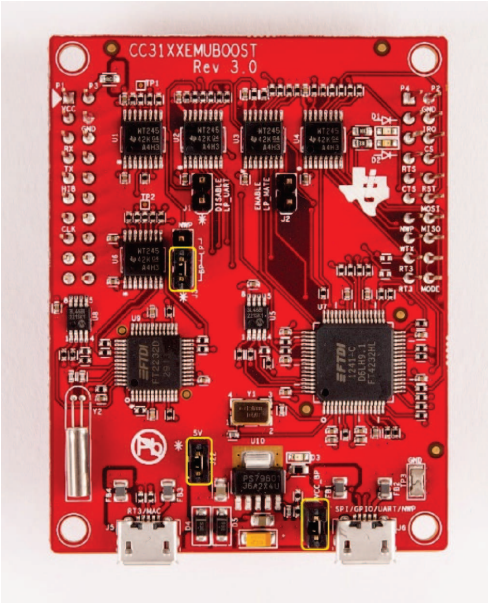
Figure 1. CC3120BOOST Board With Jumper Configuration



Copyright © 2017, Texas Instruments Incorporated

Figure 2. BOOSTXL-CC3120MOD Board With Jumper Configuration

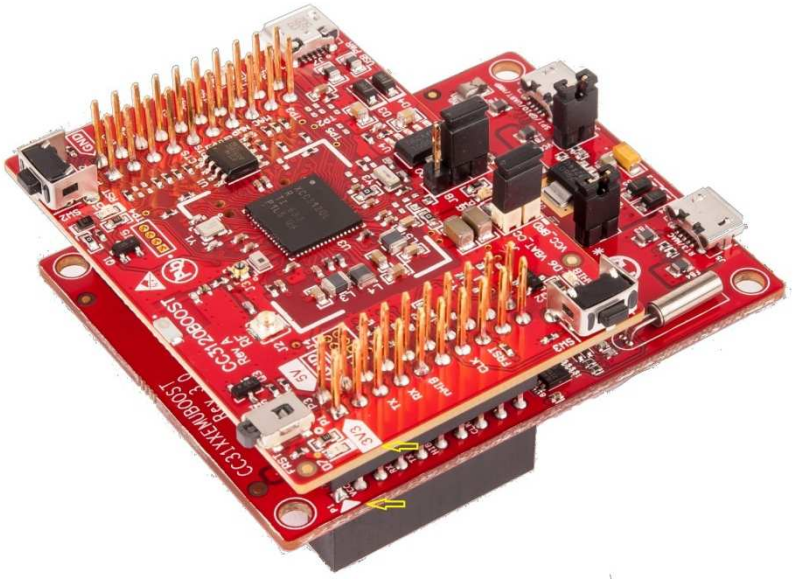
Copyright © 2017–2018, Texas Instruments Incorporated



Copyright © 2017, Texas Instruments Incorporated

Figure 3. CC31XXEMUBOOST Board With Jumper Configuration

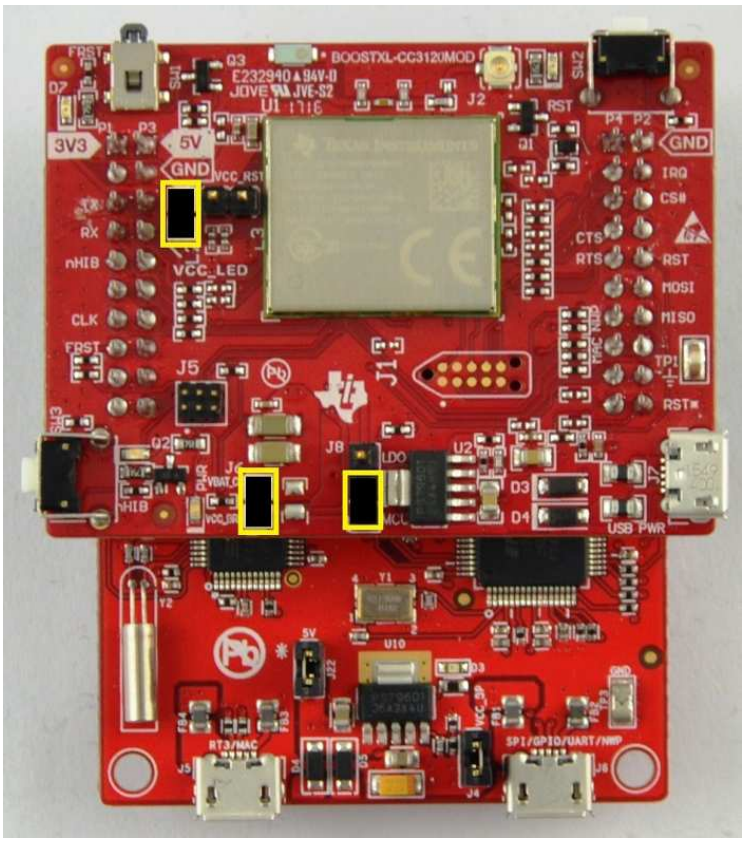
Figure 4 shows how to mount the CC3120BOOST board on top of the CC31XXEMUBOOST board. Figure 5 shows how to mount the BOOSTXL-CC3120MOD on top of the CC31XXEMUBOOST board. Ensure P1.1 of both boards are aligned with each other, as indicated by the yellow arrows on both boards. The LDO and MCU jumper (J8) configuration is used for choosing the power source. This guide uses the MCU configuration. To supply power directly to either of the BoosterPack boards, change the jumper to LDO, and connect USB PWR (J7) with a USB cable.



Copyright © 2017, Texas Instruments Incorporated

Figure 4. CC3120BOOST Board Mounted on Top of CC31XXEMUBOOST Board

Copyright © 2017–2018, Texas Instruments Incorporated

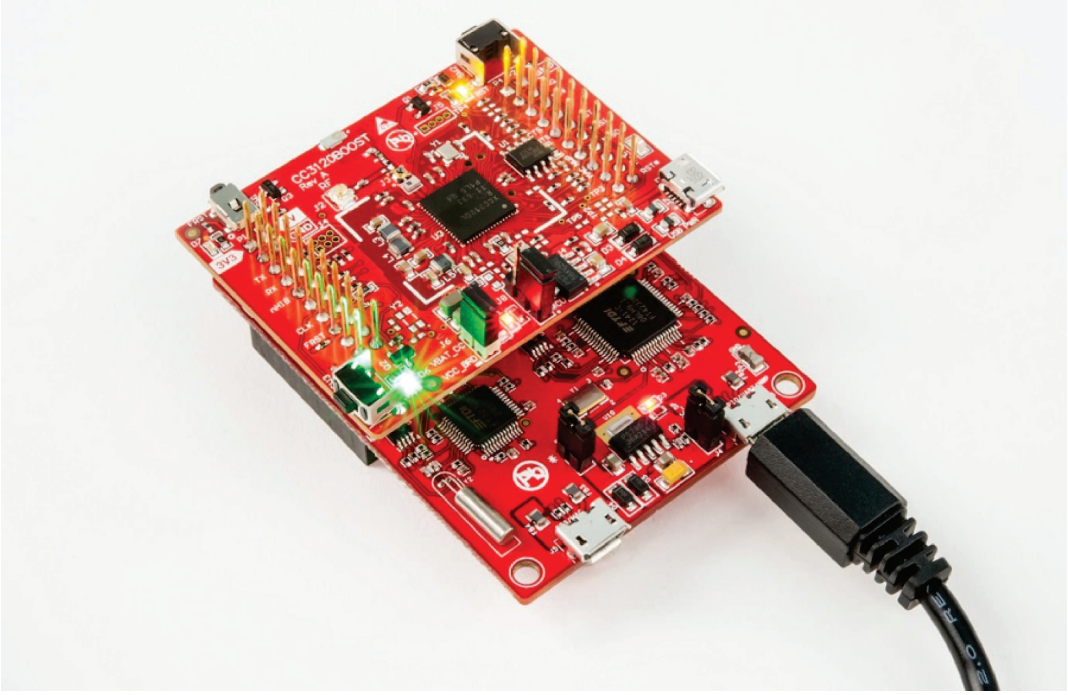


Copyright © 2017, Texas Instruments Incorporated

Figure 5. BOOSTXL-CC3120MOD Mounted on Top of CC31XXEMUBOOST

Copyright © 2017–2018, Texas Instruments Incorporated

To power the platforms, connect the J6 port of the CC31XXEMUBOOST board to the Windows PC using the micro USB cable (see Figure 6).



Copyright © 2017, Texas Instruments Incorporated

Figure 6. Powering the Platforms

If the drivers are properly installed as mentioned during the SDK installation, then the PC should recognize and list the connected device in the Device Manager (see Figure 7). The boards are now ready for programming.

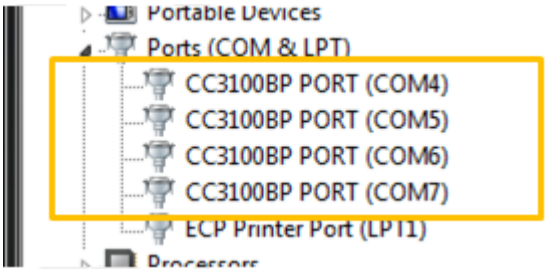


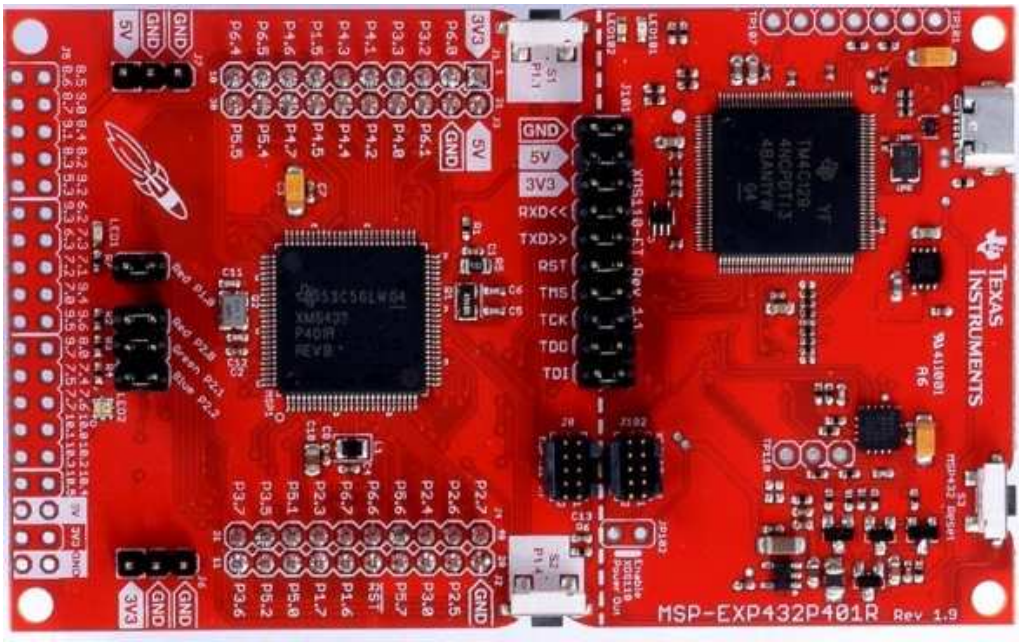
Figure 7. Device Manager Showing Ports

Copyright © 2017–2018, Texas Instruments Incorporated

3.2 Board Setup With An External MCU

The CC3120 BoosterPack evaluation board requires a host MCU to execute the user application. This guide uses the MSP-EXP432P401R LaunchPad as a reference.

Figure 8 shows how the MSP-EXP432P401R LaunchPad jumpers must be configured.

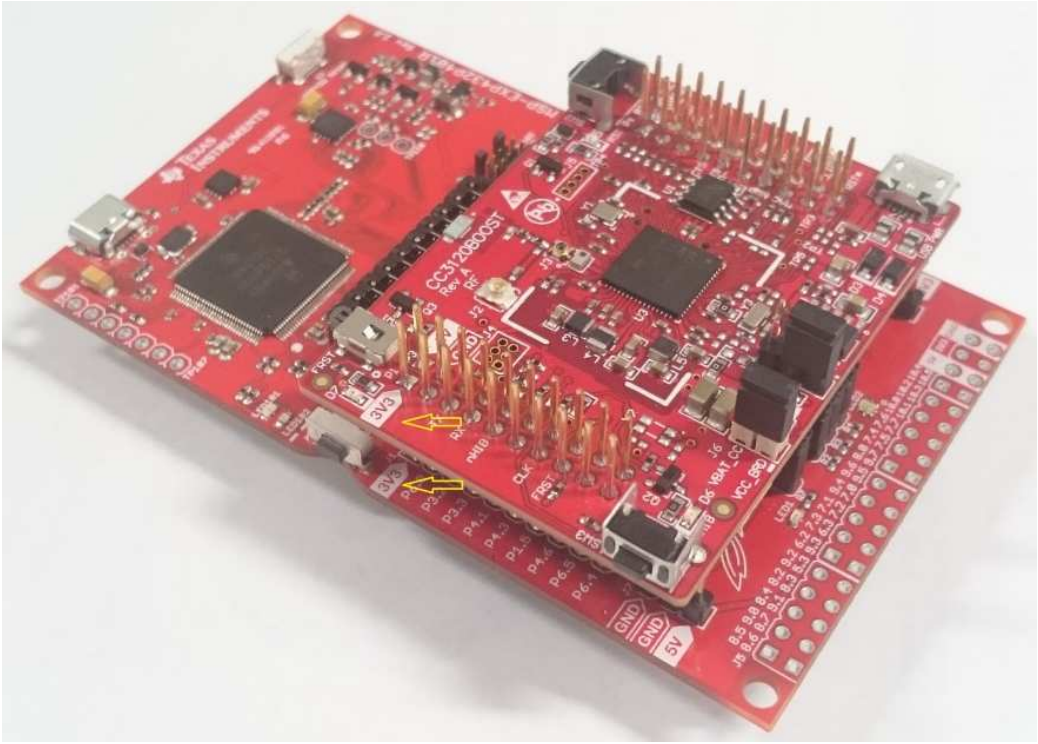


Copyright © 2017, Texas Instruments Incorporated

Figure 8. MSP-EXP432P401R With Jumper Configuration

Copyright © 2017–2018, Texas Instruments Incorporated

Figure 9 shows how to mount the CC3120BOOST board on top of the MSP-EXP432P401R board. Ensure P1.1 of both boards are aligned with each other, as indicated by the white arrows on both boards. The micro USB connector on the MSP-EXP432P401R board is used as the power source and also as the programming interface for the user application.



Copyright © 2017, Texas Instruments Incorporated

Figure 9. CC3120BOOST Board Mounted on Top of the MSP-EXP432P401R Board

Copyright © 2017–2018, Texas Instruments Incorporated

4 Programming the CC3120 BoosterPack™

Regardless of the host MCU (MSP432 or Windows PC as SimpleLink Studio mode), the CC3120BOOST or BOOSTXL-CC3120MOD board must be programmed. As a minimum, the Service Pack is required. Other system and configuration files are also programmed by default.

To program a CC3120 BoosterPack, the CC31XXEMUBOOST board is required. Section 3.1 describes the setup.

To program a CC3120 BoosterPack, install the UniFlash utility and perform the following steps:

1. Connect the CC31XXEMUBOOST board to the PC using the micro USB cable.
2. Launch UniFlash, select *CC3120/CC3220* device, and then click the *Start Image Creator* button (see Figure 10).

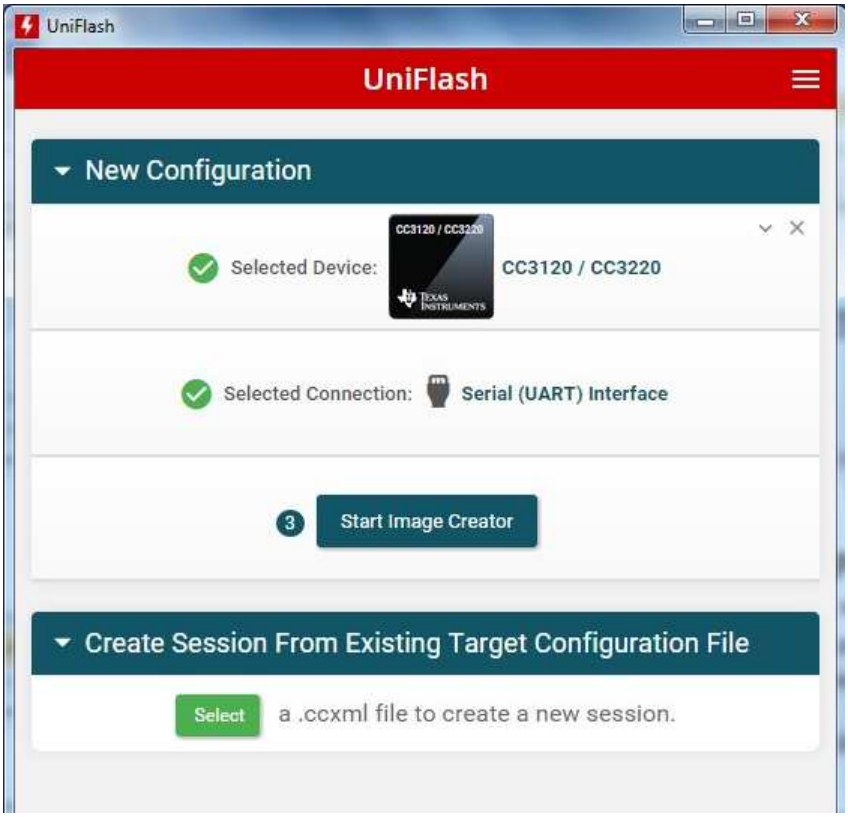


Figure 10. UniFlash Main Configuration Page

Copyright © 2017–2018, Texas Instruments Incorporated

3. Select the *New Project* button to enter the project information (see Figure 11).
4. Enter a project name.
5. Select *CC3120* as the Device Type.
6. Select *Develop* as the Device Mode (choose *Production* mode in final production).
7. Click the *Create Project* button.



Figure 11. UniFlash Create Project

Copyright © 2017–2018, Texas Instruments Incorporated

8. Click the *Connect* button on the right side of the screen to connect to the board. If the connection is successful, the device information appears on the right side of the web page (see Figure 12).

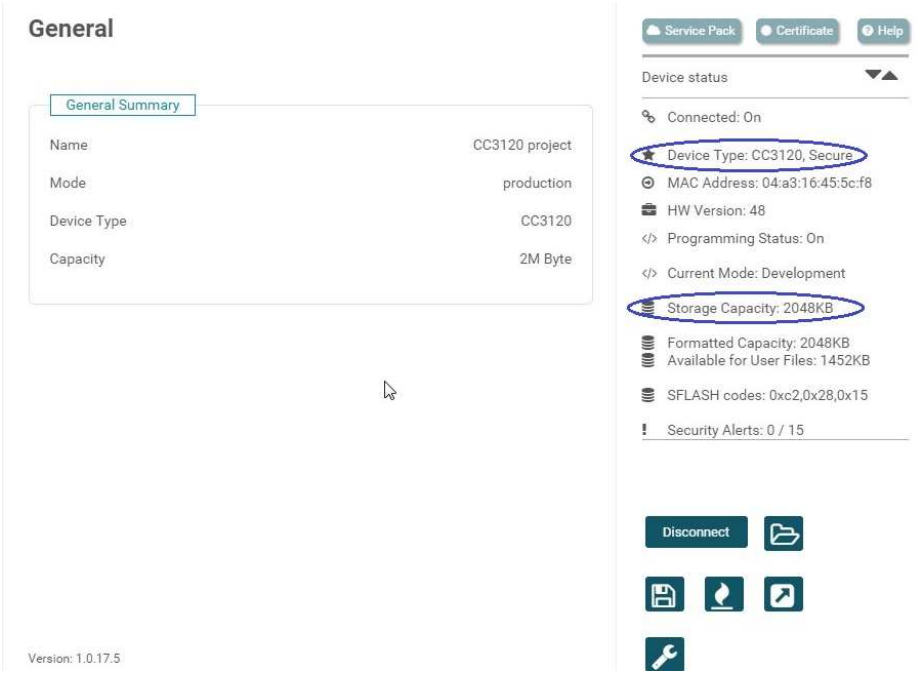


Figure 12. UniFlash Connect to CC3120

9. On the left side of the page, click the *Service Pack* tab to bring up the Service Pack File Name selection dialog.
10. Click the *Browse* button and navigate to the Service Pack directory to select the binary (see Figure 13).

    The Service Pack is in the SDK under the <SDK install Dir>/tools/cc31xx_tools/servicepack-cc3x20 directory.

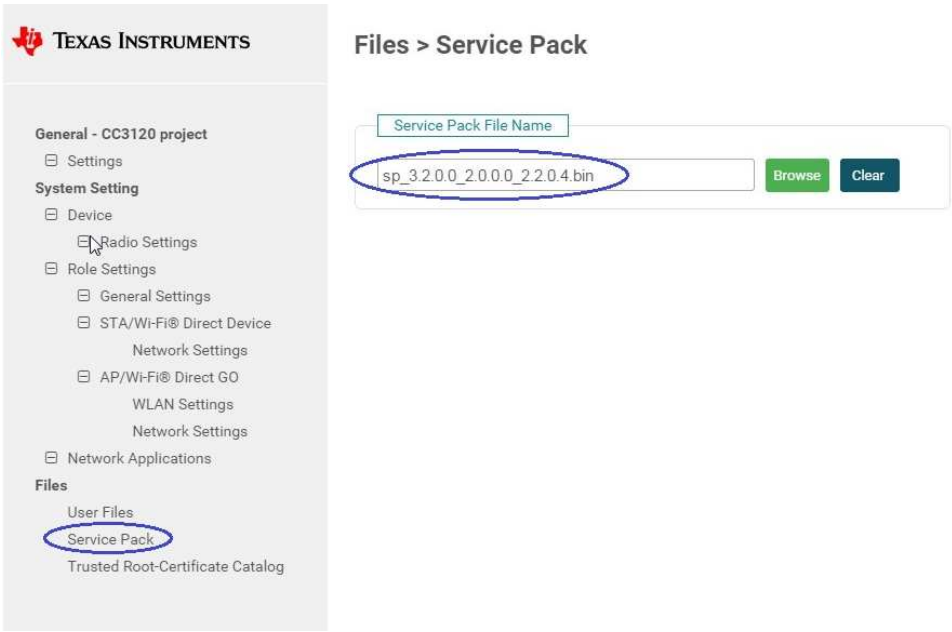


Figure 13. UniFlash Service Pack

Copyright © 2017–2018, Texas Instruments Incorporated

11. Now click the *Generate Image* button  on the right side of the web page to enter the programming page.
12. Click the center *Program Image (Create & Program)* button to create the image and start flashing it (see Figure 14). This operation may take a while to complete. Once the process is finished, a confirmation dialog appears.

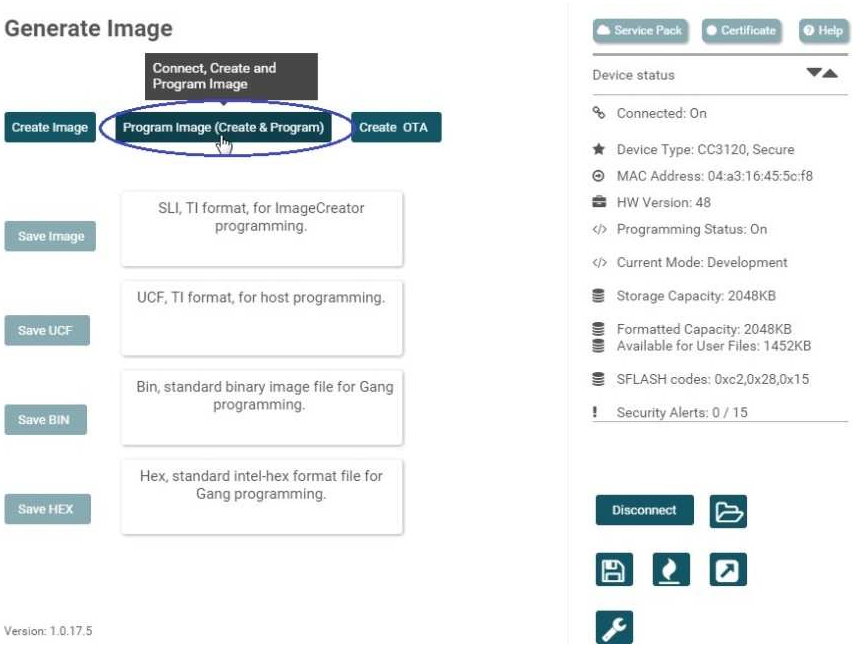


Figure 14. UniFlash Program Image

13. Once programming is complete, click the *Disconnect* button on the right side.

Copyright © 2017–2018, Texas Instruments Incorporated

5 Getting Started With the MSP-EXP432P401R LaunchPad™

5.1 Connecting the CC3120BOOST or BOOSTXL-CC3120MOD Board With the MSP-EXP432P401R Board

Section 3.2 fully describes building this setup. Before connecting these two boards, the CC3120BOOST or BOOSTXL-CC3120MOD board must be programmed using UniFlash. Section 3.1 describes the programming setup.

5.2 Terminal Setting

The *network terminal* example demonstrates an interactive user interface in the form of a command line. To let the user type commands, get responses, and debug messages, a terminal utility is essential.

1. Connect the MSP-EXP432P401R LaunchPad to the PC with a micro USB cable. Windows should search for the driver automatically. To verify that the driver is properly installed, open the Device Manager on the PC, and then navigate to *Ports*. Two enumerated XDS110 ports, as shown in Figure 15, indicate if the driver was installed properly during CCS installation.

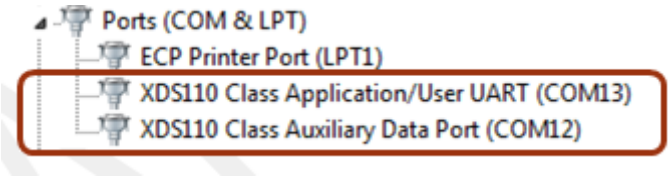


Figure 15. Enumerated XDS Ports for MSP-EXP432P401R LaunchPad™

2. Launch a terminal emulation such as TeraTerm or PuTTY and configure it to the settings in Table 1.

Table 1. Serial Connection Settings

Parameter	Value
Port number	(The XDS110 application UART port number)
Baud rate	115,200
Data bits	8
Stop bits	1
Parity	None
Flow control	(Ignored)

The following sections describe setting up two different IDEs to compile and execute your first project: CCS and IAR.

5.3 SDK Installation

The SDK installation procedure involves two components: SimpleLink MSP432 SDK as the base SDK and SimpleLink CC3120 SDK as a Wi-Fi plug-in.

Install the base SDK first and then the plug-in in the default location pointed to by the installer.

Copyright © 2017–2018, Texas Instruments Incorporated

5.4 Setting Up Code Composer Studio™

5.4.1 CCS™ Installation

Download CCS version 7.0.0 or later. During installation, TI recommends using the default directory of c:\ti\. When you are prompted to select the processor support, select the following options, as shown in Figure 16.

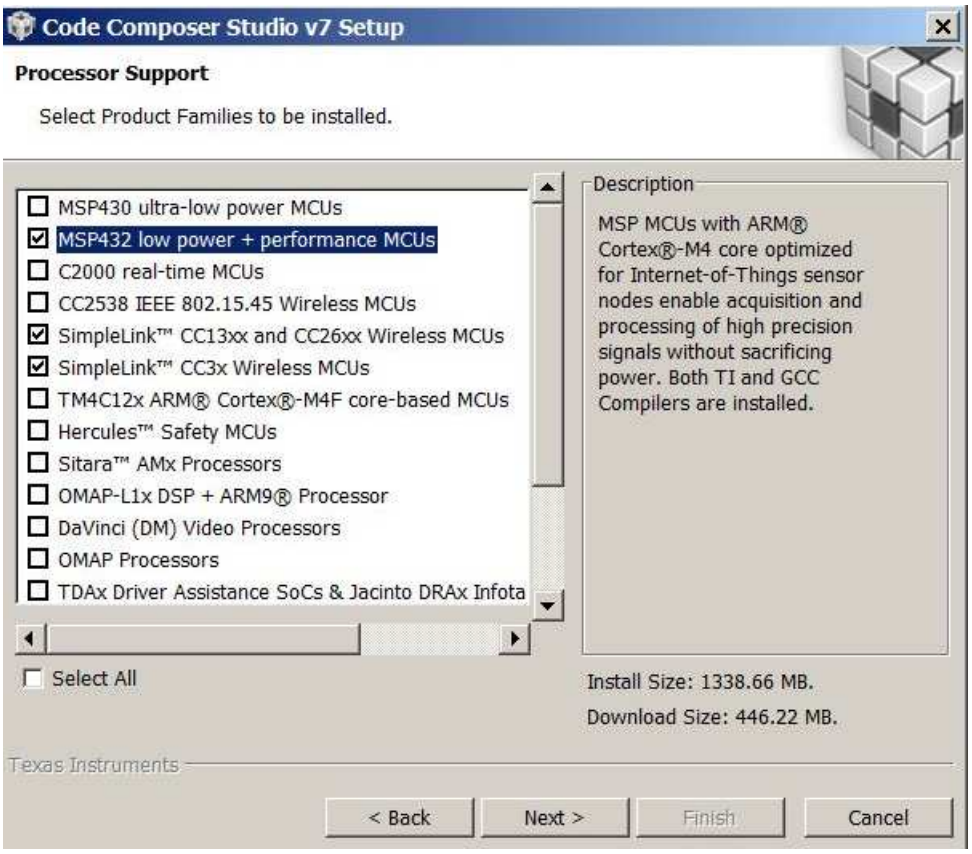


Figure 16. Code Composer Studio™ Installation Packages Selection

Copyright © 2017–2018, Texas Instruments Incorporated

To use a debug probe, select TI XDS Debug Probe Support and any other options you would like, as shown in Figure 17.

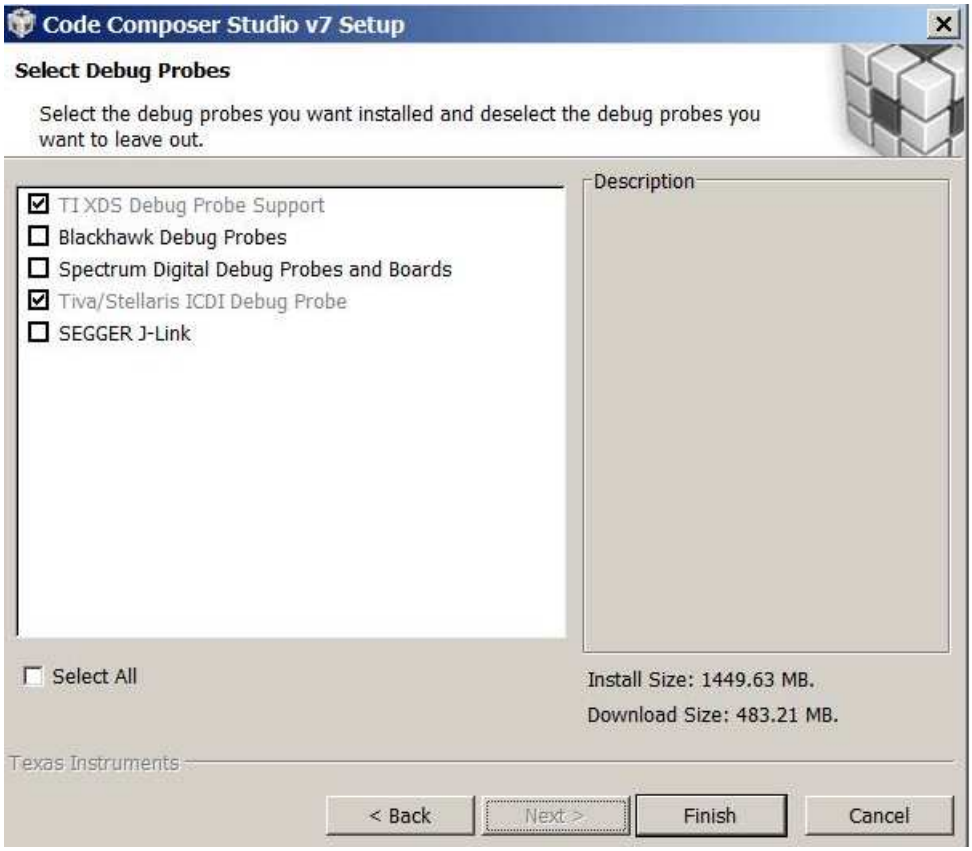


Figure 17. Code Composer Studio™ Installation Debug Probes

NOTE: The installation may take a while to complete, depending on the number of selections and the speed of the network, because CCS needs to download these files. Once the installation is finished, start CCS and users will be prompted to choose a workspace folder, which is the folder where the project files reside.

Copyright © 2017–2018, Texas Instruments Incorporated

5.4.2 Post Code Composer Studio™ Installation Setup

Support for freeRTOSv9 must be added. To add the support, do the following:

1. Download the FreeRTOS official version 9 and install into file C:
2. Run CCS and choose Window → Preferences → Code Composer Studio → Build → Variables → Add (see Figure 18).

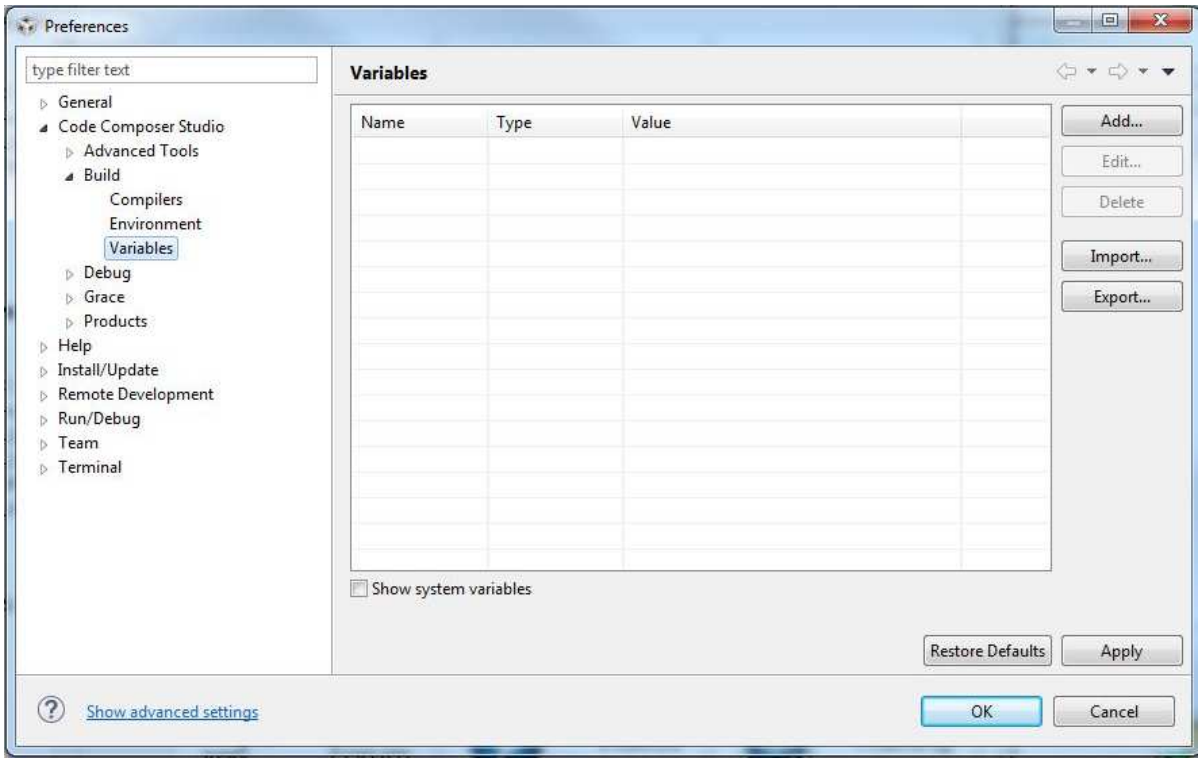


Figure 18. Code Composer Studio™ Installation – Add Variable

3. Fill the Variable name field with *FREERTOS_INSTALL_DIR* (see Figure 19).
4. Change the Type field to directory.
5. Click the *Browse* button to populate the Value field with the <FREERTOS_INSTALL_PATH> and then click *OK*.

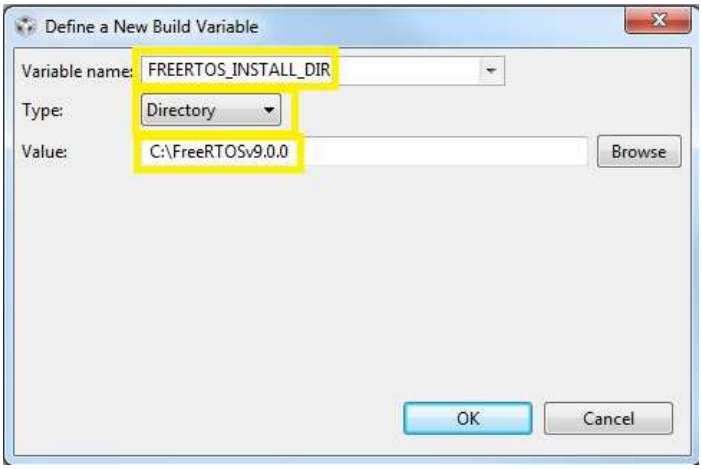


Figure 19. Code Composer Studio™ Installation – FreeRTOS Support

Copyright © 2017–2018, Texas Instruments Incorporated

Finally, installed products must be verified.

1. Choose Window → Preferences → Code Composer Studio → Products (see Figure 20).

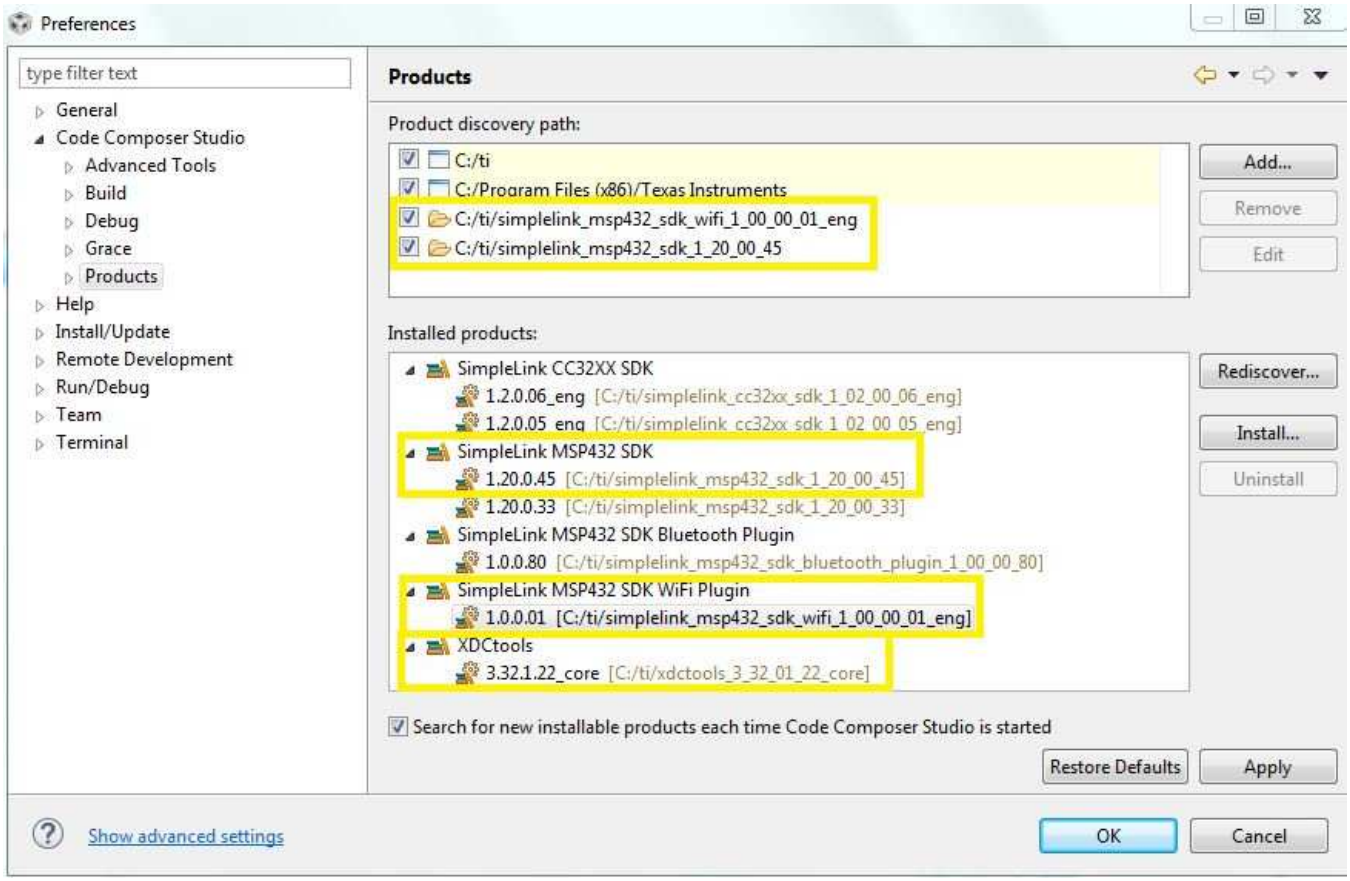


Figure 20. Code Composer Studio™ Installation Products

2. Check that the paths in the Product Discovery Path include the following:
   - SimpleLink MSP432 SDK WiFi plug-in
   - SimpleLink MSP432 SDK
   - XDCtools

Copyright © 2017–2018, Texas Instruments Incorporated

3. If the paths are not in the Product Discovery Path, add them and then click the Rediscover button.

   An Install Discovered Products window may pop up (see Figure 21). If the window appears, it means that new products have been discovered. If the products in Step 2 are in the list, choose them and select install.

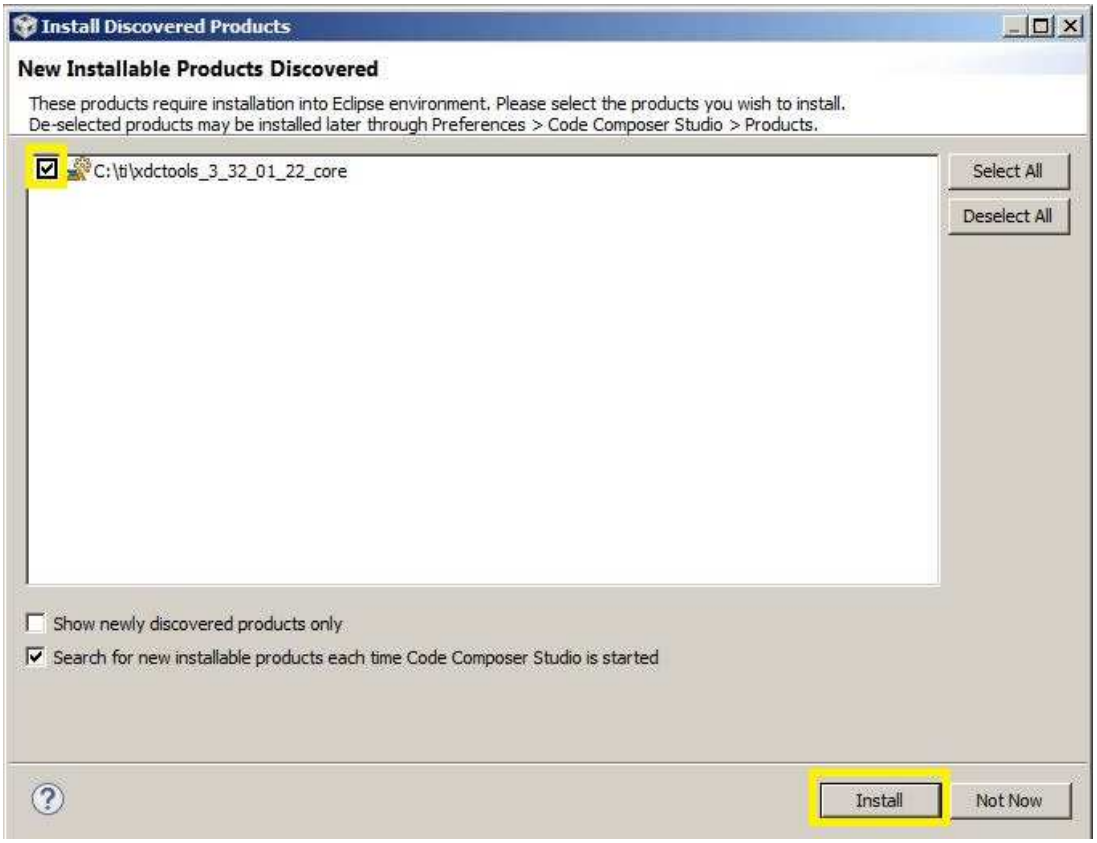


Figure 21. Code Composer Studio™ Installation – New Products Discovered

5.4.3 Importing network_terminal Example in Code Composer Studio™

1. Open CCS.
2. Click on the *Project* button, go to import CCS projects, and browse to the location of the plug-in installation (<Plugin_INSTALL_DIR>/examples). CCS lists all the CCS projects inside the specified directory and its subdirectories.

   The SDK provides TI-RTOS and FreeRTOS-based examples. TI-RTOS and FreeRTOS examples have the word *tirtos* or *freertos* in the project name. Also, CCS lists both CCS- and GCC-based examples.

   Importing TI-RTOS and FreeRTOS examples also imports kernel projects in the same workspace. The kernel project is a dependent project and is automatically built when the example is built.

   For more details and information on how to change configurations, see the <Plugin_INSTALL_DIR>/docs/simplelink_mcu_sdk/Users_Guide.html document.

Copyright © 2017–2018, Texas Instruments Incorporated

3. Select the network_terminal-based example and click *Finish* (see Figure 22).

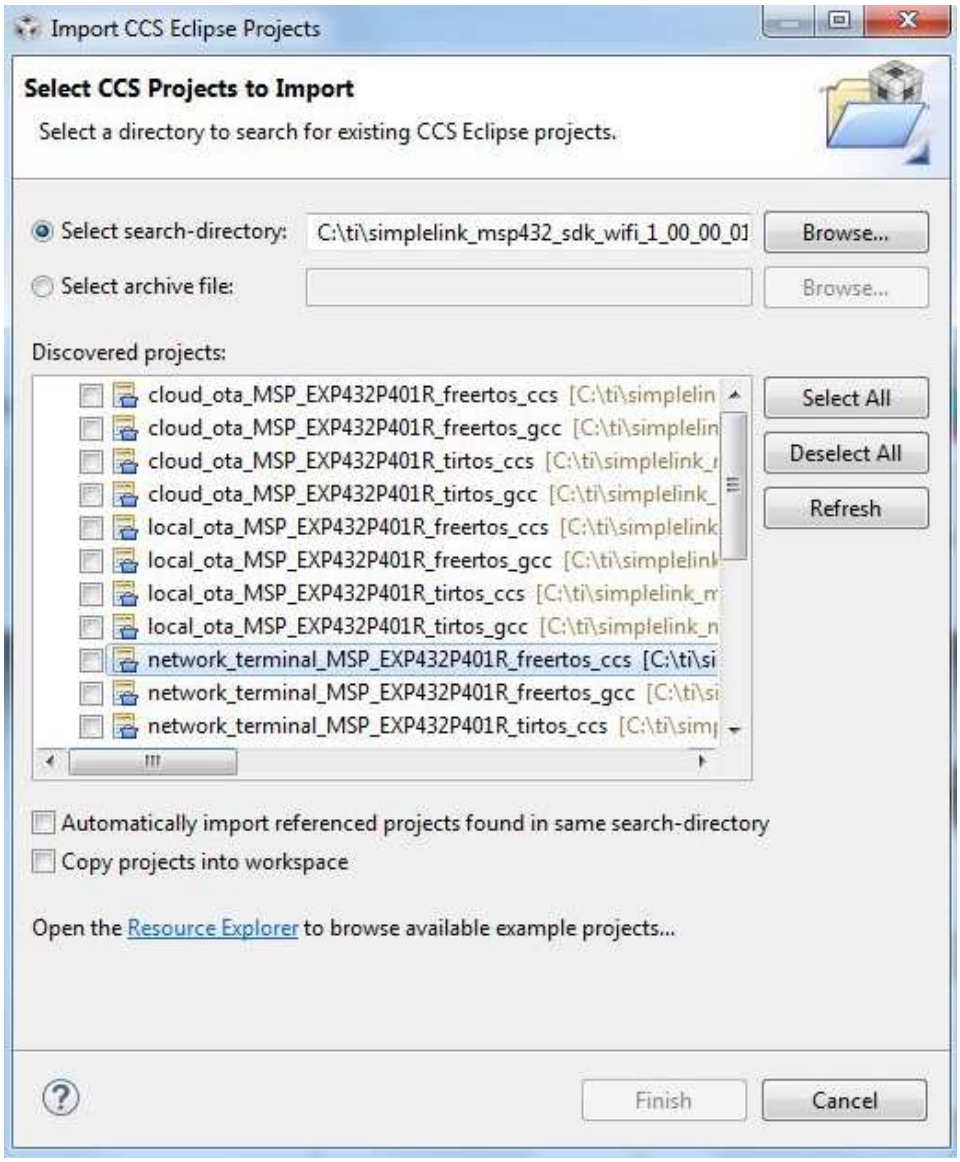


Figure 22. Code Composer Studio™ Import network_terminal Example

5.4.4 Compiling and Debugging the Project in Code Composer Studio™

Right-click on the network_terminal project from the Project Explorer window, and select *rebuild project*. CCS compiles the chosen kernel first and then the netwok_terminal example. To debug the example, follow these steps:

1. Open a terminal window and set it as described in Section 5.2.
2. Right-click on the project from the Project Explorer window, and select Debug As → Code Composer Debug Session, to start the debug process. This process downloads the code to the device and begins debugging. If this is your first time debugging the code, the process may take longer to complete.
3. Click the green arrow on the top of the screen to start executing your code.

Copyright © 2017–2018, Texas Instruments Incorporated

4. Go back to the terminal window. You can see the message being printed as the application runs on the MSP432 device (see Figure 23).

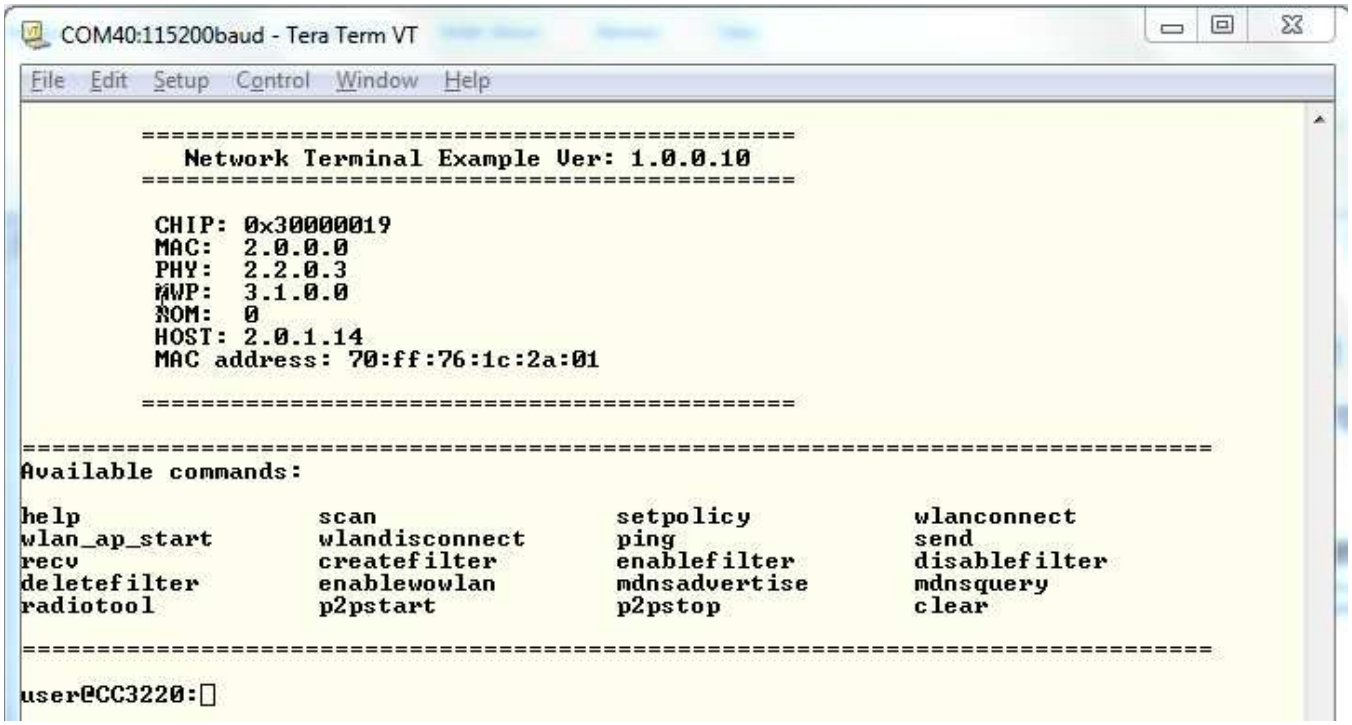


Figure 23. network_terminal Shell Command

5.5 Setting Up IAR Embedded Workbench® for ARM

5.5.1 IAR Installation

Download and install IAR Embedded Workbench for ARM Version 7.80.1 or later.

Ensure to select TI XDS as one of the debug probe drivers. Users may also select other drivers. The driver is installed toward the end of the installation of IAR.

During the installation process, users are prompted to install multiple additional software and drivers. Follow the instructions on the screen to finish the installation.

Installation may take a while to complete, depending on the number of selections and network speed, because IAR must download these files. When installation is complete, start IAR and enter your license information.

5.5.2 Post IAR Installation Setup

Once the IAR is installed and before starting to work with any of the projects in the SDK, you need to load a set of environment variables that will be set for your current workspace.

1. Browse to Tools → Configure Custom Argument Variables.
2. Go to the Global tab and click *Import*.
3. Choose the argvars which are stored in the .custom_argvars file in the <SDK install dir>\tools\iar directory of the SDK.
4. Edit the argvars to match your settings.
5. Restart IAR.

Copyright © 2017–2018, Texas Instruments Incorporated

5.5.3 Importing the network_terminal Example in IAR

Before importing any example to IAR, building the kernel is required.

First, installation paths must be updated. Edit imports.mak on the top-level directory of the MSP432 SDK and adjust the locations of the compiler tools as well as the FreeRTOS and XDCtools installation paths (see Figure 24). XDCTools installation is included in the SimpleLink MSP432 Wi-Fi plug-in.

```
XDC_INSTALL_DIR      ?= c:/ti/xdctools_3_32_01_22_core

FREERTOS_INSTALL_DIR ?= c:/FreeRTOSv9.0.0

CCS_ARMCOMPILER      ?= c:/ti/ccsv7/tools/compiler/ti-cgt-arm_16.9.0.LTS
GCC_ARMCOMPILER      ?= c:/ti/ccsv7/tools/compiler/gcc-arm-none-eabi-4_9-2015q3
IAR_ARMCOMPILER      ?= C:/Program Files (x86)/IAR Systems/Embedded Workbench 7.5/arm
```

Figure 24. IAR Update Variables

To build the kernel, follow this procedure:

1. To work with FreeRTOS, copy and replace FreeRTOSConfig.h from the Wi-Fi plug-in <SDK install dir\kernel\freertos\builds\MSP_EXP432P401R\release\FreeRTOSConfig.h> into the <MSP432 SDK install dir\kernel\freertos\builds\MSP_EXP432P401R\release\FreeRTOSConfig.h> (TI recommends saving the original FreeRTOSConfig.h file).
2. To build all the TI-RTOS and FreeRTOS config projects, open the command shell at <MSP432 SDK install dir\kernel and execute c:/ti/xdctools_3_32_01_22_core/gmake.exe.

To import the network_terminal example into IAR, follow this procedure:

1. Create a new empty project by browsing to Project → Create New Project, choose the ARM toolchain and an empty project, then click *OK* (see Figure 25).

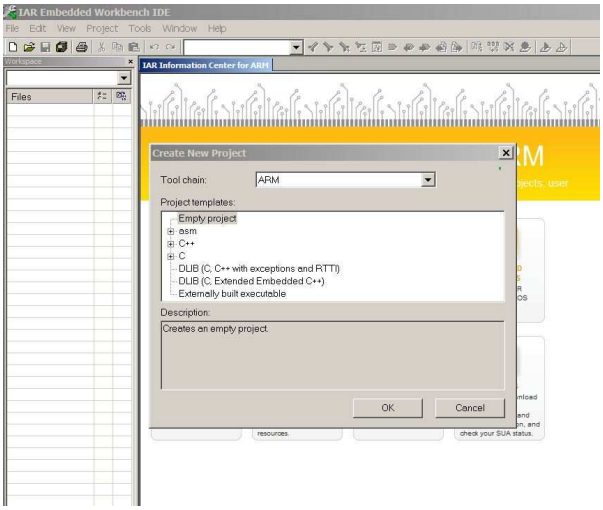


Figure 25. IAR Create New Project

2. Browse to the location where you want to save this project, and type a filename for the project file (*.ewp). Click *Save*.

Copyright © 2017–2018, Texas Instruments Incorporated

3. Choose Help → IAR Information Center for ARM, choose INTEGRATED SOLUTIONS, and then scroll down and choose Texas Instruments – Example projects (see Figure 26).



Figure 26. IAR TI Example Project

4. Click on the *example applications* link (see Figure 27). This link refers to the EXAMPLE_ROOT link in argvars.



Figure 27. IAR Example Application Link

Copyright © 2017–2018, Texas Instruments Incorporated

5. Click on an example according to the type of device and rtos/nortos flavor (see Figure 28). All project files should be imported to the workspace.



Figure 28. IAR Choose network_terminal Example

6. Save the workspace.
7. Browse to the location where you want to save this workspace, and type a filename for the workspace file (*.eww). Click *Save*.
8. Make your changes and rebuild the project.
9. To create a binary image in addition to the *.out file (in case the MCU image is required as part of UniFlash image), a post build step is also required. Right-click the project and choose Options.

Copyright © 2017–2018, Texas Instruments Incorporated

10. On the Output Converter tab, select the Generate additional output checkbox, and change the output format to binary (see Figure 29).

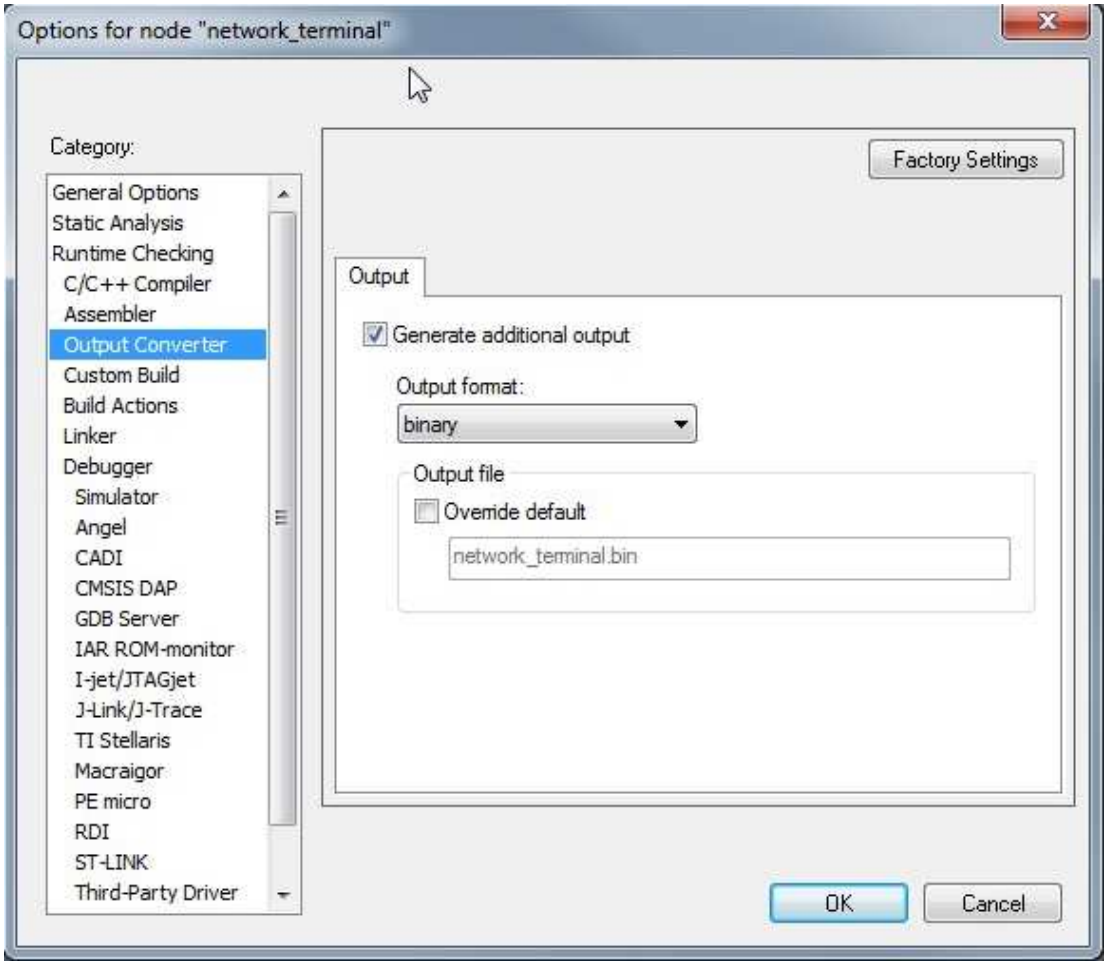


Figure 29. IR Generate Binary Output

11. Recompile and the binary should reside in Debug/Exe.

Copyright © 2017–2018, Texas Instruments Incorporated

5.5.4 Compiling and Debugging the Project in IAR

For the debug session, complete the following steps:

1. Select Project → options from the menu, and select the Debugger category. In the Setup tab, choose TI XDS as the driver and click *OK* (see Figure 30).

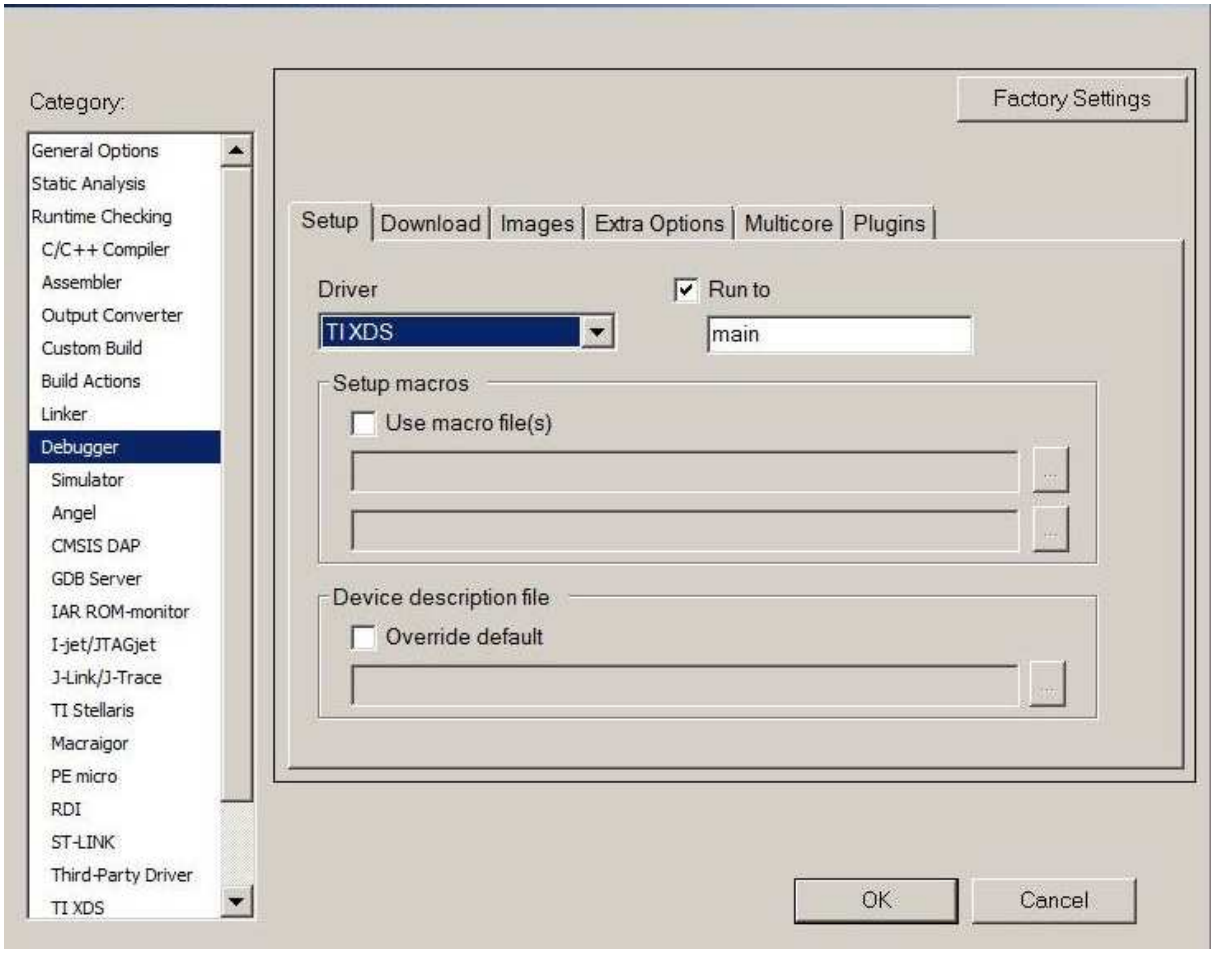


Figure 30. IAR TI XDS Debugger

Copyright © 2017–2018, Texas Instruments Incorporated

2. Go to the TI XDS category, choose TI XDS110 Emulator as the emulator and JTAG (4-pin) as the interface (see Figure 31).

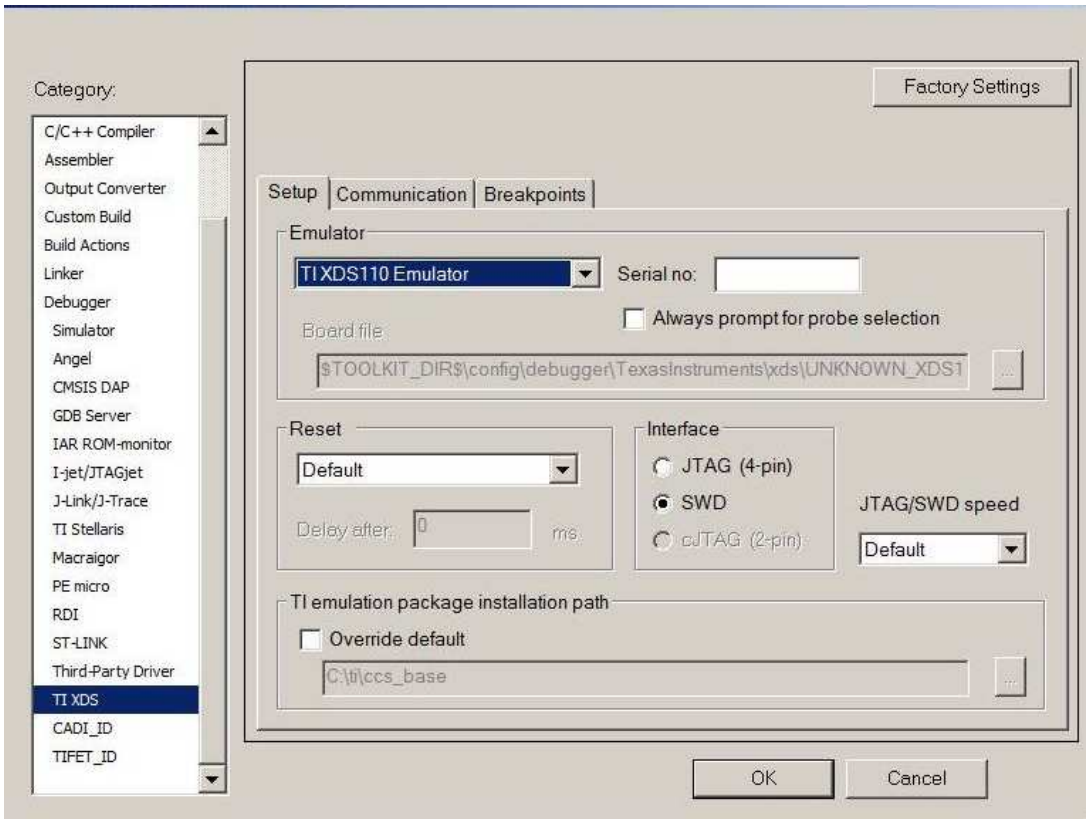


Figure 31. IAR XDS110 Debugger

Copyright © 2017–2018, Texas Instruments Incorporated

3. Check the Use flash loader(s) checkbox in the Debugger → Download tab (see Figure 32).

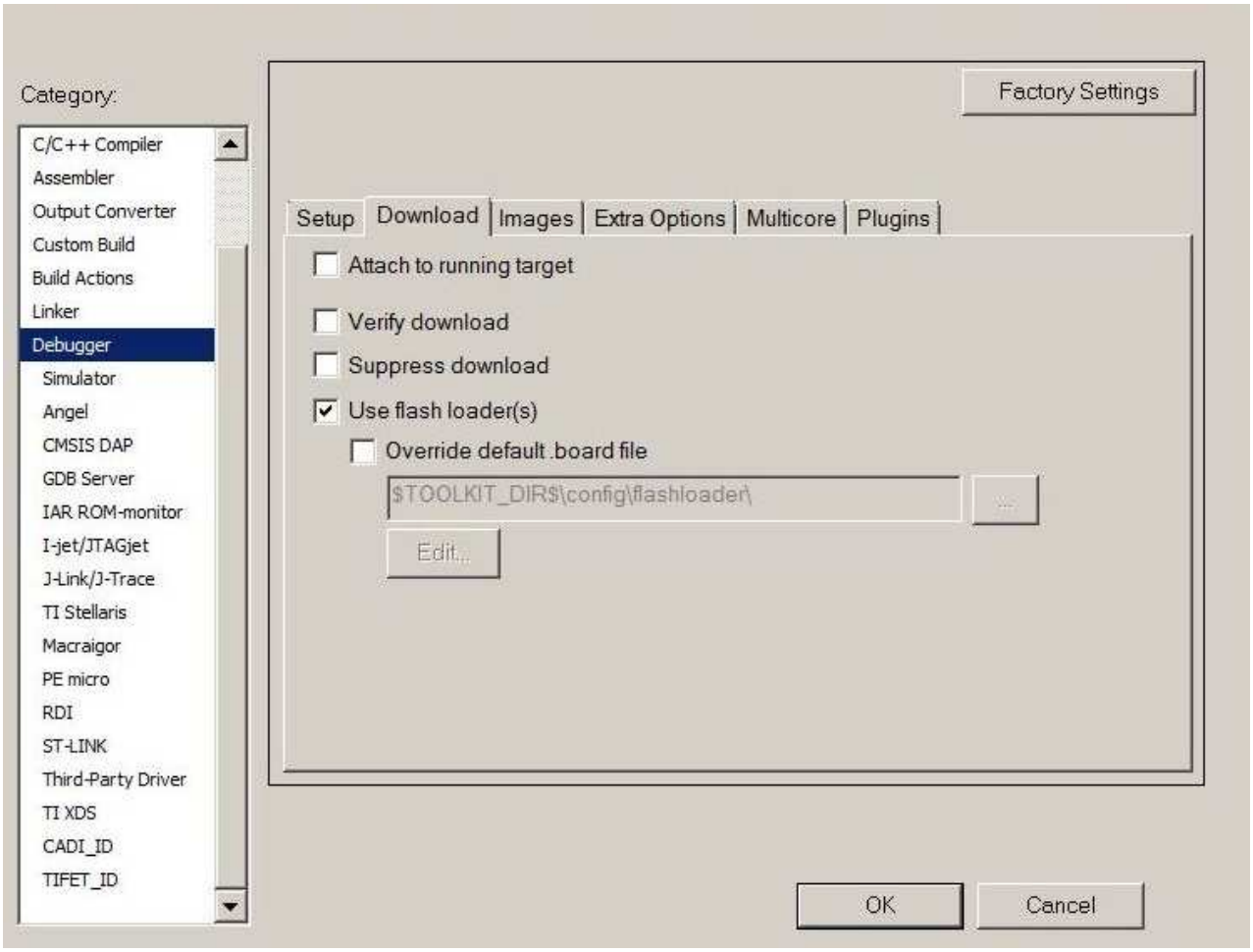


Figure 32. IAR Debugger Settings

4. Start debugging by clicking the green arrow on the top of the window to start the debugger.

Copyright © 2017–2018, Texas Instruments Incorporated

6 Getting Started With SimpleLink™ Studio

SimpleLink Studio is a Microsoft Windows-based software used to aid in the development of applications designed to work with the SimpleLink Wi-Fi CC31xx family of wireless chips.

Using SimpleLink Studio, application code can be written and executed in a desktop IDE, such as Visual Studio and Eclipse, and communicate directly with the CC31xx device through the USB cable without a MCU. This process allows for easy testing of code while it is under development, and then later ported to a MCU without needing modification.

TI provides support for SimpleLink with Visual Studio 2015. This software requires the CC31XXEMUBOOST debug board to operate.

6.1 Visual Studio Installation

Download and install Visual Studio. In this tutorial, we are using the Visual Studio 2015 Community Edition.

1. Run the installer to install Visual Studio. On the Customization page, select *Custom*. On the Select Features page, ensure the Visual C++ option under Programming Languages is checked, as shown in Figure 33.

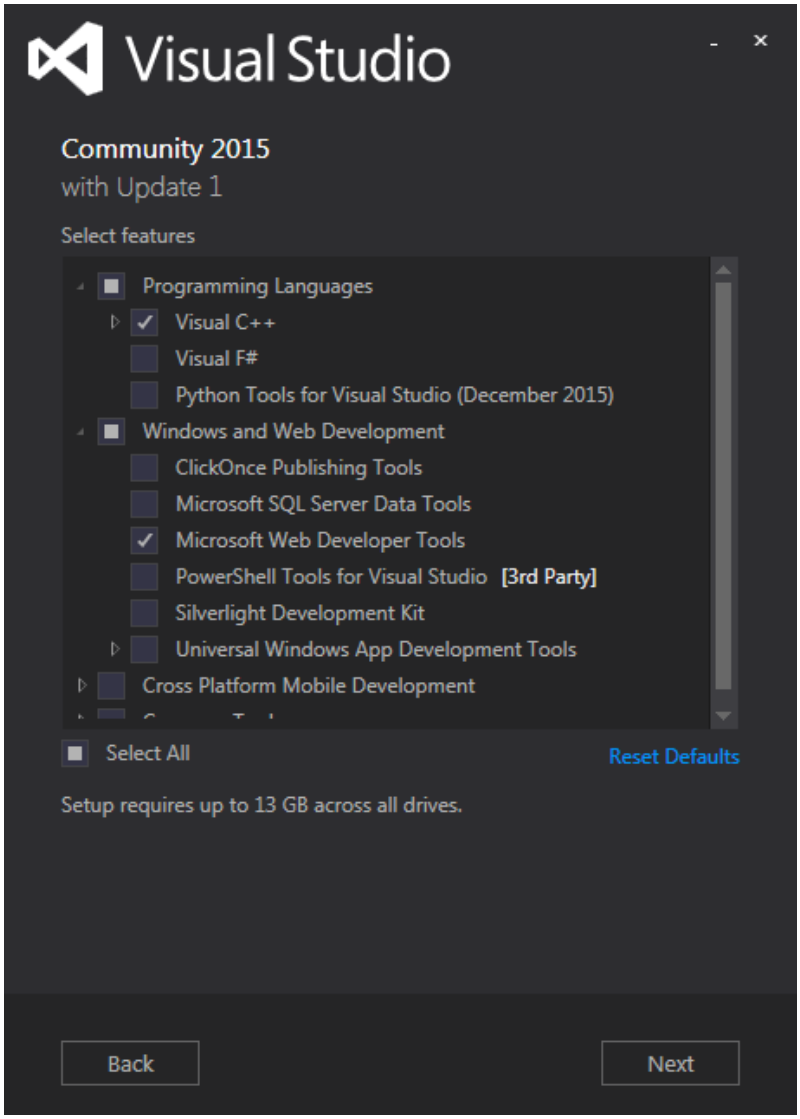


Figure 33. Visual Studio Installation

Copyright © 2017–2018, Texas Instruments Incorporated

NOTE: Installation may take a while to complete, depending on the number of selections and network speed, because Visual Studio must download these files.

2. Once installation is complete, restart Visual Studio by clicking the *Restart* button.
3. After restarting, a dialog box requiring an update to additional packages may appear, as shown in Figure 34. Click *Next* and then *Update*.

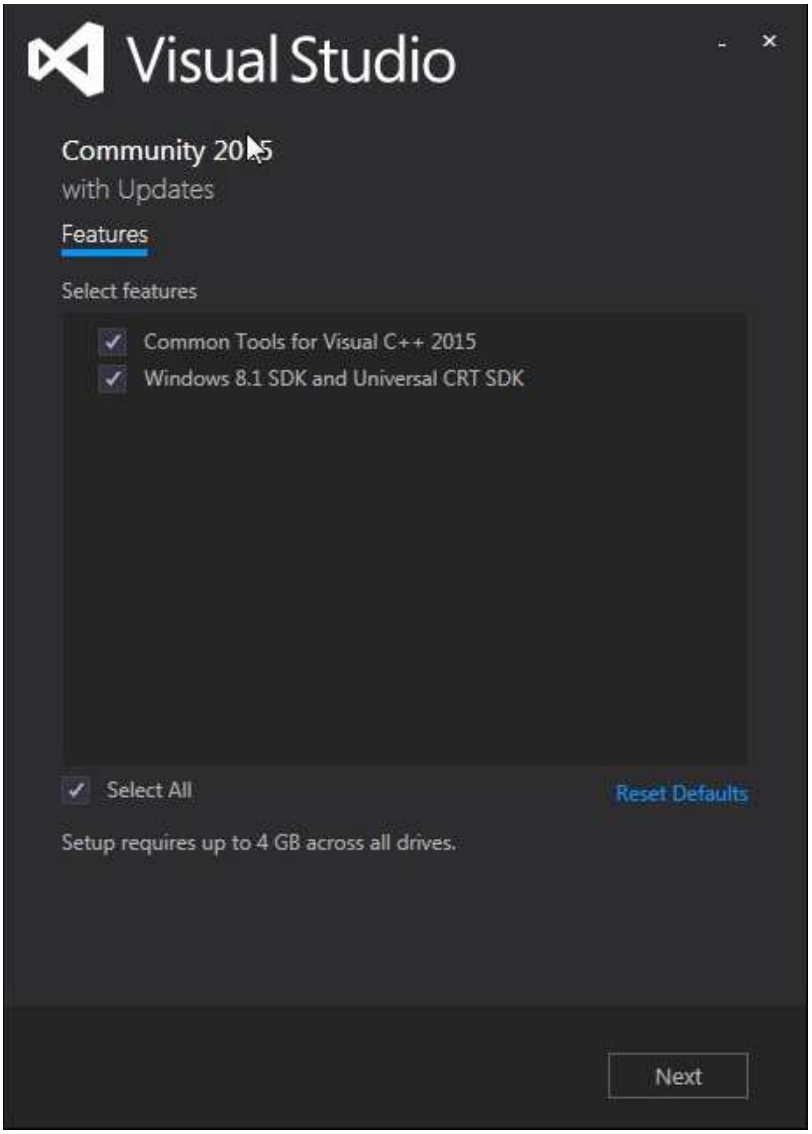


Figure 34. Visual Studio Updates

Copyright © 2017–2018, Texas Instruments Incorporated

6.2 Importing network_terminal Example in Visual Studio

The example can be loaded in two ways:

- Double-click the solution *.sln Launch Visual Studio, located under <SLS install dir>\examples\network_terminal\sl_studio>.
- Open Visual Studio, select File → Open → Project/Solution. Browse to <SLS install dir>\examples\network_terminal\sl_studio> and choose the *.sln.

The SDK provides the Network Terminal application, a Kernel project based on Windows and the Host Driver project.

6.3 Compiling and Debugging the Project in Visual Studio

To compile the project, select Rebuild solution, as shown in Figure 35.

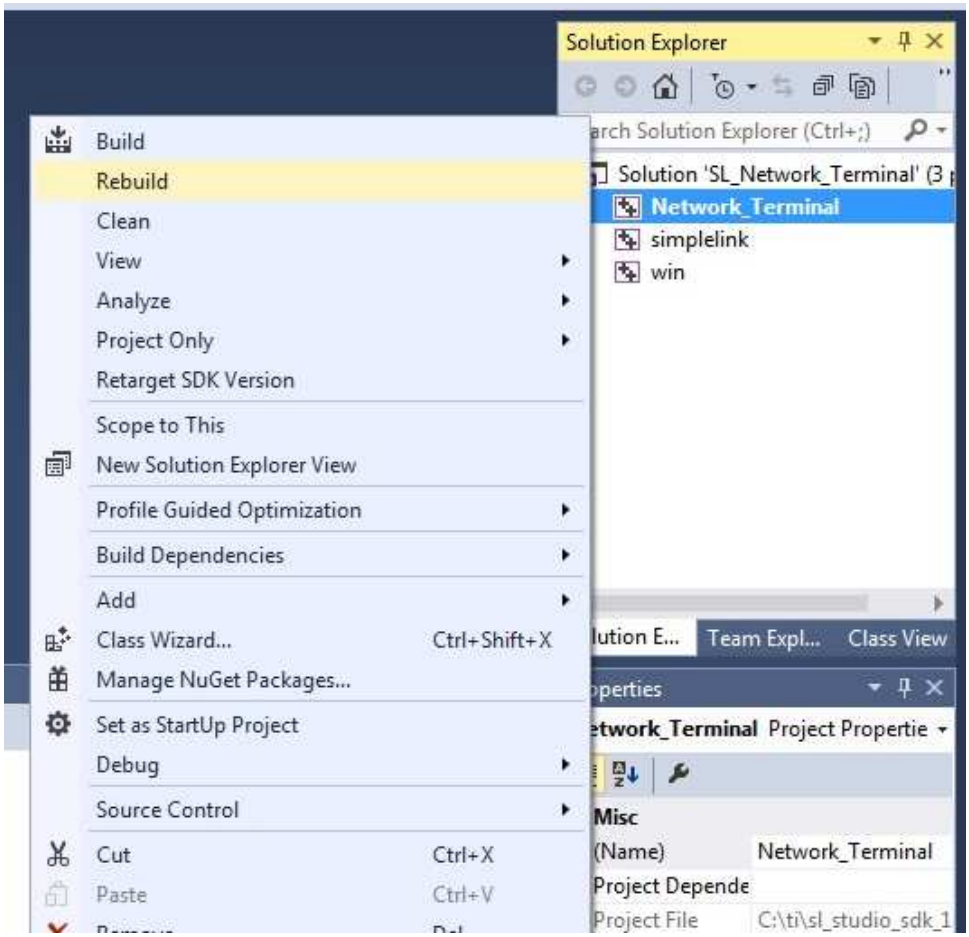


Figure 35. SimpleLink™ Studio Solution

Copyright © 2017–2018, Texas Instruments Incorporated

When the build is finished, there will be three succeeded projects (see Figure 36).

```
3>  network_terminal.c
3>  netapp_cmd.c
3>  main.c
3>  Generating Code...
3>  Network_Terminal.vcxproj -> C:\ti\sl_studio_sdk_1.00.00.03\examples\network_terminal\sl_studio\Debug\Network_Terminal.exe
========== Rebuild All: 3 succeeded, 0 failed, 0 skipped ==========
```

Figure 36. SimpleLink™ Studio Compiled Solution

To start debugging, run the debugger (see Figure 37).



Figure 37. SimpleLink™ Studio Debugger

A command line window opens and you can see the message printing as the application executes (see Figure 23).

Copyright © 2017–2018, Texas Instruments Incorporated

Revision History

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

Changes from February 8, 2017 to June 30, 2018 (from * Revision (February 2017) to A Revision)	Page
• Added BOOSTXL-CC3120MOD Board information	2
• Updated SimpleLink SDK links in Software Installation section.	2
• Added BOOSTXL-CC3120MOD Board With Jumper Configuration image to *Board Configuration*	3
• Added BOOSTXL-CC3120MOD Mounted on Top of CC31XXEMUBOOST image to *Board Configuration*	4
• Updated Getting Started With SimpleLink™ Studio section.	28

Copyright © 2017–2018, Texas Instruments Incorporated

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/sampterms.htm).

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2018, Texas Instruments Incorporated