CC3100 SimpleLink™ Wi-Fi® and IoT Solution Getting Started Guide

User's Guide

![Texas Instruments logo]

Contents

List of Figures

CC3100 SimpleLink™ Wi-Fi® and IoT Solution Getting Started Guide

ABSTRACT

This guide is intended to assist users in the initial setup and demonstration of the *Getting Started with WLAN Station* application. The guide explains how to install an Integrated Development Environment (IDE), and then compile, download, and debug *Getting Started with WLAN Station*.

1 Introduction

1.1 Prerequisites

The user should have the following items:
- One CC3100BOOST
- One CC31XXEMUBOOST or MSP430F5529 Launchpad
- An 802.11b/g/n Wireless Access Point (AP)
- A computer running Microsoft® Windows® 7 or XP operating systems.

2 Getting Started

2.1 Download and Install Software

Download and install the following software:
- CC3100 SDK package http://www.ti.com/tool/cc3100sdk
  - This guide assumes the use of the default installation folder *C:\TI\CC3100SDK_<x.x.x>*.

2.2 Update Service Pack

If the board is not already flashed with the service pack for SDK_<x.x.x>, the latest service pack for SDK_<x.x.x> must be flashed on the CC3100. The latest service pack can be downloaded from http://www.ti.com/tool/cc3100sdk. Refer to the UNIFLASH Quick Start guide for details on flashing the service pack to the CC3100 (http://processors.wiki.ti.com/index.php/CC31xx_%26_CC32xx_UniFlash#Service_Pack_Programming). Details on setting up the hardware for flashing can be found in *Section 3.1*.

SimpleLink is a trademark of Texas Instruments.
Microsoft, Windows are registered trademarks of Microsoft.
Wi-Fi is a registered trademark of Wi-Fi Alliance.

3   Getting Started with SimpleLink Studio

3.1   *Configure Boards*

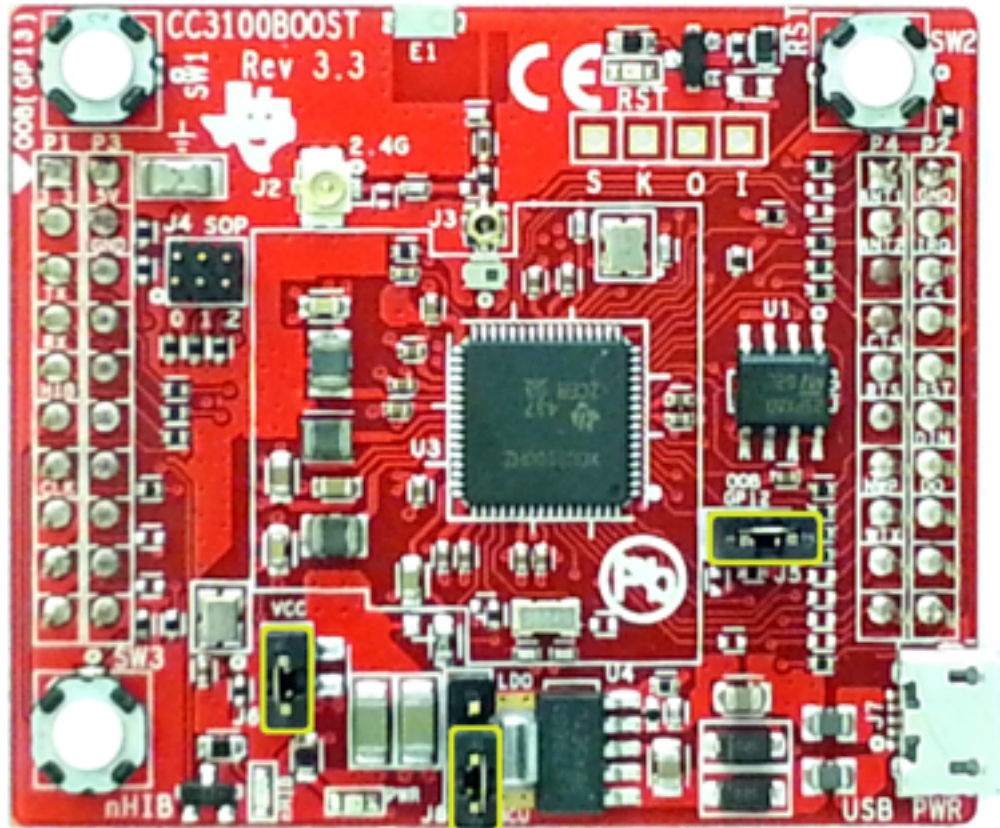1. The jumpers on the CC3100BOOST should be connected as shown in Figure 1.

Figure 1. CC3100BOOST

CC3100 SimpleLink™ Wi-Fi® and IoT Solution Getting Started Guide

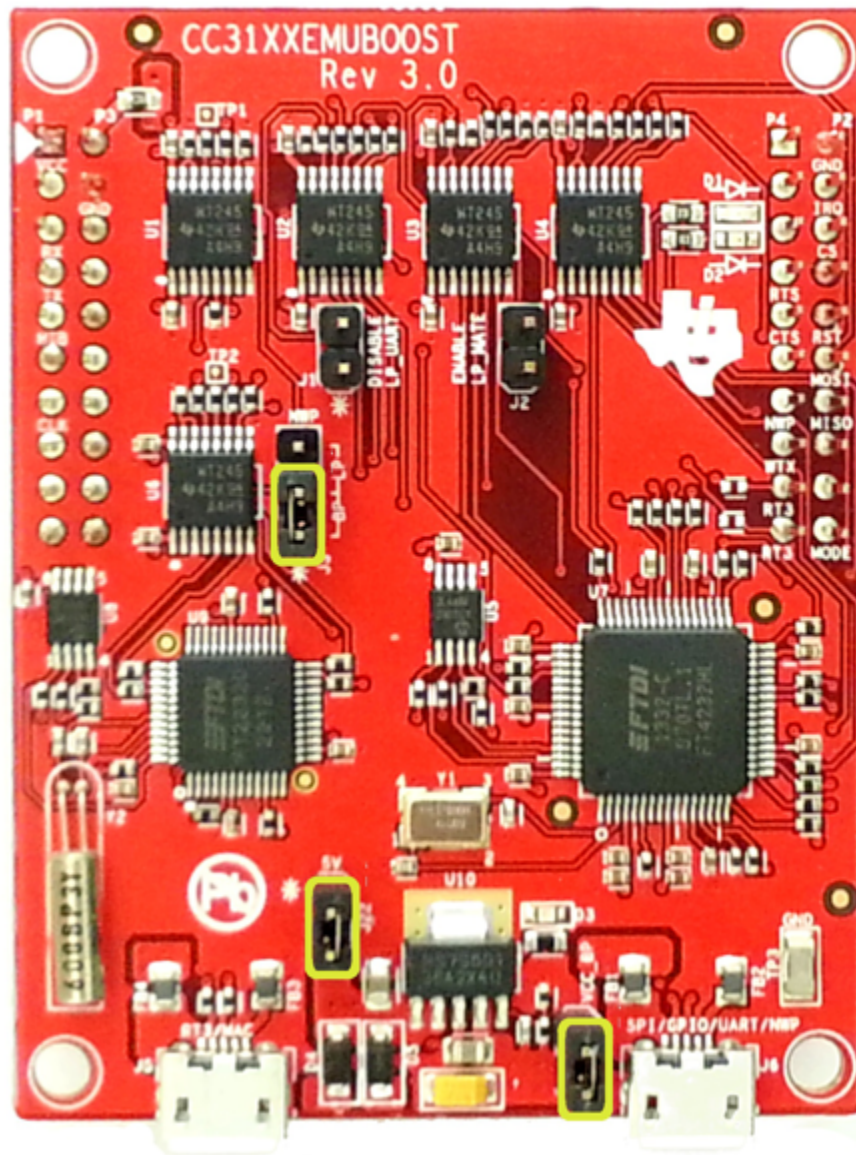2. The jumpers on the CC31XXEMUBOOST should be connected as shown in Figure 2.

Figure 2. CC31XXEMUBOOST

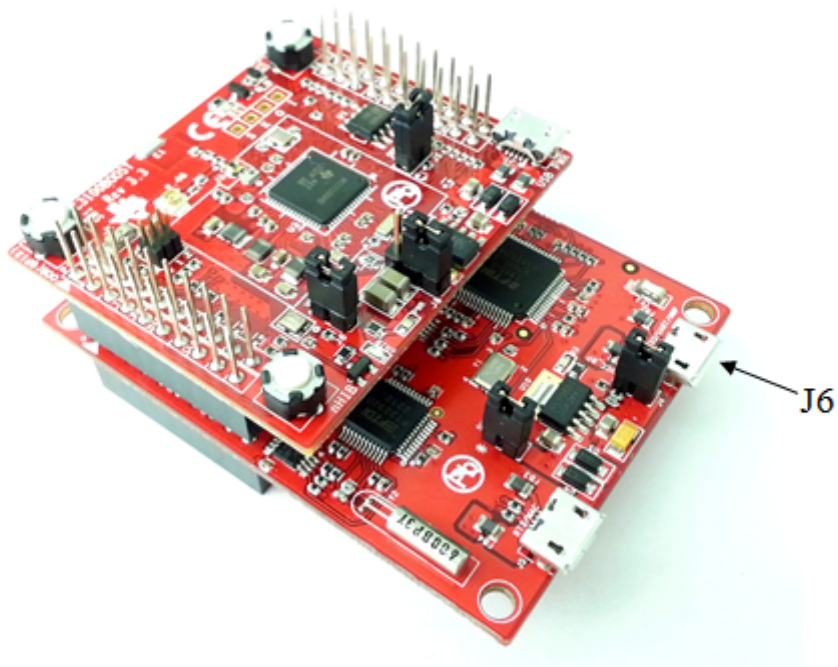3. Connect the CC3100BOOST to the CC31XXEMUBOOST as shown in Figure 3.

Figure 3. Connect the CC3100BOOST to the CC31XXEMUBOOST

4. Connect the J6 port CC31XXEMUBOOST to the PC using the provided micro-USB cable.
5. The CC3100BOOST is now visible in the Device Manager as shown in Figure 4. The user may see two COM ports instead of four.
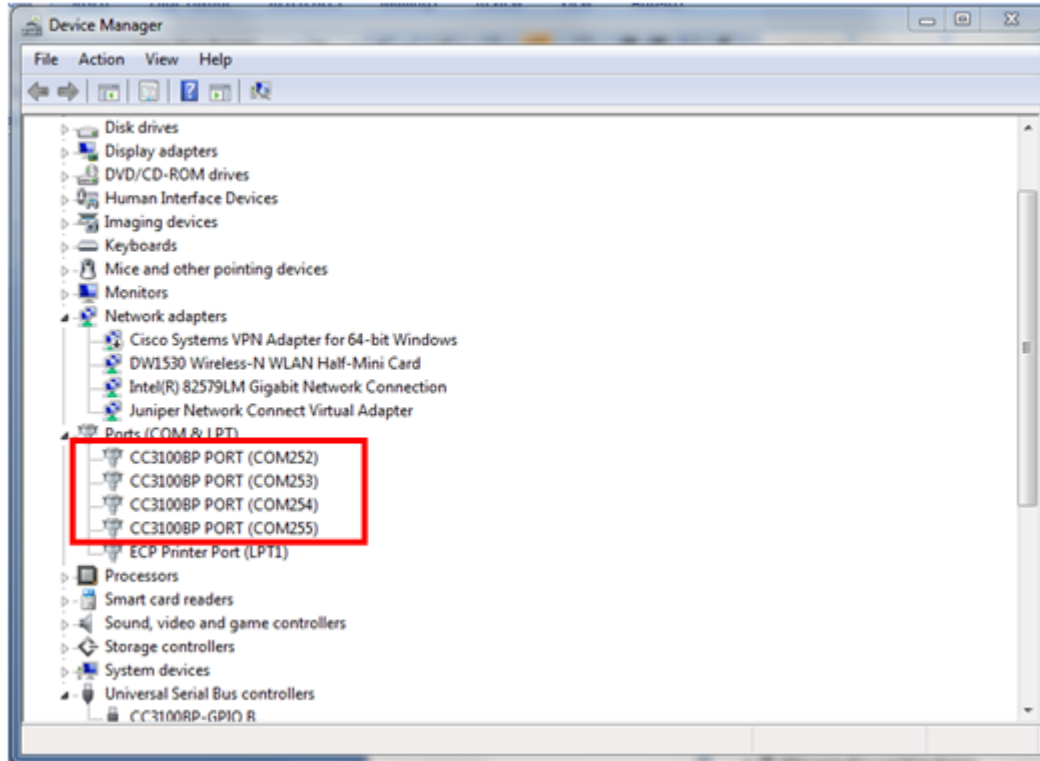
Copyright © 2014–2016, Texas Instruments Incorporated

Figure 4. Device Manager

3.2 Run the Software

The *Getting Started with WLAN Station* example uses SimpleLink Studio. This example performs the following functions:

1. Program restores WLAN configuration to factory default
2. Prompts the user for the SSID of an AP
3. Prompts the user for the security type
4. Prompts the user for the password to the AP
5. Attempts to acquire an IP address through DHCP
6. Attempts to reach the internet

Option 1. Microsoft Visual Studio:

1. Download and install Microsoft Visual Studio Express 2010 from http://www.visualstudio.com/en-us/downloads/download-visual-studio-vs#DownloadFamilies_4.
2. Open Microsoft Visual Studio Express, and select *File>Open>Project/Solution*.
3. Navigate to *C:\TI\CC3100SDK_<x.x.x>\cc3100-sdk\platform\simplelinkstudio\example_project_vs\getting_started_with_station*, and open *getting_started.sln*.
4. Select *Build>Build Solution* from the menu.
5. When building is complete, select *Debug>Start Debugging* from the menu.

Option 2. Eclipse:

1. Download and install the latest version of Java: https://www.java.com/en/download/. Install the correct version for the system (64-bit or 32-bit).

2. Download and extract Eclipse from http://www.eclipse.org/downloads/. Choose the Eclipse IDE for C/C++ Developers package. Install the correct version for the system (64-bit or 32-bit).

3. Download and install MinGW from http://sourceforge.net/projects/mingw/files/latest/download?source=files. During installation, ensure the following configurations are selected:

    (a) Set the installation location as *C:\MinGW*.

    (b) In the MinGW Installation Screen, select packages for **mingw32-base** and **mingw32-gcc-g++** as shown in Figure 5.
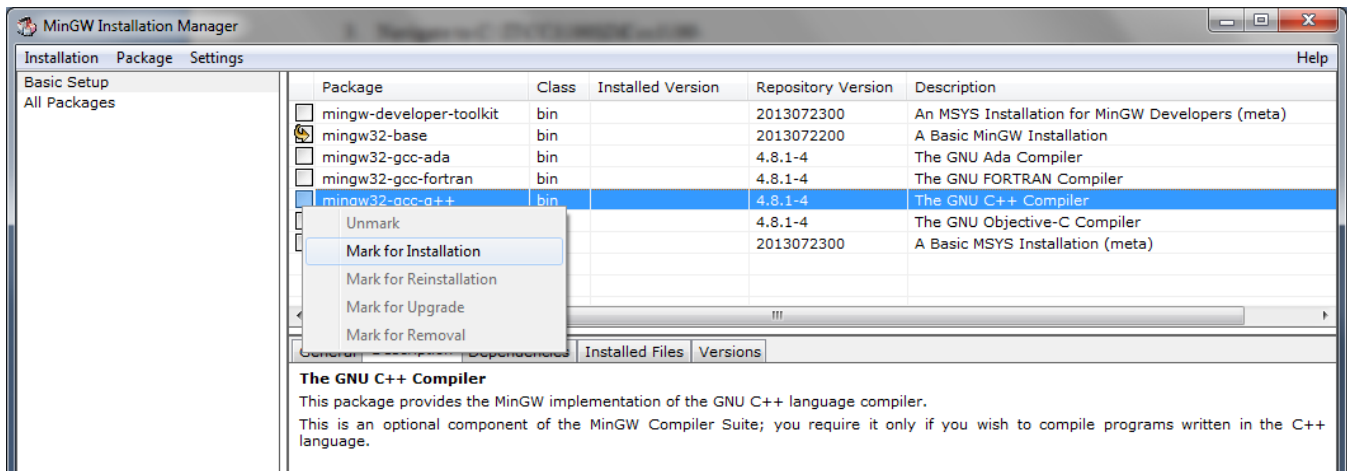

Figure 5. MinGW Installation Screen

    (c) After selecting the packages, choose *Installation>Apply Changes* from the menu, then press Apply.

    (d) Select *Installation>Quit* from the menu to quit the installer.

4. After a successful MinGW installation, add the path (*c:\MinGW\bin*) to the Windows environment variable path by going into *Control Panel>System>Advanced System Settings>Environment Variables*. Under *System Variables*, select PATH and press Edit. Append ";C:\MinGW\bin\" to the end of the line and press Ok.

5. Open Eclipse by running *eclipse.exe* from the extracted Eclipse folder.

6. In the "Select a workspace," choose your desired workspace directory (for example: *C:\Users\myself\Desktop\eclipse_workspace*). This directory should be different from the user's other versions of Eclipse.

7. From the menu, select *File>New>Makefile Project with Existing Code*.

8. Enter 'Getting_started' as the project name.

9. For Existing Code Location, enter *C:\TI\CC3100SDK_<x.x.x.>\cc3100-sdk\platform\simplelinkstudio\example_project_eclipse\getting_started_with_station*.

10. For Toolchain for indexer settings, choose "MinGW GCC" as shown in Figure 6.
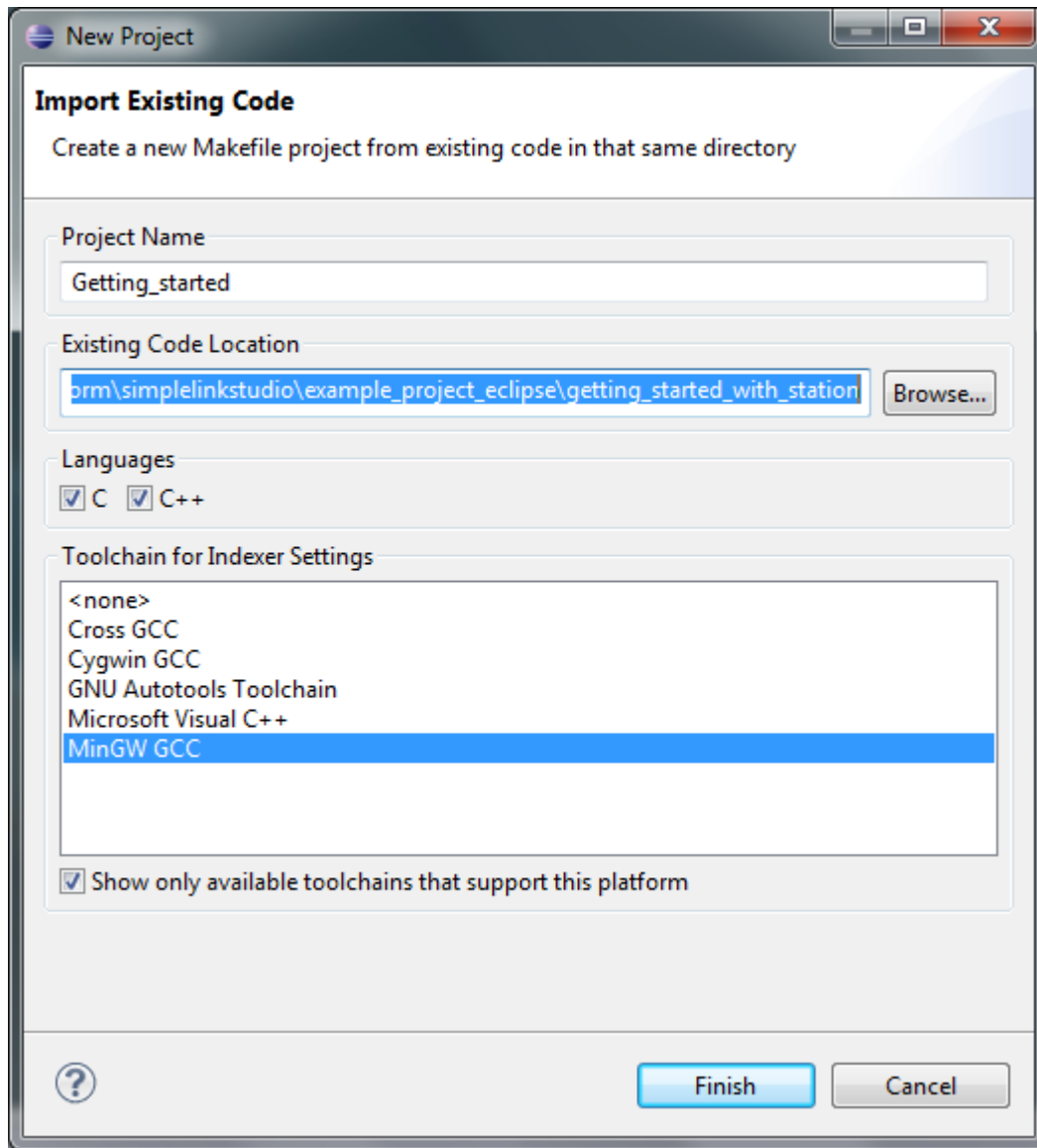
Figure 6. Import Existing Code

11. Press Finish.
12. Select *Window>Show View>Project Explorer* from the menu.
13. Select the Getting_started project in the Project Explorer, and select *File>Properties* from the menu.
14. In the Project Explorer window, right click on *<Project_Folder_Name>*, then select Property.
15. Expand the C/C++ Build menu, and select Tool Chain Editor. Under Current Builder, Select "Gnu Make Builder", then click Apply.
16. Click the C/C++ Build menu and do the following:
    (a) Uncheck 'Use the default build command.'
    (b) Type *mingw32-make -f Makefile* in the Build command.
    (c) Uncheck 'Generate Makefiles automatically.'
    (d) Set the Build Directory as: *${workspace_loc:/Getting_started}*
17. Select Environment in the C/C++ Build menu. Ensure the value of MSYS_HOME is empty, then click Apply.

18. Click the C/C++ General menu, and select Paths and Symbols. Under the Includes tab, in the Languages column, select GNU C. Press the Add button to add the directory: *C:\TI\CC3100SDK_<x.x.x>\cc3100-sdk\simplelink\include*.
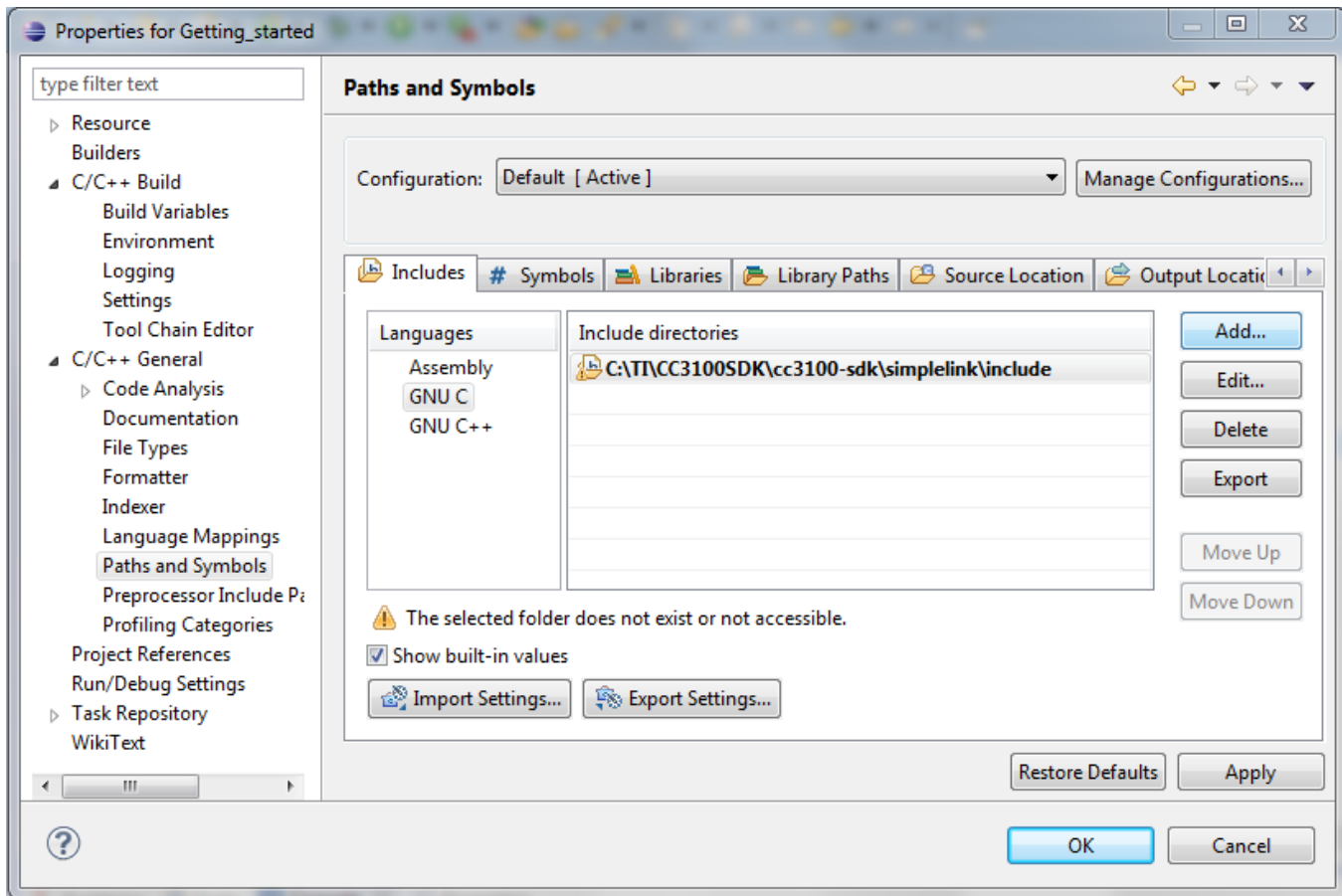

Figure 7. Paths and Symbols

19. Press OK.

20. To fix a known bug in the Eclipse console output, add the following line of code to the beginning of the main function:

```
setvbuf(stdout, NULL, _IONBF, 0);
```

21. Save the file, and select *Project>Clean* from the menu.

22. Select the Getting_started project and press OK.

23. Select the Getting_started project from Project Explorer, and from the menu select *Project>Build Project*.

24. Press Ctrl+F11 to start the program.

4   Getting Started with the MSP430F5529

4.1   Configure Boards

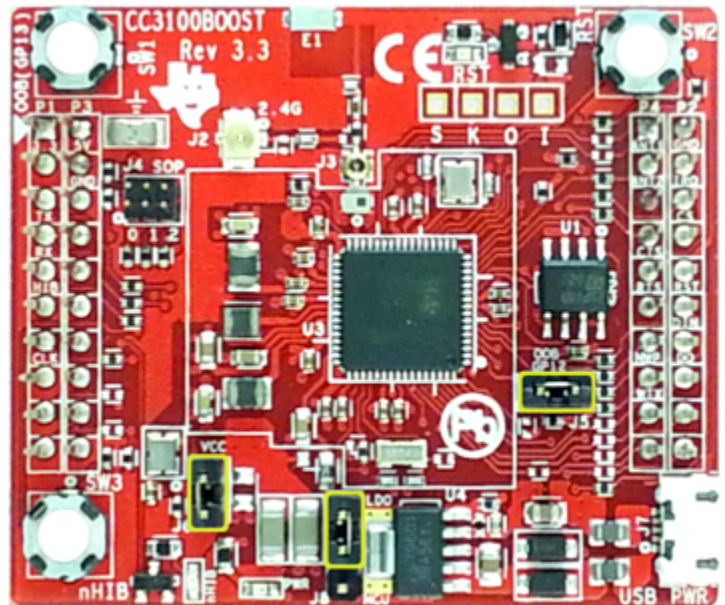1. The jumpers on the CC3100BOOST should be connected as shown in Figure 8.

Figure 8. CC3100BOOST

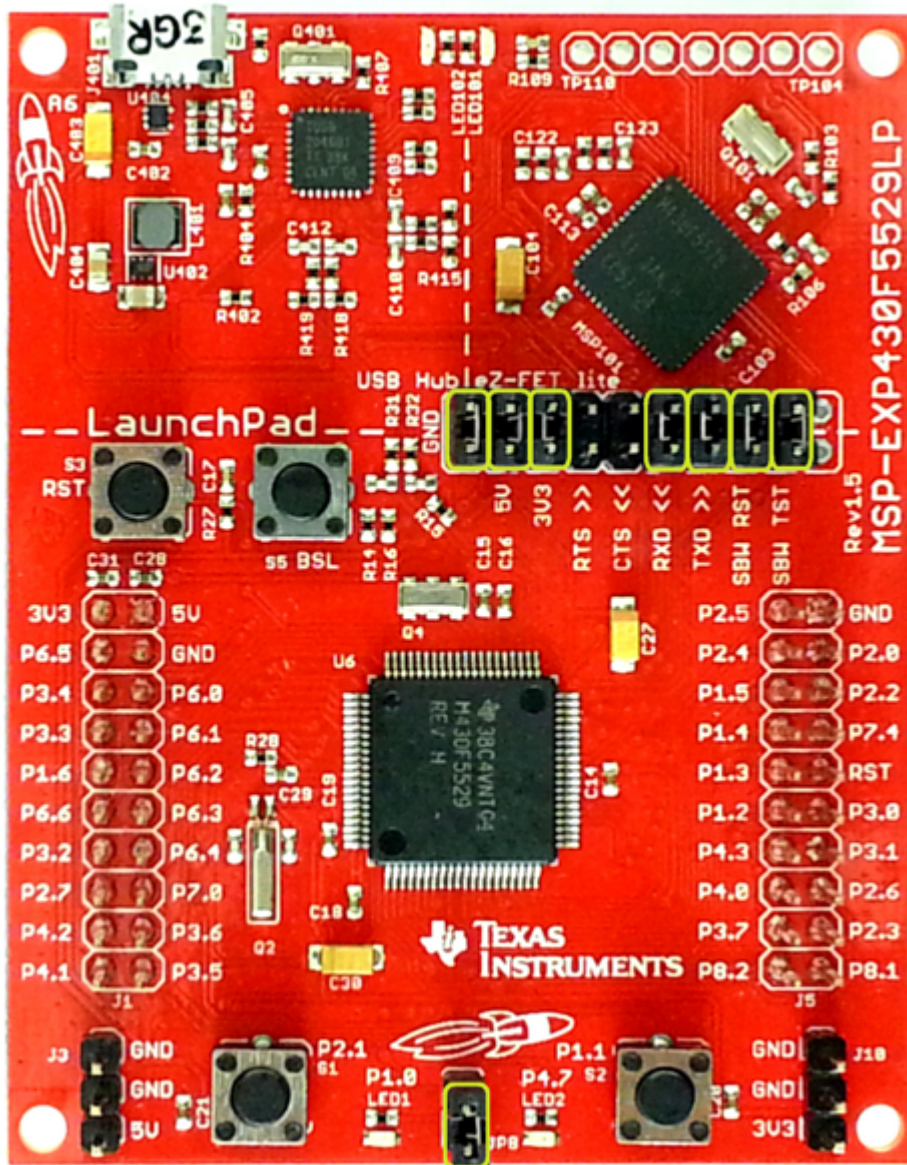2. The jumpers on the MSP430F5529 Launchpad should be connected as shown in Figure 9.

Figure 9. Jumpers on the MSP430F5529 Launchpad

3. Connect the CC3100BOOST to the MSP430F5529 Launchpad as shown in Figure 10.
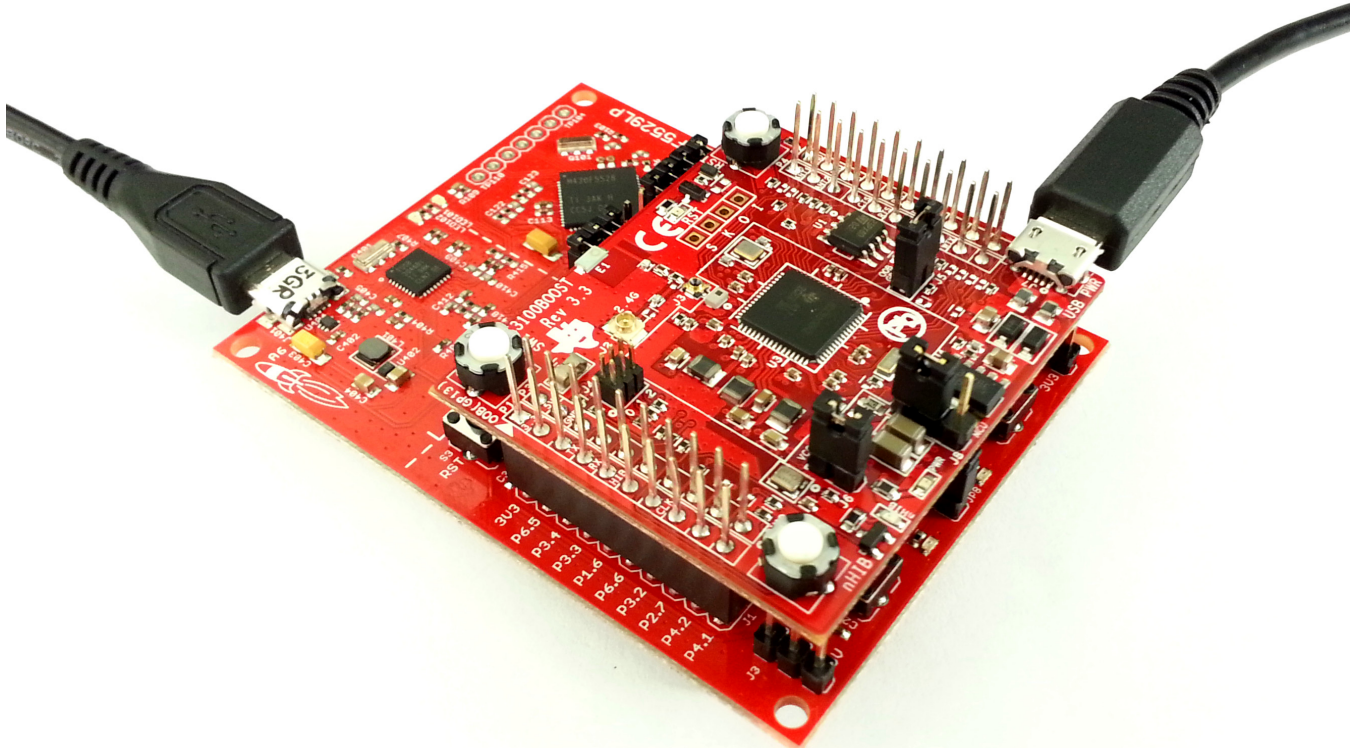

Figure 10. Connect the CC3100BOOST to the MSP430F5529 Launchpad

4. Connect a Micro-USB cable from J7 on the Boosterpack to the Windows PC, and connect a Micro-USB cable from J401 on the MSP430F5529 Launchpad to the Windows PC.

4.2 Run the Software

The example shown provides output using UART. To see the output from the program, download Tera Term (or similar software) from http://en.sourceforge.jp/projects/ttssh2/releases/

Option 1. Code Composer Studio (CCS):

1. Download and run the Code Composer Studio 6.1.0 (CCS) installation wizard (*ccs_setup_win32.exe*) from the TI website or from the CCS Wiki page. The program must be Version 6.1.0.00104 or later. When prompted to select processor support, select the *MSP Ultra Low Power MCUs* processor support option. The remaining options for the installer should be left as the default. Installation may take up to an hour.

2. Open CCS, and choose *File>Import* from the menu. Under C/C++, choose CCS Projects.

3. Under *Select Search Directory*, enter the path: *C:\TI\CC3100SDK_<x.x.x>\cc3100-sdk\platform\msp430f5529lp\example_project_ccs*.

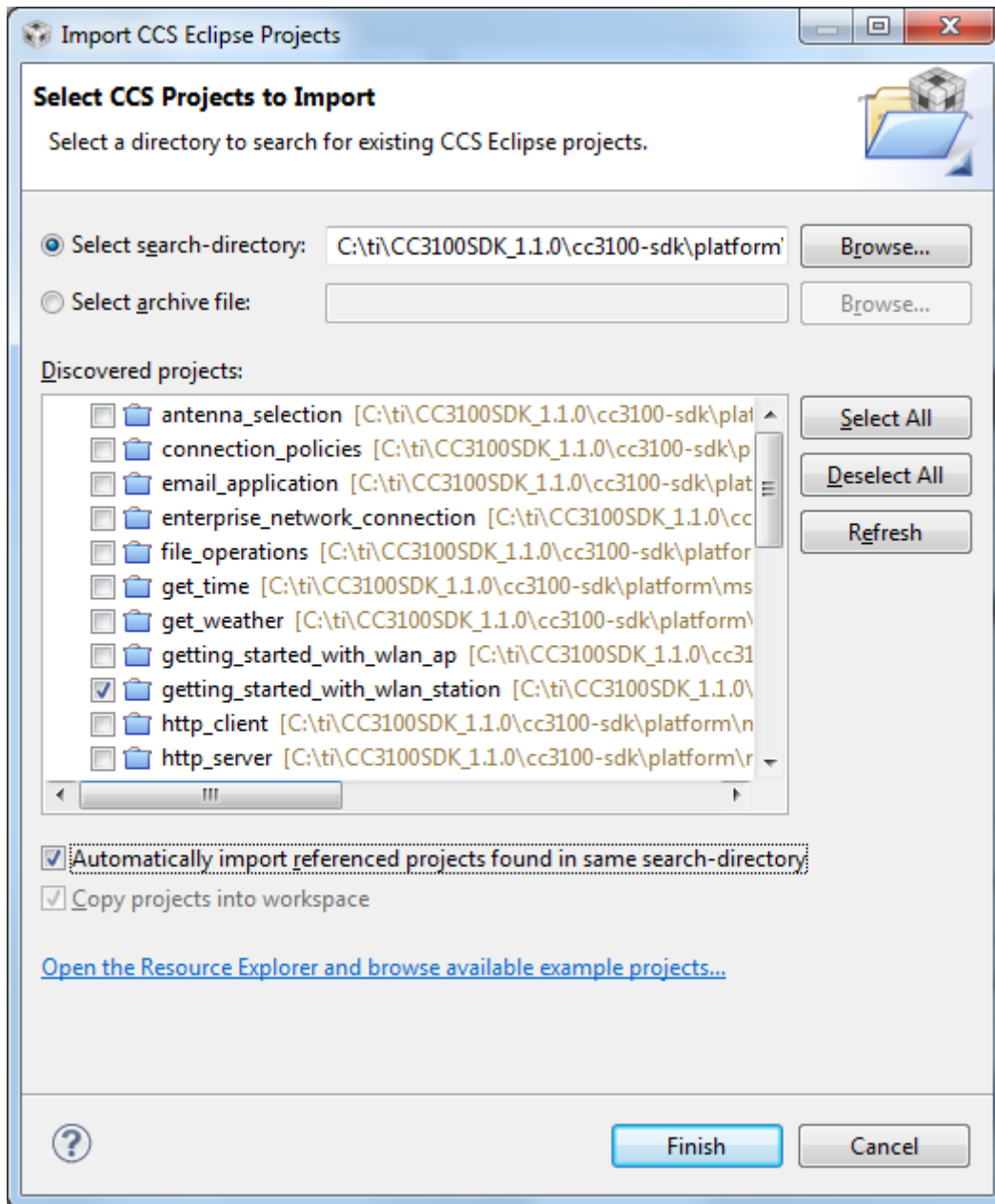4. Check the project *getting_started_with_wlan_station* and press Finish.

Figure 11. Select CCS Projects to Import

5. Open the *sl_common.h* file located at the path C:\TI\CC3100SDK_<x.x.x>\cc3100-sdk\examples\common\.

6. Edit *sl_common.h* to use the SSID, security type, and security key of the AP being used. Edit the macros SSID_NAME, SEC_TYPE, and PASSKEY to contain the AP information as shown in Figure 12. The security types supported for this demo are WPA/WPA2, WEP (5/13 ASCII), and Open. WEP supports key index 1 only. For Open security, define SEC_TYPE as SL_SEC_TYPE_OPEN. For WPA and WPA2 security, define it as SL_SEC_TYPE_WPA. For WEP security, define it as SL_SEC_TYPE_WEP.

Figure 12. Define SSID_Name

7. Select the *getting_started_with_wlan_station* project in Project Explorer and select *Project>Build Project* from the menu.

8. Launch Tera Term, and create a new serial connection to the MSP430F5529 Launchpad COM port as shown in Figure 13. The baud rate should remain at 9600.
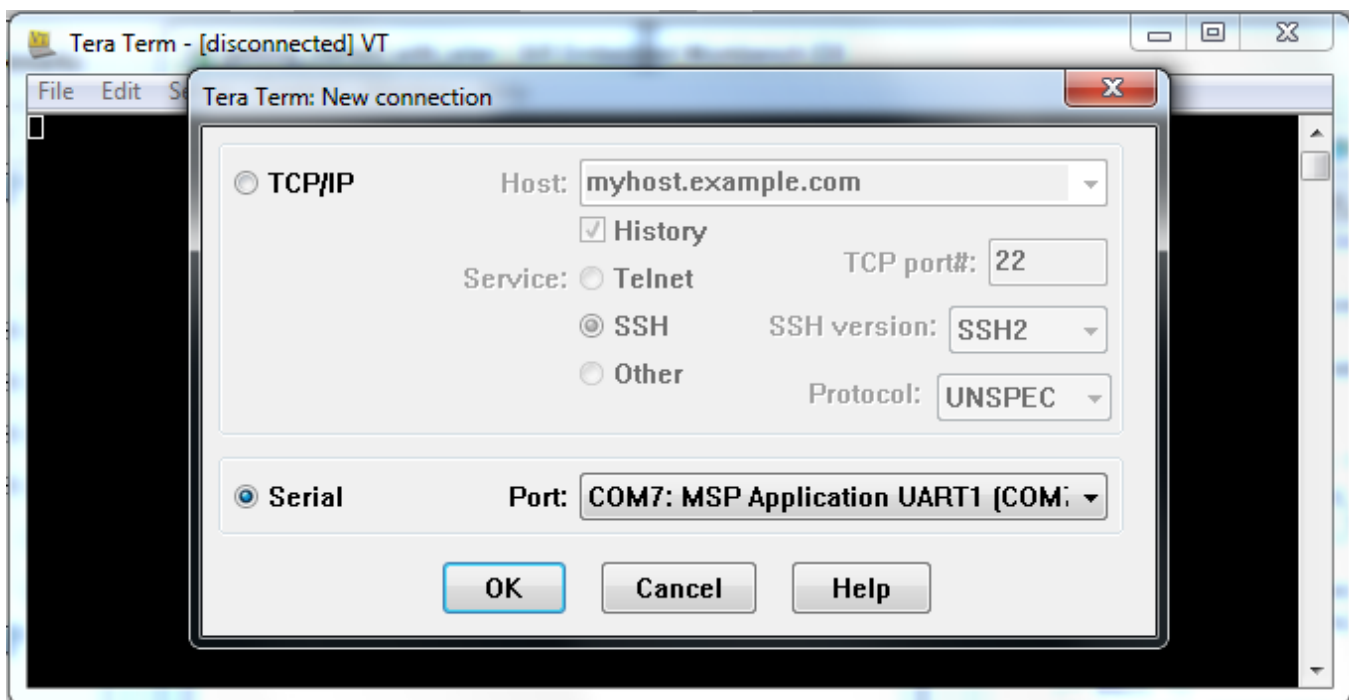
9. Press F11on the CCS window to start debugging.

Figure 13. Launch Tera Term

10. If the CC3100 successfully completes all steps, the serial output appears as shown in Figure 14.

Figure 14. Tera Term VT

Option 2. IAR Workbench

1. Install IAR Workbench for MSP430 version 6.20 or later: http://www.iar.com/en/Products/IAR-Embedded-Workbench/TI-MSP430/.

2. Open IAR Workbench and select *File>Open>Workspace* from the menu.

3. Select the project: *C:\TI\CC3100SDK_1.2.0\cc3100-sdk\platform\msp430f5529lp\example_project_iar\getting_started_with_wlan_station\getting_started_with_wlan.eww*.

4. Open the *sl_common.h* file located at the path C:\TI\CC3100SDK_<x.x.x>\cc3100-sdk\examples\common\.

5. Edit *sl_common.h* to use the SSID, security type and security key of the AP being used. Edit the macros SSID_NAME, SEC_TYPE, and PASSKEY to contain the AP information as shown in Figure 15. The security types supported for this demo are WPA/WPA2 and Open. For Open security, define SEC_TYPE as SL_SEC_TYPE_OPEN. For WPA and WPA2 security, define it as SL_SEC_TYPE_WPA.

Figure 15. Macro Definition of SSID_Name

6. Select *Project>Rebuild All* from the menu.

7. Launch Tera Term, and create a new serial connection to the MSP430F5529 Launchpad COM port as shown in Figure 16. The baud rate should remain at 9600.

8. After building is finished, select *Project>Download and Debug* from the menu to start debugging.

Figure 16. Launch Tera Term

9. If the CC3100 successfully completes all steps, the serial output appears as shown in Figure 17.
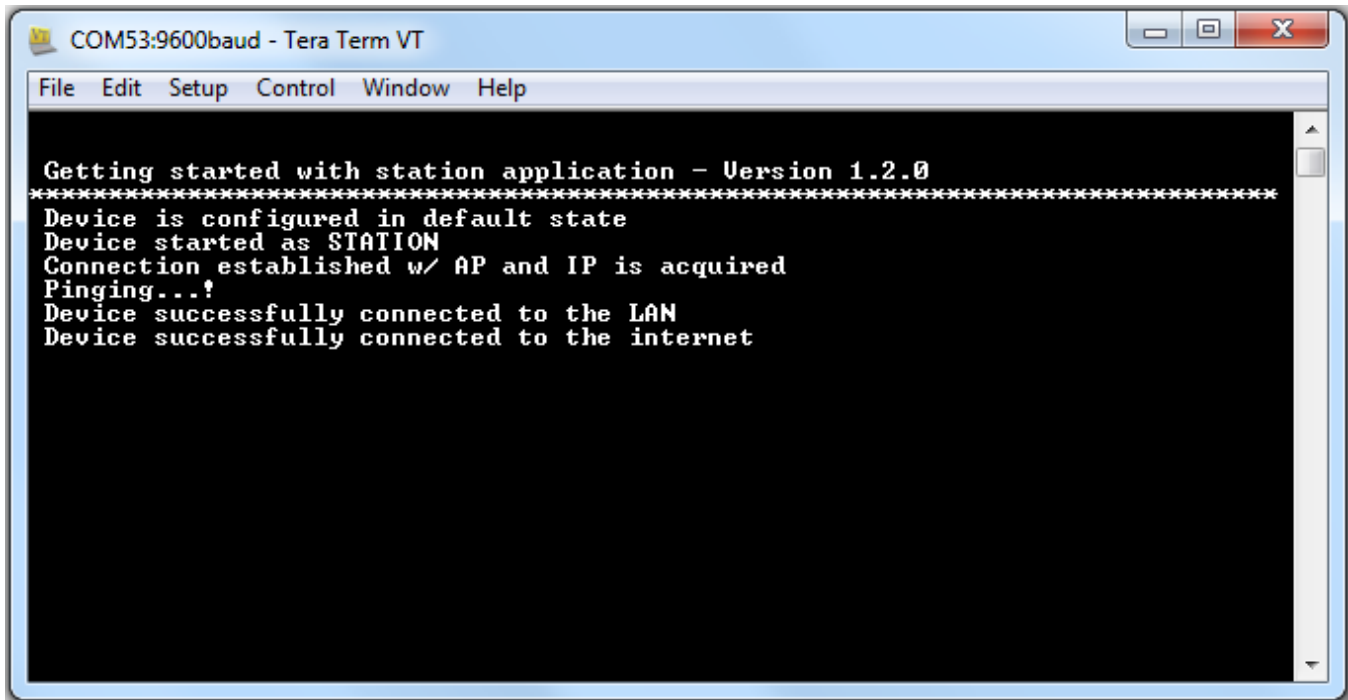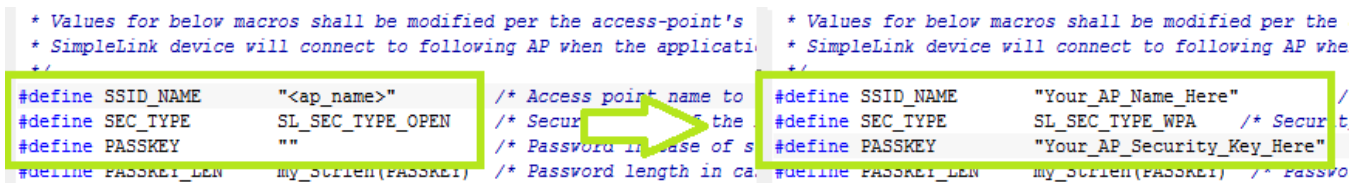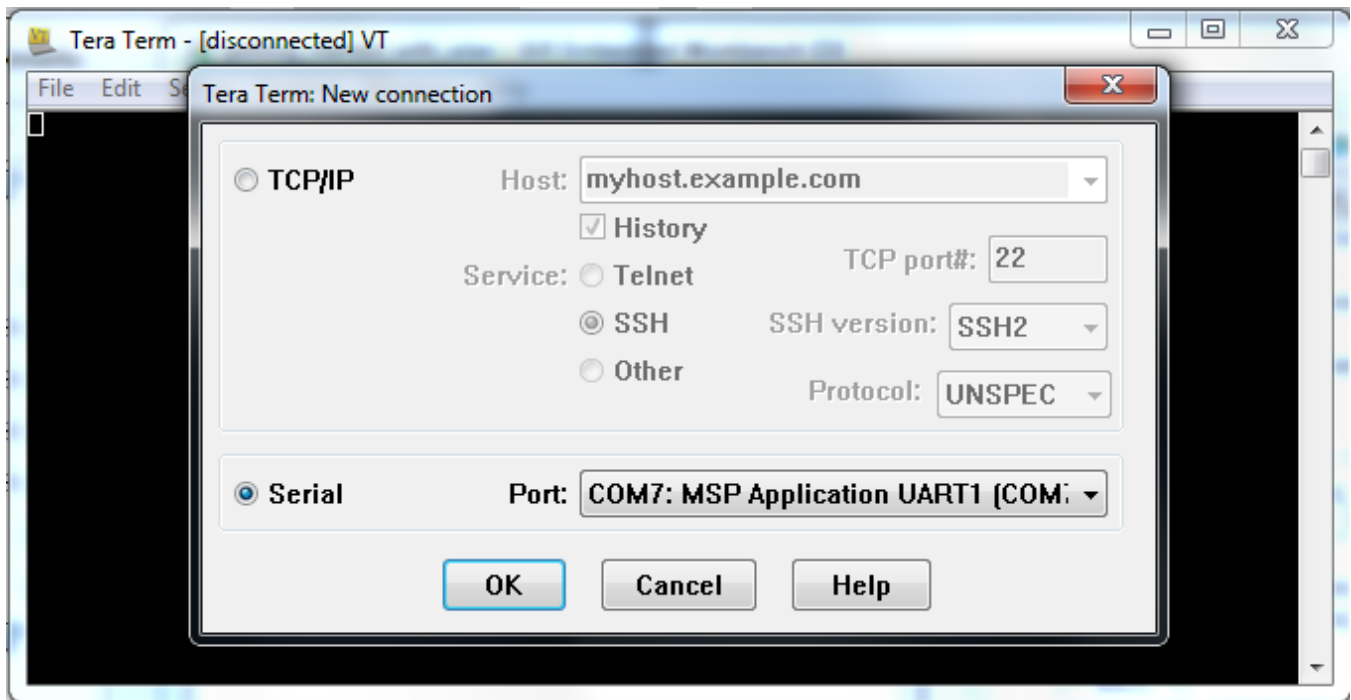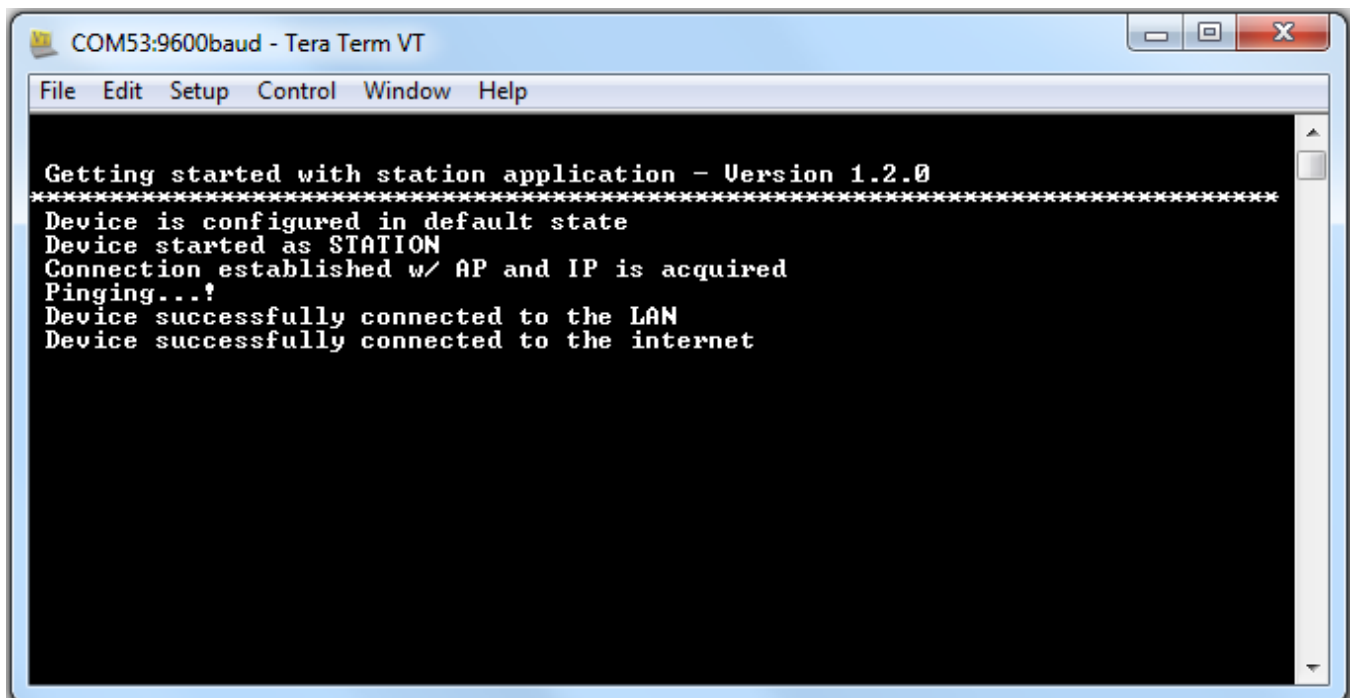

Figure 17. Tera Term VT

5   Summary

After the development environment has been set up, see the following resources for further assistance in development:

- CC3100 Programmer's Guide – This guide contains information on how to use the SimpleLink API for writing WLAN-enabled applications.
- Uniflash – The Uniflash tool is used for manually storing files on the external serial flash. This includes the SimpleLink firmware patch file and any configuration files, security certificates, web pages, and so forth.
- CC3100 Wiki – All information and tools for the CC3100, including the above, can be found on the CC3100 Wiki page.

6   Acronyms Used

STA – Wi-Fi Station

AP – Wi-Fi Access Point

WLAN – Wireless LAN

CCS – Code Composer Studio

GCC – GNU Compiler Collection

Revision History

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

Revision History

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as "components") are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI's terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers' products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers' products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information 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.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI's goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or "enhanced plastic" are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have *not* been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products		Applications	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		