

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

User's Guide



Literature Number: SWRU375D
June 2014–Revised May 2018

1	Trademarks	4
2	Introduction.....	4
2.1	Prerequisites	4
3	Getting Started	4
3.1	Download and Install Software	4
3.2	Update Service Pack	4
4	Getting Started with SimpleLink Studio	5
4.1	Configure Boards.....	5
4.2	Run the Software.....	8
5	Getting Started with the MSP430F5529.....	13
5.1	Configure Boards	13
5.2	Run the Software	15
6	Summary.....	20
7	Acronyms Used.....	20
	Revision History.....	21

List of Figures

1	CC3100BOOST	5
2	CC31XXEMUBOOST	6
3	Connect the CC3100BOOST to the CC31XXEMUBOOST	7
4	Device Manager.....	8
5	MinGW Installation Screen	9
6	Import Existing Code.....	10
7	Paths and Symbols	11
8	Eclipse Output Console	11
9	CC3100BOOST	13
10	Jumpers on the MSP430F5529 Launchpad	14
11	Connect the CC3100BOOST to the MSP430F5529 Launchpad	15
12	Select CCS Projects to Import.....	16
13	Define SSID_Name	17
14	Launch Tera Term	17
15	Tera Term VT.....	18
16	Macro Definition of SSID_Name	18
17	Launch Tera Term	19
18	Tera Term VT.....	19

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 Trademarks

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

2 Introduction

2.1 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.

3 Getting Started

3.1 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>\.

3.2 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 4.1](#).

4 Getting Started with SimpleLink Studio

4.1 Configure Boards

1. The jumpers on the CC3100BOOST should be connected as shown in [Figure 1](#).

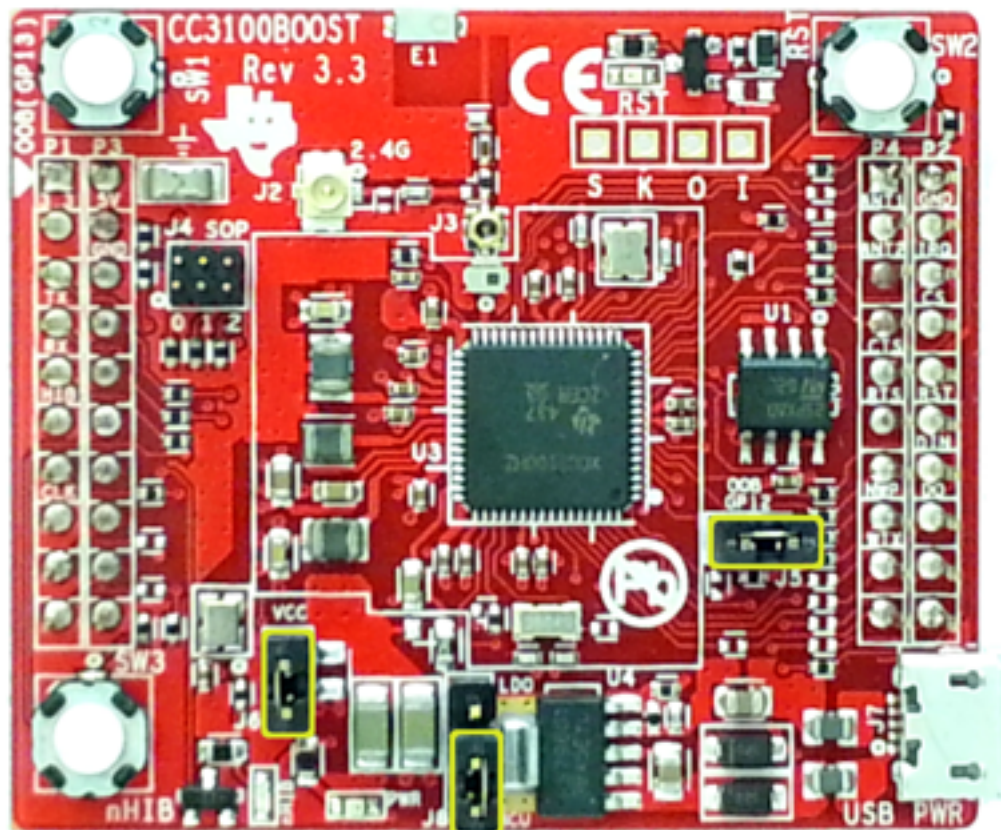


Figure 1. CC3100BOOST

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

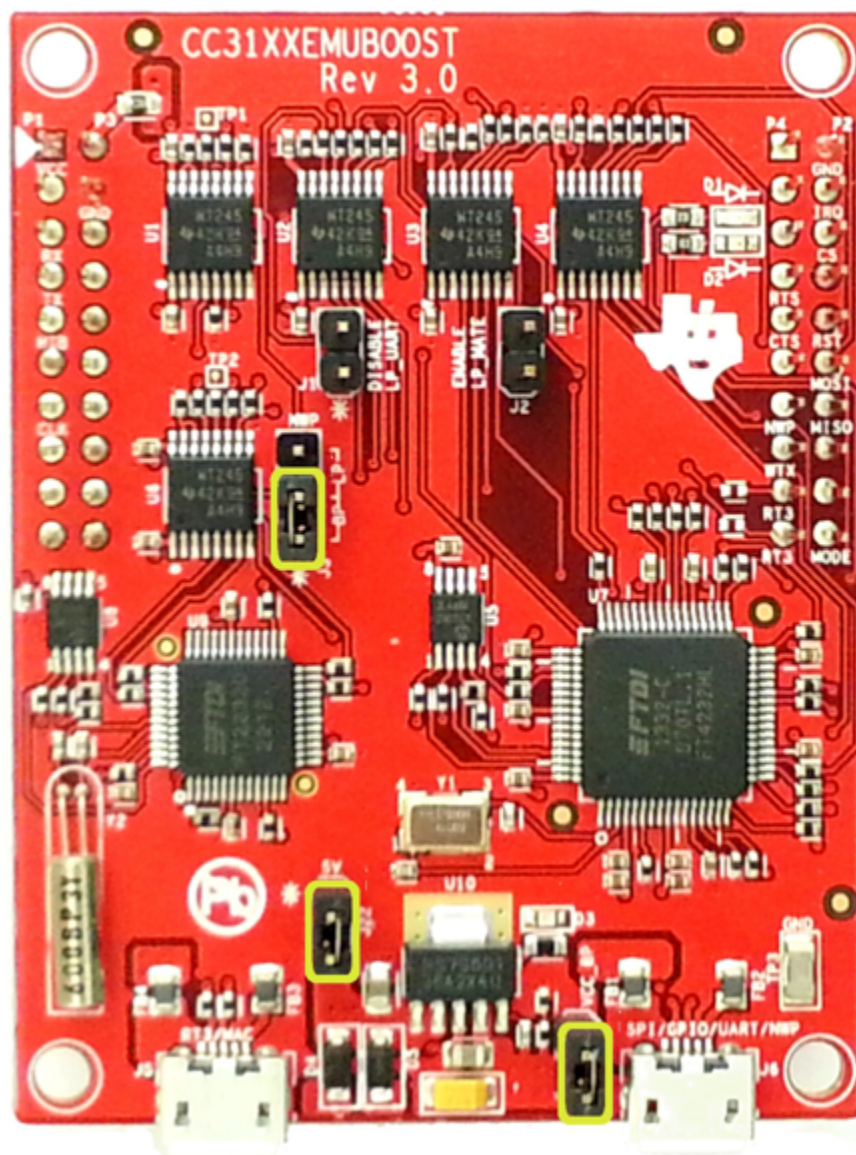


Figure 2. CC31XXEMUBOOST

3. Connect the CC3100BOOST to the CC31XXEMUBOOST as shown in [Figure 3](#).

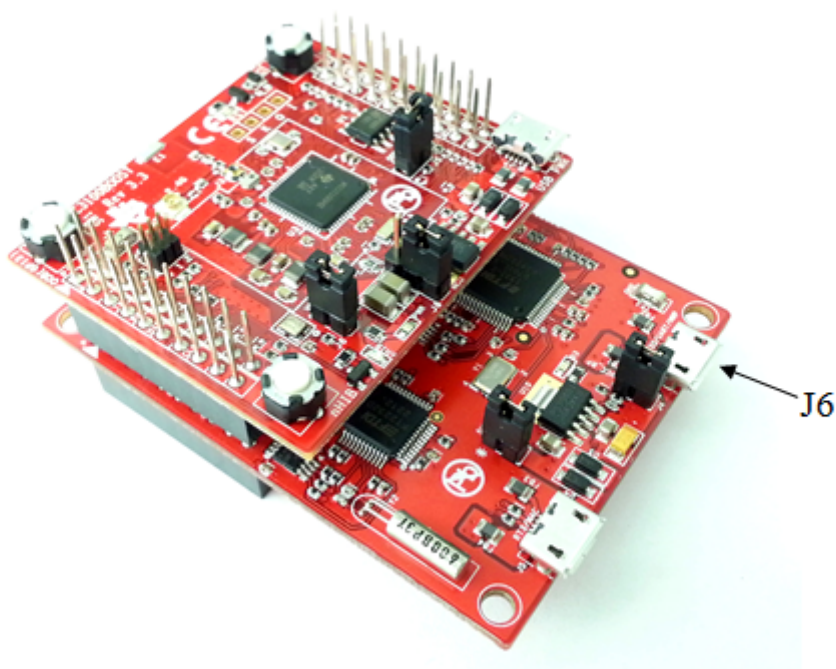


Figure 3. Connect the CC3100BOOST to the CC31XXEMUBOOST

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.

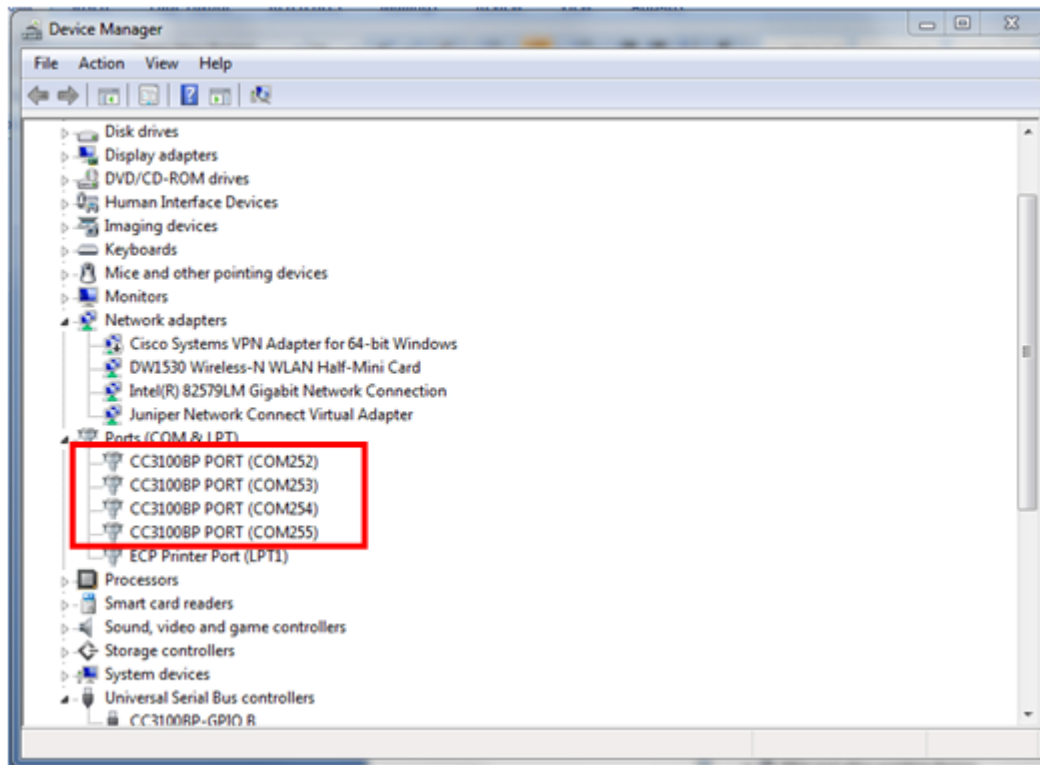


Figure 4. Device Manager

4.2 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

NOTE: For full compatibility ensure Visual Studio 2010 is used. Newer versions of Visual Studio will not work with the device. Visual Studio 2012 works with the "getting started" example, but gives the error code -2003 for the weather example, and error code -2005 for the email app.

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.

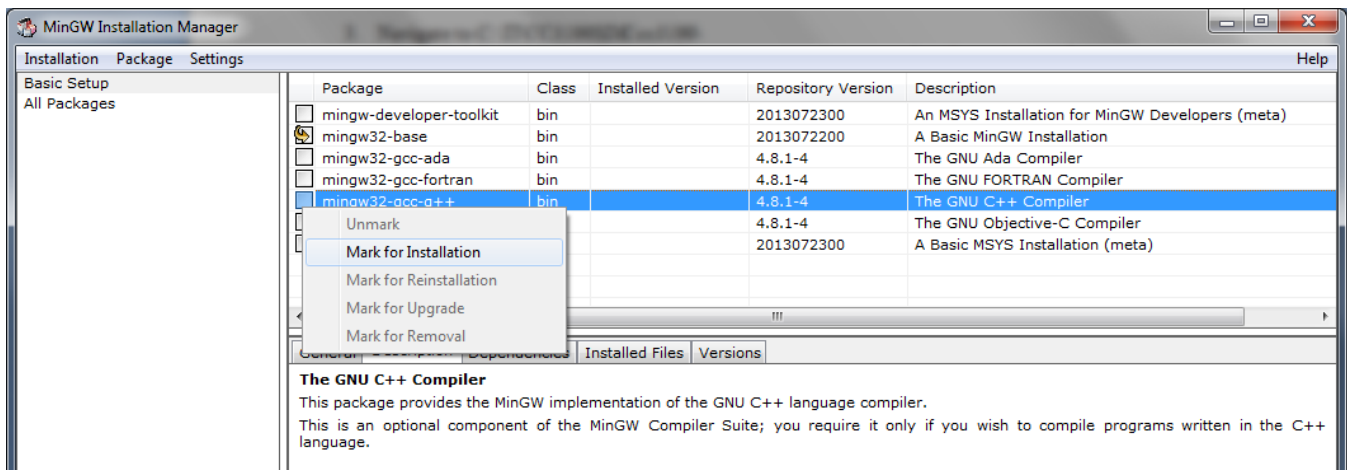


Figure 5. MinGW Installation Screen

- 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.

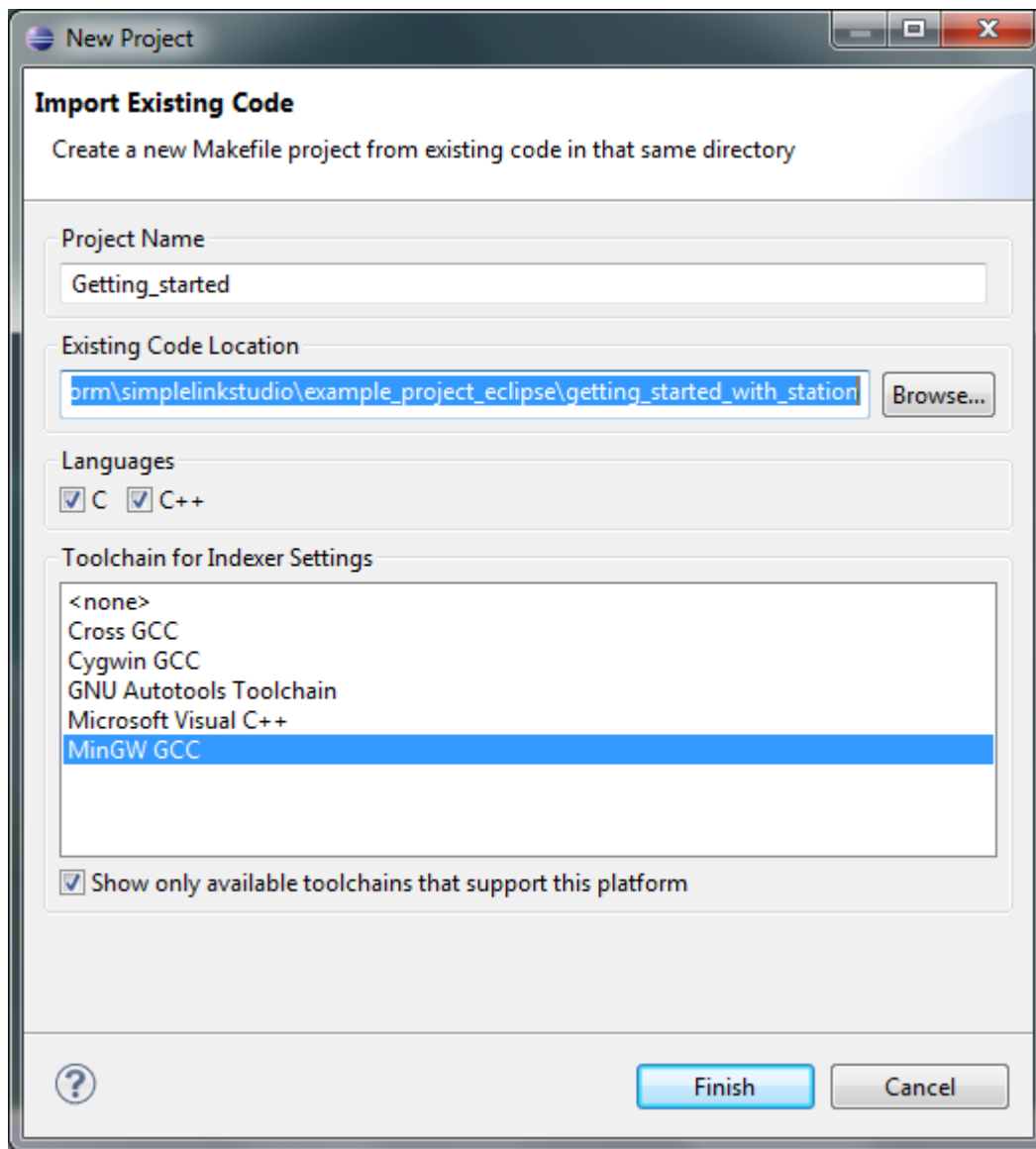


Figure 6. Import Existing Code

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 .

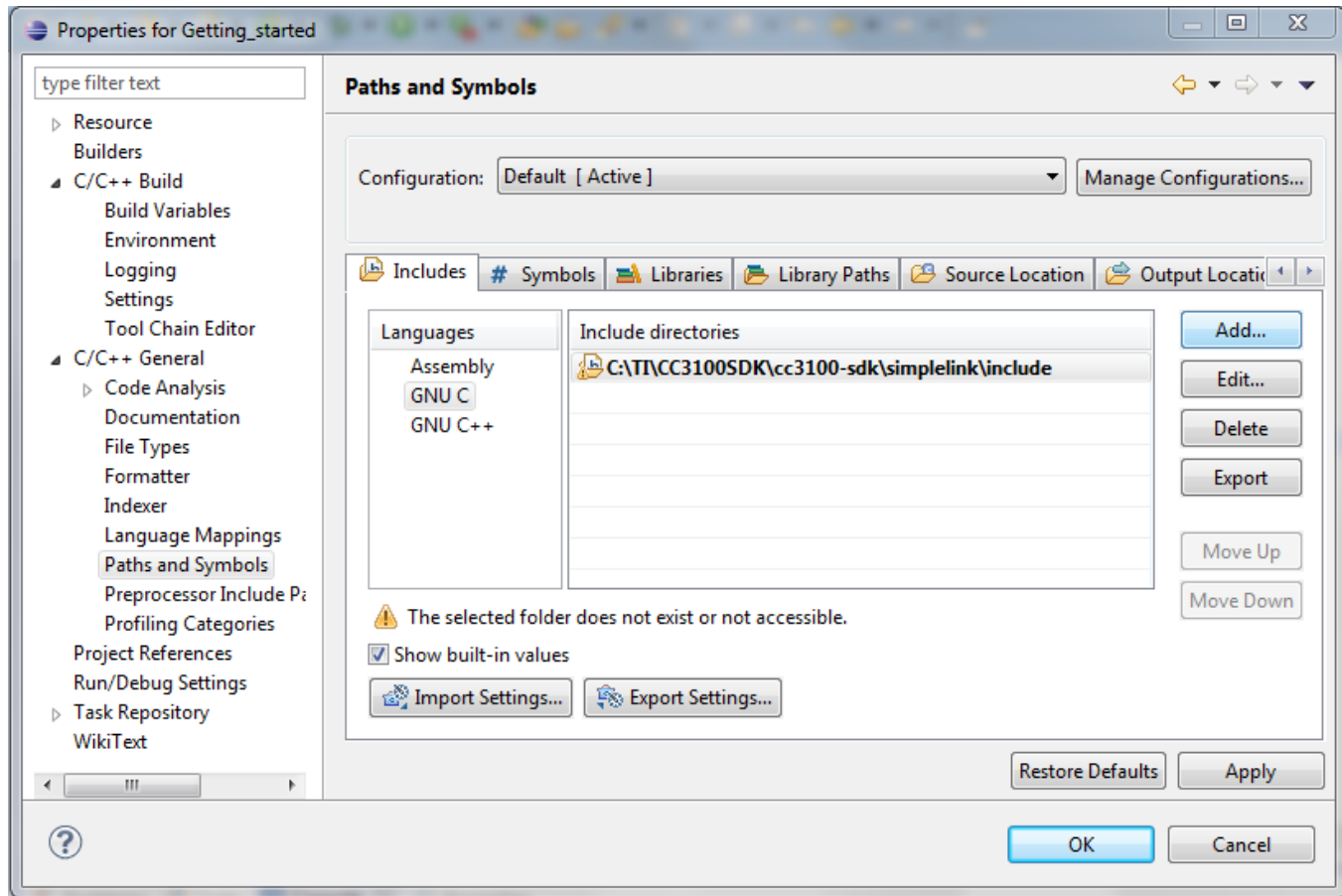


Figure 7. Paths and Symbols

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);
```



Figure 8. Eclipse Output Console

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.

5 Getting Started with the MSP430F5529

5.1 Configure Boards

1. The jumpers on the CC3100BOOST should be connected as shown in [Figure 9](#).

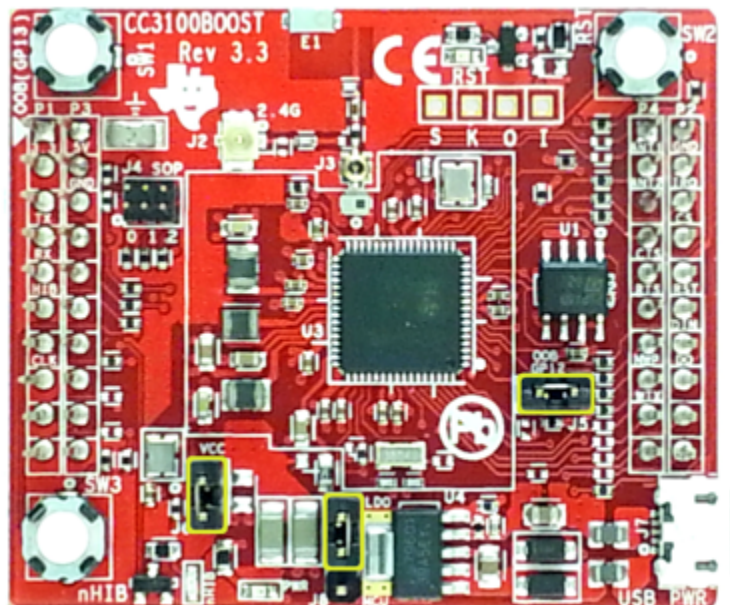


Figure 9. CC3100BOOST

- The jumpers on the MSP430F5529 Launchpad should be connected as shown in Figure 10.

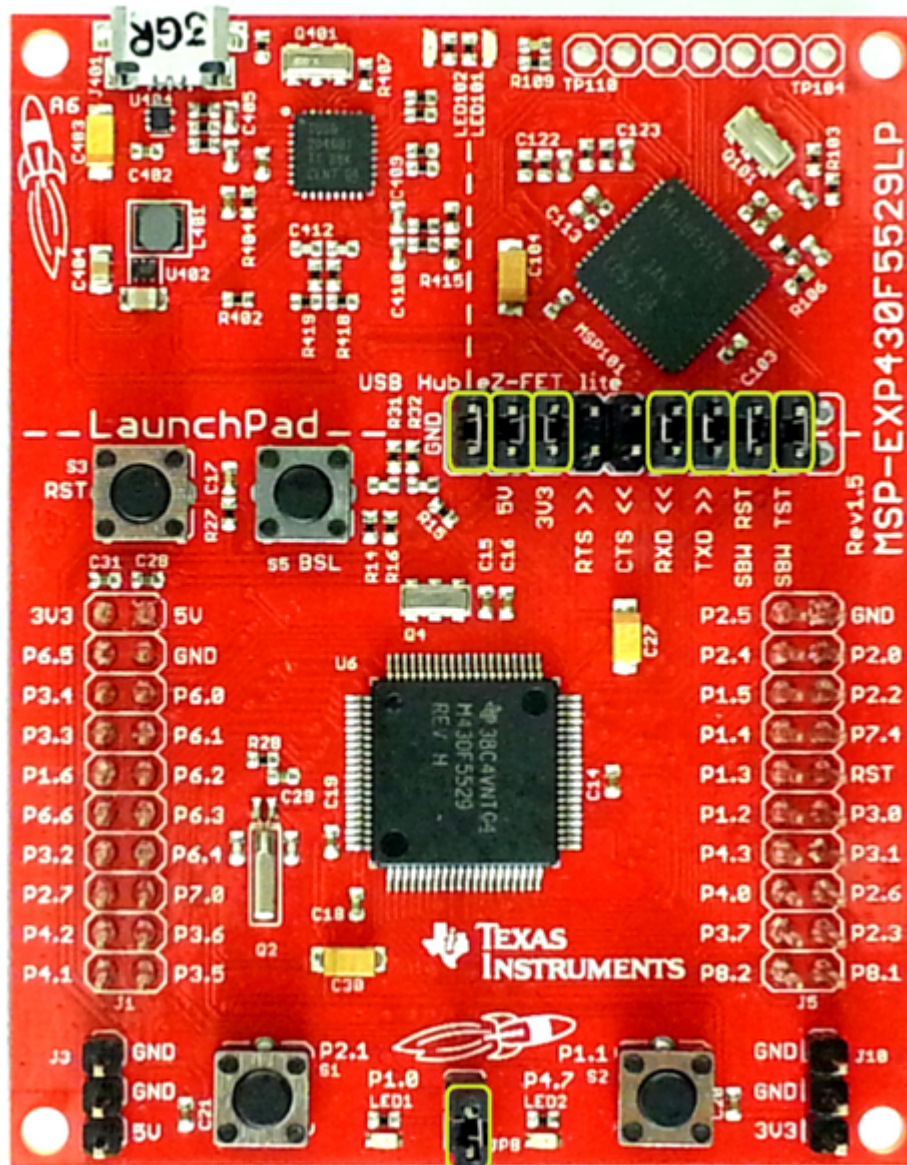


Figure 10. Jumpers on the MSP430F5529 Launchpad

3. Connect the CC3100BOOST to the MSP430F5529 Launchpad as shown in Figure 11.

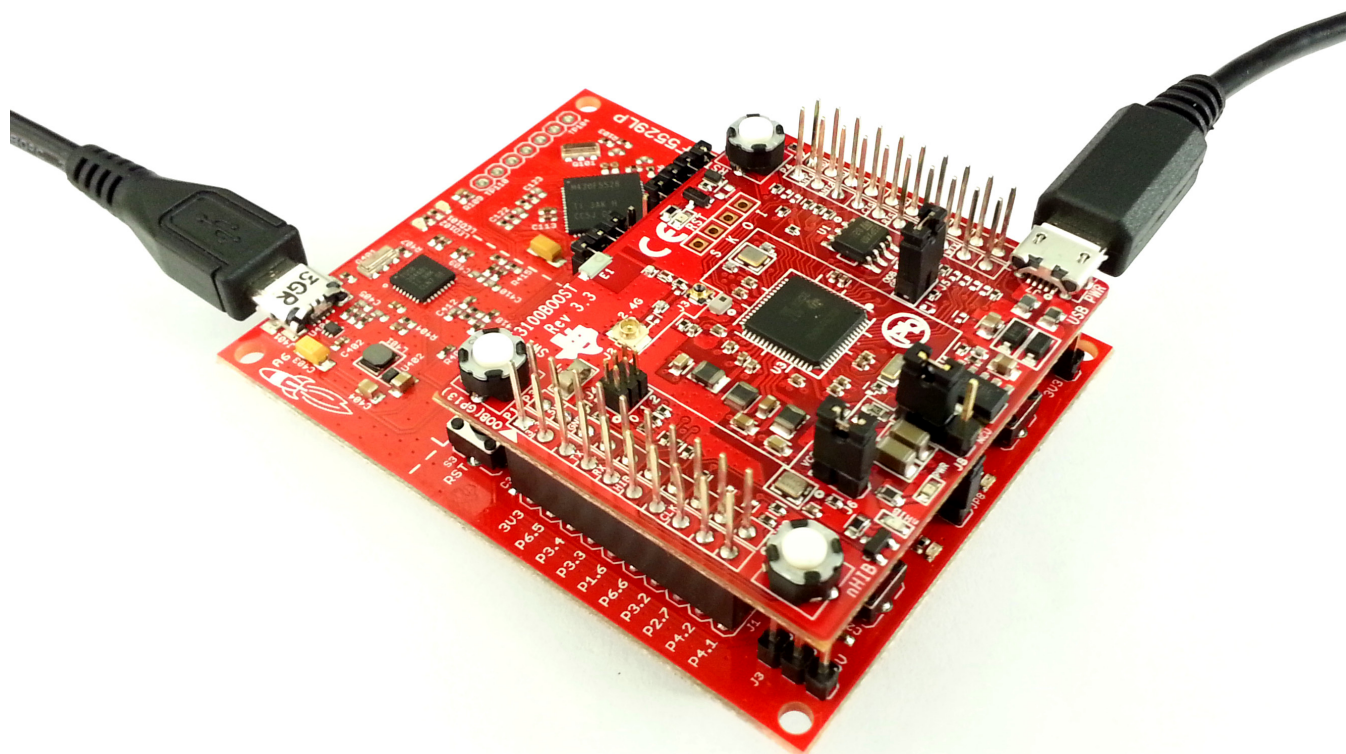


Figure 11. Connect the CC3100BOOST to the MSP430F5529 Launchpad

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.

5.2 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\msp430f5529\p\example_project_ccs`.
4. Check the project *getting_started_with_wlan_station* and press Finish.

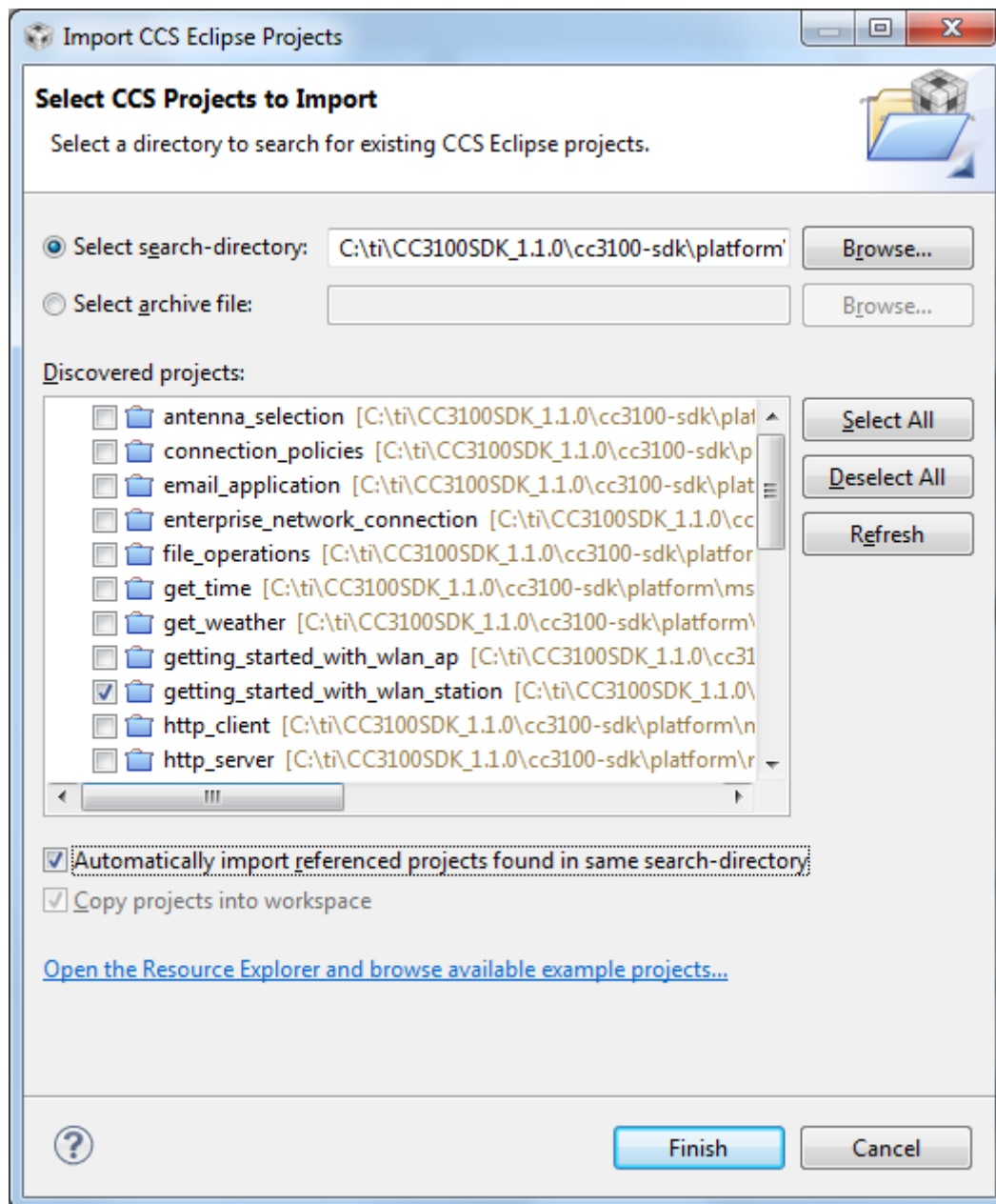


Figure 12. Select CCS Projects to Import

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 13](#). 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`.

```

* Values for below macros shall be modified per the access-point's
* SimpleLink device will connect to following AP when the applicati
*/
#define SSID_NAME      "<ap_name>" /* Access point name to
#define SEC_TYPE       SL_SEC_TYPE_OPEN /* Secur
#define PASSKEY        "" /* Password in case of s
#define PASSKEY_LEN    my_strlen(PASSKEY) /* Password length in ca
*/
* Values for below macros shall be modified per the
* SimpleLink device will connect to following AP whe
*/
#define SSID_NAME      "Your_AP_Name_Here" /
#define SEC_TYPE       SL_SEC_TYPE_WPA /* Secur t
#define PASSKEY        "Your_AP_Security_Key_Here"
#define PASSKEY_LEN    my_strlen(PASSKEY) /* Passwo

```

Figure 13. 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 14. The baud rate should remain at 9600.
9. Press F11 on the CCS window to start debugging.

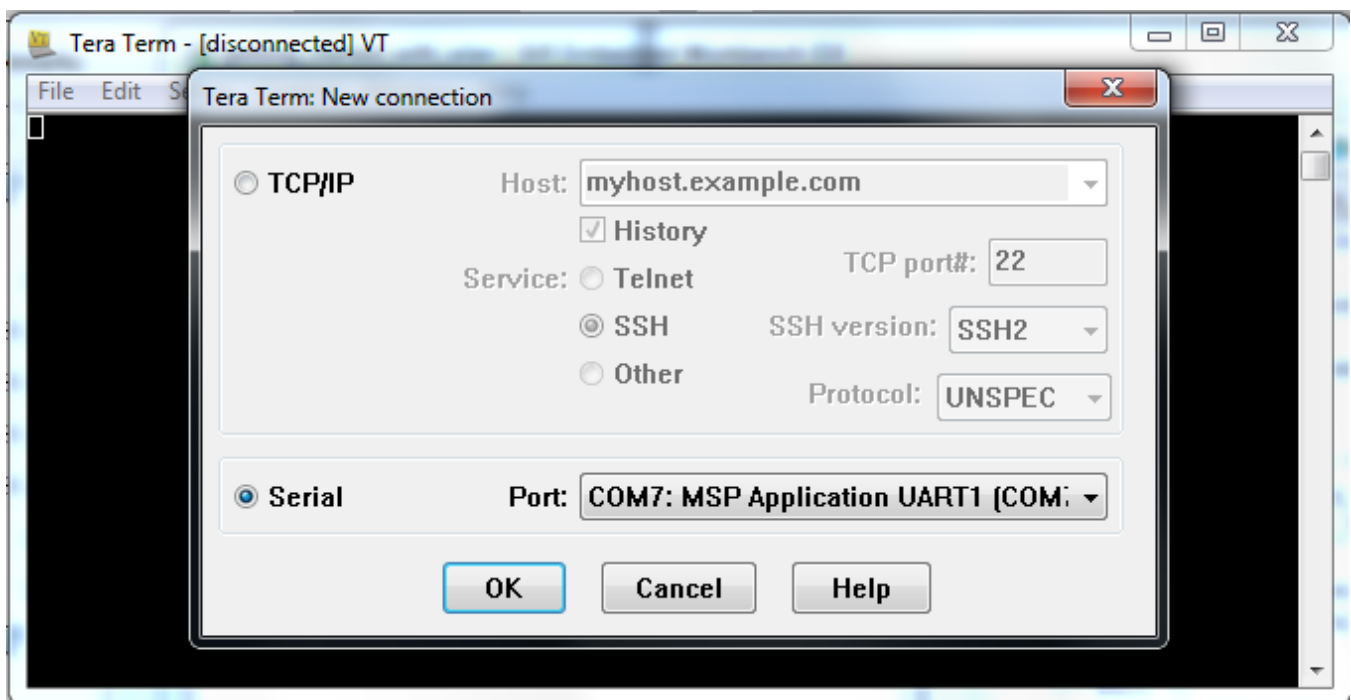


Figure 14. Launch Tera Term

10. If the CC3100 successfully completes all steps, the serial output appears as shown in Figure 15.

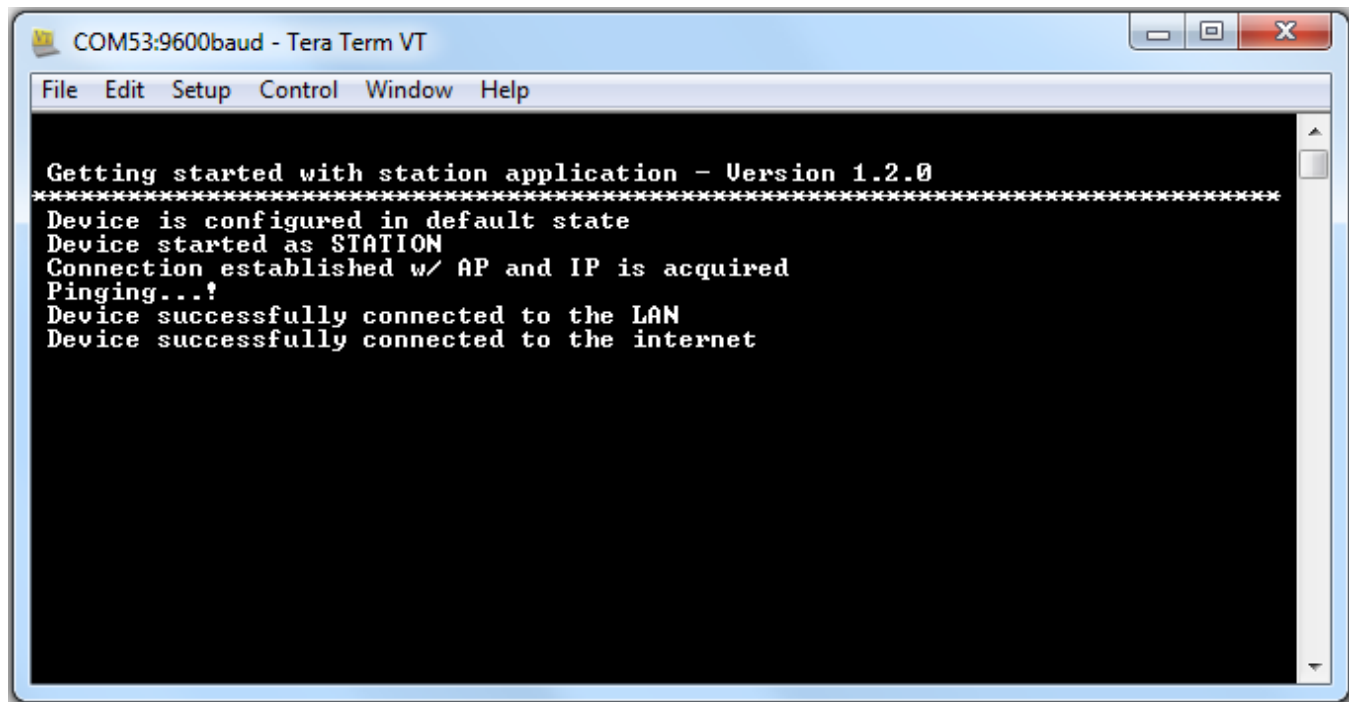


Figure 15. Tera Term VT

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\msp430f5529\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 16. 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.

```

/* Values for below macros shall be modified per the access-point's
 * SimpleLink device will connect to following AP when the applicati
 */
#define SSID_NAME      "<ap_name>"      /* Access point name to
#define SEC_TYPE       SL_SEC_TYPE_OPEN /* Security type of the
#define PASSKEY        ""               /* Password in case of s
#define PASSKEY_LEN    my_strlen(PASSKEY) /* Password length in ca

/* Values for below macros shall be modified per the
 * SimpleLink device will connect to following AP whe
 */
#define SSID_NAME      "Your_AP_Name_Here" /*
#define SEC_TYPE       SL_SEC_TYPE_WPA    /* Secur t
#define PASSKEY        "Your_AP_Security_Key_Here"
#define PASSKEY_LEN    my_strlen(PASSKEY) /* PASSWO

```

Figure 16. 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 17. The baud rate should remain at 9600.
8. After building is finished, select *Project>Download and Debug* from the menu to start debugging.

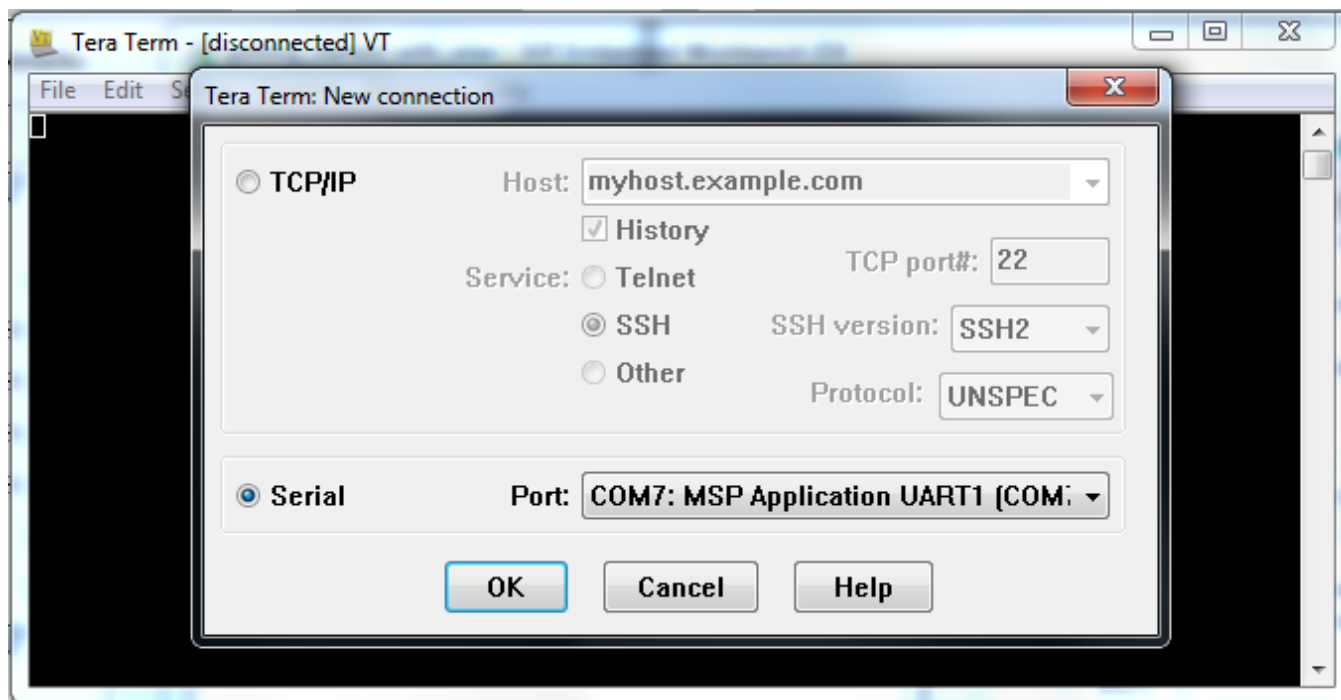


Figure 17. Launch Tera Term

9. If the CC3100 successfully completes all steps, the serial output appears as shown in [Figure 18](#).

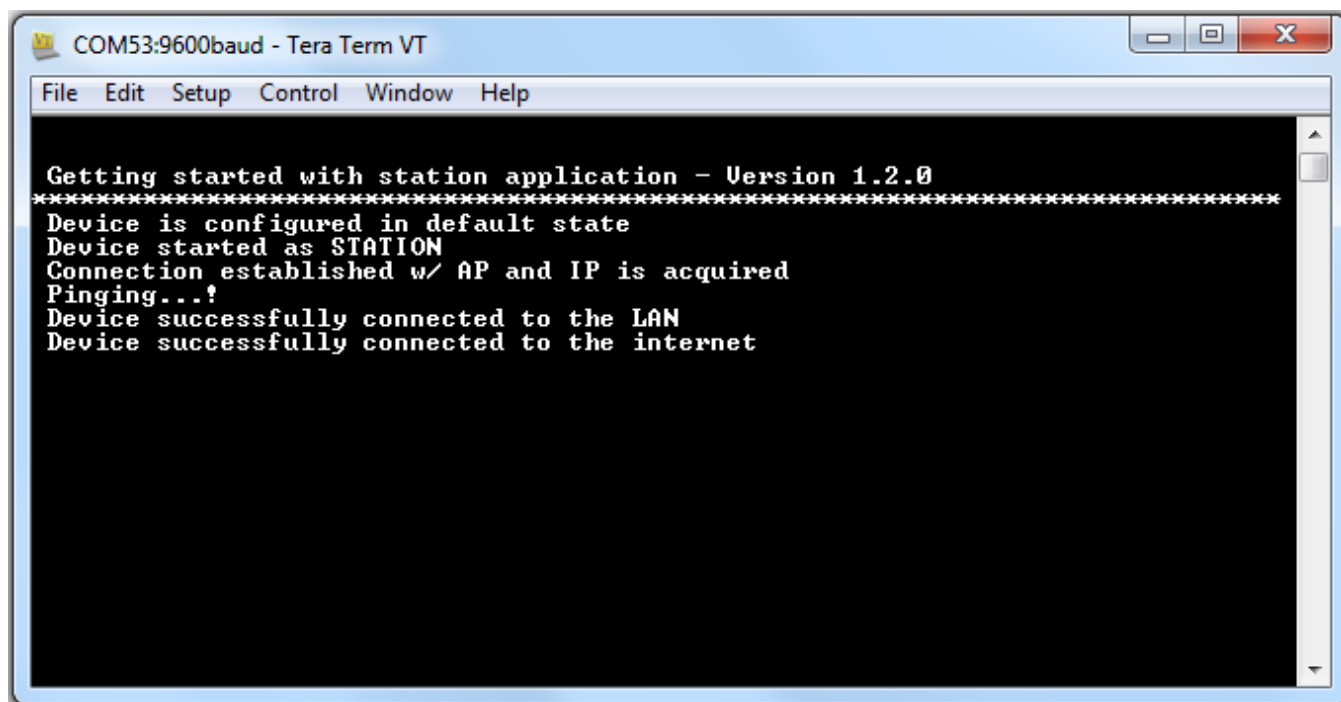


Figure 18. Tera Term VT

6 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.

7 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.

Changes from C Revision (March 2016) to D Revision	Page
• Updated Note.	8
• Added Eclipse Output Console image.	11

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), 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, 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 (www.ti.com/legal/termsofsale.html) 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.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2019, Texas Instruments Incorporated