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

User's Guide

TEXAS INSTRUMENTS

Literature Number: SWRU375
June 2014

Contents

List of Figures

User's Guide
SWRU375–June 2014

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.

SimpleLink is a trademark of Texas Instruments.
Wi-Fi is a registered trademark of Wi-Fi Alliance.

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\*.

3 Getting Started with SimpleLink Studio

3.1 Configure Boards

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

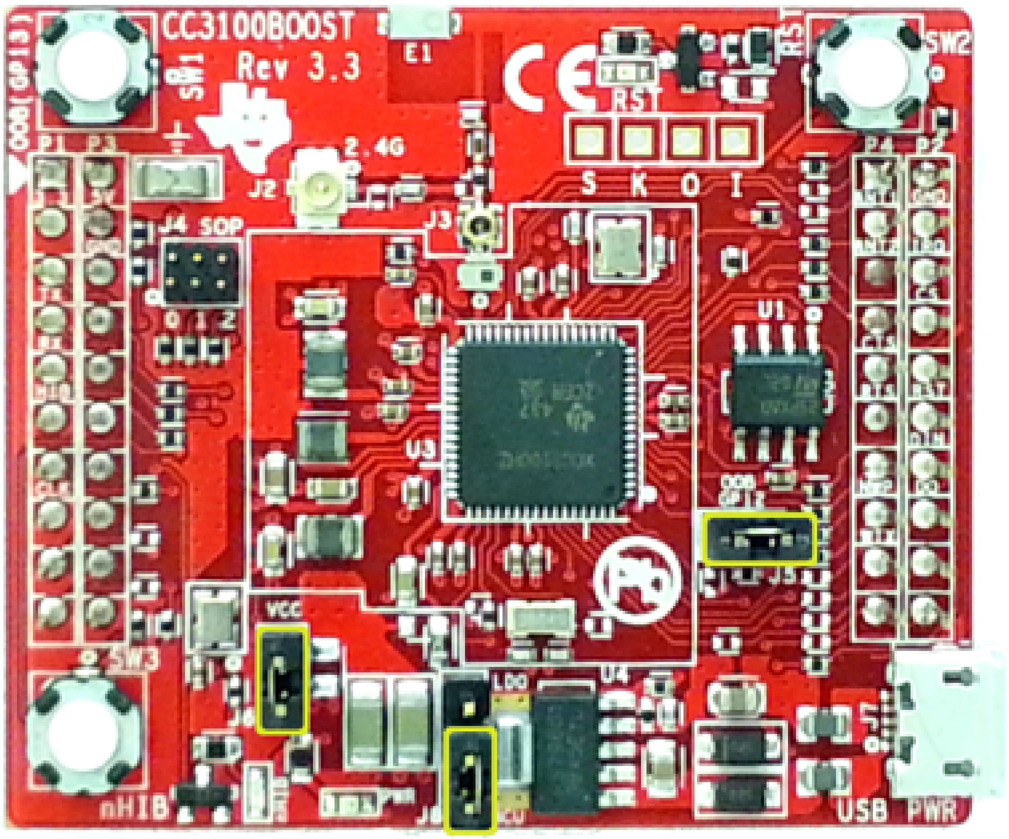


Figure 1. CC3100BOOST

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

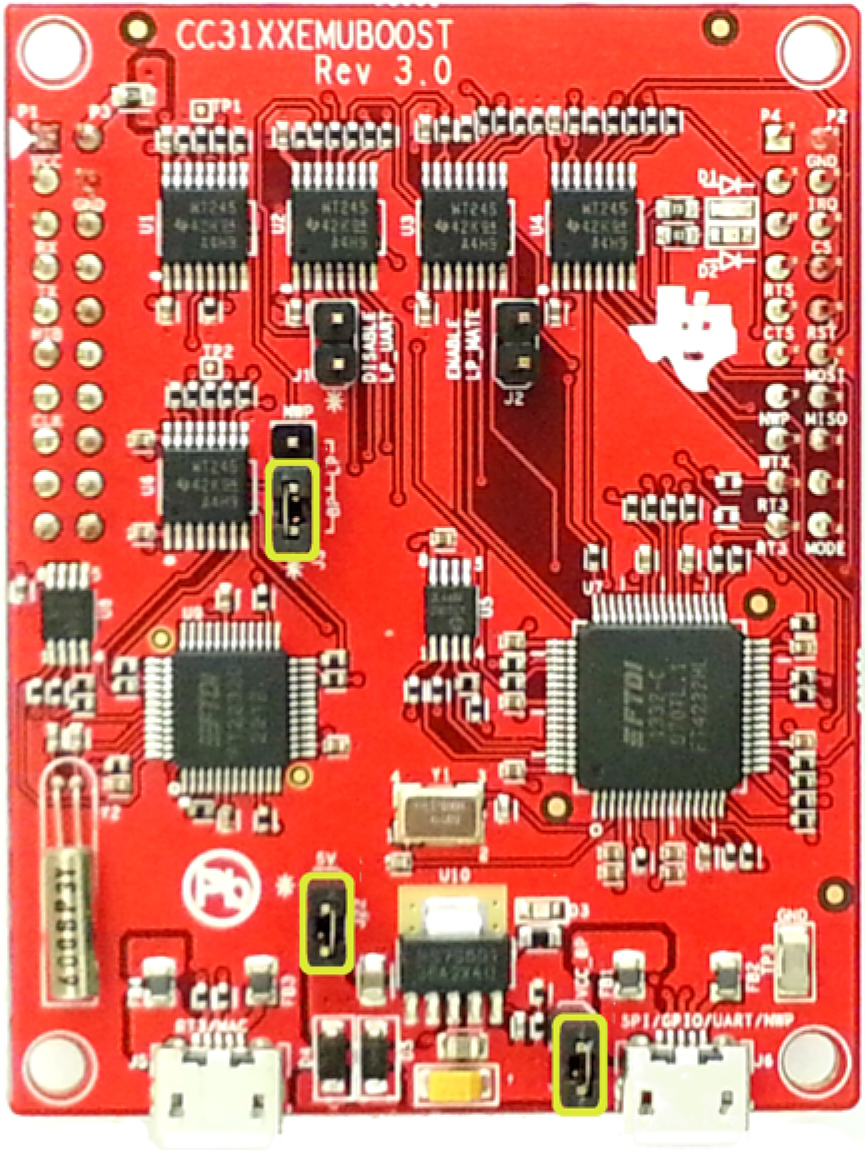


Figure 2. CC31XXEMUBOOST

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

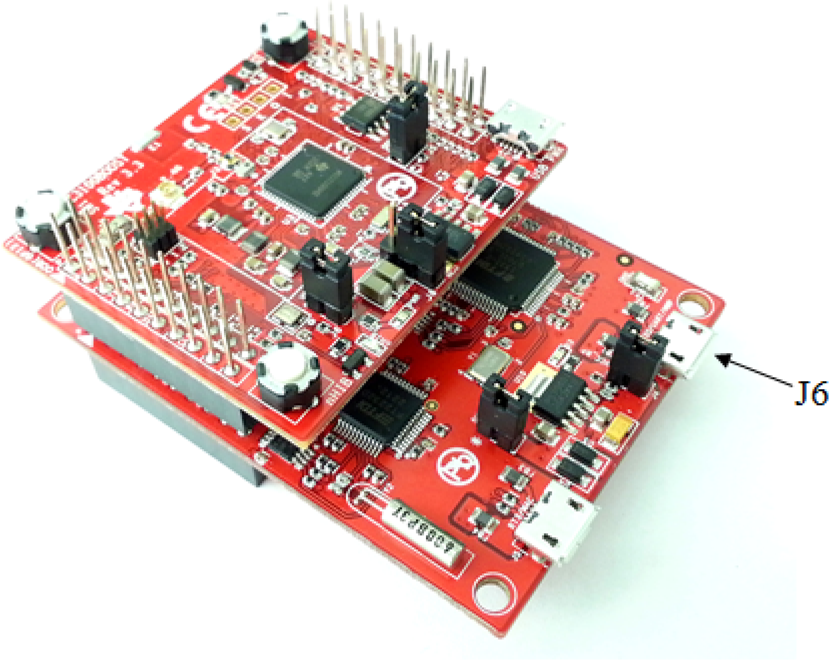
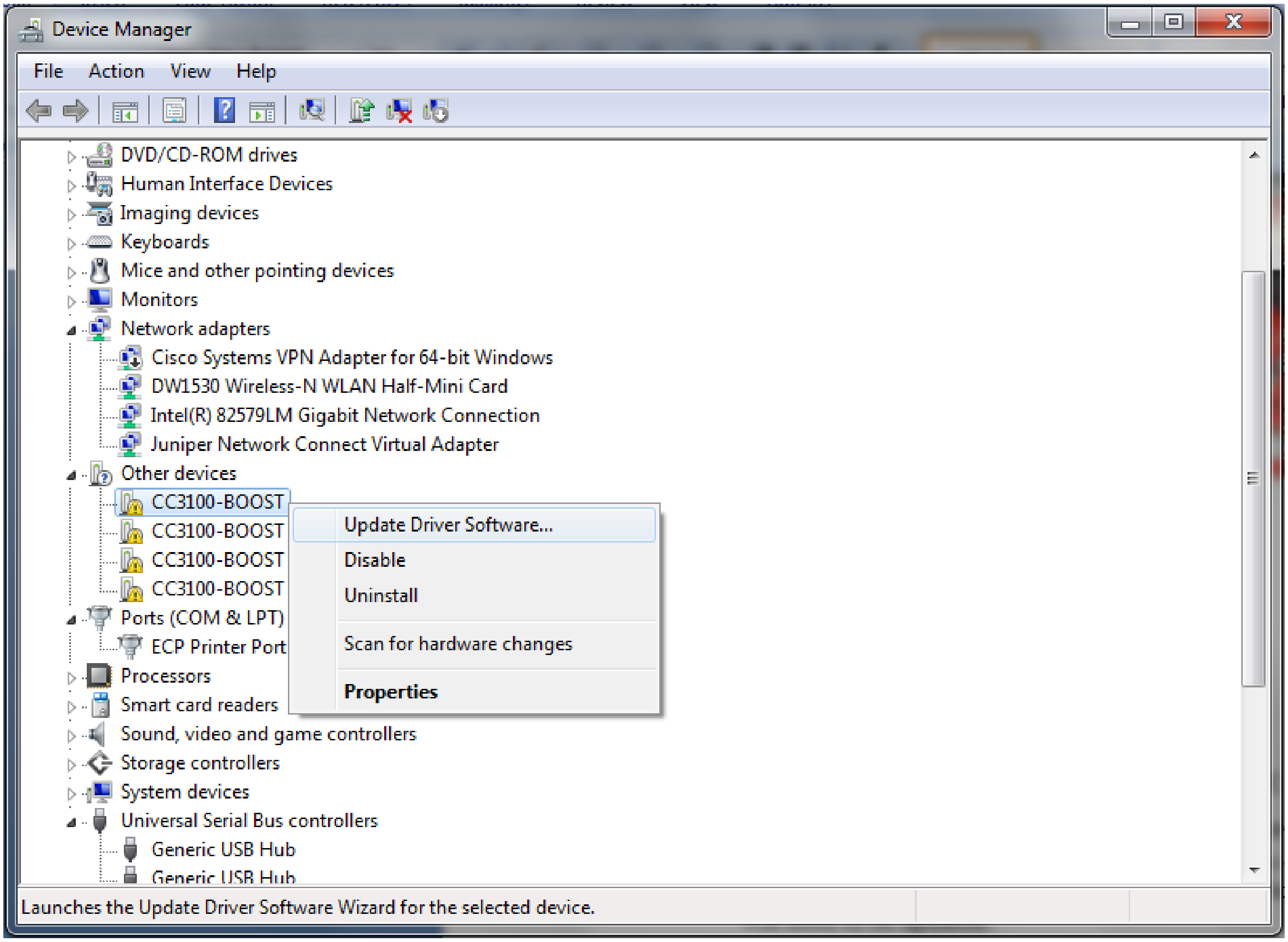


Figure 3. Connect the CC3100BOOST to the CC31XXEMUBOOST

3.2 Install USB Driver

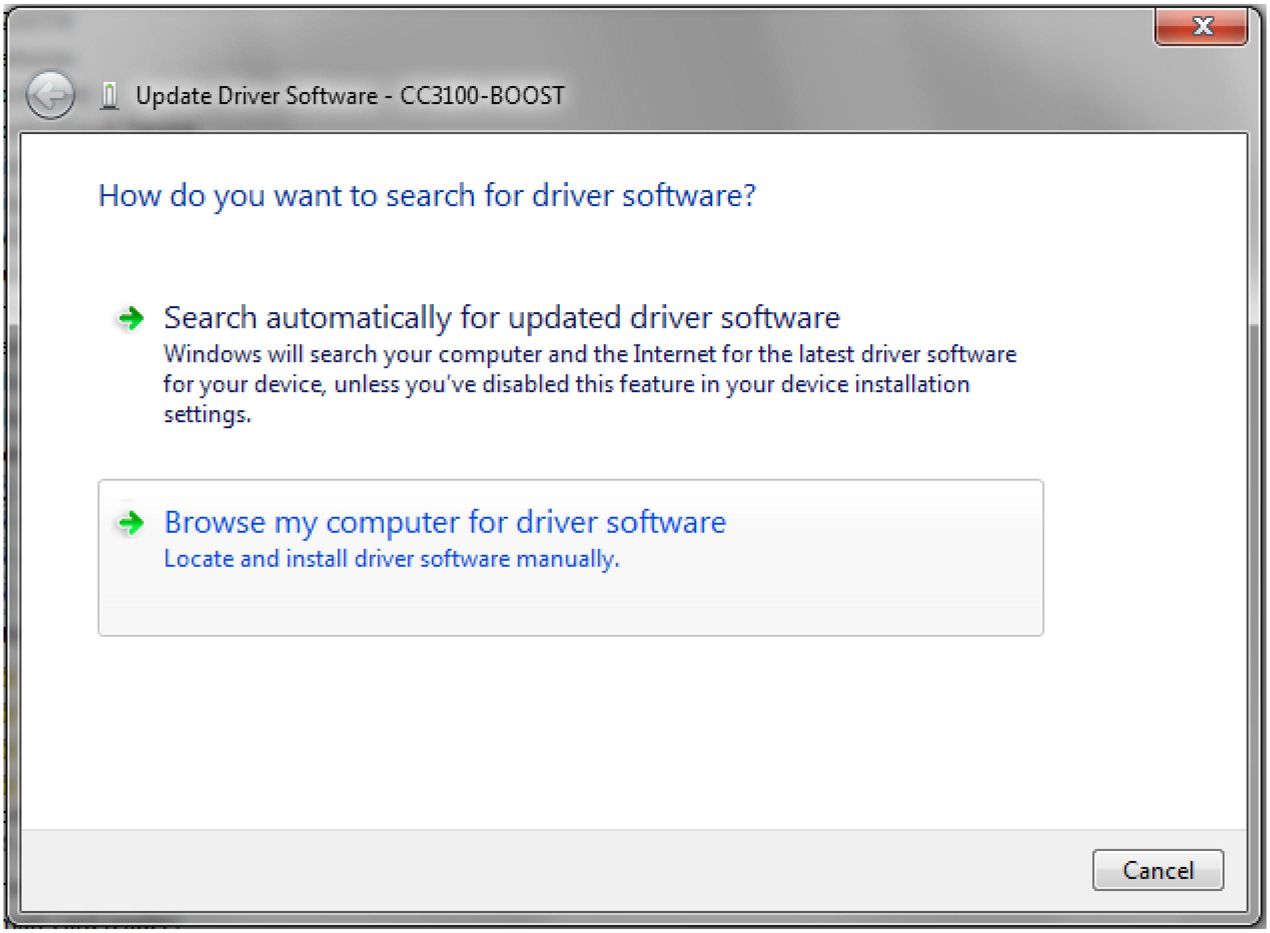
1. Connect the J6 port CC31XXEMUBOOST to the PC using the provided micro-USB cable.
2. Open Windows Device Manager by selecting *Start Menu>Control Panel>Device Manager*. The CC3100BOOST will appear as four instances of “CC3100-BOOST” under the category *Other Devices* as shown in Figure 4. For all of these instances, the driver software will need to be updated.



Launches the Update Driver Software Wizard for the selected device.

Figure 4. Device Manager

3. Right click on the first instance of “CC3100-BOOST” and select “Update Driver Software…”
4. Select “Browse my computer for driver software”.



Update Driver Software - CC3100-BOOST

How do you want to search for driver software?

Search automatically for updated driver software

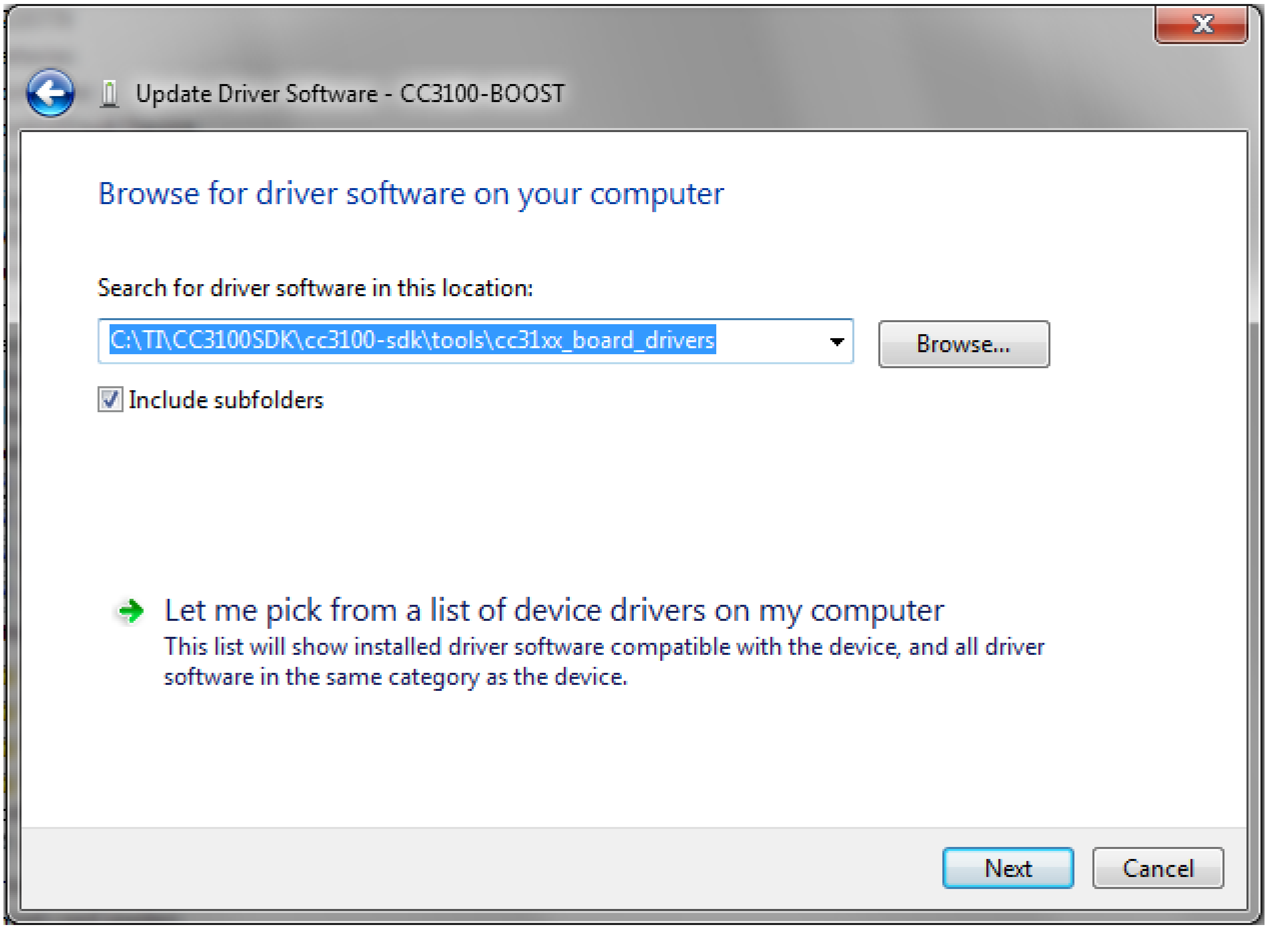
Windows will search your computer and the Internet for the latest driver software for your device, unless you've disabled this feature in your device installation settings.

Browse my computer for driver software

Locate and install driver software manually.

Figure 5. Update Driver Software

5. Fill the search path as *C:\TI\CC3100SDK\cc3100-sdk\tools\cc31xx_board_drivers*, and press Next. There is no need to restart the PC.



Update Driver Software - CC3100-BOOST

Browse for driver software on your computer

Search for driver software in this location:

C:\TI\CC3100SDK\cc3100-sdk\tools\cc31xx_board_drivers

☑ Include subfolders

Browse...
Let me pick from a list of device drivers on my computer
This list will show installed driver software compatible with the device, and all driver software in the same category as the device.
Next
Cancel

Figure 6. Update Driver Software

6. Repeat the three steps above for each of the three other instances of “CC3100-BOOST.”
7. Repeat the same steps for the four instances of “USB Serial Port” that should have appeared as shown in Figure 7.

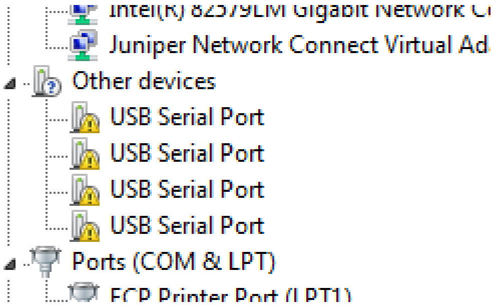


Figure 7. USB Serial Port

8. The CC3100BOOST will now be visible in the Device Manager as shown in Figure 8. The user may see two COM ports instead of four.

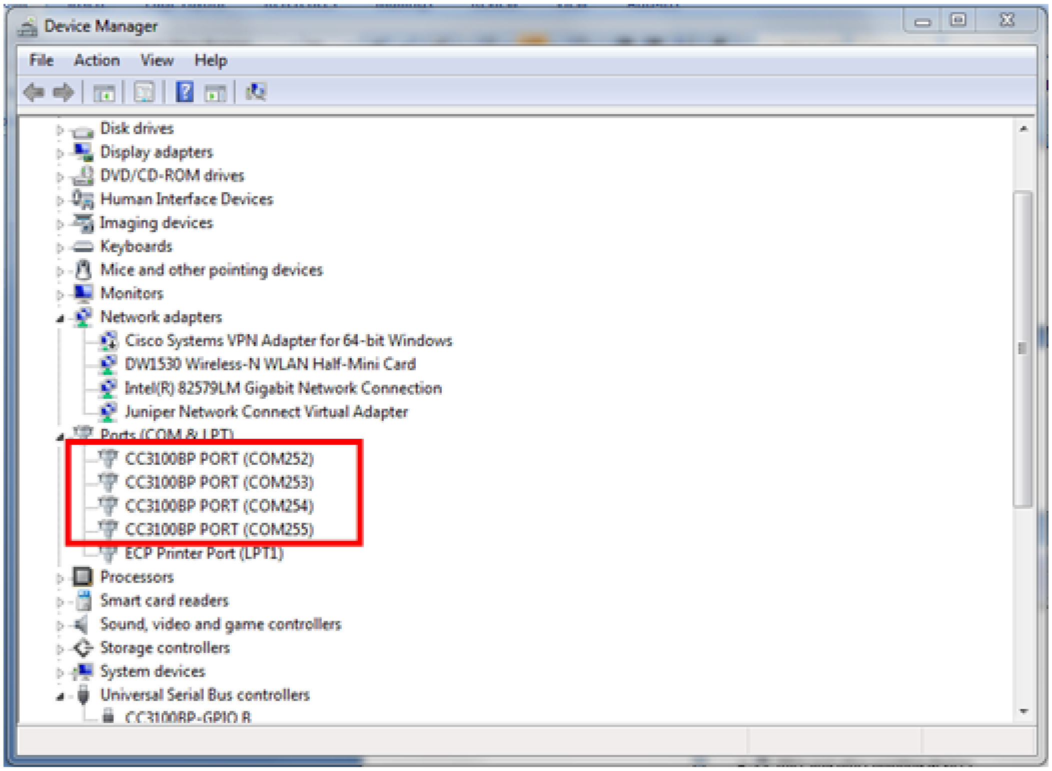


Figure 8. Device Manager

3.3 Run the Software

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

1. Prompts the user for the SSID of an AP to connect to.
2. Prompts the user for the security type.
3. Prompts the user for the password to the AP.
4. Attempts to acquire an IP address through DHCP.
5. Attempts to reach the internet.

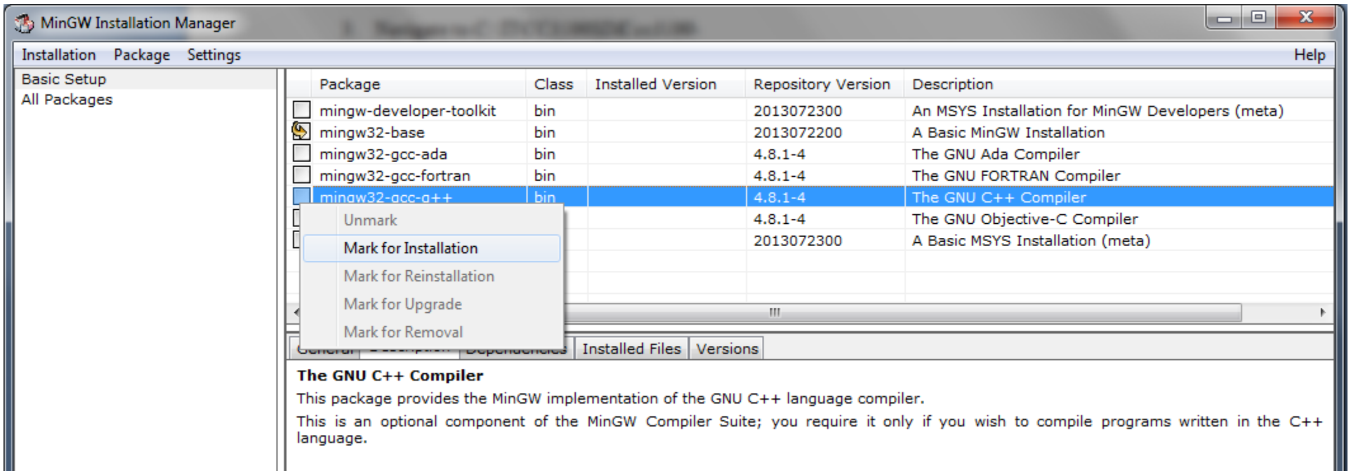
Option 1. Microsoft Visual Studio:

1. Download and install Microsoft Visual Studio Express (2010 or later) from http://www.microsoft.com/en-us/download/details.aspx?id=40787.
2. Open Microsoft Visual Studio Express, and select *File>Open>Project/Solution*.
3. Navigate to *C:\TI\CC3100SDK\cc3100-sdk\platform\simplelinkstudio\example_project_vs\getting_started_with_station*, and open *getting_started.sln*. Update the project file if using a Visual Studio version later than 2010.
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/.
2. Download and Extract Eclipse from http://www.eclipse.org/downloads/. Choose the Eclipse IDE for C/C++ Developers package.

3. Download and install MinGW from http://sourceforge.net/projects/mingw/files/latest/download?source=files. During installation, make sure you have the following configurations 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 9.



	Package	Class	Installed Version	Repository Version	Description
□	mingw-developer-toolkit	bin		2013072300	An MSYS Installation for MinGW Developers (meta)
☑	mingw32-base	bin		2013072200	A Basic MinGW Installation
□	mingw32-gcc-ada	bin		4.8.1-4	The GNU Ada Compiler
□	mingw32-gcc-fortran	bin		4.8.1-4	The GNU FORTRAN Compiler
□	mingw32-gcc-g++	bin		4.8.1-4	The GNU C++ Compiler
□				4.8.1-4	The GNU Objective-C Compiler
□				2013072300	A Basic MSYS Installation (meta)

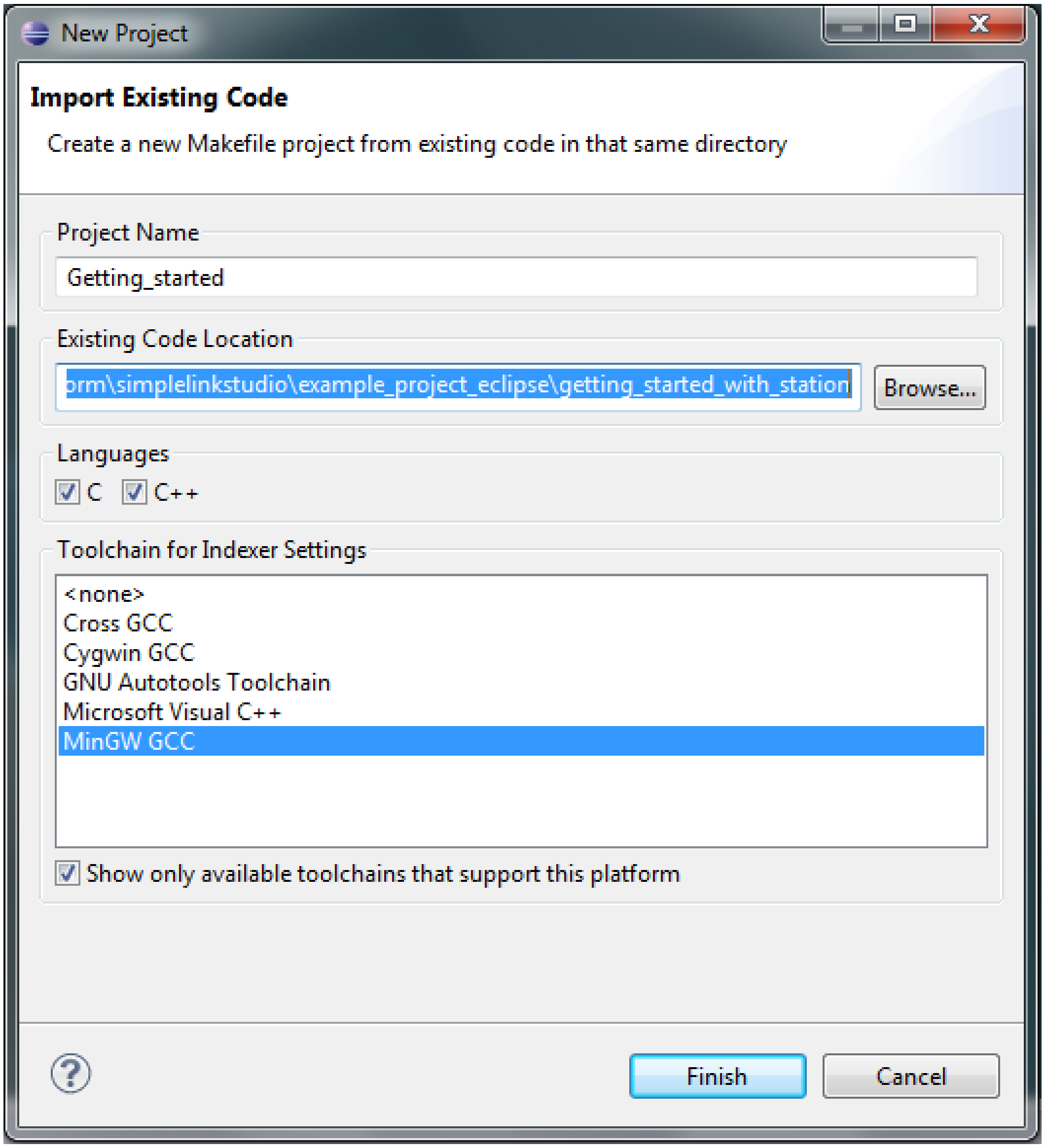
The GNU C++ Compiler

This package provides the MinGW implementation of the GNU C++ language compiler.

This is an optional component of the MinGW Compiler Suite; you require it only if you wish to compile programs written in the C++ language.

Figure 9. 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 its 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\cc3100-sdk\platform\simplelinkstudio\example_project_eclipse\getting_started_with_station*.
10. For Toolchain for Indexer Settings, choose “MinGW GCC” as shown in Figure 10.



New Project

Import Existing Code

Create a new Makefile project from existing code in that same directory

Project Name

Getting_started

Existing Code Location

orm\simplelinkstudio\example_project_eclipse\getting_started_with_station

Languages

☑ C ☑ C++

Toolchain for Indexer Settings

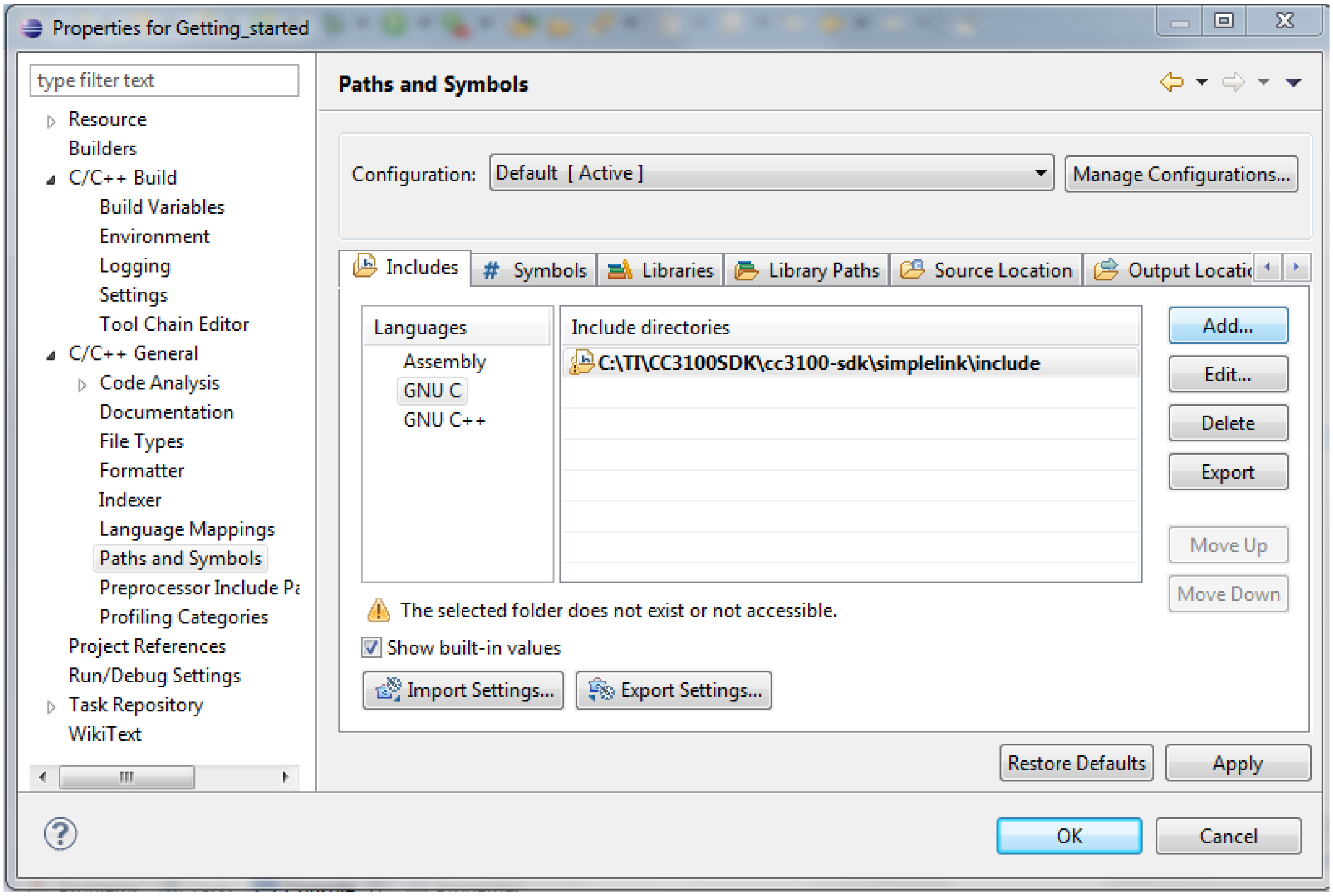
<none>
Cross GCC
Cygwin GCC
GNU Autotools Toolchain
Microsoft Visual C++
MinGW GCC

☑ Show only available toolchains that support this platform

Figure 10. 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. Click the **C/C++ Build** menu and do the following:
    (a) Uncheck **Use the default build command**.
    (b) Type *mingw32-make -f Makefile* in **Build command**.
    (c) Uncheck **Generate Makefiles automatically**.
    (d) Set the Build Directory as: *${workspace_loc:/Getting_started}*
16. Expand the **C/C++ Build** menu, and select **Tool Chain Editor**. Under **Current Builder**, Select "Gnu Make Builder", then click Apply.
17. Select **Environment** in the **C/C++ Build** menu. Make sure 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\cc3100-sdk\simplelink\include*.



Configuration: Default [Active]

Include directories: C:\TI\CC3100SDK\cc3100-sdk\simplelink\include

The selected folder does not exist or not accessible.

☑ Show built-in values

Figure 11. Paths and Symbols

19. Press OK.
20. To fix a known bug in 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 12.

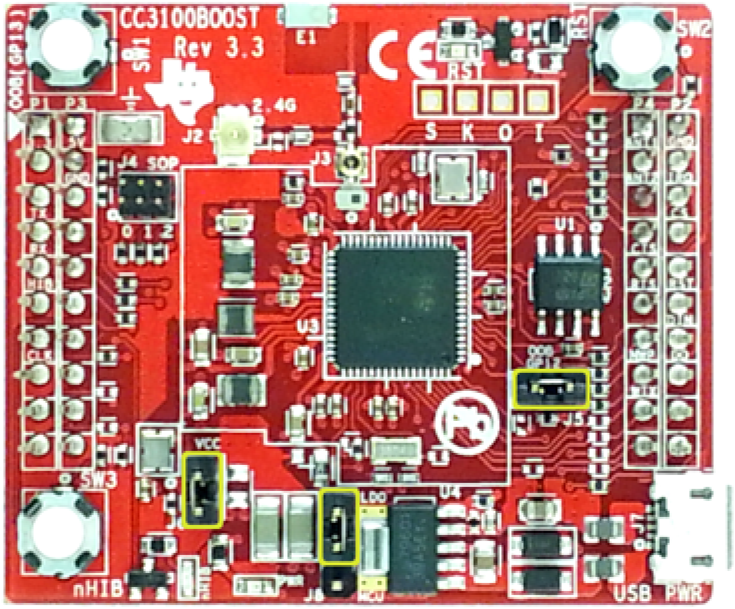


Figure 12. CC3100BOOST

2. The jumpers on the MSP430F5529 Launchpad should be connected as shown in Figure 13.

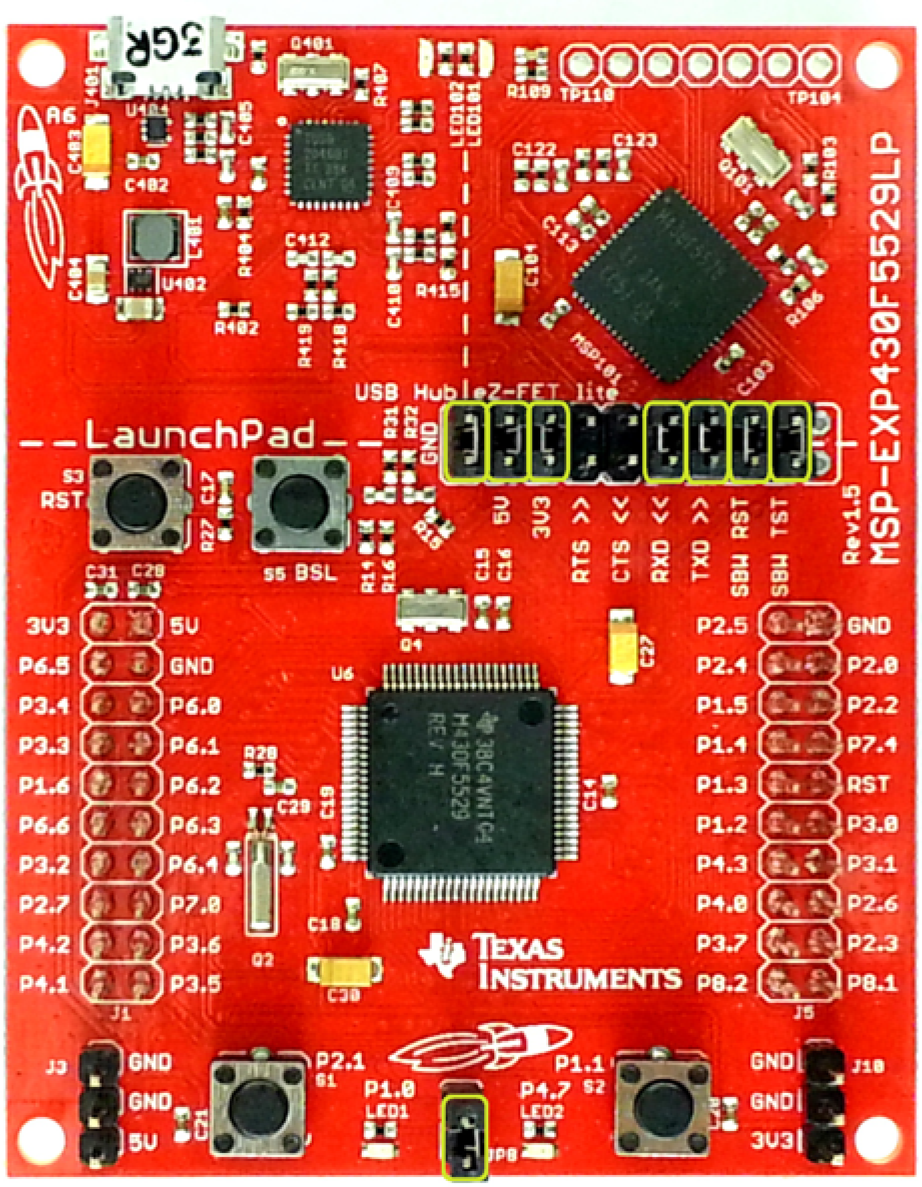


Figure 13. Jumpers on the MSP430F5529 Launchpad

3. Connect the CC3100BOOST to the MSP430F5529 Launchpad as shown in Figure 14.

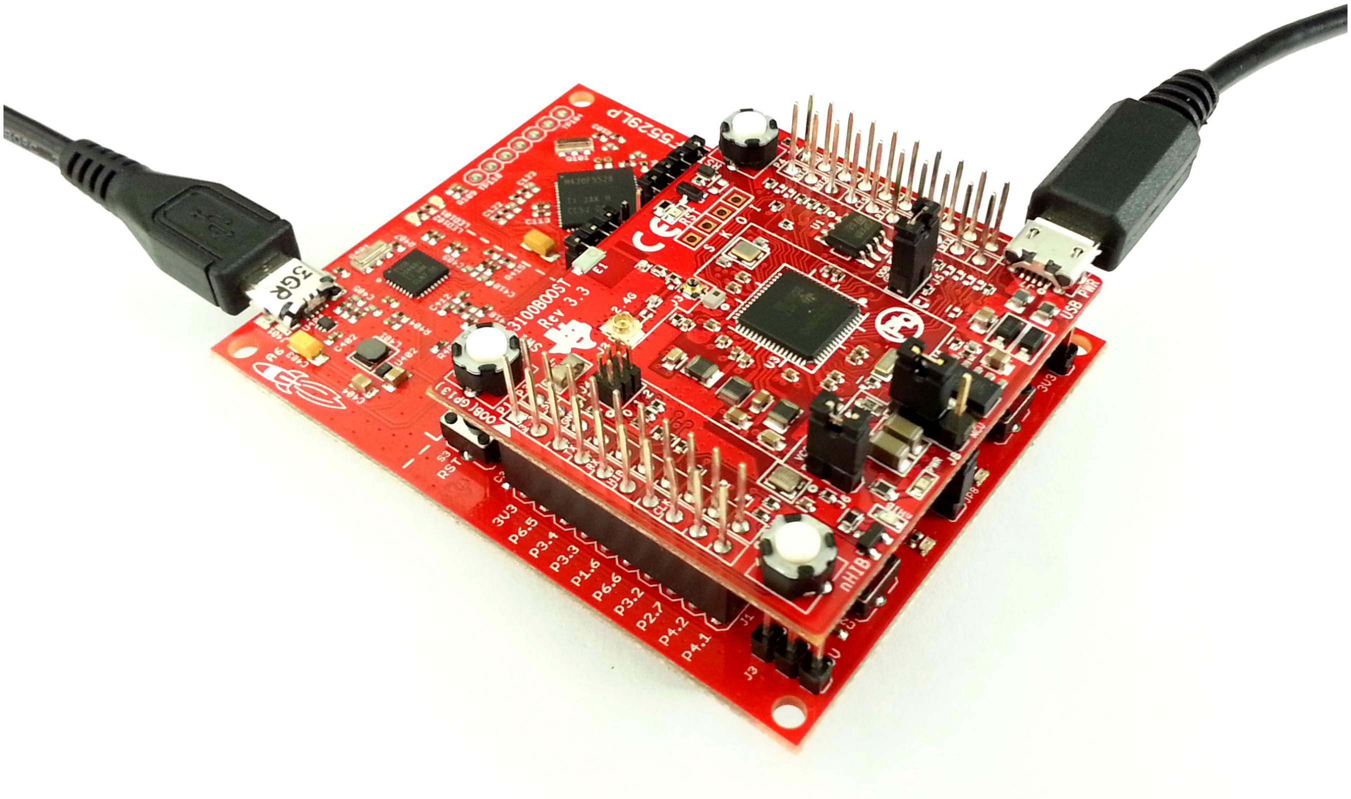


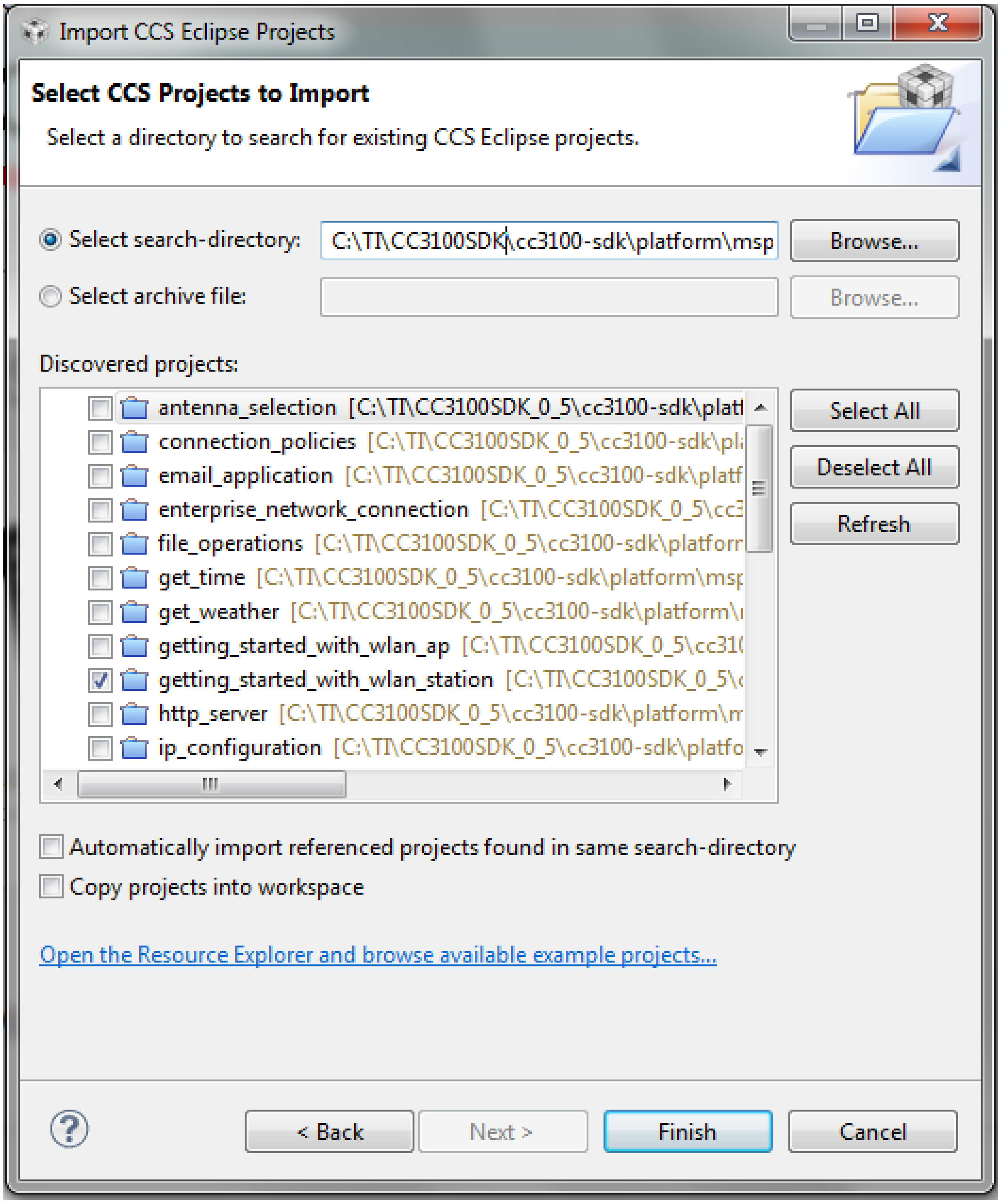
Figure 14. Connect the CC3100BOOST to the MSP430F5529 Launchpad

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

Option 1. Code Composer Studio (CCS):

1. Download and run the Code Composer Studio 6.0 (CCS) installation wizard (*ccs_setup_win32.exe*) from the TI website or from the CCS Wiki page. Must be **Version 6.0.0.00190** 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\cc3100-sdk\platform\msp430f5529lp*.
4. Check the project *getting_started_with_wlan_station* and press Finish.



Import CCS Eclipse Projects

Select CCS Projects to Import

Select a directory to search for existing CCS Eclipse projects.

● Select search-directory: C:\TI\CC3100SDK\cc3100-sdk\platform\msp

○ Select archive file:

Discovered projects:

- □ antenna_selection [C:\TI\CC3100SDK_0_5\cc3100-sdk\platf
- □ connection_policies [C:\TI\CC3100SDK_0_5\cc3100-sdk\pl
- □ email_application [C:\TI\CC3100SDK_0_5\cc3100-sdk\platf
- □ enterprise_network_connection [C:\TI\CC3100SDK_0_5\cc3
- □ file_operations [C:\TI\CC3100SDK_0_5\cc3100-sdk\platforr
- □ get_time [C:\TI\CC3100SDK_0_5\cc3100-sdk\platform\msp
- □ get_weather [C:\TI\CC3100SDK_0_5\cc3100-sdk\platform\
- □ getting_started_with_wlan_ap [C:\TI\CC3100SDK_0_5\cc31
- ☑ getting_started_with_wlan_station [C:\TI\CC3100SDK_0_5\
- □ http_server [C:\TI\CC3100SDK_0_5\cc3100-sdk\platform\m
- □ ip_configuration [C:\TI\CC3100SDK_0_5\cc3100-sdk\platfo

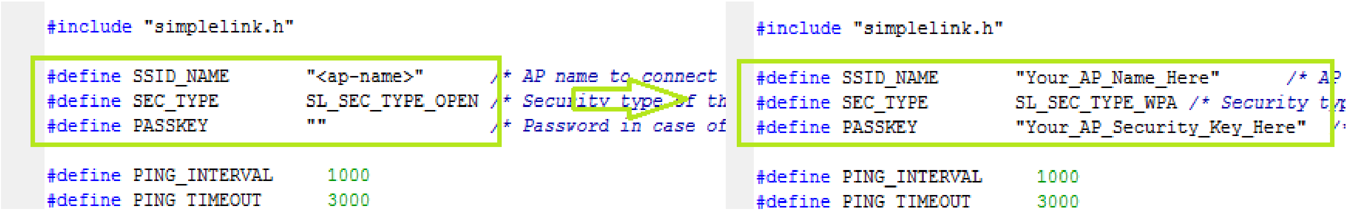
□ Automatically import referenced projects found in same search-directory

□ Copy projects into workspace

Open the Resource Explorer and browse available example projects...

Figure 15. Select CCS Projects to Import

5. Open the *main.c* file of the *getting_started_with_wlan project* for editing (*C:\TI\CC3100SDK\cc3100-sdk\examples\getting_started_with_wlan_station\main.c*).
6. Edit *main.c* to use the SSID, security type and security key of the Access Point being used. Edit the macros SSID_NAME, SEC_TYPE and PASSKEY to contain the Access Point's 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.



```
#include "simplelink.h"

#define SSID_NAME       "<ap-name>"      /* AP name to connect
#define SEC_TYPE        SL_SEC_TYPE_OPEN /* Security type of th
#define PASSKEY         ""               /* Password in case of

#define PING_INTERVAL    1000
#define PING_TIMEOUT     3000
```

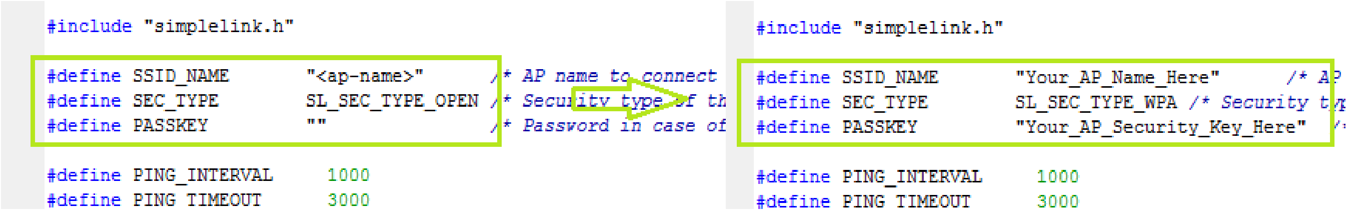
```
#include "simplelink.h"

#define SSID_NAME        "Your_AP_Name_Here"       /* AP
#define SEC_TYPE         SL_SEC_TYPE_WPA /* Security ty
#define PASSKEY          "Your_AP_Security_Key_Here"  /*

#define PING_INTERVAL    1000
#define PING_TIMEOUT     3000
```

Figure 16. Define SSID_Name

7. Select the *getting_started_with_wlan_station* project in Project Explorer and select *Project>Build Project* from the menu.
8. Press F11on the CCS window to start debugging.

Option 2. IAR Workbench

1. Install IAR Workbench for MSP430 version 6.10 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\cc3100-sdk\platform\msp430f5529lp\example_project_iar\getting_started_with_wlan_station\getting_started_with_wlan.eww*.
4. Open the *main.c* file of the *getting_started_with_wlan project* for editing (*C:\TI\CC3100SDK\cc3100-sdk\examples\getting_started_with_wlan_station\main.c*).
5. Edit *main.c* to use the SSID, security type and security key of the Access Point being used. Edit the macros SSID_NAME, SEC_TYPE and PASSKEY to contain the Access Point's information as shown in Figure Figure 17. 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.



```
#include "simplelink.h"

#define SSID_NAME       "<ap-name>"      /* AP name to connect
#define SEC_TYPE        SL_SEC_TYPE_OPEN /* Security type of th
#define PASSKEY         ""               /* Password in case of

#define PING_INTERVAL    1000
#define PING_TIMEOUT     3000
```

```
#include "simplelink.h"

#define SSID_NAME        "Your_AP_Name_Here"       /* AP
#define SEC_TYPE         SL_SEC_TYPE_WPA /* Security ty
#define PASSKEY          "Your_AP_Security_Key_Here"  /*

#define PING_INTERVAL    1000
#define PING_TIMEOUT     3000
```

Figure 17. Macro Definition of SSID_Name

6. Select *Project>Rebuild All* from the menu.
7. After building is finished, select *Project>Download and Debug* from the menu to start debugging.

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

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		

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265