User's Guide
SWRU473–February 2017

SimpleLink™ Wi-Fi® CC3220 Out-of-Box Application

This application demonstrates the out-of-box (OOB) experience with the CC3220 LaunchPad™ Development Kit from Texas Instruments™.

The CC3220 device is part of the SimpleLink™ microcontroller (MCU) platform which consists of Wi-Fi®, *Bluetooth*® low energy, Sub-1 GHz and host MCUs, which 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 enables you to add any combination of the portfolio's devices into your design, allowing 100 percent code reuse when your design requirements change. For more information, visit www.ti.com/simplelink.

Contents

Trademarks

1 Introduction

The CC3220 device is part of the SimpleLink™ microcontroller (MCU) platform which consists of Wi-Fi®, *Bluetooth*® low energy, Sub-1 GHz and host MCUs, which 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 enables you to add any combination of the portfolio's devices into your design, allowing 100 percent code reuse when your design requirements change. For more information, visit www.ti.com/simplelink.

This application demonstrates the out-of-box (OOB) experience with the CC3220 LaunchPad Development Kit from Texas Instruments. The following features are highlighted.

- Easy connection to the CC3220 LaunchPad:
  - Using the SimpleLink Wi-Fi Starter Pro application (available on iOS and Android™), users can use Access Point (AP) provisioning or SmartConfig™ provisioning for a fast CC3220 connection.
  - Configuring the device in AP mode gives users a direct connection to the CC3220 LaunchPad.

Once the device is provisioned and connected to an AP in station mode, the profile is stored on the local file system so that any reset to the CC3220 automatically connects it to the AP.

- Easy access to the CC3220 through its internal web server using either:
  - The SimpleLink Wi-Fi Starter Pro application
  - Any browser, web pages stored on the serial flash are loaded on the browser, to provide ease of use.

This feature demonstrates configuring and reading onboard sensors.

- Over-The-Air (OTA) updates that demonstrates an update of a full image. OTA service enables in-system updates of the MCU application, CC3220 firmware releases (Service Pack) made available by TI, and other vendor files. An update procedure executed in a full-system integrity fashion, such as failure to upgrade any image components, leads to rolling back to the previous valid version.

This guide is intended for two types of audiences:

- Basic level users – those who would like to program the OOB image as it is and experience it.
- Advanced level users – those who would like to build their own project. Section 9 is intended for these users.

This guide is structured chronologically starting with required downloads and installations, then building the setup, flashing the OOB image, and finally experiencing the demo.

2 Download and Installation

In an effort to make the out-of-box guide as self-contained as possible, all download and installation steps are described in detail in this document.

3 Installation for Basic Users

3.1 UniFlash

The UniFlash utility lets the developer download the application image, service pack, and other files on the serial flash of the CC3220 device. Follow these steps to install UniFlash.

1. Download UniFlash for CC3x20. If UniFlash is not available online, use the offline version (v4.x or later).
2. Run the installer by double clicking on it.
3. Read and accept the license agreement to proceed.
4. Choose the desired path to place the package (otherwise the default is chosen).
5. Proceed with the installation, and click the Finish button when done.

3.2 SimpleLink™ Wi-Fi® Starter Pro Mobile Application

This application can be downloaded and installed through the application store. It is available on both iOS™ and Android. Look for *SimpleLink Wi-Fi Starter Pro* (see Figure 1).

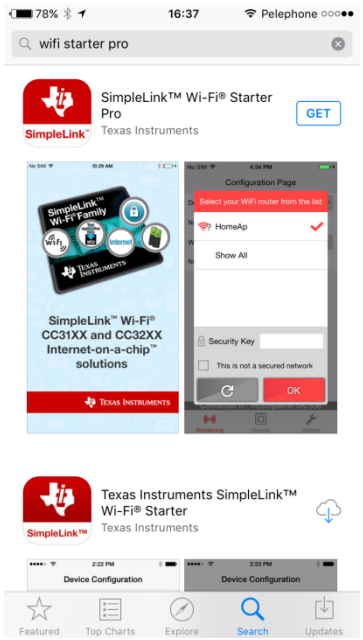

Figure 1. SimpleLink™ Wi-Fi Starter Pro

3.3 Serial Terminal

Many sample applications come with UART support for printing the debug information, or status of any operation. Some applications require user input through the UART, so TI advises installing a serial terminal application. Tera Term is used for demonstration here, but the Code Composer Terminal may also be used. Follow these steps to instal Tera Term:

1. Download Tera Term, and install as per the instructions.
2. Run the Tera Term application.
3. Select the Serial Port shown as XDS110 Class Application/User UART (see Figure 2).

   Be sure to install the XDS110 drivers for the PC to enumerate these ports for the serial terminal. Refer to Section 3.4 for installation of the XDS110 drivers.

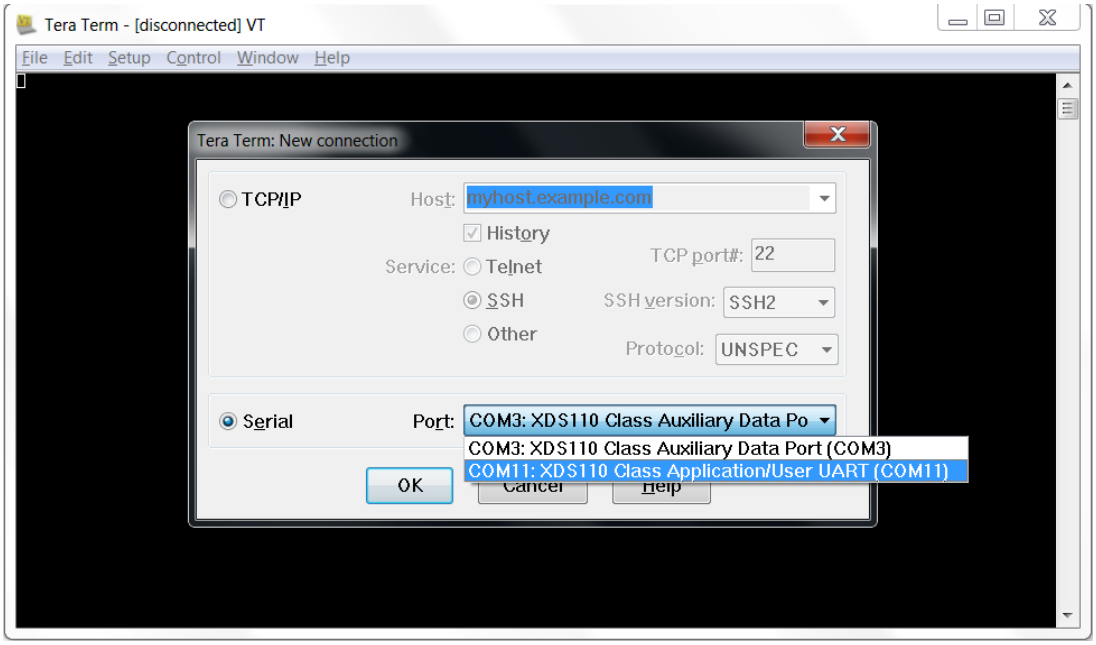

Figure 2. Tera Term New Connection Window

4. Go to Setup → Serial port (see Figure 3).

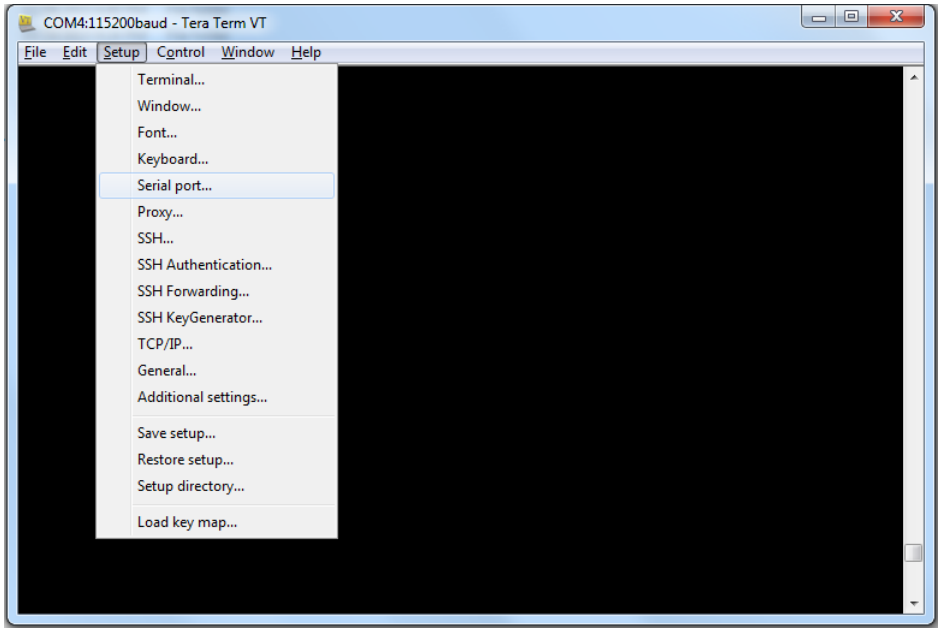

Figure 3. Tera Term Serial Port Tab

5. Configure the settings as per Figure 4.

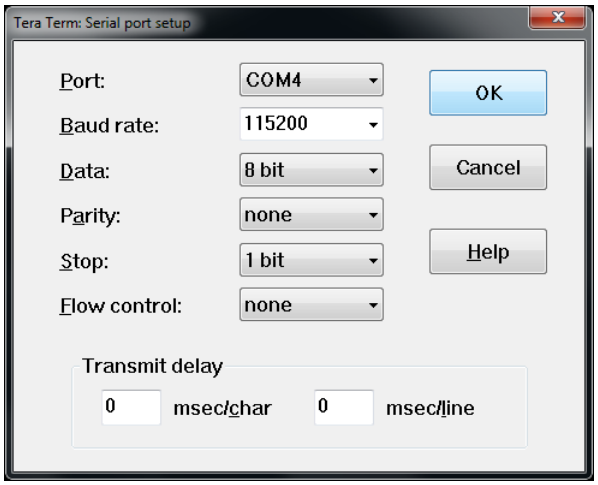

Figure 4. Tera Term Serial Port Settings

3.4 XDS110 Driver Installation

XDS110 drivers must be installed before using the debugger or image creator. The drivers also enumerate the serial terminal port, which can be used for printing the debug messages over the UART. The XDS drivers can be obtained through the XDS110 driver installation.

Follow these steps for the installation:

1. Run the installer in Administrator mode, and click the Next button (Figure 5).

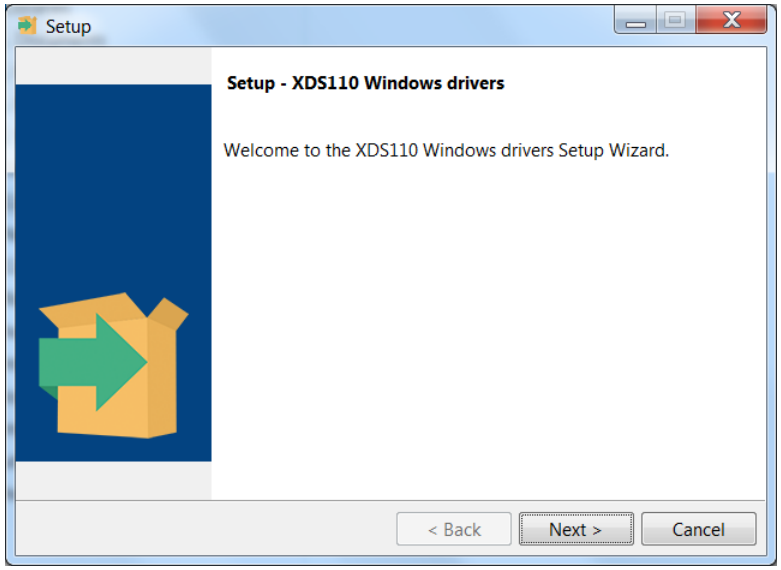

Figure 5. XDS110 Driver Setup Wizard

2. Read and accept the license agreement, and click the Next button (see Figure 6).

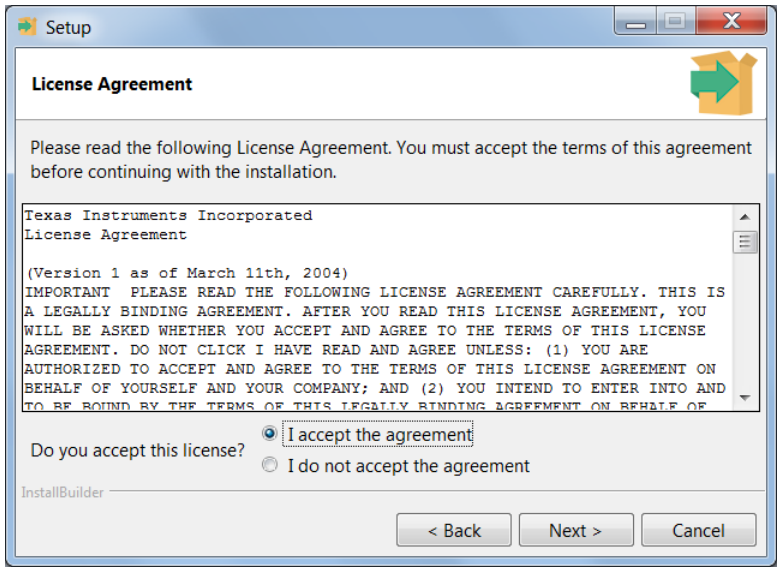

Figure 6. XDS110 Driver License Agreement

3. Specify the installation path (default is c:\ti\), and proceed with the installation (see Figure 7).

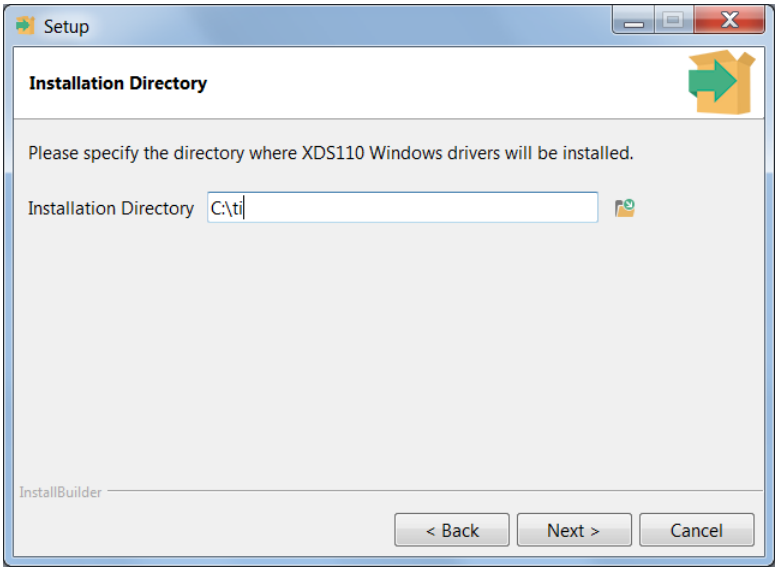

Figure 7. XDS110 Driver Installation Directory

4. Click the Finish button after the installation is done (Figure 8).

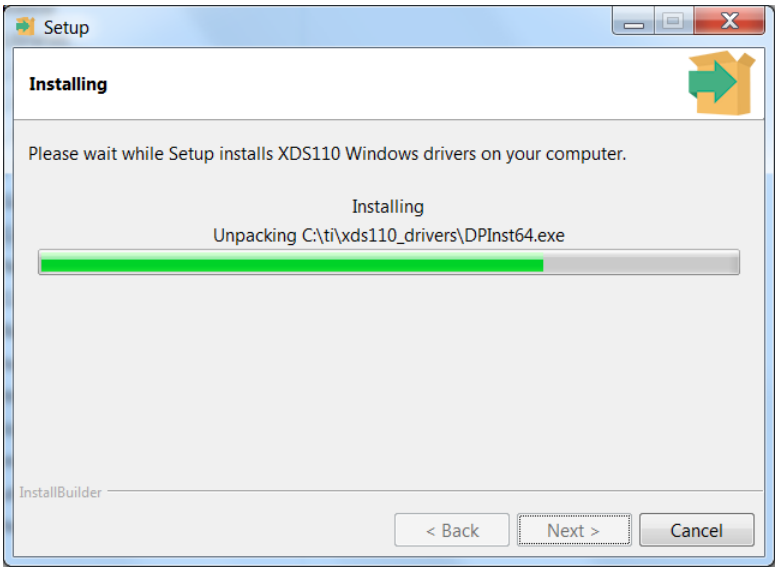

Figure 8. XDS110 Driver Installation Complete

4 Building the Setup

The following components are required.

- CC3220 LaunchPad flashed with the OOB package, and correctly set jumpers. See Figure 9 when powering from the USB.
- Mobile or tablet device with the SimpleLink Wi-Fi Starter Pro application installed. Alternatively, any mobile, tablet, or PC running a web browser can be used.
- Internet connection is required for the OTA procedure, because the software package is retrieved from the cloud.
- PC connected to the CC3220 LaunchPad if a terminal emulator is desired (it enables better visibility of the device state and debugging capabilities). In this setup, the PC powers the LaunchPad
- Local AP

To power up the CC3220 LaunchPad, connect it to the PC or any USB power supply. Out of the box, the CC3220 LaunchPad already contains the application, so there is no need to flash it, therefore the instructions in Section 5 are not required.

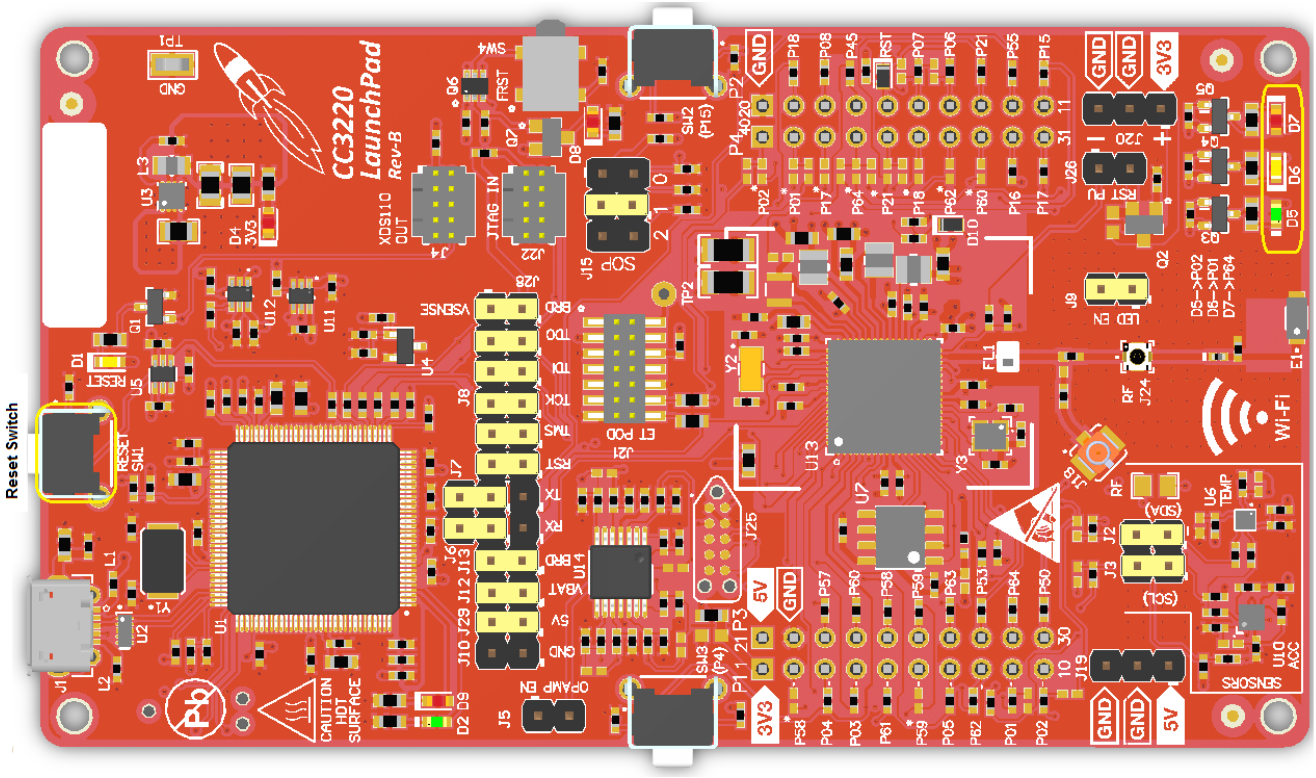

Figure 9. CC3220 Default Jumpers Setting

5 Flashing the Out-of-Box Project

The CC3220 LaunchPad comes with all the required content flashed to run the application. If the device has never been used and just came out of the box, the following steps are not necessary. If users have already flashed other SDK applications and need to reprogram the OOB content.

Because the OOB project is set in development mode (not production mode), the project matches a specific device MAC address. The image can be set to a new device MAC address through the GUI or through the command line (CLI). If users try to flash the image as is, the error shown in Figure 10 will appear.

Figure 10. UniFlash MAC Address Error

Using the GUI, perform the following steps.

1. Ensure the setup is built according to Section 4. Jumpers setting must be positioned as in Figure 9.
2. Ensure no other utility uses the XDS110 COM port. (1)
3. Open the UniFlash utility to program the OOB image.
4. Choose the CC3120/CC3220 device and then click the Start Image Creator button (see Figure 11).

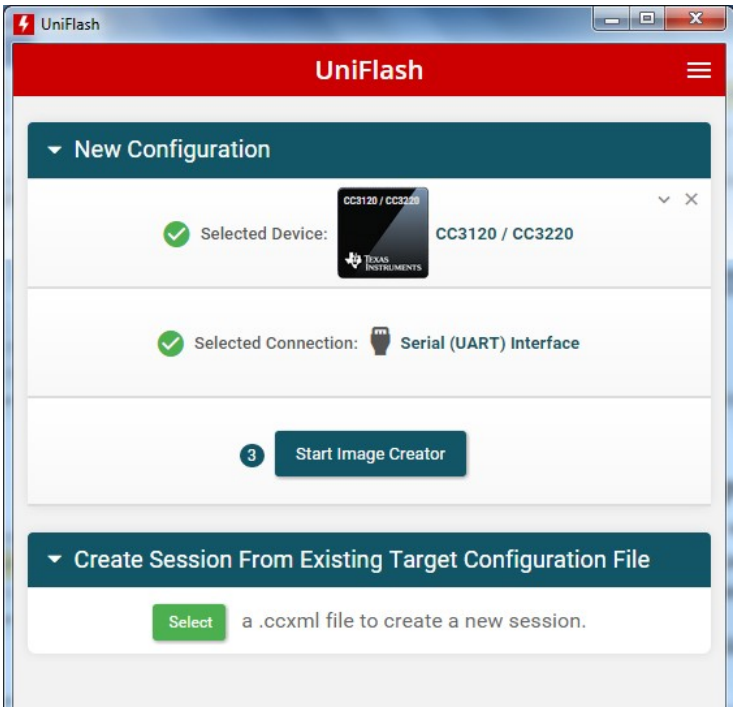

Figure 11. UniFlash – Choose CC3120 / CC3220 Device

(1) (1) Code Composer Studio (CCS) can create a conflict in the serial port preventing the communication with Image Creator. In this case, CCS should be closed.

5. Click the Manage Projects button and then click the Import project from ZIP file button. Locate the OOB image according to the connected CC3220 and the desired OS (freertos or tirtos). Already imported projects shall appear on the list of available projects.

   For CC3220S secured device:

   - <SDK install dir>\examples\rtos\CC3220S_LAUNCHXL\demos\out_of_box\uniflash\OOB_RS_tirtos.zip
   - <SDK install dir>\examples\rtos\CC3220S_LAUNCHXL\demos\out_of_box\uniflash\OOB_RS_freertos.zip

   For CC3220SF secured device:

   - <SDK install dir>\examples\rtos\CC3220SF_LAUNCHXL\demos\out_of_box\uniflash\OOB_SF_tirtos.zip
   - <SDK install dir>\examples\rtos\CC3220SF_LAUNCHXL\demos\out_of_box\uniflash\OOB_SF_freertos.zip

6. Navigate to General → Settings (see Figure 12) by using the left-side navigation panel and ensure that the image mode is set to Development.
7. Ensure the CC3220 LaunchPad is connected and click the *Connect* button. Users can see the device information of the connected device by using the MAC address; this is important because the development mode is matched to the MAC address of the device.

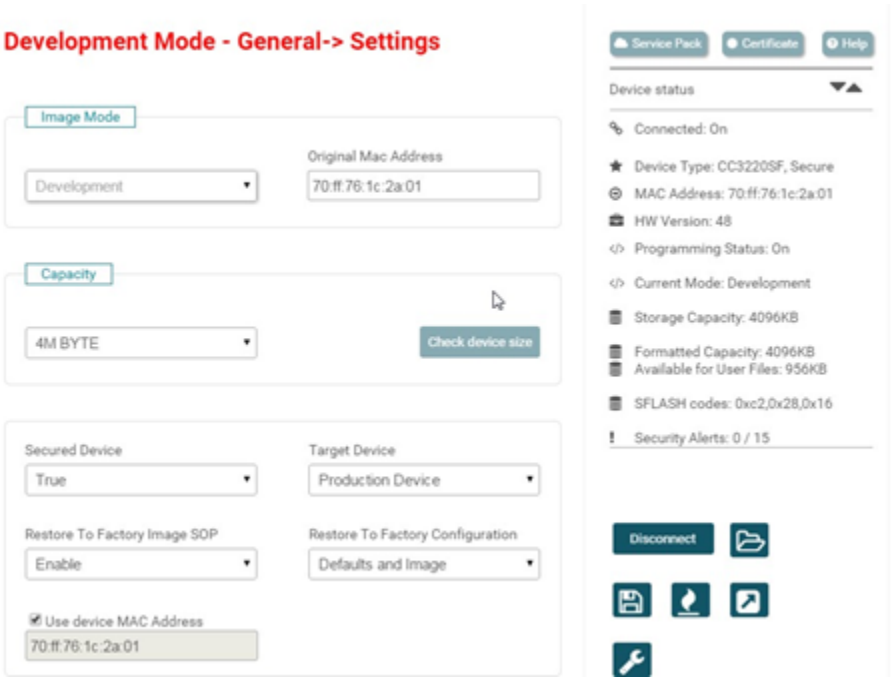

Figure 12. UniFlash General → Settings

8. Ensure the Use device MAC address option is selected (see Figure 12) and that the MAC address appear on the top.
9. Save the project and click on the Generate Image icon (see Figure 13).
10. Click the Program Image button (see Figure 13). The image is now programmed to the device.

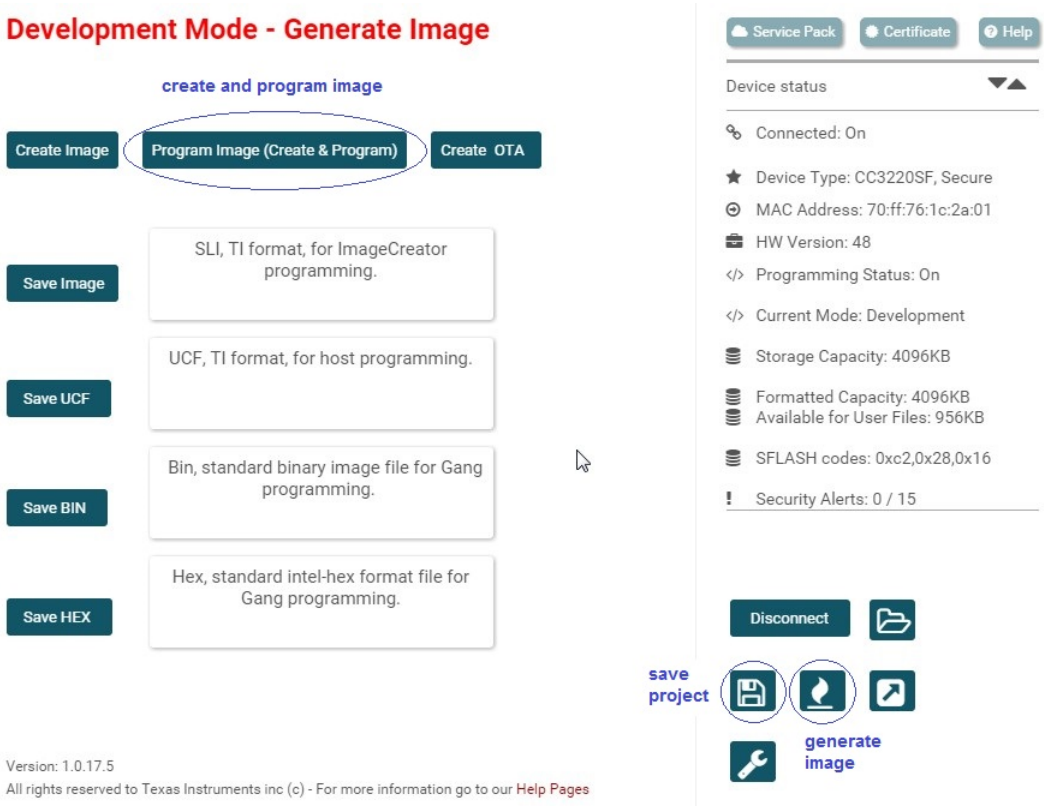

Figure 13. Program Image and Generate Image

If using the CLI, perform the following steps.

1. Import the OOB project (detailed in the previous steps 1 through 5). This step is required at least once so the project is imported into the UniFlash workspace.
2. Open a Windows command prompt and navigate to the UniFlash command line directory (UniFlash install directory where dslite.bat is located).
3. Invoke the following command.

   For the CC3220SF device:

   - dslite.bat --mode cc3220 project program --name OOB_SF --port com5 –dev

   For the CC3220R and CC3220S devices:

   (a) dslite.bat --mode cc3220 project program --name OOB_RS --port com5 –dev

NOTE: com5 is an example. Users must modify the COM port according to the assigned COM port.

Programming is considered successful is all progress bars indicate 100% and the command returns without errors. Users must reset the board at the end so the programmed image is executed.

6 Getting Started With the OOB Demonstration

6.1 Connecting to the CC3220

Connecting to the CC3220 can be achieved by either provisioning it, or by directly connecting when the CC3220 is set as an AP.

6.1.1 CC3220 Provisioning

Using the SimpleLink Wi-Fi Starter Pro application, users can start AP provisioning or SmartConfig provisioning for a fast CC3220 connection. During this procedure, the AP credentials are decoded by the CC3220, and a profile is stored on the serial flash for future connections.

By default, the OOB application starts in provisioning mode. Provisioning mode is reflected by a flashing red LED (D7) once every 2 seconds. The terminal can also be used for extra debug messages. The Waiting to be Provisioned message appears when the device is ready to be provisioned.

The provisioning procedure for SmartConfig is described as follows.

1. Open the application and navigate to the Settings tab. Ensure the Enable Smart Config setting is set to ON (Smart Config is the preferred provisioning method, see Figure 14).

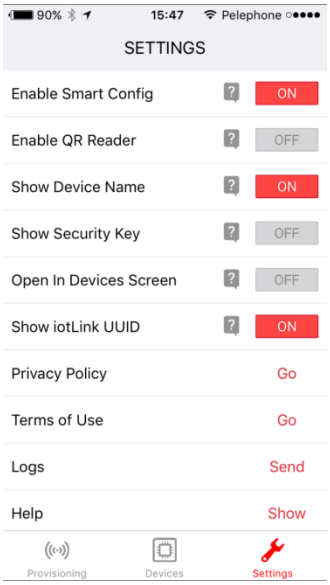

Figure 14. Wi-Fi Starter Settings Tab

2. Navigate to the provisioning tab to configure the AP credentials. Click on the START CONFIGURATION button to start the process (see Figure 15).

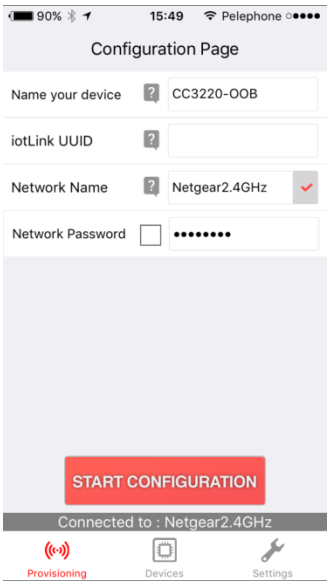

Figure 15. Wi-Fi Starter Provisioning Tab

3. Successful provisioning is indicated on the mobile application as shown in Figure 16, Figure 17, and Figure 18. On the LaunchPad, the red Led (D7) turns solid on.

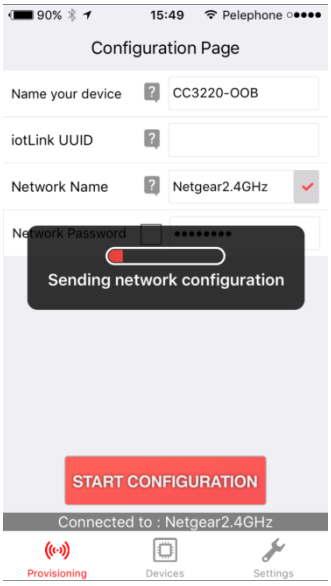

Figure 16. Wi-Fi Starter Provisioning Process (1/3)

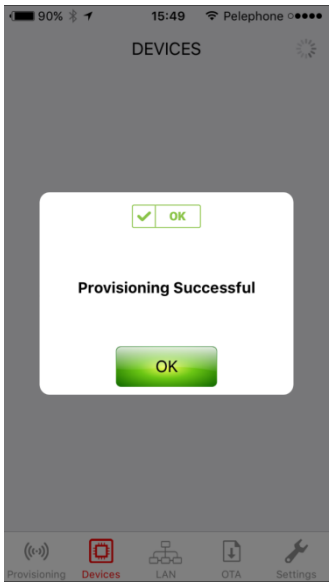

Figure 17. Wi-Fi Starter Provisioning Process (2/3)

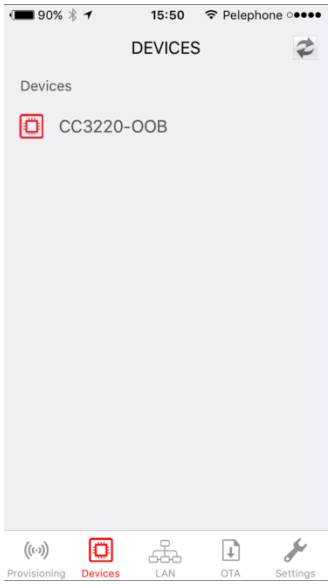

Figure 18. Wi-Fi Starter Provisioning Process (3/3)

Additional LAN and OTA tabs are applicable only to the OOB and automatically open after provisioning is successfully completed. If these tabs do not show up, navigate to the Devices tab and select the device by pressing and holding it.

6.1.2 CC3220 as AP

When the SimpleLink Wi-Fi Starter Pro application is not available, the user can set the CC3220 as an AP, and connect to it directly. Setting the CC3220 as an AP is very convenient when no AP is available. AP mode is applied by pressing the SW2 switch on the CC3220 LaunchPad. The terminal shows the Switching to AP Mode message, followed by a message indicating the IPv4 address of the CC3220 server.

To connect to the CC3220, the user must look for a network name that begins with mysimplelink-xxyyzz in which xxyyzz stands for the 3 least significant bytes of the unique MAC address of the CC3220. Successful connection to the CC3220 is indicated in the terminal where the MAC address, and the IP address of the connected station are presented.

Configuring the CC3220 in AP mode is not persistent, which means that manual reset of the board makes the CC3220 return to its default, configure as Station mode, and try to connect to a stored profile. If unsuccessful, the CC3220 is configured in provisioning mode. The only time AP is persistent is upon automated reset of the CC3220, which occurs as part of the OTA update procedure.

6.2 Browse Onboard Website

Users can browse the CC3220 website by simply opening a browser and typing the IP address of the device. The IP address of the device is printed on the terminal emulator. If the CC3220 is configured in AP mode, users can enter the URL link http://mysimplelink.net. The http:// prefix is mandatory. Another way to browse the CC3220 website by using the SimpleLink Wi-Fi Starter Pro mobile application by navigating to the Devices tab and selecting your device from the list.

The CC3220 streams the onboard web pages to the user's web browser. This procedure may take a few seconds to complete. Clicking the Start button automatically loads the demo page. Navigate through the different tabs by using the drop-down menu selector on the upper left corner of the screen.

Figure 19 shows the welcome page.

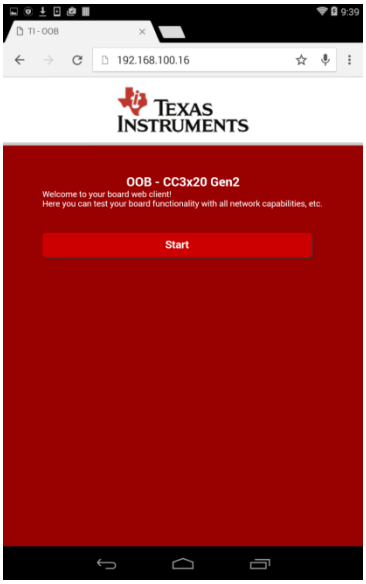

Figure 19. OOB Main Web Page

6.2.1 Local Network Demo

Users can navigate to the demo tab in three ways:

- Explicitly type demo.html as the requested resource on the browser URL, demo.html must be prefixed by the CC3220 IP address (see Figure 20).
- Use the drop-down menu selector on the upper left corner of the screen.
- Click the Start button on the main screen.

On this screen, users can control and get the state of the onboard sensors. These sensors include the red LED (D7) and the accelerometer. Additionally, users can fetch some device-specific information.

Figure 20 shows device information, status of the onboard red LED (D7), and periodic accelerometer readings, together with the equivalent 2D image of the CC3220 LaunchPad board.

Figure 20. Demo Web Page

The LED state is indicated as textual off or on under the LED control frame. To change the state of the red LED (D7), slide the switch to the opposite direction. To show board movement, move the CC3220 board. The update rate is once every 200 ms.

6.2.2 OTA Update

OTA updates through web pages will be supported in future releases.

6.2.3 Advanced Settings

Users can navigate to the demo tab in two ways:

- Explicitly type settings.html as the requested resource on the browser URL. The requested resource must be prefixed by the CC3220 IP address.
- Use the drop-down menu selector on the upper left corner of the screen.

This screen includes the ROM onboard pages (when no web pages exist on the serial flash). These pages can be used as advanced settings, letting users get information on the device, add, remove, or view profiles, ping other devices, and more. Figure 21 shows the advanced settings page.

Figure 21. Advanced Settings Web Page

6.3 Using the Mobile Application

Another way to control the CC3220 running the OOB demo is to use the SimpleLink Wi-Fi Starter Pro application.

6.3.1 Connect to the CC3220

The application includes the following tabs: Provisioning, Devices, and Settings. Additional LAN and OTA tabs are applicable to the OOB only, and are automatically opened after provisioning is successfully completed. If these tabs do not appear, navigate to the Devices tab and select the device by pressing and holding it.

If the device has never been provisioned, follow the Section 6.1.1. If the device has already been provisioned, it automatically connects to an AP, so users can start playing with LAN and OTA demonstrations. If the AP credentials have not been provisioned by the user's SimpleLink Wi-Fi Starter Pro application, or if the SimpleLink Wi-Fi Starter Pro application has been started while the device is already provisioned and connected to the AP, it is required to bind the application to the desired device. Users would need to navigate to the Devices tab and select the device by pressing and holding it.

6.3.2 Local Network Demo

On this screen, users can control and get the state of the onboard sensors. These sensors include the red LED (D7) and the accelerometer. Additionally, users can also fetch some device-specific information. It is important to know that the OTA update is adapted to the OOB application. If the user either programmed other examples or modified the OOB application, it is mandatory to program an official OOB again. See Section 5.

Figure 22 shows device information, status of the onboard red LED (D7), and periodic accelerometer readings of the CC3220 LaunchPad board.

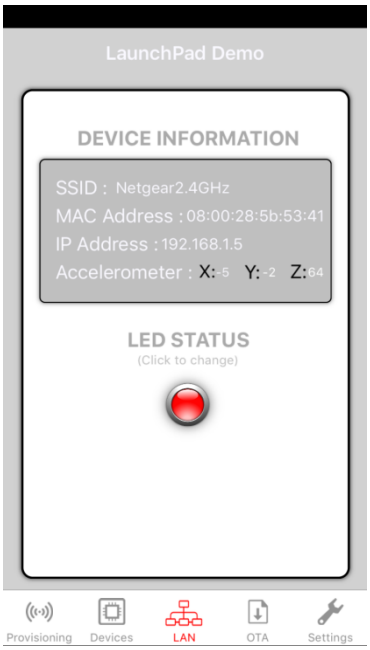

Figure 22. Mobile App LAN Tab

To change the state of the red LED (D7), tap on the LED icon. To show board movement, just move the CC3220 board. The update rate is once every second.

6.3.3 OTA Update

On this screen, users can see the current software version running on the CC3220 OOB application. Users can also check for new software versions by clicking the Check for software update button. The application then connects to a cloud server where the software version is stored, and downloads it. Users may modify this link by pressing and holding the download icon next to Check for software update button, and filling in the new URL. The procedure is monitored by an upload progress bar which gets feedbacks from the CC3220 device during the process.

It is important to know that the OTA update is adapted to the OOB application. If the user either programmed other examples, or modified the OOB application, it is mandatory to program an official OOB again. See Section 5. The procedure follows.

1. Click the Check for software update button to test whether a new software version exists. In Figure 23, the Current software version shows that no version file exists on the device, and New software version shows that no file has been downloaded from a data store server. If the device is up to date, an appropriate informative message appears asking the user to manually approve it if desired.

Figure 23. Mobile App OTA Tab

2. In Figure 24, clicking the Check for software update button triggers a new software version download from a data store server as indicated by New software version.

Figure 24. Mobile App Check for Software Update Completed

3. Alternatively, users may also modify this link by pressing and holding the download icon next to the Check for software update button, and filling in the new URL as shown in Figure 25.

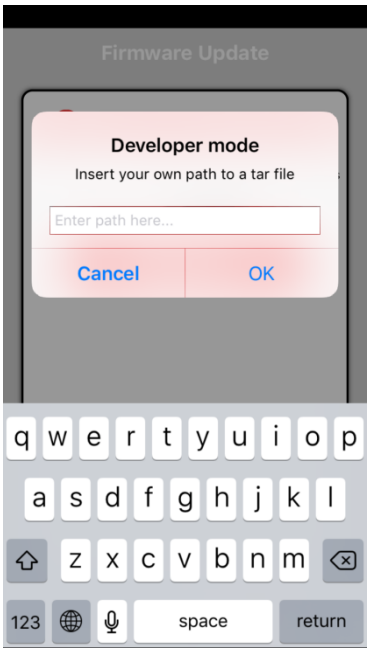

Figure 25. Mobile App Private OTA Repository

4. If a new software version has been downloaded, start the software update by clicking PROGRAM button (see Figure 26 and Figure 27).

Figure 26. Mobile App OTA Started

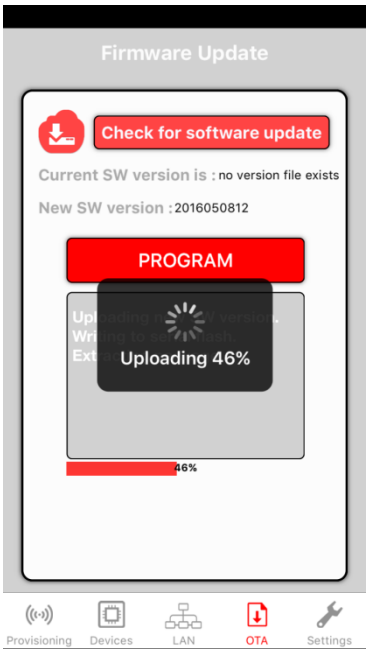

Figure 27. Mobile App OTA in Progress

The upload progress bar appears with the following messages:

- Download started
  - Uploading new software version
  - Extracting archive file
  - Writing to serial flash
- Download done
  - Rebooting…
  - Testing new software version
- Verdict
  - When successful → Done
  - When failed → OTA Update failed

Upon successful update, the new version appears under the New software version section (see Figure 28). The Upload finished! message appears on the bottom of the screen, and the screen is enabled again.

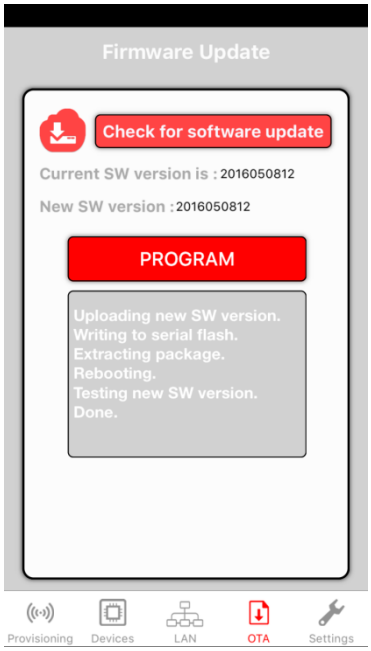

Figure 28. Mobile App OTA Done

Table 1 lists some abnormal behaviors from the OTA procedure.

Table 1. OTA Abnormal Behavior From Mobile App

Behavior	Cause	Corrective Action
Progress bar does not start	Client is not able to open connection to the CC3220 report server.	Usually, it is harmless. Probe the red LED (D7), terminal messages, and the mobile application for more information. Upon a successful process, the red LED (D7) stops rapidly blinking, momentarily turns solid red, and then follows the Local Network Connection procedure described under Section 7.1. Terminal messages should indicate a successful OTA update process as described under Section 7.3. The mobile application shows no error message, and the Current software version field updates to the new version. Users can then navigate to the Demo tab and continue with the demonstration.
Progress bar is stuck	Client is not able to communicate with the report server, or some error occurred during the process.	If the client is not able to communicate, the OTA procedure may still finish successfully. The same applies as when the progress bar does not even start (follow previous item on table). If some errors occur, the red LED (D7) stops rapidly blinking and goes off. Users can restart the OTA.

6.3.3.1 Local OTA When CC3220 is in AP Mode

When the CC3220 is set in AP mode, the mobile application can communicate to it just as if both were connected though a local AP. The OTA feature is the only use case that needs clarification. The reason for this is the image for update is on a cloud server, and because the mobile application is connected to the CC3220, it is not able to fetch and download the image. To make it possible, follow these steps.

1. Set the CC3220 LaunchPad as AP mode.
2. Go to the Wireless Settings, locate the SSID of the CC3220 LaunchPad, and connect to it (starts with mysimplelink prefix). At this point, the terminal displays a printout stating a device has been connected.
3. Open the SimpleLink Wi-Fi Starter Pro, and navigate to the Devices tab. The connected CC3220 with the MAC address as a prefix appears on the list.
4. Press and hold the CC3220 icon. A message appears with the IP address, and the LAN and OTA tabs also become visible.
5. Go back to the Wireless Settings, and connect to an AP that has Internet connection.
6. Go back to SimpleLink Wi-Fi Starter Pro and navigate to the OTA tab. Click the Check for software update button and the image is downloaded.
7. Go to Wireless Settings, locate the SSID of the CC3220 LaunchPad, and connect to it.
8. Go back to the SimpleLink Wi-Fi Starter Pro, and navigate to the OTA tab. Now users can upgrade the version.

6.4 Returning to Factory Image

The OOB application lets users return to the factory image. Factory image means that the original image stored on the serial flash is extracted, erasing all content from the serial flash. This assumes that an OOB image resides on the serial flash; otherwise, users would need to reprogram the CC3220 using the ImageCreator utility. Returning to the factory image is equivalent to reprogramming an image from scratch.

One scenario in which users would need to return to the factory image is if a profile exists, and must be replaced by another one. Returning to the factory image is an embedded feature and is not application dependent. The procedure is described in the *Restore to Factory by Using the SOP* section of the *SimpleLink™ Wi-Fi® and Internet of Things CC3120 and CC3220 Network Processor Programmer's Guide*.

For convenience, the procedure is as follows.

1. Set the SOP to 011 (SOP2 = 0, SOP1 = 1, SOP0 = 1), and perform power on reset (POR). Device reset can be simply done by pressing SW1 Reset switch. See Figure 29 for reference of SOP jumpers.

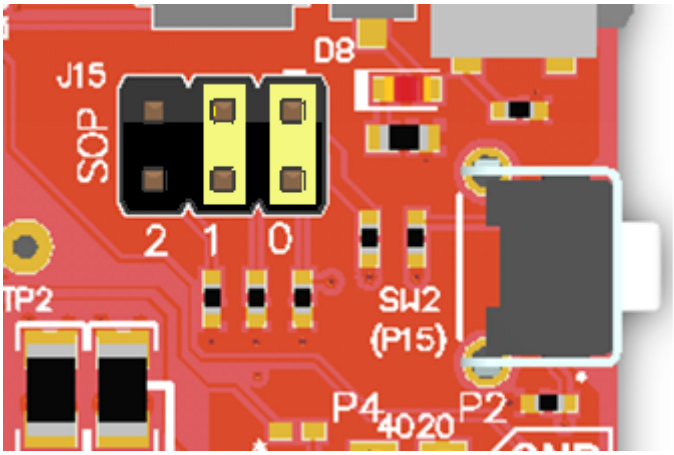

Figure 29. SOP Jumpers Configuration

2. The device is initiated in Restore to Factory phase, set the SOP jumpers to 000 (SOP2 = 0, SOP1 = 0, SOP0 = 0).
3. At this point, returning to the factory image occurs because the image is large, and the entire file system is recreated, which may take up to 1 minute to complete. Upon successful completion, the default OOB application executes.
   - The terminal message should instruct the user to perform POR.
   - Because the application is initiated, the red LED starts flashing once every second.
4. The OOB application requests the user perform POR. Manual reset is mandatory in this case.
5. The user performs POR and the default OOB application is executed.

7 Troubleshooting

7.1 Red LED Legend

This section lists all state options of the red LED (D7). The red LED (D7) indication can help users understand the application flow, and debug cases where the application does not behave as expected. Only the red LED (D7) is used for status indication, and similar indications may be applied in more than one occasion. To distinguish similar indications, the user must be aware of the executed procedure. Table 2 lists all options.

Table 2. LED Indication Legend

Red LED indication	Procedure	Interpretation
Blinking once per 2 seconds	Provisioning	CC3220 is being provisioned.
Solidly on	Provisioning	Connection to the AP is up.
Solidly off	Provisioning	Connection to the AP is down.
Blinking once per second	Local network connection	For Station mode, connection to the AP is in progress. For AP mode, initialization is in progress.
Solidly on	Local network connection	For Station mode, connection to the AP is up. For AP mode, initialization is done.
Solidly off	Local network connection	Applies in Station mode only. Connection to the AP is down.
Blinking 5 times per second	OTA update	OTA update of the CC3220 is in progress.
Solidly on	OTA update	OTA update of the CC3220 succeeded.
Solidly off	OTA update	OTA update of the CC3220 failed.
Blinking once per second	Return to default	Return to default procedure is done. Waiting for the user to manually restart the CC3220 LaunchPad

7.2 Extra Debugging

Extra debug level messages can be configured to help users and TI representatives get a better view of the application internals. To do so, users must uncomment the OOB_DEBUG_PRINT definition in the oob_common.h file and recompile.

7.3 Terminal Messages

The terminal can also be used for debug messages. The default configuration is 115200 bps, 8 bits, no parity, and 1 stop bit. Most of the debug messages are self-explained. For better tracking, each message is preceded by the task and module generating the message enclosed in brackets.

Table 3 lists some common messages to help users track the application status.

Table 3. Terminal Messages

Message	What Does it Mean?
[Provisioning task] Provisioning Started. Waiting to be provisioned..!!	The CC3220 is in provisioning mode. The user must use the SimpleLink Wi-Fi Starter Pro application to provision the device.
[ProvisioningEvent] Connection to AP succeeded	During provisioning, the CC3220 managed to decode the AP credentials and connect to it successfully.
[ProvisioningEvent] Confirmation Success!	During provisioning, the CC3220 supplied feedback to the SimpleLink Wi-Fi Starter Pro application.
[Provisioning task] Provisioning completed successfully..!	Provisioning completed successfully.
[ProvisioningEvent] Provisioning stopped	Provisioning is stopped, which may indicate the following: • Successful provisioning • Inactivity time-out elapsed • General error
[Provisioning task] committing new OTA download...	Upon end of OTA procedure, indicates a new software package is being committed.

Table 3. Terminal Messages (continued)

Message	What Does it Mean?
[Provisioning task] commit succeeded	New OTA software package committed successfully.
[Provisioning task] Rollback error sl_FsCtl	New OTA software package failed its validation testing (connect to AP while in Station mode or initialize while in AP mode), reverting to previous software package.
[Provisioning task] failed to commit new download, reverting to previous copy by resetting the device	New OTA software package failed to commit, reverting to previous software package.
Return To Factory Image successful. Do a power cycle (POR) of the device using switch SW1-Reset	After a return-to-factory is triggered and the device is up, a manual reset is required from the user to complete the operation.
Connection Success (feedback to Smartphone app failed)	During provisioning, the CC3220 connected to the AP successfully, but failed to supply feedback to the SimpleLink Wi-Fi Starter Pro application.
[Link local task] HTTP GET Request	Indicates client HTTP GET request. For details see Table 4.
[Link local task] characteristic is:	During HTTP request, indicates the relevant resource.
[Link local task] HTTP POST Request	Indicates client HTTP POST request. For details see Table 4.
[Link local task] value is:	During HTTP request, indicates the value of the resource.
[Link local task] HTTP PUT Request	Indicates client HTTP PUT request. For details see Table 4.
[Link local task] Received OTA filename	Archive filename received during the OTA update. Filename and file size follows.
[Link local task] OTA filename is in *.tar format	Archive filename received during the OTA update is not in tar format.
[Link local task] OTA bundle version file does not exist	OTA version file does not exist on the file system. This is informative, not an error.
[OtaArchive_CheckVersion] accept the new version	OTA version file does not exist on the file system. New version is accepted. Version number follows.
[OtaArchive_CheckVersion] newer version update	New OTA version is newer than the stored version.
[OtaArchive_CheckVersion] older version update	New OTA version is older than the stored version.
[OtaArchive_RunParseTar] Create/Open for write file	File is updated on file system during the OTA update. Filename follows.
[OtaArchive_RunParseTar] Downloading File Completed	File has been downloaded and updated on file system successfully during the OTA update.
[OTA report task] OTA progress	The status of the progress bar displays in percentages during the OTA update.
[Link local task] sl_extLib_OtaRun: — Download file completed	OTA procedure completed successfully.
[Link local task] sl_NetAppRecv error	Error in communication with the client during the OTA update. Error code follows.
[Link local task] OtaArchive error	Error in the archive module during the OTA update. Error code follows. For details refer to the OtaArchive.h header file.
[Control task] switching to AP mode	The CC3220 is switching to AP mode.
[Control task] device not started in AP role	Configuring the CC3220 as AP failed.
[Control task] device started in AP role, rebooting device...	Switching to AP mode succeeded, rebooting application.
[Control task] device cannot start in AP mode, please reset the board	Switching to AP mode failed. Manual reset is required.
[ERROR] - FATAL ERROR	Fatal error occurred. Manual reset is required.

8 Limitations and Known Issues

- The OTA update through web pages will be supported in future releases.
- Configuring the CC3220 in AP mode is not persistent, which means that manual reset of the board makes the CC3220 return to its default, configure as Station mode, and try to connect to a stored profile. If unsuccessful, the board is configured in provisioning mode.
- Up to 20 non-default files can be updated during the OTA update. Default files include files that are nonsecured, part of a bundle, fail-safe, and with the size of the original file.
- Maximal file length inside a tar is 128 bytes. It is the full path as it appears in the tar file itself (including all directories from the root).
- Rarely, after provisioning is complete, the LAN and OTA tabs do not appear. If these tabs do not appear, navigate to the Devices tab and select the device by pressing and holding it.
- Rarely, it is observed that the progress bar does not start, or starts but freezes in the middle; however, the OTA update procedure is successful. In these cases, the red LED (D7) indication and terminal printouts show the true status of the procedure, and also the Web client or mobile application eventually indicates a successful process.
- The Internet Explorer 11 browser does not work in its default state. To make it work, a possible workaround is to use development mode (F12), and switch the Edge drop-down menu on the top right to 10 instead of default.

9 Out-of-Box for Advance Users

9.1 Installations for Advance Users

The following installations are additional to those required for basic users. See Section 3.

9.1.1 CC3220 Software Development Kit (SDK)

1. Download the following software from the CC3220 SDK package.
2. Run the installer by double clicking on the CC3220 SDK installer.
3. Read and accept the license agreement to proceed.
4. Choose the desired path to place the package (otherwise the default is chosen).
5. Proceed with the installation and click the Finish button when done.

9.1.2 Service Pack

If the board is not already flashed with the service pack for the SDK, the latest service pack for the SDK must be flashed on the CC3220 wireless MCU. The service pack installer can be found in <SDK install dir> \tools\cc32xx_tools\servicepack-cc3x20. .

9.1.3 CCS IDE

The latest Code Composer Studio™ (CCS) installer can be downloaded here. CCS is a free tool provided by TI, which enables developers to work with various TI devices. The SDK supports CCS version 6.2.0 or later.

- Double click on the installer and follow the instructions to install the tool.
- TI recommends using the default installation directory.
- Ensure to select the SimpleLink CC3x Wireless MCUs option for processor support as shown in Figure 30.

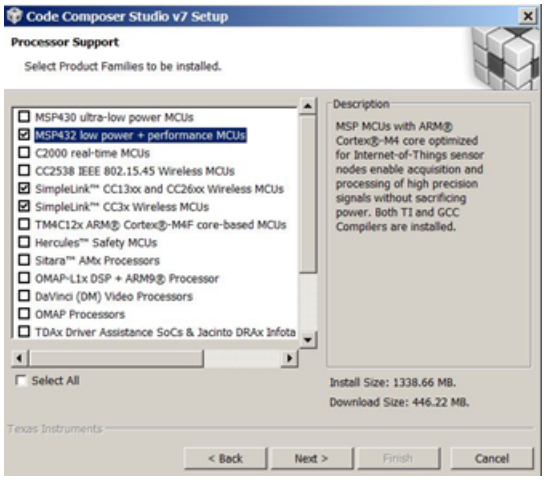

Figure 30. CCS Setup – Choosing CC32xx SimpleLink Devices

- To use a debug probe, select TI XDS Debug Probe Support as shown in Figure 31.

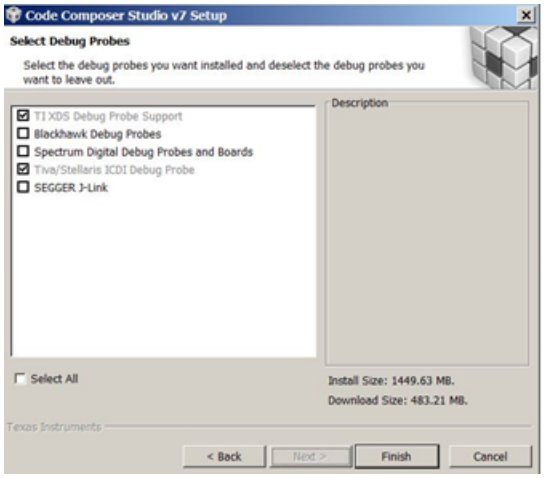

Figure 31. CCS Setup – XDS Installation

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

9.1.3.1 Add CCS Support for FreeRTOSv9

See the following instructions if users want to use FreeRTOS.

1. Download FreeRTOS version 9.
2. Install the software in the C: drive.
3. Copy the content of the patch (CCS folder) from <SDK install path>\tools\cc32xx_tools\FreeRTOS_patch\CCS and paste it to <FreeRTOS install path>\FreeRTOS\Source\portable\CCS.
4. Run CCS.
5. Choose Window → Preferences → Code Composer Studio → Build → Variables → Add (see Figure 32).

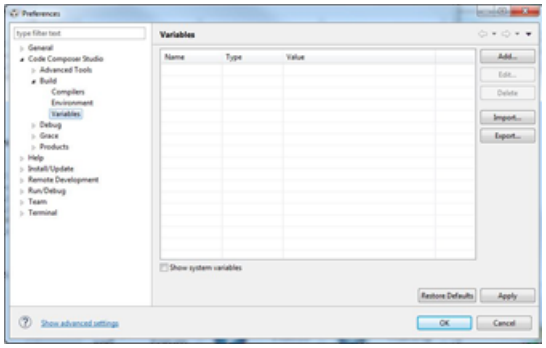

Figure 32. FreeRTOS Support in CCS (1 of 2)

6. Fill the Variable field with FREERTOS_INSTALL_DIR.
7. Change the Type to directory.

8. Change the Value to the <FreeRTOS install path> and then press OK (see Figure 33).

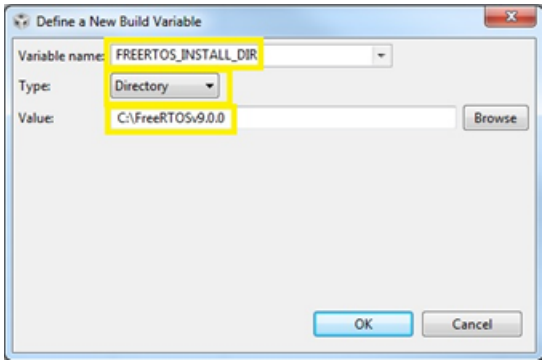

Figure 33. FreeRTOS Support in CCS (2 of 2)

9.1.3.2 Product Settings Verification

Follow the instructions to verify that the required products are installed.

1. Choose Window → Preferences → Code Composer Studio → Products (see Figure 34).

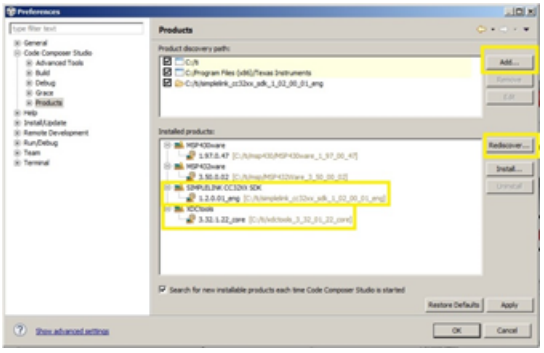

Figure 34. CCS Installed Products

2. Check that the paths in Product Discovery Path include the paths of:
   - SDK install path
   - XDCtools

   If not:
3. Choose Add, then add the paths and choose Rediscover. After the rediscover, an Install Discovered Products window may appear. If the window appears, the new products have been discovered. If the products mentioned are in the list, select them and click Install (see Figure 35).

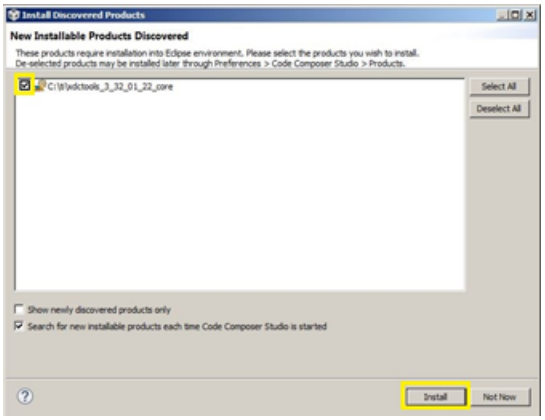

Figure 35. CCS Discovered Products

9.2 Source Files Briefly Explained

out_of_box.c – main file creates all tasks in the system. The tasks include the simplelink task, provisioning task, link local task, OTA task, and control task. This file also includes implementation of the control task which implements the switching to AP mode and registration of all SimpleLink callbacks

Provisioning_task.c – includes all provisioning-related implementation and procedures. The file also includes a piece of code for the OTA bundle committing procedure. The flow of the task follows:

1. Decide whether the CC3220 is in station or AP mode.
2. If the CC3220 is in station mode:
   - Connect to an AP using a stored profile.
   - Check if the CC3220 is coming out of an OTA update. If so, check whether the CC3220 managed to connect to the AP.
     - If connected, commit the OTA.
     - If not connected, reboot the application so the new image is reverted.
   - Check whether the CC3220 managed to connect to the AP.
     - If the CC3220 is not connected, start provisioning.
3. If the CC3220 is in AP mode:
   - Check if it is coming out of an OTA update. If so, check whether the CC3220 managed to initialize an AP.
     - If successful, commit the OTA.
     - If unsuccessful, reboot the application so the new image is reverted.
   - Check whether the CC3220 managed to initialize an AP.
     - If unsuccessful, start provisioning.

Link_local_task.c – includes implementation of local link communication between the CC3220 web server and the connected web client. The set of available operations include retrieving data from the CC3220 database, as well as updating the CC3220 database. All operations use RESTful API.

Table 4 lists the set of all operations.

Table 4. HTTP Server Methods

Service	Operation	Description
Device	Get	Fetches MAC address, IP address, and SSID of the AP, to which the CC3220 server is connected. In case CC3220 is in AP mode, SSID is the name of CC3220.
Light	Get and Post	Fetches and updates the state of the onboard red LED (D7).
Sensor	Get	Fetches the value of X, Y, and Z accelerometer axis.
OTA	Put	Uploads the OTA image to the CC3220 web server.
OTA	Get	Fetches the current OTA version.

Ota_task.c – implements a tiny HTTP report server (running on proprietary TCP port). This server reports the status of the progress bar during the OTA update.

OtaArchive.c – extracts the archive image received during the OTA update, and updates the local file system.

OtaJson.c – includes minimal Json parser services tailored for parsing the metadata header file during the OTA update.

oob_common.c – includes common resources for all tasks. These tasks include synchronization and mailbox objects, sensors readings, and platform reboot.

bma222drv.c – implements an accelerometer sensor driver.

tmp006drv.c – implements a temperature sensor driver.

9.3 Building the OOB Project Using CCS

To build the OOB project, follow these steps.

1. Open CCS and navigate to Project → Import CCS Projects
2. Browse to the <SDK install path>\examples directory. The SDK provides NoRTOS, TI-RTOS, and FreeRTOS based examples. TI-RTOS and FreeRTOS examples have tirtos or freertos in the project name.
3. Import the OOB project using CCS IDE from the following location, according to the connected CC3220 device and the desired OS (freertos or tirtos, see Figure 36).
   - CC3220S secured device:
     - <SDK install path>\examples\rtos\CC3220S_LAUNCHXL\demos\out_of_box\tirtos\ccs
     - <SDK install path>\examples\rtos\CC3220S_LAUNCHXL\demos\out_of_box\freertos\ccs
   - CC3220SF secured device:
     - <SDK install path>\examples\rtos\CC3220SF_LAUNCHXL\demos\out_of_box\tirtos\ccs
     - <SDK install path>\examples\rtos\CC3220SF_LAUNCHXL\demos\out_of_box\freertos\ccs

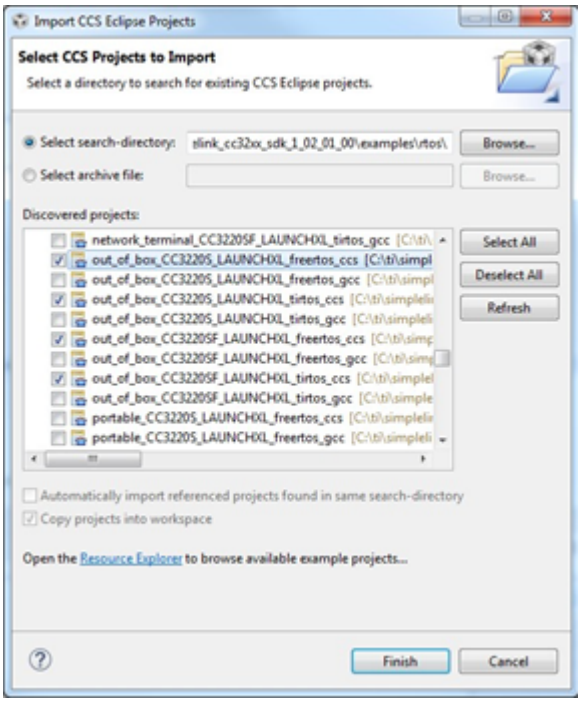

Figure 36. Import OOB Project

4. Make the required changes and rebuild the project.

   Importing TI-RTOS and FreeRTOS examples will import kernel projects in the same workspace. The kernel project is a dependent project that automatically builds when the example is built.

There are two options to execute the code:

- Run in debug mode.
- Program the OOB binary using UniFlash.

9.3.1 Executing in Debug Mode

To execute in debug mode, the device must first be opened in development mode.

To have all OOB content, such as the service pack, web pages, and other system and configuration files, an UniFlash project is required so they can be programmed into the device. The UniFlash projects are available under the following locations, according to the connected CC3220 flavor.

- For CC3220S secured device:

  <SDK install dir>\examples\rtos\CC3220S_LAUNCHXL\demos\out_of_box\uniflash\OOB_RS_tirtos.zip

  <SDK install dir>\examples\rtos\CC3220S_LAUNCHXL\demos\out_of_box\uniflash\OOB_RS_freertos.zip
- For CC3220SF secured device:

  <SDK install dir>\examples\rtos\CC3220SF_LAUNCHXL\demos\out_of_box\uniflash\OOB_SF_tirtos.zip

  <SDK install dir>\examples\rtos\CC3220SF_LAUNCHXL\demos\out_of_box\uniflash\OOB_SF_freertos.zip

Follow these instructions to program the image into the device:

1. Open the UniFlash utility to create and program the OOB image.
2. Choose CC3120/CC3220 device, and then click the Start Image Creator button (see Figure 37).

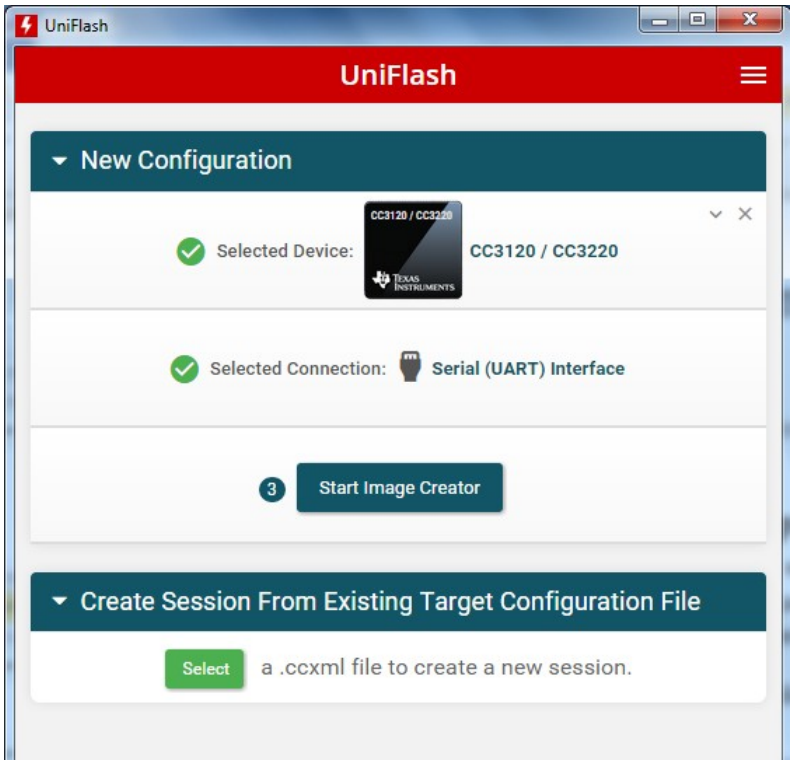

Figure 37. UniFlash – Choose CC3120 / CC3220 Device

3. Click the Manage Projects button, and then click the Import project from ZIP file button. Locate the OOB project according to the connected CC3220 device. Projects already imported appear on the list of available projects (see).
   - For CC3220S secured device:

     <SDK install dir>\examples\rtos\CC3220S_LAUNCHXL\demos\out_of_box\uniflash\OOB_RS_tirtos.zip

     <SDK install dir>\examples\rtos\CC3220S_LAUNCHXL\demos\out_of_box\uniflash\OOB_RS_freertos.zip
   - For CC3220SF secured device:

     <SDK install dir>\examples\rtos\CC3220SF_LAUNCHXL\demos\out_of_box\uniflash\OOB_SF_tirtos.zip

     <SDK install dir>\examples\rtos\CC3220SF_LAUNCHXL\demos\out_of_box\uniflash\OOB_SF_freertos.zip
4. Navigate to General → Settings on the left side tree and change the image mode to Development.

5. Ensure the CC3220 LaunchPad is connected and click the Connect button. Users can see the device information of the connected device with their device MAC address (see Figure 38). This is important because the development mode is matched to the device MAC address.

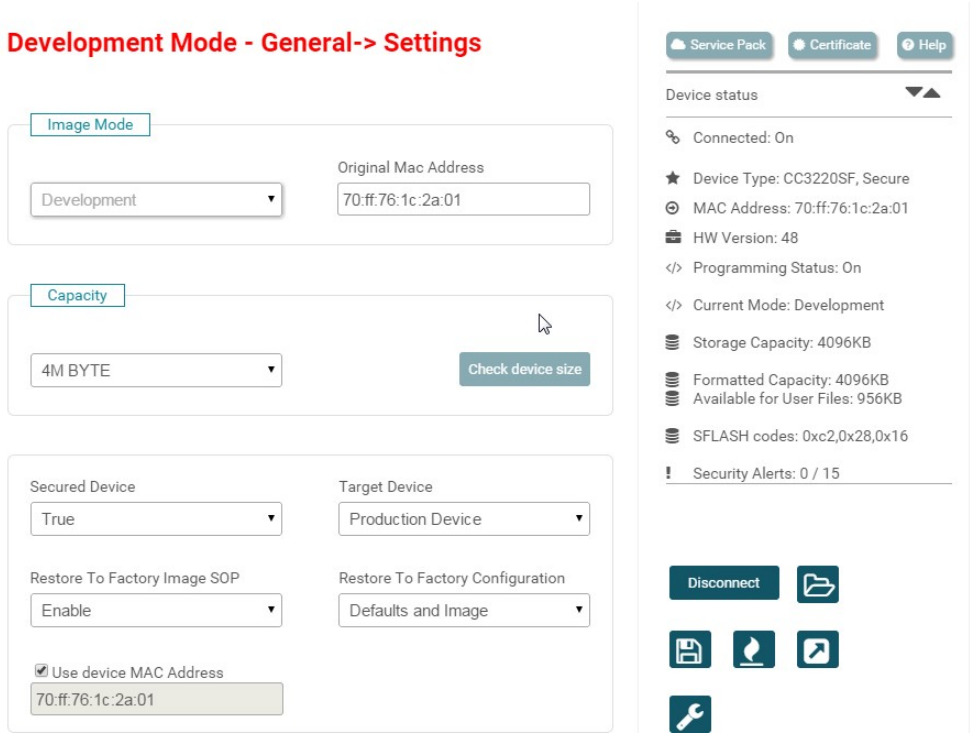

Figure 38. UniFlash General Settings

6. Ensure the Use device MAC Address box is checked and that your device MAC address appears on the top.
7. Browse and see all files under Files → User Files on the left-side navigation tree.

   It is good practice if mcuimg.bin for the CC3220S device or mcuflashimg.bin for the CC3220SF device is deleted from the project (a backup resides in the original zipped project). This step ensures that the programmed application binary is not executed and that the code downloaded through CCS is executed (see Figure 39).
8. Save the project and click on the Generate Image icon on the right side.

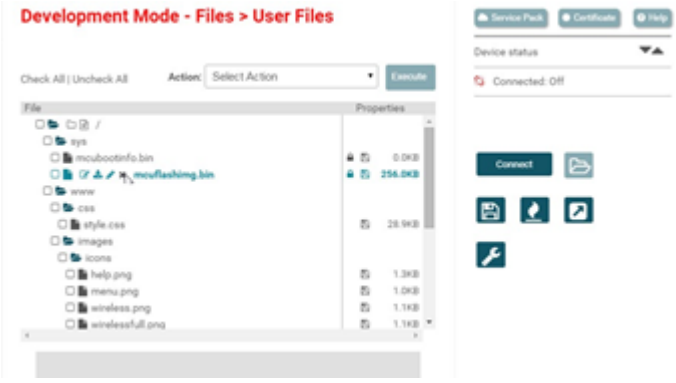

Figure 39. Deleting Application Binary

9. Finally, click on the Program Image (Create and Program) button (see Figure 40). The image is now programmed to the device.

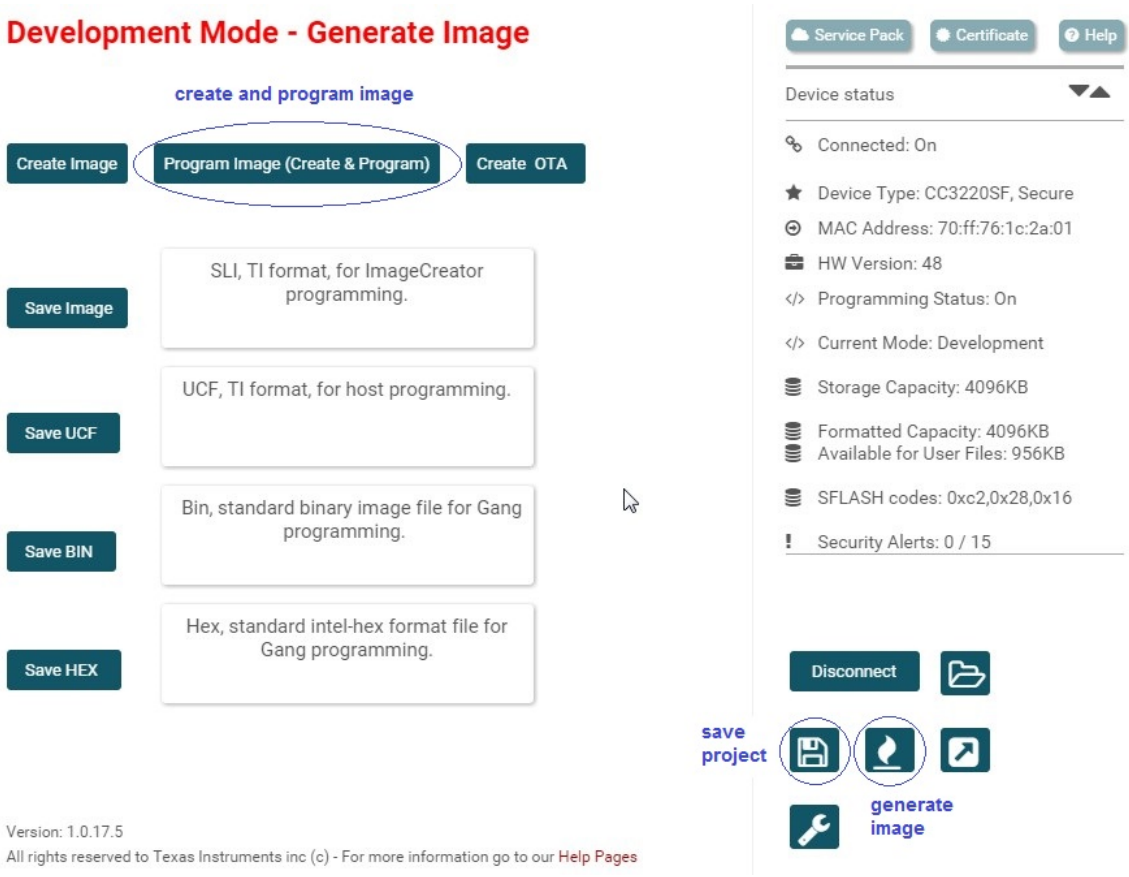

Figure 40. UniFlash Generate and Program Image

10. Upon successful programming, open CCS, make your modifications and recompile.
11. Click the debug icon and start debugging.

9.3.2 Program a User Modified OOB Binary

When working in debug mode, the application binary is not programmed to the serial flash and is volatile. The compiled binary can be added to the project and create an image that is programmable to the device; this way, the application binary is programmed to the serial flash and becomes non-volatile.

For secured devices, users must add their own certificate chain (the TI private key which pairs the certificate cannot be shared). Nevertheless, to simplify the procedure TI also provides the option to use a self-signed certificate store (playground).

To have all out-of-box content such as the service pack, web pages, and other system and configuration files, a UniFlash project is required so the image can be programmed into the device. The UniFlash projects are available under the following locations according to the connected CC3220 flavor.

- For CC3220S secured device:

  <SDK install dir>\examples\rtos\CC3220S_LAUNCHXL\demos\out_of_box\uniflash\OOB_RS_tirtos.zip

  <SDK install dir>\examples\rtos\CC3220S_LAUNCHXL\demos\out_of_box\uniflash\OOB_RS_freertos.zip

- For CC3220SF secured device:

  <SDK install dir>\examples\rtos\CC3220SF_LAUNCHXL\demos\out_of_box\uniflash\OOB_SF_tirtos.zip

  <SDK install dir>\examples\rtos\CC3220SF_LAUNCHXL\demos\out_of_box\uniflash\OOB_SF_freertos.zip

Follow these instructions to program the image into the device (the first four steps are identical to Section 9.3.1, but are listed again here for completeness).

1. Open the UniFlash utility to create and program the OOB image.
2. Choose the CC3120/CC3220 device and then click the Start Image Creator button (see).
3. Click the Manage Projects button and then click Import project from ZIP file button. Locate the OOB project according to the connected CC3220 device. Projects already imported should appear on the list of available projects.
   - For CC3220S secured device:

     <SDK install dir>\examples\rtos\CC3220S_LAUNCHXL\demos\out_of_box\uniflash\OOB_RS_tirtos.zip

     <SDK install dir>\examples\rtos\CC3220S_LAUNCHXL\demos\out_of_box\uniflash\OOB_RS_freertos.zip
   - For CC3220SF secured device:

     <SDK install dir>\examples\rtos\CC3220SF_LAUNCHXL\demos\out_of_box\uniflash\OOB_SF_tirtos.zip

     <SDK install dir>\examples\rtos\CC3220SF_LAUNCHXL\demos\out_of_box\uniflash\OOB_SF_freertos.zip
4. Browse to see all files under Files → User Files on the left side tree. As shown in Figure 41, a dummy-root-ca-cert certificate is used in case an MCU image is added to the project.

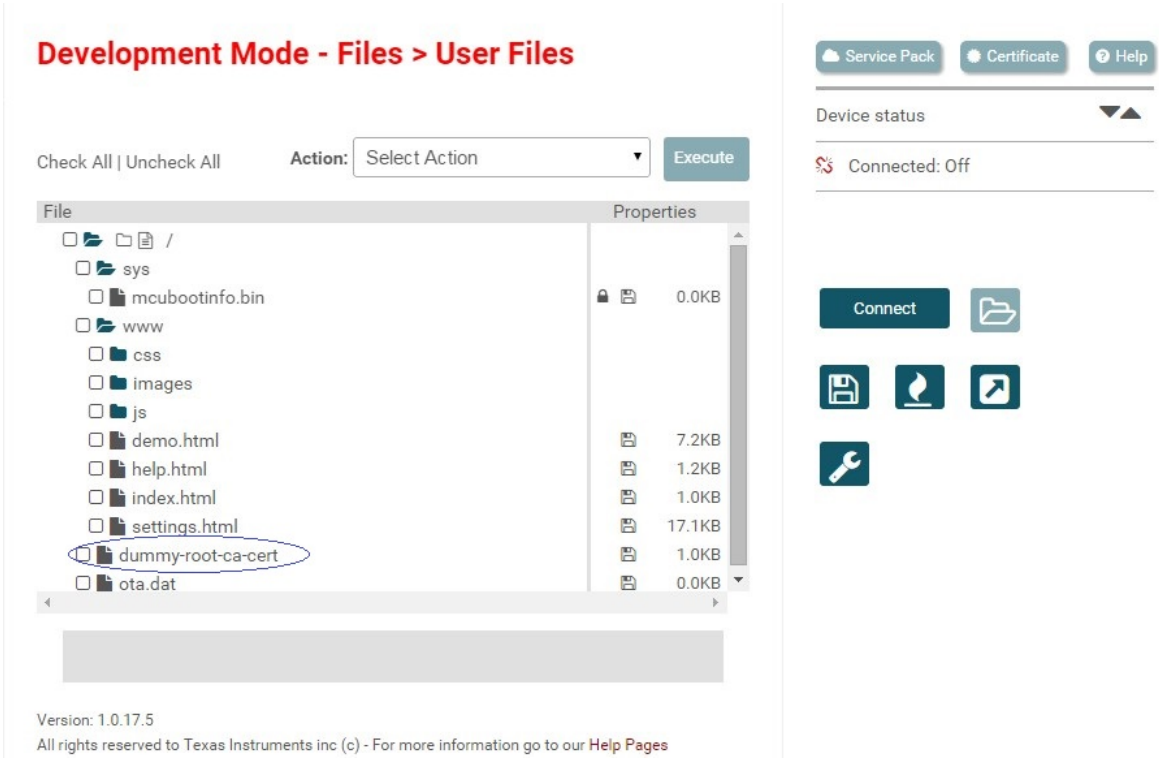

Figure 41. Root Certificate in UniFlash Project

5. Create the MCU image for the project by making code modifications and recompiling.
6. Add the created application binary to the Image Creator project (see Figure 42):
   (a) Browse to Files → User Files on the left side tree. The dummy-root-ca-cert certificate must be used if an MCU image is added to the project.
   (b) Choose Select MCU Image from the Action drop-down menu and click the Browse button.
   (c) Locate the compiled binary from Step 5 and select it.
   (d) Because the MCU image is secured, the certificate and signature must be configured.
      (i) To choose the certificate, choose dummy-root-ca-cert under the Certification File Name drop-down menu.
      (ii) To create a signature, leave the default Private Key File Name from the drop-down, click the browse button, and choose the dummy-root-ca-cert-key key which matches dummy-root-ca-cert-key and click the Write button. UniFlash automatically creates the signature. The key file is located on the SDK under <SDK install dir>\tools\certificate-playground

NOTE: For secured flash devices (CC3220SF), the default MCU image size is automatically set to 524288 bytes. This size is too large for the image (there are many other files as well). Set it to 262144 bytes.

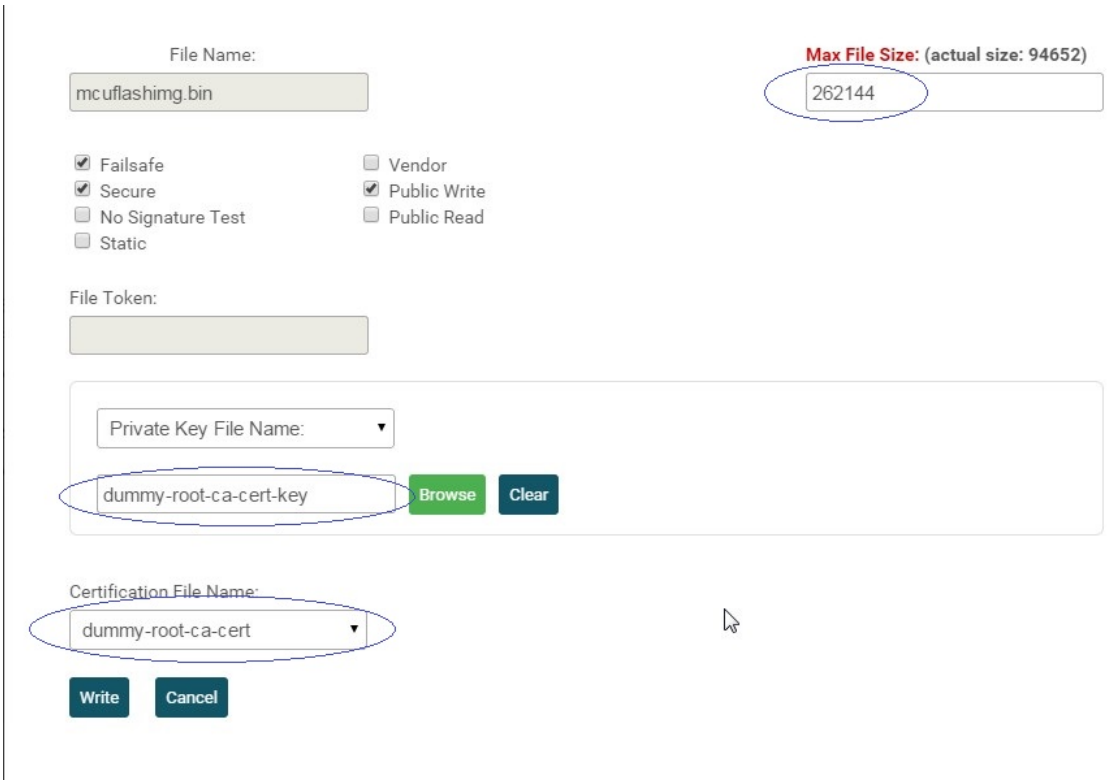

Figure 42. UniFlash Add Secured MCU Image

7. Browse to General → Settings on the left side tree and configure the image mode to either Production or Development, either works (choosing Development enables debugging).
8. Ensure the CC3220 LaunchPad is connected and click the Connect button. Users can see the device information of the connected device with their device MAC address.
9. Save the project and click on the Generate Image icon on the right.
10. Finally, click on the Program Image (Create & Program) button. The image is now programmed to the device.

Upon successful programming, the application starts running.

Revision History

Date	Revision	Notes
February 2017	SWRU473*	Initial release

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