Getting Started Guide

CC3230x-CC26x2EM-7ID Getting Started Guide

ABSTRACT

This guide is intended to assist users in understanding the hardware features of the CC3230x-CC26x2EM-7ID Coex Board, how to get started with programming each device, and how to evaluate each device's RF performance independently.

Table of Contents

1 Hardware Description......2
  1.1 Wired Connections, Jumper Settings, Buttons, and LEDs......2
  1.2 Power......8
  1.3 RF Connections......8
2 Wi-Fi Test Procedure......9
  2.1 Prerequisites......9
  2.2 Configure Board......9
  2.3 Programming the Wi-Fi Device......11
3 Bluetooth Low Energy Test Procedure......15
  3.1 Prerequisites......15
  3.2 Program the CC3230......15
  3.3 Configure the CC26x2 Board......15

List of Figures

Figure 1-1. CC3230x-CC26x2EM-7ID Board Overview......2
Figure 1-2. SOP Jumpers (Default Setting Shown)......3
Figure 1-3. Independent Device Reset (Default Configuration)......3
Figure 1-4. Synchronized Device Reset Configuration......4
Figure 1-5. CC3230 Porgramming Header (J6)......4
Figure 1-6. CC3230 Debug Logs Header (J4)......5
Figure 1-7. CC26x2 Auxiliary Header......5
Figure 1-8. SWx MUX – CC26x2......6
Figure 1-9. SWx MUX – CC3230......6
Figure 1-10. Using Onboard Antenna......8
Figure 1-11. Board Modified for Conducted Measurements......8
Figure 2-1. Wiring for Debugging, Programming, and UART Communication......9
Figure 2-2. Programming and UART Communication – Wiring Diagram......10
Figure 2-3. sysconfig Custom Board......11
Figure 2-4. sysconfig - Configure Board......12
Figure 2-5. Radio Tool......12
Figure 2-6. Select Device and Launch ImageCreator......12
Figure 2-7. Create New Project in Image Creator......13
Figure 2-8. Set up New Project......13
Figure 2-9. Load and Program ServicePack and MCU Image......14
Figure 2-10. Load and Program ServicePack and MCU Image......14
Figure 3-1. Wiring for Full Debugging, Programming, and UART Communication......16

List of Tables

Table 1-1. SOP[2:0] (J11)......3
Table 1-2. Push Button Definitions......7
Table 1-3. LED Indicators......7

Trademarks

LaunchPad™, SimpleLink™, and SmartRF™ are trademarks of Texas Instruments.
Arm® is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.
Wi-Fi® is a registered trademark of Wi-Fi Alliance.
Bluetooth® is a registered trademark of Bluetooth SIG.
All trademarks are the property of their respective owners.

1 Hardware Description

Figure 1-1 shows the CC3230x-CC26x2EM-7ID Coex Board.

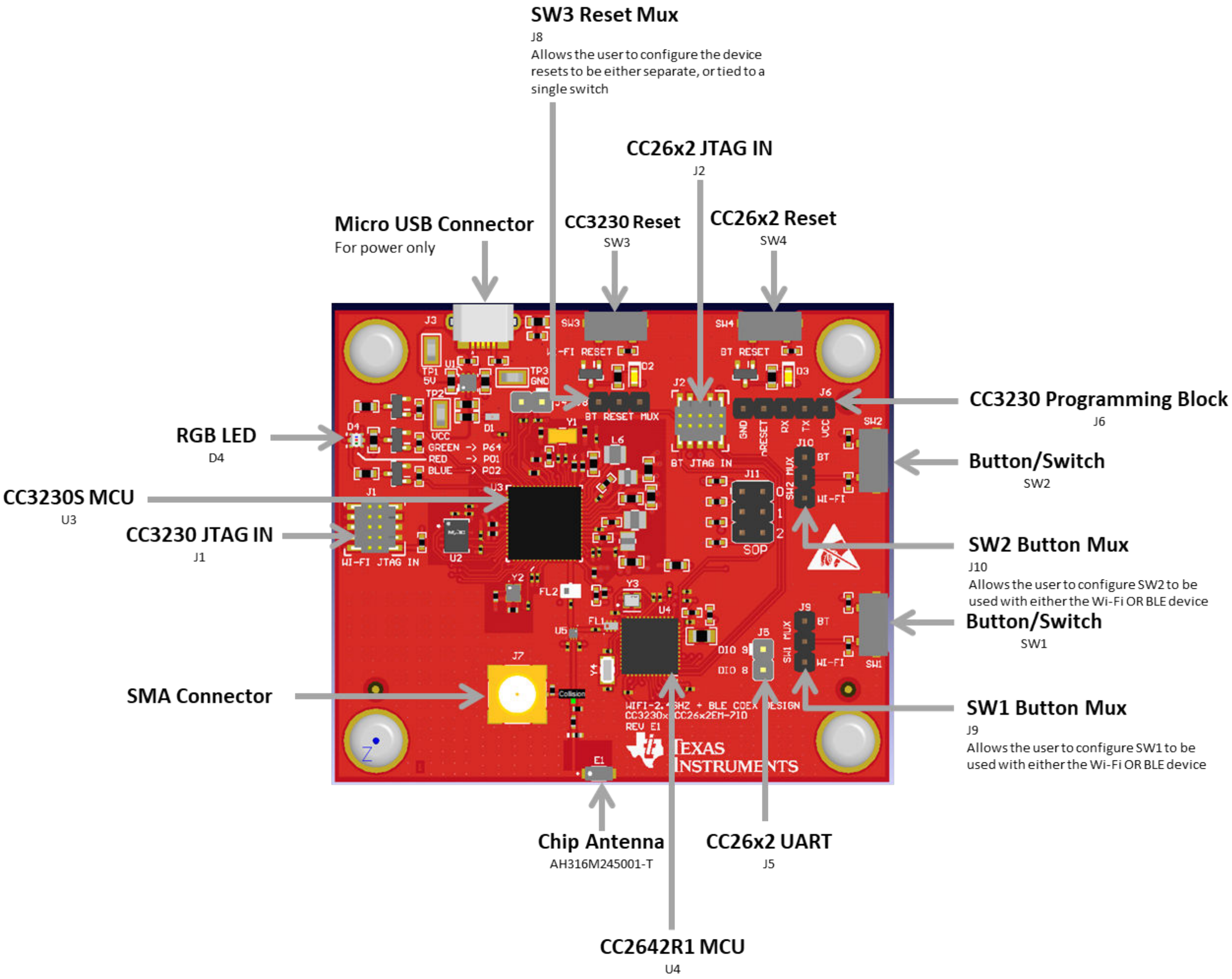


Figure 1-1. CC3230x-CC26x2EM-7ID Board Overview

1.1 Wired Connections, Jumper Settings, Buttons, and LEDs

1.1.1 JTAG

The CC3230x-CC26x2EM-7ID Coex Board is designed to be used in conjunction with a LaunchPad™ development kit for debugging. Connecting J2 to an XDS110 LaunchPad allows for debugging and programming the CC26x2. Connecting J1 to an XDS110 LaunchPad allows for debugging of the CC3230.

1.1.2 Sense on Power (SOP)

The CC3230 can be set to operate in four different modes, based on the state of the sense-on-power (SOP) lines. These SOP lines are pins 21, 34, and 35 on the CC3230 device. Table 1-1 describes the state of the device, and Figure 1-2 shows the SOP jumpers.

Table 1-1. SOP[2:0] (J11)

Binary Value	Function
000	Functional mode and 4-wire JTAG
001	Functional mode and 2-wire JTAG
010	Functional mode and flash programming
011	Factory default
100	Flash programming

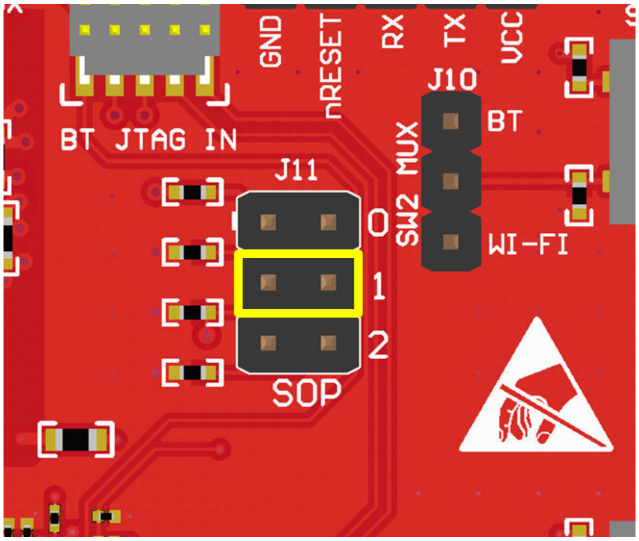


Figure 1-2. SOP Jumpers (Default Setting Shown)

1.1.3 Header Functions

1.1.3.1 BT RESET MUX Header

The BT Reset Mux Header (J8) is used for muxing the CC26x2 Reset signal. By default, the board comes with it muxed to SW4, giving the CC26x2 a separate and independent reset switch. This configuration is shown in Figure 1-3.

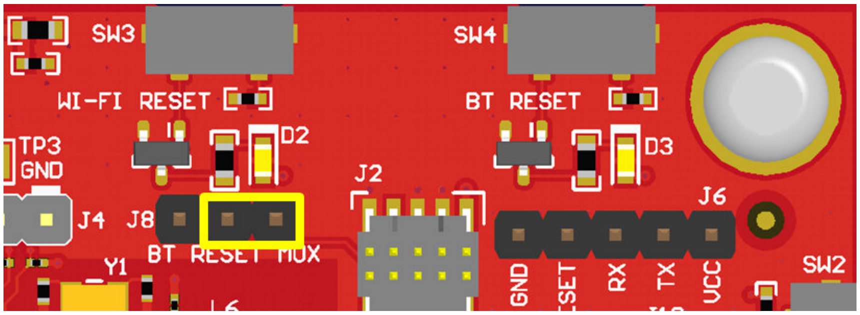


Figure 1-3. Independent Device Reset (Default Configuration)

To mux the CC26x2 Reset signal to SW3 and create a synchronized reset switch, simply move the shunt on J8 as shown in Figure 1-4.

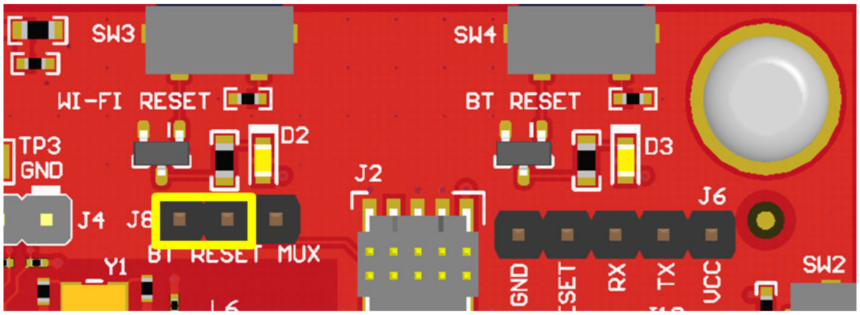


Figure 1-4. Synchronized Device Reset Configuration

1.1.3.2 CC3230 Programming Header (J6)

The CC3230 Programming Header (J6) breaks out all the necessary signals for flashing the device and using other UART-based tools, such as RadioTool. For J6 header pinout specifications, see Figure 1-5.

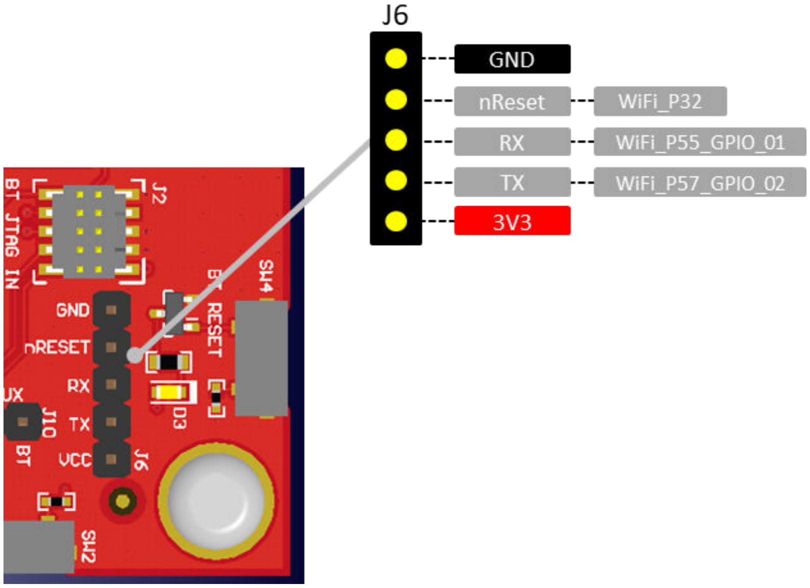


Figure 1-5. CC3230 Porgramming Header (J6)

1.1.3.3 CC3230 Debug Logs (J4)

MAC and NWP logs can help TI engineers to debug various types of issues. They can be read from a dedicated UART pin as an encrypted binary content. J4 breaks out the pins needed to capture these logs. For board location and pinout specifications, see Figure 1-6. For more information on capturing NWP logs, see the *SimpleLink™ Wi-Fi® CC3x20, CC3x3x Network Processor User's Guide*.

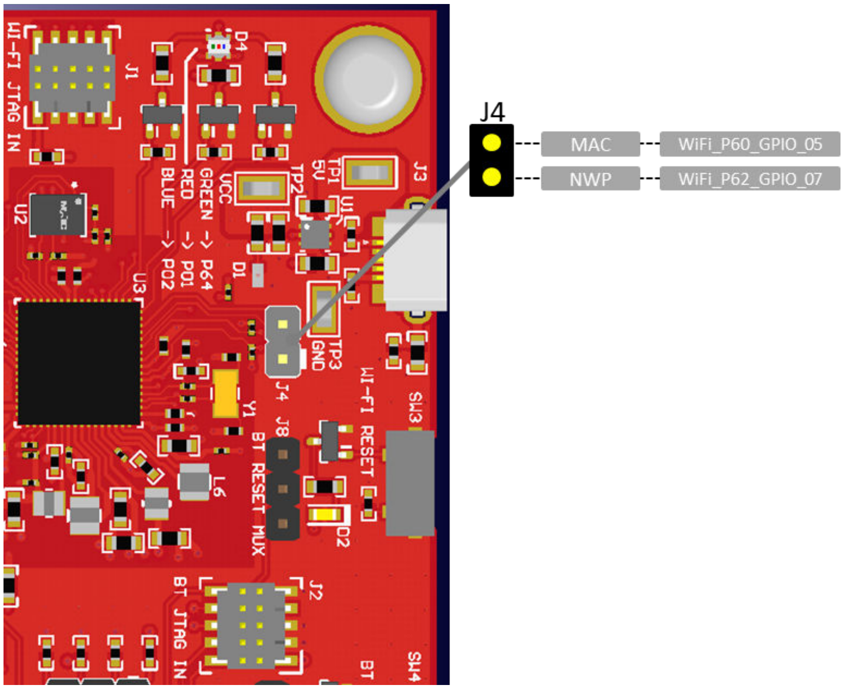


Figure 1-6. CC3230 Debug Logs Header (J4)

1.1.3.4 CC26x2 Auxiliary Header (J5)

The CC26x2 I/O controller configures I/O pins and maps peripheral signals to physical pins (DIOx). This provides the flexibility to to map different peripheral functions to the pins chosen by the user. J5 breaks out two signals for use by the user. Figure 1-7 shows the header location and its mapped DIOs.

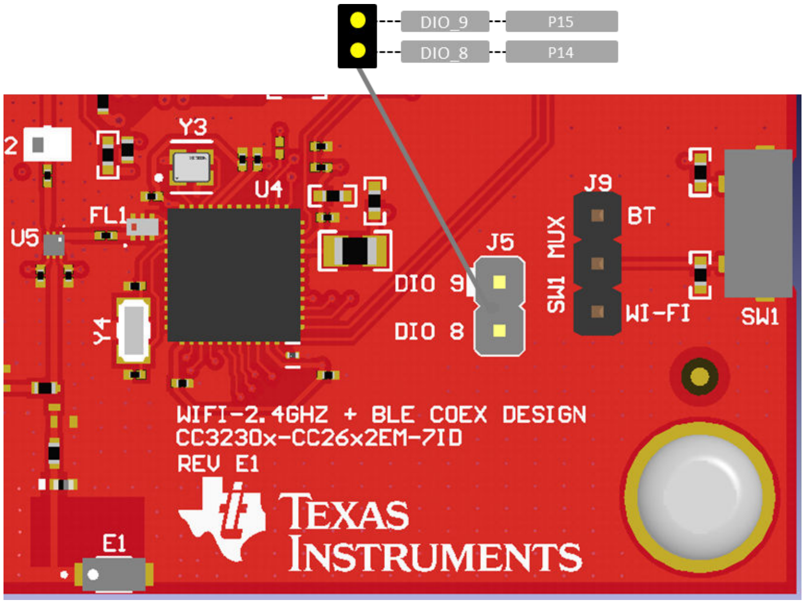


Figure 1-7. CC26x2 Auxiliary Header

1.1.3.5 SWx Mux Headers (J9 & J10)

The SWx MUX Headers (J9 and J10) are used for muxing the push-buttons to either the CC26x2 or the CC3230. By default, the board comes with both buttons muxed to the CC26x2. This configuration is shown in Figure 1-8.

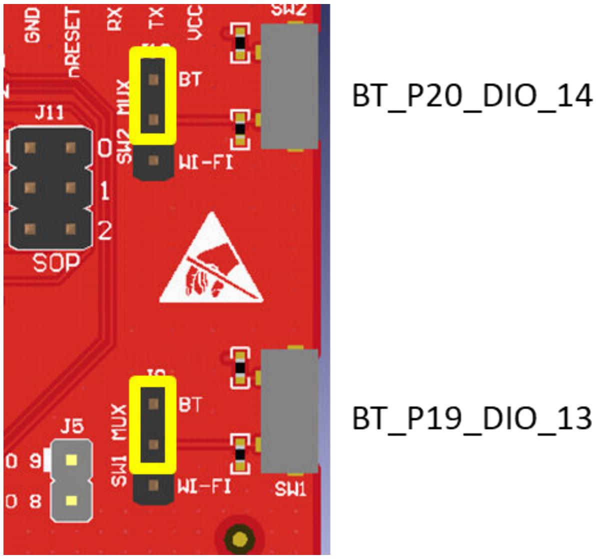


Figure 1-8. SWx MUX – CC26x2

To mux the push-buttons to the CC3230, simply move the shunts on J9 and J10 as shown in Figure 1-9.

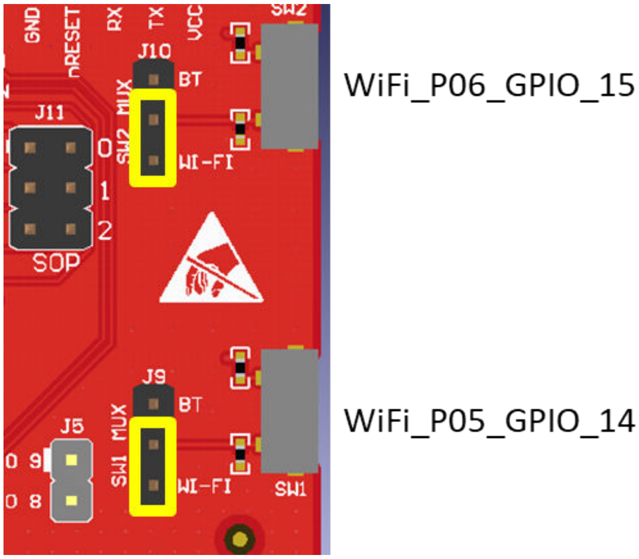


Figure 1-9. SWx MUX – CC3230

1.1.4 Push-Buttons and LED Indicators

Table 1-2 lists the push-button definitions

Table 1-2. Push Button Definitions

Reference	Use	Comments
SW1	CC3230x – GPIO_14((1)) CC26x2 – DIO_13	When pushed, GPIO_13 is pulled to VCC.
SW2	CC3230x – GPIO_15((1)) CC26x2 – DIO_14	When pushed, GPIO_13 is pulled to VCC.
SW3	CC3230x RESET OR Synchronized Device RESET((2))	This is used to reset the CC3230 device. This signal is also output on J6
SW4	CC26x2 RESET	This is used to reset the CC26x2 device

(1) For more information on SWx mux capabilities, see Section 1.1.3.5.

(2) For information on the reset mux capabilities, see Section 1.1.3.1.

Table 1-3 lists the LED indicators.

Table 1-3. LED Indicators

Reference	Color	Use	Comments
D1	Red	Power	Indicates when the 3.3V power is supplied to the board.
D2	Yellow	CC3230x RESET OR Synchronized Device RESET (1)	Indicates the state of the nRESET pin. If this LED is on, the device is functional.
D3	Yellow	CC26x2 RESET (1)	Indicates the state of the nRESET pin. If this LED is on, the device is functional.
D4	RGB	CC3230 - GPIO_10 CC3230 - GPIO_11 CC3230 - GPIO_09	On when the GPIO_xx is logic-1.

(1) For information on the reset mux capabilities, see Section 1.1.3.1.

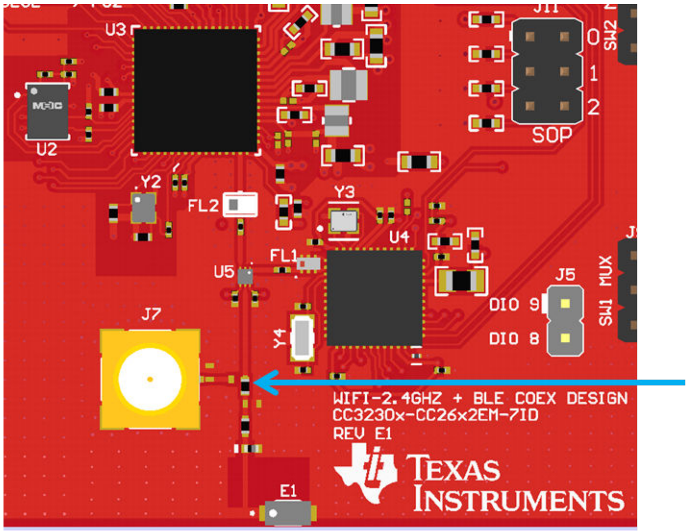
1.2 Power

1.2.1 USB Power

The CC3230x-CC26x2EM-7ID Coex Board is designed to work from the USB-provided power supply.

1.3 RF Connections

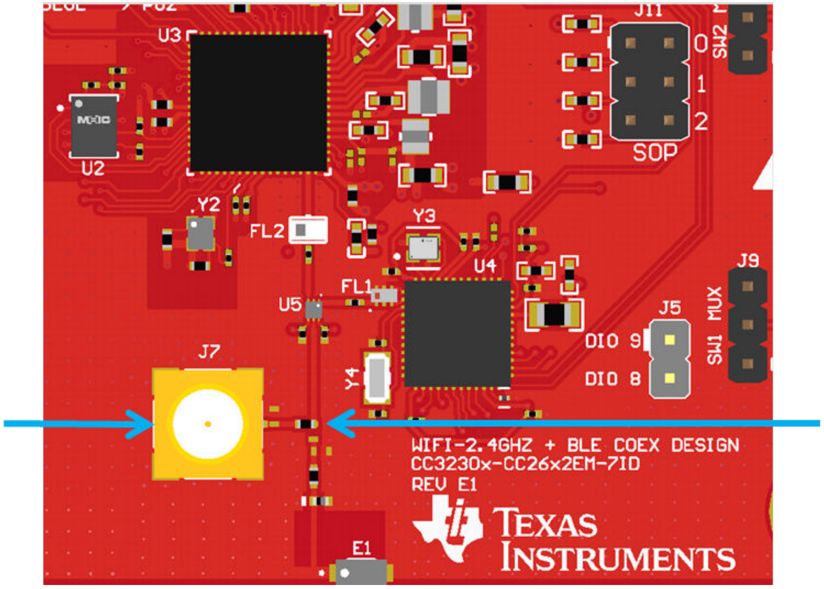
By default, the board ships with the RF signals routed to the on-board chip antenna, as shown in Figure 1-10.



Capacitor mounted for RF signal towards antenna

Figure 1-10. Using Onboard Antenna

An SMA connector provides a way to test in the lab using a compatible cable. A rework must be performed before this connector can be used; this involves swapping the position of a capacitor. The modified board would appear as in Figure 1-11.



SMA Connector

Capacitor mounted for RF signal towards SMA Connector

Figure 1-11. Board Modified for Conducted Measurements

2 Wi-Fi Test Procedure

2.1 Prerequisites

The user should have the following items:

- Hardware:
  - 1x CC3230x-CC26x2EM-7ID Coex Board
  - 1x LAUNCHXL-CC3235 OR any LaunchPad with XDS110-based JTAG emulation with serial port for flash programming
  - 2x Micro USB cables
  - 1x 10-pin JTAG cable, such as the FFSD-05-D-06.00-01-N
  - 2-5x female-to-female jumpers
- Software:
  - UniFlash v6.1.0 or newer
  - RadioTool v1.0.3.16 or newer
  - CC32xx SDK v4.40 or newer

2.2 Configure Board

2.2.1 Full Debugging, Programming, and UART Communication Option

The CC3230x-CC26x2EM-7ID Coex Board is designed to be used in conjunction with a LaunchPad development kit for debugging. Connecting these to an XDS110 LaunchPad shown in Figure 2-1 allows for full debugging, programming, and UART communication with the CC3230.

1. Disconnect the debug isolation jumpers on your LaunchPad.
2. Connect the Arm® 10-pin JTAG cable to XDS110 OUT header on your LaunchPad.
3. Connect the other end of the Arm10-pin cable to the Wi-Fi® JTAG IN (J1) on the Coex Board.
4. Connect the 2-pin jumper cable to the top pins of RX and TX (purple wire to RX, white wire to TX).
5. Connect the other end of the 2-pin jumper to the RX and TX pins on J6 on the Coex Board (Purple to RX, white to TX).
6. Connect both boards to your PC.

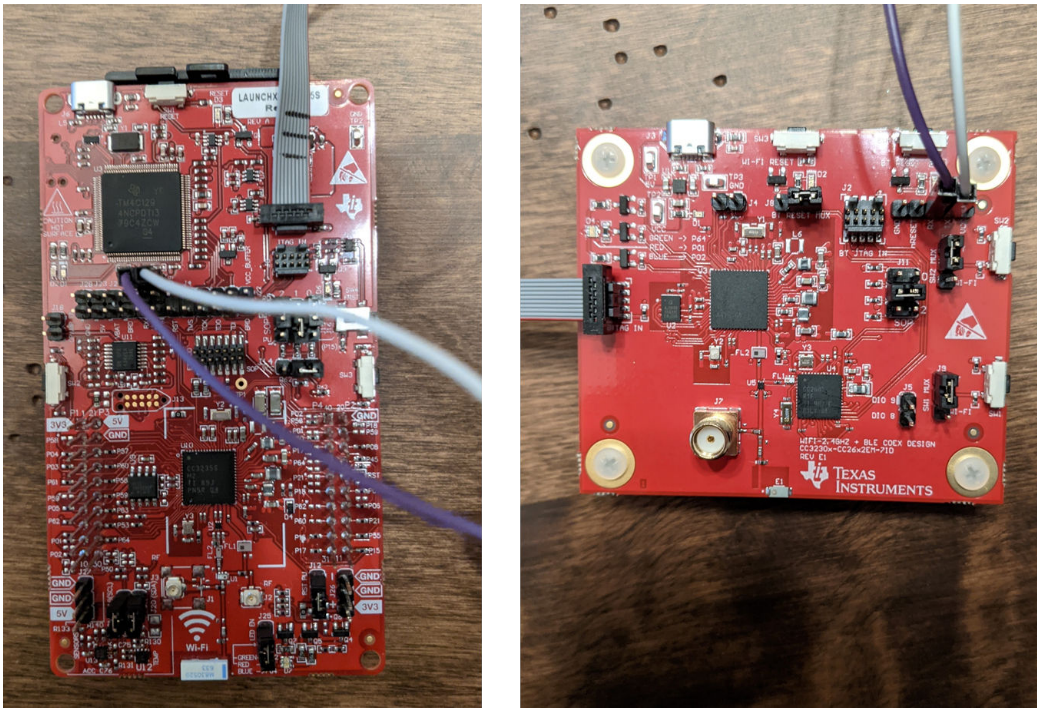


Figure 2-1. Wiring for Debugging, Programming, and UART Communication

2.2.2 Programming and UART Communication Option

If debugging capability is not needed or you do not have a 10-pin JTAG cable, there is an alternative way to set the boards up which omit the use of the 10-pin JTAG cable.

1. Disconnect the debug isolation jumpers on your LaunchPad.
2. Connect the 5-pin jumper cable to the boards as shown in Figure 2-2.
3. Connect both boards to your PC.

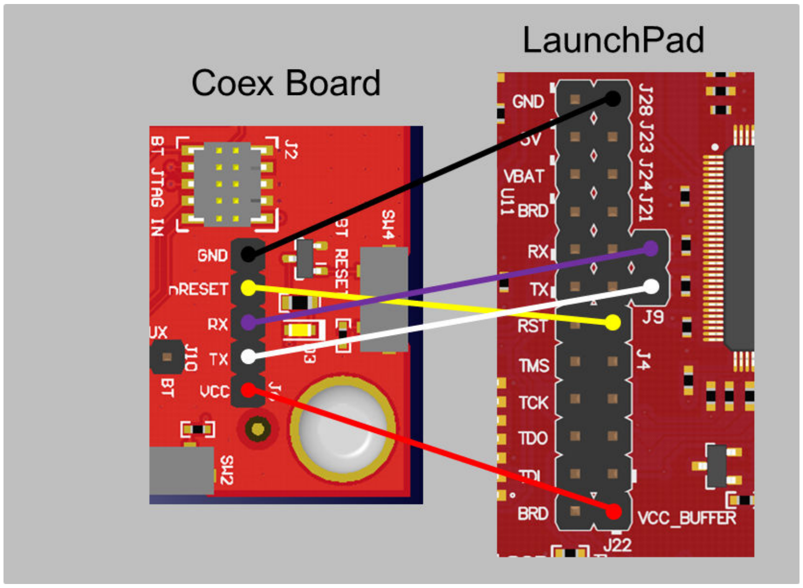


Figure 2-2. Programming and UART Communication – Wiring Diagram

2.3 Programming the Wi-Fi Device

The Coexistence feature does not work on the CC3230, if it is in Test Mode. The RadioTool project has to be modified to control the RF Coex switch and give the CC3230 control of the RF path.

2.3.1 Modifying the RadioTool Source Project

1. Open the RadioTool Source Project in CCS: (*C:\ti\CC3xxx_RadioTool_xx.xx.xx.xx\Source Files\Source Projects\radiotool_CC3235S_LAUNCHXL_tirtos_ccs*)
2. In radiotool.sysconfig, navigate to the board view and select the option to “use custom board” as shown in Figure 2-3.
3. Select the GPIO module in the Software view and set up GPIO_CONFIG_0 as shown in Figure 2-4.
4. Add the line of code to 'mcu_radiotool.c', shown in Figure 2-5.
5. Build the project.

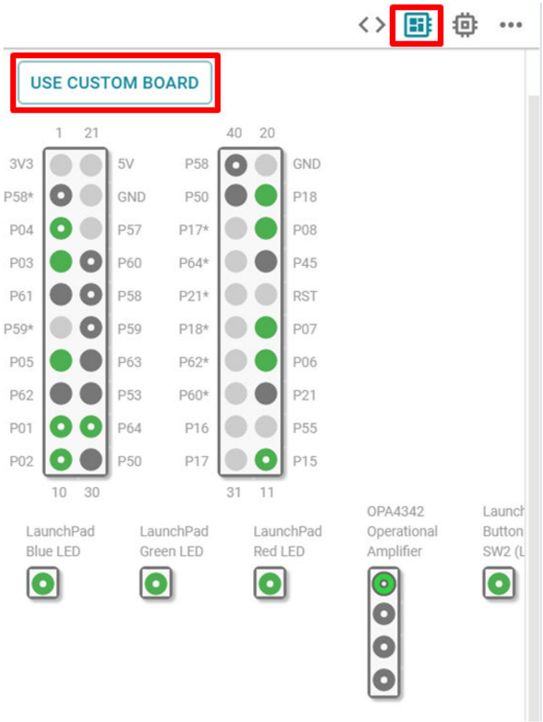
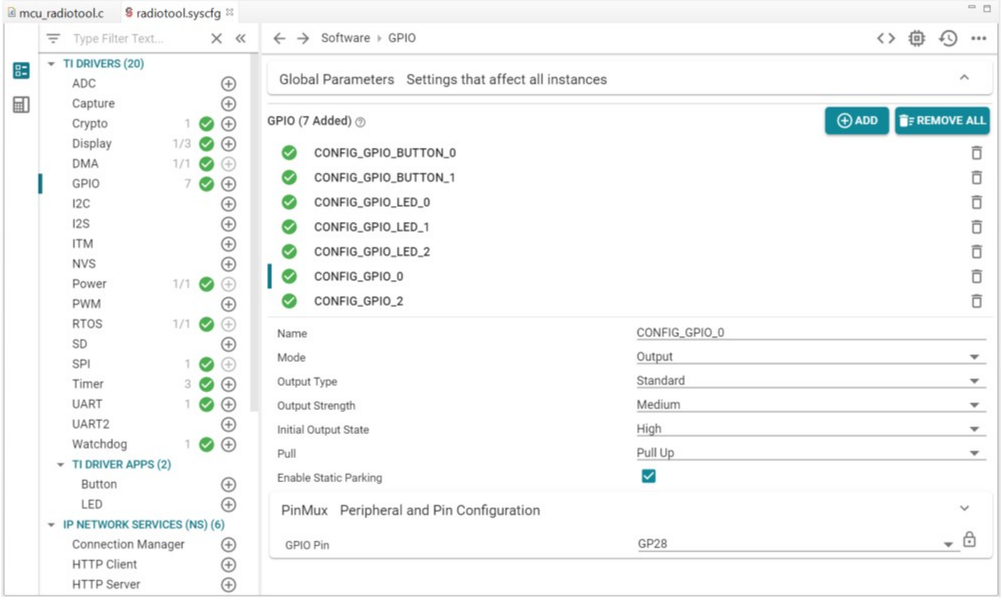
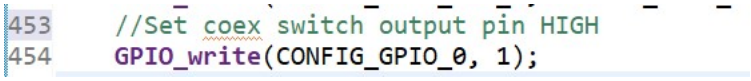


Figure 2-3. sysconfig Custom Board



Name: CONFIG_GPIO_0
Mode: Output
Output Type: Standard
Output Strength: Medium
Initial Output State: High
Pull: Pull Up
Enable Static Parking: ☑
GPIO Pin: GP28

Figure 2-4. sysconfig - Configure Board

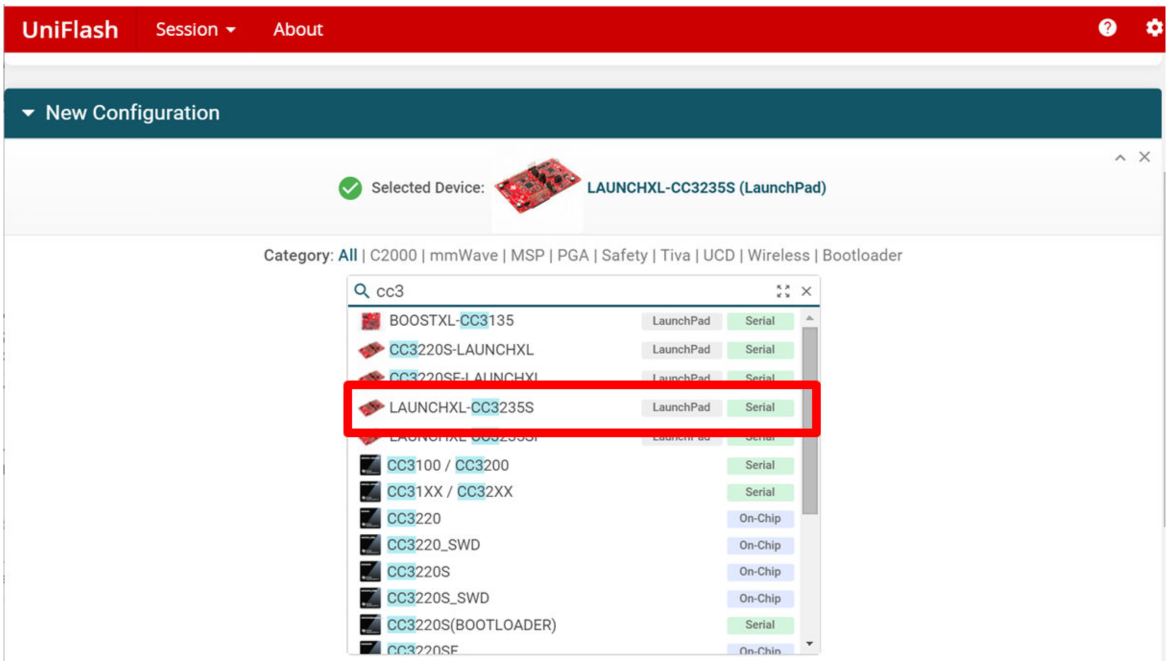


```
	//Set coex switch output pin HIGH
	GPIO_write(CONFIG_GPIO_0, 1);
```

Figure 2-5. Radio Tool

2.3.2 Flashing the New RadioTool MCU Image

1. Type "CC3" into the search bar and you'll see a number of options. Select the CC3220S-LAUNCHXL or CC31XX / CC32XX Serial device and click Start Image Creator to launch the ImageCreator tool.

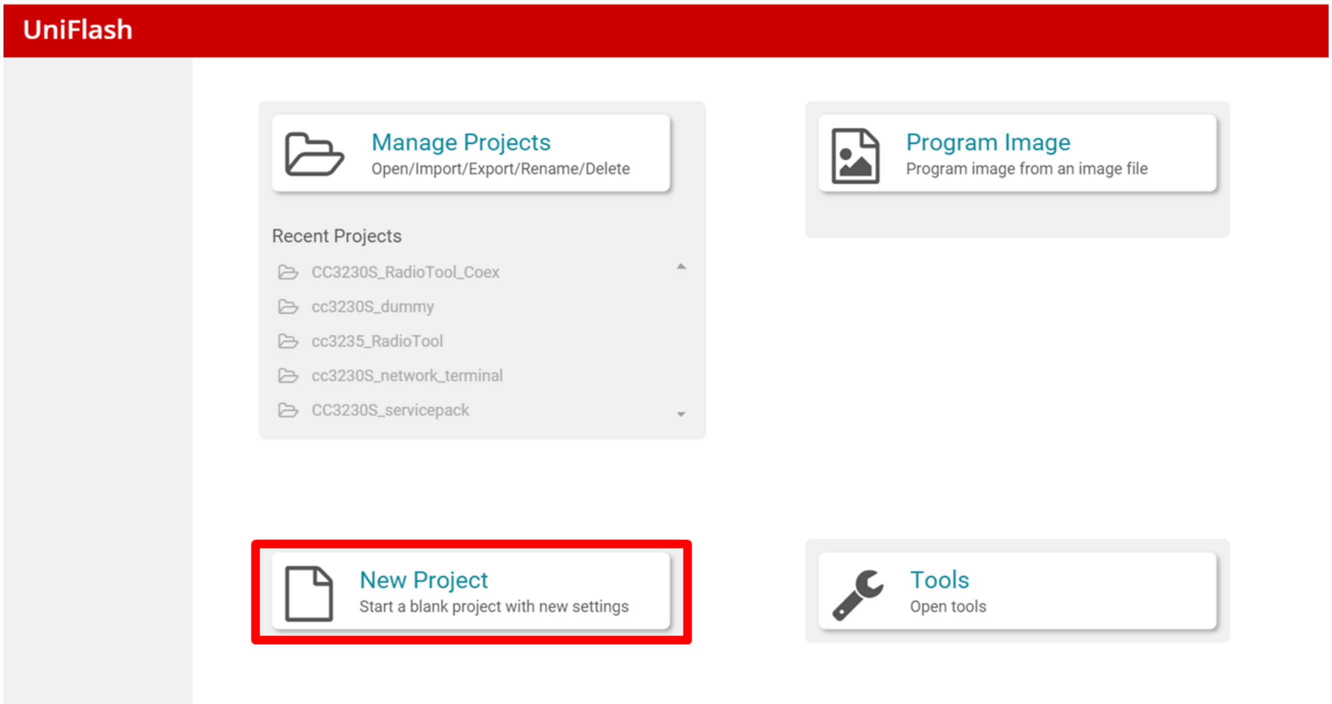

UniFlash
Session
About
New Configuration
Category: All | C2000 | mmWave | MSP | PGA | Safety | Tiva | UCD | Wireless | Bootloader
BOOSTXL-CC3135 LaunchPad Serial
CC3220S-LAUNCHXL LaunchPad Serial
CC3220SF-LAUNCHXL LaunchPad Serial
LAUNCHXL-CC3235S LaunchPad Serial
LAUNCHXL-CC3235SF LaunchPad Serial
CC3100 / CC3200 Serial
CC31XX / CC32XX Serial
CC3220 On-Chip
CC3220_SWD On-Chip
CC3220S On-Chip
CC3220S_SWD On-Chip
CC3220S(BOOTLOADER) Serial
CC3220SF On-Chip

Selected Device: LAUNCHXL-CC3235S (LaunchPad)

cc3

Figure 2-6. Select Device and Launch ImageCreator

2. Select New Project from the home page.



UniFlash

Manage Projects
Open/Import/Export/Rename/Delete

Program Image
Program image from an image file

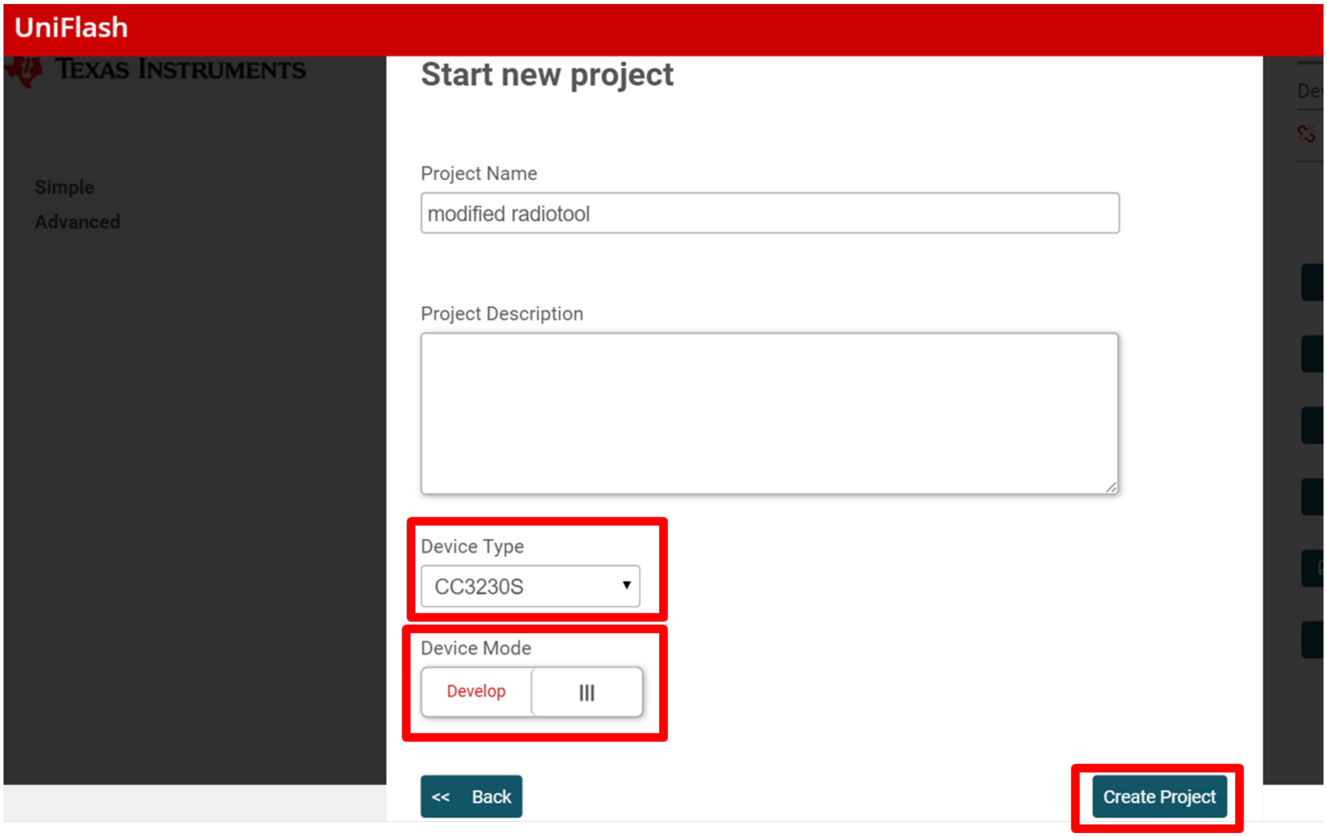
Recent Projects
- CC3230S_RadioTool_Coex
- cc3230S_dummy
- cc3235_RadioTool
- cc3230S_network_terminal
- CC3230S_servicepack

New Project
Start a blank project with new settings

Tools
Open tools

Figure 2-7. Create New Project in Image Creator

3. The next screen will ask you to name the project and select a Device Type and Device Mode. Be sure to select CC3230S for the device type and Develop for the device mode.



UniFlash

Start new project

Project Name: modified radiotool

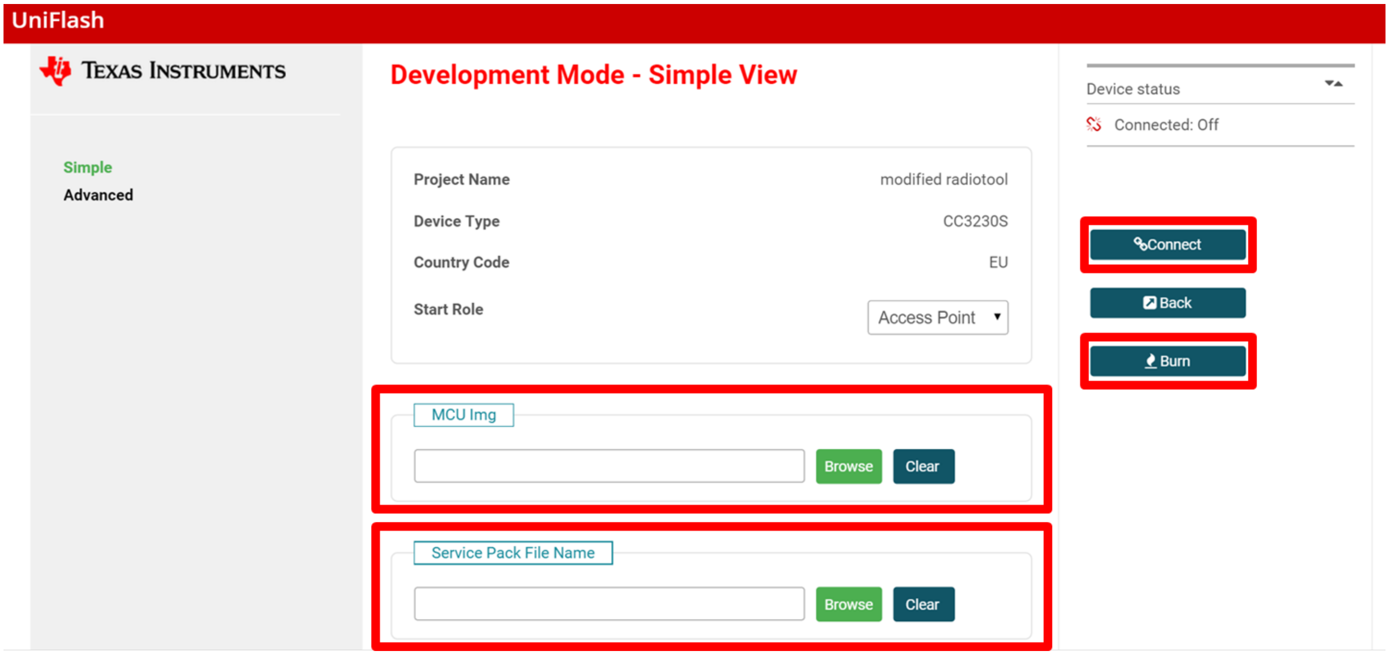
Project Description:

Device Type: CC3230S

Device Mode: Develop

Figure 2-8. Set up New Project

4. To add the servicepack to the project, click Browse and navigate to the sp_xxxx_xxxx_xxxx.bin file provided in the latest SimpleLink™ CC32xx SDK.
5. To add the MCU image to the project, click Browse and navigate to the radiotool_CC3235S_LAUNCHXL_tirtos_ccs.bin file created in Section 2.3.
6. Click the Connect button on the right.

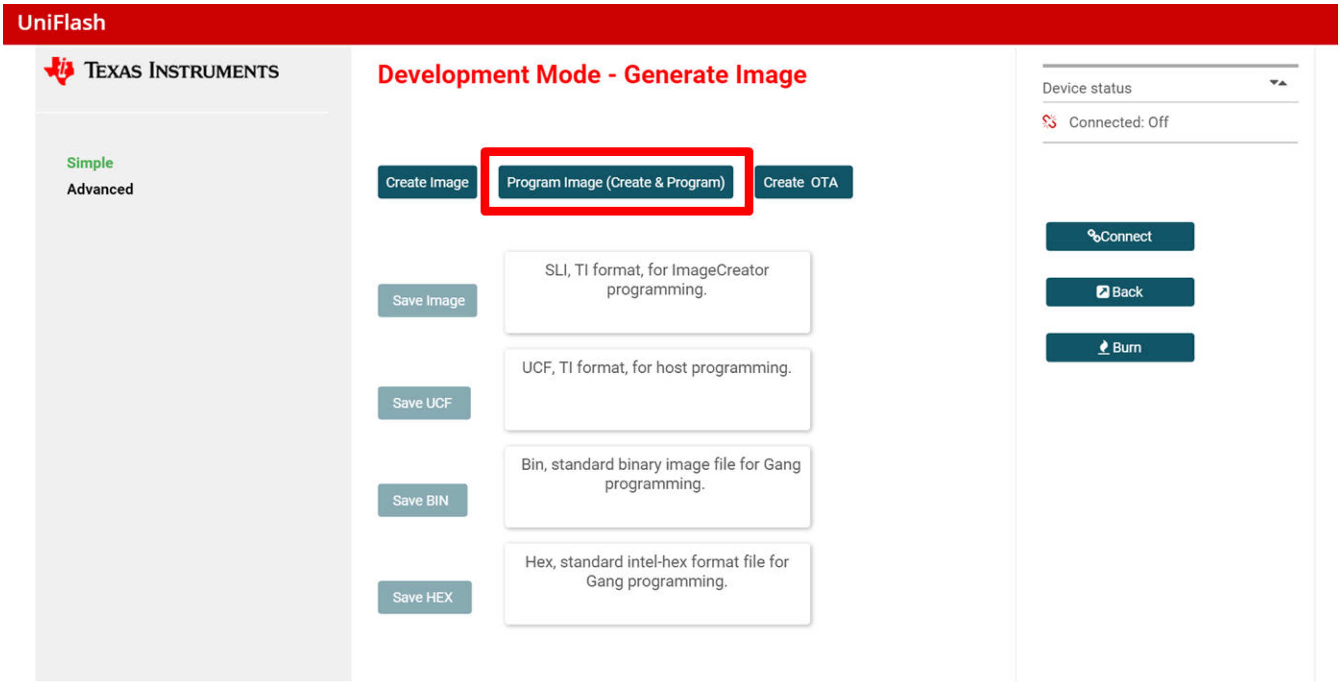


Development Mode - Simple View

Project Name: modified radiotool
Device Type: CC3230S
Country Code: EU
Start Role: Access Point

Figure 2-9. Load and Program ServicePack and MCU Image

7. The Burn button on the right will take you to the Generate Image page where you can create and program an image to the device. Select Program Image to generate a project image.



Development Mode - Generate Image

Save Image: SLI, TI format, for ImageCreator programming.
Save UCF: UCF, TI format, for host programming.
Save BIN: Bin, standard binary image file for Gang programming.
Save HEX: Hex, standard intel-hex format file for Gang programming.

Figure 2-10. Load and Program ServicePack and MCU Image

At this point the device is ready for RadioTool using a UART connection. For help setting up RadioTool, see the *SimpleLink™ Wi-Fi® CC3x20, CC3x3x Radio Tool User's Guide*.

3 Bluetooth Low Energy Test Procedure

3.1 Prerequisites

- You should have the following items:
- Hardware
  - 1x CC3230x-CC26x2EM-7ID Coex Board
  - 1x LAUNCHXL-CC3235 OR any LaunchPad with XDS110-based JTAG emulation with serial port for flash programming
  - 2x Micro USB cables
  - 1x 10-pin JTAG cable, such as the FFSD-05-D-06.00-01-N
  - 2x female-to-female jumpers
- Software
  - UniFlash v6.1.0 or newer
  - SmartRF™ Studio 7
  - CC32xx SDK v4.40 or newer

3.2 Program the CC3230

In order to give the CC26x2 control of the RF path, the CC3230 needs to be programmed with a “dummy image” that pin parks the GPIOs that control the coex switch.

3.2.1 Configure the Board

Follow the steps outlined in Section 2.2 to add the lines of code shown below 'empty.c' in the ***empty_CC3235S_LAUNCHXL_tirtos_ccs*** project.

```
/* Set the Coex Switch Ouput and Input Pins to give the BLE Device control of the RF path */
GPIO_setConfig (GPIO_COEX_OUT, GPIO_CFG_OUT_STD | GPIO_CFG_OUT_LOW);
GPIO_setConfig (GPIO_COEX_IN, GPIO_CFG_OUT_STD | GPIO_CFG_OUT_HIGH);

GPIO_write(GPIO_COEDX_OUT, 0);
GPIO_write(GPIO_COEX_IN, 1);
```

3.2.2 Program the “Dummy Image”

Follow the steps outlined in Section 2.3 to flash the ***empty_CC3235S_LAUNCHXL_tirtos_ccs.bin*** file to the device.

3.3 Configure the CC26x2 Board

The CC3230x-CC26x2EM-7ID Coex Board is designed to be used in conjunction with a LaunchPad development kit for debugging. Connecting these to an XDS110 LaunchPad shown in Figure 3-1 allows for full debugging, programming, and UART communication with the CC2642R. If UART communication to the PC isn’t needed, simply skip steps 4 and 5.

1. Disconnect the debug isolation jumpers on your LaunchPad.
2. Connect the Arm 10-pin JTAG cable to XDS110 OUT header on your LaunchPad.
3. Connect the other end of the Arm10-pin cable to the BT JTAG IN (J2) on the Coex Board.
4. Connect the 2-pin jumper cable to the top pins of RX and TX.
5. Connect the other end of the 2-pin jumper to DIO_8 and DIO_9 on J5 on the Coex Board (orientation does not matter, this will be configured in software).
6. Connect both boards to your PC.

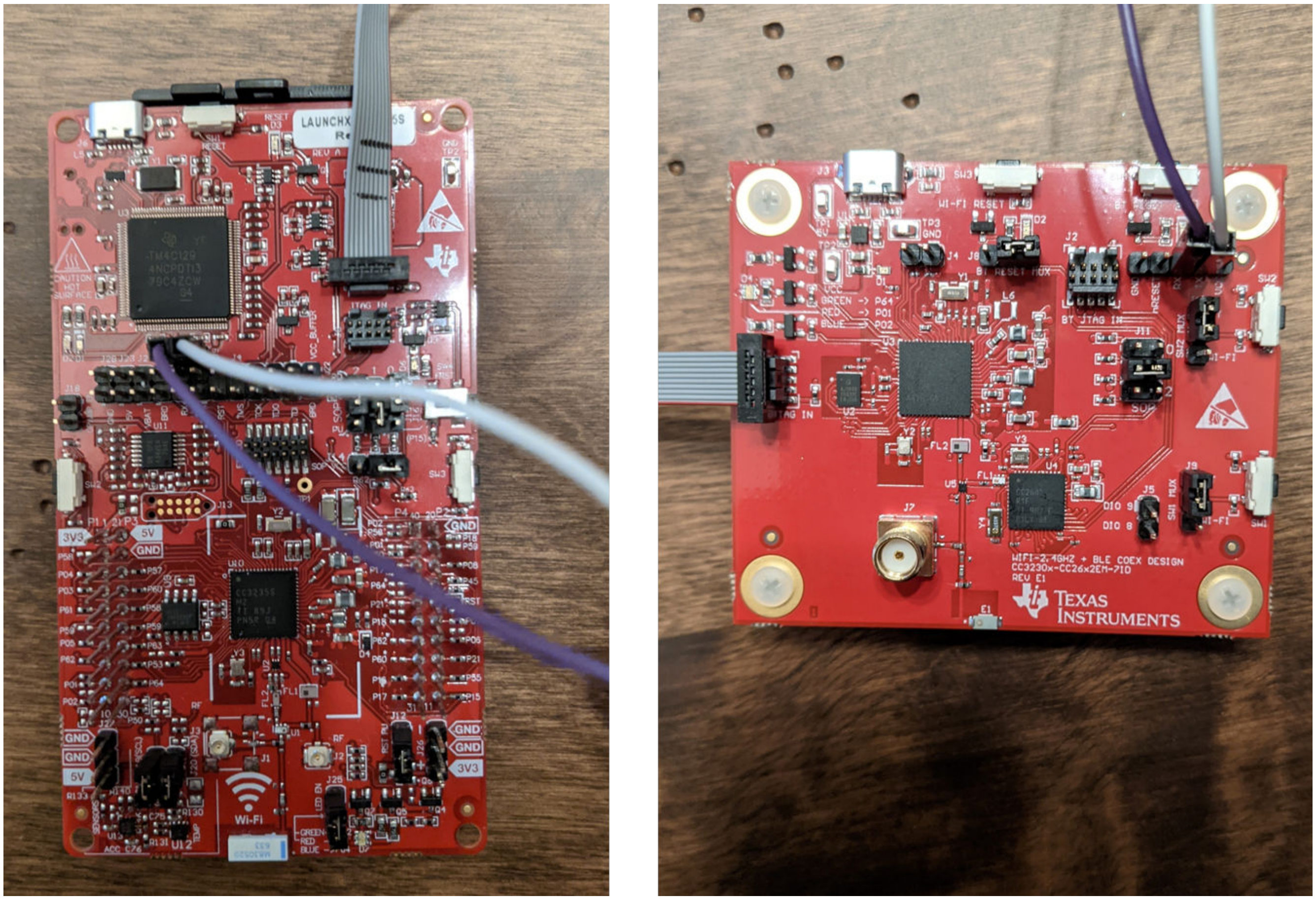


Figure 3-1. Wiring for Full Debugging, Programming, and UART Communication

At this point, the board is ready to perform Bluetooth® Low Energy testing via Smart RF Studio7. For help setting up SmartRF Studio 7, see the SmartRF Studio 7 Documentation.