# CC3200 SimpleLink™ Wi-Fi® and Internet of Things Solution With MCU LaunchPad™ Hardware

# User's Guide

Literature Number: SWRU372C
June 2014–Revised March 2020

# Contents

## List of Figures



## List of Tables

# CC3200 SimpleLink™ Wi-Fi® and Internet of Things Solution With MCU LaunchPad™ Hardware

## 1 Introduction

### 1.1 CC3200 LaunchPad Development Kit

The high-performance CC3200 is the industry's first single-chip microcontroller (MCU) with built-in Wi-Fi® connectivity for the LaunchPad™ ecosystem. Created for the Internet of Things (IoT), the SimpleLink Wi-Fi CC3200 device is a wireless MCU that integrates a high-performance Arm® Cortex®-M4 MCU allowing customers to develop an entire application with a single IC. With on-chip Wi-Fi, internet, and robust security protocols, no prior Wi-Fi experience is needed for faster development.

The CC3200 LaunchPad development kit is a low-cost evaluation platform for Arm Cortex-M4F-based microcontrollers. The LaunchPad kit design highlights the CC3200 Internet-on-a-chip™ solution and Wi-Fi capabilities. The CC3200 LaunchPad kit also features programmable user buttons, RGB LED for custom applications, and onboard emulation for debugging. The stackable headers of the CC3200 LaunchPad XL interface demonstrate how easy it is to expand the functionality of the LaunchPad kit when interfacing with other peripherals on many existing BoosterPack™ plug-in modules, such as graphical displays, audio codecs, antenna selection, environmental sensing, and more. Figure 1 shows a photo of the CC3200 LaunchPad kit.

Free software development tools are also available, including TI's Eclipse-based Code Composer Studio™ and IAR Embedded Workbench®. More information about the LaunchPad kit, the supported BoosterPack modules, and the available resources can be found at TI's LaunchPad portal. Also visit the CC3200 Wiki page for design resources and example projects.

**NOTE:** The antennas used for this transmitter must be installed to provide a separation distance of at least 20 cm from all persons and must not be co-located or operating in conjunction with any other antenna or transmitter.

**NOTE:** All figures and references in this document apply to the Rev3.2. Most of the document also applies to the Rev4.1, unless otherwise stated. For the exact list of changes made across board revisions, refer to Section 2.8.2.

### 1.2 Key Features

- CC3200, SimpleLink Wi-Fi, Internet-on-a-chip solution with integrated MCU
- 40-pin LaunchPad standard that leverages the BoosterPack ecosystem
- FTDI-based JTAG emulation with serial port for flash programming
- Two buttons and three LEDs for user interaction
- Backchannel universal asynchronous receiver/transmitter (UART) through USB to PC
- On-board chip antenna with U.FL for conducted testing
- On-board accelerometer and temperature sensor for out-of-box demo
- Micro USB connector for power and debug connections

## 1.3 What's Included

### 1.3.1 Kit Contents

- CC3200 LaunchPad development kit
- Micro USB cable
- Quick start guide

## 1.4 FCC/IC Regulatory Compliance

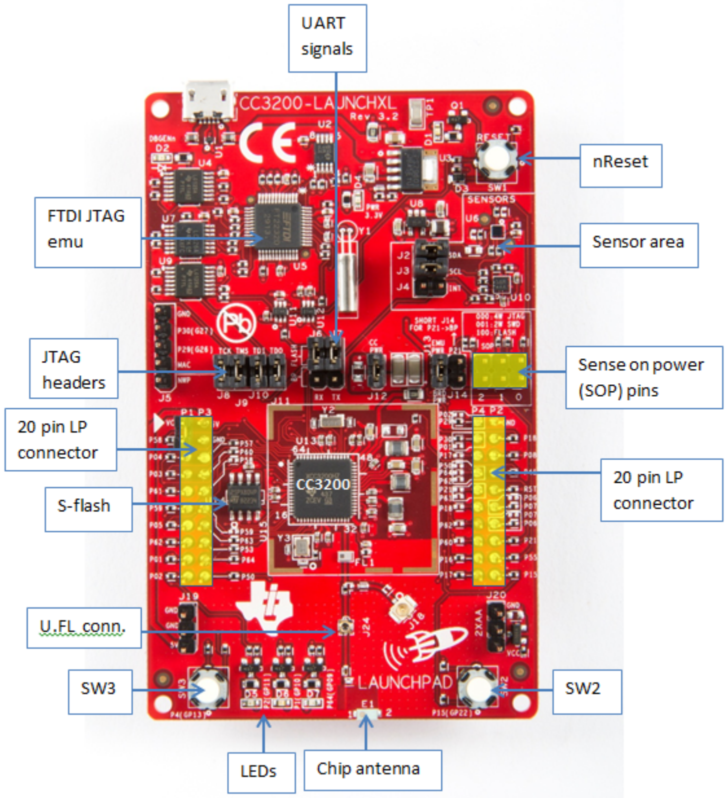
The CC3200 SimpleLink Wi-Fi and IoT solution with MCU LaunchPad kit hardware is FCC Part 15 and IC ICES-003 Class A compliant.

## 1.5 Trademarks

LaunchPad, Internet-on-a-chip, BoosterPack, Code Composer Studio, E2E are trademarks of Texas Instruments.
Arm, Cortex are registered trademarks of Arm Limited.
IAR Embedded Workbench is a registered trademark of IAR Systems AB.
Wi-Fi is a registered trademark of Wi-Fi Alliance.
All other trademarks are the property of their respective owners.

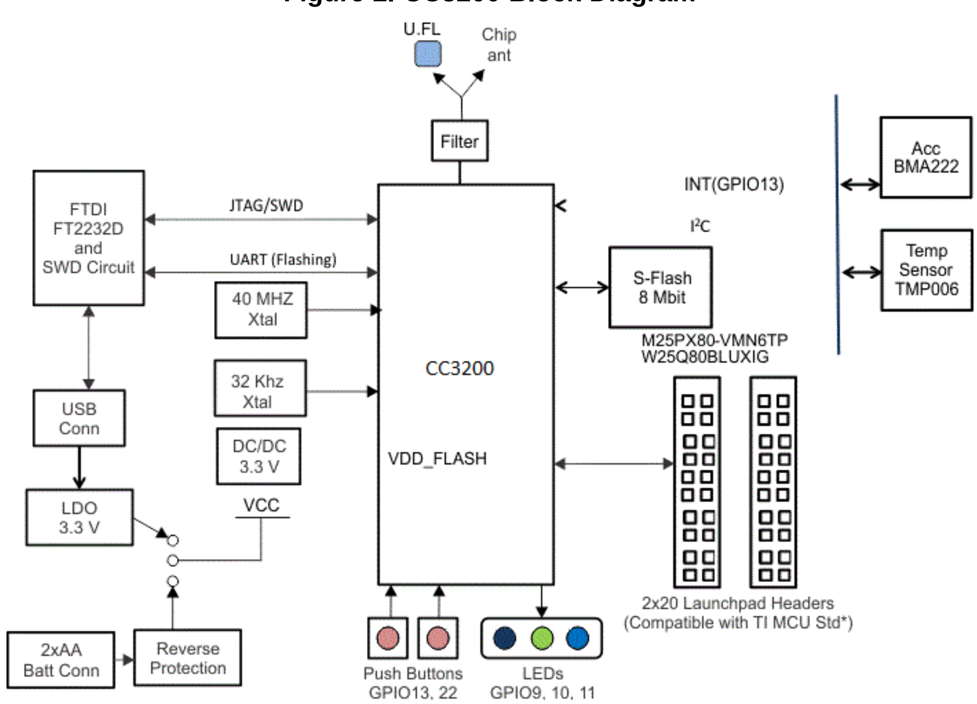
# 2 Hardware Description

**Figure 1. CC3200 LaunchPad Development Kit Overview**




## 2.1 Block Diagram

**Figure 2. CC3200 Block Diagram**



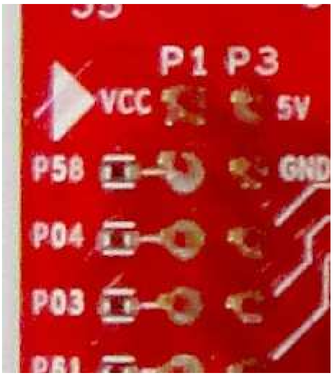
## 2.2 Hardware Features

- CC3200, SimpleLink Wi-Fi, Internet-on-a-chip solution with integrated MCU
- 40-pin LaunchPad standard that leverages the BoosterPack ecosystem
- FTDI-based JTAG emulation with serial port for flash programming
- Supports both 4-wire JTAG and 2-wire SWD
- Two buttons and three LEDs for user interaction
- Virtual COM port UART through USB on PC
- On-board chip antenna with U.FL for conducted testing
- On-board accelerometer and temperature sensor for out-of-box demo, with option to isolate them from the inter-integrated circuit ($I^2C$) bus
- Micro USB connector for power and debug connections
- Headers for current measurement and external JTAG connection
- Bus-powered device with no external power required for Wi-Fi
- Long-range transmission with highly optimized antenna (200 m typical in open air with a 6-dBi antenna AP)
- Can be powered externally, with 2xAA or 2xAAA alkaline batteries working down to 2.3 V typical

## 2.3 Connecting a BoosterPack Plug-in Module

A compatible BoosterPack module can be stacked on top of the LaunchPad kit using the 2x20 pin connectors. Note that the connectors do not have a key to prevent the misalignment of the pins or reverse connection. Ensure that VCC and 5V pins are aligned with the BoosterPack module header pins. On the CC3200 LaunchPad kit, a small white triangle symbol is provided near Pin-1 (see Figure 3) to orient all BoosterPack modules. This same marking, provided on compatible BoosterPack modules, must be aligned before powering up the boards.

**Figure 3. Pn-1 Marking on the LaunchPad Kit (White Triangle)**



## 2.4 Jumpers, Switches, and LEDs

### 2.4.1 JTAG Headers

The headers are provided on the board to isolate the CC3200 device from the mounted FTDI JTAG emulator. These jumpers are shorted by default when the board is shipped from TI. To connect an external emulator, remove these jumpers and place the external emulator on the pins closer to the CC3200 device. When a battery is used, disconnect all the JTAG headers to prevent any reverse leakage current.

**Figure 4. JTAG Headers**


**Table 1. JTAG Headers**

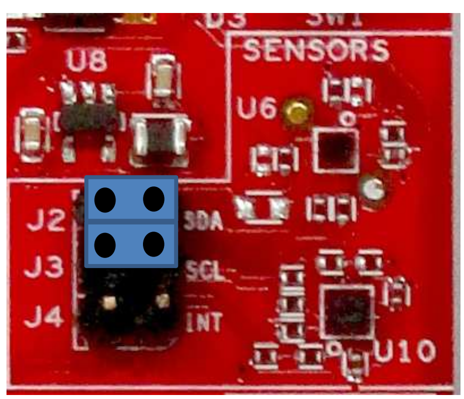
| Reference | Usage | Comments |
|---|---|---|
| J8 (TCK) [(1)] | JTAG | Short: Routes the on-board emulator to the CC3200 |
| J9 (TMS) [(1)] | | |
| J10 (TDI) | | Open: Isolate the on-board emulator from the CC3200. |
| J11(TDO) | | |

(1) For the SWD mode, only TCK and TMS must be shorted to the CC3200.

### 2.4.2 I²C Connections

The board features an accelerometer and a temperature sensor for the out-of-box demo. These are connected to the I²C bus and can be isolated using the jumpers provided. By removing J2 and J3, the accelerometer and the temperature sensors are isolated from the I²C bus. This also removes any pullup resistor from the I²C bus.

**Figure 5. I²C Connections**



#### 2.4.2.1 Jumper Settings

**Table 2. Jumper Settings**

| Reference | Usage | Comments |
|---|---|---|
| J2 | I²C SDA | Short: Connect the CC3200 I²C bus to the on-board sensors with pullup<br>Open: Isolate the sensors from the CC3200 |
| J3 | I²C SCL | Short: Connect the CC3200 I²C bus to the on-board sensors with pullup<br>Open: Isolate the sensors from the CC320 |
| J4 | INT | Short: Connect the accelerometer interrupt to the CC3200 on GPIO13 |

#### 2.4.2.2 Default I²C Address

**Table 3. Default I²C Addresses**

| Sensor Type | Ref | Part Number | Slave Address |
|---|---|---|---|
| Temp sensor | U6 (1) | TMP008 | 0x41 |
| Accelerometer | U10 | BMA222 | 0x18 |

(1) U6 is DNP on newer CC3200-LAUNCHXL builds.

### 2.4.3 Power Connections

The board can be powered by using the on-board micro USB connector. An on-board LDO provides 3.3 V for the CC3200 and the rest of the board to operate. This supply can be isolated from the LDO using the jumpers on the board.

**Table 4. Jumper Settings**

| Reference | Usage | Comments |
|---|---|---|
| J12 | Current measurement | Measures the current flowing into the CC3200 device. |
| J13 | Board power | Short: Supply the board power from the on-board LDO.<br>Open: Supply the board power from the J20 (battery connector) |
| J19 | 5-V power | 5-V output from the USB VBUS (has a diode drop of up to 0.4 V) |
| J20 | 3.3-V power input | Can be used to power the board from an external 2xAA battery pack. It has in-built reverse voltage protection to prevent the battery from being plugged in the reverse manner. |

### 2.4.4 UART Signals

The board supports a USB-based virtual COM port, which is used on the FTDI device FT2232D. There are two ports on the FT2232: the first port is dedicated for the emulation (JTAG/SWD) and the second port is used for the virtual COM port. The UART can also be routed to the 20-pin connector and the selection is performed using jumpers on the board.

**Figure 6. UART Signals**

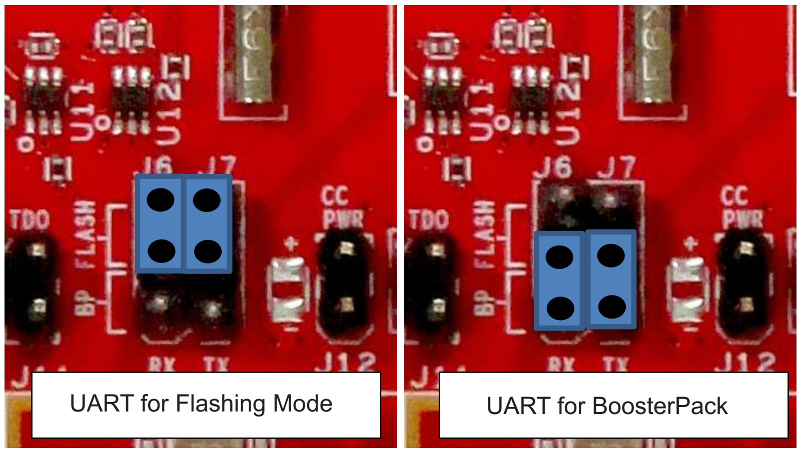



**Table 5. UART Signals**

| Reference | Usage | Comments |
|---|---|---|
| J6, J7 | UART for flash programming | Short 1-2: Route the signals to the 20-pin connector.<br>Short 2-3: Route the signals to the FTDI for flash programming. |

### 2.4.5 Sense on Power

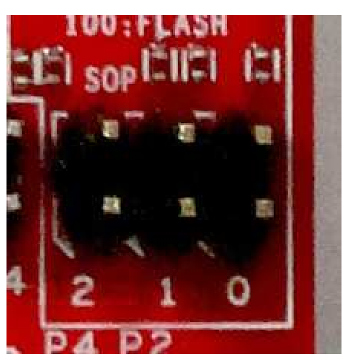
The CC3200 can be set to operate in three different modes based on the state of the sense-on-power (SOP) lines. These are pins 21, 34, and 35 on the CC3200 device. The state of the device is described in Table 6.

**Table 6. SOP Lines**

| Usage | Comments |
|---|---|
| SOP[2:0] | 100 = Flash programming<br>000 = Functional mode + 4-Wire JTAG<br>001 = Functional mode + 2-Wire JTAG |

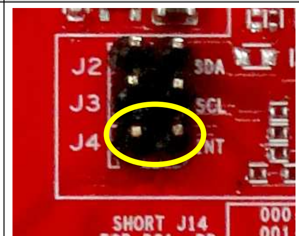
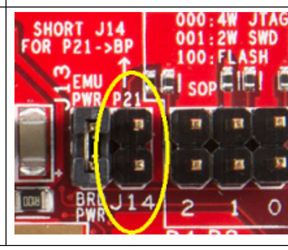
SOP[2:0] corresponds to J15, J16, and J17 in the LaunchPad kit schematic design.

**Figure 7. SOP Jumpers**



### 2.4.6 Other Miscellaneous

**Table 7. Miscellaneous Settings**

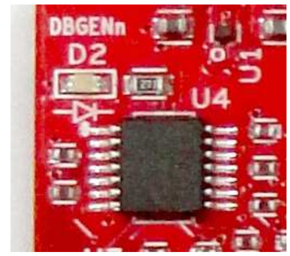
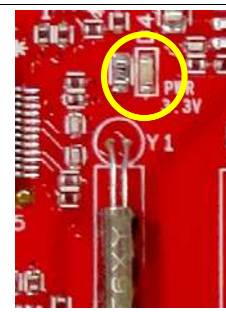
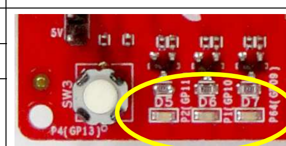
| Reference | Usage | Comments | |
|---|---|---|---|
| J4 | Accelerometer interrupt | Short = Route the accelerometer sensor interrupt to the GPIO_13<br>Open = Isolates the interrupt to the GPIO_13 |  |
| J5 | Debug header | To observe the network processor (NWP), MAC logs. |  |
| J14 | SOP2 isolation | Isolate SOP2 (GPIO_25) from the 20-pin connector |  |

### 2.4.7 Push Buttons and LEDs

**Table 8. Push Buttons**

| Reference | Usage | Comments | |
|---|---|---|---|
| SW1 | RESET | This is used to RESET the CC3200 device. This signal is also output on the 20-pin connector to RESET any external BoosterPack module which may be stacked. |  |
| SW2 | GPIO_22 | When pushed, the GPIO_22 is pulled to VCC. |  |
| SW3 | GPIO_13 | When pushed, the GPIO_13 is pulled to VCC. |  |

## Table 9. LEDs

| Reference | Colo4 | Usage | Comments | |
|---|---|---|---|---|
| D1 | Yellow | nRESET | This LED indicates the state of nRESET pin. If this LED is glowing, the device is functional. |   |
| D2 | Green | Debug | This LED glows whenever debugging is enabled over the JTAG. |   |
| D4 | RED | Power | Indicates when the 3.3-V power is supplied to the board. |   |
| D5 | GREEN | GPIO_11 (1) | Glows when the GPIO is logic-1 |   |
| D6 | YELLOW | GPIO_10 (1) | Glows when the GPIO is logic-1 | |
| D7 | RED | GPIO_09 | Glows when the GPIO is logic-1 | |

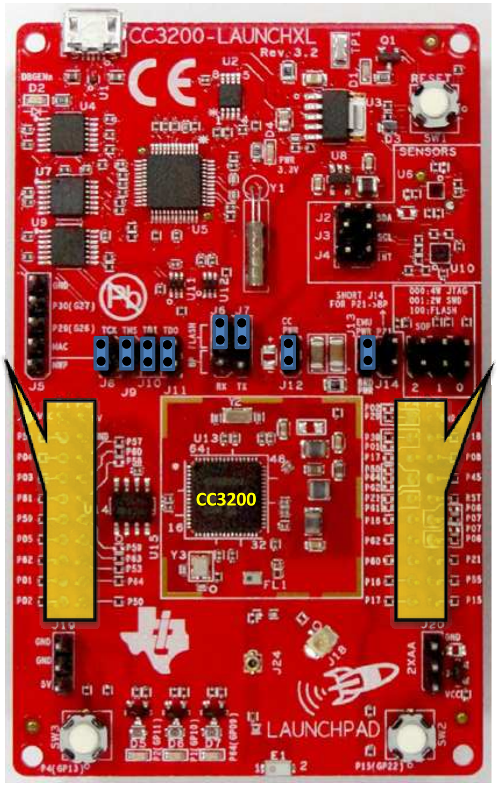
(1) GPIO_10 and GPIO_11 are also used as $I^2C$. Thus, when the pullups are enabled, the LEDs glow.

### 2.4.8 2x20 Pin Connector Assignment

The signal assignment on the 2x20 pin connector is shown in Figure 8. The P1-Pn naming convention is used for 2x20 pin connectors only.

**Figure 8. 2x20 Pin Connector**

| | P1 | | P3 | |
|---|---|---|---|---|
| Ref | Signal | Dev Pin# | Dev Pin# | Signal |
| 1 | 3.3V | | | 5V |
| 2 | ADC_CH1 | 58 | | GND |
| 3 | UART0_RX | 4 | 57 | ADC_CH0 |
| 4 | UART0_TX | 3 | 60 | ADC_CH3 |
| 5 | GPIO | 61 | 58* | ADC_CH1 |
| 6 | ADC_CH2 | 59 | 59* | ADC_CH2 |
| 7 | SPI_CLK | 5 | 63 | AUD_SYNC |
| 8 | GPIO | 62 | 53 | AUD_CLK |
| 9 | I2C_SCL | 1 | 64 | AUD_DOUT |
| 10 | I2C_SDA | 2 | 50 | AUD_DIN |




| P4 | | P2 | | |
|---|---|---|---|---|
| Signal | Dev Pin# | Dev Pin# | Signal | Ref |
| PWM | 2* | | GND | 1 |
| PWM | 1* | 18 | GPIO | 2 |
| PWM | 17* | 8 | SPI_CS | 3 |
| PWM | 64* | 45 | GPIO | 4 |
| CCAP/GPIO | 21* | | RESET_OUT | 5 |
| CCAP/GPIO | 18* | 7 | SPI_DOUT | 6 |
| GPIO | 62* | 6 | SPI_DIN | 7 |
| GPIO | 60* | 21 | GPIO | 8 |
| GPIO | 16 | 55 | GPIO | 9 |
| GPIO | 17 | 15 | GPIO | 10 |

The signal mappings are as indicated in Figure 8. All the signals are referred to by the pin number in the SDK; Figure 8 shows the default mappings. Some of the pins are repeated across the connector. For instance, pin 62 is available on P1 and P4, but only P1 is connected by default. The signal on P4 is marked with a *(star) to signify that it is not connected by default. It can be routed to the pin by using a 0-Ω resistor in the path. For the exact resistor placement, see the schematics and placement diagram.

## *2.5 Power*

The LaunchPad kit is designed to be powered by the USB connection or by external 2xAA or 2xAAA batteries.

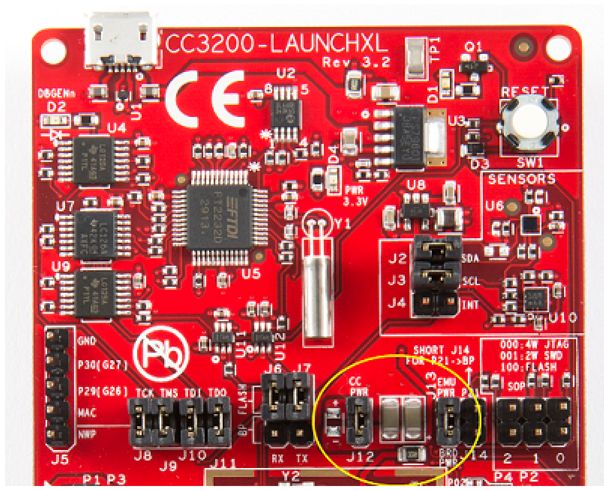
### 2.5.1 USB Power

The LaunchPad kit is designed to work from the USB-provided power supply. The LaunchPad kit enumerates as a bus-powered device on the computer. When the board is powered from the USB connector, the jumpers must be placed on the following headers, as shown in Figure 9.

J12 (shorted)

J13 (shorted)

**Figure 9. Powering From USB**

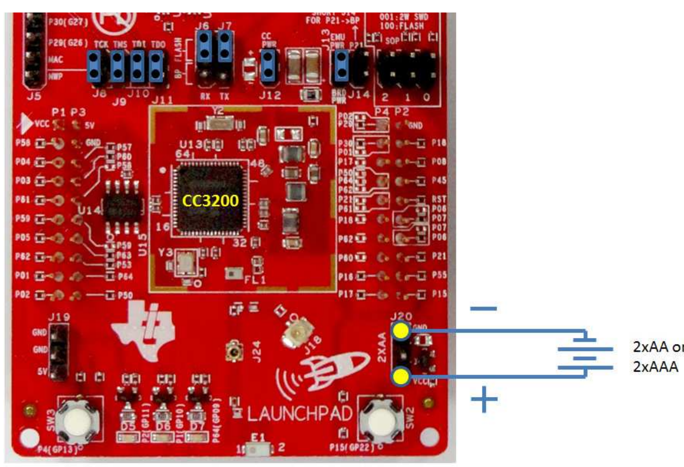


### 2.5.2 Battery Power (2 × 1.5 V)

The LaunchPad kit can also be powered from an external battery pack by feeding the voltage on the J20 header. This input features reverse voltage protection to ensure that the board is not damaged due to an accidental reverse voltage. The following care should be taken while using the board with a battery.

1. Remove the USB cable.
2. Plug-in the battery pack on J20 with correct polarity (see Figure 10).

**Figure 10. Battery Power**




### 2.5.3 BoosterPack Module Power Supply

The CC3200 LaunchPad kit can be powered by a stacked booster-pack, which can provide a 3.3-V power on P1.1. During this mode, remove the J13 so that the on-board LDO is not overloaded.

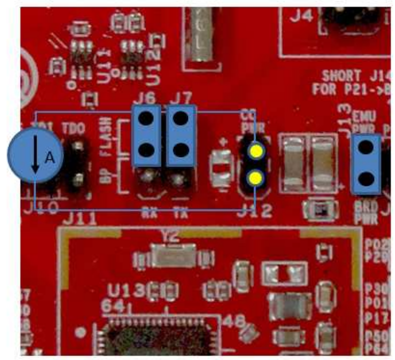
## 2.6 Measure CC3200 Current Draw

To measure the current draw of the CC3200, use the 3V3 jumper on the jumper isolation block (J12). The current measured in this mode includes only the CC3200 current and no external blocks. However, if a GPIO of the CC3200 is driving a high current load like LED, then that is also included in this measurement.

### 2.6.1 Measuring Low Power (<1 mA)

Follow these steps to measure ultra-low power:

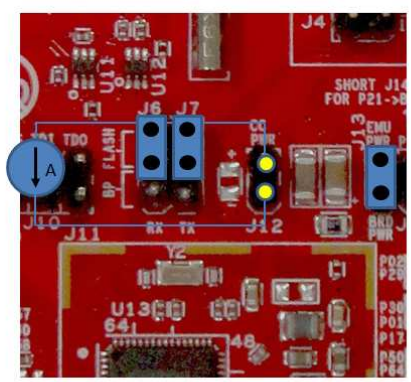
**Figure 11. Measuring Low Power**




1. Remove the 3V3 jumper (J12); attach an ammeter across this jumper.
2. The CC3200 should not drive any high-current loads directly, such as an LED, as this can draw a large current.
3. Begin target execution and set the device to low-power modes (LPDS or hibernate).
4. Measure the current. If the current levels are fluctuating, it may be difficult to get a stable measurement. It is easier to measure quiescent states.

### 2.6.2 Measuring Active Power

**Figure 12. Measuring Active Power**




1. Remove the 3V3 jumper (J12).
2. Solder a 0.1-Ω resistor on the board at R62. Or, attach a jumper wire between J12 so that it can be used with a current probe.
3. Measure the voltage across the R62 using an oscilloscope with a differential probe. (For the current probe, coil the wire around the sensor multiple times for good sensitivity).
4. An ammeter can also be used for this measurement, but the results may be erroneous due to the switching nature of the current.

## 2.7 RF Connections

### 2.7.1 Radiated Testing (AP connection)

By default, the board ships with the RF signals routed to the on-board chip antenna. An on-board u.fl (Murata) connector provides a means to perform the testing in the lab using a compatible cable.

**Figure 13. Radiated Testing Using Chip Antenna**

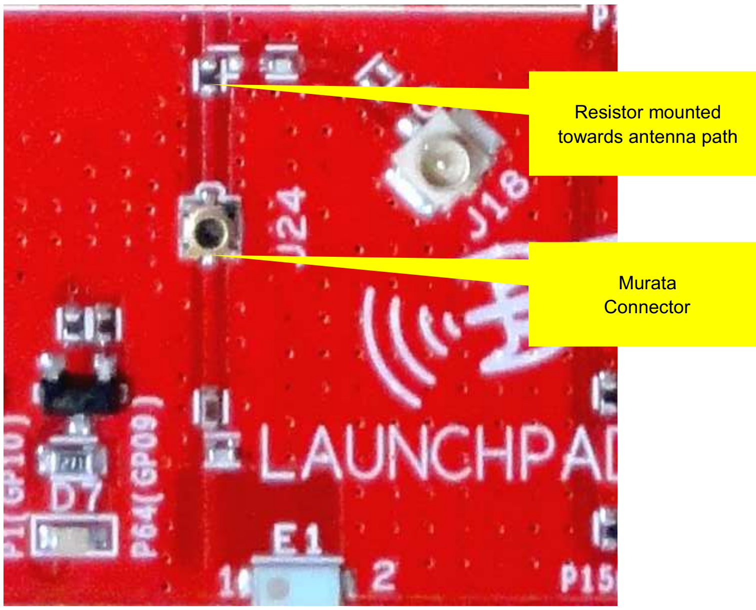
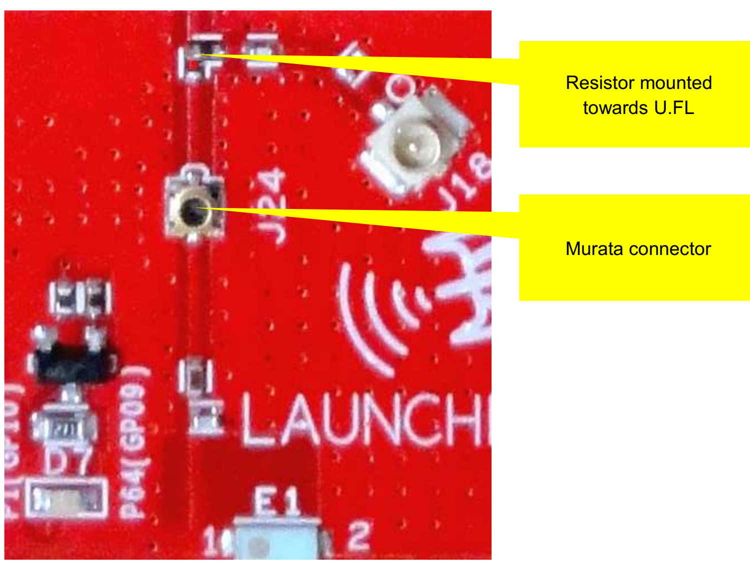



**Figure 14. Board Set for Conducted Testing**



## 2.8 Design Files

### 2.8.1 Hardware

All design files include schematics, layout, Bill of Materials (BOM), Gerber files, and documentation, which are made available for download from the following URL: http://www.ti.com/tool/cc3200-launchxl-rd.

### 2.8.2 Revision History

**Table 10. Change Log**

| PCB Revision | Description |
|---|---|
| Rev 3.0B | • First baseline revision |
| Rev 3.1 | The main changes pertain to the bill of materials (BOM) and the layout:<br>• Replaced the caps C23, C24 with ceramic ones to minimize leakage current<br>• R62 is made to DNP by default so that the jumper is used to measure the hibernate current<br>• Miscellaneous silk screen changes to clearly annotate components on the board. |
| Rev 3.2 | • Layout changes for the DC-DC section to improve the mask margin<br>• Updated the silk screen to reflect the final markings. |
| Rev 4.1 | • Added pullup and pulldowns for the serial flash. (Reduces hibernate current to around 17 uA)<br>• Moved the nRESET pull from VCC_BRD to VBAT_CC (ensures always pulled high).<br>• Added pullup on UART_TX going to the FTDI to prevent false start bits.<br>• Added pullup resistor for acccelerometer address to avoid conflict with audio booster pack<br>• Added 100K pullup on RESET_OUT net for any BP without RESET pulls.<br>• Changed R61 to 2.7K, R57-> 270 Ω (To solve false entering to bootloader mode)<br>• Miscellaneous silk changes |

### 2.8.3 Software

All design files including firmware patches, software example projects, and documentation are made available from the SimpleLink Wi-Fi Platform page.

The software development kit (SDK) for the CC3200 LaunchPad kit can be obtained from http://www.ti.com/tool/cc3200sdk.

# 3 Software Examples

## 3.1 Development Environment Requirements

The following software examples with the LaunchPad kit require an integrated development environment (IDE) that supports the CC3200 device.

For more details on where to download the latest IDE, see Section 4.3.

The CC3200 programmer's guide (SWRU369) has detailed information on software environment setup, with examples. Refer to this document for further details on the software sample examples.

### 3.1.1 CCS

CCS 6.0 or higher is required. When CCS has been launched, and a workspace directory chosen, use *Project → Import Existing CCS Eclipse Project*. Direct it to the desired demo's project directory containing main.c.

### 3.1.2 IAR

IAR 6.70 or higher is required. To open the demo in IAR, choose *File → Open → Workspace…*, and direct it to the *.eww workspace file inside the \IAR subdirectory of the desired demo. All workspace information is contained within this file.

The subdirectory also has an *.ewp project file; this file can be opened into an existing workspace, using *Project → Add-Existing-Project…*.

# 4 Additional Resources

## 4.1 LaunchPad Kit Wiki

Most updated information is available on the CC3200 Wiki page.

## 4.2 Information on the CC3200

For more information on CC3200, visit the product page (http://www.ti.com/product/cc3200), which includes the data sheet (SWAS032) and key documents such as the technical reference manual (SWRU367) and Wiki (http://www.ti.com/simplelinkwifi-wiki). These resources contain information on getting started, hardware details, software details including porting information, testing and certification, support, and the CC3200 community.

## 4.3 Download a Development Environment

Although the files can be viewed with any text editor, more can be done with the projects if they are opened with a development environment such as Code Composer Studio (CCS), IAR, or Energia.

CCS and IAR are each available in a full version, or a free, code-size-limited version. The full out-of-box demo cannot be built with the free version of CCS or IAR (IAR Kickstart), due to the code-size limit. To bypass this limitation, a code-size-limited CCS version is provided that has most functionality integrated into a library. The code built into the library is able to be viewed by the user, but it cannot be edited. For full functionality, download the full version of either CCS or IAR.

## 4.4 The CC3200 Code Examples

The user's guide for each example can be found within the Software Development Kit (SDK) or on the SimpleLink Wiki.

## 4.5 CC3200 Application Notes

There are many application notes with practical design examples and topics located at the SimpleLink(TM) Wi-Fi(R) Wiki page and the main landing page.

### 4.6 Support Resources

TI E2E™ support forums are an engineer's go-to source for fast, verified answers and design help — straight from the experts. Search existing answers or ask your own question to get the quick design help you need.

Linked content is provided "AS IS" by the respective contributors. They do not constitute TI specifications and do not necessarily reflect TI's views; see TI's Terms of Use.

## 5 Known Limitations

### 5.1 Hardware Limitations

#### 5.1.1 Floating IO (All Revisions)

All GPIO outputs from the CC3200 device float while the device enters hibernate state. This can cause glitches on the lines if they are not pulled externally.

#### 5.1.2 Board Modification for LPDS Mode

There must be a 100K pulldown resistor on pin19 (JTAG_TCK) for the device to reliably enter LPDS mode. This is not present on the boards.

#### 5.1.3 Floating S-Flash Lines (Rev 3.2 and Earlier)

The SPI lines routed from the CC3200 to the on-board serial flash are not pulled up or down using resistors on the board. When the device enters hibernate state, these pins can be floating, and high currents can be drawn by the serial flash.

# Revision History

| Changes from B Revision (January 2015) to C Revision | Page |
|---|---|
| • Added note (1) to Table 3 *Default $I^2C$ Addresses* | 9 |