CC3200 SimpleLink™ Wi-Fi® and Internet-of-Things Solution, a single-chip wireless MCU

Pre-production devices and
Software Development Kit (SDK) v0.5.2 Release Notes

TABLE OF CONTENTS

1 Introduction

This document describes the pre-production Software Development Kit (SDK) version 0.5.2 for use with the pre-production CC3200 SimpleLink Wi-Fi MCU device mounted on the CC3200 LaunchPad evaluation kit.

This pre-production release of the SDK and the CC3200 device has minor performance limitations which are listed as errata in section 9 of this document.

Texas Instruments will release the production version of the CC3200 device with updated development boards and an updated SDK during summer of 2014.

2 Getting Started

Please follow the on-line CC3200 Quick Start Guide to start using the CC3200 LaunchPad development platform.

Please download the CC3200 Getting Started Guide to get started with your project development.

3 Release Content

Item	Version	Type
Device	XCC3200HZ	Pre-production device
Development boards	CC3200-LAUNCHXL Rev3.2	Orderable from TI
SDK Installer	CC3200SDK-0.5.2-windows-installer.exe For Windows 7 and Windows XP	Provided with a click wrap license
Firmware	2.0.7.0.31.0.0.4.2.1.5.3.3	Binary, pre-flashed on the CC3200LAUNCHXL board
Network Processor host driver	Version 0.5.1	Source code
MCU Peripherals Drivers	Version 0.5.1	Source code
Power Management Framework Library	Version 0.5.1	Source code
Supported IDE	IAR version : 7.20 CCS version : 6.0	Delivered separately
Demo	Embedded HTML web-site	Pre-flashed on LaunchPad and source code provided
User guides	CC3200 Getting started guide CC3200LAUNCHXL User Guide SimpleLink Host Driver Programmer's Guide	PDF PDF Doxygen HTML
Tools	USB driver for CC3200LAUNCHXL for Windows7 and Windows XP	Executable

4 Directory structure of SDK

Double-Click on the package to copy the directories (and files) to the preferred location. The first level directory structure is as shown in the table below.

Directory Name	Content
Docs	• SimpleLink Host Driver Programmer's Guide • Peripheral Driver Library User's Guide • Power Management Framework Guide • Application notes for sample applications
Examples	Example application in source code
Driverlib	• Peripheral driver library source files • The driverlib.a is also provided in CCS and IAR directories
Inc	• Register definition header files
middleware	• Power management framework providing an easy to use infrastructure for power aware solution
Oslib	• Interface file to configure Free-RTOS or TI-RTOS
SimpleLink	• The SimpleLink Network Processor host driver code. • simplelink.a, simplelink_nonos.a and simplelink_pm are available in CCS and IAR directories. • For GCC compiler simplelink.a and simplelink_nonos.a are available in gcc folder.
third_party	• FatFS source files • FreeRTOS souce files
ti_rtos	• Abstraction layer files for TI-RTOS
tools	• **ccs_patch –** Files required for CCS-FTDI-LP connection • **Iar_patch –** Files required for IAR-FTDI-LP connection • **ftdi-** Contains CC3200 FTDI-USB driver for Windows

5 Networking features

5.1 Wi-Fi

Standards	802.11b/g/n **Station** and Wi-Fi Direct **Client**
Supported Channels	1-13
Personal Security	WEP, WPA and WPA2
Enterprise Security	WPA-2 Enterprise EAP Fast, EAP PEAPv0 MSCHAPv2, EAP PEAPv0 TLS, EAP PEAPv1 TLS, EAP TLS, EAP TTLS TLS, EAP TTLS MSCHAPv2
Provisioning	SmartConfig™ technology Wi-Fi Protected Setup (WPS2) Access Point mode with internal HTTP Web Server
Standards	802.11b/g **Access Point** and Wi-Fi Direct **Group Owner**
Clients	1
Personal Security	WEP, WPA and WPA2

5.2 Networking protocols

IP	IPv4	
Transport	UDP TCP RAW ICMP	
Cross-Layer	DHCP ARP DNS	
Application	mDNS DNS-SD HTTP 1.0 web server	
Transport Layer Security	SSLV3 SSLV3 TLSV1 TLSV1 TLSV1 TLSV1 TLSV1 TLSV1 TLSV1_1 TLSV1_1 TLSV1_1 TLSV1_1	SSL_RSA_WITH_RC4_128_SHA SSL_RSA_WITH_RC4_128_MD5 TLS_RSA_WITH_RC4_128_SHA TLS_RSA_WITH_RC4_128_MD5 TLS_RSA_WITH_AES_256_CBC_SHA TLS_DHE_RSA_WITH_AES_256_CBC_SHA TLS_ECDHE_RSA_WITH_RC4_128_SHA TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA TLS_RSA_WITH_RC4_128_SHA TLS_RSA_WITH_RC4_128_MD5 TLS_RSA_WITH_AES_256_CBC_SHA TLS_DHE_RSA_WITH_AES_256_CBC_SHA

	TLSV1_1	TLS_ECDHE_RSA_WITH_RC4_128_SHA
	TLSV1_1	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
	TLSV1_2	TLS_RSA_WITH_RC4_128_SHA
	TLSV1_2	TLS_RSA_WITH_RC4_128_MD5
	TLSV1_2	TLS_RSA_WITH_AES_256_CBC_SHA
	TLSV1_2	TLS_DHE_RSA_WITH_AES_256_CBC_SHA
	TLSV1_2	TLS_ECDHE_RSA_WITH_RC4_128_SHA
	TLSV1_2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
User application sockets	Up to 8 open sockets Up to 2 secured application sockets: - One server (listen socket and accept socket) + client (data socket) - Up to two clients (data socket)	

5.3 Advanced Features

802.11 Transceiver	Transmit and Receive raw Wi-Fi packets with full control over payload. Wi-Fi disconnect mode. Can be used for general-purpose applications (e.g. tags, sniffer, RF tests)
Traffic Filters	Embedded filters to reduce power consumption and Wake-on-LAN trigger packets (IP and MAC layer)

5.4 Power modes

MCU Power Management framework library allows user applications to exploit the following device power management modes:

- Sleep
- Low Power Deep Sleep
- Hibernate

The Wi-Fi network processor core supports the following low power policies

Low Power mode	Uses 802.11 Power Save and Device Deep Sleep Power with three user configurable policies
Configurable Power Policies	• Normal (Default) - Best tradeoff between traffic delivery time and power performance • Low power –Used only for Transceiver mode application (Disconnect mode) • Long Sleep Interval – wakes up for the next DTIM after a configurable sleep interval, up to 2 seconds. This policy is only applicable for client socket mode

5.4.1 MCU Peripheral Interface Drivers

Drivers and example applications are provided in the SDK package for the following peripheral interfaces -

ADC	Analog to digital converter.
AES	Advanced encryption standard
CAMERA	Parallel interface (8 bits) that can be used to connect to camera sensors.
CRC	Cyclic redundancy check
DES	Data encryption standard.
GPIO	General purpose input/output
I2C	Standard I2C interface to communicate with various sensors
Interrupt	Interrupt module
I2S	Audio signal processor
PinMux	Pinmux setting for Pads
PRCM	Power reset and clock module
SDHost	Secure digital host controller
SHAMD5	Secure hash algorithm, message digest algorithm
SPI	Serial peripheral interface
Systick	System tick functionality
Timer	General purpose timers
UART	Standard UART interface
UDMA	Direct memory access
WDT	Watchdog timer

6 Advanced information

- TCP/IP
  - TCP Window size is 16KB, divided between application sockets.
  - IP Fragmentation is not supported for Tx UDP and RAW sockets
  - Max Tx payload for Raw packet with IP header is 1460 bytes
  - Max Tx payload for Raw Transceiver is 1488 bytes
- SSL/TLS Certificates
  - Certificate Authority (CA) certificates needs to be installed if server authentication is required
  - CA Certificate key size up to 2048 bit
- WEP
  - Supporting only WEP open using ASCII pre shared key
- WPS
  - Up to 4 seconds delay between association and EAPOL-Start
- SmartConfig
  - Not supported with 5GHz AP (802.11a/n/ac)
  - Not supported for MIMO-capable configuration devices
  - Only Group 0 is supported in auto start mode
- Tx Power
  - Tx power in AP mode takes effect only after reset
- Wi-Fi Direct
  - In Group Owner mode FAST connection policy has to be set to TRUE
- Rx Filters
  - BSSID can't be filtered while STA is connected (if filtered will cause disconnection)
- Power Management
  - The device will remain in active after initialization until the host reads all events
- File System
  - Up to 100 user files, file size is not limited
- MCU Deep Sleep Mode
  - User Application should handle the behavior that MCU Deep Sleep mode invocation leads to change in Peripheral Clocks (Frequency gets reduced by a factor of ‘2’)
- Serial Flash
  CC3200 supports JEDEC specification compliant Serial Flash devices with 4Kbyte sector size erase. The following parts were validated:

Micron	N25Q128-A13BSE40	128Mbit
Spansion	S25FL208K	8Mbit
Winbond	W25Q16V	16Mbit
Adesto	AT25DF081A	8Mbit
Macronix	MX25L12835F-M2	128Mbit

7 Networking sample applications

The release package includes several sample applications developed for the CC3200 Cortex M4 processor. The applications come with

- Detailed Application Note (readme documents)
- Project files for IAR, CCS or Both as well as GCC makefiles for a few applications

Recommend to use “Getting Started in STA mode” application as sample reference code to write new application.

7.1 Antenna Selection

This is a reference implementation for antenna-selection scheme running on the CC3200 MCU, to enable improved radio performance inside buildings

7.2 Connection Policies

This application demonstrates the usage of the CC3200 profiles and connection-policies.

7.3 Send Email

This application sends an email using SMTP to a user-configurable email address at the push of a button.

7.4 Enterprise Network Connection

This application demonstrates the procedure for connecting the CC3200 to an enterprise network.

7.5 File System

This application demonstrates the use of the file system API to read and write files from the serial Flash.

7.6 Get Time

This application connects to an SNTP cloud server and receives the accurate time.

7.7 Get Weather

This application connects to ‘Open Weather Map’ cloud service and receives weather data.

7.8 Getting Started in AP Mode

This application configures the CC3200 in AP mode. It verifies the connection by pinging the connected client.

7.9 Getting Started in STA Mode

This application configures the CC3200 in STA mode. It verifies the connection by pinging the connected Access Point.

7.10 HTTP Server

This application demonstrates using the on-chip HTTP Server APIs to enable static and dynamic web page content.

7.11 MDNS

This application registers the service for broadcasting and attempts to get the service by the name broadcasted by another device.

7.12 Mode Configuration

This application demonstrates switching between STA and AP modes.

7.13 NWP Filters

This application demonstrates the configuration of Rx-filtering to reduce the amount of traffic transferred to the host, and to achieve lower power consumption.

7.14 P2P (Wi-Fi Direct)

This application configures the device in P2P (Wi-Fi Direct) mode and demonstrates how to communicate with a remote peer device.

7.15 Provisioning AP

This application demonstrates the use of the on Chip HTTP server for Wi-Fi provisioning in AP Mode, building upon example application 7.8 above.

7.16 Provisioning with SmartConfig

This application demonstrates the usage of TI's SmartConfig™ Wi-Fi provisioning technology. The *Wi-Fi Starter Application* for iOS and Android is required to use this application. It can be downloaded from following link: http://www.ti.com/tool/wifistarter or from the Apple App store and Google Play.

7.17 Provisioning with WPS

This application demonstrates the usage of WPS Wi-Fi provisioning with CC3200.

7.18 Scan Policy

The application demonstrates the scan-policy settings in CC3200.

7.19 SSL/TLS

The application demonstrates the usage of certificates with SSL/TLS for application traffic privacy and device or user authentication

7.20 TCP Socket

The application demonstrates simple connection with TCP traffic.

7.21 Transceiver Mode

The application demonstrates the CC3200 transceiver mode of operation.

7.22 UDP Socket

The application demonstrates simple connection with UDP traffic.

7.23 XMPP Client

The application demonstrates instant messaging using a cloud based XMPP server.

7.24 File Download

This application demonstrates file download from a cloud server to the on board serial Flash.

7.25 Deep-sleep

This application demonstrates the deep-sleep low power mode a networking application.

7.26 Hibernate

This application demonstrates the hibernate ultra-low-power mode in a networking application using an UDP client.

7.27 Serial Wi-Fi

This application implements a wireless terminal over a Wi-Fi network and an "AT commands" like control interface

7.28 Out of Box Application

This application demonstrates the out of box experience where user can view different demo and SDK web links on their web-browser.

7.29 Wi-Fi Audio

This application demonstrates Bi-directional wireless audio on a CC3200 LaunchPad. It uses two LaunchPad boards in STA mode and streams the audio from one LaunchPad to the other.

7.30 Camera Application

This example demonstrates image capture using the CC3200 fast parallel interface

7.31 Sensor Profile

This application shows how to use hibernate mode through the Power Management Framework

7.32 Idle Profile

This Application exercises low power modes (LPDS) using Power Management Framework (middleware).

8 MCU sample applications

8.1 LED Blink

This application uses the GPIO DriverLib APIs to blink an LED on the CC3200 Launchpad.

8.2 Timer Demo

This application demonstrates the CC3200 timer DriverLib APIs. It uses 16 bit timers to generate interrupts which in turn toggle the state a GPIO driving LEDs.

8.3 Watchdog Demo

This application demonstrates the use of the Watch Dog timer (WDT) DriverLib APIs. It shows how the watchdog timer resets the device on system failure.

8.4 UART Demo

This application uses the UART DriverLib APIs to demonstrate typing echo on a UART terminal.

8.5 Interrupt Application

This application uses the Interrupt DriverLib APIs to demonstrate the CC3200 MCU interrupt preemption and tail-chaining capabilities.

8.6 I2C Demo

This application uses the I2C DriverLib APIs to read and write the I2C peripherals on the CC3200 LaunchPad.

8.7 MCU Sleep-DS

This application demonstrates the Sleep and Deep Sleep modes of the CC3200 MCU.

8.8 uDMA Application

This application uses the UDMA DriverLib APIs to show various DMA mode functionalities.

8.9 FreeRTOS Demo

This application demonstrates using FreeRTOS for multiple task creation and inter-task communication using queues.

8.10 AES Demo

This application uses the AES Driverlib APIs to exercise various AES encryption modes.

8.11 DES Demo

This application uses the DES Driverlib APIs to exercise various AES encryption modes.

8.12 CRC Demo

This application uses the CRC Driverlib APIs to exercise various CRC calculation modes.

8.13 SHA-MD5 Application

This application uses the SHA-MD5 Driverlib APIs to exercise various SHA-MD5 modes.

8.14 ADC Demo Application

This application demonstrates the C3200 ADC module using the Driverlib APIs.

8.15 PWM Demo

This application demonstrates the PWM mode of the CC3200 General Purpose Timers (GPT).

8.16 SD Host Demo

This application demonstrates the functionality of the Secure Digital Host (SD Host) controller of CC3200, which interfaces with standard SD memory cards in 1-bit transfer mode.

8.17 SDHost FatFS Demo

This application uses the FatFS library for block level read/write access to SD card, using the SD Host controller on CC3200.

8.18 SPI Demo

This application shows the initialization sequence that enables the CC3200 SPI module in full duplex 4-wire master and slave modes.

8.19 UART DMA

This application demonstrates using the CC3200 UART interface with uDMA and interrupts.

8.20 Timer Count Capture

This application demonstrates measuring the frequency of an external signal using the CC3200 Timer count capture feature.

9 Revision History

SDK Version	Updates from previous version
0.5.2	• Added a function to configure the firmware to default state across all applications. • Added error handling to Host driver API calls in application "Getting Started in STA mode". This can be used as sample reference code for writing new application.
0.5.1	First Release

10 Errata

The following section covers known issues and performance limitations with SDK 0.5.2 and the CC3200 pre-production devices. TI will release the CC3200 production devices with SDK 1.0.0 during summer 2014 that will remove some of the limitations as described ahead.

10.1 Functionality

10.2 Pre-regulated 3.3v to Pin 47

For preproduction devices connect an external pre-regulated 3.3v +/- 5% supply to pin 47 (VDD_ANA2). This adds 12mA average current and up to 100mA peak current over 20uSec to the total system current at 3.3V.

The CC300 LaunchPad version 3.2 already includes the correct supply configuration for the pre-production device and also adds a 10uF capacitor to filter the peak currents. No further action is required.

The external 3.3V supply is not required in the CC3200 production device in which case pin 47 can be left not connected. If pin 45 is used in the production device as a GPIO, pin 47 has to be connected to any on-board power supply.

10.3 Power consumption increase

Power consumption of the CC3200 pre-production device in all active modes is 1-2 mA higher compared to the CC3200 production devices

10.4 RAM size

Preproduction device have 192 Kbyte RAM available for user application. The production devices will have 256 Kbyte RAM available

10.5 MCU Low-Power Deep Sleep (LPDS)

Pre-production devices do not support LPDS mode.

10.6 Peripherals driver library

On CC3200 pre-production devices the peripheral driver library needs to be compiled and executed from RAM. On CC3200 production device, the peripheral driver library is burned in the device ROM leaving more RAM space for user applications.

10.7 Network Processor Performance

Item	SDK 0.5.2	SDK 1.0.0**
MCU-NWP SPI link	13 MHz	20 MHz
Init time from hibernate until device ready for networking	250 mSec	50 mSec
Init time from hibernate until WPA2 connection*	300 mSec	95 mSec
Maximum UDP throughput, open socket	13 Mbps	16 Mbps
Maximum TCP throughput, open socket	11 Mbps	12 Mbps
Maximum TLS/SSL throughput with RC4_128 cipher	5 Mbps	8 Mbps
Maximum TLS/SSL throughput with AES_256 cipher	7 Mbps	8 Mbps
Minimum TLS/SLL connection time with ECC cipher	2.5 Sec	1.4 Sec
Minimum TLS/SSL connection time with RSA cipher	200 mSec	150 mSec

* Excluding user application boot time

** SDK 1.0.0 target performance based on CC3200 production device

10.8 Wi-Fi known issues

ID	MCS00130040
Title	Wi-Fi Direct connection can fail at first attempt
Description	Negotiation with peer device is not always successful at first attempt
Impact	The first connection doesn't success
Workaround	Repeat the connection attempt
Fix Expected	Will be improved in SDK 1.0.0

ID	MCS00130160
Title	Scan during connection process
Description	Cannot invoke a scan command while trying to connect
Impact	Scan command might interfere with connection process
Workaround	Avoid calling scan during connection
Fix Expected	Fix expected in SDK 1.0.0

ID	MCS00130368
Title	Adding profile using Fast connection-policy
Description	The profile has to be explicitly added when using 'Fast' connection-policy
Impact	Connection policy with 'Fast' will not connect automatically if profile is not added explicitly.
Workaround	Add the profile manually
Fix Expected	Fix expected in SDK 1.0.0

10.9 Networking known issues

ID	MCS00127876
Title	sl_NetAppDnsGetHostByName returns with no answer
Description	In high Rx traffic conditions some DNS packets can be dropped, causing GetHostByName to fail
Impact	No answer on request
Workaround	Upon error return status call the API again

ID	MCS00128959
Title	Failing IXIA invalid DHCP test case
Description	Failing IXIA test case for RFC2131
Impact	Device IP address will not be updated when receiving a malformed DHCPACK packet

ID	MCS00129407
Title	Failing IXIA malformed ICMP header test case
Description	Failing IXIA test case 5.1 for handling malformed ICMP header
Impact	None

10.10 Host applications known issues

ID	MCS00130291
Title	WPS PIN Connect failure if pin code is not null-terminated
Description	If the PIN code from the HOST is not null terminated connection can fail in some cases
Impact	Connection failure
Workaround	Add null termination to the PIN code string

ID	MCU00004896
Title	During Debug, the path for driverlib or simplelink related files needs to be specified explicitly by the user
Description	This is due to IAR/CCS IDE generate *.a library with static path embed in it.
Impact	Clean Build Required for debug sessions
Workaround	Build Driverlib and Simplelink project before building any example project.
Fix Expected	TBD

10.11 Applications known issues

ID	MCS00130240
Title	In AP mode the internal DNS Server cannot be disabled
Impact	Cannot use external DNS server in AP mode

ID	MCS00130241
Title	'AnyP2P' and 'Auto smart config' policies can be changed only in station or P2P mode
Impact	Can't change these specific configurations from the HTTP server in AP mode
Workaround	Change the configurations while in STA mode

ID	MCS00130114
Title	HTTP Server: Cannot add Enterprise or P2P profile from HTTP Server
Description	Adding enterprise or P2P profile is not possible from HTTP pages
Impact	Can't add enterprise or P2P profiles from HTTP pages
Workaround	Add the specific profiles from MCU
Fix Expected	Fix expected in SDK 1.0.0