

Table of Contents

- [Introduction](#)
- [Getting Started](#)
- [What's New](#)
- [Updating the servicepack](#)
- [Dependencies](#)
- [Device Support](#)
- [Directory Structure](#)
- [Fixed Issues](#)
- [Known Issues](#)
- [Notes for CC3200 Application Developers](#)
- [Technical Support](#)

Introduction

The SimpleLink™ Wi-Fi® CC3200 device is a single-chip microcontroller (MCU) with built-in Wi-Fi connectivity. The 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.

For more information, visit www.ti.com/simplelink.

This is v1.5.0 of the CC3200 SDK.

Getting Started

To get started using the CC3200 LaunchPad with the Out of Box experience, please see the [CC3200 Quick Start Guide](#).

To get started with your project development, please see the [CC3200 Getting Started Guide](#).

What's New

- New APIs added to the host driver.
 - API to mask supported rates in the RX and TX
 - API to extend scan duration on selected channels
 - API to configure the device to use NULL frames instead of PS-POLL frames when fetching packets from the AP in power save mode
- New event added
 - An unsolicited event to indicate the same 3rd EAPOL packet is received again during 4-way handshake

Updating the servicepack

It is recommended to update the service pack to the latest available, which can be downloaded from the [CC3200 SDK download page](#).

For more information, see the [CC3100 & CC3200 UniFlash Guide](#).

Dependencies

This release was validated with the following components:

- [Code Composer Studio 8.3](#)
 - Compiler TI 18.1.4.LTS
- IAR Code Generation Tools
 - [ARM 8.20.2](#)
- SimpleLink host driver 1.0.1.14
- [CC3100/CC3200_ServicePack 1.0.1.14-2.12.2.6](#)

- [TI-RTOS](#) 2.16.01.14
- [FreeRTOS](#) 10.0.1

Device Support

- CC3200R1 [ES1.33 Chip Id : 0x4000010]
 - CC3200-LAUNCHXL Rev3.2 onwards

Directory Structure

Directory Name	Content
Docs	Quick Start Guide for LaunchPad Out of Box Experience Getting Started Guide for application development SimpleLink Host Driver Programmer's API Guide (simplelink_api folder) CC3200 TI-RTOS User's Guide Power Management Framework Guide OTA Update Application Note SimpleLink OTA Extlib API Guide CC3200 Peripheral Driver Library API Guide API Guides for MQTT and HTTP libraries (netapps folder)
Driverlib	Peripheral driver library source files. driverlib.a is provided in CCS, GCC and IAR directories.
Examples	Example applications in source code
Inc	Register definition header files
Middleware	Power Management Framework providing an easy-to-use infrastructure for low power solutions
Netapps	HTTP client library source files HTTP server library source files (webserver) JSON parser library source files MQTT client library source files MQTT client-server library source files SMTP client library source files TFTP client library source files XMPP client library source files
Oslib	Interface file to configure Free-RTOS or TI-RTOS
SimpleLink	The SimpleLink Network Processor host driver code. Pre-built simplelink library (simplelink.a) for OS, NON_OS, PM_framework and NON_OS_PM are available in CCS and IAR directories, plus debug versions of these libraries. For GCC compiler, libsimplelink.a and libsimplelink_nonos.a are available in gcc folder.
SimpleLink Ext Libraries	FLC library source files OTA library source files Provisioning library source files
Third Party	FATFS library source files FreeRTOS library source files
TI-RTOS	Abstraction layer files for TI-RTOS
Tools	ccs_patch – Files required for CCS-FTDI-LP connection gcc_scripts – Script files required for GCC compiler ftdi - Contains CC3200 FTDI-USB driver for Windows iar_patch – Files required for IAR-FTDI-LP connection

Examples

Networking Applications

--	--	--	--

Example	Description	OS Support	Documentation
Antenna Selection	Antenna Selection is the method by which users can configure the WLAN RF antenna on the CC3200 device to improve radio performance.	TI-RTOS	antenna_selection README
Connection Policy	Demonstrates the connection policies which determine how the CC3200 connects to an AP.	Non-OS	connection_policy README
Email	Demonstrates how to send emails via SMTP on a hardware interrupt.	TI-RTOS	email README
Enterprise Security	Demonstrates the connection to an enterprise network using the flashed certificate. Certificate is flashed onto the serial flash.	Non-OS	ent_wlan README
File Download	Demonstrates how to connect to a server to download a file and save it on the serial flash.	Non-OS	file_download README
File Operations	Demonstrates the file operations on the serial flash.	Non-OS	file_operations README
Get Time	Connects to a SNTP server and requests time information.	FreeRTOS	get_time README
Get Weather	Connects to a weather service website and requests weather data.	FreeRTOS	get_weather README
Getting Started with WLAN AP	Demonstrates device capability to behave as an AP in a typical networking system.	TI-RTOS	getting_started_with_wlan_ap README
Getting Started with WLAN Station	Demonstrates device capability to behave as a station in a typical networking system.	TI-RTOS	getting_started_with_wlan_station README
Hibernate	Showcases hibernate mode as a power-saving tool in a networking context, demonstrated as a UDP client.	FreeRTOS	hib README
HTTP Client Demo	Demonstrates the HTTP Client library to enable the device as an HTTP Client.	Non-OS	http_client_demo README
HTTP Server	Demonstrates the internal HTTP Server hosted on the CC3200.	FreeRTOS	httpserver README
Idle Profile (Non OS)	Demonstrates a low power mode (LPDS) using the Power Management Framework in a Non-OS environment.	Non-OS	idle_profile_nonos README
Idle Profile	Demonstrates a low power mode (LPDS) using the Power Management Framework.	FreeRTOS	idle_profile README
mDNS	Demonstrates mDNS functionality in CC3200. This application showcases “mDNS advertise” and “mDNS listen” functionality.	Non-OS	mdns README
Mode Configuration	Configure the device either to a station or an Access Point.	Non-OS	mode_config README
MQTT Client	Showcases the device acting as a MQTT client in a fully functional MQTT network.	TI-RTOS/FreeRTOS	mqtt_client README
MQTT Server	Showcases the device acting as an MQTT Server managing multiple local clients and allowing local clients to communicate with remote MQTT clients.	FreeRTOS	mqtt_server README
NWP Filter	Demonstrates the RX filtering feature on the CC3200	Non-OS	nwp_filter README

Example	Description	OS Support	Documentation
OTA Update	Shows the Over-The-Air (OTA) firmware update including MCU image, service pack, and additional user files.	FreeRTOS/Non-OS	OTA Application Note
Out of Box	Demonstrates the Out of Box Experience with the CC3200 Launch Pad.	FreeRTOS	out_of_box README
P2P (Wi-Fi Direct)	Demonstrates Wi-Fi direct (Peer to Peer) feature on the CC3200.	Non-OS	p2p README
Power Measurement	Provides users the ability to configure the device in various low power modes and profiles for power measurements.	Non-OS	power_measurement README
Provisioning AP	Demonstrates the AP provisioning feature. AP Provisioning is the method by which users can configure AP information on the CC3200 device from a web browser.	Non-OS	provisioning_ap README
Provisioning SmartConfig	Demonstrates the SmartConfig provisioning feature. TI SmartConfig is the proprietary method where users can use a Wi-Fi enabled device to configure the CC3200 to an AP on the fly.	Non-OS	provisioning_smartconfig README
Provisioning WPS	Demonstrates how to use WPS for Wi-Fi provisioning.	Non-OS	provisioning_wps README
Scan Policy	Demonstrates how to set the scan policy and enables the scan.	Non-OS	scan_policy README
Sensor Profile	Demonstrates a low power mode (hibernate) using the Power Management Framework.	FreeRTOS	sensor_profile README
Serial Wi-Fi	Serial Wi-Fi is a capability designed to provide easy, self-contained terminal access behavior over a UART interface.	FreeRTOS	serial_wifi README
SSL	Showcases SSL implementation on the CC3200.	Non-OS	ssl README
TCP Socket	Demonstrates device communication over a network using TCP protocol.	Non-OS	tcp_socket README
TFTP Client	Demonstrates how a client application can read/write a file from/to the TFTP server running on a network.	FreeRTOS	tftp_client README
Transceiver Mode	The RX Statistics feature is used to inspect the medium in terms of congestion, distance, and validate the RF hardware. Using the RSSI information it can help position the CC3200 in an ideal location.	Non-OS	transceiver_mode README
UDP Socket	Demonstrates device communication over a network using UDP protocol.	Non-OS	udp_socket README
Watchdog System	Demonstrates the use of a watchdog timer (WDT) in a complete system with MCU and networking subsystem.	Non-OS	watchdog_system_demo README
WebSock Simple	Demonstrates the real-time streaming capability of websocket protocol.	FreeRTOS	websocket_simple README
Wi-Fi Audio	Demonstrates audio streaming on the CC3200. Device can act as a speaker or microphone.	FreeRTOS	wifi_audio_app README
XMPP	Demonstrates the connection and communication via a XMPP server.	TI-RTOS	xmpp README

MCU Only Applications

More demo applications

Example	Description	OS Support	Documentation
ADC	Demonstrates the functionality of the CC3200 ADC module using driverlib APIs.	Non-OS	adc README
AES	Demonstrates the AES encryption feature on the CC3200.	Non-OS	aes README
Application Bootloader	Demonstrates the secondary bootloader operations to manage updates to application image.	Non-OS	application_bootloader README
Blinky	Demonstrates the GPIOs using driverlib APIs to control LEDs on the CC3200. LaunchPad.	Non-OS	blinky README
CRC	Demonstrates the functionality of the CC3200 CRC module using driverlib APIs.	Non-OS	crc README
DES	Demonstrates the DES encryption feature on the CC3200.	Non-OS	des README
Dynamic Library Loader	Demonstrates dynamic loading of an application binary from non-volatile memory while the program is being executed.	FreeRTOS	dynamic_lib_loader README
FreeRTOS Demo	Showcases FreeRTOS functionality like task creation and inter-task communication using message queues.	FreeRTOS	freertos_demo README
I2C	Demonstrates the functionality of the CC3200 ADC module using driverlib APIs.	Non-OS	i2c_demo README
Interrupt	Demonstrates interrupt preemption and tail-chaining capabilities.	Non-OS	interrupt README
MCU Sleep	Demonstrates the sleep mode of the MCU.	Non-OS	sleep README
Pulse Width Modulation (PWM)	Demonstrates the functionality of the CC3200 PWM module using driverlib APIs.	Non-OS	pwm README
SDHost	Demonstrates the functionality of the CC3200 SDHost module using driverlib APIs.	Non-OS	sdhost README
SDHost FatFS	Demonstrates the functionality of the CC3200 SDHost FATFS module using driverlib APIs to provide the block level read/write access to the SD card.	Non-OS	sdhost_fatfs README
SHA-MD5	Demonstrates the SHA-MD5 Hash Algorithm on the CC3200.	Non-OS	shamd5 README
SPI	Demonstrates the functionality of the CC3200 SPI module using driverlib APIs.	Non-OS	spi_demo README
Timer	Demonstrates the usage of 16 bit timers to generate interrupts that toggle GPIOs.	Non-OS	timer README
Timer Count Capture	Demonstrates the timer count capture feature to measure the frequency of an external signal.	Non-OS	timer_cc README
UART	Demonstrates the functionality of the CC3200 UART module using driverlib APIs.	Non-OS	uart_demo README
UART DMA	Demonstrates UART using driverlib APIs with uDMA and interrupts.	Non-OS	uart_dma README
uDMA	Demonstrates different DMA modes of transfer.	Non-OS	udma README
Watchdog Demo Application	Demonstrates the watchdog timer functionality to reset the system whenever the system fails.	Non-OS	watchdog README

Fixed Issues

- WPA2 enterprise re-authentication issue. The client supplicant may get disconnected upon RADIUS server re-authentication.
- APs are not included in scan list in a congested environment since the device doesn't stay on the channel too long. A new extended scan API is added as noted above.
- A very high L2 retry rate observed in congested environments, especially when high rates are used. A new rates masking API is added as noted above.
- Sudden drops in TX throughput during Bidirectional TCP data on the same socket.
- Bidirectional maximum TCP throughput may cause NWP abort when using TLS1.0 and TLS_RSA_WITH_AES_256_CBC_SHA or TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA.
- Data patterns may get injected into packets pulled from NWP causing integrity failure.
- Setting scan interval does not behave properly when using big-endian host processor.
- TCP fast retransmission is not triggered upon 3 duplicate ACK frames as expected.
- Rare disconnections from an AP with personal WPA2. A new unsolicited event is added as noted above.
- When working in non-blocking secured sl_Connect(), after some time the maximum 2 SSL sockets is reached although sockets are closed properly.
- Device may stay on SD although LSI is set.
- Once in a while, when AID0 and VBM are signaled simultaneously, the device may not send a PS-POLL or NULL frame.
- In case of AID0 and VBM are signaled simultaneously, the PS-POLL or NULL frame is transmitted immediately and not at the end of the broadcast/multicast chain.
- Specific APs like Engenius may occasionally disconnect the device. Fixed by using the new No Ps Poll feature as noted above.
- When RF conditions are poor and 1Mbps/2Mbps are used, the device may consume more power and loose synchronization with the AP.
- Device may stay stuck in awake (consuming constant high current) after disconnection from the AP.
- Device may stay stuck in awake (consuming constant high current) in congested environments where AP starts suppressing/deferring beacons.
- Device may consume high power when AP's clock drifts forward.
- Failure to exchange data after full 4-way handshake rekey (either as a standalone or enterprise).
- High power consumption in connected idle if during first time connection, the AP beacons are missing.
- Device does not disconnect if beacons are missing and AP responds to Probes (150 consecutive missed beacons).

Host Driver

ID	Summary
SLWIFICSR-229	Request x509 serial number to be exposed via getsockopts
CC3X00SDK-203	Add host driver macro to disable WPA enterprise server authentication
SLWIFICSR-434	Add host driver macro to enable extended scan
SLWIFICSR-435	Add host driver macro to enable extended scan
SLWIFICSR-436	Add host driver wlan option to mask supported rates
SL_WIFI_R1-1665	Add host driver wlan option to enable/disable PSpOLL mode

Known Issues

SDK

ID	Summary
SLWIFICSR-425	HTTPCli_getResponseStatus does not timeout in file_download example
SLWIFICSR-420	Memory leak in webserver library when receiving data (WSCore_DataRecv)

Network Processor / Host Driver

ID	Title	Description	Impact	Workaround
SL_WIFI_R1-400	WiFi Security: CC3100 and CC3200 Supports only WEP with Key Index 0 (==> AP Key	When using WEP security – only WEP index 0 is supported	Can't use more than one key in WEP	None

ID	index 1) Title	Description	WEPP Impact security	Workaround
SL_WIFI_R1-149	WiFi Security: Traffic Stop while WPA EAP-TLS Enterprise and Reauthentication enabled	In WPA EAP-TLS security the traffic stopped when Reauthentication packet is received	Traffic stopped	Disable reauthentication or set it to a very long time
SL_WIFI_R1-1258	Scan: Results list contain duplicate networks	Device might returns duplicate networks when the network list is not totally filled and the get scan results ask for fewer entries than what was actually found.	Duplicate networks in Scan results list	Read the maximum entries at once (20 entries) or to read one by one starting from the end to the beginning and check for duplicates. Once a duplicate was found the list is completed
SL_WIFI_R1-747	sl_NetAppDnsGetHostByName return with no answer in high traffic	In high Rx traffic some DNS packets can get lost	No answer on request	Run API again
SL_WIFI_R1-808	UDP/RAW socket data payload is limited to MTU size	Tx IP Fragmentation is not supported for UDP and RAW Tx	Packet bigger than MTU size will lead that portion of the packet will be discard	Use packet size <= MTU size
SL_WIFI_R1-934	DHCP: SL continues using its previous IP address if an invalid IP in the DHCPACK (before lease time expired)	DHCPACK arrives to SL with invalid address in the DHCPACK params address field but also the IP destination is the same invalid address (MAC address is the valid SL address). SL does not listen to other IPs address as destination but his own therefore this DHCPACK is not processed and SL continue to use his old address until the lease time expires	The device will continue to use the previous IP address	N/A
SL_WIFI_R1-1016	NS: SL device should discard ICMP Req datagram with problem in IP Header	According to the RFC – if the gateway or host processing a ICMP Req datagram and finds a problem with the header parameters such that it cannot complete processing the datagram it must discard the datagram	Low impact - The SL device sends ICMP reply message	N/A
SL_WIFI_R1-1313	NS: Error -105 when trying to open 4 TCP server sockets while the internal HTTP server is running	While the HTTP server is running one of the TCP server is been used and limit the number of user TCP Servers. Error -105 - SL_ENOBUFS [No buffer space available]	Only 3 TCP servers can be used while the HTTP is running	Disable the internal HTTP Server if 4 TCP Server need to be used
SL_WIFI_R1-1347	NS: blocking accept on secure socket doesn't return	Procedure: open secured socket, bind, listen, select on socket. Select will not return when other side is connected	Select doesn't return	Don't use select method for accept on secure socket

ID	Title	Description	Impact	Workaround
SL_WIFI_R1-1320	Transceiver mode: Can't configuring the channel of a RAW Socket if it's already open	Changing the channel while a RAW socket is open to receive by using SetSockOpt command can halt the Host. The command response on SetSockOpt doesn't return. As a result, the host is might get stuck if it configured to blocking mode.	Host might get stuck	Close the socket and open it again with the correct channel
CC3x00SDK-69	WPS PIN Connect might fail if pin code is not null-terminated	If the PIN code from the HOST is not null terminated the string can be wrongly used and in some cases the connection doesn't succeed	Connection doesn't succeed	Add null termination to the PIN code string
CC3x00SDK-166	Host: Set/Get time is limited up to year 2038	time.h (standard time library) is limiting the structure to a signed 32-bit integer, and this number is interpreted as the number of seconds since 00:00:00 UTC on 1 January 1970	Low impact – Certifications that are bounded by date will expire after year 2038	N/A
SL_WIFI_R1-930	In Enterprise network the device will Frequently Wakeup due to IPV4 BRDCST Rx frames	On enterprise network there a lot of BRDCST packets	Increase in power consumption	Add a filter to block the broadcast packets. Filter is specific to the enterprise network.

Notes for CC3200 Application Developers

SimpleLink library build: Release vs. Development

A core reset from the debugger will only reset the applications MCU and leave the networking subsystem active. This means while using the debugger on the CC3200 LaunchPad, the developer would have to manually reset the LaunchPad after every debug session before loading the new application image.

To improve the developer's experience, the following was added to the implementation of `sl_Start`:

- `sl_DeviceEnablePreamble` defined as `NwpPowerOnPreamble` in `cc_pal.c` resets the networking services
- Delay in `sl_DeviceEnable` defined as `NwpPowerOn` introduced for proper operation
- With these changes, the networking engine has to be gracefully stopped and started again for the next debug session. This includes some additional delays which result in greater overall current consumption. As these additional steps and delays are required only for debug purposes, they should not be a part of deployment applications.

For ease of use, the latest CC3200 SDK provides separate configurations of the SimpleLink library for development (debug) and deployment scenarios.

Release libraries

- NON_OS – SimpleLink for NON-OS environment
- OS – SimpleLink for OS environment
- PM_Framework – SimpleLink with Power Management Framework for OS environment
- NON_OS_PM – SimpleLink with Power Management Framework hook

Development libraries

- NON_OS_debug – SimpleLink for NON-OS environment with debug support
- OS_debug – SimpleLink for NON-OS environment with debug support
- PM_Framework_debug – SimpleLink with Power Management Framework for OS environment with debug support
- NON_OS_PM_debug – SimpleLink with Power Management Framework hook with debug support

All the above configurations are pre-built in the SDK package along with their generated output library

All the above configurations are pre-built in the SDK package along with their generated output library.

Note that the debug builds will result in overall greater current consumption. Developers should link to the release library builds for deployment by updating the library path to `simplelink.a` in their IDE.

Updating SimpleLink spawn task stack size

By default, the SimpleLink spawn task stack size is set to 2048 bytes. While this size is sufficient for the applications available in the SDK, this value may need to be changed depending on the callback implementation/call flows. In order to change the stack size, the `oslib` library must be rebuilt with the macro `SPAWN_TASK_STACK` set to the optimum stack size value (for example `-DSPAWN_TASK_STACK=3072`).

Implementation of `osi.h` is shown below:

```
#ifndef SPAWN_TASK_STACK
#define STACK_LEN (2048) /*Stack Size*/
#else
#define STACK_LEN (SPAWN_TASK_STACK)
#endif
```

Post-LPDS settings

`PRCMCC3200MCUInit` API must be called after coming out of LPDS. If this API is not called post-LPDS, then the interrupts and DMA requests for GPIO and I2C modules would not work since their initial configuration is lost after LPDS. `PRCMCC3200MCUInit` has these settings and will reconfigure these modules.

The Idle Profile example demonstrates this from the `lp3p0_restore_soc_data` implementation.

Default GPIO settings

All GPIO pins default to Mode 1 on power-up, unless programmed by the MCU. Set all unused pins except the serial flash SPI (pins 11, 12, 13, 14) and JTAG (pin 16, 17, 19, 20) to Mode 0 to prevent any bus contention.

Add following code snippet to `void PinMuxConfig(void)`:

```
PinModeSet(PIN_X, PIN_MODE_0);
```

For example, by default I2C is muxed to pins 1 and 2. If the application assigns other pins to the I2C module, this would cause contention. To resolve contention on I2C lines, add following code snippet to `void PinMuxConfig(void)`:

```
PinModeSet(PIN_01, PIN_MODE_0);
PinModeSet(PIN_02, PIN_MODE_0);
```

Technical Support

- [TI SimpleLink Solutions](#)
- [E2E Forums](#)
- www.ti.com

TI is a global semiconductor design and manufacturing company. Innovate with 100,000+ analog ICs and embedded processors, along with software, tools and the industry's largest sales/support staff.

© Copyright 1995-2020, Texas Instruments Incorporated. All rights reserved.

Trademarks | [Privacy policy](#) | [Terms of use](#) | [Terms of sale](#)