Application Report
SWRA646–January 2019

CC3235x Production Line Guide

ABSTRACT

Contents

List of Figures

List of Tables

Trademarks

1 Production Line Overview

Texas Instruments™ provides a number of resources that assist manufacturers using the CC3235x device to produce products quickly and efficiently. To ensure products are designed with efficient production in mind, TI provides reference design collateral and application notes for schematic and PCB design. Software and hardware tools have been developed for programming and testing the CC3235x device in the production line. In addition, Over-The-Air programming functionality allows for products to have their software updated periodically, even after they have been deployed.

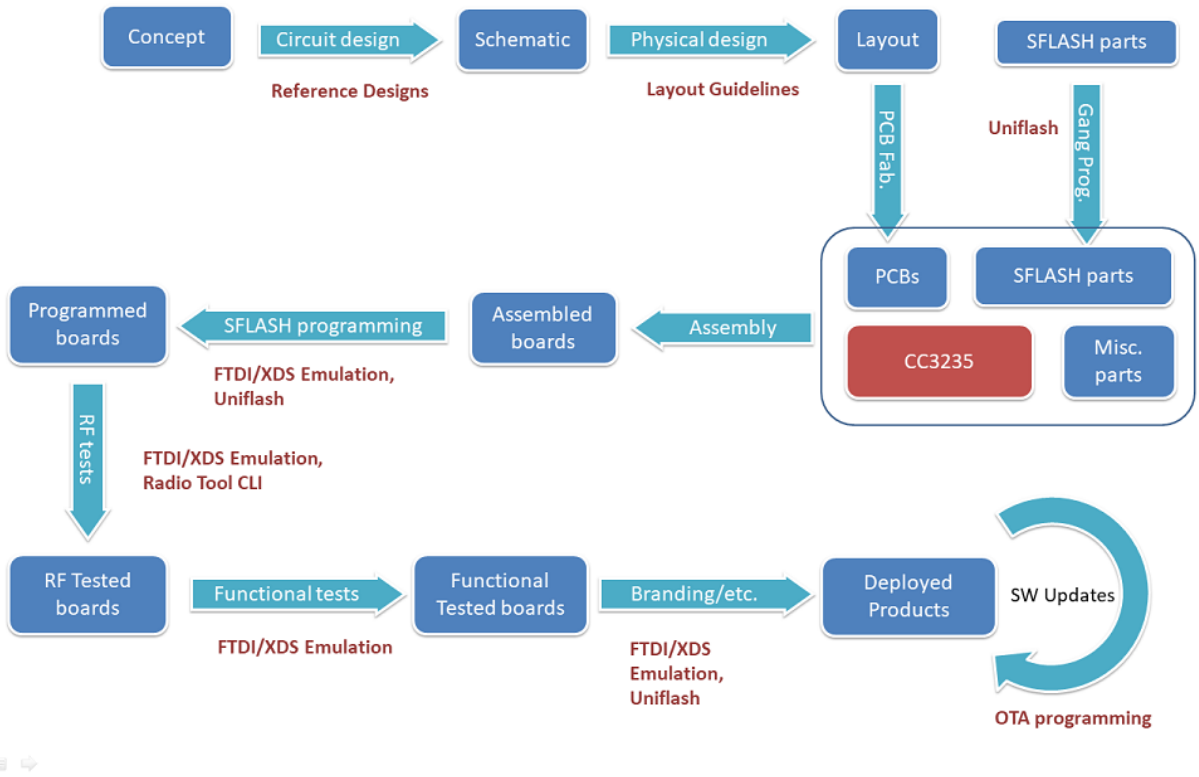

Figure 1. Production Line Overview

2 Programming the CC3235x QFN in the Production Line

Production with CC3235x device requires a file system to be created on the attached serial flash device for proper operation. At the minimum, this includes the service pack that contains the necessary software updates and additional features. The host program running on the internal Arm™ Cortex M4™ must also be programmed to the serial flash. Configuration files may also be written, which provide an initial configuration for the device upon startup. Security certificates and other content such as web pages, images, scripts, and so forth, can also be included. Although most of this content is usually written during production, all content including the host program and the service pack can be continuously updated over the lifetime of the product.

All CC3235x devices must initially be programmed with a gang image. The gang image usually contains all of the data content necessary for a functional product. The serial flash vendor may be able to pre-program the serial flash parts with the gang image before they are sent to the manufacturer. There are two methods of loading the gang image onto the CC3235x serial flash:

- Direct SPI programming – Write directly to the serial flash through SPI, using an external off-the-shelf programming tool.
- UART programming – A PC-based utility or embedded device can be used for indirectly programming the serial flash through UART.

Upon boot, the CC3235x device detects the presence of a gang image, and converts it to the target file system of the device. This formatting process is performed exclusively by the SimpleLink™ device, and does not require any inputs from external interfaces. It does, however, extend the duration of the first power-up. The formatted file system is created alongside the gang image. Thus, the serial flash should be sized to contain both the gang image and the filesystem. The gang image is retained to allow for factory default restore functionality. See *Using Serial Flash on CC3135 and CC3235x SimpleLink™ Wi-Fi® and Internet-of-Thing Devices* for more details about serial flash usage in the CC3235x device.

Generally, a single gang image is used per product release. There are some device-specific parameters that can be reconfigured through Uniflash CLI during the programming phase. Additional files can be added to the serial flash using the MCU program afterwards. If the MCU program can perform over-the-air programming, the serial flash contents can be updated by downloading content from the internet or from a local connection. Loading more content through serial interface is also possible if the MCU program allows for it.

Configure the gang image correctly for production, because some things cannot be updated through over the air updating:

- The gang image should be configured for production mode.
- The Trusted Root-Certificate Catalog used should be the production version.
- Do not use the dummy certificates and keys provided in the certificate-playground of the SDK for production.
- The formatted serial flash size cannot be changed.

3 Creating the Gang Image

The software tool used for creating gang images for the CC3235x device is Uniflash (4.1 or greater). The country code for the device should be set for allowing RF compliance with regional governmental regulations (such as US, EU, or JP). This ensures maximum RF performance, while remaining compliant with regional law. PHY calibration mode should also be set during gang programming. Other configuration options, such as startup, WLAN role, and WLAN connection policy, can also be set during programming. The image can be secured with a private key. This done when the code must be secured, and the serial flash is programmed in a location different than where the assembly takes place. See *CC3x20, CC3x35 SimpleLink™ Wi-Fi® Internet-on-a chip™ Solution Built-In Security Features* for more details about this feature.

4 Programming Directly Through SPI

The serial flash device is programmed directly, starting at memory offset 0, with one of the gang image files created by Uniflash: either Programming.bin or Programming.hex, depending on the SPI programmer. The serial flash device may be programmed after assembly on the board, provided some schematic and layout considerations are taken:

- The serial flash SPI interface pins must be brought out for physical contact with the programmer (such as headers or test pads).
- The SPI lines must not be driven by any other source while programming.
- The CC3235x device is held in reset during programming to prevent I/O contention.

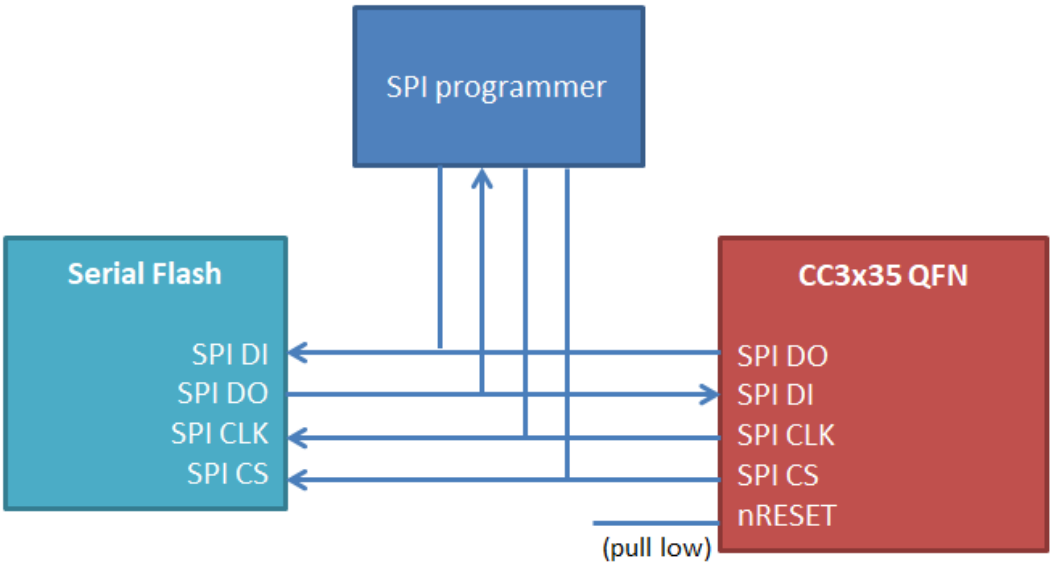

Figure 2. Programming Through SPI

5 Programming Over UART

The gang image can be programmed to the serial flash through the CC3235x device through UART communication. The CC3235x device must be put into UARTLOAD or UARTLOAD_FUNCTIONAL_4WJ boot mode using the SOP pins, see Table 1. The device can then be communicated with by using a bootloader protocol over UART. See the *CC3235S and CC3235SF SimpleLink™ Wi-Fi®, Dual-Band, Single-Chip Solution Data Sheet* for a full description of these pins.

Table 1. Programming Over UART

Mode	SOP[2]	SOP[1]	SOP[0]	
UARTLOAD	Pullup	Pulldown	Pulldown	Factory, lab flash, and SRAM loads through the UART. The device waits indefinitely for the UART to load code. The SOP bits then must be toggled to configure the device in functional mode. Also puts JTAG in 4-wire mode.
UARTLOAD_FUNCTIONAL_4WJ	Pulldown	Pullup	Pulldown	Supports flash and SRAM load through UART and functional mode. The MCU bootloader tries to detect a UART break on UART receive line. If the break signal is present, the device enters the UARTLOAD mode. Otherwise, the device enters the functional mode. TDI, TMS, TCK, and TDO are available for debugger connection.

The production process itself must include provisions for temporarily changing the state of these pins using external pulls during programming. When using Uniflash, this can be accomplished using the XDS110 on the CC3235 LaunchPad™. For other methods of programming through UART, toggling these pins must be accomplished using other automated mechanisms. The SOP pins must not be left in programming mode after production is completed, because this can cause the device to become inoperable during a restore to factory default sequence.

5.1 Using the Uniflash CLI

Uniflash communicates with the CC3235x device using the bootloader protocol. The protocol includes a command line interface, which can be used in batch files and scripts for the purposes of programming the CC3235x device in the production line through UART. Refer to *UniFlash CC3x20, CC3x35 SimpleLink™ Wi-Fi® and Internet-on-a chip™ Solution ImageCreator and Programming Tool* for information on how to use Uniflash ImageCreator.

5.2 Using Embedded Programming

In addition to using Uniflash, an embedded device can be used to program the CC3235x device through UART. This can be advantageous if a larger number of programming setups are needed to operate simultaneously, and using multiple PCs with Uniflash would be cost prohibitive. Refer to *CC313x and CC323x Simplelink™ Wi-Fi® Embedded Programming User's Guide* for information on programming using an embedded device for UART programming.

5.3 Configuration of the UART

Programming the serial flash device through the CC3235x UART interface requires the use of the following CC3235x pins:

- 55 – UART1 TX
- 57 – UART1 RX
- 32 – nRESET
- 35, 34, 21 – SOP pins

The UART TX and RX pins are used for data transfer. RTS and CTS signals are not used. The nRESET pin is used to reset the device. The UART data transfer occurs at 921600 bps.

The UART configuration is as follows:

- Baud rate: 921600
- Data bits: 8 bits
- Flow control: None
- Parity: None
- Stop bits: 1
- Polarity: Positive

The CMOS logic level specifications for the UART can be found in the *Electrical Characteristics* of the *CC3235S and CC3235SF SimpleLink™ Wi-Fi®, Dual-Band, Single-Chip Solution Data Sheet*.

5.4 UART Hardware Connection Using the CC3235 LaunchPad™

The CC3235 LaunchPad™ can provide the required USB to the UART/GPIO interface for programming the serial flash through Uniflash. The onboard XDS110 uses a logic level of 3.3 V. The PC drivers for this board are included in the CC3235 SDK, and are installed during installation of the SDK. The CC3235 LaunchPad™ is connected through USB from socket J1 to the PC. On the product being programmed, the relevant CC3235x pins must be brought out for physical contact with the programmer (such as male headers, or test pads), and must be driven by no other source while programming.

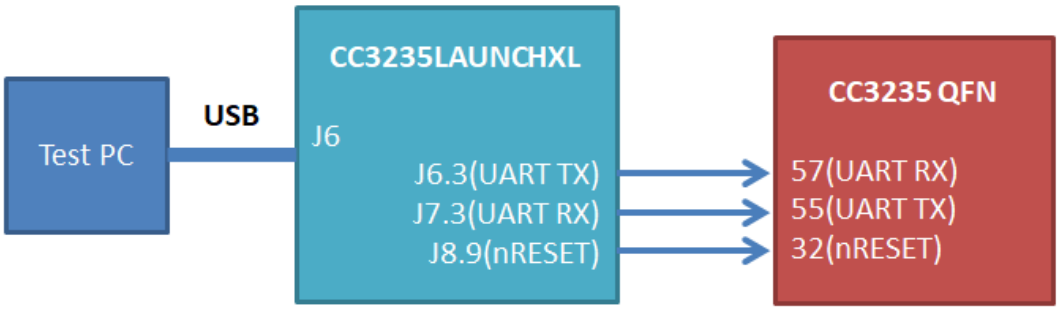

Figure 3. Connecting Through USB

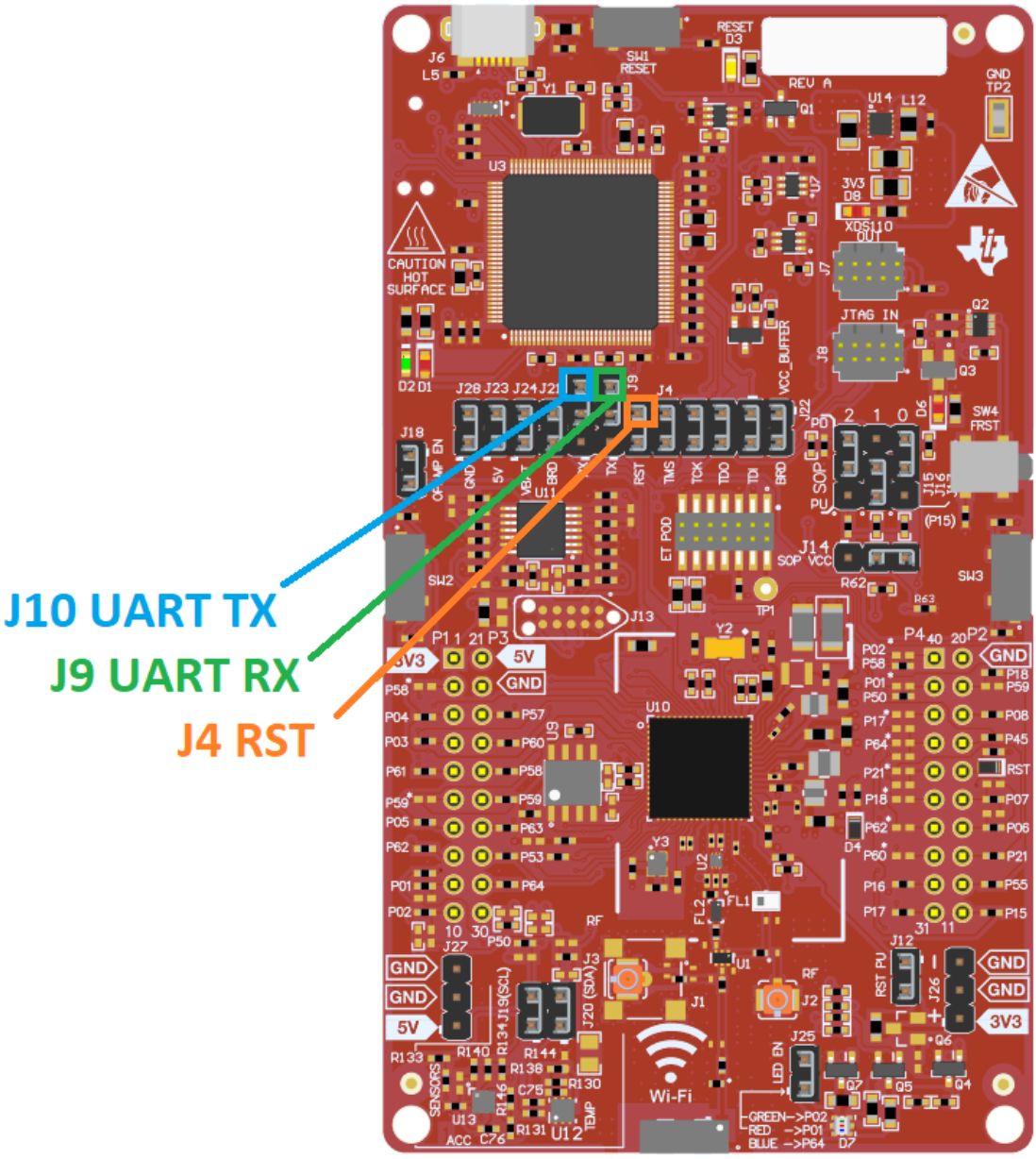

Figure 4. CC3235 LaunchPad™

6 Over-the-Air Programming

The CC3235x device has the capability for Over-the-Air programming, which allows files to be written and updated over a network connection. An OTA programming library is available in the CC3235 SDK. Using OTA in the production line can enable faster data transfer over other methods in some instances. To use OTA in the production line, a PC on the local network can run the OTA server. The service pack should be updated before doing OTA or using any wireless functionality. For the fastest transfer of data using OTA, TI recommends minimizing RF congestion in the production environment. See the *CC3x20/CC3x35 SimpleLink™ Wi-Fi® and Internet-of-Things Over the Air Update* for more information.

7 Production Line RF Testing

Testing of hardware and software functionality is highly specific to each product, but there are some tools Texas Instruments has made available to assist with testing RF performance. The CC3235x device can be instructed to perform RF testing operations in a number of ways:

- The CC3235x program may have a built-in subroutine dedicated to RF testing. This could be run once upon first power-up, or could be triggered using a special external command.
- A script using the Radio Tool CLI could control the CC3235x device from a PC, as detailed in *CC3x20, CC3x35 SimpleLink™ Wi-Fi® and Internet-on-a chip™ Solution Radio Tool*. This requires the CC3235x device to be connected to the PC through a UART to USB connection, and for a special Radio Tool program to be loaded as the MCU application.
- The CC3235x device can be controlled by interfacing with a dedicated RF tester.

7.1 Testing Software Options

7.1.1 MCU-Controlled RF Testing

SimpleLink™ API functions are available that can put the CC3235x device into modes used for RF testing. This allows for:

- Transmission of packets at specified channels, modulations, and so forth.
- Receipt of packets while gathering statistics for RSSI, modulation, and so forth.
- Carrier wave transmission

For information about the SimpleLink™ API for transceiver mode, see Chapter 13 in the *CC3x20, CC3x35 SimpleLink™ Wi-Fi® and Internet of Things Network Processor Programmer's Guide*.

7.1.2 Testing With an Access Point

A straightforward method of checking for acceptable RF performance is to put the device being tested through a trial run in an RF environment with worst-case conditions. The trial run begins with the device under test connecting to an access point, then communicating with either a PC on the local network or with a remote cloud server. The communication between the device under test and its peer can be monitored for reliability and speed. To get consistent and relevant results for all devices being tested, some actions may be taken with respect to the controlling RF environment for this type of testing:

- Minimize unintentional RF congestion in the test area. This can be accomplished by turning off other nearby 2.4-GHz band devices, or performing the testing in an RF-shielded enclosure.
- Introduce controlled RF congestion. This can involve something such as having another device connected to the same access point, which transmits a steady stream of packets to the access point.
- Introduce attenuation in the antenna path for the access point, or place at a distance from the device being tested.
- Set the access point to communicate only on a specific channel, modulation, and so forth.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265