Application Report
SWRA613–January 2019

Using Serial Flash on CC3135 and CC3235x SimpleLink™ Wi-Fi® and Internet-of-Things Devices

ABSTRACT

This application report is divided into two parts. The first part provides important guidelines and best-practice design techniques to consider when choosing and embedding a serial flash paired with the CC3135 and CC3235x (CC3x3x) devices. The second part describes the file system, along with guidelines and considerations for system designers working with the CC3x3x devices.

The CC3135 and CC3235x devices are part of the SimpleLink™ microcontroller (MCU) platform, which consists of Wi-Fi®, *Bluetooth*® low energy, Sub-1 GHz and host MCUs. All share a common, easy-to-use development environment with a single core software development kit (SDK) and rich tool set. A one-time integration of the SimpleLink™ platform lets you add any combination of devices from the portfolio into your design. The ultimate goal of the SimpleLink™ platform is to achieve 100 percent code reuse when your design requirements change. For more information, visit www.ti.com/simplelink.

Contents

Trademarks

1 Factors to Consider When Designing With Serial Flash

Many embedded systems contain a serial flash component to store firmware, configuration files, and user data for usage by a microcontroller (MCU) or processor. The processor sporadically writes data into this serial flash when updating the contents. Inclusion of serial flash memory poses unique challenges for the system designer.

- A typical serial flash has a data endurance of 100K write cycles per sector and a 20-year data retention. The write endurance and data retention characteristics must be considered by the application developer.
- The serial nature of reads and writes results in long access times, increasing the challenge of maintaining a stable system-supply voltage for the duration of the access.

1.1 CC3x3x Compatible Serial Flash Devices

Serial flash components from some vendors may appear to be equivalent in memory capacity, but an examination of the serial flash data sheets can reveal significant parametric differences between components in areas such as operating voltage and access times.

1.1.1 Supported Instructions

For compatibility with the CC3x3x device, the serial flash device must support the following commands and format:

- Uniform sector erase size of 4K.
- Command 0x9F (read the device ID [JEDEC]). Procedure: SEND 0x9F, READ 3 bytes.
- Command 0xB9 (power down). Procedure: SEND 0xB9, read status until busy bit is cleared.
- Command 0xAB (power up). Procedure: SEND 0xAB, read status until busy bit is cleared.
- Command 0x05 (read the status of the SFLASH). Procedure: SEND 0x05, READ 1 byte. Bit 0 is busy and bit 1 is write enable.
- Command 0x01 (write the status of the SFLASH). Procedure: SEND 0x01, SEND 1 byte. Bit 0 is busy and bit 1 is write enable.
- Command 0x06 (set write enable). Procedure: SEND 0x06, read status until write-enable bit is set.
- Command 0xC7 (chip erase). Procedure: SEND 0xC7, read status until busy bit is cleared.
- Command 0x20 (sector erase). Procedure: SEND 0x20, SEND 24-bit address, read status until busy bit is cleared.
- Command 0xD8 (sector erase). Procedure: SEND 0xD8, SEND 24-bit address, read status until busy bit is cleared.
- Command 0x03 (read data). Procedure: SEND 0x03, SEND 24-bit address, read n bytes.
- Command 0x02 (write page). Procedure: SEND 0x02, SEND 24-bit address, write n bytes (0<n≤256).

1.1.2 Supported Vendors and Parts Numbers

The serial flash components listed in TI BOM tables for the CC3x3x device reference designs can be used, because TI has system-tested these serial flash components. Table 1 lists the different parts tested with the CC3x3x devices. TI found these parts reliable through a series of system-level tests, to ensure robustness under various operating conditions. However, this does not ensure data integrity under extreme operating conditions, as specified in subsequent sections of this document.

More information can be found in the *CC31xx, CC32xx SimpleLink™ Wi-Fi® and Internet of Things Network Processor Programmer's Guide* (see the *Design Consideration* section under the File System paragraph).

Table 1. Serial Flash Parts Tested With the CC3x3x Devices

Vendor	Part Number	Size	Voltage Power	Recommendations
Macronix	MX25R3235FM1IL0	32Mb	1.65 V to 3.6 V	Battery and line-powered systems
ISSI	IS25LQ016B	16Mb	2.3 V to 3.6 V	Battery and line-powered systems
ISSI	IS25LQ032B	32Mb	2.3 V to 3.6 V	Battery and line-powered systems
Adesto	AT25DF321A	32Mb	2.7 V to 3.6 V	Battery and line-powered systems

To get the best access time performance, designers should consider the following parameters: Page Program Time, 4KB Sector Erase Time, and 64KB Block Erase Time. In addition, when high-performance mode is available, it can be used to leverage performance (at the expense of power consumption). Table 2 lists some access time parameters for the tested parts.

Table 2. Serial Flash Parameters

Vendor	Part Number	4KB Sector Erase Time [mSec]	64KB Block Erase Time [mSec]	Page Program Time [mSec]
Macronix (high performance)	MX25R3235FM1IL0	40 to240	480 to 3000	0.85 to 4
ISSI	IS25LQ032B	70 to 300	200 to 1000	0.5 to 2
Adesto	AT25DF321A	50 to 200	400 to 950	1 to 3

1.1.3 Serial Flash Write or Erase Endurance Limitations

The lifespan of a serial flash component is subject to a maximum number of write/erase cycles. This should be considered when designing the system application software.

General guidelines for increasing flash endurance follow:

- Minimize the number of application writes to flash, especially after reset. For example, ensure that application configuration writes to flash occur only at initial reset, and not every time the CC3x3x device is reset.
- The creation and deletion of files requires updates to the File Allocation table (FAT). When an update to an existing file is required, do not delete and then recreate the file, because unnecessary access to the FAT should be avoided.]
- More information can be found in the *CC31xx, CC32xx SimpleLink™ Wi-Fi® and Internet of Things Network Processor Programmer's Guide* (see the *Design Consideration* section in the *File System* chapter).

A typical serial flash ensures a data endurance of 100K write cycles per sector, and a 20-year data retention. Table 3 details the maximum number of writes-per-day to the same sector, allowing the device to operate for a given number of years.

Table 3. Serial Flash Endurance

Desired Product Life [Years]	Maximum Writes-per-Day(1)
20	14
15	18
10	27
5	55
2	137

(1) Maximum number of writes each day = 100000 / (product life in years × 365)

In the CC3x3x system, the serial flash may be written by the user application or by periodic activity of the CC3x3x on-chip firmware. The total number of writes from these two sources should not exceed the budget for the maximum number of flash writes calculated for the desired product lifetime.

1.2 CC3x3x-Incompatible Serial Flash Devices

Some serial flash components have supplementary mechanisms that may not be compatible with the CC3x3x devices. Careful attention must be taken whether these mechanisms are enabled by default, because the serial flash driver is implemented on the device firmware and cannot be altered easily.

Among these mechanisms:

- Serial flash protection: Some parts have a global protection for the entire flash, and some have per-sector protection where each sector has a dedicated bit in a register. To “unlock” a sector (or the entire flash), a dedicated SPI command is usually necessary (usually writing to a status register). Additionally, check whether the unlocking configuration is persistent over reset or not. If it is persistent, then the unlocking operation can be applied externally, even before the serial flash part is soldered on the PCB.
- Quad SPI: Some parts have a Quad SPI support (4 lines instead of 2). Users must ensure that QSI is not enabled by default, whether it can be modified programmatically, and if it is persistent or not.

Table 4 lists some vendors and parts numbers which implement a special mechanism.

Table 4. Serial Flash Parts Not Compatible With CC3x3x Devices

Vendor	Part number	Incompatibility Reasons
Microchip	SST26VF032BA	Default with SIO2 and SIO3 pins enabled, which initiates QUAD I/O operations Default protected after a power-on reset cycle, requires a dedicated global unlock command
Microchip	SST26VF032B	Default protected after a power-on reset cycle, requires a dedicated global unlock command
Adesto	AT45DB321E	Default per-sector protection – no global unprotect
Winbond	W25Q32JVSSIQ	Default with SIO2 and SIO3 pins enabled, which initiates QUAD I/O operations

1.3 Best Practice Design Techniques for System Robustness

1.3.1 Overview

The CC3x3x devices are proven, robust WLAN solutions when operated in accordance with the supply and signal parameters described in the data sheet. This section describes design techniques to maximize system robustness. These techniques include minimizing the possibility of inadvertent corruption of the serial flash memory connected to the CC3x3x devices for battery-powered and hybrid line/battery-powered designs. The techniques described focus on ensuring the integrity of the power supply to the serial flash, to avoid situations where the supply voltage falls below the minimum threshold specified by the serial flash manufacturer, causing corruption of flash data during write or erase operations. The nature of such corruption systems determines whether or not systems continue to operate, and in some cases a physical access to the system may be required to recover.

1.3.2 General Guidelines

General guidelines for achieving system robustness follow:

- Apply primarily to systems that are powered from batteries or by a hybrid line and battery scheme. This includes designs where the CC3x3x device connects directly to the battery, and systems where a DC-DC converter is used between the battery and the CC3x3x device.
- If the application uses a DC-DC converter, designers should ensure that the output of the DC-DC converter satisfies the supply requirements of both the CC3x3x device and the attached serial flash device.
- The CC3x3x devices should be enabled only when the supply voltage is greater than or equal to 2.3 V. This minimum is typically determined by the serial flash minimum supply voltage and not the CC3x3x minimum supply voltage, which is defined in the data sheet as 2.1 V. As specified in the data sheet, the supply must tolerate a CC3x3x transmit current or calibration current load without sagging below 2.3 V; therefore, the unloaded supply voltage may have to be greater than 2.3 V to account for internal resistance effects of the supply.
- The supply voltage applied to the CC3x3x device should never exceed 3.8 V, which is specified as the absolute maximum supply voltage in the data sheet. The corresponding absolute maximum voltage constraints from the chosen serial flash data sheet must also be followed.
- For better flash endurance, follow the guidelines in Section 1.1.3.
- For maximum system robustness, use a serial flash such as the Macronix MX25R3235. This device supports a wide supply voltage range, which tends to improve system immunity to supply fluctuations.
- For access time performance, choose serial flash parts that have better data sheet numbers. The important parameters include: Page Program Time, 4KB Sector Erase Time, and 64KB Block Erase Time.
- If the serial flash part has a high performance mode, it may be configured to leverage performance.
- The CC3x3x WLAN transmission can result in sudden increases in the loading on the power supply. The sudden increases may result in a momentary decrease in supply voltage. Consult the CC3x3x data sheet section that describes how to handle supply brownout.

1.3.3 Sudden Power Off

All systems using serial flash are vulnerable to the effects of sudden power removal. As noted in most serial flash data sheets, a data corruption may occur if the power is removed while a write or erase operation is in progress. This can occur if the system operating voltage goes below V_{min} of the serial flash (2.3 V typical) before the erase or write operation is completed. Figure 1 shows a typical scenario.

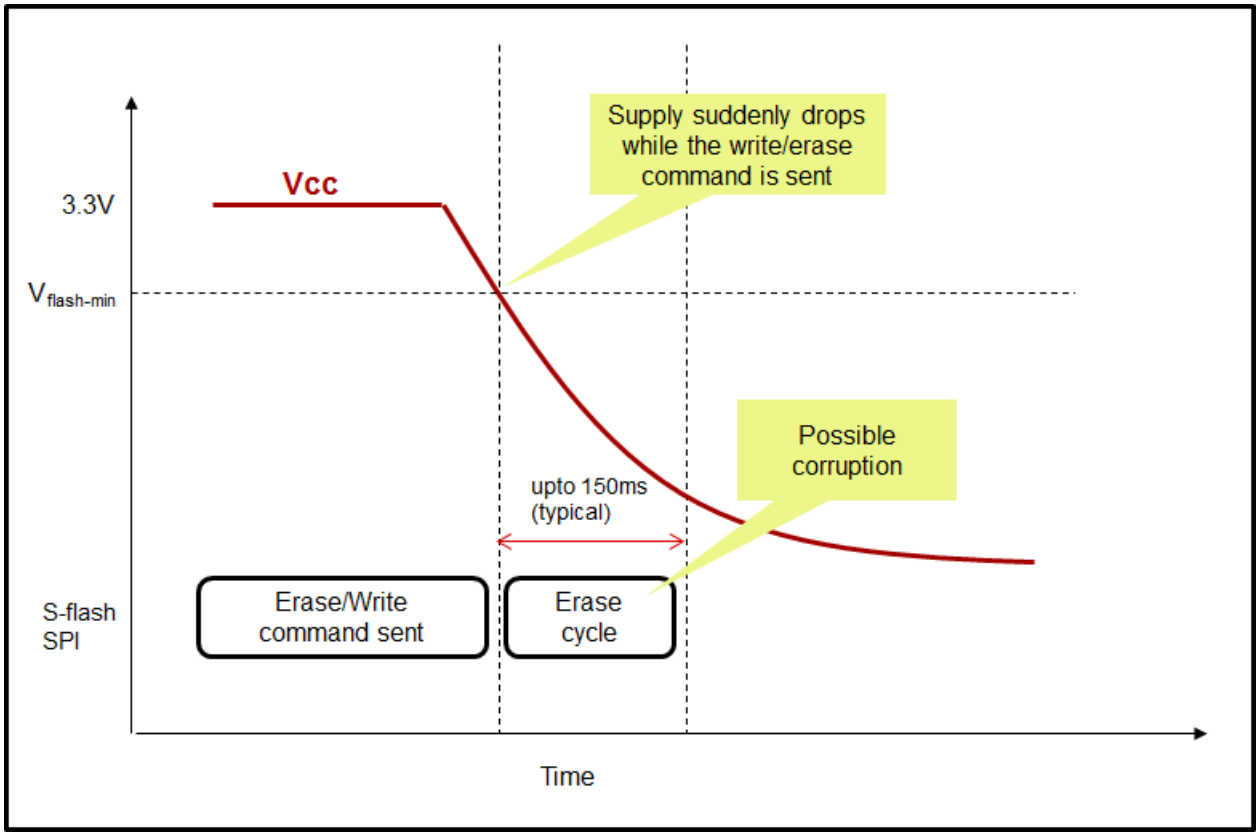

Figure 1. Sudden Power Off During Flash Erase

This scenario may occur in battery or line-powered end equipment:

- Battery-powered products: Removal of battery from the product without a soft shutdown
- Line-powered products: Electrical supply failure or sudden unplugging without a soft shutdown

The following sections explain how the potential for serial flash corruption can be minimized in both types of end equipment.

1.3.3.1 Battery-Powered Systems

In a battery-powered system, a sudden removal of power could coincide with an ongoing serial flash access. The chance of this occurring can be minimized by battery removal difficult while the product is in use, such as in a Wi-Fi® weighing scale, where the user stands on the scale when the product is in use. Alternatively, product instructions could discourage users from removing the batteries while the product is in use. In systems desiring a higher degree of protection, a soft power-down push-button that maps to a GPIO of the system processor could also be provided to give a warning in advance that the user wants to remove the battery. In this system, the processor could use an LED to indicate that soft shutdown has completed, and that the batteries can now be removed.

1.3.3.2 Line-Powered Systems

In an AC main line-powered system, the failure of the grid can cause the supply of the Wi-Fi® subsystem to suddenly drop. In the unlikely event that a grid failure and corresponding sudden drop in the Wi-Fi® subsystem would coincide with the erase operation of the serial flash, data corruption may occur. One of the ways to minimize the chance of a flash corruption is to ensure that the DC voltage ramps down slowly after the input power is removed. Ensure that the DC voltage ramps down slowly with the help of bulk capacitors, which hold the charge while the input supply is removed. The value of the capacitor must be estimated from the maximum time for the SFLASH erase operation, voltage thresholds, and the current drawn by the system. The embedded system senses the sudden fall in the input voltage and initiates a soft shutdown of the Wi-Fi® subsystem, thus safely completing all serial flash operations before $V_{flash-min}$ is reached. The charge stored on the capacitor would be used during this brief interval. Figure 2 shows this sequence.

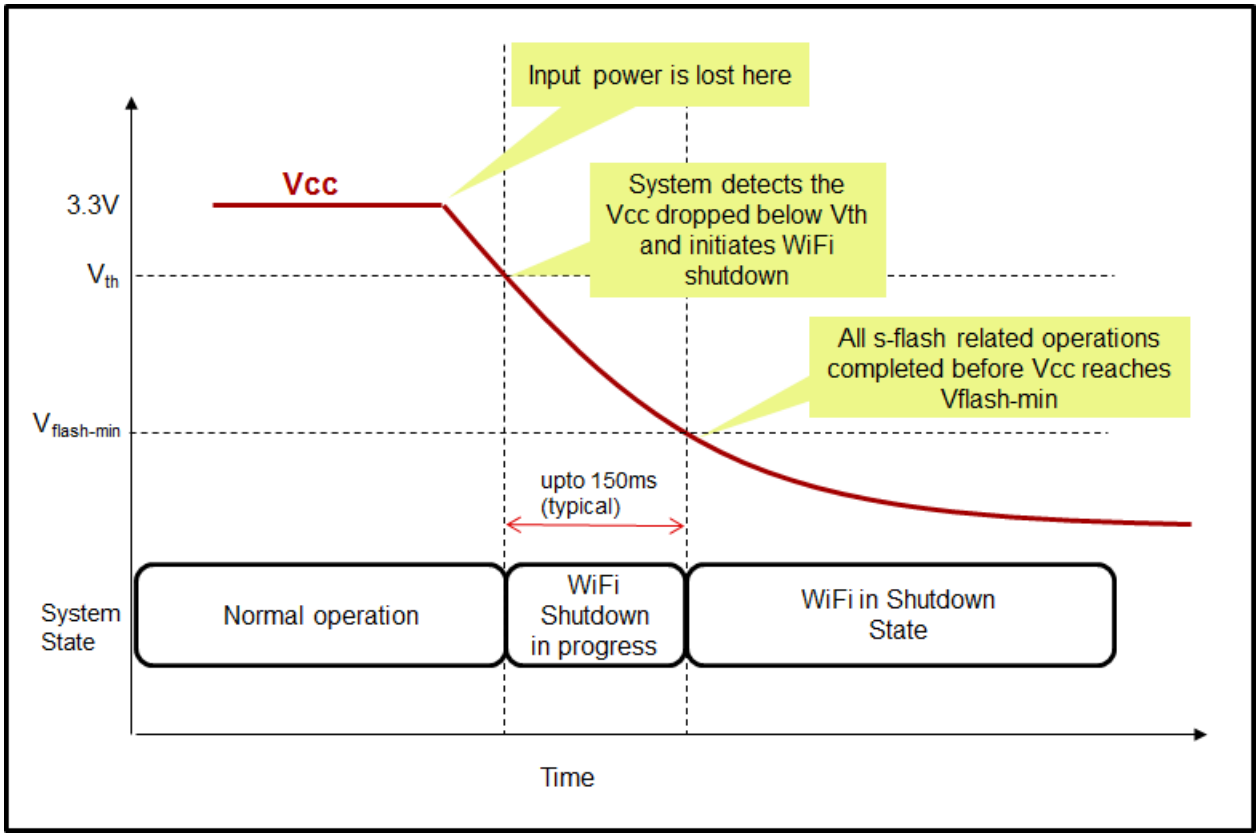

Figure 2. Sudden Power Off in Line-Powered Device

2 Factors to Consider When Using CC3x3x File System

2.1 Overview

The second part of this guide discusses the CC3x3x file system implemented on the serial flash. Unlike the serial flash, which can be chosen by system designers, the CC3x3x file system is common for all devices. System and configurations files can only be created and maintained using the CC3x3x file system.

More detailed information on the CC3x3x file system can be found in the *CC31xx, CC32xx SimpleLink™ Wi-Fi® and Internet of Things Network Processor Programmer's Guide* (see the *File System* section).

2.2 File System Guidelines

Most of file system guidelines can be found in *CC31xx, CC32xx SimpleLink™ Wi-Fi® and Internet of Things Network Processor Programmer's Guide* (see the *File System Characteristics* section under the *Secure File System* paragraph). In addition to those guidelines, system designers should also consider the following:

- The file system allocation table consumes five blocks (20KB).
- The serial flash storage type that the CC3x3x device supports has a minimal block size of 4096 bytes.
- Each file consumes at least:
  - One block (4KB) for a file with no fail-safe support
  - Two blocks (8KB) for a file with fail-safe support
- File size cannot be enlarged after the file has been created. To increase the file content during the device life cycle, the maximum size attribute should be set when the file is created (the file system reserves space).
- File attributes cannot be modified after the file has been created (apart from commit and rollback attributes).
- The file system does not handle fragmentation.

2.3 File Memory Space Mathematics

Because users may use the file system to store their own files, it is important to be able to accurately calculate the occupied memory space per file. The total occupied size on the flash is a function of the file content length (or the maximum size attribute upon creation), file attributes, and file system metadata.

Figure 3 shows the process for calculating how much memory is consumed on the serial flash.

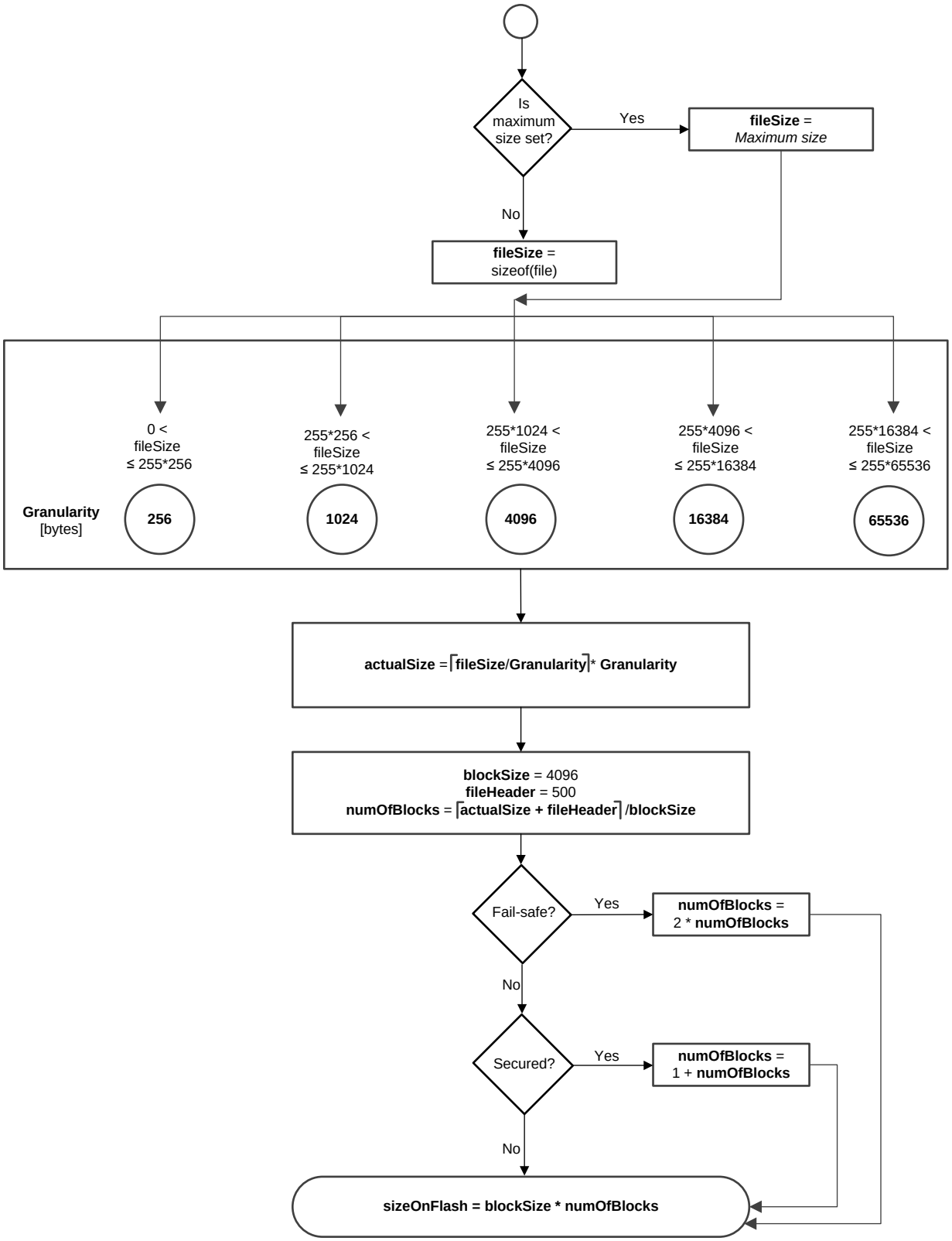

Figure 3. File Memory Consumption on Flash

2.4 System Files

2.4.1 Overview

System files include either a subset or all configuration files that set the device into a predefined state. System files can be implicitly created by the user (such as through invoking a host driver API), or internally created by the device. Because these files are essential for proper operation of the device, system designers must understand when these files are created and how to monitor the serial flash occupancy.

Failing to preserve enough space for the system files may cause unexpected behavior in the system.

2.4.2 Host Driver Mapping

Most of the system files are created implicitly by users when invoking a host driver API. Some system files are internally created by the network processor subsystem, and the user does not have control over these files (for example, calibration files, ARP table files, and so forth).

For detailed information on the host driver API implicitly attached with the system files, see the *CC31xx, CC32xx SimpleLink™ Wi-Fi® and Internet of Things Network Processor Programmer's Guide* (see the *Persistency* appendix).

2.4.3 How to Profile Serial Flash Content

Designing and monitoring the content on serial flash is mandatory, especially in embedded devices where memory resources are limited. New generation SimpleLink™ devices provide some options for system designers to both help design the system and monitor it.

2.4.3.1 Using UniFlash

UniFlash ImageCreator is a tool used to program the external serial flash used by the SimpleLink™ Wi-Fi® CC3220 and CC3120 devices. More information can be found in the*UniFlash CC3120, CC3220 SimpleLink™ Wi-Fi® and Internet-on-a chip™ Solution ImageCreator and Programming Tool User's Guide*.

2.4.3.1.1 Monitor the Storage Breakdown (Available Only During Operational Mode)

For this option to work, the device must be opened in development mode. All that is required is to connect to the device and click the file listing button.

Figure 4 is taken from the Out-of-Box project and shows the file listing (the file listing button is marked with a blue circle).

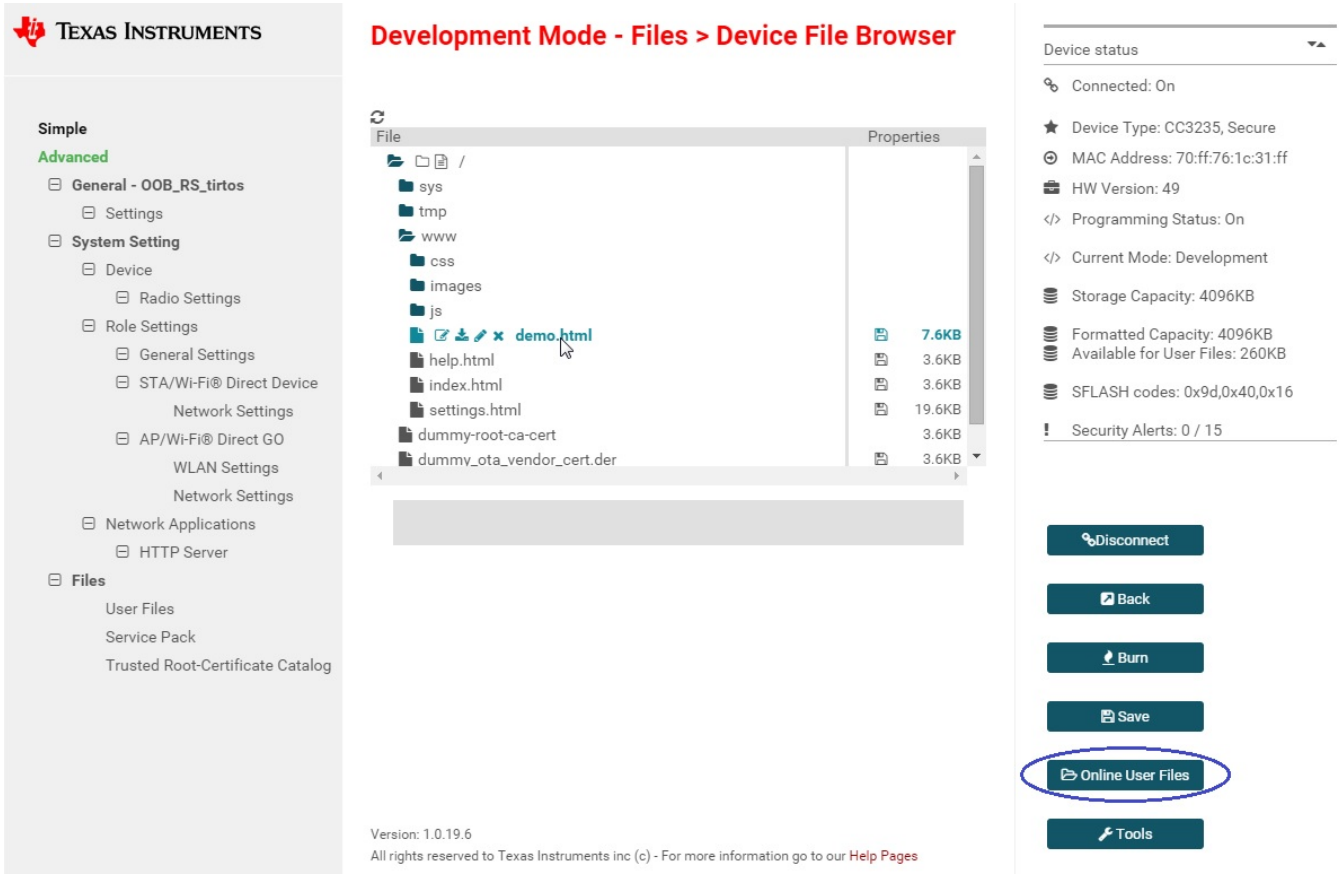

Figure 4. UniFlash File Listing

2.4.3.2 Using Host Driver APIs

Storage content can be monitored in real time using a host driver API.

There are two APIs for this purpose:

- The file list interface displays information regarding the existing files and the number of allocated blocks per file.
- The Get storage information command contains information about the device usage; it contains information regarding the device memory usage, information about security alerts in the system, the number of times FAT has been accessed, and so forth.

Figure 5 shows the file listing feature for the out-of-box invoked through the file system API and printed on the terminal.For this purpose, the AT Commands application was used and programmed into the device instead of the out-of-box application.

The file system API for listing the file system content is sl_FsGetFileList() API.

The AT command for listing the file system content is At+FileGetFileList.

The ordered parameters retrieved per file through the AT commands terminal are:

- The full path on the SimpleLink™ file system.
- Allocated size in bytes (not including the file header of 440 bytes). The size is for a single copy of the file and does not take into account the second copy in case the file is fail-safe.
- File attributes.
- Allocated size in blocks including fail-safe copy.

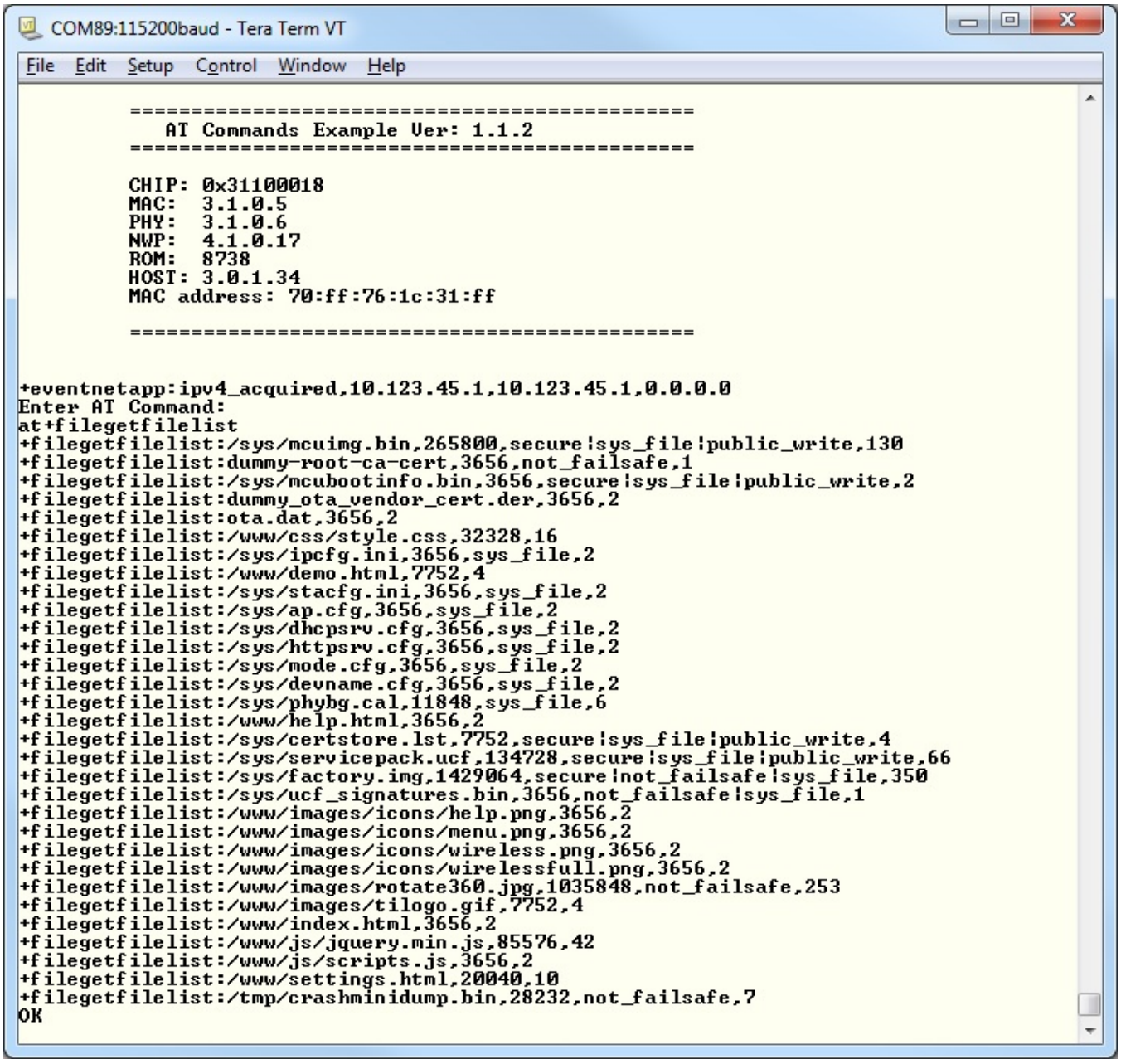

Figure 5. File Listing Through Host API

Figure 6 shows the storage info feature for the out-of-box invoked through the file system API and printed on the terminal. Storage information can be invoked by executing sl_FsCtl() API with the SL_FS_CTL_GET_STORAGE_INFO command.

The AT command for listing the storage information is At+FileCtl=GET_STORAGE_INFO,,,.

The ordered parameters retrieved through the AT commands terminal are:

- Device block size
- Device capacity in blocks
- Number of allocated blocks
- Number of reserved blocks
- Number of reserved blocks for system files
- Largest allocated memory gap in blocks
- Number of available blocks for user files
- Maximum possible files in the file system
- Is the device formatted in development mode?
- The file system bundle state
- Maximum number of files reserved for system files
- Actual number of user files
- Actual number of system files
- Number of system security alerts
- System security alerts threshold
- Counter of FAT write access

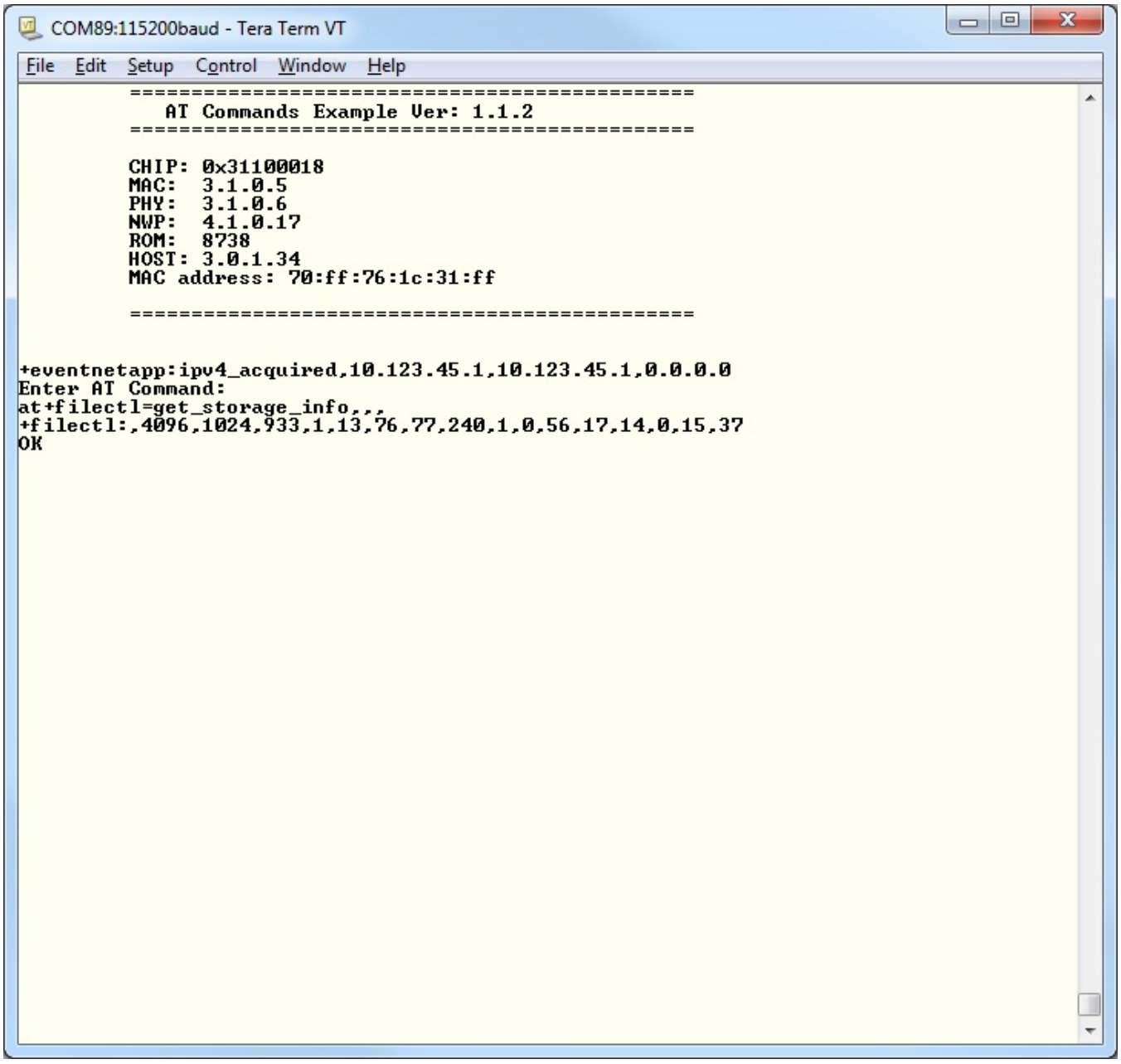

Figure 6. Get Storage Information Through Host API

2.4.4 Flash Recommended Size

The recommended size of the serial flash is determined by the memory space occupied by all system files, and by the programming image if restore-to-factory is enabled.

System designers can either reserve the entire space, or customize and reserve only the required space. In the latter case, take caution because not preserving enough space may cause unexpected behavior in the system.

Table 5 lists the minimal required memory consumption under the following assumptions:

- System files in use consume 64 blocks (256KB).
- Vendor files are not considered.
- MCU code is taken as the maximal possible size for the CC3235x with fail-safe enabled to account for future updates, such as through OTA.
- Gang image:
  - Storage for the gang image is rounded up to 32 blocks (meaning 128KB resolution).
  - Gang image size depends on the actual content size of all components. Service pack, system files, and the 32-block resolution are assumed to occupy 256KB.
- All calculations consider that the restore-to-default is enabled.

Table 5. Recommended Flash Size

Element	CC3135 [KB]	CC3235S [KB]	CC3235SF [KB]
File system allocation table	20	20	20
System and configuration files	256	256	256
Service pack	264	264	264
MCU code	N/A	512(1)	2048(1)
Gang image size	256	256 + MCU	256 + MCU
Total	796	1308 + MCU	2844 + MCU
Minimal flash size(2)	8Mb	16Mb	32Mb
Recommended flash size(2)	16Mb	16Mb	32Mb

(1) Including fail-safe.
(2) For maximum MCU size.

3 Production Line Programming

3.1 Overview

Production line programming is the process of flashing the gang image created by the Uniflash utility into the device serial flash. The gang image, at a minimum, contains the service pack and the binary host application for the CC3235x. Secure parts must also include security-related files such as certificates, keys, and the certificates catalog. In addition, system and configuration files are included, as well as optional user files.

There are two main methods for image programming:

- Direct SPI programming – Directly write to the serial flash through the SPI lines, using an external off-the-shelf programming tool.
- UART programming – Either using the Uniflash utility or the embedded programming package to indirectly program the serial flash through UART.

In both cases, during first boot the device detects the gang image, and starts extracting the image and creating the file system. This formatting process is performed solely and exclusively by the SimpleLink™ device, and does not require any user intervention.

This section discusses the factors that contribute to the total programming time duration. More information can be found in the *CC31xx, CC32xx SimpleLink™ Wi-Fi® and Internet of Things Network Processor Programmer's Guide* , *CC3135 Production Line Guide Application Report*, and *CC3235x Production Line Guide Application Report*.

3.2 Factors Affecting Programming Time

3.2.1 SPI Programming

The following factors contribute to the total programming time duration when programming through SPI lines.

- Serial flash access time parameters, as listed in Table 6.
- SPI clock rate
- Serial flash mode, if applicable. Some flash parts may be configured to work in high performance mode, which is faster than working in low power mode.
- The off-the-shelf tool used for programming. Specifically, some tools work faster when using the hexadecimal format of the image. Users can also program the tool to work faster with some utilities.
- The entire serial flash space must be erased before programming.
- For the CC3235SF device, the default size of the host application binary is 1MB with fail-safe, that is, 2MB.
- For the CC3235S device, the default size of the host application binary is 256KB with fail-safe, that is, 512KB.
- Size of the image. The image gets larger if:
  - Files are configured in fail-safe mode.
  - Files are secured.
  - Files are configured with a predefined size that is larger than the original file size.

3.2.2 UART Programming

The following factors contribute to the total programming time duration when programming through UART lines.

- Serial flash access time parameters, as listed in Table 6.
- Serial flash mode, if applicable. Some flash parts may be configured to work in high performance mode, which is faster than working in low power mode.
- UART interface speed is constant and set to 921,600bps.
- The device is in boot loader mode during programming. The programming protocol over UART has overhead.
- For the CC3235SF device, the default size of the host application binary is 1MB with fail-safe, that is, 2MB.
- For the CC3235S device, the default size of the host application binary is 256KB with fail-safe, that is, 512KB.
- Size of the image. The image gets larger if:
  - Files are configured in fail-safe mode.
  - Files are secured.
  - Files are configured with a predefined size that is larger than the original file size.

3.3 Image Programming Examples

To get a estimated time duration for the serial flash listed in Table 6, the out-of-box (OOB) image is used. Because the OOB image is large, the pre-allocated size of the binary host application must be set to 256KB for the CC3235SF device (otherwise, the image would not fit into a 32Mb serial flash). Using the same size binary host application allows the user to use either the CC3235S or CC3235SF device.

Because the size of the image and the extracted file system occupies almost the entire flash memory space, the total programming time is an estimation of the maximal duration for the specific flash part. The flash content breakdown for the OOB image is (blocks of 4KB):

- Image size – 352 blocks
- Size of user files after extraction – 357 blocks
- Size of system files after extraction – 214 blocks

The total size that must be erased is thus 923 blocks, which is 3692KB.

Table 6. Serial Flash Parameters

Vendor	Part Number	4KB Sector Erase Time [mSec]	64KB Block Erase Time [mSec]	Page Program Time [mSec]	Image Programming [seconds]	Image Extraction [seconds]
Macronix (low power)	MX25R3235FM1IL0	58 to 240	800 to 3500	3.2 to 10	55	59
ISSI	IS25LQ032B	70 to 300	200 to 1000	0.5 to 2	41	21
Adesto	AT25DF321A	50 to 200	400 to 950	1 to 3	36	22

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