Application Report
SWRA515–February 2017

Using Serial Flash on CC3120 and CC3220 SimpleLink™ Wi-Fi® and Internet-of-Things Devices

ABSTRACT

The CC3120 and CC3220 devices are part of the SimpleLink™ microcontroller (MCU) platform, which consists of Wi-Fi®, *Bluetooth*® low energy, Sub-1 GHz and host MCUs, which 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 enables you to add any combination of the portfolio's devices into your design, allowing 100 percent code reuse when your design requirements change. For more information, visit www.ti.com/simplelink.

This application note 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 CC3120 and CC3220 (CC3x20) devices. The second part describes the file system, along with guidelines and considerations for system designers working with the CC3x20 devices.

Contents

SimpleLink is a trademark of Texas Instruments.
Bluetooth is a registered trademark of Bluetooth SIG.
Wi-Fi is a registered trademark of Wi-Fi Alliance.
All other trademarks are the property of their respective owners.

1 Factors to Consider in 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 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 20 years 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, raising the challenge of maintaining a stable system-supply voltage for the duration of the access.

1.1 Serial Flash Vendor and Part Number Selection

Serial flash components from some vendors may appear to be equivalent in memory capacity, but close examination of the serial flash data sheets can reveal significant parametric differences between components in areas such as operating voltage and access times. The serial flash components listed in TI BOM tables for the CC3x20 reference designs can be used, because these serial flash components have been system-tested by TI. Table 1 lists the different parts tested with the CC3x20 devices. These parts were found reliable by TI 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 *CC3x20 SimpleLink Network Processor Programmer's Guide* (see the *Secure File System* paragraph).

Table 1. Serial Flash Parts Tested With the CC3x20 Devices

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

1.2 Supported Flash Types

For compatibility with the CC3x20 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 0x05 (read the status of the SFLASH). Procedure: SEND 0x05, READ 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 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 ($0<n\leq256$)
- Command 0x20 (sector erase). Procedure: SEND 0x20, SEND 24-bit address, read status until busy bit is cleared

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 are:

- 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 CC3x20 is reset.
- The creation and deletion of files requires updates to the FAT table. Avoid deleting files when not required, as a rewrite can be done over an existing file. This reduces unnecessary access to the FAT table.
- More information can be found in the *CC3x20 SimpleLink Network Processor Programmer's Guide* (refer to the *Software Design Consideration* section under the *Secure File System* paragraph).

A typical serial flash ensures a data endurance of 100K write cycles per sector, and 20 years data retention. Table 2 details the maximum number of writes-per-day to the same sector, allowing the device to operate for a given number of years.

Table 2. 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/day = 100000 / (product life in years * 365)

In the CC3x20 system, the serial flash may be written by the user application or by periodic activity of the CC3x20 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.

1.4 Best Practice Design Techniques for System Robustness

1.4.1 Overview

The CC3x20 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, including minimizing the possibility of inadvertent corruption of the serial flash memory connected to the CC3x20 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. Depending on the nature of such corruption, systems may or may not continue to operate, and in some cases a physical access to the system may be required to recover.

1.4.2 General Guidelines

- Apply primarily to systems that are powered from batteries or by a hybrid line and battery scheme. This includes designs where the CC3x20 connects directly to the battery, and systems where a DC2DC converter is used between the battery and the CC3x20.
- If the application uses a DC2DC converter, designers should ensure that the output of the DC2DC converter satisfies the supply requirements of both the CC3x20 and the attached serial flash device.
- The CC3x20 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 CC3x20 minimum supply voltage, which is defined in the data sheet as 2.1 V. The supply must tolerate a CC3x20 transmit current or calibration current load, as specified in the data sheet, without sagging

below 2.3 V, so that 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 CC3x20 should never exceed 3.8 V, 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 guidelines under Section 1.3.
- For maximum system robustness, use a serial flash such as the Macronix MX25R6435.This device supports a wide supply voltage range, which tends to improve system immunity to supply fluctuations.
- The CC3x20 WLAN transmission can result in sudden increases in the loading on the power supply, which may result in a momentary decrease in supply voltage. Consult the CC3x20 data sheet section describing how to handle supply brown out.

1.4.3 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 happen 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. A typical scenario is shown in Figure 1.

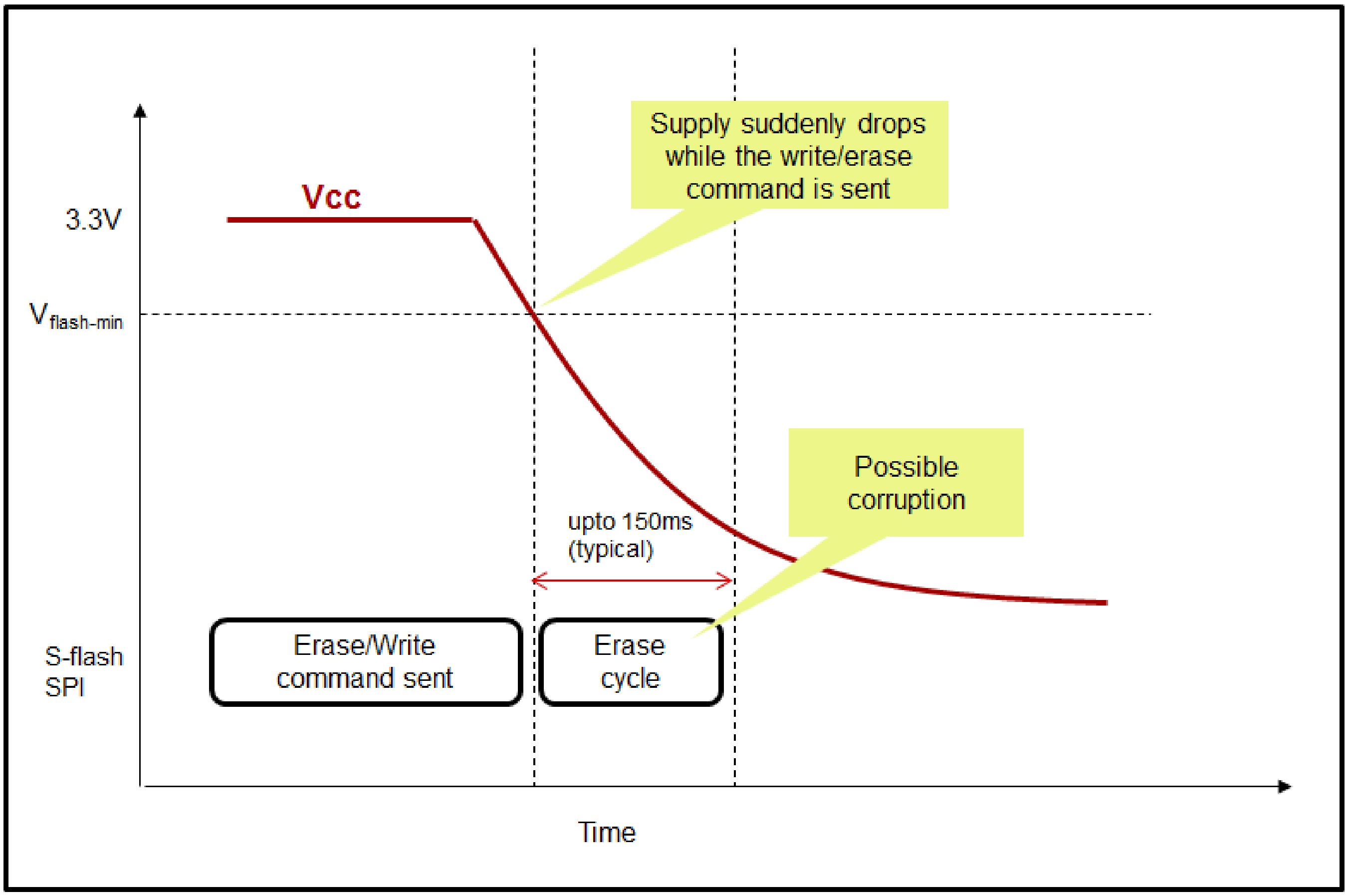


Figure 1. Sudden Power Off During Flash Erase

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.

1.4.3.1 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 happening can be minimized by making it difficult to remove the batteries while the product is in use, such as in a Wi-Fi weighing scale, where the user stands on the scale when it 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 could also be provided that maps to a GPIO of the system processor, giving it 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 shut down has completed, and that the batteries can now be removed.

1.4.3.2 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 this coincides with the erase operation of the serial flash, there is a chance of data corruption. 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. This can be achieved 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 s-flash 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 sub-system, thus safely completing all serial flash operations before $V_{flash\text{-}min}$ is reached. The charge stored on the capacitor would be used during this brief interval. This sequence is illustrated in Figure 2.

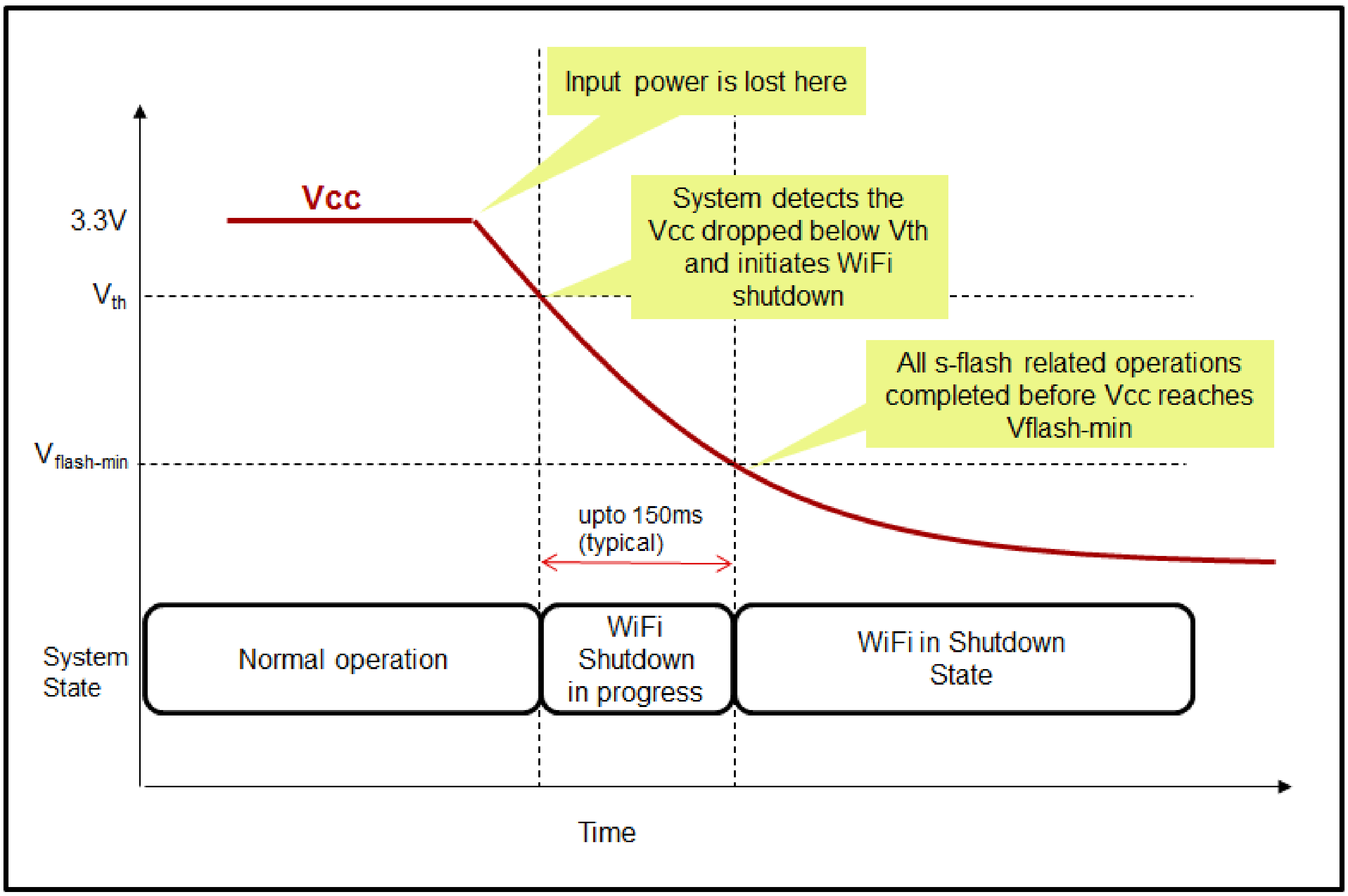


Figure 2. Sudden Power Off in Line-Powered Device

1.4.4 Brownout Mitigation Techniques for Self-Hosted CC3220 Designs

1.4.4.1 Overview

This section describes how to mitigate the effects of a substantially discharged battery on a self-hosted CC3220 system. It is assumed that the device is the main controller and has the ability to control all high power components in the system.

1.4.4.2 Problem Description

The brownout problem can occur when operating with a substantially discharged battery that has enough energy to power up the CC3220 processor, but not enough to power the Wi-Fi transmitter. This can cause a loop where the CC3220 device powers up and reaches the point where it does some high-power activity. This activity causes the battery voltage to drop below the brownout threshold, causing a reset. Once reset, the device consumes no power, the voltage rises back above the brownout threshold, and the device powers up again.

1.4.4.3 Suggested Solution

As a generic approach, CC3220 initialization does not turn on the internal networking processor function, so the CC3220 application can start running prior to Wi-Fi activity.

In this scenario, the application code can use the CC3200 A2D converter to monitor battery voltage, but the A2D accuracy must be factored into any decision to activate the NWP for normal operation or not. To avoid the above cycle, use a secondary bootloader to load the user application and keep track of whether the application was loaded successfully (without causing another brownout event). This is done by keeping a counter in an on-chip register (OCR), which will likely be retained if voltage drops due to excessive power usage. The power up flow should be as described in Figure 3.

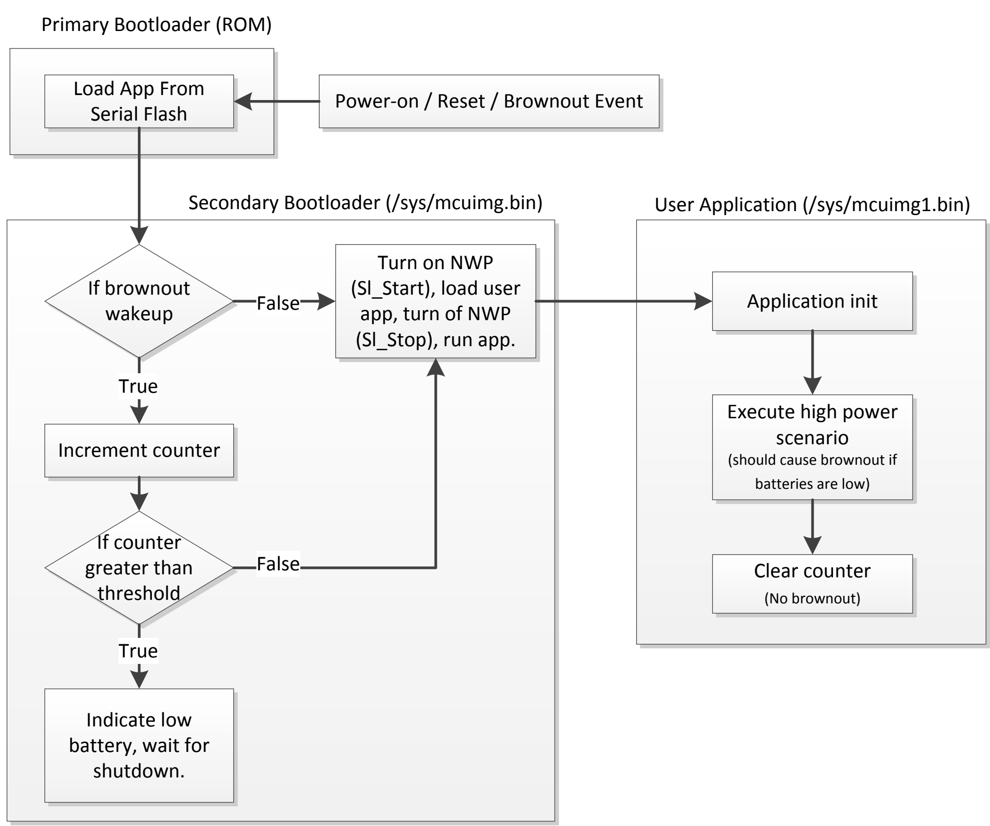


Figure 3. Brownout Suggested Flow

2 Factors to Consider When Using CC3x20 File System

2.1 Overview

The second part of this guide deals with the CC3x20 file system implemented on the serial flash. Unlike the serial flash, which can be chosen by system designers, the CC3x20 file system is common for all devices. System and configurations files can only be created and maintained using the CC3x20 file system. More detailed information on the CC3x20 file system can be found in the *CC3x20 SimpleLink Network Processor Programmer's Guide* (see the *Secure File System* section).

2.2 File System Guidelines

Most of file system guidelines can be found in *CC3x20 SimpleLink 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 5 blocks (20KB).
- The serial flash storage type that the CC3x20 supports has a minimal block size of 4096 bytes.
- Each file consumes at least:
  - 1 block (4KB) for a file with no fail-safe support
  - 2 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 upon file creation (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 make use of 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 4 illustrates the process for calculating how much memory is actually consumed on the serial flash.

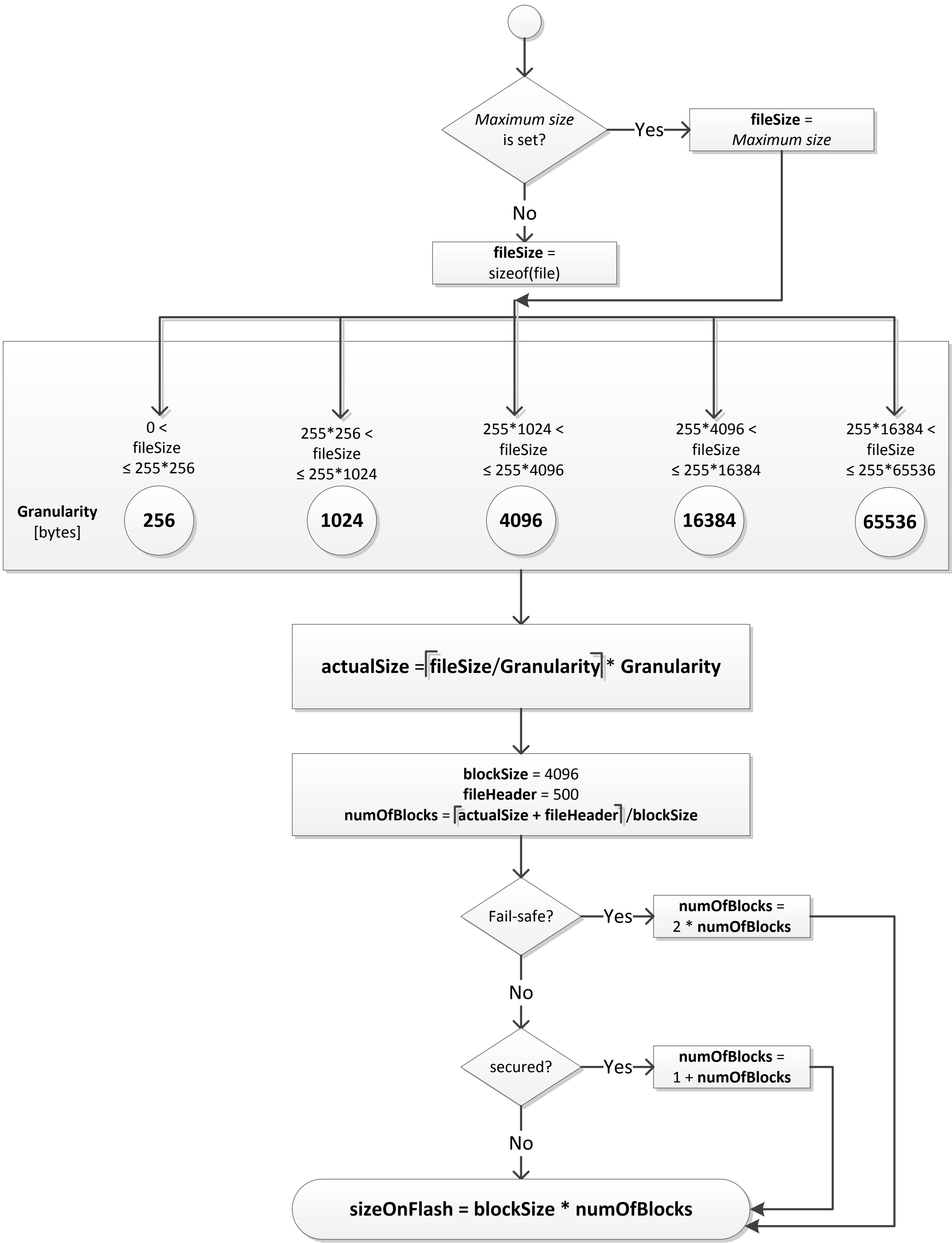


Figure 4. File Memory Consumption on Flash

2.4 System Files

2.4.1 Overview

System files include either a subset or all configuration files that set the device into a pre-defined 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 device operation, 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 sub-system, and the user does not have control over these (such as calibration files, ARP table files, and so forth).

For detailed information on the host driver API implicitly attached with the system files, refer to the *CC3x20 SimpleLink 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

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 5 is taken from the Out-of-Box project and shows the file listing.

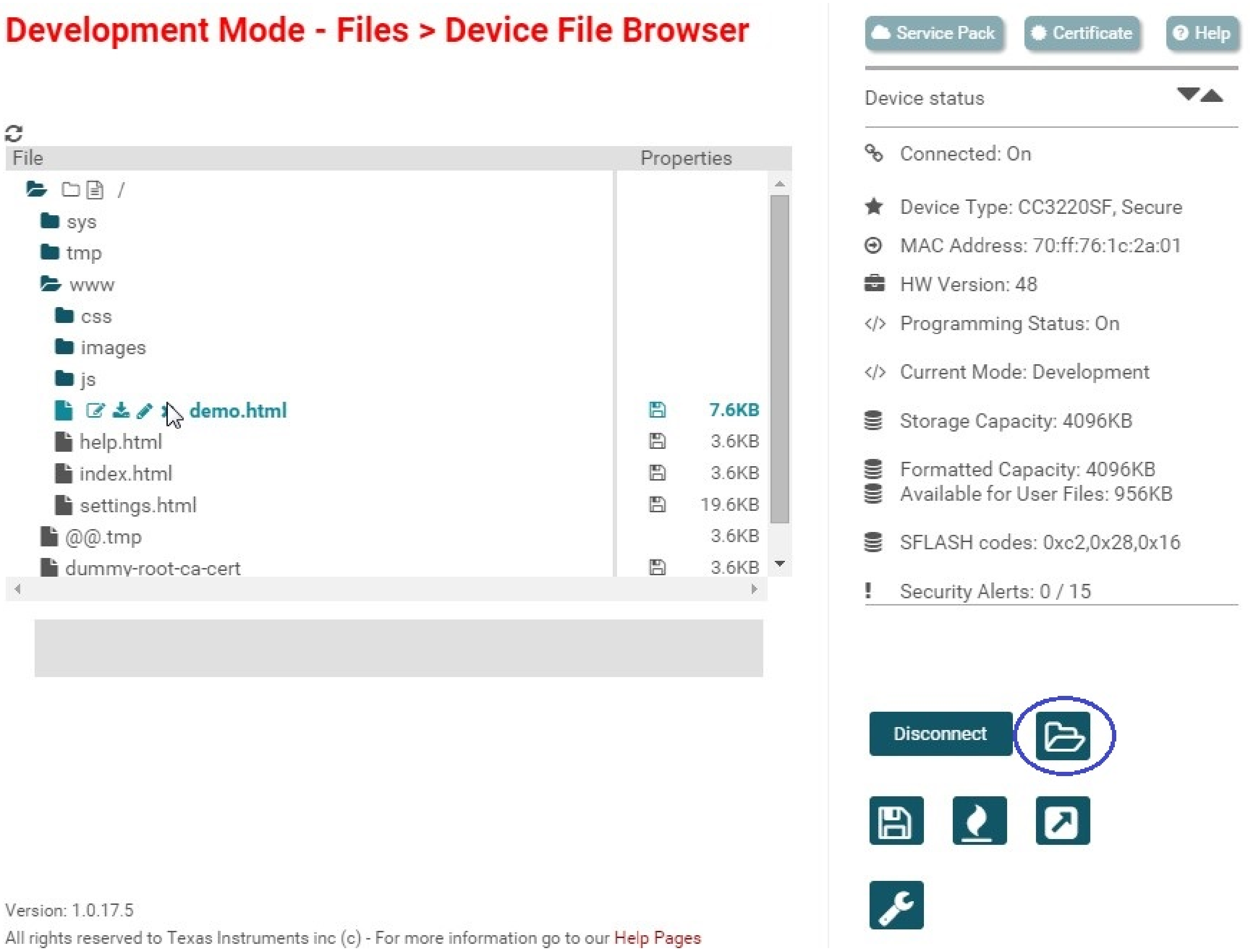


Figure 5. UniFlash File Listing

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 info command contains information about the device usage; it contains information regarding the device memory usage, information about security alerts in the system, number of times FAT has been accessed, and so forth.

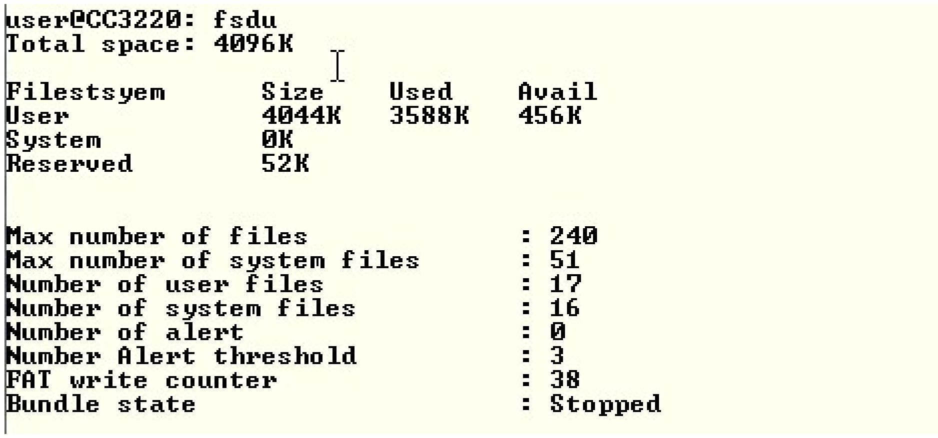
Figure 6 shows the file listing feature for the out-of-box invoked through the file system API, and printed on the terminal.

```
user@CC3220: fslist -lh
ca.der                                   3K      0x020
/sys/mcubootinfo.bin                     3K      0x2d0
/www/demo.html                           7K      0x000
/www/help.html                           3K      0x000
/www/index.html                          3K      0x000
/sys/ipcfg.ini                           3K      0x040
/www/ota.html                           11K      0x000
/sys/stacfg.ini                          3K      0x040
/sys/pref.net                            7K      0x050
/sys/ap.cfg                              3K      0x040
/sys/mdns.cfg                            3K      0x040
/sys/dhcpsrv.cfg                         3K      0x040
/sys/mode.cfg                            3K      0x040
/sys/devname.cfg                         3K      0x040
/tmp/phy.cal                            11K      0x040
/www/settings.html                      19K      0x000
/sys/certstore.lst                       7K      0x2d0
/sys/servicepack.ucf                   131K      0x2d0
/sys/mcuflashimg.bin                   179K      0x2d0
/sys/mcuflashimghash.bin                 3K      0x070
/sys/factory.img                      1347K      0x070
/sys/ucf_signatures.bin                  3K      0x060
/www/info.html                          15K      0x000
/www/css/style.css                      31K      0x000
/www/images/rotate360.jpg             1011K      0x020
/www/images/tilogo.gif                   7K      0x000
/www/images/icons/help.png               3K      0x000
/www/images/icons/menu.png               3K      0x000
/www/images/icons/wireless.png           3K      0x000
/www/images/icons/wirelessfull.          3K      0x000
/www/js/jquery.min.js                   83K      0x000
/www/js/scripts.js                       3K      0x000
/tmp/crashminidump.bin                  27K      0x020

 File properties flags description:
 0x001 - Open file commit
 0x002 - Open bundle commit
 0x004 - Pending file commit
 0x008 - Pending bundle commit
 0x010 - Secure file
 0x020 - No file safe
 0x040 - System file
 0x080 - System with user access
 0x100 - No valid copy
 0x200 - Public write
 0x400 - Public read
```

Figure 6. File Listing Through Host API

Figure 7 shows the storage info feature for the out-of-box invoked through the file system API, and printed on the terminal.



```
user@CC3220: fsdu
Total space: 4096K

Filestsyem       Size    Used    Avail
User             4044K   3588K   456K
System           0K
Reserved         52K


Max number of files               : 240
Max number of system files        : 51
Number of user files              : 17
Number of system files            : 16
Number of alert                   : 0
Number Alert threshold            : 3
FAT write counter                 : 38
Bundle state                      : Stopped
```

Figure 7. Get Storage Info 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, extra caution should be taken because not preserving enough space may cause unexpected behavior in the system.

Table 3 lists the minimal required memory consumption under the following assumptions:

- System files in use consume 64 blocks (256KB).
- Vendor files are not taken into account.
- MCU code is taken as the maximal possible size for the CC3220 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 3. Recommended Flash Size

	CC3120 [KB]	CC3220 [KB]	CC3220SF [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)	8MBit	16MBit	32MBit
Recommended flash size (2)	16MBit	16Mbit	32MBit

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

Revision History

Date	Revision	Notes
February 2017	SWRA515*	Initial release

IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated ('TI") technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

TI RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI's standard terms for semiconductor products http://www.ti.com/sc/docs/stdterms.htm), evaluation modules, and samples (http://www.ti.com/sc/docs/sampterms.htm).