

FileSystem Memory Calculator User Guide

Version 1.20

For use with **FSmem.exe** Versions 3.1 and above

Date: 26-Jun-2017 14:06

All rights reserved. This document and the associated software are the sole property of HCC Embedded. Reproduction or duplication by any means of any portion of this document without the prior written consent of HCC Embedded is expressly forbidden.

HCC Embedded reserves the right to make changes to this document and to the related software at any time and without notice. The information in this document has been carefully checked for its accuracy; however, HCC Embedded makes no warranty relating to the correctness of this document.

Table of Contents

Introduction	3
Change History	4
Using the Calculator	5
About Tab	5
RAM Driver Tab	6
NOR Flash Driver Tab	8
NAND Flash Driver Tab	11
DATAflash Driver Tab	13

1 Introduction

The FileSystem Memory Calculator is a component of the SafeFLASH file system. You can use it to calculate the memory requirements of a file system implemented in any of the following media:

- RAM driver.
- NOR flash driver.
- NAND flash driver.
- Adesto[®] DATAflash driver. (This DataFlash technology was formerly owned by Atmel[®] but is now owned by Adesto[®] Technologies.)

For each media type, you are asked to supply various parameters which define the system. After doing this, just click **Calculate** to display the space requirements. Depending on the results, you can modify the parameters as required then perform the calculation again. You can repeat this process until the results match your needs.

For a RAM driver the result shows:

- The free space and the reserved/bad space.
- The maximum number of directory entries and file handles, the number of sectors, and the sector size.

For the other drivers the result shows:

- The total memory required, the free space, and the reserved/bad space.
- The maximum number of directory entries and file handles, the number of sectors, and (for Adesto[®] DATAflash) the sector size.

The calculator is the file **src/util/FSmem.exe** in the SafeFLASH base package, **fs_safe**.

Note:

- If you need help with creating a configuration for your system, please contact support@hcc-embedded.com.
- HCC provides a wide range of fail-safe solutions for flash storage. To discuss how this product or other HCC products can be used to meet your requirements, please contact sales@hcc-embedded.com.

2 Change History

This section describes past changes to this manual.

- To view or download earlier manuals, see [Archive: FileSystem Memory Calculator User Guide](#).

The current version of this manual is 1.20. The full list of versions is as follows:

Manual version	Date	Software version	Reason for change
1.20	2017-06-26	3.1	New <i>Change History</i> format.
1.10	2015-01-09	3.1	Updated for new software.
1.00	2014-08-19	2.1	First online version.

3 Using the Calculator

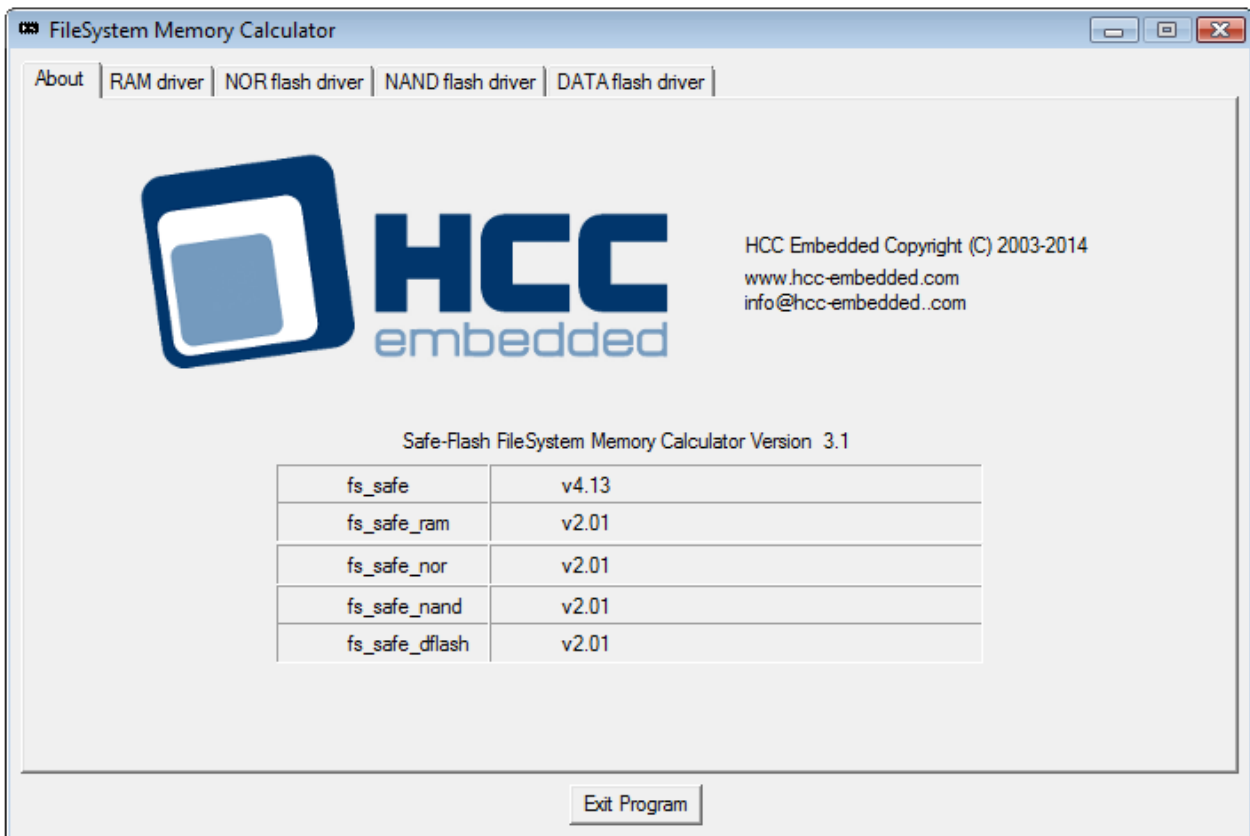
The pages which follow describe each of the calculator tabs.

Note: In the parameter lists, references to the file **config_safe.h** are to the SafeFLASH configuration file in the base **fs_safe** package.

3.1 About Tab

The **About** tab is the **FSmem.exe** home tab. It shows the calculator version and lists the supported versions of SafeFLASH and the different driver types.

Click on the tab for your media type.



Click **Exit Program** when you have finished with the calculator.

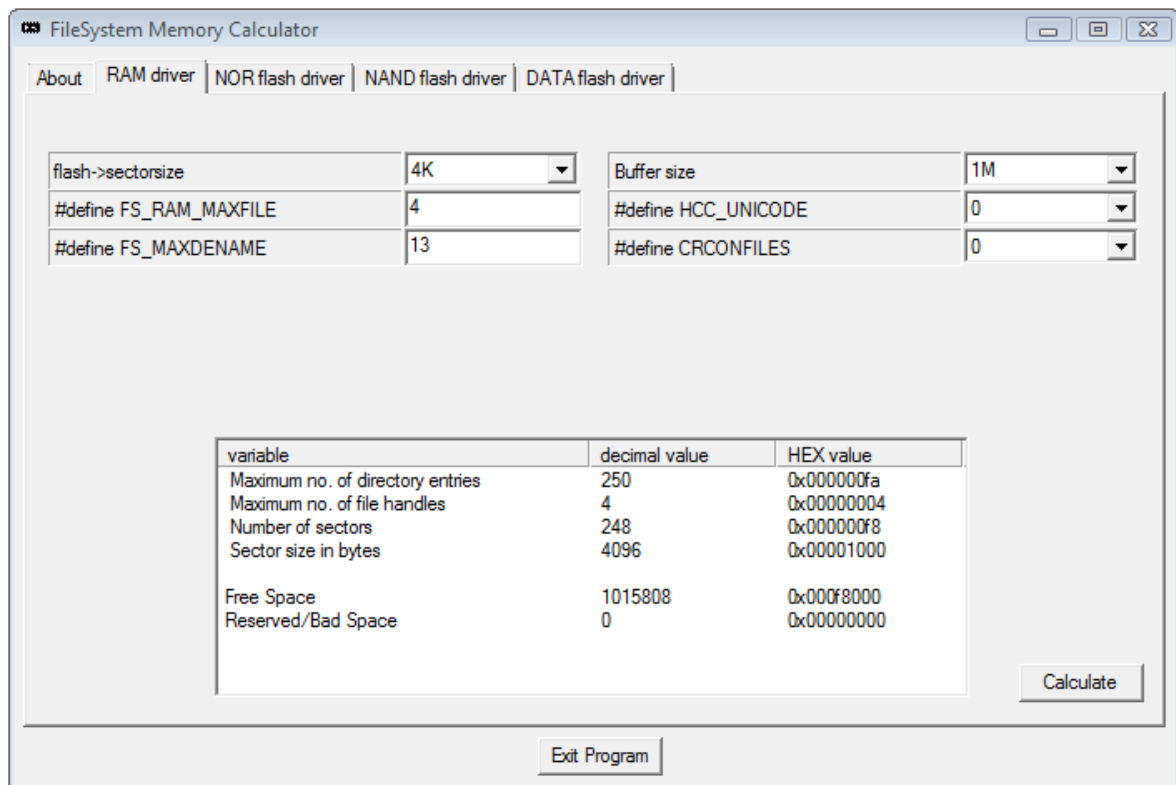
3.2 RAM Driver Tab

Use the **RAM driver** tab to calculate parameters for a RAM driver file system. The following parameters appear on this tab:

Parameter	Description
<i>flash->sectorsize</i>	The sector size for file allocation. Select a value from those listed.
<i>Buffer size</i>	The total size of the RAM drive.
<i>#define FS_RAM_MAXFILE</i>	The maximum number of files that can be open simultaneously. This option is set in the file config_safe_ram.h .
<i>#define HCC_UNICODE</i>	Select 1 to enable use of Unicode characters in file and pathnames. This option is set in the file config_safe.h .
<i>#define FS_MAXDENNAME</i>	The maximum size of a name in a base directory entry. This option is set in the file config_safe.h .
<i>#define CRCONFILES</i>	Select 1 to enable CRC protection on files. This option is set in the file config_safe.h .

Do the following:

1. Enter the parameter values.
2. Click **Calculate**. The results are displayed as shown below:



3. Make any required changes and click **Calculate** again. You can repeat this as many times as required.

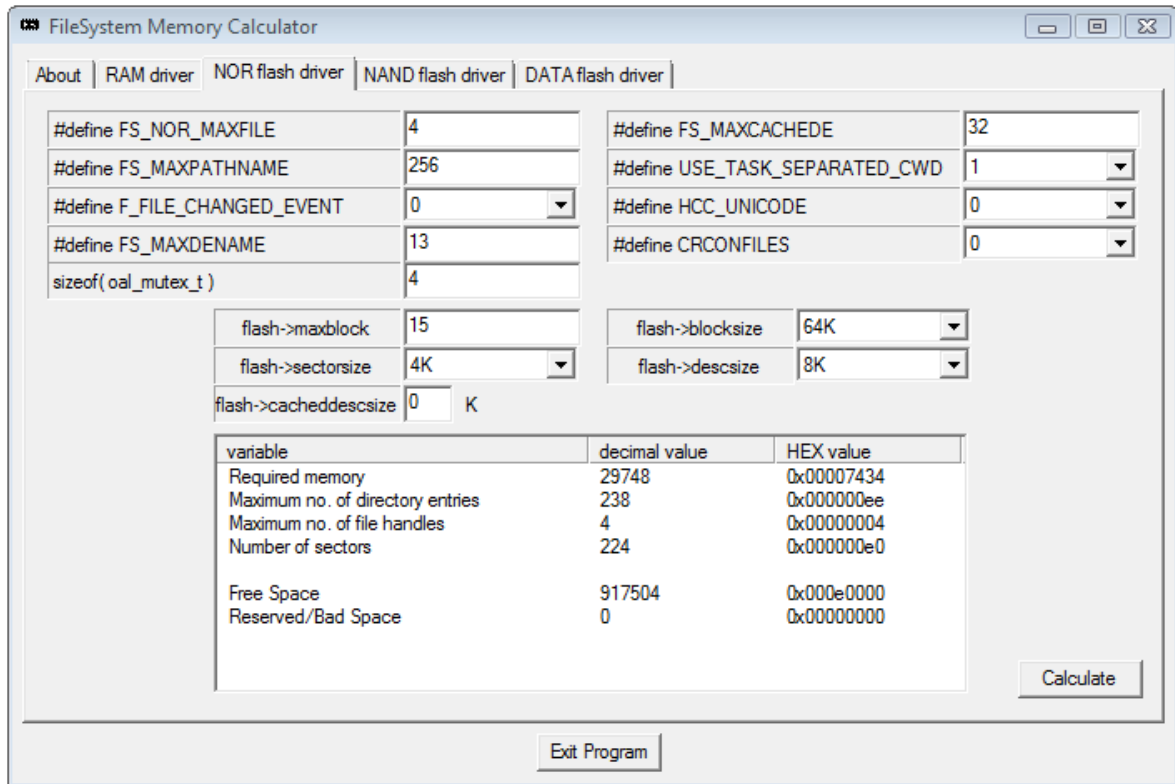
3.3 NOR Flash Driver Tab

Use the **NOR flash driver** tab to calculate parameters for a NOR flash driver file system. The following parameters appear on this tab:

Parameter	Description
<i>#define FS_NOR_MAXFILE</i>	The maximum number of files that can be open simultaneously on the volume. This option is set in the file config_safe_nor.h .
<i>#define FS_MAX_CACHEDE</i>	The maximum number of cacheable directory entries. This is defined in the file api_safe.h .
<i>#define FS_MAXPATHNAME</i>	The maximum length of a pathname. This option is set in the file config_safe.h .
<i>#define USE_TASK_SEPARATED_CWD</i>	Select 1 to give every task its own current working directory. This option is set in the file config_safe.h .
<i>#define F_FILE_CHANGED_EVENT</i>	Select 1 to call a callback function when a file state changes. This option is set in the file config_safe.h .
<i>#define HCC_UNICODE</i>	Select 1 to allow use of Unicode characters in file and path names. This option is set in the file config_safe.h .
<i>#define FS_MAXDENAME</i>	The maximum size of a name in a base directory entry. This option is set in the file config_safe.h .
<i>#define CRCONFILES</i>	Select 1 to enable CRC protection on files. This option is set in the file config_safe.h .
<i>sizeof(oal_mutex_t)</i>	The size of the <i>oal_mutex_t</i> structure in bytes, if an RTOS is used.
<i>flash>maxblock</i>	The number of erasable data blocks (not descriptor blocks) that are available for file storage.
<i>flash>blocksize</i>	The physical block size to use in the file storage area. This is an erasable unit of the flash chip. All blocks in the file storage area must be the same size.
<i>flash>sectorsize</i>	The sector size for file allocation. Each block is divided into a number of sectors. The sector size is the smallest usable unit in the system so is the minimum file storage area. To make best use of the flash blocks, the sector size must be a power of two.
<i>flash>descsize</i>	The size of a descriptor block. This is the maximum size of FAT+directory+block index. This may differ from <i>blocksize</i> .
<i>flash>cacheddescsize</i>	The descriptor write cache size inside the descriptor block. This number must be less than <i>descsize</i> , since the cache is allocated in the descriptor block. If you do not want to use the descriptor write cache, set this to zero.

Do the following:

1. Enter the parameter values.
2. Click **Calculate**. The results are displayed as shown below:



3. Make any required changes and click **Calculate** again. You can repeat this as many times as required.

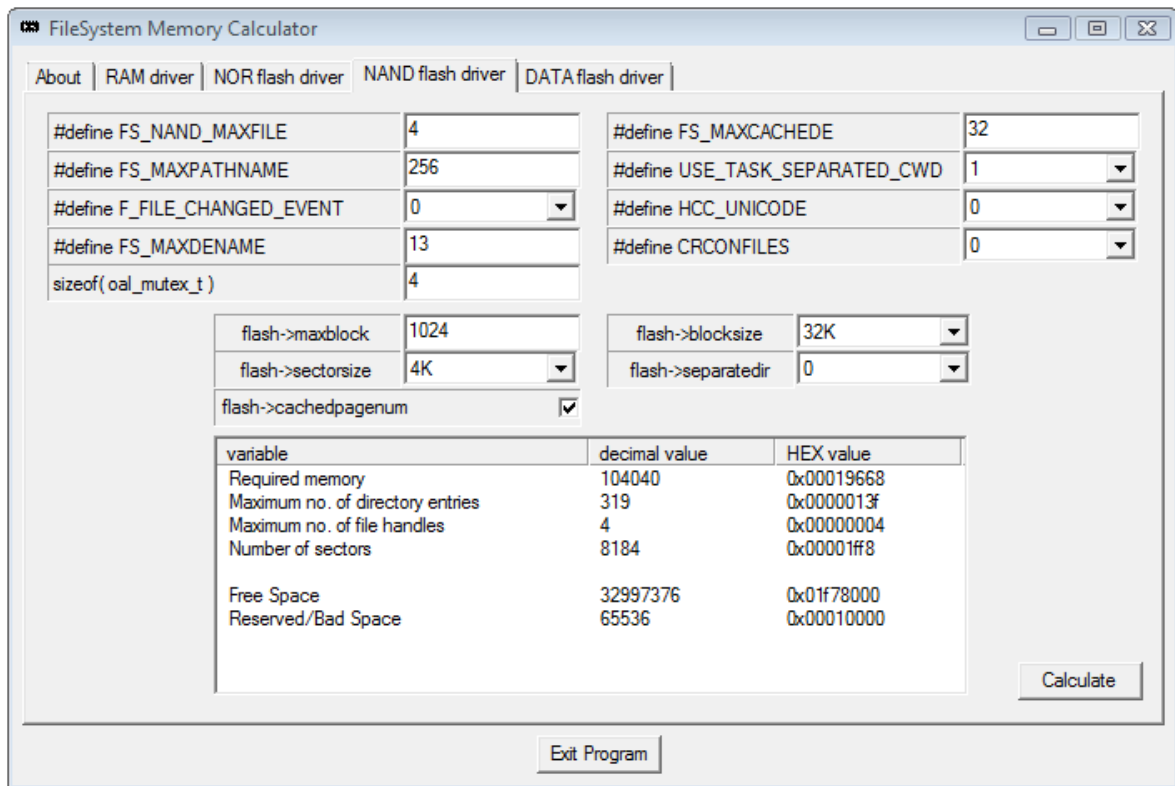
3.4 NAND Flash Driver Tab

Use the **NAND flash driver** tab to calculate parameters for a NAND flash driver file system. The following parameters appear on this tab:

Parameter	Description
<i>#define FS_NAND_MAXFILE</i>	The maximum number of files that can be open simultaneously on the volume. This option is set in the file config_safe_nand.h .
<i>#define FS_MAX_CACHEDE</i>	The maximum number of cacheable directory entries. This is defined in the file api_safe.h .
<i>#define FS_MAXPATHNAME</i>	The maximum length of a pathname. This option is set in the file config_safe.h .
<i>#define USE_TASK_SEPARATED_CWD</i>	Select 1 to give every task its own current working directory. This option is set in the file config_safe.h .
<i>#define F_FILE_CHANGED_EVENT</i>	Select 1 to call a callback function when a file state changes. This option is set in the file config_safe.h .
<i>#define HCC_UNICODE</i>	Select 1 to enable use of Unicode characters in file and path names. This option is set in the file config_safe.h .
<i>#define FS_MAXDENAME</i>	The maximum size of a name in a base directory entry. This option is set in the file config_safe.h .
<i>#define CRCONFILES</i>	Select 1 to enable CRC protection on files. This option is set in the file config_safe.h .
<i>sizeof(oal_mutex_t)</i>	The size of the <i>oal_mutex_t</i> structure in bytes, if an RTOS is used.
<i>flash>maxblock</i>	The number of data blocks (not descriptor blocks) that are available for file storage.
<i>flash>blocksize</i>	The physical block size to use in the file storage area. This is an erasable unit of the flash chip. All blocks in the file storage area must be the same size.
<i>flash>sectorsize</i>	The sector size for file allocation. Each block is divided into a number of sectors. The sector size is the smallest usable unit in the system so is the minimum file storage area. To make best use of the flash blocks, the sector size must be a power of two.
<i>flash>separatedir</i>	The number of dedicated directory blocks.
<i>flash>cachedpagenum</i>	Select this if the cache is used.

Do the following:

1. Enter the parameter values.
2. Click **Calculate**. The results are displayed as shown below:



3. Make any required changes and click **Calculate** again. You can repeat this as many times as required.

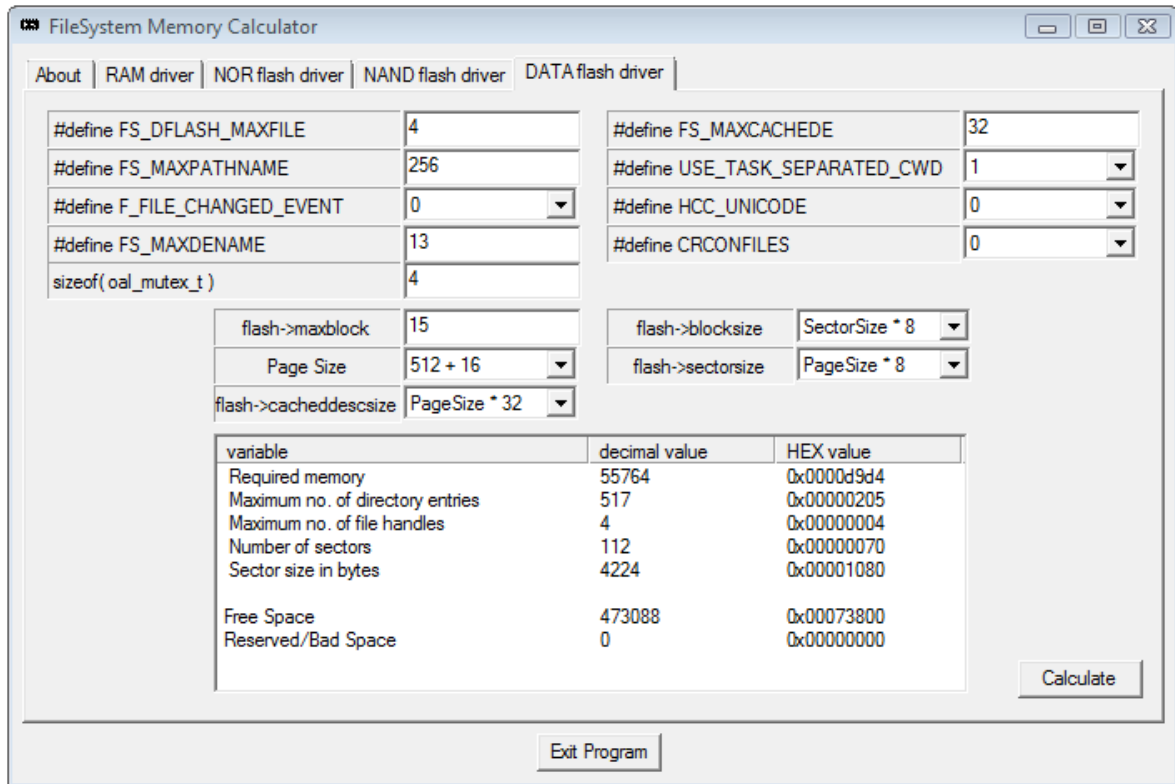
3.5 DATAflash Driver Tab

Use the **DATAflash driver** tab to calculate parameters for an Adesto[®] DATAflash driver file system. The following parameters appear on this tab:

Parameter	Description
<i>#define FS_DFLASH_MAXFILE</i>	The maximum number of files that can be opened simultaneously on the volume. This option is set in the file config_safe_dflash.h .
<i>#define FS_MAX_CACHEDE</i>	The maximum number of cacheable directory entries. This is defined in the file api_safe.h .
<i>#define FS_MAXPATHNAME</i>	The maximum length of a pathname. This option is set in the file config_safe.h .
<i>#define USE_TASK_SEPARATED_CWD</i>	Select 1 to give every task its own current working directory. This option is set in the file config_safe.h .
<i>#define F_FILE_CHANGED_EVENT</i>	Select 1 to call a callback function when a file state changes. This option is set in the file config_safe.h .
<i>#define HCC_UNICODE</i>	Select 1 to enable use of Unicode characters in file and path names. This option is set in the file config_safe.h .
<i>#define FS_MAXDENAME</i>	The maximum size of a name in a base directory entry. This option is set in the file config_safe.h .
<i>#define CRCONFILES</i>	Select 1 to enable CRC protection on files. This option is set in the file config_safe.h .
<i>sizeof(oal_mutex_t)</i>	The size of the <i>oal_mutex_t</i> structure in bytes, if an RTOS is used.
<i>flash>maxblock</i>	The number of erasable blocks (not descriptor blocks) that are available for file storage.
<i>flash>blocksize</i>	The physical block size to use in the file storage area. This is an erasable unit of the flash chip. All blocks in the file storage area must be the same size.
<i>Page size</i>	The page size.
<i>flash>sectorsize</i>	The sector size for file allocation. Each block is divided into a number of sectors. The sector size is the smallest usable unit in the system so is the minimum file storage area. To make best use of the flash blocks, the sector size must be a power of two.
<i>flash>cacheddescsize</i>	The descriptor write cache size inside the descriptor block. This number must be less than the block size. If you do not want to use the descriptor write cache, set this to zero.

Do the following:

1. Enter the parameter values.
2. Click **Calculate**. The results are displayed as shown below:



3. Make any required changes and click **Calculate** again. You can repeat this as many times as required.