Renesas Synergy™ Platform

Block Media RAM Framework Module Guide

R11AN0315EU0101
Rev.1.011
Dec 13, 2018

Introduction

This module guide will enable you to effectively use a module in your own design. Upon completion of this guide, you will be able to add this module to your own design, configure it correctly for the target application, and write code using the included application project code as a reference and efficient starting point. References to more detailed API descriptions and suggestions of other application projects that illustrate more advanced uses of the module are available on the Renesas Synergy Knowledge Base (as described in the References section at the end of this document) and should be valuable resources for creating more complex designs.

The Block Media Framework Module implements a file system on RAM for read, write, and control of the read/write region of RAM memory. The framework has all the functionality needed to interface with a file system through a block media interface.

Contents

1. Block Media RAM Framework Module Features 2

2. Block Media RAM Framework Module APIs Overview 2

3. Block Media RAM Framework Module Operational Overview 3

3.1 Block Media RAM Framework Module Important Operational Notes and Limitations 3

3.1.1 Block Media RAM Framework Module Operational Notes 3

3.1.2 Block Media RAM Framework Module Limitations 3

4. Including the Block Media RAM Framework Module in an Application 3

5. Configuring the Block Media RAM Framework Module 4

5.1 Block Media RAM Framework Module Clock Configuration 5

5.2 Block Media RAM Framework Module Pin Configuration 5

6. Using the Block Media RAM Framework Module in an Application 5

7. The Block Media RAM Framework Module Application Project 6

8. Customizing the Block Media RAM Framework Module for a Target Application 8

9. Running the Block Media RAM Framework Module Application Project 8

10. Block Media RAM Framework Module Conclusion 9

11. Block Media RAM Framework Module Next Steps 9

12. Block Media RAM Framework Module Reference Information 9

Revision History 11

1. Block Media RAM Framework Module Features

- Enables FileX® to run on linear memory-mapped devices
- Temporary and fast storage of data in RAM

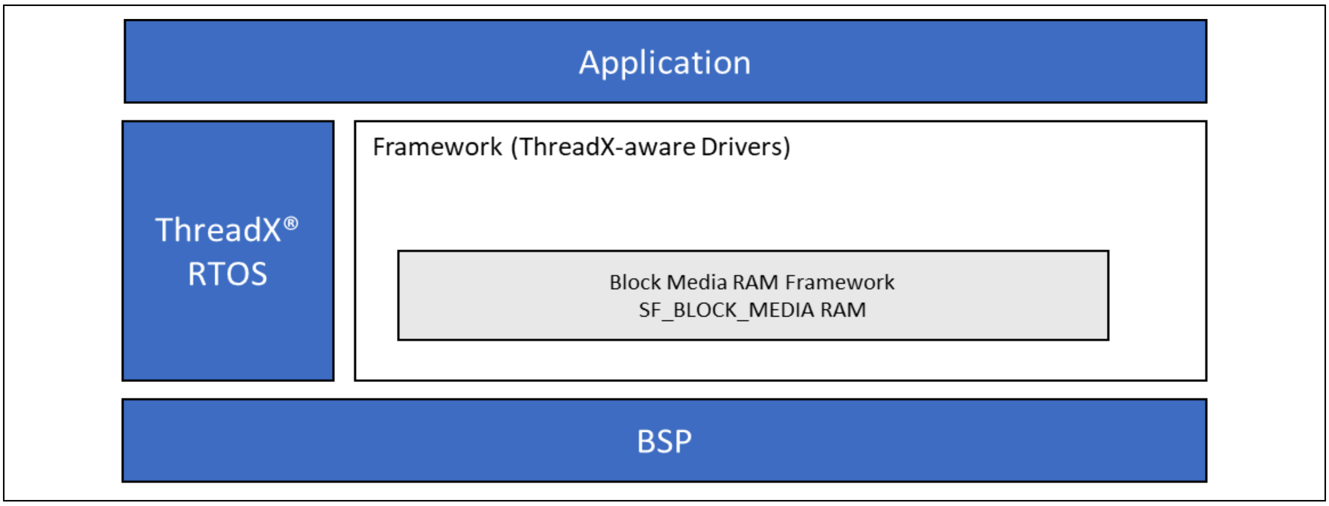


Figure 1 Block Media RAM Framework Module

2. Block Media RAM Framework Module APIs Overview

The Block Media RAM Framework module defines APIs to open, read, write, and close the module. A complete list of the available APIs, an example API call, and a short description of each can be found in the following table. A table of status return values follows the API summary table.

Table 1 Block Media RAM Framework Module API Summary

Function Name	Example API Call and Description
.open	`g_sf_block_media_ram0.p_api->open (g_sf_block_media_ram0.p_ctrl, g_sf_block_media_ram0.p_cfg);` Open device for read, write and control.
.read	`g_sf_block_media_ram0.p_api->read (g_sf_block_media_ram0.p_ctrl, p_dest, start_block, block_count);` Read data from RAM buffer.
.write	`g_sf_block_media_ram0.p_api->write (g_sf_block_media_ram0.p_ctrl, p_source, start_block, block_count);` Write data to RAM buffer.
.ioctl	`g_sf_block_media_ram0.p_api->ioctl (g_sf_block_media_ram0.p_ctrl, command, p_data);` Send control commands to and receive status of RAM buffer.
.close	`g_sf_block_media_ram0.p_api->close (g_sf_block_media_ram0.p_ctrl);` Close the framework.
.versionGet	`g_sf_block_media_ram0.p_api->versionGet(&version);` Retrieve the API version with the version pointer.

Note: For details on operation and definitions for function data structures, typedefs, defines, API data, API structures and function variables, review the *SSP User's Manual* API References for the associated module.

Table 2 Status Return Values

Name	Description
SSP_SUCCESS	API Call Successful
SSP_ERR_UNSUPPORTED	Command not supported.
SSP_ERR_NOT_OPEN	Unit is not open
SSP_ERR_ASSERTION	A parameter is NULL.
SSP_ERR_IN_USE	Framework is already open.
SSP_ERR_INVALID_BLOCKS	Invalid block passed.

Note: Lower-level drivers may return common error codes. Refer to the *SSP User's Manual* API References for the associated module for a definition of all relevant status-return values.

3. Block Media RAM Framework Module Operational Overview

The Block Media Framework Interface is simply an abstract interface using function pointers instead of direct function calls. Functions are called between FileX and the SSP block media drivers, such as SDMMC, SPI Flash and SDRAM/RAM. The interface remains the same for any media driver so that all media drivers appear functionally identical at file I/O layer and can be interchanged with one another without changing code. Device adaptation drivers, such as sf_block_media_ram, are accessed through the Block Media Framework Interface and provide device specific code needed to perform media I/O operations. Configuration and control structures passed through block media function calls are generally device specific as well.

3.1 Block Media RAM Framework Module Important Operational Notes and Limitations

3.1.1 Block Media RAM Framework Module Operational Notes

The media must be formatted before you can open the media and begin creating, writing, and reading files.

The memory area used must be directly readable and writable by the core. Internal SRAM or SDRAM size should be sufficient for the block media RAM and there will be no persistence of data across the power cycles.

Block count must be assigned by considering the boot record, FAT area, root directory and directory sector. If the File System is used on RAM then the minimum block count must be 4, as the first 3 sectors are reserved for boot record, FAT area and root directory and the 4th sector is used for data sector.

The "Format media during initialization" property needs to be enabled in order to use RAM for FileX.

The "File System is on block media" property needs to be true in order to configure the block count and block size from the block media RAM.

3.1.2 Block Media RAM Framework Module Limitations

Refer to the most recent SSP Release Notes for any additional operational limitations for this module.

4. Including the Block Media RAM Framework Module in an Application

This section describes how to include the Block Media RAM Framework module in an application using the SSP configurator.

Note: It is assumed you are familiar with creating a project, adding threads, adding a stack to a thread, and configuring a block within the stack. If you are unfamiliar with any of these items, refer to the first few chapters of the *SSP User's Manual* to learn how to manage each of these important steps in creating SSP-based applications.

To add the Block Media RAM Framework module to an application, simply add it to a HAL /Common thread using the stacks selection sequence given in the following table. (The default name for the Block Media RAM Framework module is g_sf_block_media_ram0. This name can be changed in the associated Properties window.)

Table 3 Block Media RAM Framework Module Selection Sequence

Resource	ISDE Tab	Stacks Selection Sequence
g_ sf_block_media_ram0 Block Media Framework on sf_block_media_ram	Threads	New Stack> Framework> File System> Block Media Framework on sf_block_media_ram

When the Block Media RAM Framework module on sf_block_media_ram is added to the thread stack, the configurator automatically adds any lower-level drivers that are required. Any drivers needing additional configuration information are box-text highlighted in Red. Modules with a Gray band are individual modules that stand alone. Modules with a blue band are shared or common and added only once to be used by multiple stacks. Modules with a pink band can require the selection of lower-level drivers; sometimes these are optional or recommended as text in the block indicates. If additional lower-level drivers are required, the module description includes “Add” in the text. Click any pink-banded modules to display the “New” icon and show possible choices.

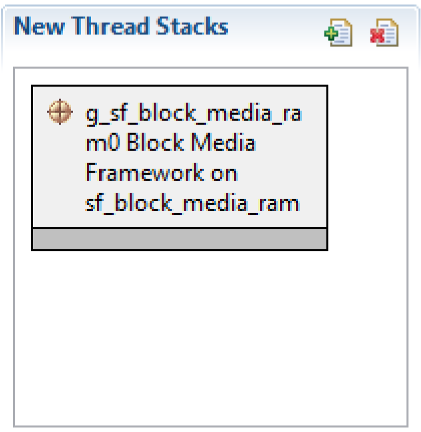

New Thread Stacks
g_sf_block_media_ram0 Block Media Framework on sf_block_media_ram

Figure 2 Block Media RAM Framework Module Stack

5. Configuring the Block Media RAM Framework Module

The user configures the Block Media RAM Framework module for the desired operation. The SSP configuration window automatically identifies (by highlighting the block in red) the selections required for successful operation, configuration selections, such as interrupts or operating modes for lower-level modules. Only non-conflicting properties are available for change. Unavailable properties are ‘locked’ with a lock icon in the ISDE Properties window. This approach simplifies the configuration process and makes it much less error-prone than previous ‘manual’ approaches. The SSP Configurator Properties tab shows available configuration settings and defaults for all the user-accessible properties. They are also listed in the following tables for easy reference.

Interrupt properties are often identified as a property requiring a change and these configuration settings are available within the Properties window of the associated module. Simply select the indicated module and then view the Properties window; the interrupt settings are toward the bottom of the properties list, so scroll down until they become available. Also, note that the interrupt priorities listed in the ISDE Properties window indicates the setting’s validity based on the targeted MCU (CM4 or CM0+). These details are not in the following tables, but are easily visible within the ISDE when configuring interrupt-priority levels.

Note: You may want to open your ISDE, create the module, and explore the property settings in parallel with reviewing the following configuration table settings. Reviewing the settings helps orient you and can be a useful ‘hands-on’ approach to learning the ins and outs of developing with the SSP.

Table 4 Configuration Settings for the Block Media RAM Framework Module on sf_block_media_ram

ISDE Property	Value	Description
Parameter Checking	BSP, Enabled, Disabled Default: Enabled	If selected code for parameter checking is included in the build.
Name	g_sf_block_media_ram0	Module name
Block size of media in bytes	512	Specify the size of the block used on the target media

ISDE Property	Value	Description
Number of blocks to allocate	16	Specify the number of blocks to allocate on the target media. The target must have sufficient storage to support the total number of bytes specified.
Enter the valid section for RAM buffer allocation	noint	Specify the section of RAM to be used as the target media.

Note: The example values and defaults are for a project using the Synergy S7G2. Other MCUs may have different default values and available configuration settings.

5.1 Block Media RAM Framework Module Clock Configuration

The Block Media RAM Framework module uses the ICLK as the source for the internal RAM.

5.2 Block Media RAM Framework Module Pin Configuration

The Block Media RAM Framework module uses internal RAM so no external pins are required.

6. Using the Block Media RAM Framework Module in an Application

The “Format media during initialization” property needs to be enabled to use RAM for FileX.

The “File System is on block media” property needs to be true in order to configure the block count and block size from the block media RAM.

The typical steps for sf_block_media_ram framework in an application are:

1. Initialize media using the `open` API
2. Read media as required using the `read` API
3. Write media as required using the `write` API
4. Operate on data as required.
5. Close media as required using the close API

The following operational flow diagram outlines these common steps:

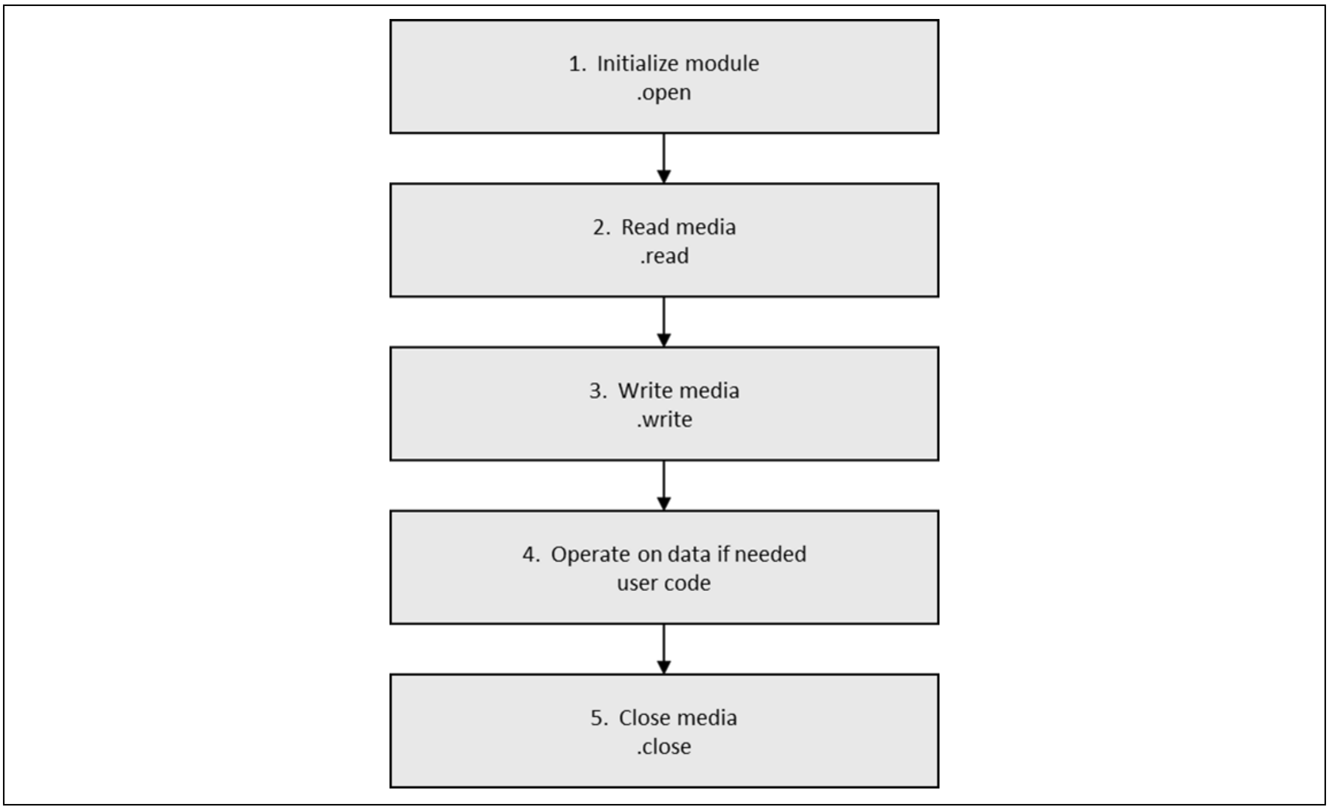


Figure 3 Typical Block Media RAM Framework Module application operational flow

The typical steps for sf_block_media_ram using sf_el_fx, in an application are:

1. Initialize media using the FileX API `fx_system_initialize` (Synergy generated code calls it automatically)
2. Format media using FileX API `fx_media_format` (Synergy generated code calls it automatically if “Format media during initialization” property is set to Enabled)
3. Open the media using FileX API `fx_media_open` (Synergy generated code opens the media automatically if “Auto Initialization” property is set to Enable)
4. Read media as required using the `read` API (Block Media Framework) or one of the FileX APIs, such as `fx_media_read()`, `fx_file_read()`.
   Write to media as required using the `write` API (Block Media Framework) or one of the FileX APIs, such as `fx_media_write`(), or `fx_file_write`().

Note: After a successful fx_media_open call, all FileX APIs can be used, not only the FileX `read` and FileX `write`.

The following typical operational flow outlines these common steps:

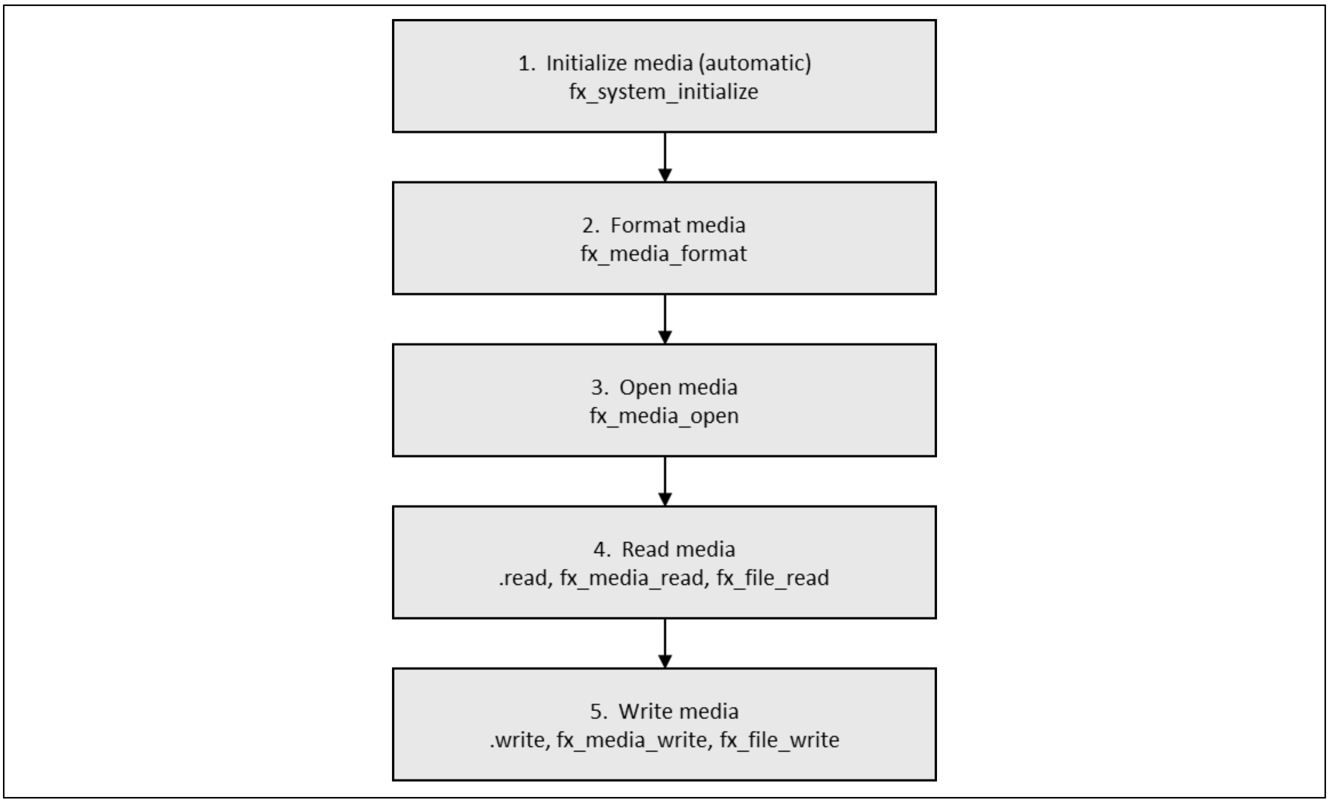


Figure 4 Typical Block Media RAM Framework Module Application for FileX operational flow

7. The Block Media RAM Framework Module Application Project

The application project associated with this module guide demonstrates typical steps in a full design. The project can be found using the link provided in the References section at the end of this document. You may want to import and open the application project within the ISDE and view the configuration settings for the FileX Port Block Media Framework Module. You can also read over the code (in `ram_thread_entry.c`), which is used to illustrate the FileX on Block Media APIs in a complete design.

The application project demonstrates the typical use of the FileX on Block Media Framework APIs. The application project auto-generated code initializes FileX framework using the FileX Port Block Media Framework. It is a driver that aims to implement FileX-specific accesses using the Block Media Framework. The underlying implementation of the Block Media Framework uses the Block Media on RAM Framework. The following table identifies the target versions for the associated software and hardware used by the application project.

Table 5 Software and Hardware Resources Used by the Application Project

Resource	Revision	Description
e^2 studio	6.2.1 or later	Integrated Solution Development Environment
SSP	1.5.0 or later	Synergy Software Platform
IAR EW for Renesas Synergy	8.23.1 or later	IAR Embedded Workbench® for Renesas Synergy™
SSC	6.2.1 or later	Synergy Standalone Configurator
SK-S7G2	v3.0 to v3.1	Starter Kit

The following flow diagram shows steps in the application project:

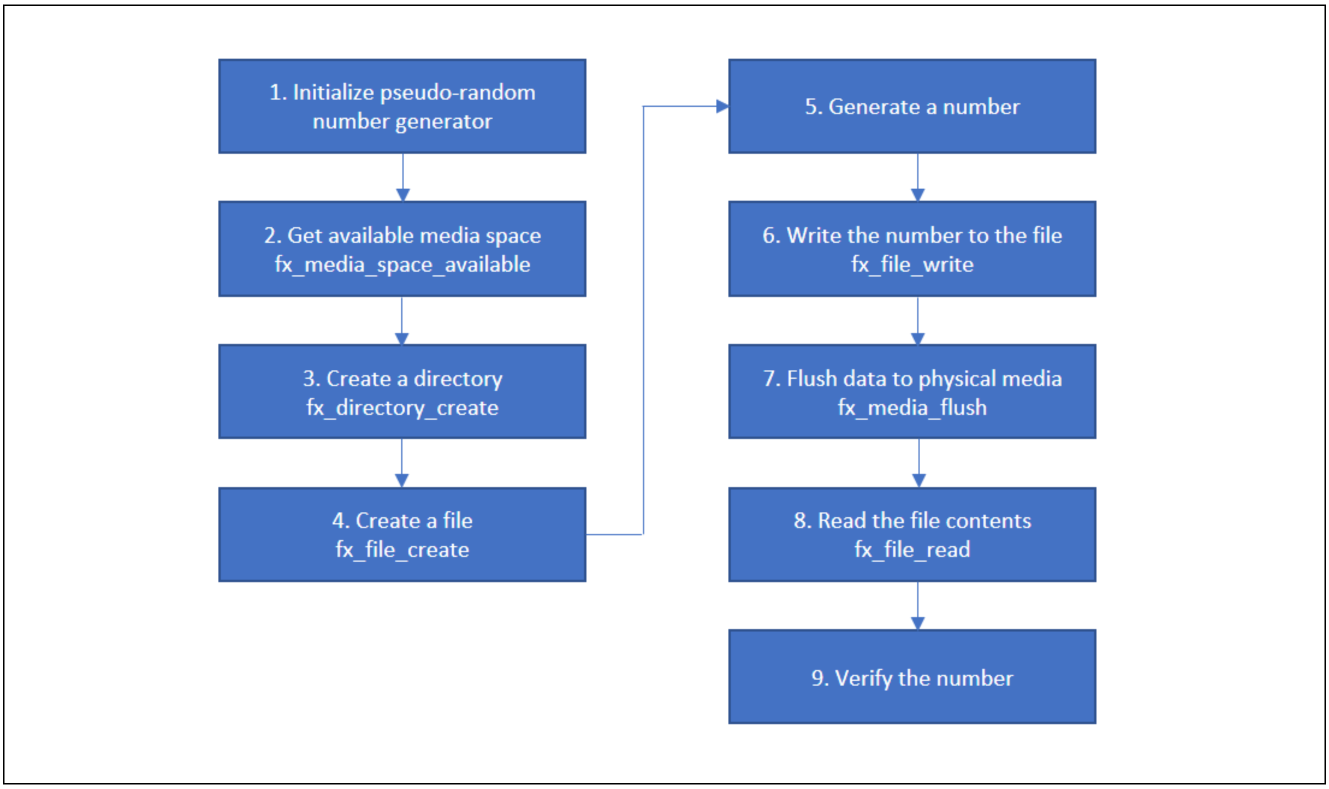

1. Initialize pseudo-random number generator
2. Get available media space fx_media_space_available
3. Create a directory fx_directory_create
4. Create a file fx_file_create
5. Generate a number
6. Write the number to the file fx_file_write
7. Flush data to physical media fx_media_flush
8. Read the file contents fx_file_read
9. Verify the number

Figure 5 Block Media RAM Framework Module application project flow diagram

The complete application project can be found using the link provided in the References section at the end of this document. The `ram_thread_entry.c` file is located in the project once it has been imported into the ISDE. You can open this file within the ISDE and follow along with the description provided to help identify key uses of APIs.

The first section of `ram_thread_entry.c` has the header files which reference the FileX Media structure and a code section which allows semi-hosting to display results using printf(). The next section contains macro constants and variable definitions. Then, there are function prototypes. The first function generates a pseudo-random number. Then, two functions for two-way conversion between integers and strings are defined. There is also simple code section for error handling. If the semi-hosting is enabled, it displays the error code. The next function is used for writing a number to a file. At first, the number is converted into a string. Then it accesses `FileX` APIs in order to open, clear, write, close the file, and flush data to physical media. The last step is necessary because the FileX operations are buffered, and not all recent changes might be reflected after reset. Alternatively, FileX flushes data when `fx_media_close` is called. The following function is very similar to the previous one, except that it is used for reading a number from given file instead of performing a write.

The last section is the thread entry function section. At the beginning, pseudo-random number generator is initialized using a predefined number. If the semi-hosting is enabled, the available space is displayed. Then, the application needs to create a file, but a directory is created first. Then the software enters the ‘forever’ while loop, and it starts with generating a number. This number is then written to the previously created file. Afterwards, the number is read and verified with the generated one. In case of an error, the message is provided. Then, a thread sleep function pauses execution for several ThreadX timer ticks and the while loop functions are repeated.

Note: This description assumes that you are familiar with using printf() the Debug Console in the Synergy Software Package. If you are unfamiliar with printf(), refer to, *How do I Use Printf() with the Debug Console in the Synergy Software Package*, listed in the Reference Section at the end of this document. Alternatively, the user can see results using the watch variables in the debug mode.

A few key properties are configured in this Application Project to support the required operations and the physical properties of the target board and MCU. Following are the properties with the values set for this specific project. You can also open the application project and view these settings in the property window as a hands-on exercise.

Table 6 Block Media Framework Module Configuration Settings for the Application Project

ISDE Property	Value Set
Name	g_fx_media
Format media during initialization	Enabled
File System in on block media	True
Formatting Options	
Volume Name	Volume 1
Number of FATs	1
Directory Entries	256
Hidden Sectors	0
Total Sectors	3751936
Bytes per Sector	512
Sectors per Cluster	1
Working media memory size	512
Name of generated initialization function	fx_media_init0
Auto Initialization	Enable

Table 7 Block Media RAM Framework Module Configuration Settings for the Application Project

ISDE Property	Value Set
Name	g_sf_block_media_ram0
Block size of media in bytes	512
Number of blocks to allocate	16
Enter the valid section for RAM buffer allocation	noinit

8. Customizing the Block Media RAM Framework Module for a Target Application

Some configuration settings are normally changed from those shown in the application project, by the developer. For example, the user can easily change the configuration settings for the FileX instance, especially the formatting options.

9. Running the Block Media RAM Framework Module Application Project

To run the Block Media RAM Framework application project and to see it executed on a target kit, you can simply import it into your ISDE, compile, and run debug.

Note: The following steps are described in sufficient detail for someone experienced with the basic flow through the Synergy development process. If these steps are not familiar, review the first few Chapters in the *SSP User's Manual*, as described in the Reference Section at the end of this document. To create and run the Block Media RAM Framework application project, simply follow these steps:

1. Import and build the example project included with this module guide according to the *Renesas Synergy Project Import Guide* (r11an0023eu0121-synergy-ssp-import-guide.pdf).
2. Connect to the host PC using the USB cable (J19 DEBUG_USB connector)
3. Start to debug the application.
   The output can be viewed in the Renesas Debug Console.

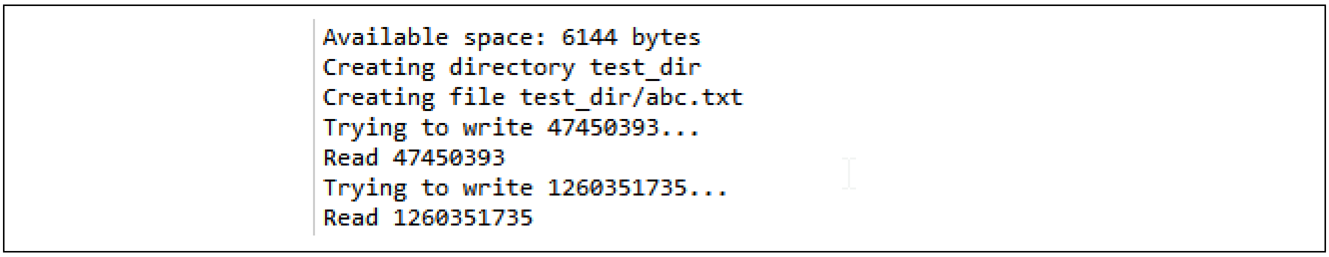
To implement the Block Media RAM Framework application in a new project, use the following steps to define, configure, auto-generate files, add code, compile, and debug the framework on the target kit. Following these steps is a hands-on approach that can help make the development process with SSP more practical, while just reading over this guide tends to be more theoretical.

Note: The following steps are described in sufficient detail for someone experienced with the basic flow through the Synergy development process. If these steps are not familiar, refer to the first few Chapters in the *SSP User's Manual*, as described in the Reference Section at the end of this document.

To create and run the FileX Port Block Media Framework application project, simply follow these steps:

1. Create a new Renesas Synergy project for the SK-S7G2 board called **RAM_HAL_FRWK_FX_MG_AP**.
2. Select the **Threads** tab.
3. Add a new thread as follows:
   - Symbol **ram_thread**
   - Name **RAM Thread**
4. Add **g_fx_media FileX** on Block Media to the RAM Thread.
5. Add **g_sf_block_media_ram0** Block Media Driver as Block Media Framework on sf_block_media_ram.
6. Copy the files **`ram_thread_entry.c`** and **`semihosting_cfg.h`** to the src folder. (Refer the package for the sample project).
7. Connect to the host PC using the USB cable (J19 DEBUG_USB connector)
8. Start to debug the application.

   The output can be viewed in the Renesas Debug Console.



```
Available space: 6144 bytes
Creating directory test_dir
Creating file test_dir/abc.txt
Trying to write 47450393...
Read 47450393
Trying to write 1260351735...
Read 1260351735
```

Figure 6 Example Output from Block Media RAM Framework Module application project

10. Block Media RAM Framework Module Conclusion

This module guide has provided all the background information needed to select, add, configure, and use the module in an example project. Many of these steps were time consuming and error-prone activities in previous generations of embedded systems. The Renesas Synergy Platform makes these steps much less time consuming and removes the common errors like conflicting configuration settings or the incorrect selection of lower-level modules. The use of high-level APIs (as demonstrated in the application project) illustrates additional development time savings by allowing work to begin at a high level and avoiding the time required in older development environments to use, or, in some cases, create, lower-level drivers.

11. Block Media RAM Framework Module Next Steps

After you have mastered a simple FileX Block Media RAM Framework project, you may want to review a more complex example. You may find that the USBX Device Class Mass Storage is a better fit for your target application. The USBX Device Class Mass Storage Module Guide illustrates the use of the mass storage device and can be easily adopted to work with FileX. This guide is available at the link shown in the References section at the end of this document.

12. Block Media RAM Framework Module Reference Information

SSP User Manual: Available in html format in the SSP distribution package and as a pdf from the Synergy Gallery.

Links to all the most up-to-date Block Media RAM Framework module reference materials and resources are available on the Synergy Knowledge Base: https://en-support.renesas.com/search/sf_block_media_ram%20Module%20Guide%20Resources

Website and Support

Visit the following vanity URLs to learn about key elements of the Synergy Platform, download components and related documentation, and get support.

Synergy Software	www.renesas.com/synergy/software
Synergy Software Package	www.renesas.com/synergy/ssp
Software add-ons	www.renesas.com/synergy/addons
Software glossary	www.renesas.com/synergy/softwareglossary
Development tools	www.renesas.com/synergy/tools
Synergy Hardware	www.renesas.com/synergy/hardware
Microcontrollers	www.renesas.com/synergy/mcus
MCU glossary	www.renesas.com/synergy/mcuglossary
Parametric search	www.renesas.com/synergy/parametric
Kits	www.renesas.com/synergy/kits
Synergy Solutions Gallery	www.renesas.com/synergy/solutionsgallery
Partner projects	www.renesas.com/synergy/partnerprojects
Application projects	www.renesas.com/synergy/applicationprojects
Self-service support resources:	
Documentation	www.renesas.com/synergy/docs
Knowledgebase	www.renesas.com/synergy/knowledgebase
Forums	www.renesas.com/synergy/forum
Training	www.renesas.com/synergy/training
Videos	www.renesas.com/synergy/videos
Chat and web ticket	www.renesas.com/synergy/resourcelibrary

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Jun 11, 2018	—	Initial Release
1.01	Dec 13, 2018	7–9	Updated to SSP v1.5.0

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
6. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

RENESAS

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A.
Tel: +1-408-432-8888, Fax: +1-408-434-5351

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700, Fax: +44-1628-651-804

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709 Quantum Plaza, No.27 ZhichunLu, Haidian District, Beijing, 100191 P. R. China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, 200333 P. R. China
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HAL 2nd Stage, Indiranagar, Bangalore 560 038, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
17F, KAMCO Yangjae Tower, 262, Gangnam-daero, Gangnam-gu, Seoul, 06265 Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5338

Colophon 7.0