Renesas Synergy™ Platform

USBX™ Device Class Mass Storage Module Guide

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 in 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 USBX™ Device Class Mass Storage module is a high-level API for USB mass storage applications and is implemented on `sf_el_ux` for USB Full Speed (USBFS) or USB High Speed (USBHS). The USBX Device Class Mass Storage module uses the USB and data-transfer peripherals on the Synergy MCU.

Contents

1. USBX Device Class Mass Storage Module Features

- ThreadX®-aware framework.
- Storage Media Parameter Setup
  - Last LBA
  - Byte-per-sector
  - Type of storage media
  - Removable flag
- USB Device Configuration (Device Configuration)
  - Vendor ID
  - Product ID
  - Device Release Number
  - Index of Serial Number String Descriptor
- Supported USB Specification (DCD)
  - USBFS
  - USBHS
- USB Device interrupts (DCD)

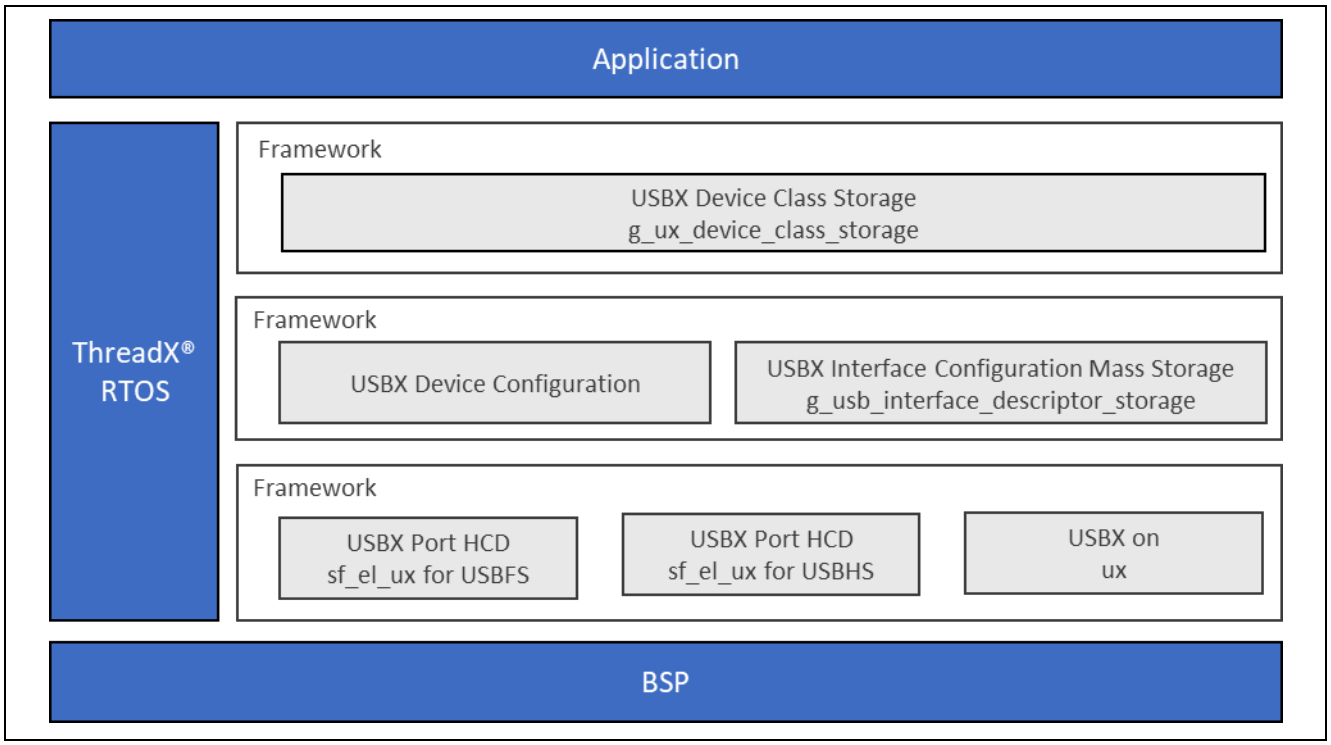

Figure 1. USBX Device Class Mass Storage Module Organization, Options, and Stack Implementations

2. USBX Device Class Mass Storage APIs Overview

The USBX Device Class Mass Storage module automatically adds an initialization process; the user application only needs to prepare the callback functions for media access. Unless the functionality of the USBX Device Class Mass Storage module is required, there is no need to use it.

Note: For details on the USBX Device Stack, see *USBX Device Stack User's Manual*.

3. USBX Device Class Mass Storage Module Operation

The USBX Device Class Mass Storage module automatically adds an initialization process. The process initializes internal information with the given parameters and creates an internal thread for processing the mass-storage class; this internal thread processes all USB messages.

On requests from the connected host side, the user application's callback functions enable access to the storage media from this internal thread.

3.1 USBX Device Class Mass Storage Module Operational Notes and Limitations

3.1.1 USBX Device Class Mass Storage Module Operational Notes

- The USBX device storage class supports multiple logical unit numbers (LUNs), making it possible to create a storage device that acts simultaneously as a CD-ROM and flash disk.

3.1.2 USBX Device Class Mass Storage Module Limitations

- This module does not support the complex device.
- See the latest *SSP Release Notes* for any other operational limitations that apply to this module.

4. Including the USBX Device Class Mass Storage Module in an Application

This section describes how to include the USBX Device Class Mass Storage 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 need to learn how to perform these any of these tasks in creating SSP-based applications, see the *SSP User's Manual*.

To add the USBX Device Class Mass Storage module to an application, simply add it to a thread using the stacks selection sequence given in the following table. (The default name for the USBX Device Class Mass Storage module is `g_ux_device_class_storage`. This name can be changed in the associated Properties window.)

Table 1. USBX Device Class Mass Storage Selection Sequence

Resource	ISDE Tab	Stacks Selection Sequence
`g_ux_device_class_storage` USBX Device Class Mass Storage	Threads	New Stack> X-Ware> USBX> Device> Classes> Mass Storage> USBX Device Class Mass Storage

The following figure shows that when the USBX Device Class Mass Storage module is added to the thread stack, the configurator automatically adds the needed lower-level drivers. 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 need only be added once, since they can be used by multiple stacks. Modules with a Pink band can require the selection of lower-level drivers. Sometimes these are optional or recommended and this is indicated in the block with the inclusion of this text. If the addition of lower-level drivers is required, the module description shows **Add** in the text. Clicking on any Pink banded modules brings up the **New** icon and shows possible choices.

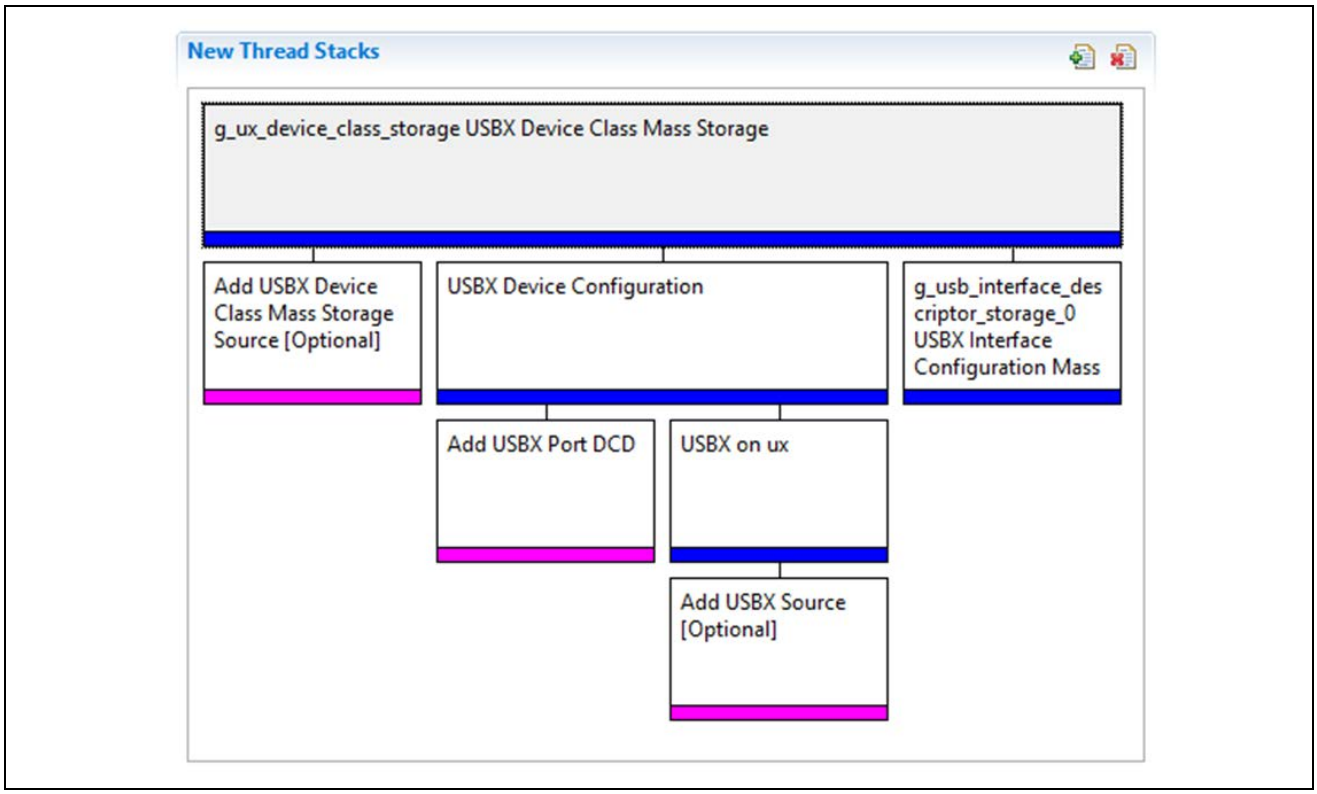

Figure 2. USBX Device Class Mass Storage Module Stack

5. Configuring the USBX Device Class Mass Storage Module

The USBX Device Class Mass Storage module must be configured for the desired operation. The SSP configuration window automatically identifies (by highlighting the block in red) any required configuration selections, such as interrupts or operating modes, which must be configured for lower-level modules for successful operation. Only those properties that can be changed without causing conflicts are available for modification. Other properties are locked and are not available for changes and are identified with a lock icon for the locked property in the Properties window in the Integrated Solution Development Environment (ISDE). This approach simplifies the configuration process and makes it much less error prone than previous manual approaches to configuration. The available configuration settings and defaults for all the user-accessible properties are given in the properties tab within the SSP configurator and are shown in the following tables for easy reference.

One of the properties most often identified as requiring a change is the interrupt priority; this configuration setting is available within the Properties window of the associated module. Simply select the indicated module and then view the Properties window; the interrupt settings are often toward the bottom of the properties list, so scroll down until they become available. Also note that the interrupt priorities listed in the Properties window includes an indication as to the validity of the setting based on the targeted MCU (CM4 or CM0+). This level of detail is not included in the following configuration properties tables, yet easily visible when configuring interrupt-priority levels.

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

Table 2. Configuration Settings for the USBX Device Class Mass Storage Module

ISDE Property	Value	Description
Name	g_ux_device_class_storage	Module name.
Mass Storage Class Parameter Setup	Auto (Simple Auto Setup if LUN is 1), Manual (User Manual Setup if LUN is greater than 1) (Default: Auto)	Mass storage class parameter setup selection.
User Setup Callback (Only valid if Parameter Setup is Auto)	ux_device_class_storage_user_setup	User setup callback selection.
Last LBA of Storage Media (Only valid if Parameter Setup is Auto)	0	Last LBA of Storage Media selection
Bytes Per Sector of Storage Media (Only valid if Parameter Setup is Auto)	512	Bytes Per Sector of Storage Media selection
Type of Storage Media (Only valid if Parameter Setup is Auto)	0	Type of Storage Media selection
Removable Flag of Storage Media (Only valid if Parameter Setup is Auto)	0x80	Removable Flag of Storage Media selection
Media Read Function Callback (Only valid if Parameter Setup is Auto)	ux_device_msc_media_read	Media Read Function Callback selection
Media Write Function Callback (Only valid if Parameter Setup is Auto)	ux_device_msc_media_write	Media Write Function Callback selection
Media Status Function Callback (Only valid if Parameter Setup is Auto)	ux_device_msc_media_status	Media Status Function Callback selection

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

5.1 Configuration Settings for the USBX Device Class Mass Storage Module Low-Level Modules

Typically, only a small number of settings must be modified from the default for lower-level modules. (These are indicated via the red text in the thread stack block.) Notice that some of the configuration properties must be set to a certain value for proper framework operation and are locked to prevent user modification. The following tables identify all the values within the properties section for the module.

Table 3. USBX Device Configuration Settings

ISDE Property	Value	Description
Vendor ID	0x045B	Vendor ID selection
Product ID	0x0000	Product ID selection
Device Release Number	0x0000	Device Release Number selection
Index of Manufacturing String Descriptor	0x00	Index of Manufacturing String Descriptor selection
Index of Product String Descriptor	0x00	Index of Product String Descriptor selection
Index of Serial Number String Descriptor	0x00	Index of Serial Number String Descriptor selection
Class Code	Communications (CDC), HID, Mass Storage, Miscellaneous, Vendor specific (Default: Communications (CDC))	Class Code selection
Index of String Descriptor describing this configuration	0x00	Index of String Descriptor describing this configuration selection
Size of USB Descriptor in bytes for	0x00	Size of USB Descriptor in bytes

ISDE Property	Value	Description
this configuration (Modify this value only for Vendor-specific Class, otherwise set zero)		for this configuration selection
Number of Interfaces (Modify this value only for Vendor-specific Class, otherwise set zero)	0x00	Number of Interfaces selection
Self-Powered	Enable, Disable Default: Enable	Self-Powered selection
Remote Wakeup	Enable, Disable Default: Disable	Remote Wakeup selection
Maximum Power Consumption (in 2 mA units)	50	Maximum Power Consumption selection
Supported Language Code	0x0409	Supported Language Code selection
Name of USBX String Framework	NULL	Name of USBX String Framework selection
Total index number of USB String Descriptors in USB String Framework	0	Total index number of USB String Descriptors in USB String Framework selection
Name of USBX Language Framework	NULL	Name of USBX Language Framework selection
Number of Languages to support (US English is applied if zero is set)	0	Number of Languages to support selection

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.

Table 4. USBX Interface Configuration Mass Storage Class Settings

ISDE Property	Value	Description
Name	`g_usb_interface_descriptor_storage_0`	Module name
Interface Number of Communications Class Interface	0x00	Interface Number of Communications Class Interface selection
Endpoint Number to be used for Bulk Out Transfer	Endpoint 1-9 Default: Endpoint 1	Endpoint Number to be used for Bulk Out Transfer selection
Endpoint Number to be used for Bulk In Transfer	Endpoint 1-9 Default: Endpoint 2	Endpoint Number to be used for Bulk In Transfer selection

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

Table 5. USBX DCD on sf_el_ux for USBFS Settings

ISDE Property	Value	Description
Full Speed Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) (Default: Disabled)	Full speed interrupt priority selection.
Name	g_sf_el_ux_dcd_fs_0	Module name.
USB Controller Selection	USBFS	USB controller selection.

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

Table 6. USBX DCD on sf_el_ux for USBHS Settings

ISDE Property	Value	Description
High Speed Interrupt Priority	Priority 0 (highest), Priority 1:2, Priority 3 (CM4: valid, CM0+: lowest- not valid if using ThreadX), Priority 4:14 (CM4: valid, CM0+: invalid), Priority 15 (CM4 lowest - not valid if using ThreadX, CM0+: invalid) (Default: Disabled)	High speed interrupt priority selection.
Name	`g_sf_el_ux_dcd_hs_0`	Module name.
USB Controller Selection	USBHS	USB controller selection.

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

Table 7. USBX on ux settings

ISDE Property	Value	Description
USBX Pool Memory Name	`g_ux_pool_memory`	USBX pool memory name selection.
USBX Pool Memory Size	18432	USBX pool memory size selection.
User Callback for Host Event Notification (Only valid for USB Host)	NULL	User callback for host event notification (only valid for USB host)

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

5.2 USBX Device Class Mass Storage Module Clock Configuration

The USB peripheral module uses UCLK as its clock source. UCLK should be configured for 48 MHz operation. In the **SSP Configuration Window**, select the **Clocks** tab to view the clock-source setting.

5.3 USBX Device Class Mass Storage Module Pin Configuration

The USB peripheral module uses MCU pins to communicate with external devices. Select **I/O pins** and configure to the external device requirements. The following table lists the pin selection method within the **SSP Configuration Window**, with the subsequent table demonstrating the selection process using USB pins as an example.

Note: For some peripherals, the operation mode selection determines what peripheral signals are available and the MCU pins required.

Table 8. USBFS and USBHS Pin Selection Sequence

Resource	ISDE Tab	Pin Selection Sequence
USBFS	Pins	Select Peripherals > Connectivity: USBFS> USBFS0
USBHS	Pins	Select Peripherals > Connectivity: USBHS> USBHS0

Note: The selection sequence assumes USBFS0 or USBHS0 is the desired hardware target for the driver.

Table 9. USBFS Pin Configuration Settings

Property	Value	Description
Operation Mode	Disabled, Custom, Device, Host, OTG (Default: Custom)	Select device as the Operation Mode
USBDP	USBDP	USBDP pin
USBDM	USBDM	USBDM pin
OVRCURB	None	OVRCURB pin
OVRCURA	None	OVRCURA pin
VBUSEN	None	VBUSEN pin

Property	Value	Description
VBUS	None, P407 (Default: P407)	VBUS pin
EXICEN	None	EXICEN pin
ID	None	ID Pin
VCCUSB	VCCUSB	VCCUSB pin
VSSUSB	VSSUSB	VSSUSB pin

Table 10. USBHS Pin Configuration Settings

Property	Value	Description
Operation Mode	Disabled, Custom, Device, Host, OTG (Default: Custom)	Select Device as the Operation Mode
USBHSDP	USBHSDP	USBHSDP pin
USBHSDM	USBHSDM	USBHSDM pin
OVRCURB	None	OVRCURB pin
OVRCURA	None	OVRCURA pin
VBUSEN	PB00	VBUSEN pin
VBUS	PB01	VBUS pin
EXICEN	None	EXICEN pin
ID	None	ID pin
USBHSRREF	USBHSRREF	USBHSRREF pin
AVCCUSBHS	AVCCUSBHS	AVCCUSBHS pin
AVSSUSBHS	AVSSUSBHS	AVSSUSBHS pin
PVSSUSBHS	PVSSUSBHS	PVSSUSBHS pin
VCCUSBHS	VCCUSBHS	VCCUSBHS pin
VSS1USBHS	VSS1USBHS	VSS1USBHS pin
VSS2USBHS	VSS2USBHS	VSS2USBHS pin

Note: The example values are for a project using the S7G2 Synergy MCU Group and the SK-S7G2 Kit. Other Synergy MCUs and other Synergy Kits may have different available pin configurations.

6. Using the USBX Device Class Mass Storage Module in an Application

The USBX Device Class Mass Storage module does not need the usual initialization by an application; the application simply prepares the three user callbacks that the USBX Device Class Mass Storage module requires.

7. Using the USBX Device Class Mass Storage Module in an Application

The application project associated with this module guide demonstrates a full design. You may want to import and open the application project within the ISDE and view configuration settings for the USBX Device Class Mass Storage module. You can also read over the code in `ux_user_callback.c`, demonstrating the USBX Device Class Mass Storage APIs in a complete design.

In the application project, the program accesses the RAMDISK that uses a memory area; this memory is defined into `ramdisk_image.c`. Used as a media for the application, `ramdisk_image.c` functions as a USB memory. Connect the media to a PC with a USB cable to check the media description.

Table 11. Application Project Applicable Software and Hardware Resources

Resource	Revision	Description
e² studio	V7.3.0 or later	Integrated Solution Development Environment
SSP	v1.6.0 or later	Renesas Synergy™ Software Platform
IAR EW for Synergy	v8.23.3 or later	IAR Embedded Workbench® for Renesas Synergy™
SSC	V7.3.0 or later	Synergy Standalone Configurator
SK-S7G2	v3.0, v3.3	Starter Kit
DK-S7G2	v3.0, v3.1	Developer Kit

Resource	Revision	Description
DK-S3A7	v2.0	Developer Kit
DK-S124	v2.0, v3.0	Developer Kit
PK-S5D9	v1.0	Promotional Kit
PE-HMI1	v2.0	Product Example Kit

The following flow diagram shows a simple application project:

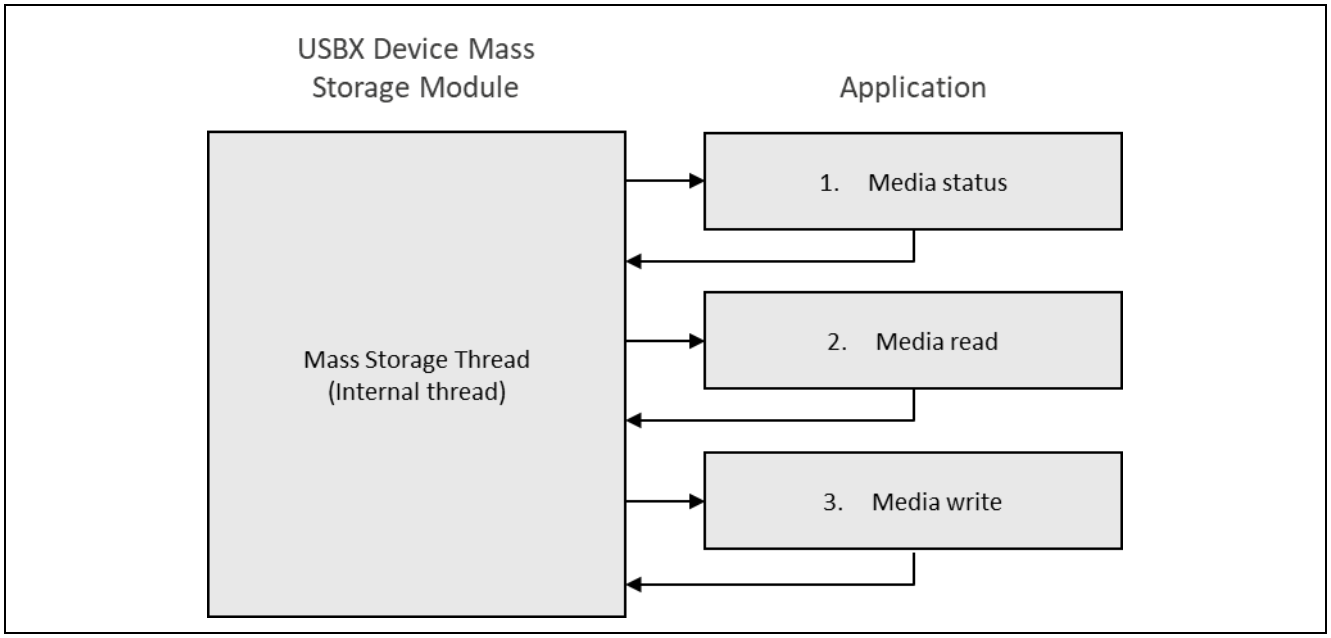

Figure 3. USBX Device Class Mass Storage Module Application Project Flow Diagram

Once the project has been imported into the ISDE, the `ux_user_callback.c/ramdisk_image.c` files are located in the project. You can open these files within the ISDE. The following guide helps you to create callback functions.

In `ux_user_callback.c`, the first section is `ux_device_msc_media_status()`; this function sets the media status. If there is a problem in the media access, set the error information to `media_status` for an argument.

The next section is `ux_device_msc_media_read()`; this function reads the description of media by reading the media data specified by the LBA from the `data_pointer` as well as the block number specified.

The last section is `ux_device_msc_media_write()`; this function writes the description to the media by writing the media data specified by the LBA to the `data_pointer` as well as the block number specified.

The return value of these functions can be either `UX_SUCCESS` or `UX_ERROR`, indicating a successful or unsuccessful operation; these operations do not need to return any other error codes. If there is an error in any operation, the storage class invokes the status-callback function.

Note: For USBX Device Stack callback functions and error codes, see the *USBX Device Stack User's Manual*.

An initializing process is prepared by the USBX Device Class Mass Storage module; it is not necessary to prepare an initializing process by the user application.

A few key properties are configured in this application project to support required operations and the physical properties of the target board and MCU. The following table lists the properties with values set to this specific project. You can also open the application project and view these settings in the **Properties** window as a hands-on exercise.

Table 12. USBX Device Class Mass Storage Module Configuration Settings for the Application Project

ISDE Property	Value Set
Name	`g_ux_device_class_storage`
Mass Storage Class Parameter Setup	Auto (Simple Auto Setup if LUN is 1)
User Setup Callback (Only valid if Parameter Setop is Auto)	`ux_device_class_storage_user_setup`
Last LBA of Storage Media (Only valid if Parameter Setup is Auto)	31
Bytes Per Sector of Storage Media (Only valid if Parameter Setup is Auto)	512
Type of Storage Media (Only valid if Parameter Setup is Auto)	0
Removable Flag of Storage (Only valid if Parameter Setup is Auto)	0x80
Media Read Function Callback (Only valid if Parameter Setup is Auto)	`ux_device_msc_media_read`
Media Write Function Callback (Only valid if Parameter Setup is Auto)	`ux_device_msc_media_write`
Media Status Function Callback (Only valid if Parameter Setup is Auto)	`ux_device_msc_media_status`

8. Customizing the USBX Device Class Mass Storage Module for a Target Application

Some configuration settings are normally changed by the developer from those shown in the application project. For example, you can change the Last LBA and the Byte Per Sector. The Last LBA and the Byte Per Sector can be configured for application supported storage media information. Running the USBX Device Class Mass Storage Module Application Project

To run the USBX Device Class Mass Storage module application project and to see it executed on a target kit, you can simply import it into your ISDE, compile, and run debug. See the included Application Note, *Renesas Synergy™ Project Import Guide* (r11an0023eu0121-synergy-ssp-import-guide.pdf) for instructions on importing the project into e^2 studio ISDE or IAR EW for Synergy, and building/running the application.

To implement the USBX Device Class Mass Storage module application in a new project, follow the steps to define, configure, auto-generate files, add code, compile, and debug on the target kit. Following these steps is a hands-on approach that can help make the development process with SSP more practical.

Note: The instructions are provided 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 of the *SSP User's Manual* for a description of how to accomplish these steps.

To create and run the USBX Device Class Mass Storage application project, simply follow these steps:

1. Create a new Renesas Synergy project for the SK-S7G2 called **USBX_Mass_Storage_Device**.
2. Select the **BSP** project template.
3. Open the **Configuration.xml** file from the project.
4. Select the Threads tab
5. Add a new thread called
   Symbol new_thread0
   Name New Thread
6. In the Threads tab, select the newly created **New Thread** and add **USBX Device Class Mass Storage** from New Thread Stacks X-Ware>USBX>Device>Classes>Mass Storage.
7. Change the configuration settings for USBX Device Mass Storage from the following properties:
   A. Set the **Last LBA** (31: The Last LBA value for `ramdisk_image.c`).
   B. Set the **Byte Per Sector** (512: The Byte Per Sector value for `ramdisk_image.c`).
   C. Set user-callback function name (the default names are **ux_device_msc_media_status**, **ux_device_msc_media_read**, and **ux_device_msc_media_write**).

8. Right click to **Add USBX Port DCD>New>USBX Port DCD on sf_el_ux for USBFS**.
9. Set **Property>Common>Full speed Interrupt Priority**: **Priority 3** (CM4: valid, CM0+; lowest—not valid if using ThreadX).
10. Click the **Generate Project Content** button.
11. Add the code from the supplied project file `ux_user_callback.c/ramdisk_image.c` or copy over the `generated ux_user_callback.c` file.
12. Compile the application without errors and warnings.
13. Connect to the USB micro cable at J19 on SK-S7G2 board. Connect the USB cable's other end to the host.
14. Start to debug the application.
15. Connect to the USB micro cable to J5 on SK-S7G2 board. Connect the USB cable's other end to the host.
16. On your PC, the added drive and the following initial file appears.

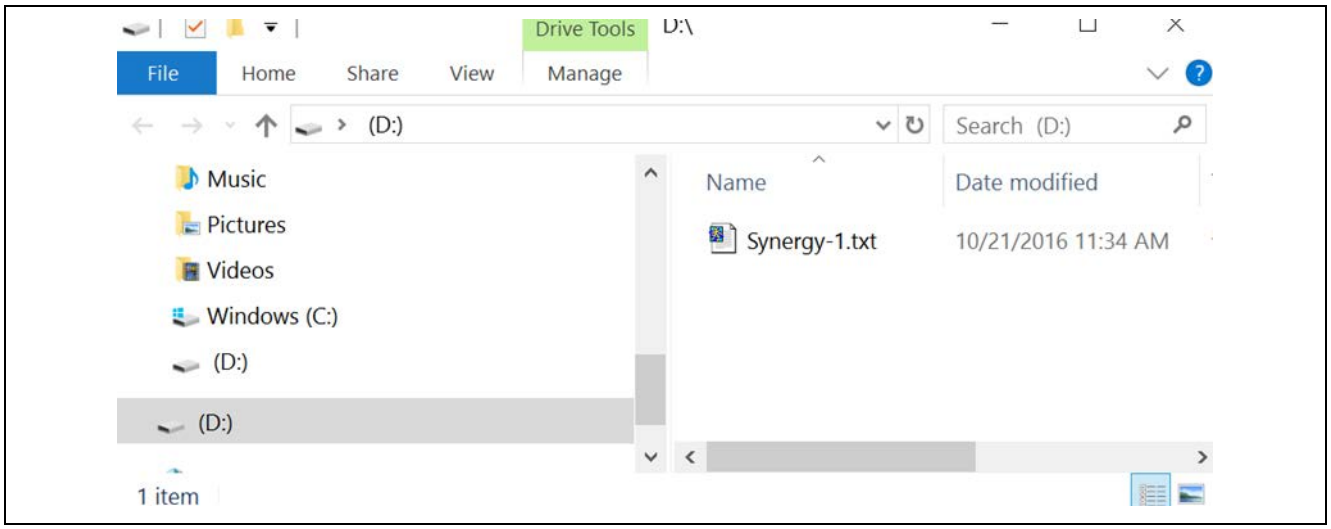

Figure 4. Example of USBX Device Class Mass Storage Module Application Project Initial File

9. USBX Device Class Mass Storage 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 low-level modules. The use of high-level APIs (as demonstrated in the application project) shows 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.

10. USBX Device Class Mass Storage Module Next Steps

After you have mastered a simple USBX Device Class Mass Storage module project, you may want to review a more complex example. Other application projects and application notes that demonstrate USBX Device Class Mass Storage use can be found as described in the next Reference Information section of this document.

11. USBX Device Class Mass Storage Module Reference Information

SSP User Manual: Available in html format in the SSP distribution package and as a pdf from the Synergy Gallery: www.renesas.com/synergy/ssp.

Links to all the most up-to-date ux_device_class_storage module reference materials and resources are available on the Synergy Knowledge Base: https://en-support.renesas.com/knowledgeBase/16977572.

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.09.17	—	Initial version
1.01	Sep.14.17	—	Update to hardware and software resources table
1.02	Jan.03.18	—	Minor edits for grammar and usage
1.03	Dec.10.18	—	Updated to 1.5.0
1.04	Mar.22.19	—	Updated to 1.6.0

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.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/.