Renesas Synergy™ Platform

USBX™ Synergy Port Framework 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™ Synergy Port Framework module (sf_el_ux) is integrated into the SSP. This driver is intended to be used with USBX APIs from Express Logic. For more information about USBX, including the API reference, see USBX documentation available in the References section at the end of this document.

Contents

1. USBX Synergy Port Framework Module Features

The USBX Synergy Port Framework module supports the following features:

- Implements Express Logic USBX in SSP—supports USBX APIs
- Supports the Port Device Controller Driver (DCD) for USBHS peripheral
- Supports the Port Device Controller Driver (DCD) for USBFS peripheral
- Supports the Port Host Controller Driver (HCD) for USBHS peripheral
- Supports the Port Host Controller Driver (HCD) for USBFS peripheral
- Supports transfer module operation (optional)

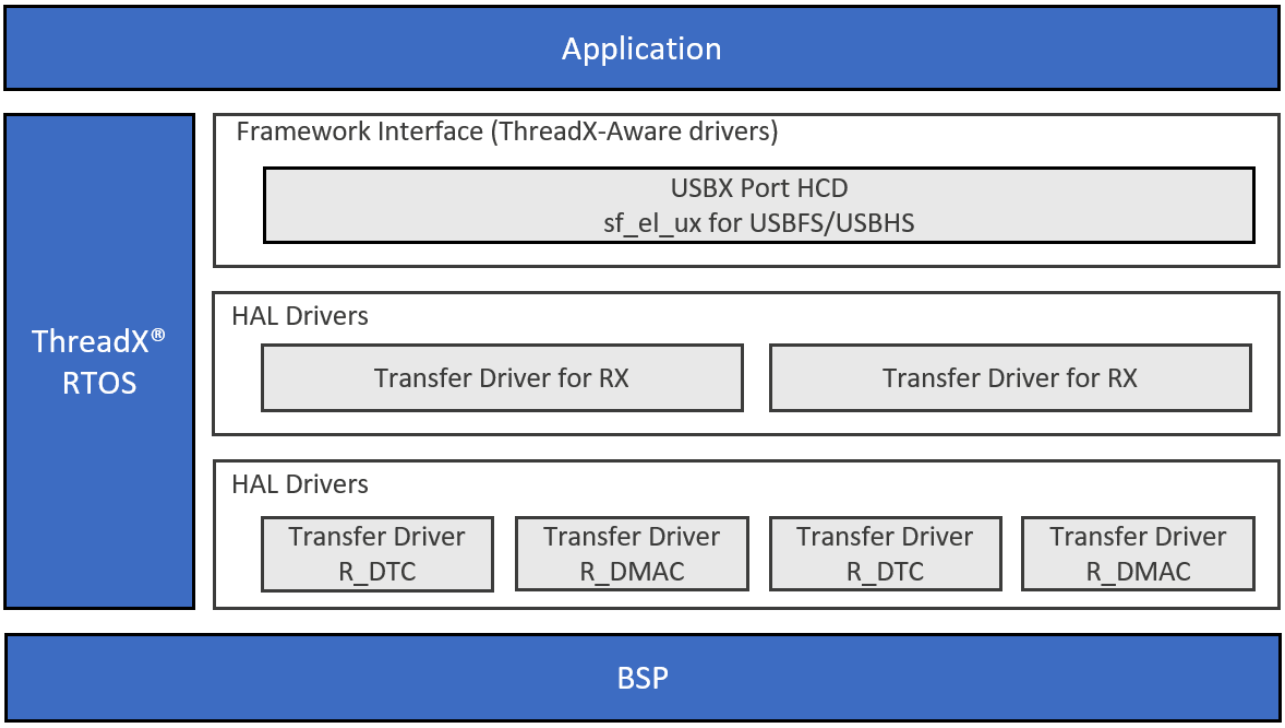

Figure 1. USBX Synergy Port Framework Module Organization, Options and Stack Implementations

2. USBX Synergy Port Framework Module API Overview

This module does not have its own API calls and implements USBX API calls from Express Logic. For details on available API calls, see Express Logic USBX documentation in the References section at the end of this document.

3. USBX Synergy Port Framework Module Operational Overview

This module provides USB hardware port functions that are required to use the USBX stack on Renesas Synergy hardware. Application code using this module must use USBX API calls from Express Logic.

3.1 USBX Synergy Port Framework Module Important Operational Notes and Limitations

Support for Express Logic USBX APIs is included for this module in the SSP. For details on available APIs, see Express Logic USBX documentation in the References section at the end of this document.

The USBX Synergy Port Framework module supports the Port Device Controller Driver (DCD) for the USBHS and USBFS peripherals, as well as the Port Host Controller Driver (HCD) for the USBHS and USBFS peripherals.

Users may use the Transfer Module option for the USBX Port Framework module. This option provides better USB data throughput by transferring data in block transfer mode. To enable the Transfer module, add two instances of components to the USBX Class stack in the Synergy Configuration tool, and in the property, enable the interrupts. The Synergy Configuration tool auto-generates the driver setup code to enable DMAC or DTC transfer in `common_data.c`.

3.2 USBX Synergy Port Framework Module Operational Notes

- The module uses the interrupt of a USB Controller. Set appropriate interrupt priority level in Synergy Configuration tool; otherwise, the interrupt will not work.
- The module uses the interrupt of a Transfer module (implemented as DMAC or DTC), if it is used. Set appropriate priority level in the Synergy Configuration tool. The level has to be higher than that of a USB Controller; otherwise, the interrupt will not work.
- The current version of the USBX Driver does not support the isochronous transfer. There is no support for any USBX classes using isochronous transfer.

See the latest SSP Release Notes for other limitations to operation of this module.

4. Including the USBX Synergy Port Framework Module in an Application

This section describes how to include the Express Logic USBX Synergy Port Framework module in an application using the SSP configurator.

Note: This section assumes 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 USBX Synergy Port 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 Synergy Port is `g_sf_el_ux`. This name can be changed in the associated Properties window.)

Table 1. USBX Synergy Port Selection Sequence

Resource	ISDE Tab	Stacks Selection Sequence
`g_sf_el_ux_hcd_hs_0` USBX Port HCD on sf_el_ux for USBHS	Threads	New Stack> X-Ware> USBX> Host > Synergy Port> USBX Port HCD on sf_el_ux for USBHS
`g_sf_el_ux_hcd_fs_0` USBX Port HCD on sf_el_ux for USBFS	Threads	New Stack> X-Ware> USBX> Host > Synergy Port> USBX Port HCD on sf_el_ux for USBFS
`g_sf_el_ux_dcd_hs_0` USBX Port HCD on sf_el_ux for USBHS	Threads	New Stack> X-Ware> USBX> Device > Synergy Port> USBX Port HCD on sf_el_ux for USBHS
`g_sf_el_ux_dcd_fs_0` USBX Port HCD on sf_el_ux for USBFS	Threads	New Stack> X-Ware> USBX> Device > Synergy Port> USBX Port HCD on sf_el_ux for USBFS

When the USBX Synergy Port Framework module on sf_el_ux is added to the thread stack, the configurator automatically adds any needed lower-level modules. Any drivers needing additional configuration information have 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, need to only be added once, and can be used by multiple stacks. Modules with a Pink band can require the selection of lower-level drivers; these are either optional or recommended (this is indicated in the block with the inclusion of this text.) If the addition of lower-level drivers is required, the module description includes "Add" in the text. Clicking on any Pink banded modules brings up the "New" icon and then displays possible choices.

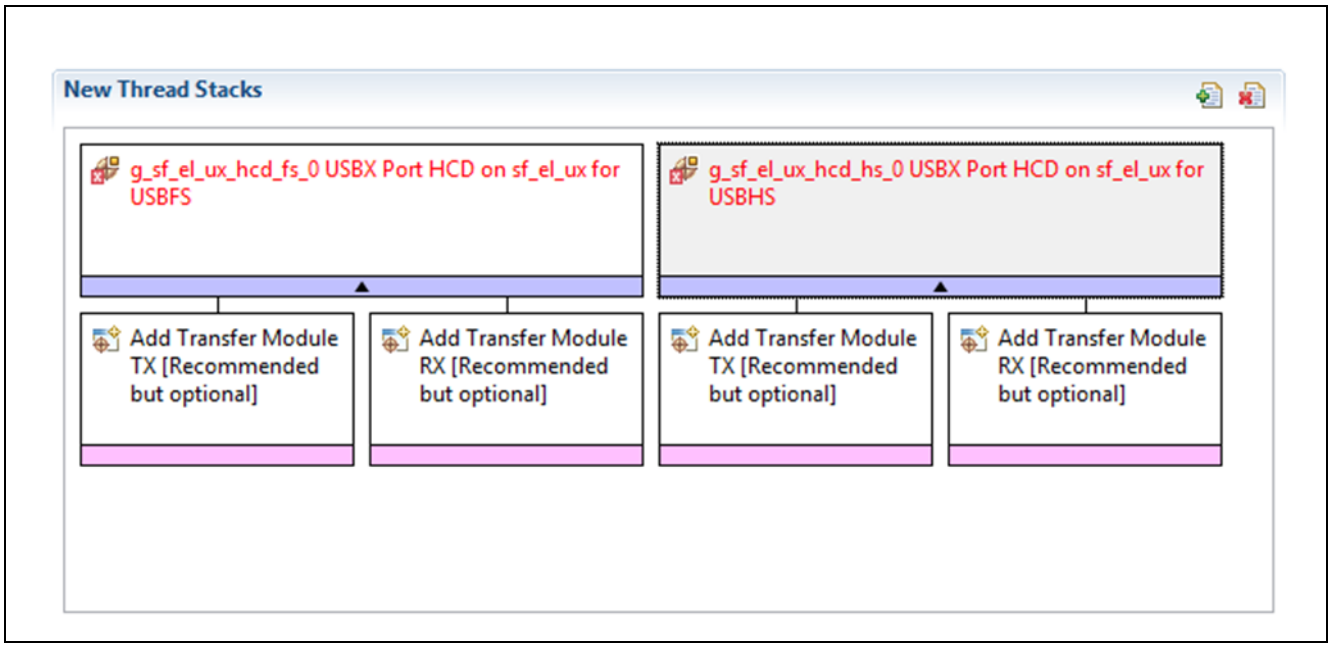

Figure 2. Express Logic USBX Synergy Port Framework Module Stack

5. Configuring the USBX Synergy Port Framework Module

Configure the USBX Synergy Port Framework module for your 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 not available for changes; these properties are identified with a lock icon for the 'locked' property in the Properties window in the 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 this module guide for easy reference.

One of the properties that often requires changing 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 in the ISDE indicates the validity of the setting based on the MCU targeted (CM4 or CM0+). This level of detail is not included in the following configuration properties tables, but is easily visible with 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 looking over the following configuration table settings. This will help 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 Synergy Port DCD on sf_el_ux for USBFS

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)	Sets the priority level of the full speed interrupt
Name	G_sf_el_ux_dcd_fs_0	Module name.
USB Controller Selection	USBFS	USB Controller type

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

Table 3. Configuration Settings for the Express Logic USBX Port DCD on sf_el_ux for USBHS

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	Sets the priority level of the full speed interrupt
Name	g_sf_el_ux_dcd_hs_0	Module name.
USB Controller Selection	USBHS	USB Controller type

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

Table 4. Configuration Settings for the USBX Synergy Port HCD on sf_el_ux for USBFS

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	Sets the priority level of the high-speed interrupt
VBUSEN pin Signal Logic	Active High, Active Low (Default: Active High)	VBUS pin level setting
Name	g_sf_el_ux_hcd_fs_0	Module name.
USB Controller Selection	USBFS	USB Controller type

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

Table 5. Configuration Settings for the USBX Synergy Port HCD on sf_el_ux for USBHS

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	Sets the priority level of the high-speed interrupt
VBUSEN pin Signal Logic	Active High, Active Low (Default: Active High)	VBUS pin level setting
Name	g_sf_el_ux_dcd_fs_0	Module name.
USB Controller Selection	USBFS	USB Controller type

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

In some cases, settings other than the defaults for lower-level modules can be desirable. For example, it might be useful to select the DAC Channel based on the target hardware implementation. The configurable properties for the lower-level stack modules are given in the following sections for completeness and as a reference.

Note: Most of the property settings for lower-level modules are intuitive and usually can be determined by inspection of the associated Properties window from the SSP configurator.

5.1 Configuration Settings for USBX Synergy Port Framework Module Low-Level Modules

Typically, only a small number of settings must be modified from the default for lower-level modules as 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 settings within the properties section for the module.

Note: Only one copy of the lower level drivers is provided. The three other transfer drivers sets are similar to those provided, with only minor differences between RX and TX implementations. To see the differences, view the properties in the SSP configuration window.

Table 6. Configuration Settings for the Transfer Driver on r_dmac (Transfer Driver Option)

ISDE Property	Value	Description
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Selects if code for parameter checking is to be included in the build
Name	g_transfer0	Module name
Channel	0	Channel selection
Mode	Block	Mode selection
Transfer Size	1 Byte	Transfer size selection
Destination Address Mode	Fixed	Destination address mode selection
Source Address Mode	Incremented	Source address mode selection
Repeat Area (Unused in Normal Mode	Source	Repeat area selection
Destination Pointer	NULL	Destination pointer selection
Source Pointer	NULL	Source pointer selection
Number of Transfers	0	Number of transfers selection
Number of Blocks (Valid only in Block Mode)	0	Number of blocks selection
Activation Source	Software Activation	Activation source selection
Auto Enable	False	Auto enable selection
Callback	NULL	Callback selection
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)	Interrupt priority selection

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

Table 7. Configuration Settings for the Transfer Driver on r_dtc (Transfer Driver Option)

ISDE Property	Value	Description
Parameter Checking	BSP, Enabled, Disabled (Default: BSP)	Selects if code for parameter checking is to be included in the build
Software Start	Enabled, Disabled (Default: Disabled)	Software start selection
Linker section to keep DTC vector table	.ssp_dtc_vector_table	Linker section selection
Name	g_transfer0	Module name
Mode	Block	Mode selection
Transfer Size	1 Byte	Transfer size selection

ISDE Property	Value	Description
Destination Address Mode	Fixed	Destination address mode selection
Source Address Mode	Incremented	Source address mode selection
Repeat Area (Unused in Normal Mode	Source	Repeat area selection
Interrupt Frequency	After all transfers have completed	Interrupt frequency selection
Destination Pointer	NULL	Destination pointer selection
Source Pointer	NULL	Source pointer selection
Number of Transfers	0	Number of transfers selection
Number of Blocks (Valid only in Block Mode)	0	Number of blocks selection
Activation Source (Must enable IRQ)	Software Activation 1	Activation source selection
Auto Enable	False	Auto enable selection
Callback (Only valid with Software start)	NULL	Callback selection
ELC Software Event 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)	ELC Software Event interrupt priority selection.

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

5.2 USBX Synergy Port Framework Module Clock Configuration

The USB peripheral module uses UCLK as its clock source. The clock frequency must be set to 48 MHz and this can be done in the Clocks tab of the configuration window.

5.3 USBX Synergy Port Framework Module Pin Configuration

The USB peripheral module uses pins on the MCU to communicate to external devices. I/O pins must be selected and configured as required by the external device. The following table illustrates the method for selecting the pins within the SSP configuration window (depending on the USB function desired- USBFS or USBHS) and the subsequent tables illustrate example selections for the USB pins in each function.

Note: The operation mode determines what peripheral signals are available and thus what MCU pins are required. The operation mode selection must be consistent with the mode used by the USBX Device Class CDC-ACM module.

Table 8. Pin Selection Sequence for USBFS and USBHS

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

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

Table 9. Pin Configuration Settings for the USBX Device Module on USBFS0

Properties	Value	Description
Operation Mode	Disabled, Custom, Device, Host, OTG (Default: Disabled) Device	Select Device as the Operation Mode for USB CDC-ACM
USBDP	USBDP	USPDP
USBDM	USBDM	USBDM
OVRCURB	None	OVRCURB
OVRCURA	None	OVRCURA
VBUSEN	None	VBUSEN
VBUS	None, P407 (Default: P407)	VBUS
EXICEN	None	EXICEN
ID	None	ID
VCCUSB	VCCUSB	VCCUSB
VSSUSB	VSSUSB	VSSUSB

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

Table 10. Pin Configuration Settings for the USBX Device on USBHS0

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

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

6. Using the USBX Synergy Port Framework Module in an Application

Once the USBX port framework module is added to a thread, the USBX APIs become available to use.

7. USBX Synergy Port Framework Module Application Project

The application project associated with this module guide demonstrates the aforementioned steps in an example application. You may want to import and open the application project within the ISDE and view the configuration settings for the Express Logic USBX Synergy Port module. The project can be found using the link provided in the Reference Section at the end of this document. You can also read over the code in *usb_thread0_entry.c.*

The application project demonstrates the typical use of user callback function from USBX. The application project main thread entry receives event flag notification from the user callback function and displays the

type of event that occurred to the debugging console. The user callback function saves the information on the event that occurred to variables and notifies it to the main thread using the event flag. The following table identifies the target versions for the associated software and hardware used by the application project.

Table 11. Software and Hardware Resources Used by the Application Project

Resource	Revision	Description
e^2 studio	5.3.1 or later	Integrated Solution Development Environment
SSP	1.2.0 or later	Synergy Software Platform
IAR EW for Synergy	7.71.2 or later	IAR Embedded Workbench® for Renesas Synergy™
SSC	5..3.1 or later	Synergy Standalone Configurator
SK-S7G2	v3.0 to v3.1	Starter Kit

A simple flow diagram of the application project is given in the following figure:

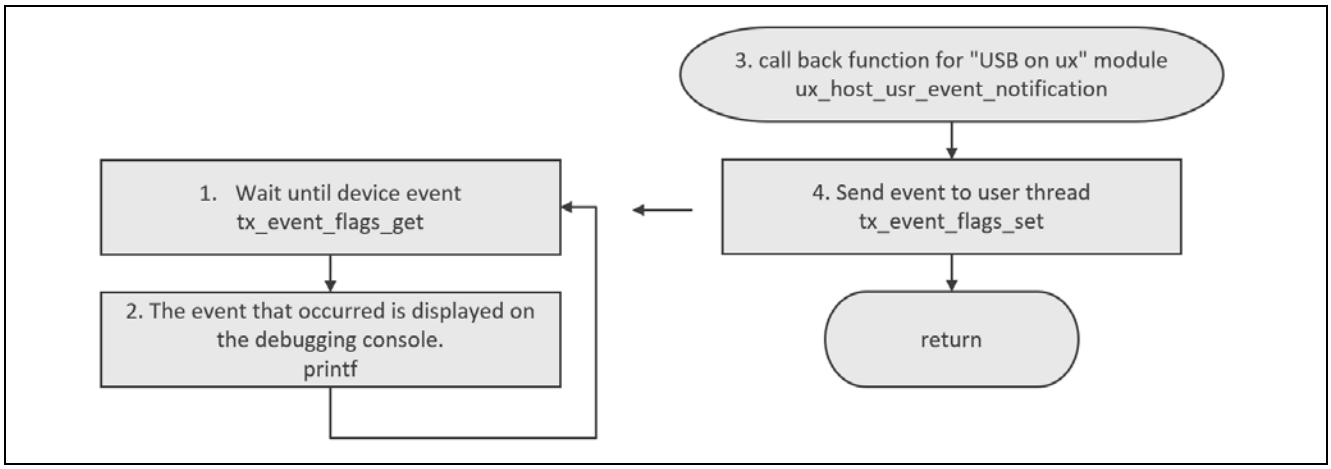

Figure 3. USBX Synergy Port Framework Module Application Project Flow Diagram

Find the complete Application Project using the link in the Reference Section at the end of this document. The *usb_thread0_entry.c* file is in the project once it has been imported into the ISDE. You can open these files within the ISDE and follow along with this section to help identify key uses of APIs.

The first section of `usb_thread0_entry.c` has a header file for the user thread. The next section contains definitions and variables used in the application. The subsequent section has a callback function used when USB memory is connected or disconnected. The last section is the user thread.

When connected to USB memory, `ux_host_usr_event_notification` is called. This callback function determines the connected device from the argument, saves the pointer of the FileX instance, and notifies the user thread. This callback function is also called when the USB memory was unplugged.

The user thread receives notification from the callback function. In the user thread, the event type and class name saved by the callback function display to the debug console. The callback function is called on both the connection and disconnection cases. Therefore, the display from the user thread is also done at connection and disconnection.

A few key properties are configured in this Application Project to support required operations, as well as the physical properties of the target board and MCU. The following table lists 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 12. USBX Synergy Port Framework Module Configuration Settings for the Application Project

ISDE Property	Value Set
USBX Port HCD on sf_el_ux for USBHS High Speed Interrupt Priority	Priority 3
VBUSEN pin Signal Logic	Active Low
USBX on ux USBX Pool Memory Size	32768
User Callback for Host Event Notification (only valid for USB Host)	ux_host_usr_event_notification

8. Customizing the USBX Synergy Port Framework Module for a Target Application

Some configuration settings can be changed by the developer from those shown in the Application Project. For example, you can also select DTC and DMAC as transfer drivers. This data transfer module can be added simply by clicking TX or RX box displayed under the USBX Port HCD box of the configurator.

For the DMAC driver, you need to select its interrupt. When adding the DMAC driver be sure to set the interrupt priority at a higher priority (a lower interrupt value setting) than the interrupt for the higher level USB module. Setting the interrupt priority higher improves driver functionality and also enables DMAC support.

9. Running the USBX Synergy Port Framework Module Application Project

To run the USBX Synergy Port Framework Application project and to see it executing on a target kit, you can simply import it into your ISDE, compile and run debug.

To implement the USBX Synergy Port Framework application in a new project, use the included steps to define, configure, auto-generate files, add code, compile, and debug the project 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 will tend to be more theoretical.

Note: The instructions provided are in sufficient detail for someone experienced with the basic flow through the Synergy development process. If these steps are not familiar, see *SSP User's Manual* listed in the Reference Section at the end of this document.

To create and run the USBX Synergy Port Framework Application Project simply follow these steps:

1. Create a new Renesas Synergy™ project for the SK-S7G2 called SF_EL_UX_MG_AP.
2. Select the **Threads** tab.
3. Add **usb_thread0** in the Threads window.
4. Add the object, **g_usb_activate_event_flags0** under the USB thread.
5. Add the **FileX on USBX Mass Storage** module to the USB Thread Stacks.
6. Click the box, **USBX on ux**, on the USB Thread Stacks.
7. Change the field, USBX Pool Memory Size, to **32768** within the property window.
8. Change the field, User Callback for Host Event Notification (only valid for USB Host), to `ux_host_usr_event_notification` within the property window.
9. Click the box, **Add USBX Port HCD**, and select **USBX Port HCD on sf_el_ux for USBHS**.
10. Change the field, High Speed Interrupt Priority, to **Priority 3**, within the property window.
11. Change VBUSEN pin Signal Logic to **Active Low** within the property window.
12. Click the button, **Generate Project Content**.
13. Add the code from the supplied project file, `usb_thread0_entry.c`, or copy over the generated `usb_thread0_entry.c` file.
14. Add **#define SEMI_HOSTING** in the code to enable printf function.
15. Build the application code and connect the board to flash the executable binary.
16. On the SK-S7G2 Kit, insert the USB memory device into the J6 connector.
    The connection message displays on the debug console.
17. On the SK-S7G2 Kit, remove the USB memory device.
    A disconnection message displays on the debug console.

```
device insertion
class name [ux_host_class_storage]
device removal
class name [ux_host_class_storage]

device insertion
class name [ux_host_class_storage]
device removal
class name [ux_host_class_storage]
```

Figure 4. Example Output from USBX Port Framework Module Application Project

10. USBX Synergy Port 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. USBX Synergy Port Framework Module Next Steps

After you have mastered a simple USBX Port Framework project you, may want to review a more complex example. Other Application Projects and Application Notes demonstrating USB usage with the Synergy Platform are available as described in the References section at the end of this document.

12. USBX Synergy Port 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 sf_el_ux module reference materials and resources are available on the Synergy Knowledge Base: https://en-support.renesas.com/search/SF_EL_UX%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	Sep.12.17	—	Initial Release
1.01	Jan.14.19	—	Minor updates to configuration tables and settings.

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/.