Renesas Synergy™ Platform

ELC HAL Module Guide

R11AN0101EU0110
Rev.1.10
Nov 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 an 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 in this document), and should be valuable resources for creating more complex designs.

The Event Link Controller (ELC) HAL module is a high-level API for ELC HAL applications and is implemented on `r_elc`. The ELC HAL module uses the ELC peripheral on the Synergy MCU. There are no callbacks associated with the ELC HAL module. The project configurator in the e^2 studio Integrated Solution Development Environment (ISDE), includes the ELC HAL module in every project by default. To configure the ELC HAL module, select it in the HAL/Common module in the **Threads** tab, and then click on it in the HAL/Common Stacks window.

Contents

1. ELC HAL Module Features

The ELC HAL module supports the following functions:

- Create an event link between two blocks.
- Break an event link between two blocks.
- Generate one of two software events that interrupt the CPU.

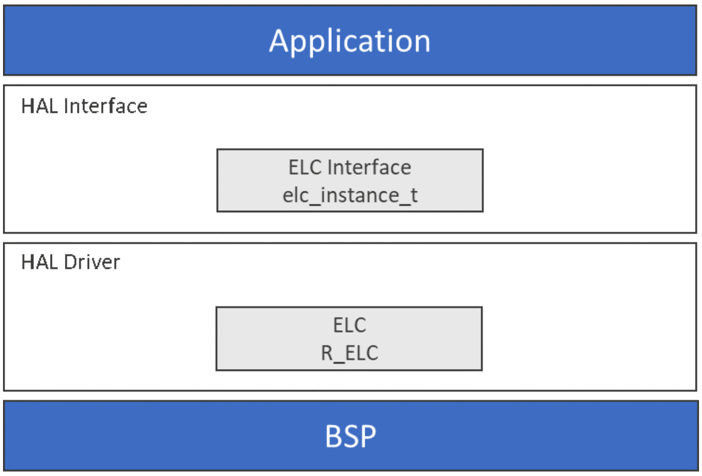

Figure 1 ELC HAL Module Block Diagram

2. ELC HAL Module APIs Overview

The ELC HAL module defines APIs for initializing, enabling, disabling and creating or breaking event links between modules. 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 ELC HAL Module API Summary

Function Name	Example API Call and Description
`.init`	`g_elc.p_api->init(g_elc.p_cfg)` Initialize all links in the Event Link Controller.
`.softwareEventGenerate`	`g_elc.p_api->softwareEventGenerate(event_num)` Generate a software event in the Event Link Controller.
`.linkSet`	`g_elc.p_api->linkSet(peripheral, signal)` Create a single event link.
`.linkBreak`	`g_elc.p_api->linkBreak(peripheral)` Break an event link.
`.enable`	`g_elc.p_api->enable()` Enable the operation of the Event Link Controller.
`.disable`	`g_elc.p_api->disable()` Disable the operation of the Event Link Controller.
`.versionGet`	`g_elc.p_api->versionGet(&version)` Retrieve the API version with the version pointer.

Note: For details on operation and definitions for the 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	Function successfully completed.
SSP_ERR_ASSERTION	p_version is NULL.

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. ELC HAL Module Operational Overview

The ELC HAL module allows the developer to link various peripheral operations by using events generated by one peripheral to trigger the start of operation of another peripheral within the Synergy MCU. The ELC HAL module APIs make it easy to create a link between two blocks (for example, from a timer to an ADC to control a periodic scan interval). By connecting various peripherals in this way, intelligent functions can be constructed that require little, if any, CPU intervention.

The following figure shows a simplified block diagram of the ELC, showing the input event sources and the peripherals that can be triggered by these events. (The number of input and output triggers is specific to the S7G2 MCU.) Other Synergy devices support a different number of events.

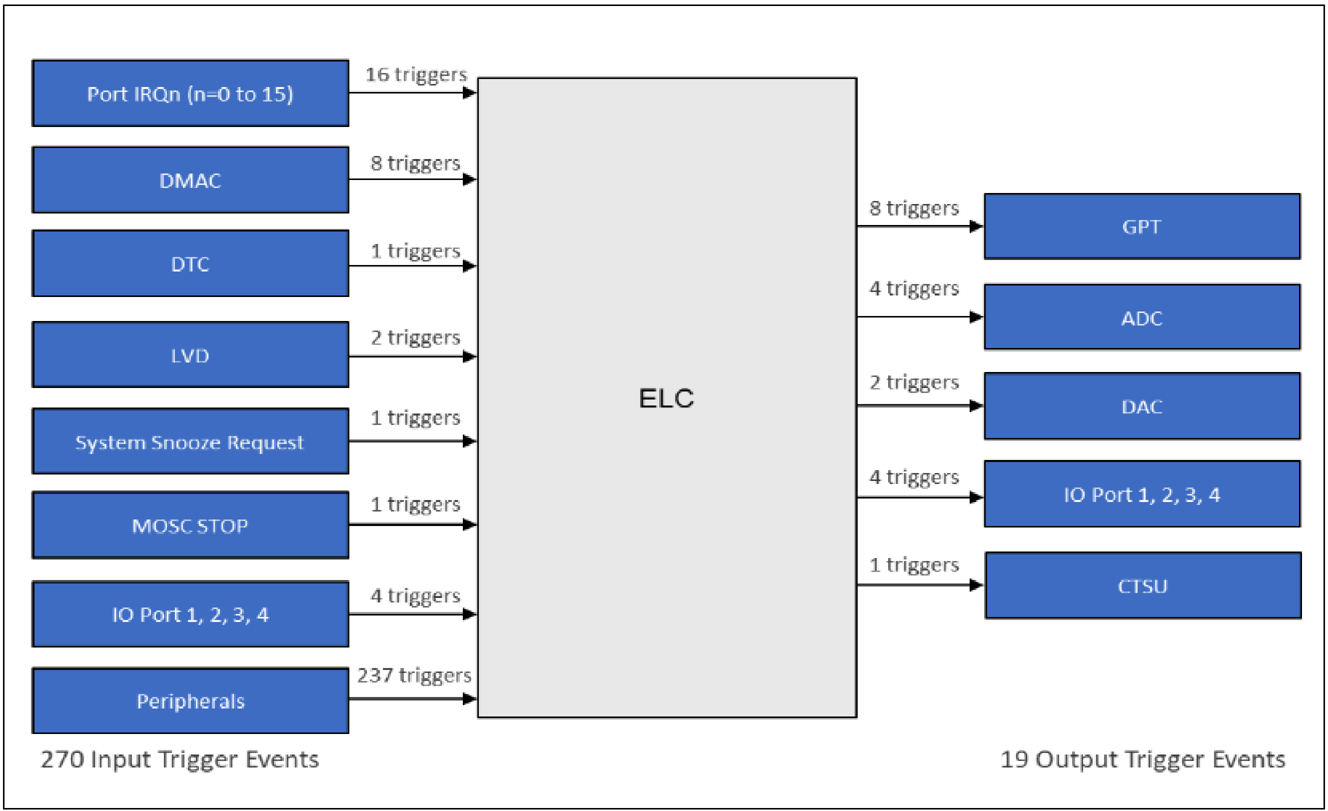

Figure 2 ELC Hardware Block Diagram

The application project associated with this module guide shows an example of the ELC in use.

It is possible to find the mapping of ELC peripherals in the file, `bsp_elc.h`, in the Synergy-generated code. The event list and peripherals for the S7G2 microcontroller unit (MCU) follows. Additional information on ELC operation is also available in the associated MCU User's Manual.

```
/** Possible peripherals to be linked to event signals */
typedef enum e_elc_peripheral
{
    ELC_PERIPHERAL_GPT_A                                        =    (0),
    ELC_PERIPHERAL_GPT_B                                        =    (1),
    ELC_PERIPHERAL_GPT_C                                        =    (2),
    ELC_PERIPHERAL_GPT_D                                        =    (3),
    ELC_PERIPHERAL_GPT_E                                        =    (4),
    ELC_PERIPHERAL_GPT_F                                        =    (5),
    ELC_PERIPHERAL_GPT_G                                        =    (6),
    ELC_PERIPHERAL_GPT_H                                        =    (7),
    ELC_PERIPHERAL_ADC0                                         =    (8),
    ELC_PERIPHERAL_ADC0_B                                       =    (9),
    ELC_PERIPHERAL_ADC1                                         =   (10),
    ELC_PERIPHERAL_ADC1_B                                       =   (11),
    ELC_PERIPHERAL_DAC0                                         =   (12),
    ELC_PERIPHERAL_DAC1                                         =   (13),
    ELC_PERIPHERAL_IOPORT1                                      =   (14),
    ELC_PERIPHERAL_IOPORT2                                      =   (15),
    ELC_PERIPHERAL_IOPORT3                                      =   (16),
    ELC_PERIPHERAL_IOPORT4                                      =   (17),
    ELC_PERIPHERAL_CTSU                                         =   (18),
} elc_peripheral_t;


/** Sources of event signals to be linked to other peripherals or the CPU1
 * @note This list may change based on device. This list is for S7G2.
 * */
typedef enum e_elc_event
{
    ELC_EVENT_ICU_IRQ0                                          =    (1),
    ELC_EVENT_ICU_IRQ1                                          =    (2),
    ELC_EVENT_ICU_IRQ2                                          =    (3),
    ELC_EVENT_ICU_IRQ3                                          =    (4),
.
.
.
    ELC_EVENT_ICU_IRQ12                                         =   (13),
    ELC_EVENT_ICU_IRQ13                                         =   (14),
    ELC_EVENT_ICU_IRQ14                                         =   (15),
    ELC_EVENT_ICU_IRQ15                                         =   (16),
.
.
.
    ELC_EVENT_GLCDC_LINE_DETECT                                 =  (506),
    ELC_EVENT_GLCDC_UNDERFLOW_1                                 =  (507),
    ELC_EVENT_GLCDC_UNDERFLOW_2                                 =  (508),
    ELC_EVENT_DRW_INT                                           =  (509),
    ELC_EVENT_JPEG_JEDI                                         =  (510),
    ELC_EVENT_JPEG_JDTI                                         =  (511),
} elc_event_t;
```

Figure 3 Mapping of ELC Peripherals in bsp_elc.h Example

3.1 ELC HAL Module Important Operational Notes and Limitations

3.1.1 ELC HAL Module Operational Notes

The ELC HAL module needs no pin, clocking, or interrupt configuration. It is just a ‘connect’ mechanism between peripherals. However, if linking I/O Ports via the ELC, the I/O pins need to be configured as inputs or outputs.

3.1.2 ELC HAL Module Limitations

See the latest *SSP Release Notes* for any additional operational limitations for this module.

4. Including the ELC HAL Module in an Application

The ISDE automatically adds the necessary ELC HAL module to the HAL/Common thread in the project by default. As such, the steps described later in this section are not necessary or even forbidden (if the ELC HAL module is already configured in terms of the HAL/Common thread — only one driver instance is allowed). The following descriptions are provided for completeness and in case the ELC HAL module is mistakenly removed.

This section describes how to include the ELC HAL module in an application using the SSP configurator.

Note: It is assumed that 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 tasks, 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 ELC Driver to an application, simply add it to a thread using the stacks selection sequence given in the following table. (The default name for the ELC HAL module is `g_elc`. This name can be changed in the associated Properties window.)

Table 3 ELC HAL Module Selection Sequence

Resource	ISDE Tab	Stacks Selection Sequence
g_elc ELC Driver on r_elc	Threads	New Stack> Driver> System> ELC Driver on r_elc

When the ELC HAL Driver on `r_elc` is added to the thread stack as shown in the following figure, the configurator automatically displays the options dialog. The only available change to the ELC HAL module is to enable or disable parameter checking.

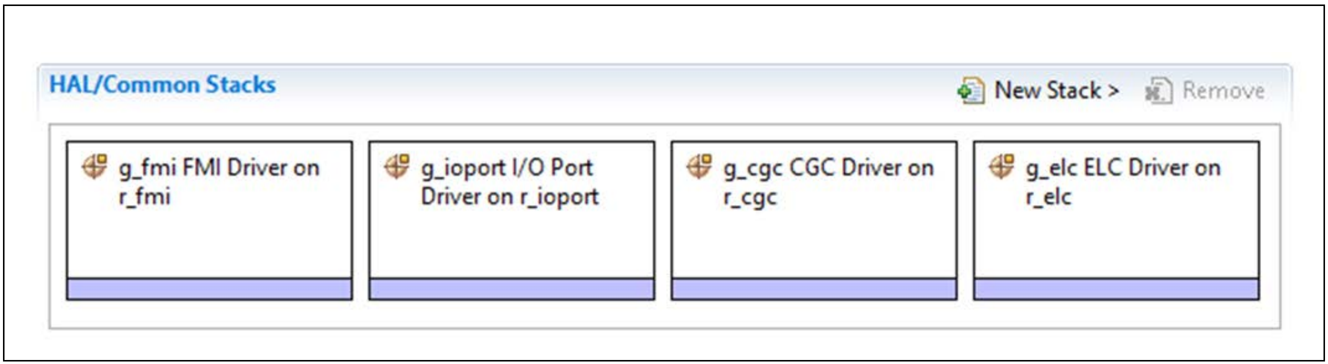

Figure 4 ELC HAL Module Stack

5. Configuring the ELC HAL Module

The ELC HAL module must be configured by the user 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 not available for changes and 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 the following tables for easy reference.

Note: You may want to open your ISDE and create the ELC HAL 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 4 Configuration Settings for the ELC HAL Module on r_elc

ISDE Property	Value	Description
Parameter Checking	BSP, Enabled, Disabled (Default: BSP)	Enable or disable the parameter error checking
Name	g_elc	Module name

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

5.1 ELC HAL Module Clock Configuration

There is no clock configuration for the ELC block.

5.2 ELC HAL Module Pin Configuration

There are no pins associated directly with the ELC HAL Module that require configuration.

6. Using the HAL Module in an Application

The typical steps in using the ELC HAL module in an application are:

1. Initialize the ELC using the `init` and `enable` APIs (automatically done by ISDE).
2. Link a peripheral with an event using the `linkSet` API.
3. Enable the linkage with the `enable` API.

The following figure illustrates the common steps in a typical operational flow diagram.

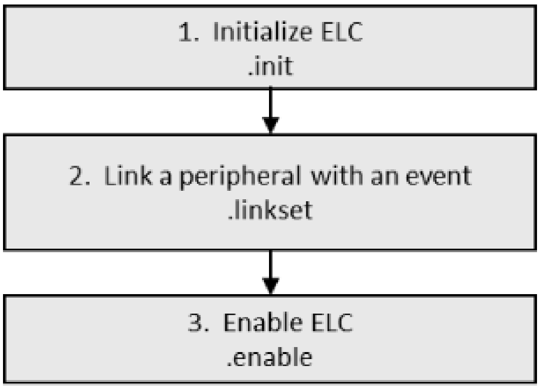

Figure 5 Flow Diagram of a Typical ELC HAL Module Application

7. The ELC HAL Module Application Project

The application project associated with this module guide demonstrates the steps performed in a full design. The project can be found using the link provided in the References section in this document. You may want to import and open the application project within the ISDE and view the configuration settings for the ELC HAL module. You can also read over the code (in `elc_hal.c`), which is used to illustrate the ELC HAL module APIs in a complete design.

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
e² studio	5.3.1 or later	Integrated Solution Development Environment
SSP	1.2.0 or later	Synergy Software Platform
IAR EW for Renesas 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

The application project demonstrates the use of the ELC HAL module APIs. The application links an external IRQ event to a DAC channel and links an AGT Timer event to an ADC channel. To enable the user to see some real-time activity, the application uses a GPT timer, an external IRQ ISR, and the DTC. These peripherals are not required for the ELC, but are used only for demonstration purposes.

The project main thread entry initializes the ADC, DAC, DTC, AGT, and GPT drivers. An ADC scan is periodically triggered by the AGT timer via an ELC event. The ADC input voltage is the output of the DAC. A physical pin connection has to be made to see the application running. The DAC output is enabled in response to pressing of the user pushbutton SW4. SW4 generates an IRQ10 event, which is linked via the ELC to the DAC peripheral. Pressing the user pushbutton SW5 generates an IRQ11 event, which in turn generates a CPU interrupt. This ISR disables the DAC output. The GPT timer is free-running and its counter value is transferred to the DAC write register by the DTC, in response to ADC scan end interrupt. This means that after every ADC scan (triggered by the AGT), a new GPT timer counter value is output by the DAC. The scanned ADC result is represented using three LEDs as a 3-bit binary display.

The following figure shows a simple flow diagram of the application project.

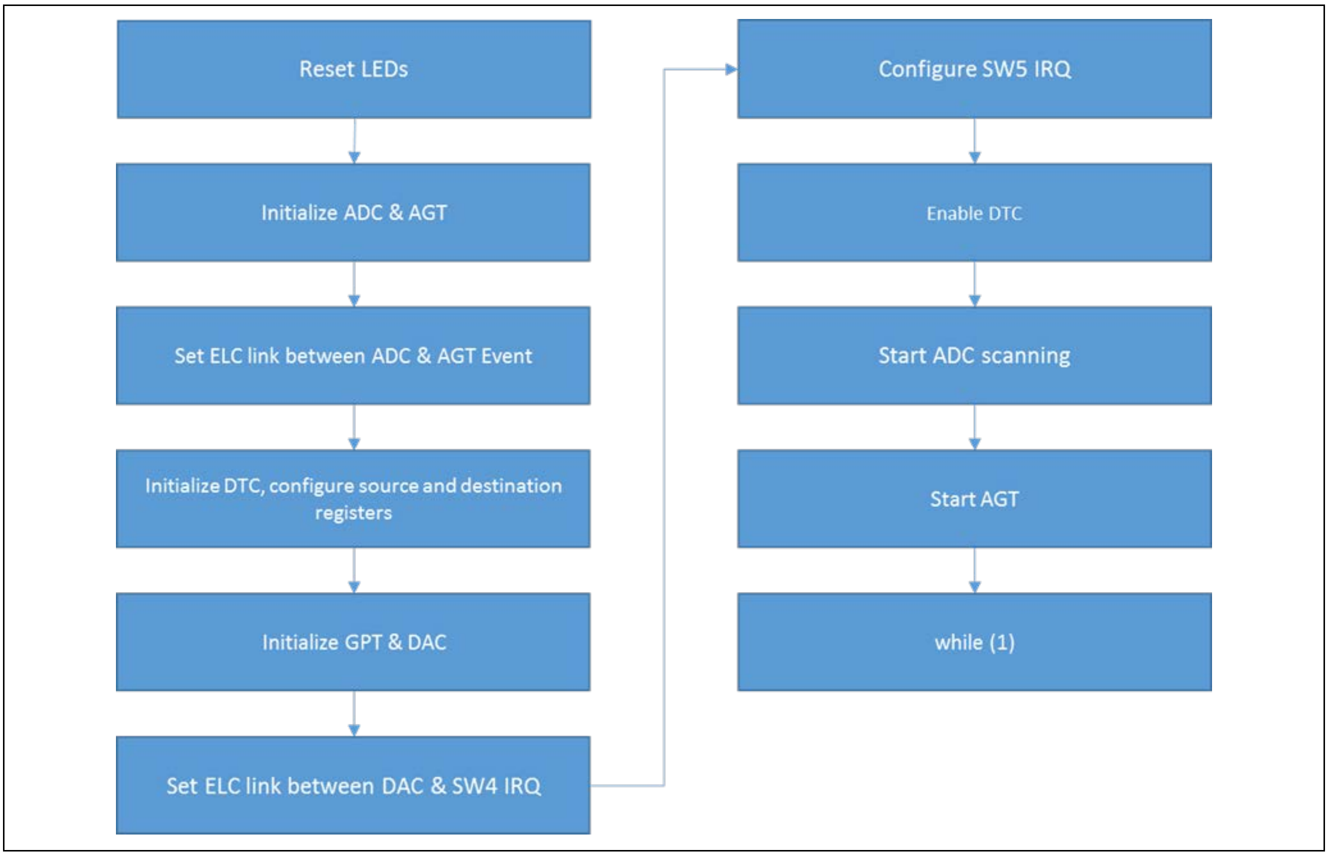

Figure 4 ELC HAL Module Application Project Flow Diagram

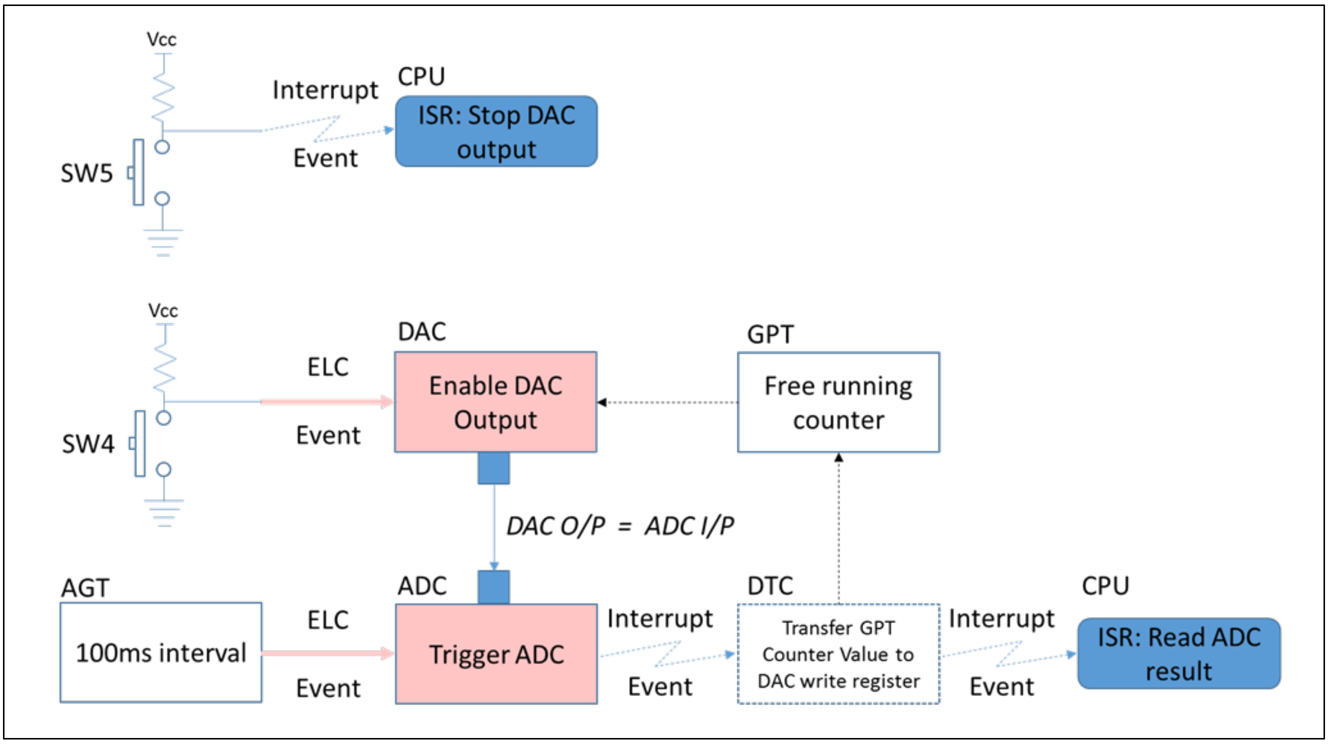

Figure 5 ELC HAL Module Application Project Architecture Diagram

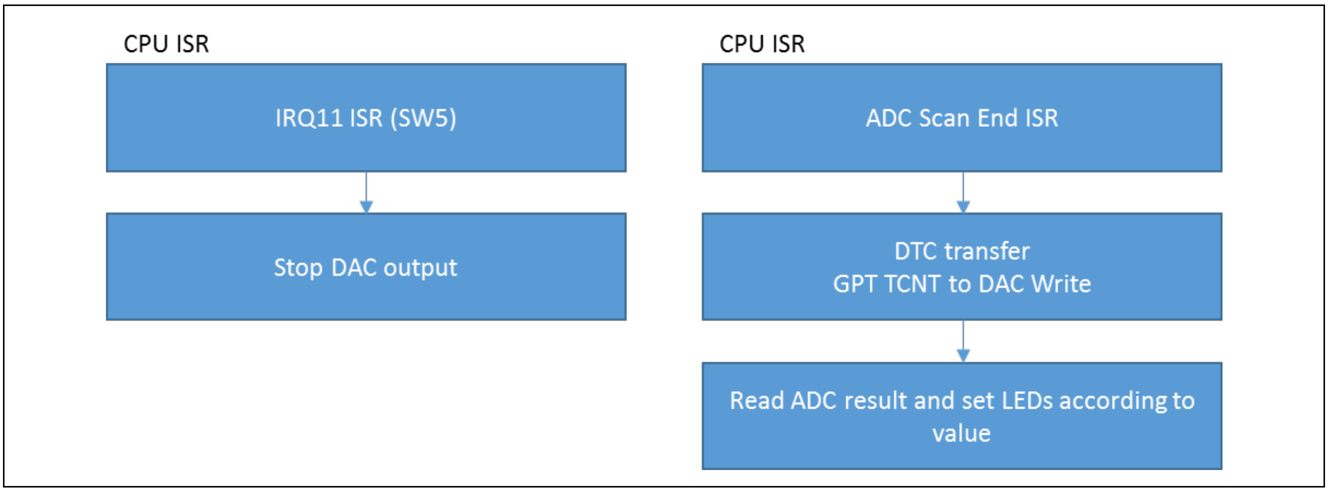

Figure 6 ELC HAL Module Application Project CPU ISR Diagrams

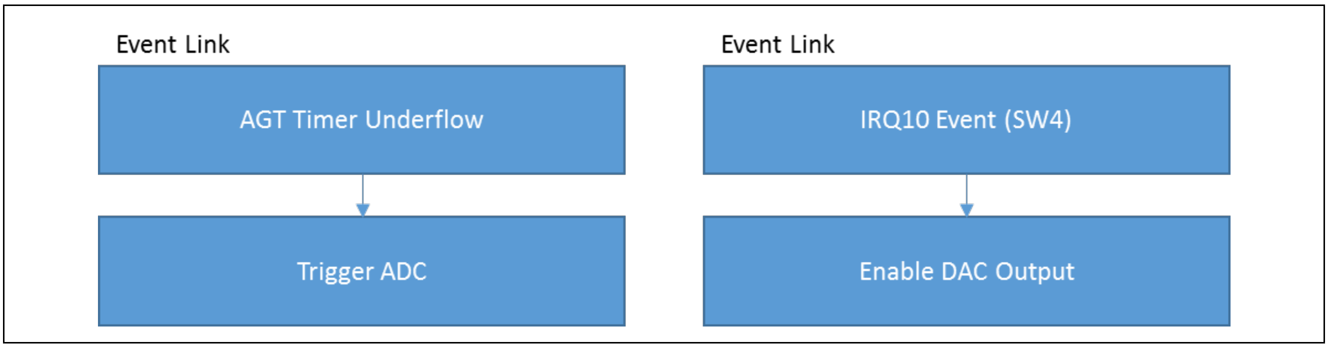

Figure 7 ELC HAL Module Application Project Event Link Diagrams

The `elc_hal.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 the key uses of APIs.

The first section of `elc_hal.c` is the #include and function prototypes section.

The second section is the `elc_hal_module_guide_project` function. This performs the initialization and configuration of the peripherals, starts the timers, and makes the application run.

The LEDs are turned off, the ADC is configured to scan channel 0, and the AGT timer is configured. With these two peripherals configured, an ELC link is set between the ADC peripheral (ELC_PERIPHERAL_ADC0) and AGT timer overflow event (ELC_EVENT_AGT0_INT). This enables the AGT to trigger an ADC scan.

Next, the DAC driver and a GPT timer are initialized and the DTC is configured to transfer data from the GPT counter value register (GTCNT) to a DAC output register (DADRn[0]). With these peripherals configured, an ELC link is set between a DAC peripheral (ELC_PERIPHERAL_DAC0) and IRQ 11 – SW4 user button – event (ELC_EVENT_ICU_IRQ11). This setting enables the SW4 user button to enable the DAC output.

Note: To make use of the Link Event Generated by the IRQ11 pin, there is no need to open and enable the IRQ with the SSP. Opening the IRQ means that the signal is passed into the NVIC, and the CPU, as an interrupt request. This is not required when using the IRQ signal via the ELC.

To disable the DAC output, the user button SW5, IRQ10 is used. The External IRQ driver for channel 10 is configured with a callback function described later on in this document.

Finally, the DTC transfer is enabled, and ADC scanning, and AGT timer are started. From now on, the application is controlled by interrupts and ELC events, and the main thread enters an infinite loop.

This is followed by the ADC callback function. This function gets called after the AGT timer overflow event (ELC_EVENT_AGT0_INT), which takes place every 100 ms. In response to the overflow event, the ADC reads ADC0 input pin (P00), which should be physically connected to the DAC output pin (P014), and depending on the read value, the LEDs are updated. The three LEDs serve as a simple binary display that can display numbers from 0 to 7 in binary format. The green LED (0) is the LSB, and the amber LED (2) is the MSB. To map the 12-bit value of ADC scan result to 3 bits, the value is divided by 512 and that result (ranging from 0 to 7) is displayed using LEDs.

The ADC callback function is followed by the IRQ10 (user button SW5) callback function section. As previously stated, SW5 disables the DAC output and makes the I/O pin high impedance. Due to the very low output impedance of the DAC, and the very high input impedance of the ADC, the voltage being displayed by the LEDs decays very slowly. Fitting a high value resistor between the ADC input and GND results in the ADC input voltage being 0, when the DAC output is disabled and the LEDs are turned off.

Finally, there is the `update_leds` function that facilitates setting up onboard LEDs. It uses I/O Port HAL module APIs to manipulate pins corresponding to LEDs on the board.

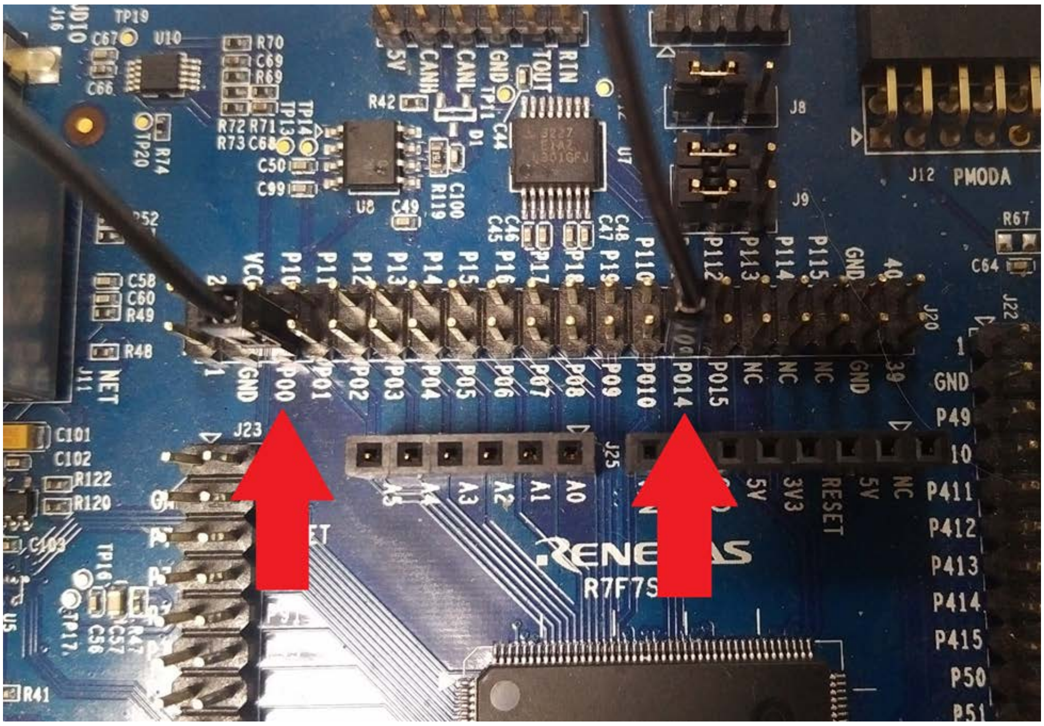

Figure 8 Connecting DAC output pin (P014) with ADC input pin (P00)

As the ELC requires peripherals such as the ADC, DAC, and timers to be able to demonstrate its capabilities, this application example uses quite a few of them, and each requires some configuration. The ELC HAL itself does not need to be manually configured. All the component settings are listed in the following tables.

Table 6 AGT HAL Module Configuration Settings for the Application Project

ISDE Property	Value Set
Name	g_agt
Channel	0
Mode	Periodic
Period Value	100
Period Unit	Milliseconds
Auto Start	False
Count Source	LOCO
AGT0 Output Enabled	False
AGTI0 Output Enabled	False
Output Inverted	False
Callback	NULL
Interrupt Priority	Disabled

Table 7 ADC HAL Module Configuration Settings for the Application Project

ISDE Property	Value Set
Name	g_adc
Unit	0
Resolution	12-Bit
Alignment	Right
Clear after read	On
Mode	Single Scan
Channel Scan Mask > Channel 0	Used in Normal/ Group A
Channel Scan Mask > Channels 1 – 27, Temperature Sensor, Voltage Sensor	Unused
Normal/Group A Trigger	ELC Event
Group B Trigger	ELC Event
Group Priority	Group A cannot interrupt Group B
Add/Average Count	Disabled
Addition/Averaging Mask > all channels	Disabled
Sample and Hold Mask > all channels	Disabled
Sample Hold States	24
Callback	adc_callback
Scan End Interrupt Priority	Priority 2
Scan End Group B Interrupt Priority	Disabled

Table 8 DAC HAL Module Configuration Settings for the Application Project

ISDE Property	Value Set
Name	g_dac
Channel	0
Synchronize with ADC	Disabled
Data Format	Right Justified
Output Amplifier	Disable

Table 9 External IRQ HAL Module Configuration Settings for the Application Project

ISDE Property	Value Set
Name	g_external_irq10_sw5
Channel	10
Trigger	Rising
Digital Filtering	Disabled
Digital Filtering Sample Clock	PCLK / 64
Interrupt enabled after initialization	True
Callback	irq10_sw5_callback
Interrupt Priority	Priority 2

Table 10 DTC HAL Module Configuration Settings for the Application Project

ISDE Property	Value Set
Name	g_transfer
Mode	Repeat
Transfer Size	2 Bytes
Destination Address Mode	Fixed
Source Address Mode	Fixed
Repeat Area	Destination
Interrupt Frequency	After each transfer
Destination Pointer	NULL
Source Pointer	NULL
Number of Transfers	255
Number of Blocks	0
Activation Source	Event ADC0 SCAN END
Auto Enable	False
Callback	NULL
ELC Software Event Interrupt Priority	Disabled

Table 11 GPT HAL Module Configuration Settings for the Application Project

ISDE Property	Value Set
Name	g_gpt
Channel	0
Mode	Periodic
Period Value	0xffff0000
Period Unit	Raw Counts
Duty Cycle Value	50
Duty Cycle Unit	Unit Raw Counts
Auto Start	True
GTIOCA Output Enabled	False
GTIOCA Stop Level	Pin Level Low
GTIOCB Output Enabled	False
GTIOCB Stop Level	Pin Level Low
Callback	NULL
Interrupt Priority	Disabled

8. Customizing the ELC HAL 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 ADC port pins to select the desired analog input. This can be done using the **Pins** tab in the configurator. The ELC application project used channel 0, which is connected to the onboard DAC unit 0 output (P014). To further modify the example, one can specify more/other ELC links between available events and peripherals. Changing the AGT timer period alters the ADC scan frequency, causing the LEDs to update more, or less, frequently. A similar effect can be achieved by modifying the GPT timer; the ADC scan frequency does not change, but the DAC output value changes, respectively.

9. Running the ELC HAL Module Application Project

To run the ELC HAL module application project and to see it executed on a target kit, you can simply import it into your ISDE, compile and run debug.

To implement the ELC Driver application in a new project, use the following steps for defining, configuring, auto-generating files, adding code, compiling and debugging on the target kit. Following these steps is a hands-on approach that will make the development process with SSP more practical, while just reading over this guide will tend 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 of the SSP User's Manual for a description of how to accomplish these steps.

To create and run the ELC HAL Module Application Project, simply follow these steps:

1. Create a new Renesas Synergy project for the S7G2 SK called ELC_HAL_MG_AP.
2. Select the **Threads** tab.
3. Add and configure the AGT HAL Module, according to section 7 above, to the HAL/Common thread.
4. Add and configure the ADC HAL Module, according to section 7 above, to the HAL/Common thread.
5. Add and configure the DAC HAL Module, according to section 7 above, to the HAL/Common thread.
6. Add and configure the DTC HAL Module, according to section 7 above, to the HAL/Common thread.
7. Add and configure the GPT HAL Module, according to section 7 above, to the HAL/Common thread.
8. Click on the **Generate Project Content** button.
9. Add the code from the supplied project file `elc_hal.c` or copy over the generated `elc_hal.c` file.
10. Connect to the host PC via a micro USB cable to J19 on SK-S7G2.
11. Start to debug the application.
12. The output can be viewed on the LEDs or a variable named `g_adc_result` can be watched in debug process.

10. ELC HAL 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 drivers. 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. ELC HAL Module Next Steps

After you have mastered setting up ELC links between the AGT and ADC, you may want to experiment with other events and peripherals supported by the ELC. Additionally, you may want to get to know more about components used in this example, such as the AGT, GPT, ADC, DAC, DTC and External IRQ HAL modules. These guides are available as described in the References section at the end of this document.

12. ELC HAL 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 r_elc module reference materials and resources are available on the Synergy Knowledge Base: https://en-us.knowledgebase.renesas.com/English_Content/Renesas_Synergy%E2%84%A2_Platform/Renesas_Synergy_Knowledge_Base/R_ELC_Module_Guide_Resources.

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

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Jun 8, 2017	-	Initial version
1.01	Aug 23, 2017	-	Update to Hardware and Software Resources Table
1.10	Nov 13, 2018	-	Table 9 callback changed

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)

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

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

Renesas Electronics Corporation
TOYOSU FORESIA, 3-2-24 Toyosu, Koto-ku, Tokyo 135-0061, Japan

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

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