Renesas Synergy™ Platform

IWDT HAL 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 Independent Watchdog Timer (IWDT) HAL module is a high-level API for watchdog timer applications and is implemented on r_iwdt. The Watchdog Timer uses the IWDT peripheral on the Synergy MCU. A user can configure the timeout period, stop control on sleep mode and standby mode, adjust the window period and generation of event on WDT underflow, or on a refresh outside permitted window.

Contents

1. IWDT HAL Module Features

The Independent Watchdog Timer (IWDT) HAL module key features includes:

- One of two possible event triggers for IWDT underflows or refreshing outside of the permitted window:
  - device reset, or
  - NMI generation.
- Internal Watchdog timer peripheral clock source to improve safety
- Automatic hardware configuration after reset

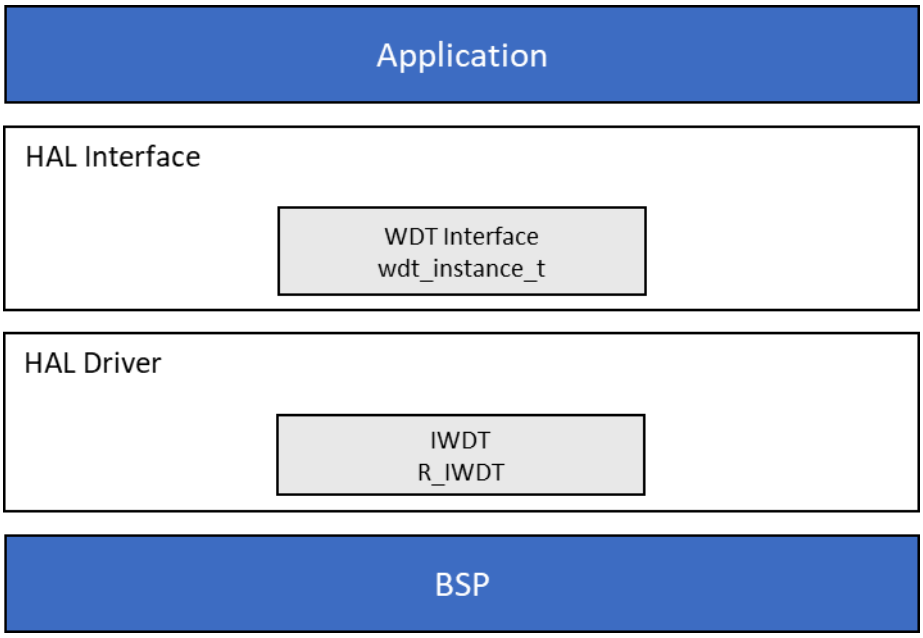

Figure 1. IWDT HAL Module Organization, Options, & Stack Implementations

2. IWDT HAL Module APIs Overview

The IWDT HAL module defines APIs to open, refresh, read, and get status. The following table summarizes the available APIs and includes an example of each API call, along with a brief description. A table of status return values follows the API summary table.

Table 1. IWDT HAL Module API Summary

Function Name	Example API Call and Description
cfgGet	`g_iwdt0.p_api->cfgGet(g_iwdt0.p_ctrl, g_iwdt0.p_cfg);` Initialize the IWDT in register start mode. In auto-start mode with NMI output it registers the NMI callback.
open	`g_iwdt0.p_api->open(g_iwdt0.p_ctrl, g_iwdt0.p_cfg);` Initialize the IWDT in register start mode. In auto-start mode with NMI output it registers the NMI callback.
refresh	`g_iwdt0.p_api->refresh(g_iwdt0.p_ctrl);` Refresh the watchdog timer.
statusGet	`g_iwdt0.p_api->statusGet( g_iwdt0.p_ctrl, &status);` Read the status of the IWDT.
statusClear	`g_iwdt0.p_api->statusClear( g_iwdt0.p_ctrl, clear);` Clear the status flags of the IWDT.
counterGet	`g_iwdt0.p_api->counterGet(g_iwdt0.p_ctrl, &counter);` Read the current IWDT counter value.
timeoutGet	`g_iwdt0.p_api->timeoutGet(g_iwdt0.p_ctrl, &timeout);` Read the watchdog timeout values.

Function Name	Example API Call and Description
versionGet	`g_iwdt0.p_api->versionGet(&version);` Retrieve the API version using 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 executed.
SSP_ERR_ASSERTION	Null Pointer(s).
SSP_ERR_INVALID_ARGUMENT	One or more configuration options is invalid.
SSP_ERR_INVALID_MODE	An attempt to open the WDT in register-start mode when OFS0 register is configured for auto-start mode. Or, to open the WDT in auto-start mode when OSF0 is configured for register start mode.
SSP_ERR_ABORTED	Invalid clock divider for this watchdog

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. IWDT HAL Module Operational Overview

The IWDT HAL module configures the IWDT Interface. When the IWDT underflows or is refreshed outside of the permitted refresh window, one of the following events can occur:

- Resetting of the device
- Generation of an NMI

The following figure shows an example of the operation of the IWDT. When refreshed in the valid refresh period of the counter the timer count value is reset. If the count underflows or IWDT refresh occurs outside of the valid refresh period, the IWDT resets the device or generates an NMI.

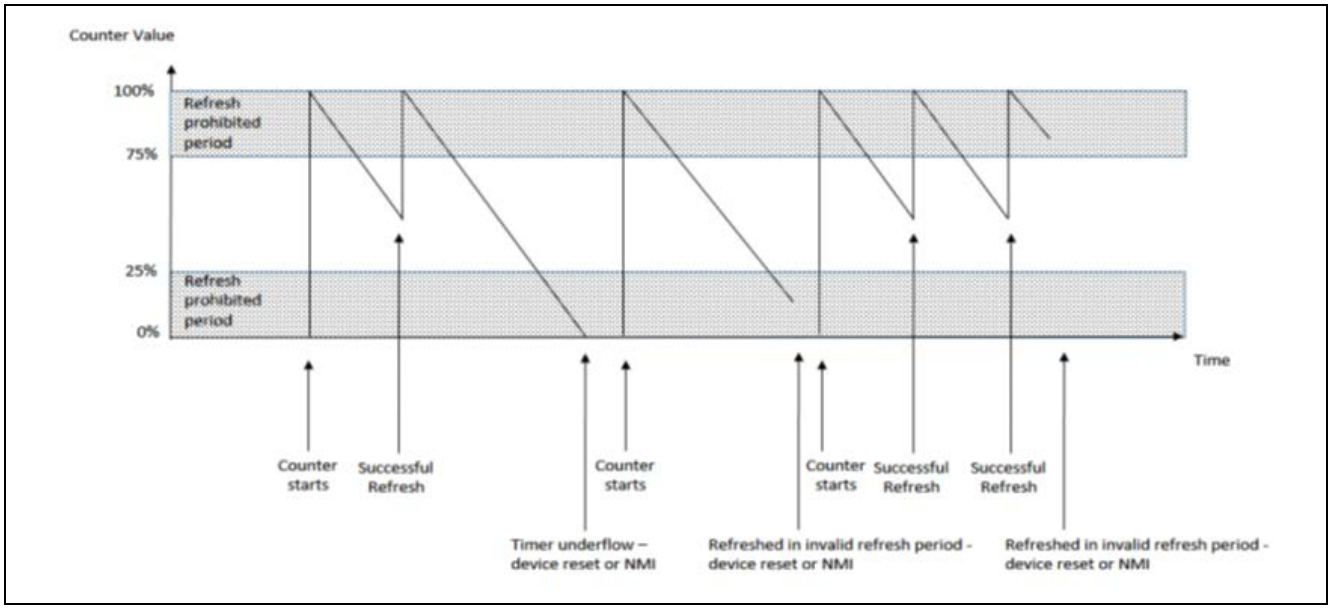

Figure 2. IWDT HAL Module Operational Diagram

All series of Synergy microcontrollers have an Option-Setting Memory which can be used to set the operating state of peripherals after a reset. The OFS can be used to set the state of the IWDT, WDT, LVD and CGC HOCO.

The following table details how the IWDT parameters are configured by the OFS registers:

Table 3. WDT Parameters

Control	Description
IWDT Start Mode Select	Automatically starts the IWDT after a Reset in enabled.
IWDT Timeout Period	Specifies the IWDT timeout (number of clock cycles). 128 cycles 512 cycles 1024 cycles 2048 cycles
IWDT-Clock Frequency Division Ratio (The IWDT is clocked from PCLKB)	IWDTCLK / 1 IWDTCLK / 16 IWDTCLK / 32 IWDTCLK / 64 IWDTCLK / 128 IWDTCLK / 256
IWDT Window End Position	0% (no window end position set) 25% 50% 75% 100%
IWDT Window Start Position	0% 25% 50% 75% 100% (no window start position set)
IWDT Reset Interrupt Request	The IWDT can either generate an Interrupt Signal or a Reset signal.
IWDT Stop Control	The IWDT can continue to count or Stop counting in Sleep, Snooze, Software Standby, or Deep Software Standby mode

Note: For details on the contents of the OFS0 register see the Synergy MCU hardware manual.

The following figures show the OFS register values are set via the properties dialog of the BSP tab of Synergy Configuration editor:

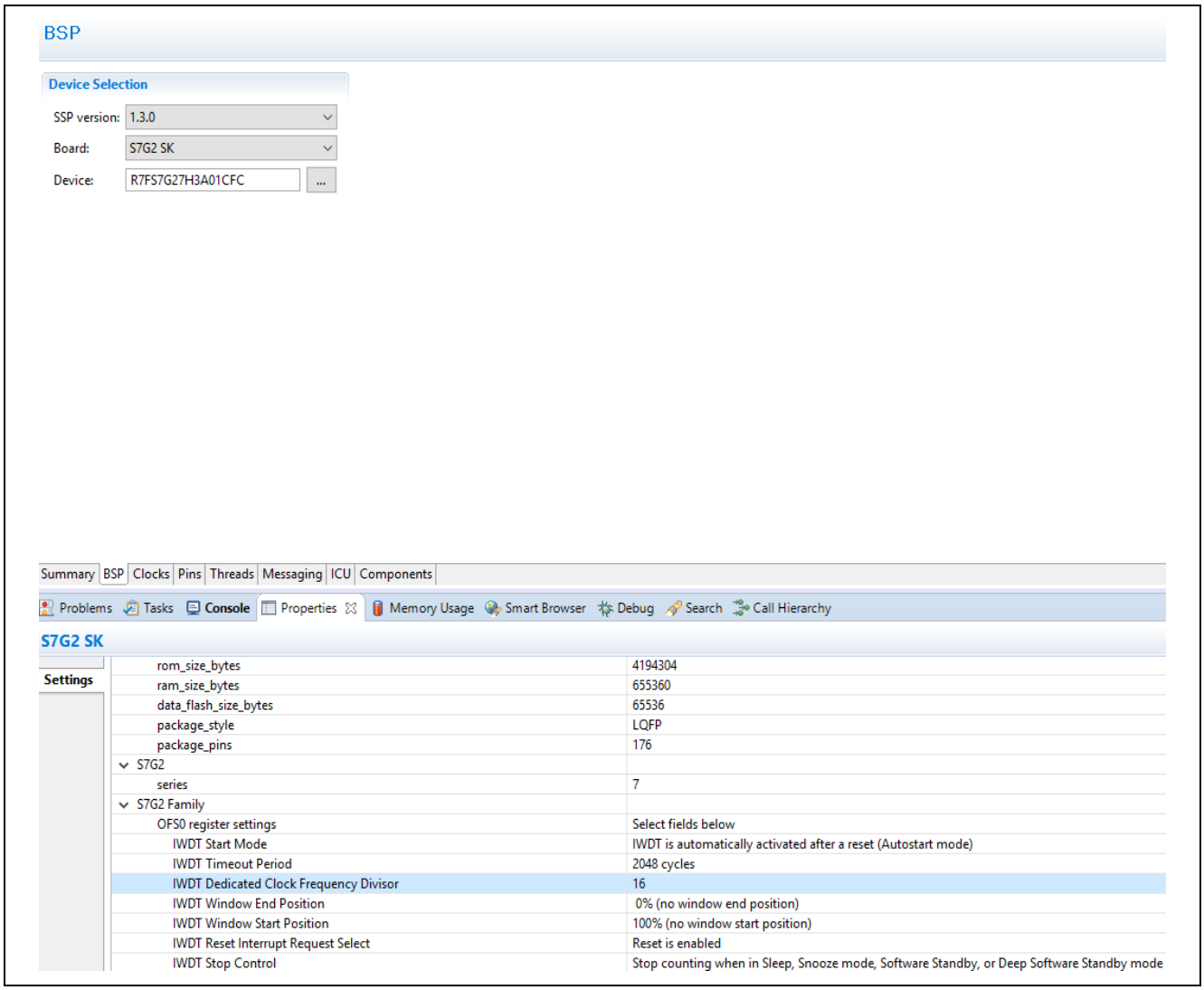

Figure 3. OFS register values

3.1 IWDT HAL Module Important Operational Notes and Limitations

3.1.1 IWDT HAL Module Period Calculation

The IWDT operates from IWDTCLK. Assuming largest parameters for the IWDT, the time from the last refresh to device reset or NMI generation is just under 35 seconds as the following table indicates.

Parameter	Equal to
IWDTCLK	15 kHz
Clock division ratio	IWDTCLK/256
Timeout period	2048 cycles
IWDT clock frequency	15 kHz / 256 = 58.59 Hz
Cycle time	1 / 58.59 Hz = 17.07 ms
Timeout	17.07 ms x 2048 cycles = 34.95 seconds

3.1.2 Triggering DMAC/DTC with the IWDT HAL Module

To trigger a transfer of data using the DMAC or DTC peripheral when the IWDT counter underflows or when a refresh is attempted outside of the valid refresh period, configure the IWDT to generate an NMI and configure the DMAC/DTC transfer with activation source set to ELC_EVENT_IWDT_UNDERFLOW. See the associated User Guide (DMAC, DTC) for further information.

3.1.3 Triggering Event Link Controller Events with the IWDT HAL Module

The IWDT can trigger the start of another peripheral using the Event Link Controller (ELC). Refer to the ELC User Guide for a complete list of available peripherals.

3.2 IWDT HAL Module Limitations

- When using a JLink debugger, the IWDT counter does not count and will not reset the device or generate an NMI.
- When there is no active task ready to run, ThreadX puts the MCU into sleep mode. If the IWDT is configured to stop the counter in low-power mode, your application must then restart the timer when used with ThreadX RTOS.

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

4. Including the IWDT HAL Module in an Application

Use the following information to include the IWDT 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 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 IWDT Driver to an application, simply add it to a thread using the stacks selection sequence provided in the following table. (The default name for the Transfer Driver is g_iwdt0. This name can be changed in the associated Properties window.)

Table 4. IWDT Driver Selection Sequence

Resource	ISDE Tab	Stacks Selection Sequence
g_wdt0 IWDT HAL on r_iwdt	Threads	New Stack> Driver> Monitoring> IWDT HAL on r_iwdt

When the IWDT HAL module on r_iwdt is added to the thread stack as shown in the following figure, the configurator automatically adds any needed lower-level modules. Any drivers that need additional configuration information are box text highlighted in Red. Modules with a Gray band are individual modules that stand alone.

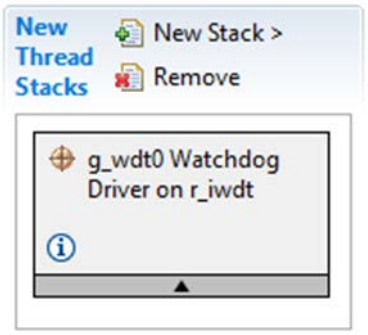

Figure 4. IWDT HAL Module Stack

5. Configuring the IWDT HAL Module

The IWDT 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 properties that can be changed without causing conflicts are available for modification. Other properties

which are ‘locked’ are unavailable for changes, and these 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 in 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 in the ISDE 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 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 helps to orient you and can be a useful ‘hands-on’ approach to learning the ins and outs of developing with SSP.

Table 5. Configuration Settings for the IWDT HAL Module on r_iwdt

ISDE Property	Value	Description
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Include parameter checking code
Name	g_wdt0	Module Name
NMI Callback	NULL	Callback. A user callback function can be registered in the open API. If this callback function is provided, it will be called from the interrupt service routine (ISR) each time the IRQn triggers. Since the callback is called from an ISR, care should be taken not to use blocking calls or lengthy processing. Spending excessive time in an ISR can affect the responsiveness of the system.

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

5.1 Configure Option Function Select Register 0 (OFS0)

All series of Synergy microcontrollers have an Option-Setting Memory which can be used to set the operating state of peripherals after a reset. The OFS can be used to set the state of the IWDT, WDT, LVD and CGC HOCO. See the description in the operational overview section earlier in this document.

5.2 Configuring the Interrupts for the IWDT HAL Module

Use the ISDE to configure the IWDT interrupts in the same way as configuring the other options for the IWDT module. If the IWDT is configured to generate an NMI interrupt on underflow or invalid refresh the interrupt must be enabled in the BSP.

Note: To enable interrupts, set the priority of the IWDT > IWDT NMIUNDF N. This sets BSP_IRQ_CFG_IWDT_UNDERFLOW in ssp_cfg/bsp/bsp_irq_cfg.h to the priority level selected.

When the IWDT NMIUNDF N interrupt is enabled in the BSP, the corresponding ISR will be defined. The ISR will call a user callback function if one was registered in the open API.

5.3 IWDT HAL Module Clock Configuration

The IWDT has its own dedicated clock operating at a set frequency which cannot be modified.

5.4 IWDT HAL Module Pin Configuration

The IWDT does not require pins for their operation.

6. Using the IWDT HAL Module in an Application

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

1. Register the IWDT NMI callback using the `open` API.
2. Read the configuration of the IWDT HAL module using the `cfgGet` API.
3. Refresh the independent watchdog timer using the `refresh` API.
4. Read the IWDT status flags using the `statusGet` API.
5. Clear the IWDT Status and error flags using the `statusClear` API.
6. Read the current count value of the IWDT using the `counterGet` API.
7. Read the timeout values of the IWDT HAL module using the `timeoutGet` API.

The following diagram shows common steps in a typical operational flow:

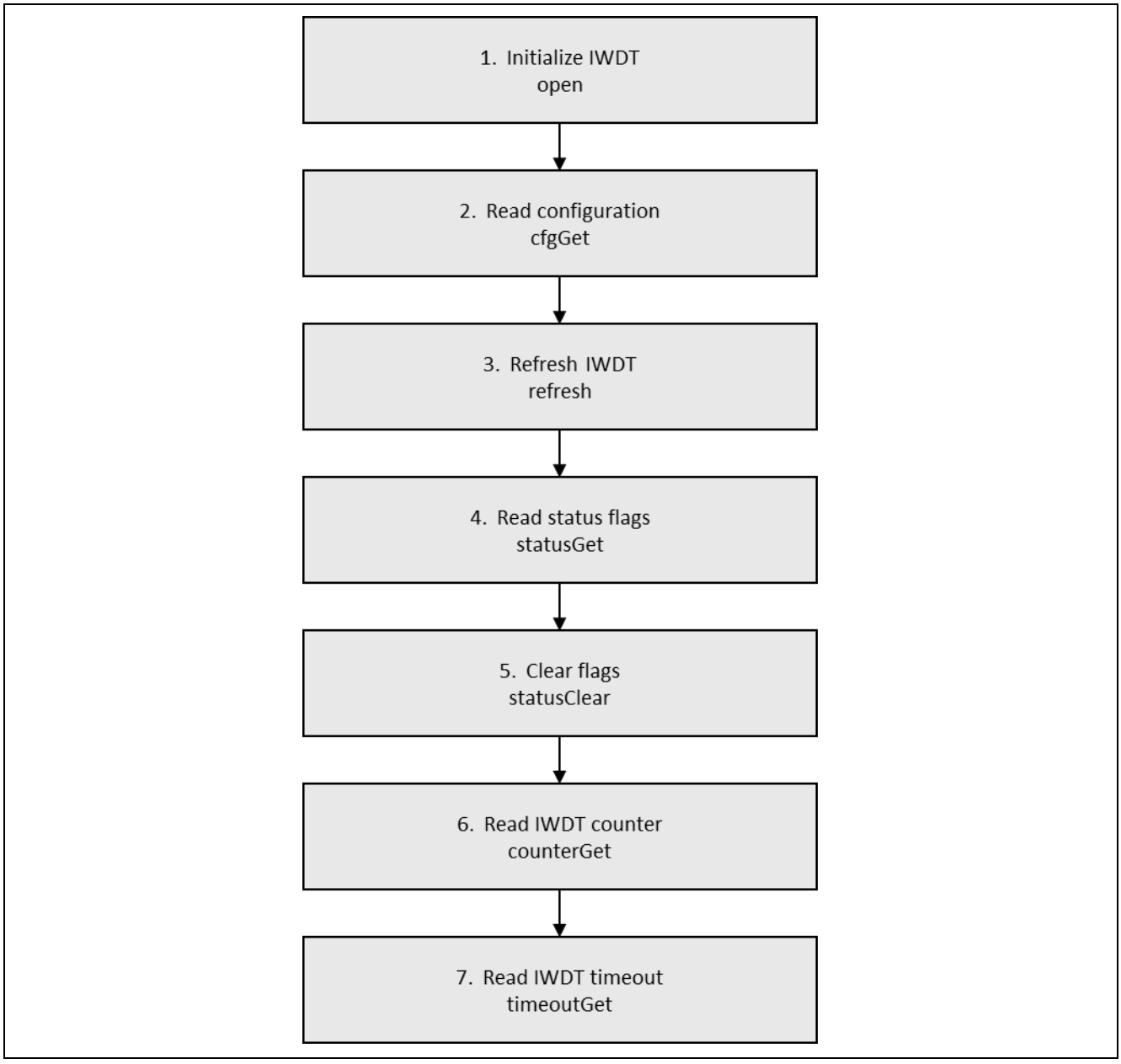

Figure 5. Flow Diagram of a Typical WDT HAL Module Application

7. The IWDT HAL Module Application Project

The application project associated with this module guide demonstrates the common steps in an example application. You may want to import and open the application project within ISDE and view the configuration settings for the IWDT HAL module. You can also read over the code (in IWDT_HAL_MG.c) which is used to demonstrates the IWDT HAL module APIs in a complete design.

The application project configures the IWDT to run in auto-start mode with (IWDTCLK (15KHz)/16) and timeout of 2048 cycles, which amounts to an approximate timeout of 2.18 seconds. The GPT timer0 is configured to refresh IWDT periodically and toggle the Green LED as an activity indication. SW5 is configured to disable IWDT refresh to generate an IWDT reset signal. On reset, the application identifies the cause of the last reset, and if the reset is caused by IWDT, the RED led is set to inform the user that IWDT reset has occurred. Pressing SW4 clears this state.

Table 6. Software and Hardware Resources Used by the Application Project

Resource	Revision	Description
e2 studio	5.4.0 or later	Integrated Solution Development Environment
SSP	1.3.0 or later	Synergy Software Platform
IAR EW for Renesas Synergy	7.71.3	IAR Embedded Workbench® for Renesas Synergy™
SSC	5.4.0 or later	Synergy Standalone Configurator
SK-S7G2	v3.0 to v3.2	Starter Kit

The following diagram shows a simple flow diagram of the application project:

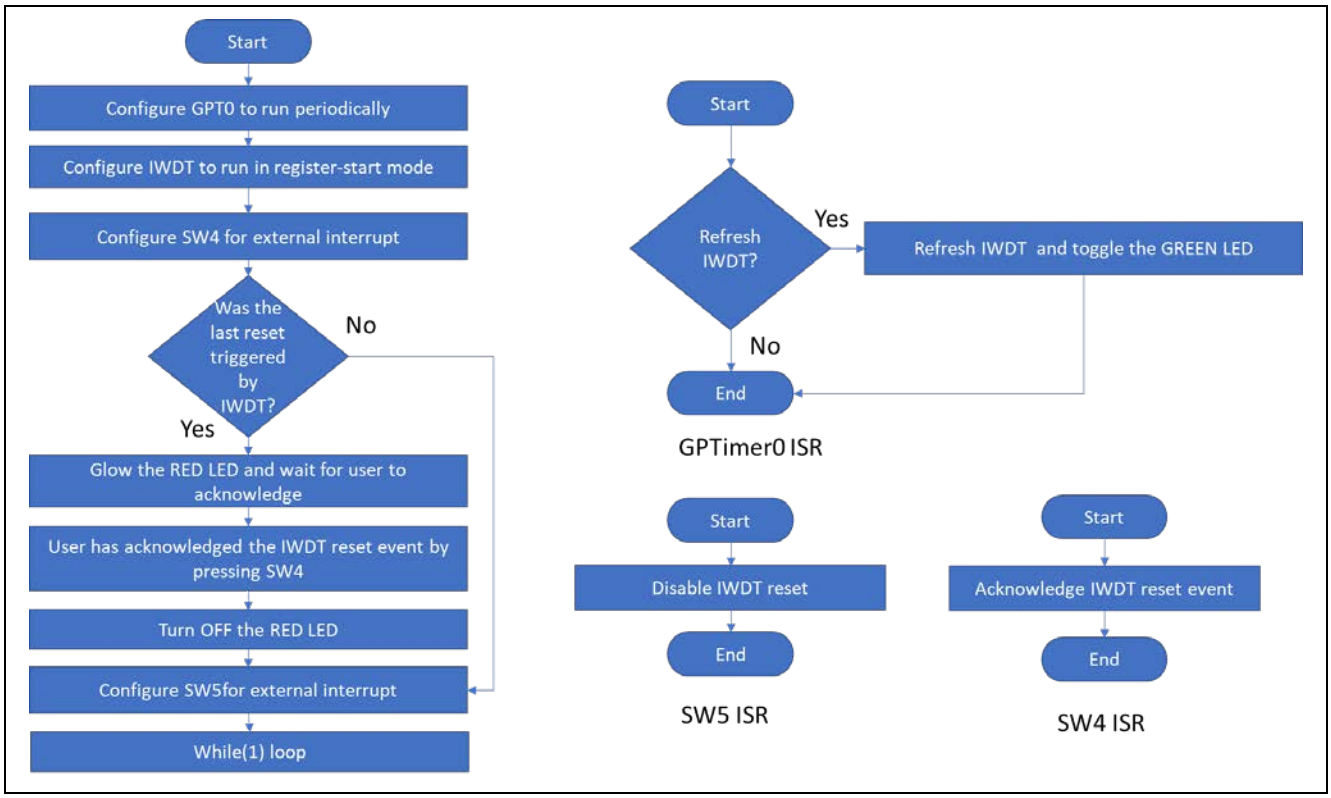

Figure 6. IWDT HAL Module Application Project Flow Diagram

The IWDT_HAL_MG_AP.c file resides in the project once it has been imported into the ISDE. You can open this file within the ISDE and follow along with the description provided to help identify key uses of APIs.

To refresh IWDT, GPT0 is set to periodic mode to refresh WDT using a 200 ms interrupt subroutine.

On startup, hardware auto-starts the IWDT with the configuration set in the synergy configurator->BSP. The application configures the IWDT interface and GPT0 by making open calls, g_timer0.p_api->open and to

g_wdt.p_api->open, respectively. The IWDT is refreshed by calls to the g_iwdt.p_api->refresh, in the callback function of the GPT0 timer.

When switch SW5 is pressed, the g_iwdt_refresh_flag is cleared, the GPT0 interrupt subroutine would skip over the IWDT refresh, causing IWDT underflow. On the next reset, application checks for the cause of the RESET. If the IWDT bit of RSTSR1_b (RSTSR1 SFR) is set, program turns on the red LED and waits in the loop for user to press switch SW4. The switch, SW5, would also be disabled until the user acknowledges the IWDT reset event by pressing SW4.

Note: To support required operations and physical properties of the target board and MCU, a few key properties are configured in this application project. The following table lists properties with values set for this specific project. You can open the project and view these settings in the property window as a hands-on exercise.

Table 7. IWDT HAL Module Configuration Settings for the Application Project (BSP settings)

ISDE Property	Value Set
IWDT Start Mode Select	IWDT is automatically activated after a reset (auto-start mode)
IWDT Timeout Period	2048 cycles
IWDT Clock Frequency Division Ratio	16
IWDT Window End Position	0% (no window end position)
IWDT Window Start Position	100%(no window start position)
IWDT Reset Interrupt Request	NReset is enabled
IWDT Stop Control	Stop counting when in Sleep, Snooze mode, Software Standby, or Deep Software Standby mode

Table 8. GPT0 Configuration Settings for the Application Project in Threads

ISDE Property	Value Set
Parameter Checking	Default (BSP)
Name	g_timer0
Channel	0
Mode	Periodic
Period Value	200
Period Unit	Milliseconds
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	gpt_callback
Interrupt Priority	Priority 4 (CM4: valid, CM0+: invalid)

8. Customizing the IWDT HAL Module for a Target Application

Some configuration settings are normally be changed by the developer from those shown in the application project, such as changing the IWDT configuration in BSP. For example, the user can change the RESET interrupt request from RESET action to NMI call back and define a callback in r_iwdt stack.

9. Running the IWDT HAL Module Application Project

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

Once successfully imported and compiled, run the IWDT application project by using the following steps:

1. Connect to the host PC using the USB cable (use J19 DEBUG_USB connector).
2. Start to debug the application.
3. Stop the debug operation. Restart the board by shorting RESET jumper J2, to test the IWDT application without J-Link.
4. The Green LED (LED1 on SK-S7G2) starts to blink at the rate of 200 ms to indicate an IWDT refresh.
5. Press switch SW5. Observe the green LED stops blinking, indicating the IWDT is not getting refreshed.
6. After approximately 2.218 seconds the IWDT RESETs the device (indicated by the Red LED (LED2 on SK- S7G2) glowing along with green flashing LED.
7. Press SW4 to acknowledge the IWDT RESET, this turns OFF the Red LED.

Note**:** Since the project runs without J-Link support, SEMI-HOSTING is not enabled. However, the Amber LED (LED3 on SK-S7G2) lights up on detection of any error in IWDT, GPT, and external interrupts API calls. In such a case, users are advised to connect J-Link and do a step-by-step debug operation to explore the bug.

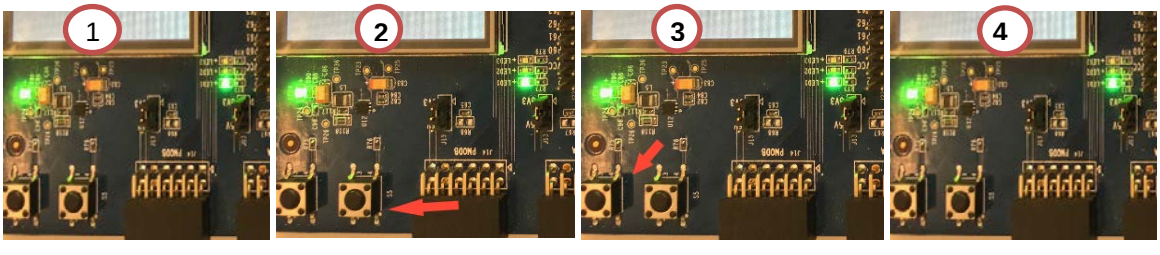

Figure 7. IWDT Application stages from IWDT Driver Application Project

Description of IWDT Application stages from Left to Right (see figure)

1. Green LED blinking
2. On press of SW5 (red arrow); after a RESET, Red LED (bottom right of image) glowing
3. On press of SW4 (bottom left of image). Red LED is turned OFF.
4. Amber LED (lower left corner of image) glowing on detection of any error caused by API calls.

10. IWDT 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 low-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. IWDT HAL Module Next Steps

After you have mastered a simple IWDT module project, you may want to review a more complex example. Other application projects and application notes that demonstrate IWDT HAL use can be found as described in the References section at the end of this document.

12. IWDT 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_iwdt module reference materials and resources are available on the Synergy Knowledge Base: https://en-support.renesas.com/search/r_iwdt%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	Aug.24.17	—	Initial release
1.01	Dec.01.17	—	Updated software versions and made minor corrections
1.02	Jan.07.19	—	Updates to section 7 and 9 procedures and minor changes throughout.

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/.