Renesas Synergy™ Platform

WDT 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 Reference section at the end of this document), and will be valuable resources for creating more complex designs.

The WDT (Watchdog Timer) HAL module is a high-level API for WDT applications and is implemented on r_wdt. The WDT HAL module uses the WDT peripheral on the Synergy MCU. You can configure the timeout period, start/stop methods, the window period and the generation of an event on WDT underflow or on refresh outside the permitted window.

Target Device

Renesas Synergy™ Starter Kit S7G2 (SK-S7G2) and the S7G2 MCU Group

Contents

1. WDT HAL Module Features

The WDT HAL module has the following key features:

- When the WDT underflows or is refreshed outside of the permitted refresh window, one of the following events can occur:
  - Resetting of the device
  - Generation of a NMI
- Supports the Watchdog Timer (WDT) peripheral, which uses a peripheral clock (PCLKB for SK-S7G2).
- The clock division ratio and cycles can be configured to produce the desired timeout of WDT.
- The WDT can be configured in Register-start mode through the WDT registers or in Auto-start mode.
- WDT can be configured to continue to run in Sleep mode.

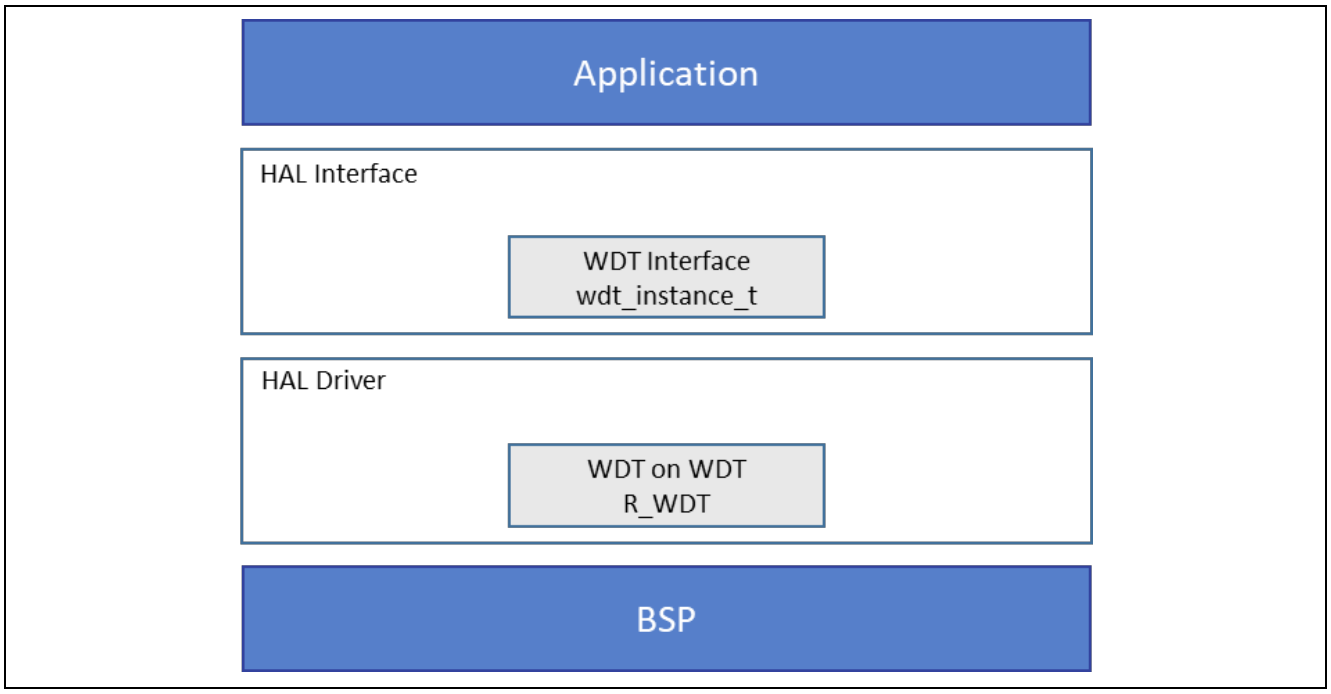

Figure 1. WDT HAL Module Block Diagram

2. WDT HAL Module APIs Overview

The WDT HAL module defines APIs for opening, closing, reading, writing, and other useful functions. A complete list of the available APIs, an example API call, and a short description of each can be found in the following table.

Table 1. WDT HAL Module API Summary

Function Name	Example API Call and Description
.cfgGet	`g_wdt0.p_api->cfgGet(g_wdt0.p_ctrl, g_wdt0.p_cfg);` Initialize the WDT in register start mode. In auto-start mode with NMI output it registers the NMI callback.
.open	`g_wdt0.p_api->open(g_wdt0.p_ctrl, g_wdt0.p_cfg);` Initialize the WDT in register start mode. In auto-start mode with NMI output it registers the NMI callback.
.refresh	`g_wdt0.p_api->refresh(g_wdt0.p_ctrl);` Refresh the watchdog timer.
.statusGet	`g_wdt0.p_api->statusGet( g_wdt0.p_ctrl, &status);` Read the status of the WDT.
.statusClear	`g_wdt0.p_api->statusClear( g_wdt0.p_ctrl, clear);` Clear the status flags of the WDT.
.counterGet	`g_wdt0.p_api->counterGet(g_wdt0.p_ctrl, &counter);`

Function Name	Example API Call and Description
	Read the current WDT counter value.
.timeoutGet	`g_wdt0.p_api->timeoutGet(g_wdt0.p_ctrl, &timeout);` Read the watchdog timeout values.
.versionGet	`g_wdt0.p_api->versionGet(&version);` Retrieve the API version using the version pointer.

Note: For more complete descriptions of 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.

3. WDT HAL Module Operational Overview

Consider the following points while configuring the Watchdog Timer (WDT) module for synergy MCUs:

- The WDT can be configured in Register-start mode, which allows WDT to be started from the application.
- The WDT can also be configured in Auto-start mode, where the hardware uses parameters stored in Option Function Select Register 0 (OFS0) to automatically configure, WDT after reset.

3.1 WDT HAL Module Important Operational Notes and Limitations

The WDT HAL module configures the WDT Interface. When the WDT 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 WDT. When refreshed in the valid refresh period of the counter the timer count value is reset. If the count can underflow or refresh occurs outside of the valid refresh period, the WDT resets the device or generates a NMI.

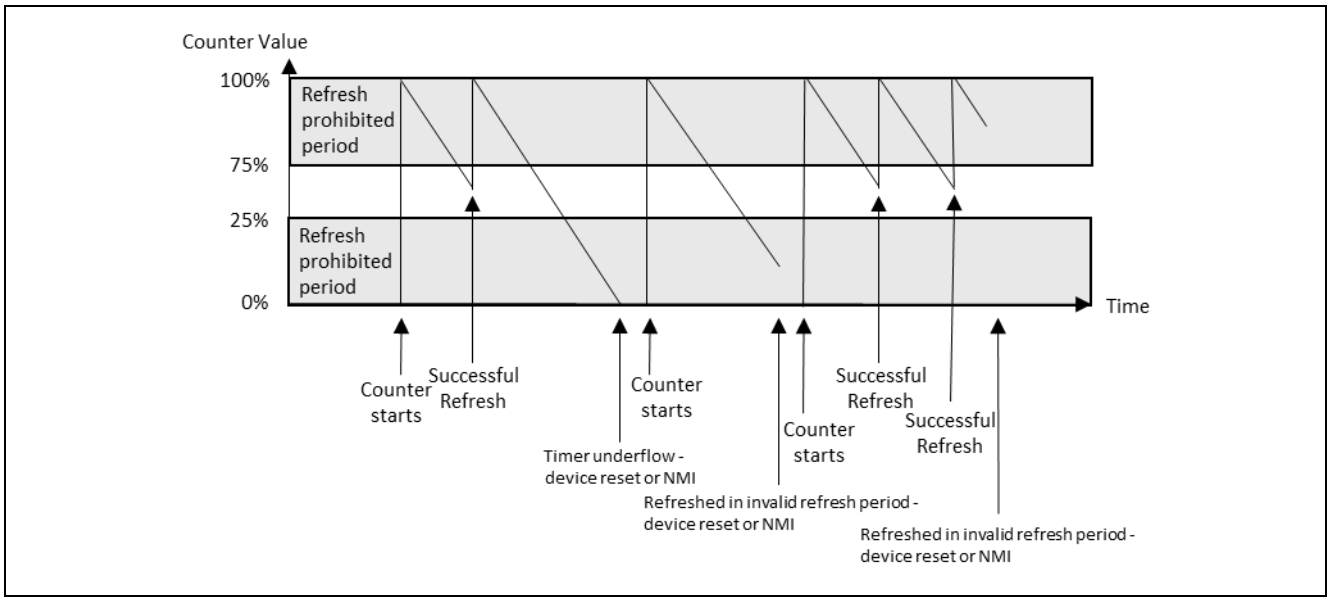

Figure 2. WDT HAL Module Operational Diagram

The WDT can be configured in Register-start mode through the WDT registers. The WDT can also be configured by hardware automatically after reset using parameters stored in Option Function Select Register 0 (OFS0) as shown in the following table.

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 which parameters of the WDT can be configured by the OFS registers.

Table 2. WDT Parameters

Control	Description
WDT Start Mode Select	Automatically starts the WDT after a Reset in enabled.
WDT Timeout Period	Specifies the WDT timeout (number of clock cycles). 128 cycles 512 cycles 1024 cycles 2048 cycles
WDT Clock Frequency Division Ratio (The WDT is clocked from PCLKB)	PCLKB / 1 PCLKB / 64 PCLKB / 128 PCLKB / 512 PCLKB / 2048 PCLKB / 8192
WDT Window End Position	25% 50% 75% 100% (no window end position set)
WDT Window Start Position	25% 50% 75% 100% (no window start position set)
WDT Reset Interrupt Request	The WDT can either generate an Interrupt Signal or a Reset signal.
WDT Stop Control	The WDT can continue to count or Stop counting in Low Power Mode.

Note: For further information on the contents of the OFS0 register see the Hardware Synergy MCU Manual.

The OFS register values are set by the **Properties** dialog of the **BSP** tab of **Synergy Configuration** editor as shown in the following figure.

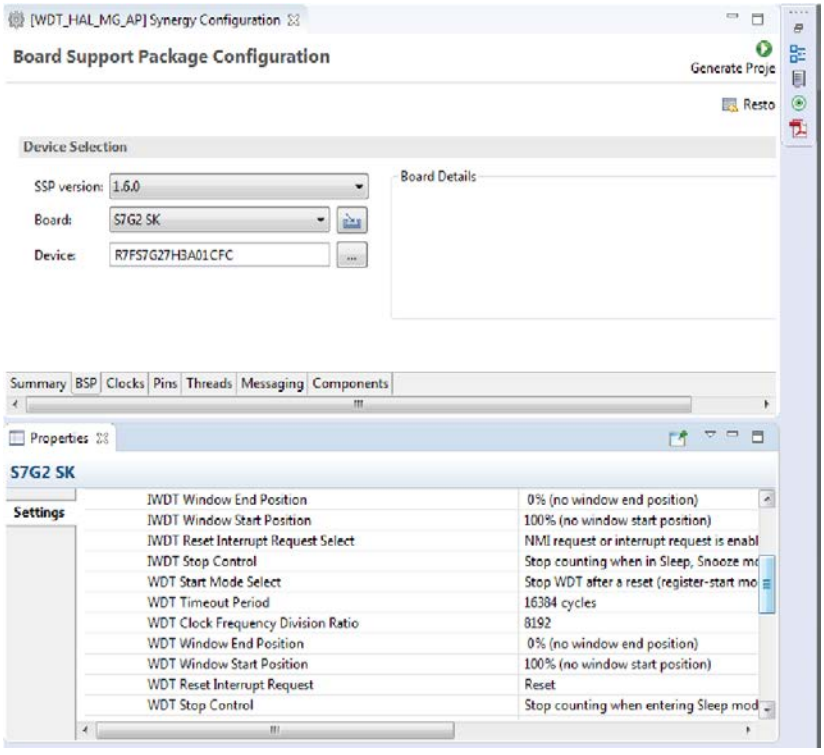

Figure 3. Setting OFS Register Values in Synergy Configurator

3.2 WDT HAL Module Important Operational Notes and Limitations

3.2.1 Period Calculation

The WDT operates from PCLKB. Assuming largest parameters for the WDT and a PCLKB of 60 MHz, the time from the last refresh to device reset or NMI generation will be just over 2.2 seconds as detailed below.

- PLCKB = 60 MHz
- Clock division ratio = PCLKB/8192
- Timeout period = 16384 cycles
- WDT clock frequency = 60 MHz/8192 = 7.324 kHz
- Cycle time = 1/7.324 kHz = 136.53 us
- Timeout = 136.53 µs x 16384 cycles = 2.23 seconds

3.2.2 Triggering DMAC/DTC

To trigger a transfer of data using the DMAC or DTC peripheral when the WDT counter underflows or when a refresh is attempted outside of the valid refresh period, configure the WDT to generate a NMI and configure the DMAC/DTC transfer with `activation_source` set to `ELC_EVENT_WDT_UNDERFLOW`. See the associated User's Guide (DMAC, DTC) for further information.

3.2.3 Triggering Event Link Controller Events

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

3.2.4 WDT HAL Module Limitations

- When using a J-Link® Debug Probe, the WDT counter does not count and therefore will not reset the device or generate a NMI.
- When there is no active task ready to run, ThreadX® puts the MCU into sleep mode. If the WDT is configured to stop the counter in low power mode, then your application must restart the timer when used with the ThreadX RTOS.
- Refer to the latest *SSP Release Note* for any additional operational limitations for this module.

4. Including the WDT HAL Module in an Application

This section describes how to include the WDT HAL 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 WDT 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_wdt0`. This name can be changed in the associated **Properties** window.

Table 3. WDT Driver Selection Sequence

Resource	ISDE Tab	Stacks Selection Sequence
`g_wdt0` Watchdog Driver on `r_wdt`	Threads	New Stack>Driver>Monitoring>Watchdog Driver on `r_wdt`

When the WDT HAL module on `r_wdt` 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 will be box text highlighted in Red. Modules with a Gray band are individual modules that stand alone.

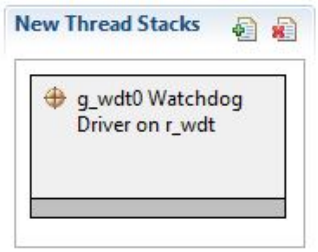

Figure 4. WDT HAL Module Stack

5. Configuring the WDT HAL Module

The WDT HAL module must be configured by you for the desired operation. The SSP configuration window will automatically identify (by highlighting the block in red) any required configuration selections, such as interrupts or operating modes, that must be configured for lower-level modules for successful operation. Only those properties that can be changed without causing conflicts are available for modification. Other properties are **locked** and are not available for changes, and are identified with a lock icon for the **locked** property in the **Properties** window in the 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 your 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. Note that the interrupt priorities listed in the Properties window in the ISDE will include 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 configuration table settings given in the table below. 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 WDT HAL Module on r_wdt

ISDE Property	Value	Description
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Enables or disables the parameter checking.
Name	`g_wdt0`	Module Name
Start Mode	Register, Auto Default: Register	Configures the start mode as register start or auto-start.
Start Watchdog After Configuration	True, False Default: True	Controls whether WDT is started during initialization
Timeout	1024 cycles, 4096 cycles, 8192 cycles, 16384 cycles Default: 16384 cycles	WDT timeout period.
Clock Division Ratio	PCLK/4, PCLK/64, PCLK/128, PCLK/512, PCLK/2048, PCLK/8192 Default: PCLK/8192	WDT clock divider.

ISDE Property	Value	Description
Window Start Position	100% (Window Position Not Specified), 75%, 50%, 25% Default: 100% (Window Position Not Specified)	Permitted refresh period start position.
Window End Position	0% (Window Position Not Specified), 25%, 50%, 75% Default: 0% (Window Position Not Specified)	Permitted refresh period end position.
Reset Control	Reset Output, NMI Generated Default: Reset Output	Select whether WDT should reset the MCU or generate an NMI.
Stop Control	WDT Count Enabled in Low Power Mode, WDT Count Disabled in Low Power Mode Default: WDT Count Disabled in Low Power Mode	Select whether the WDT should stop counting in low power modes.
NMI Callback	NULL	Callback. A user callback function can be registered in open. If this callback function is provided, it will be called from the interrupt service routine (ISR) each time the IRQn triggers. **Warning:** 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 S7G2 Synergy 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 that 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 WDT HAL Module

Configure the WDT interrupts in the same way as configuring the other options for the WDT module. If the WDT is configured to generate a NMI interrupt on underflow or invalid refresh, the interrupt must be **enabled** in the **BSP**.

To enable interrupts, set the priority of the `CWDT > CWDT NMIUNDF N`. This sets `BSP_IRQ_CFG_WDT_UNDERFLOW` in `ssp_cfg/bsp/bsp_irq_cfg.h` to the priority level selected.

When the `CWDT 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 WDT HAL Module Clock Configuration

The WDT clock is based on the PCLKB frequency. You can set the PCLKB frequency using the clock configurator in the ISDE or using the CGC Interface at run-time. The maximum timeout period with PCLKB running at 60 MHz is approximately 2.2 seconds.

5.4 WDT HAL Module Pin Configuration

The WDT does not require pins for their operation.

6. Using the WDT HAL Module in an Application

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

1. Initialize the WDT HAL module using the open API.
2. Refresh the WDT HAL module as needed using the refresh API.

These common steps are illustrated in a typical operational flow diagram in the following figure:

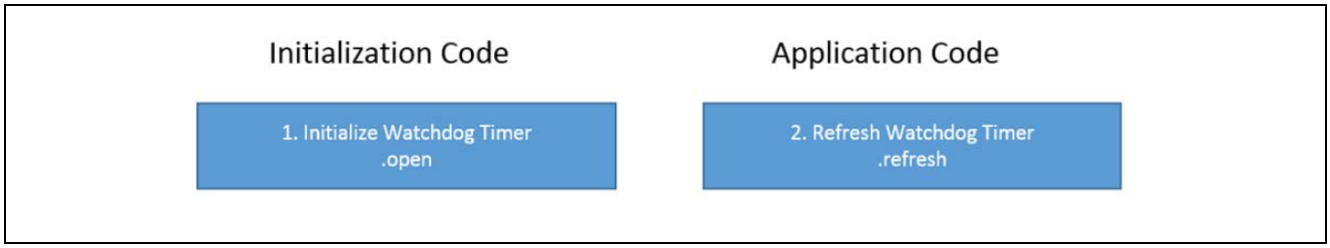

Figure 5. WDT HAL Module Application

7. WDT HAL Module Application Project

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

The application project configures WDT to run in register-start mode with PCLB/8192 and timeout of 16,384 cycles, that amounts to approximate timeout of 2.23 seconds. GPT `timer0` is configured to refresh the WDT periodically and toggle the Green LED as an indication. SW5 is configured to disable WDT refresh to generate a WDT reset signal. On reset, the application will identify the cause of the last reset, and if the reset is caused by the WDT, the RED led will be set to inform you that the WDT reset has occurred, Pressing SW4 will clear this state.

Table 5. Software and Hardware Resources Used by the Application Project

Resource	Revision	Description
e² studio	v7.3.0 or later	Integrated Solution Development Environment
SSP	v1.6.0 or later	Synergy Software Platform
IAR EW for Renesas Synergy	v8.23.3 or later	IAR Embedded Workbench® for Renesas Synergy
SSC	v7.3.0 or later	Synergy Standalone Configurator
SK-S7G2	v3.0 to v3.3	Starter Kit

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

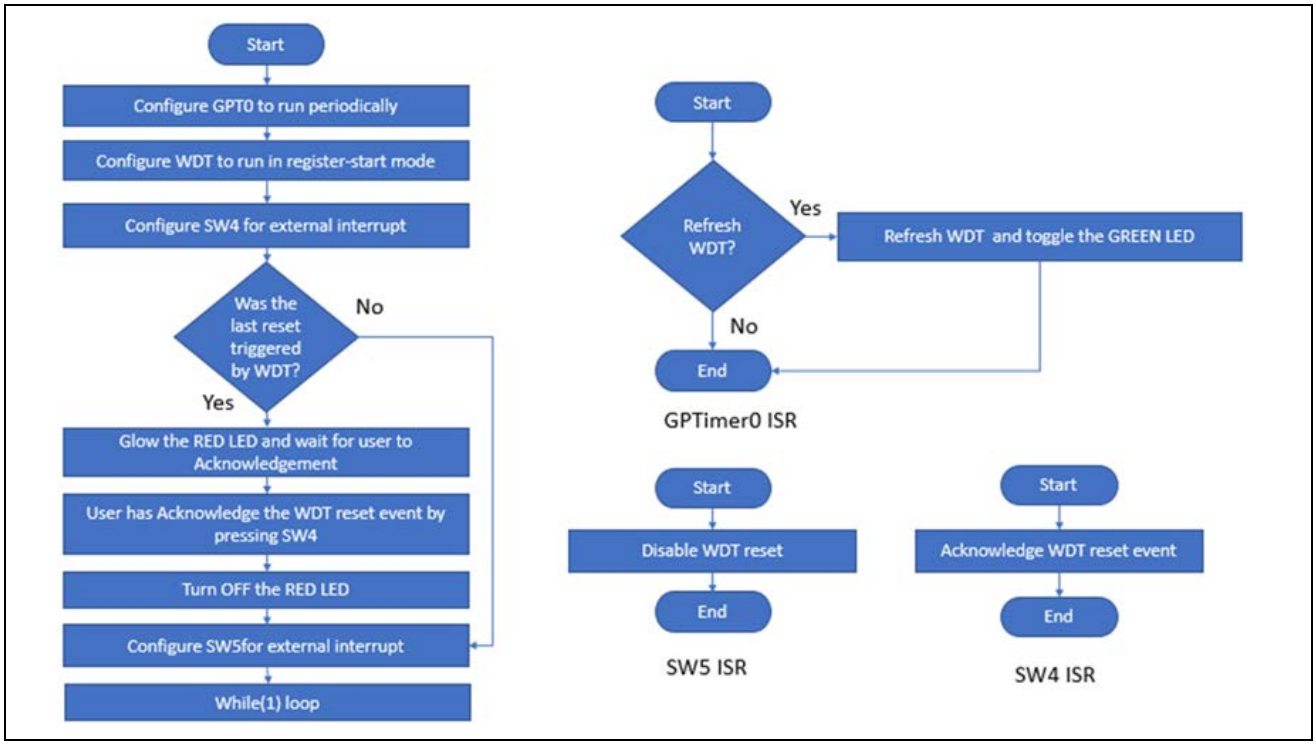

Figure 6. WDT HAL Module Application Project Flow Diagram

The `WDTIWDT_HAL_MG_AP.c` file is 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.

This Module Guide for the WDT module would describe the process for configuring and implementing a WDT module, and would use on board switches to control the application.

To refresh WDT, GPT0 is set to period time mode to refresh the WDT in 200 ms interrupt subroutine.

On startup, the application will configure the WDT and GPT0 by calling upon; open calls (`g_wdt.p_api->open` & `g_timer0.p_api->open`) respectively. WDT is configured in **register-start mode** and will be refreshed by calling `g_wdt.p_api->refresh call`, periodically by the GPTO.

When switch SW5 is pressed, `g_wdt_refresh_flag` is cleared, the GPT0 interrupt subroutine would stop the WDT refresh, causing the WDT underflow. On the next reset, the application will check for the cause of RESET, and if the WDT bit of `RSTSR1_b (RSTSR1 SFR)` is set, the program will light up the red LED and wait in the loop for you to press SW4. The SW5 to disable WDT refresh would be disabled till you acknowledge the RESET condition by pressing SW4.

Table 6. WDT Configuration Settings for the Application Project in BSP

ISDE Property	Value Set
WDT Start Mode Select	Stop WDT after a reset(register – start mode)
WDT Timeout Period	16384 cycles
WDT Clock Frequency Division Ratio	8192
WDT Window End Position	0% (no window end position)
WDT Window Start Position	100% (no window start position)
WDT Reset Interrupt Request	Reset
WDT Stop Control	Stop counting when entering sleep mode

Table 7. WDT Configuration Settings for the Application Project in Threads

ISDE Property	Value Set
Parameter Checking	Default(BSP)
Name	g_wdt0
WDT Start Mode Select	Register
Start Watchdog After Configuration	True
Timeout	16384 cycles
Clock Division Ratio	PCLKB/8192
Window Start Position	100% (no window start position)
Window End Position	0% (no window end position)
Reset Control	Reset Output
Stop Control	WDT Count disabled in Low Power Mode
NMI Callback	NULL

Select the Auto-start mode or Register-start mode by setting the WDT Start Mode Select bit (OFS0.WDTSTRT) in the OFS0 register. When Auto-start mode is selected, the settings in the WDT Control Register (WDTCR), WDT Reset Control Register (WDTRCR), and WDT Count Stop Control Register (WDTCSTPR) are disabled, and the settings in the OFS0 register are enabled. When Register-start mode is selected, the OFS0 register settings are disabled, and the settings in the WDT Control Register (WDTCR), WDT Reset Control Register (WDTRCR), and WDT Count Stop Control Register (WDTCSTPR) are enabled.

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
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

8. Customizing the WDT HAL Module for a Target Application

Some configuration settings will normally be changed by the developer from those shown in the application project. For example, you can easily change the configuration settings for the WDT clock by updating the PCLKB in the **Clock** tab. You can also change the WDT configuration in BSP and **Threads** tab.

9. Running the WDT HAL Module Application Project

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

To implement the WDT HAL module application in a new project, follow the steps below for defining, configuring, auto-generating files, adding code, compiling, and debugging on the targeted MCU kit. Following these steps in 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.

To run the WDT HAL Module application project, simply follow these steps:

1. Create a new Renesas Synergy™ project for the SK-S7G2 called `WDT_HAL_MG_AP`.
2. Select the **Threads** tab.
3. Click on **HAL/common** thread.
4. Add the WDT driver, GPT driver, External IRQ drivers for SW4 and SW5 by adding drivers through the stack's windows on the right side.
5. Add `WDT_HAL_MG.c` and `WDT_HAL_MG.h` to the new project.
6. Connect to the host PC using the USB cable (use the **J19 DEBUG_USB** connector).
7. Start to debug the application.
8. To test without debugger, stop the debug operation and recycle the power by unplugging and plugging the micro USB from J19 connector.
9. The green LED (LED1 on SK-S7G2) will start blinking at the rate of 200 ms, indicating WDT refresh.
10. Press switch SW5, and observe that the blinking of the green LED stops, indicating that WDT is not getting refreshed.
11. After approximately 2.23 seconds, the WDT will RESET the device, and as an indication of that the red LED (LED2 on SK-S7G2) will glow along with the green flashing LED.
12. Press SW4 to acknowledge the WDT RESET, this will turn OFF the red LED.

Note**:** Since the project runs without J-Link support, hence SEMI-HOSTING is not enabled for this project. However, the AMBER LED (LED3 on SK-S7G2) will light up on detection of any error in WDT, GPT and external interrupts API calls. When this happens, you are advised to connect the J-Link and do a step by step debugging to explore the bug.

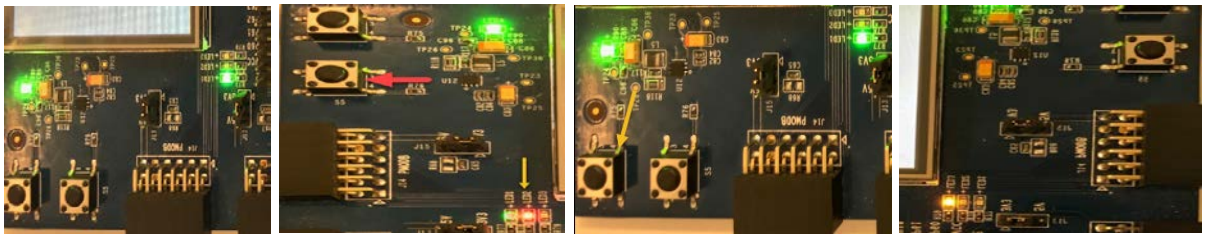

Figure 7. WDT application stages from the WDT Driver Application Project (From Left to Right)

The stages shown in the Figure 7 from left to right are: a green LED blinks on press of SW5, after a RESET red LED starts glowing. On press of SW4, the red LED is turned OFF and an amber LED starts glowing on detection of any error caused by the API calls.

10. DWDT 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) illustrate 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. WDT HAL Module Next Steps

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

12. Reference Information

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

Links to all the most up-to-date `r_wdt` module reference materials and resources are available on the Renesas Synergy Knowledge Base: https://en-us.knowledgebase.renesas.com/English_Content/Renesas_Synergy%E2%84%A2_Platform/Renesas_Synergy_Knowledge_Base/R_WDT_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	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	Mar.09.17	-	Initial release
1.01	Aug.23.17	-	Update to Hardware and Software Resources table
1.02	Dec.05.17	-	Editing and release
1.03	May.01.19	-	Updated for SSP v1.6.0 and changed AGT timer to GPT timer

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/.