Renesas Synergy™ Platform

Power Profiles V2 Framework Module Guide

Introduction

This module guide will enable you to effectively use a module in your own design. Upon completion of this guide you will be able to add this module to your own design, configure it correctly for the target application, and write code using the included application project code as a reference and efficient starting point. References to more detailed API descriptions and suggestions of other application projects that illustrate more advanced uses of the module are available in the Renesas Synergy Knowledge Base (as described in the References section at the end of this document) and should be valuable resources for creating more complex designs.

The Power Profiles V2 Framework is a generic API that can be used to control the system clocks, the I/O ports, the operating modes (indirectly through the clock control), and the low power modes of the MCU. The Power Profiles V2 Framework, when used with the LPM V2 Driver, CGC Driver and I/O Port Driver, gives the user advanced control over the power consumption of the MCU.

Contents

1. Power Profiles V2 Framework Module Features

- Uses Low Power Modes V2
- Sets CGC clock configuration and I/O Port pin configuration when entering and exiting the configured low-power mode
- Supports both threaded and non-threaded operations

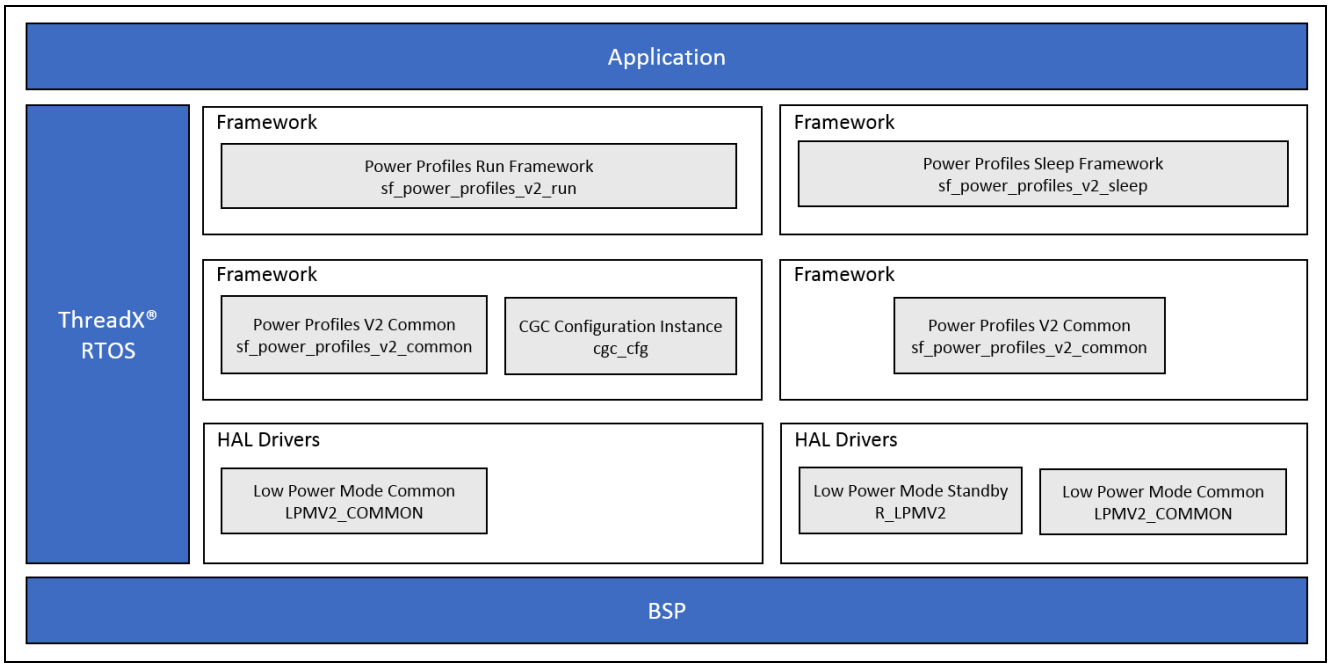

Figure 1. Power Profiles V2 Framework Organization, Options and Stack Implementations

Note: Due to the many options for lower level LPMV2 HAL modules available for different MCUs, the figure does not show every option. The short-hand <MCU> indicator is replaced with the CPU name (like S7G2) in the actual framework implementation.

2. Power Profiles V2 Framework Module APIs Overview

The table lists low-level LPM V2 HAL modules used in the framework, which varies depending on the target MCU.

MCU	Driver
S124	S124 Low Power Mode Sleep on r_lpmv2
S124	S124 Low Power Mode Standby on r_lpmv2
S128	S128 Low Power Mode Sleep on r_lpmv2
S128	S128 Low Power Mode Standby on r_lpmv2
S3A3	S3A3 Low Power Mode Sleep on r_lpmv2
S3A3	S3A3 Low Power Mode Standby on r_lpmv2
S3A7	S3A7 Low Power Mode Sleep on r_lpmv2
S3A7	S3A7 Low Power Mode Standby on r_lpmv2
S5D9	S5D9 Low Power Mode Sleep on r_lpmv2
S5D9	S5D9 Low Power Mode Standby on r_lpmv2
S5D9	S5D9 Low Power Mode Deep Standby on r_lpmv2
S7G2	S7G2 Low Power Mode Sleep on r_lpmv2
S7G2	S7G2 Low Power Mode Standby on r_lpmv2
S7G2	S7G2 Low Power Mode Deep Standby on r_lpmv2

Due to the number of lower-level LPM V2 HAL modules, identifying separate APIs and configuration settings will be limited to the S7G2 MCU in this module guide. All the APIs and configuration settings are the same (except for MCUs which do not support Deep Standby), making it easy to extrapolate to any target MCU.

The Power Profiles defines APIs for common functions such as open, sleep, and close. The following table lists all the available APIs, an example API call, and a short description. A table of status return values is also provided.

Table 1. Power Profiles V2 Framework Module API Summary

Function Name	Example API Call and Description
.open	`g_sf_power_profiles_v2_common.p_api->open(g_sf_power_profiles_v2_low_power_0.p_ctrl, g_sf_power_profiles0.p_cfg);` Initialize the Power Profiles V2 Framework.
.runApply	`g_sf_power_profiles_v2_common.p_api->runApply(g_sf_power_profiles_v2_common.p_ctrl, &g_sf_power_profiles_v2_run_0);` Apply the selected run power profile.
.lowPowerApply	`g_sf_power_profiles_v2_common.p_api->lowPowerApply(g_sf_power_profiles_v2_common.p_ctrl, &g_sf_power_profiles_v2_low_power_0);` Apply the selected low power profile.
.close	`g_sf_power_profiles_v2_common.p_api->close(g_sf_power_profiles_v2_common.p_ctrl);` Close the module.
.versionGet	`g_sf_power_profiles_v2_common.p_api->versionGet(&p_version);` Get version and store it in provided pointer p_version.

Note: For details on operation and definitions for function data structures, typedefs, defines, API data, API structures, and function variables, see the *SSP User's Manual* API References for the associated module.

Table 2. Status Return Values

Name	Description
SSP_SUCCESS	Function successful.
SSP_ERR_ASSERTION	Assertion error.
SSP_ERR_IN_USE	The framework has already been initialized.
SSP_ERR_INVALID_HW_CONDITION	Incompatible system clock configuration.
SSP_ERR_NOT_OPEN	Device not open.
SSP_ERR_UNSUPPORTED	The function is not supported by the module.
SSP_ERR_INTERNAL	Internal error.

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. Power Profiles V2 Framework Module Operational Overview

The Power Profiles V2 Framework is a generic API that can be used to control system clocks, I/O ports, operating modes (indirectly through the clock control), and low power modes of the MCU. The Power Profiles V2 Framework, when used with the LPM V2 Driver, CGC Driver, and I/O Port Driver, gives users advanced control over MCU power consumption. The Power Profiles V2 Framework provides two main functions to control power consumption, the `@ref sf_power_profiles_v2_api_t::runApply` and the `@ref sf_power_profiles_v2_api_t::lowPowerApply`. The `runApply()` function uses a CGC Clocks configuration and an I/O Port pin configuration to set the MCU's system clocks and I/O port pins. The `lowPowerApply()` function uses a LPM V2 configuration and two I/O port configurations to set low-power

mode and I/O port pins before entering the configured low power mode, and after waking from the low power mode. For details on available low-power modes, see the LPM V2 usage notes and the associated MCU Hardware Manual for details about the available low power modes.

The Power Profiles V2 Framework uses the LPM V2, IOPORT, and CGC Drivers of the Synergy Software Package and provides an easy-to-use software interface to control the power consumption of the MCU.

Comparing Frameworks: Power Profiles V2 versus Power Profiles V1

The Power Profiles V1 Framework uses the LPM V1 driver. Due to constraints in both Power Profiles V1 and LPM V1, support consists of only a subset of low-power features. The LPM V2 design allows the user to have exposure to much more low power functionality. The design allows these features to be available to a new Framework module. Power Profiles V1 uses LPM V1. To support LPM V2, Power Profiles V2 was created. Power Profiles V2 supports all the features of Power Profiles V1, as well as the other features covered in this document. While Power Profiles V1 and V2 interfaces are similar, they are not compatible.

Note: Power Profiles V1 and Power Profiles V2 cannot be used in the same project. For all new projects, it is recommended that applications use Power Profiles V2.

Power Profiles V2 Module Operational Description

The Power Profiles V2 Framework configures the system in both a Run state and a Low Power state. The system clocks, I/O pins, and low power mode of the MCU can all be handled and controlled using the Power Profiles V2 Framework.

The Power Profiles V2 Framework API function `@ref sf_power_profiles_v2_api_t::open` initializes the Power Profiles V2 Framework internal variables, instance variables, and the LPM V2 Driver. If the project uses ThreadX®, the Framework ensures its safe use in a multi-threaded environment.

The `runApply()` and `lowPowerApply()` functions optionally use I/O port pin configurations to provide control of MCU I/O Ports. The `lowPowerApply()` function can use two pin configurations: one to set the pins to a state appropriate for the low power mode when the MCU is not be executing instructions, and a second, when waking from the low power mode, when instruction execution resumes.

The Power Profiles V2 Framework `runApply()` function applies the user-defined, optional pin configuration, it then applies the user-defined CGC clocks configuration. The user can switch clocks on and off, change clock dividers, and switch the system clocks using the CGC Clocks configuration structure.

The Power Profiles V2 Framework API function `runApply()` performs tasks in the following order:

- If the project uses ThreadX, the function it acquires the ThreadX mutex prior to calling any lower-level driver.
- Applies the user-specified optional pin configuration.
- Applies the user-specified CGC Clocks configuration.
- If the project uses ThreadX, the function returns the ThreadX mutex.

The `lowPowerApply()` function uses a LPM V2 configuration to set the low-power mode, triggers to wake from the low-power mode, monitors the state of bus pins, and other settings that are MCU-specific. The `lowPowerApply()` function can optionally use an application callback function. The prototype can be found in **/src/synergy_gen/hal_data.c or /src/synergy_gen/<thread name>.c**.

The Power Profiles V2 Framework `lowPowerApply()` function performs tasks in the following order:

1. If the project uses ThreadX, the function acquires the ThreadX mutex prior to calling any lower level driver.
2. Applies the optional user-specified, low-power, entry pin configuration.
3. Call the user-specified callback function with the enumeration: **SF_POWER_PROFILES_V2_EVENT_PRE_LOW_POWER**.
4. Applies the user-specified, low-power mode configuration. Any valid LPM V2 configuration can be used.
5. Enters low-power mode.
6. If the low-power mode chosen was other than Deep Standby, the MCU resumes code execution from the same point, once the wakeup trigger is detected. (If the LPM V2 low-power mode configuration was in Deep Standby, the MCU does not resume code execution; instead, it does a soft reset once the wakeup trigger is detected.).

7. Applies the optional user-specified, low power, exit pin configuration.
8. Call the user specified callback function with the enumeration:
   SF_POWER_PROFILES_V2_EVENT_POST_LOW_POWER.
9. If the project uses ThreadX, the function returns the ThreadX mutex.

The following figure shows how a user adds the Power Profiles V2 Run or Low Power profiles to the project. The Power Profiles V2 common module is added automatically.

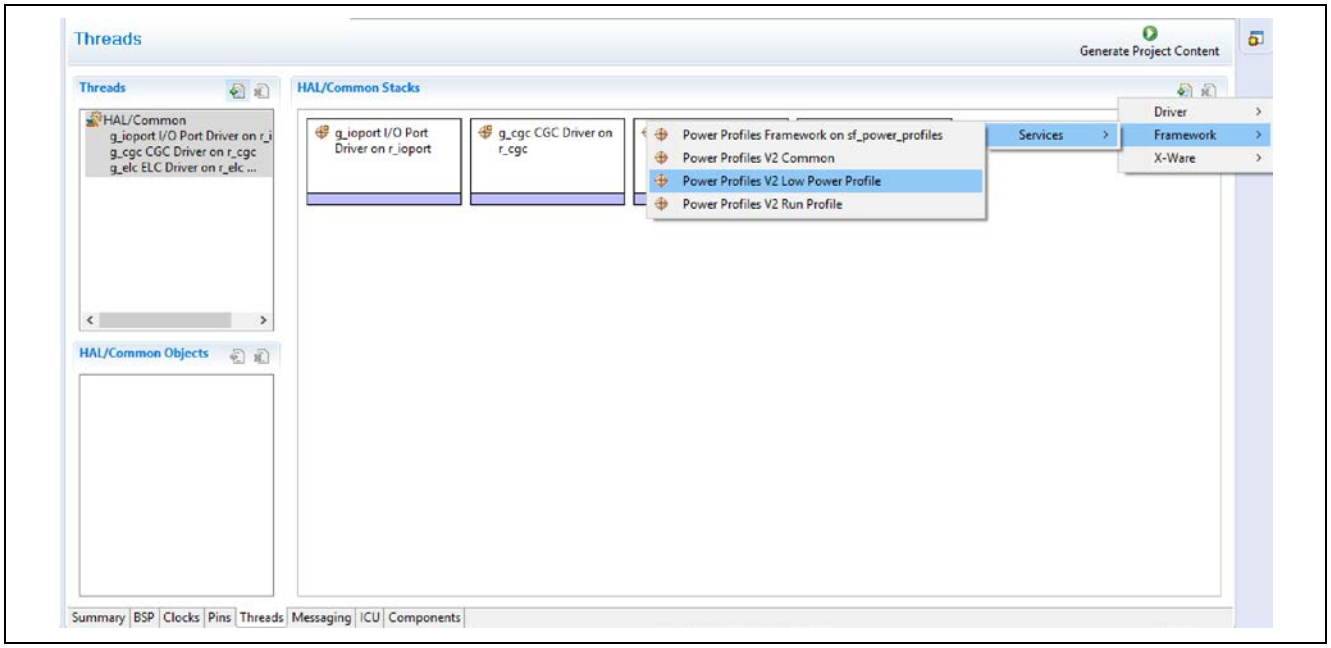

Figure 2. Adding Power Profiles V2 Run or Low Power profiles outside a thread

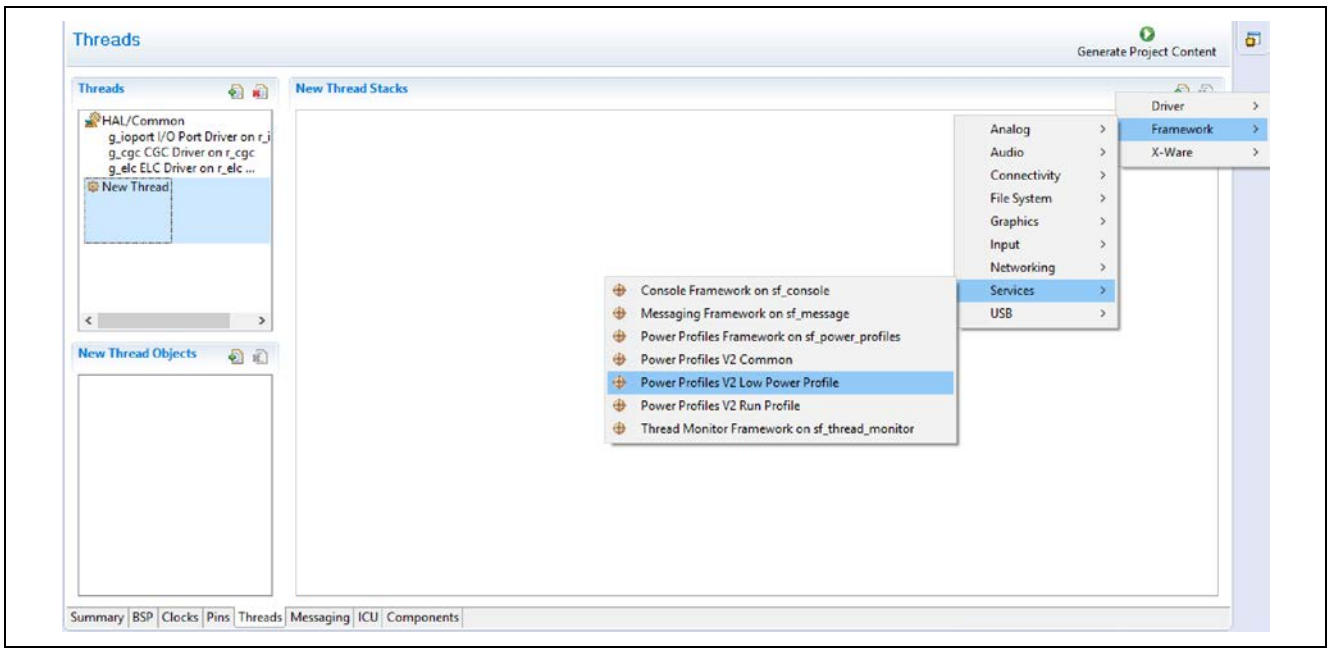

Figure 3. Adding Power Profiles V2 Run or Low Power profiles within a thread

After adding a Power Profiles V2 Run or Low Power Profile, an LPM V2 Common module is added. If an, LPM V2 Common instance already exists in the project, it is used. A Power Profiles V2 Run Profile does not directly use the LPM V2, but it is still a dependency for a successful build.

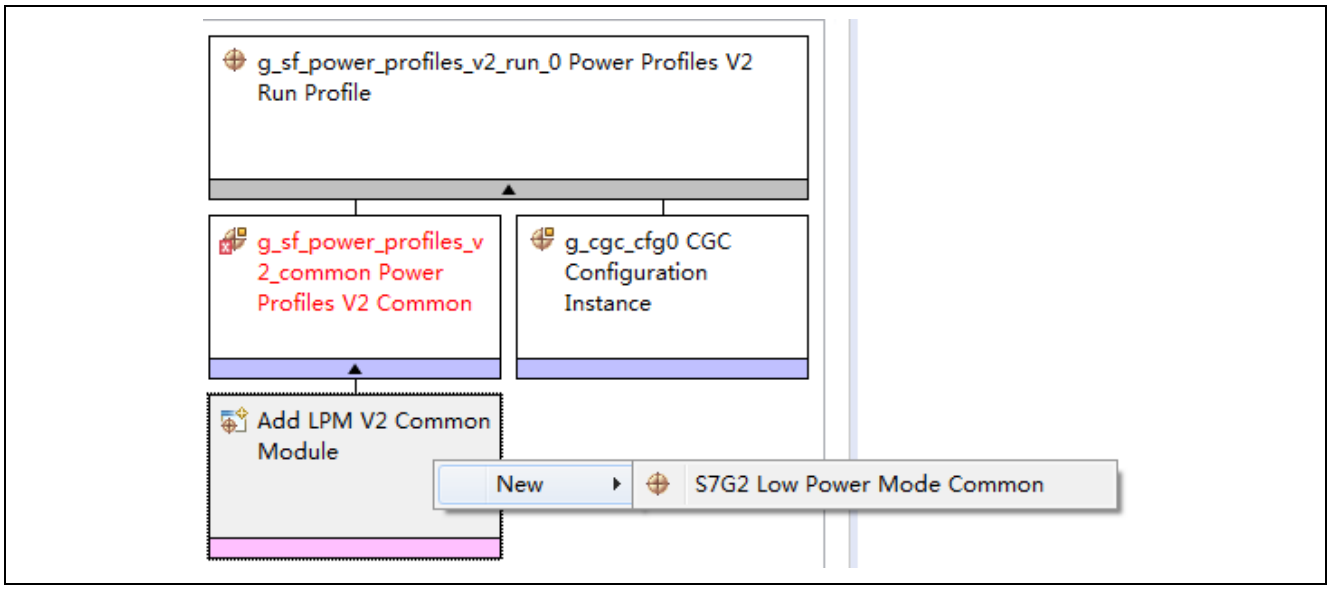

Figure 4. Adding a LPM V2 Common module

The following figure shows the Power Profiles V2 Run Profile depends upon a CGC Clocks Configuration. It shows the CGC Clocks Configuration instance and its configuration options. If a CGC Clocks Configuration instance is not present in the project, it is added automatically. If the instance is present, that instance could be used instead. Controlling system clocks is a critical in controlling MCU power consumption. For details, see the CGC usage notes.

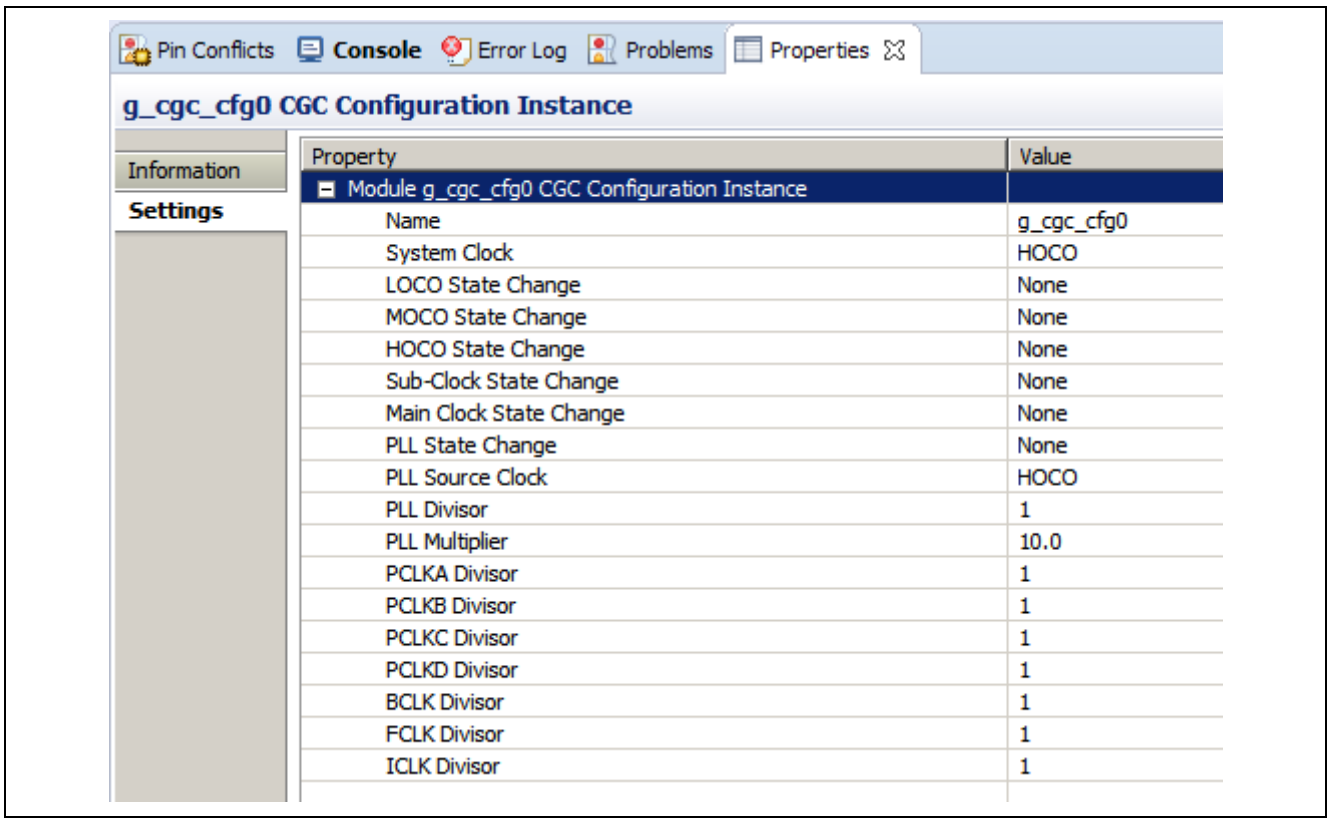

Figure 5. CGC Clocks Configuration options

The following figure shows how to add a Power Profiles V2 Low Power Profile. The Power Profiles V2 Low Power Profile uses an LPM V2 standby instance; but a LPM V2 deep standby instance or LPM V2 sleep instance could be used instead, depending on which MCU is currently being used.

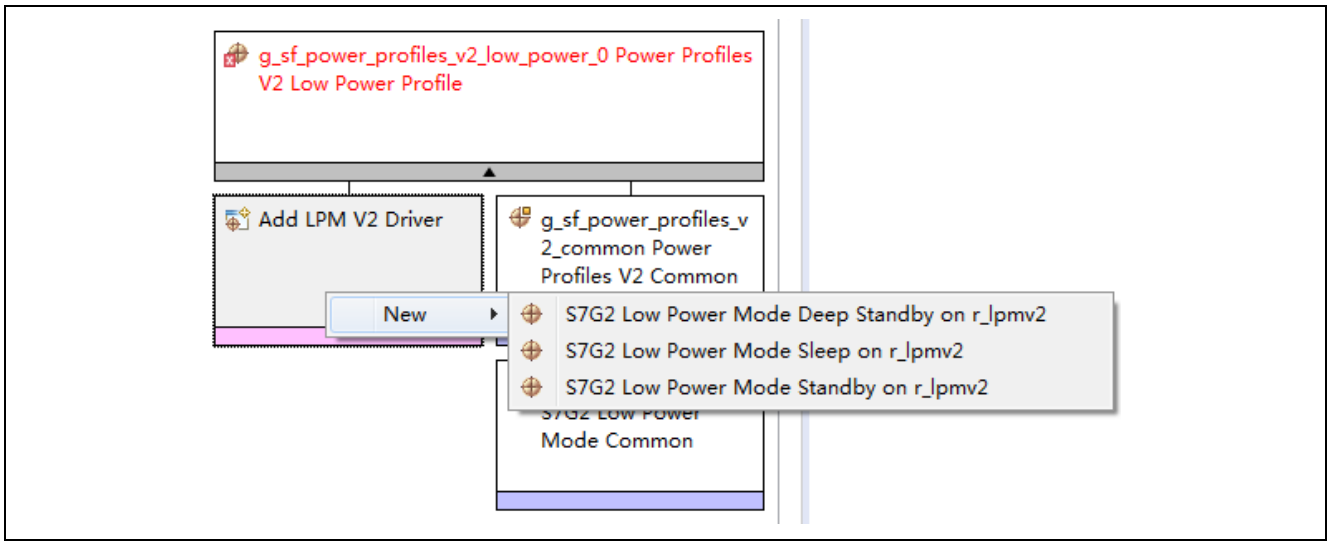

Figure 6. Add a Power Profiles V2 Low Power Profile

Properties for the LPM V2 instance can be configured; select the instance and review the Properties pane.

Configuring the Pins for the Power Profiles V2 Framework

Optional: If you wish to create additional I/O port pin configurations via the Pins tab, use the following steps:

1. In the top-level directory of your project, find the file with file extension *.**pincfg**.
2. Make a copy of this file and rename it, keeping the new file in the same top-level directory of the project. The new file is now available as an option on the Pins tab of the Synergy Configuration. Look for the file name in the drop-down list on the Pins tab, **Select pin configuration**.
3. Check the **Generate** data checkbox, then type in a pin configuration name. The checkbox and text entry can be found to the right of the pin configuration drop down.
4. Configure the pins, as desired, for either a Run or Low Power Profile, respectively, using the following screens.
5. Save the project configuration and select **Generate Project Content**.
6. View the pin configuration generated for the ioport_cfg_t structure of the same name as entered in the text box, go to: {*project_directory*}/src/synergy_gen/pin_data.c.
7. Add the name of the ioport_cfg_t structure to one of the pin configuration table entries for a Run or Low Power Profile.

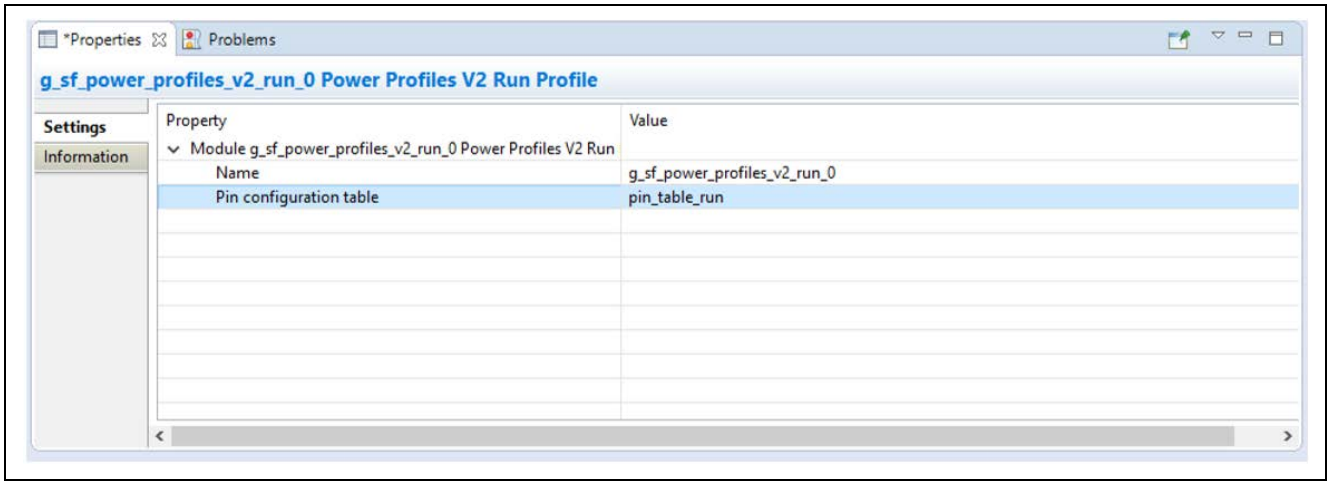

Figure 7. Configuring pin for Run Profile

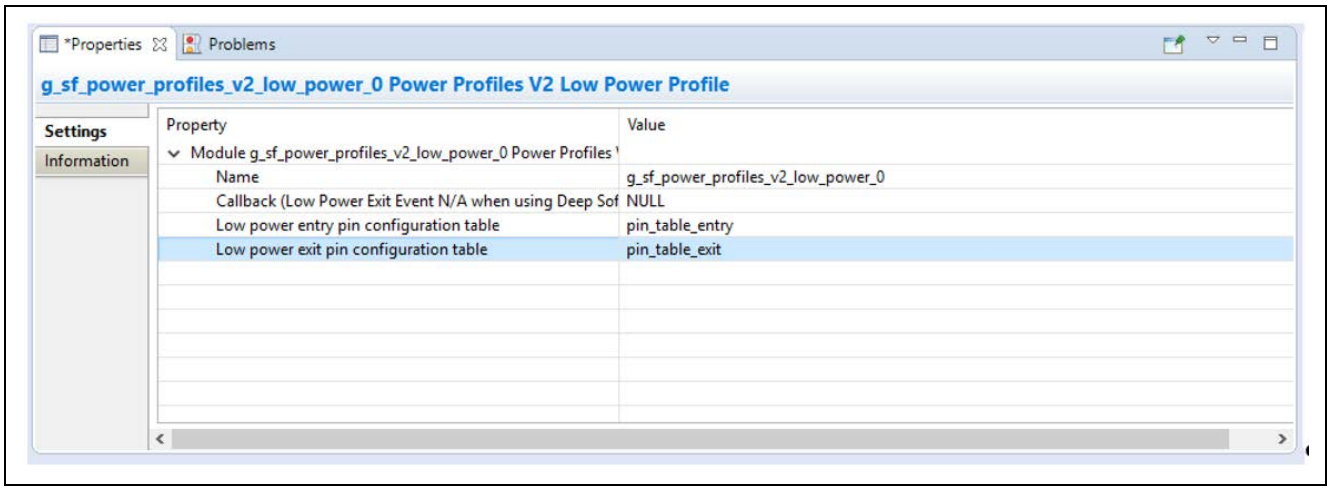

Figure 8. Configuring pin for Low Power Profile

Configuring the Interrupts for the Power Profiles V2 Framework

The Power Profiles V2 Framework does not use any interrupts directly, although any interrupt is capable of waking the MCU while in Sleep mode. This is handled through the LPM V2 driver configuration.

Configuring the Power Profiles V2 Callbacks

Power Profiles V2 Low Power Profiles can notify the application before it enters low-power mode and after waking from low-power mode. The prototype can be found in **/src/synergy_gen/hal_data.c** or **/src/synergy_gen/<thread name>.c**.

The following image shows how to fill in the callback used by the Power Profiles V2 Low Power Profile.

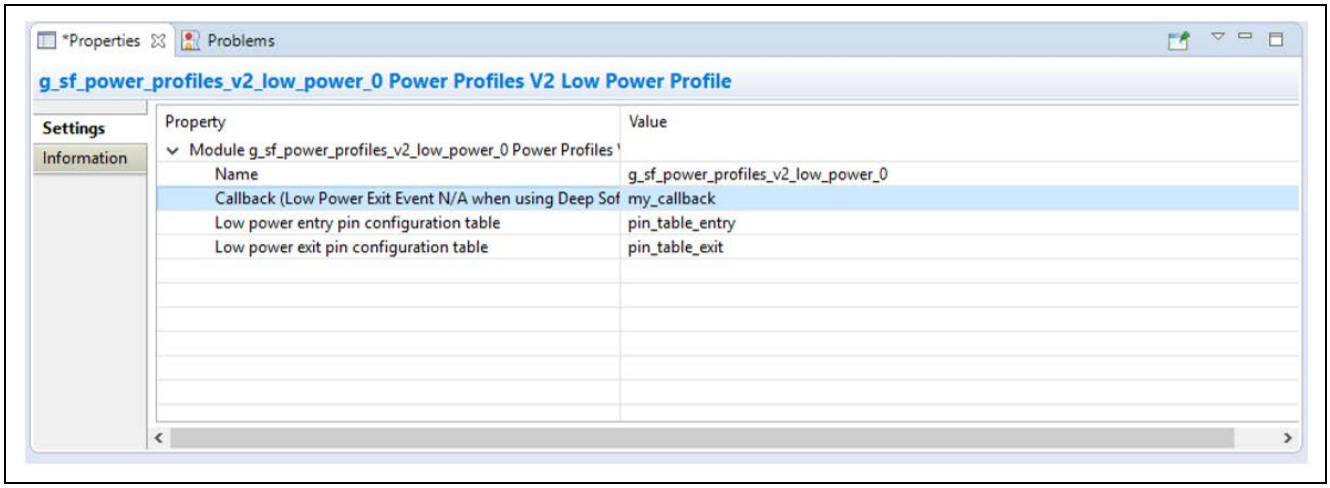

Figure 9. Filling in the callback used by Low Power Profile

Configuring the Low Power Module Parameters

See the LPM V2 usage notes for an in-depth description of how to use LPM V2.

3.1 Power Profiles V2 Framework Module Important Operational Notes and Limitations

3.1.1 Power Profiles V2 Framework Module Operational Notes

- An LPM V2 Driver instance is required to create Power Profiles V2 applications. The CGC driver is included in a Synergy project by default. To use the Power Profiles V2 runApply() function, another instance of the CGC Driver is required to create a CGC Clocks configuration. Since the CGC Driver is included in all Synergy projects by default and does not add to the code size of the project.
- The I/O Port pin configurations can be created without adding an additional instance of the I/O Port driver.
- When used with ThreadX, this framework uses ThreadX intrinsic objects like mutexes. Operation with ThreadX is optional.

- Power Profiles V1 and Power Profiles V2 cannot be used in the same project. For all new projects, it is recommended that applications use Power Profiles V2.

3.1.2 Power Profiles V2 Framework Module Limitations

- The Power Profiles V2 Framework does not handle starting or stopping MCU peripherals.
- The Power Profiles V2 Framework open function will not be called automatically prior to main if the project does not use ThreadX. The initialization must be done explicitly by calling g_common_init() as the following code shows, or by explicitly calling the API function open(). This is not a Power Profiles V2 limitation but a result of any Framework module that supports being used without an RTOS.

```
@code
#include "hal_data.h"
void hal_entry(void)
{
    g_common_init();
    g_sf_power_profiles_v2_common.p_api->runApply(
        g_sf_power_profiles_v2_common.p_ctrl,
        &g_sf_power_profiles_v2_run_0);
    g_sf_power_profiles_v2_common.p_api->lowPowerApply(
        g_sf_power_profiles_v2_common.p_ctrl,
        &g_sf_power_profiles_v2_low_power_0);
    g_sf_power_profiles_v2_common.p_api-
>close(g_sf_power_profiles_v2_common.p_ctrl);
}
@endcode
**/
```

For additional operations limitations to this module, see the latest SSP Release Notes.

4. Including the Power Profiles V2 Framework Module in an Application

This section describes how to include the Power Profiles 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 these tasks, see the initial chapters of the *SSP User's Manual*, where you can learn how to manage these important tasks in creating SSP-based applications.

To add the Power Profiles framework module to an application, simply add it to a thread using the stacks selection sequence given in the following table. (The default name for the Power Profiles is `g_sf_power_profiles_low_power_0`. This name can be changed in the associated Properties window.)

Table 3. Power Profiles V2 Framework Module Selection Sequence

Resource	ISDE Tab	Stacks Selection Sequence
g_sf_power_profiles_low_power_0 Power Profiles V2 Low Power Profile	Threads	New Stack > Framework > Services > Power Profiles V2 Low Power Profile
g_sf_power_profiles_run_0 Power Profiles V2 Run Profile	Threads	New Stack > Framework > Services > Power Profiles V2 Run Profile

When the Power Profiles V2 low power or run module is added to the Thread Stack, the configurator automatically adds any needed lower-level drivers. Any drivers that need additional configuration information are box text highlighted in Red. Modules with a Gray band are individual, standalone modules.

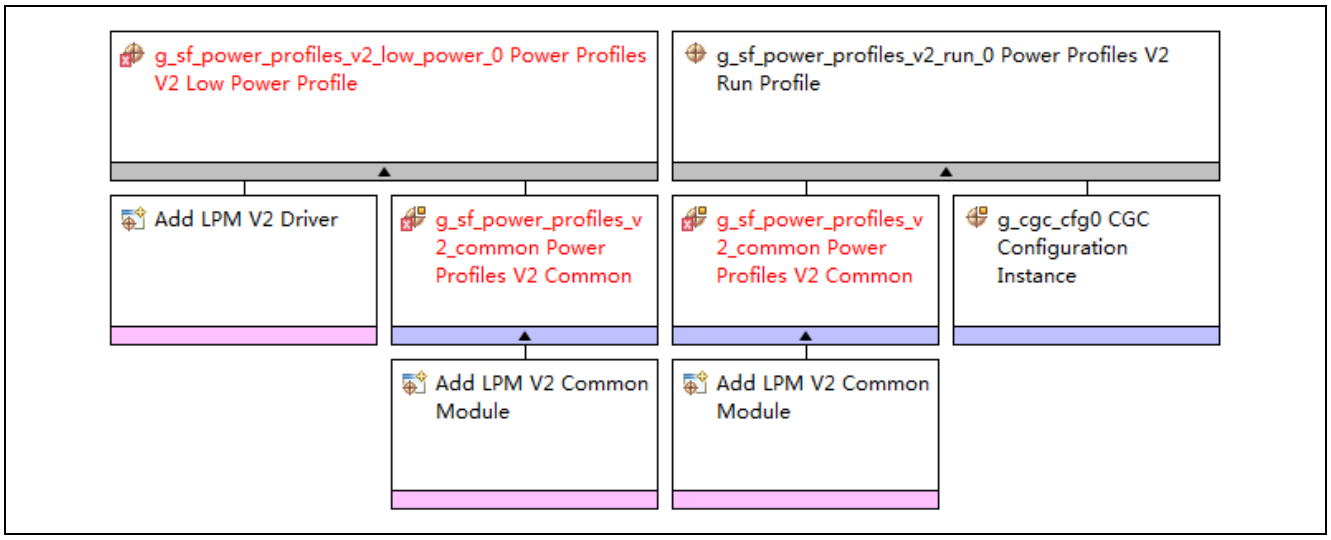

Figure 10. Power Profiles V2 Framework Module Stack

5. Configuring the Power Profiles V2 Framework Module

Configure the Power Profiles V2 run and low power modules for the desired operation you need. The SSP configuration window automatically identifies, by highlighting the block in red, any configuration selections required for lower level modules (such as interrupts or operating modes) for successful operation. Only those properties that can be changed without causing conflicts are available for modification. Properties that are 'locked' are not available for changes. They are identified with a lock icon for the 'locked' property in the property window in the ISDE. This approach simplifies the configuration process and making it less error prone than previous 'manual' approaches. The available configuration settings and defaults for all the user accessible properties are given in the properties tab within the SSP configurator. These properties are also included here in tables for easy reference.

One of the properties most often identified as requiring a change is the interrupt priority. This configuration setting is available with 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 ISDE Interrupt priorities listed in the properties window show the validity of the setting based on the MCU targeted (CM4 or CM0+). This level of detail is not included in the following tables but are easily visible in the ISDE when you configure the Interrupt Priority levels.

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

There are two different stack selections for the Power Profiles V2 Framework, the Run Profile and the Low Power Profile. The following sections describe their respective configuration settings.

5.1 Power Profiles V2 Framework Module Configuration Settings for Run Profile

Typically, only a small number of settings must be modified from the default for lower level drivers and these are indicated via the red text in the Thread Stack block. Note that some configuration properties are set to a certain value for proper framework operation and are locked to prevent user modification. The following table identifies these settings within the properties section for the module.

Table 4. Configuration Settings for Power Profiles V2 Run Profile

ISDE Property	Value	Description
Name	g_sf_power_profiles_v2_run_0	Module name
Pin configuration table	NULL	Pin configuration table selection

Table 5. Configuration Settings for Power Profiles V2 Common

ISDE Property	Value	Description
Name	g_sf_power_profiles_v2_common	Module name
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Enables or disables the parameter checking.

Table 6. Configuration Settings for Low Power Mode Common

ISDE Property	Value	Description
Name	g_lpmv2_common	Module name
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Enables or disables the parameter checking.

Table 7. Configuration Settings for CGC Configuration Instance

ISDE Property	Value	Description
Name	g_cgc_cfg0	Module name
System Clock	HOCO, MOCO, LOCO, Main Oscillator, Sub Clock, PLL Default: HOCO	System clock selection
LOCO State Change	None, Start, Stop Default: None	LOCO state change selection
MOCO State Change	None, Start, Stop Default: None	MOCO state change selection
HOCO State Change	None, Start, Stop Default: None	HOCO state change selection
Sub-Clock State Change	None, Start, Stop Default: None	Sub-clock state change selection
Main Clock State Change	None, Start, Stop Default: None	Main clock state change selection
PLL State Change	None, Start, Stop Default: None	PLL state change selection
PLL Source Clock	HOCO, MOCO, LOCO, Main Oscillator, Sub Clock, PLL Default: Main Oscillator	PLL source clock selection
PLL Divisor	1, 2, 3 Default: 2	PLL divisor selection
PLL Multiplier	10.0, 10.5, 11.0, 11.5, 12.0, 12.5, 13.0, 13.5, 14.0, 14.5, 15.0, 15.5, 16.0, 16.5, 17.0, 17.5, 18.0, 18.5, 19.0, 19.5, 20.0, 20.5, 21.0, 21.5, 22.0, 22.5, 23.0, 23.5, 24.0, 24.5, 25.0, 25.5, 26.0, 26.5, 27.0, 27.5, 28.0, 28.5, 29.0, 29.5, 30.0 Default: 20.0	PLL multiplier selection
PCLKA Divisor	1, 2, 4, 8, 16, 64 Default: 2	PCLKA divisor selection
PCKLB Divisor	1, 2, 4, 8, 16, 64 Default: 4	PCKLB divisor selection
PCLKC Divisor	1, 2, 4, 8, 16, 64 Default: 4	PCLKC divisor selection
PCLKD Divisor	1, 2, 4, 8, 16, 64 Default: 2	PCLKD divisor selection

ISDE Property	Value	Description
BCLK Divisor	1, 2, 4, 8, 16, 64 Default: 2	BCLK divisor selection
FCLK Divisor	1, 2, 4, 8, 16, 64 Default: 4	FCLK divisor selection
ICLK Divisor	1, 2, 4, 8, 16, 64 Default: 1	ICLK divisor selection

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

5.2 Power Profiles V2 Configuration Settings for Low Power Profile

Typically, only a small number of settings must be modified from the default for lower level drivers and these are indicated via the red text in the Thread Stack block. Note that some of the configuration properties must be set to a certain value for proper framework operation and will be locked to prevent user modification. The following table identifies all the settings within the properties section for the module.

Table 8. Configuration Settings for Power Profiles V2 Low Power Profile

ISDE Property	Value	Description
Callback (Low Power Exit Event N/A when using Deep Software Standby)	NULL	Callback selection
Low power entry pin configuration table	NULL	Low power entry pin configuration table selection
Low power exit pin configuration table	NULL	Low power exit pin configuration table selection

Table 9. Configuration Settings for Low Power Mode Deep Standby

ISDE Property	Value	Description
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Enables or disables the parameter checking
Name	g_lpmv2_deep_standby	Module name
Output port state in standby and deep standby, applies to address output, data output, and other bus control output pins	High impedance state, No change Default: No change	
Maintain or reset the IO port states on exit from deep standby mode	Maintain the IO port states, Reset the IO port states Default: Maintain the IO port states	
Internal power supply control in deep standby mode	Maintain the internal power supply, Cut the power supply to standby RAM, low-speed on-chip oscillator, AGTn, and USPFS/HS resume detecting unit, Cut the power supply to LVDn, standby RAM, low-speed on-chip oscillator, AGTn, and USBFS/HS resume detecting unit Default: Maintain the internal power supply	
IRQ0-15	Enabled, Disabled Default: Disabled	IRQ0-15 selection
IRQ0-15 Edge	Disabled, Rising Edge, Falling Edge Default: Disabled	IRQ0-15 Edge selection
LVD1	Enabled, Disabled Default: Disabled	LVD1 selection
LVD1 Edge	Disabled, Rising Edge, Falling Edge Default: Disabled	LVD1 Edge selection
LVD2	Enabled, Disabled Default: Disabled	LVD2 selection

ISDE Property	Value	Description
LVD2 Edge	Disabled, Rising Edge, Falling Edge Default: Disabled	LVD2 Edge selection
RTC Interval	Enabled, Disabled Default: Disabled	RTC Interval selection
RTC Alarm	Enabled, Disabled Default: Disabled	RTC Alarm selection
NMI	Enabled, Disabled Default: Disabled	NMI selection
NMI Edge	Disabled, Rising Edge, Falling Edge Default: Disabled	NMI Edge selection
USBFS	Enabled, Disabled Default: Disabled	USBFS selection
UBSHS	Enabled, Disabled Default: Disabled	UBSHS selection
AGT11	Enabled, Disabled Default: Disabled	AGT11 selection

Table 10. Configuration Settings for Low Power Mode Sleep

ISDE Property	Value	Description
Name	g_lpmv2_sleep0	Module name
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Enables or disables the parameter checking

Table 11. Configuration Settings for Low Power Mode Standby

ISDE Property	Value	Description
Name	g_lpmv2_standby0	Module name
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Enables or disables the parameter checking
Choose the low power mode	Standby, Standby with snooze Enabled Default: Standby	
Output port state in standby and deep standby, applies to address output, data output, and other bus control output pins	High impedance state, No change Default: No change	
IRQ0-15	Enabled, Disabled Default: Disabled	IRQ1-15 selection
IWDT	Enabled, Disabled Default: Disabled	IWDT selection
Key Interrupt	Enabled, Disabled Default: Disabled	Key Interrupt selection
LVD1 Interrupt	Enabled, Disabled Default: Disabled	LVD1 Interrupt selection
LVD2 Interrupt	Enabled, Disabled Default: Disabled	LVD2 Interrupt selection
Analog Comparator High-speed 0 Interrupt	Enabled, Disabled Default: Disabled	Analog Comparator High-speed 0 Interrupt selection
RTC Alarm	Enabled, Disabled Default: Disabled	RTC Alarm selection
RTC Period	Enabled, Disabled Default: Disabled	RTC Period selection

ISDE Property	Value	Description
USB High-speed	Enabled, Disabled Default: Disabled	USB High-speed selection
USB Full-speed	Enabled, Disabled Default: Disabled	USB Full-speed selection
AGT1 underflow	Enabled, Disabled Default: Disabled	AGT1 underflow selection
AGT1 Compare Match A	Enabled, Disabled Default: Disabled	AGT1 Compare Match A selection
AGT1 Compare Match B	Enabled, Disabled Default: Disabled	AGT1 Compare Match B selection
12C 0	Enabled, Disabled Default: Disabled	12C 0 selection
Snooze Entry Source	RXD0 falling edge, IRQ0-IRQ15, KINT, ACMPHS0, RTC Alarm, RTC Period, AGT1 Underflow, AGT1 Compare Match A, AGT1 Compare Match B Default: RXD0 falling edge	Snooze Entry Source selection
AGT1 Underflow	Enabled, Disabled Default: Disabled	AGT1 Underflow selection
DTC Transfer Completion	Enabled, Disabled Default: Disabled	DTC Transfer Completion selection
DTC Transfer Completion Negated Signal	Enabled, Disabled Default: Disabled	DTC Transfer Completion Negated Signal selection
ADC0 Compare Match	Enabled, Disabled Default: Disabled	ADC0 Compare Match selection
ADC0 Compare Mismatch	Enabled, Disabled Default: Disabled	ADC0 Compare Mismatch selection
ADC1 Compare Match	Enabled, Disabled Default: Disabled	ADC1 Compare Match selection
ADC1 Compare Mismatch	Enabled, Disabled Default: Disabled	ADC1 Compare Mismatch selection
SCI0 Address Match	Enabled, Disabled Default: Disabled	SCI0 Address Match selection
DTC state in Snoooze Mode	Enabled, Disabled Default: Disabled	DTC state in Snoooze Mode selection

Table 12. Configuration Settings for Power Profiles V2 Common

ISDE Property	Value	Description
Name	g_sf_power_profiles_v2_common	Module name
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Enables or disables the parameter checking.

Table 13. Configuration Settings for Low Power Mode Common

ISDE Property	Value	Description
Name	g_lpmv2_common	Module name
Parameter Checking	BSP, Enabled, Disabled Default: BSP	Enables or disables the parameter checking.

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

5.3 Power Profiles V2 Framework Module Clock Configuration

The Power Profiles V2 framework does not require any specific clock settings.

5.4 Power Profiles V2 Framework Module Pin Configuration

The application may optionally maintain the Io Port state during a low power mode.

6. Using the Power Profiles V2 Framework Module in an Application

Once you have configured the modules, and the ISDE has generated the files, use the following steps to implement the Power Profiles V2 Framework in a thread. The steps assume the user-defined name for the Power Profiles V2 Framework common instance is: g_sf_power_profiles_v2_common. The steps assume the user-defined names for the Power Profiles V2 framework Low Power and Run profiles are: `g_sf_power_profiles_v2_low_power_0` and `g_sf_power_profiles_v2_run_0`.

Before using any APIs, define the body of the callback function configured in the Low Power profile. The callback function notifies the application when the MCU is about to enter a low-power mode, and when the MCU just woke up from a low-power mode. Using a callback is optional; but if you do define a callback in the Power Profiles V2 properties, there must be a definition for it.

1. If ThreadX is being used, the Power Profiles V2 Framework `@ref sf_power_profiles_v2_api_t::open` function is called by the Synergy-generated code before the user application code is reached. If ThreadX is not used, the application must call the open function.

Apply a Run Profile at any time using the `runApply()` function. The runApply() function accepts a Run Profile as its second parameter. The parameter can be any valid Run Profile, allowing the application to easily switch between Run Profiles.

```
@code
g_sf_power_profiles_v2_common.p_api-
>runApply(g_sf_power_profiles_v2_common.p_ctrl,
&g_sf_power_profiles_v2_run_0);
@endcode
```

Apply a Low Power Profile using the lowPowerApply() function. The lowPowerApply() function accepts a Low Power Profile as its second parameter. The parameter can be any valid Low Power Profile, allowing the application to easily switch between Low Power Profiles.

```
@code
g_sf_power_profiles_v2_common.p_api-
>lowPowerApply(g_sf_power_profiles_v2_common.p_ctrl,
&g_sf_power_profiles_v2_low_power_0);
@endcode
```

Optional: Close the framework by calling the @ref sf_power_profiles_v2_api_t::close function.

```
@code
g_sf_power_profiles_v2_common.p_api-
>close(g_sf_power_profiles_v2_common.p_ctrl);
@endcode
```

The following diagram identifies these common steps in a typical operational flow.

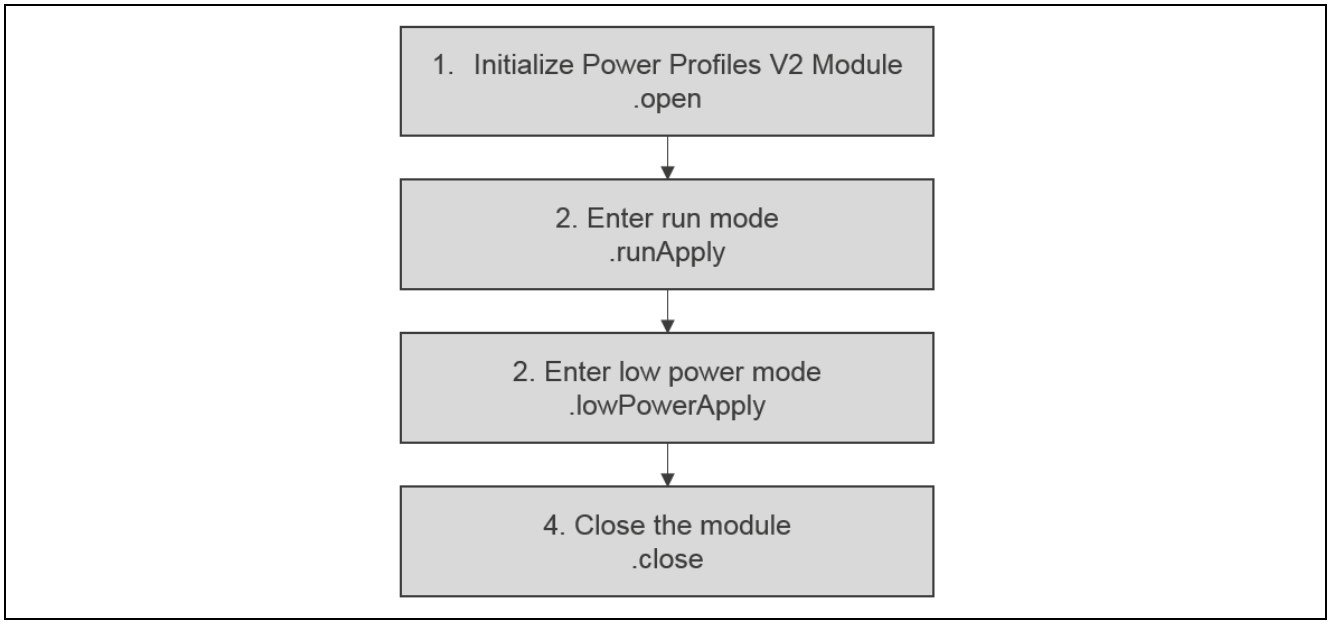

Figure 11. Typical Power Profiles V2 Framework Module Application flow

7. The Power Profiles V2 Framework Module Application Project

The application project associated with this module guide demonstrates the operational flow steps in an example application. In ISDE, you may want to import and open the application project to view the configuration settings for the Power Profiles V2 module. You can also read over the code in `power_profiles_v2_application.c`, which is used to demonstrate the Power Profiles V2 Framework module APIs in a complete design.

The application project shows how to use the Power Profiles V2 Framework module APIs. The following table identifies the target versions for the associated software and hardware used by the application project.

Table 14. Software and Hardware Resources Used by the Application Project

Resource	Revision	Description
e² studio	6.2.1	Integrated Solution Development Environment
SSP	1.5.0	Synergy Software Platform
IAR EW for Renesas Synergy	8.23.1	IAR Embedded Workbench® for Renesas Synergy™
SSC	6.2.1	Synergy Standalone Configurator
SK-S7G2	v3.0 to v3.3	Starter Kit

The following diagram shows the application project flow.

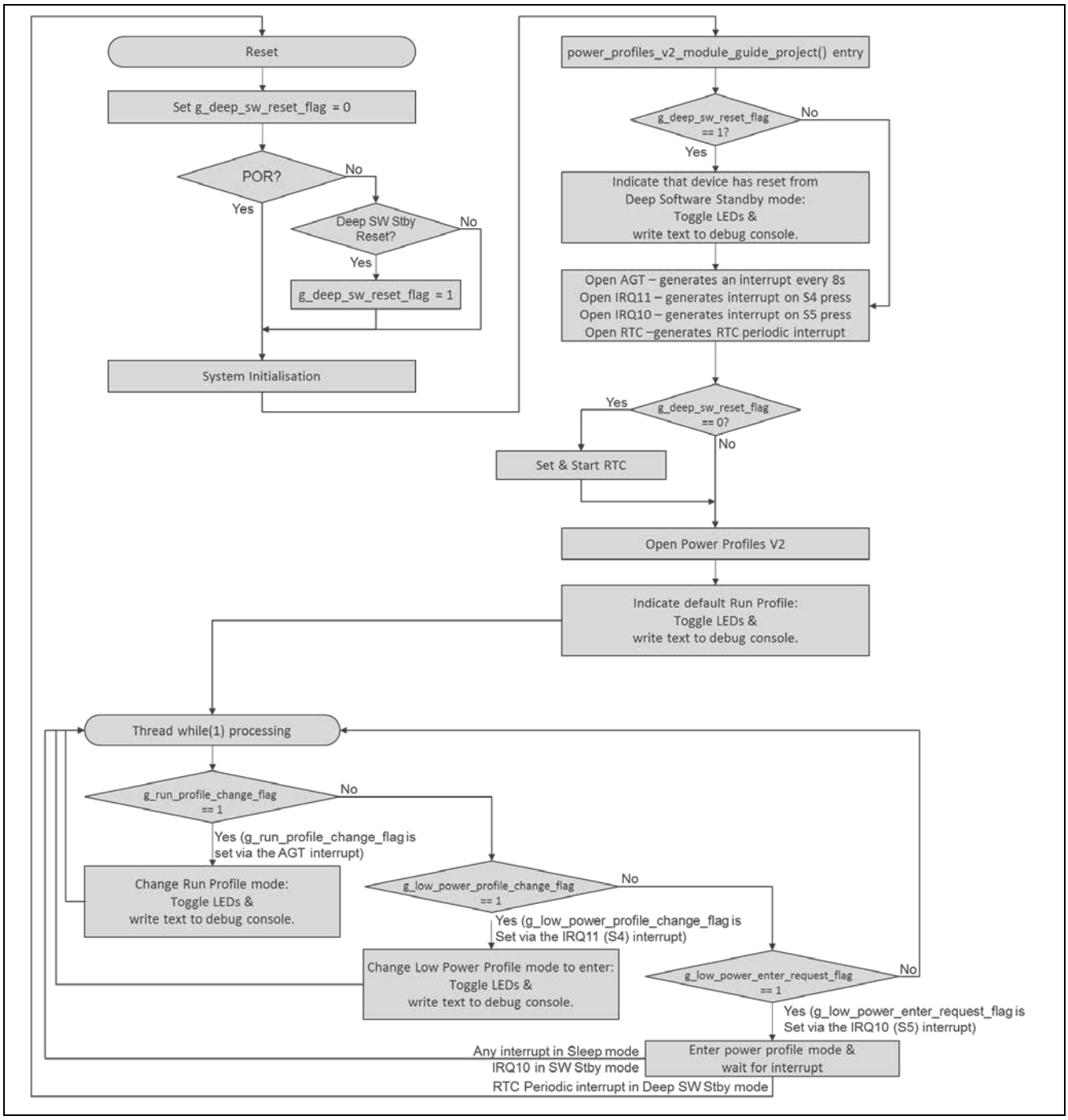

Figure 12. Power Profiles V2 Framework Module Application Project Flow

When executing, the application project cycles the S7G2 device through seven different Power Profile Run modes and allows three Low Power Profiles modes to be entered. The Power Profiles Run modes changes every 8.57 seconds when an AGT (Asynchronous General Purpose Timer) generates an interrupt. These seven different modes are continuously cycled. On the S7G2-SK board, press the S4 user button to select each of the three Low Power Profile modes. Once selected, pressing S5 enters the previously selected mode. Push buttons S4 and S5 generate external IRQ interrupts IRQ11 and IRQ10, respectively.

Once the Power Profile Run modes change on the SK-S7G2 board, the Green LED blinks ‘n’ number of times to indicate the mode. Additionally, a new pin configuration is applied. Each new pin configuration changes the color of SK-S7G2 LCD TFT screen. If the application is being run via the OCD, then the ISDE displays a text string in the debug console window.

When the selected Low Power Profile mode changes, the Red and Amber LEDs count in a binary sequence to indicate the mode. If the application is being run via the OCD, then the ISDE debug console window displays a text string.

Once the application has initialized all of its peripherals, it runs in a continuous while(1) loop, waiting on one of three software flags to be set: `g_run_profile_change_flag`, `g_low_power_profile_change_flag`, or `g_low_power_enter_request_flag`.

- g_run_profile_change_flag is set via the AGT interrupt.
- g_low_power_profile_change_flag is set via the IRQ11 interrupt – S4 has been pressed
- g_low_power_enter_request_flag is set via the IRQ10 interrupt – S5 has been pressed

The following table lists the seven Power Profile Run modes that change the clocks and clock dividers are running at on the S7G2 device.

Table 15. Power Profile Run Mode names and clock frequencies

Power Profile Run Mode	Clock(*1)	ICLK	PCLKA	PCLKB	PCLKC	PCLKD	BLCK	FLCK
g_sf_power_profiles_v2_run_PLL_240	XTAL+PLL	240M	120M	60M	60M	120M	120M	60M
g_sf_power_profiles_v2_run_PLL_240_DIV_4	XTAL+ PLL	60M	60M	60M	60M	60M	60M	60M
g_sf_power_profiles_v2_run_HOCO_20	HOCO	20M	20M	20M	20M	20M	20M	20M
g_sf_power_profiles_v2_run_HOCO_20_DIV_2	HOCO	10M	10M	10M	10M	10M	10M	10M
g_sf_power_profiles_v2_run_MOCO_8	MOCO	8M	8M	8M	8M	8M	8M	8M
g_sf_power_profiles_v2_run_MOCO_8_DIV_8	MOCO	1M	1M	1M	1M	1M	1M	1M
g_sf_power_profiles_v2_run_LOCO_32k	LOCO	32k	32k	32k	32k	32k	32k	32k

Note (1) Clock refers to the clock source that is divided down to generate the listed clocks. In all Power Profile Run Modes the XTAL and PLL continue to run to provide the LCD clock for the TFT.

The following table lists the three, selectable Low Power Profile modes used to place the device in Low Power Mode.

Table 16. Low Power Profile Names and interrupt wake up source

Power Profile Run Mode	Low Power Mode	Wake up source
g_sf_power_profiles_v2_low_power_sleep	Sleep Mode	Any interrupt
g_sf_power_profiles_v2_low_power_sw_stby	Software Standby Mode	IRQ11 (user Button S4)
g_sf_power_profiles_v2_low_power_deep_sw_stby	Deep Software Standby Mode	RTC Alarm

The effect that each mode has on current consumption can be measured via J31 on the S7G2-SK. Either measure the voltage drop across the 50m resistor (R113) via J31, or remove R113 and measure the current consumption directly placing a current meter in series with J31.

When the power profiles module guide is imported into the ISDE, the application consists of the following source files: power_profiles_v2_module_guide_project.c, user_debug_console.c, user_rtc.c, user_warm_start.c, user_lcd.c and accompanying header files.

The only file that needs to be studied understand the Power Profiles V2 API is power_profiles_v2_module_guide_project.c

The accompanying files provide other functionality to make the application run and do not need to be studied.

The first section of power_profiles_v2_module_guide_project.c includes the header files required by the application:

`stdio.h`, `hal_data.h`, `power_profiles_v2_module_guide_project.h`, `user_debug_console.h`, and `user_rtc.h & user_lcd.h`

Following this section are several global variables used by the application.

The first, `extern volatile uint8_t g_deep_sw_reset_flag;` is defined in `user_warm_start.c` and is used by the application to determine if the device has come out of Deep Software Standby Mode.

Following this section are the variables, `run_profiles_t g_run_profile` and `low_power_profiles_t g_low_power_profile`, identifying the mode the device is running in.

Following this section are the software flags set in response to the AGT interrupt or external IRQ interrupts.

Next section has two string arrays that contain the text that is sent to the debug console.

The following section is the entry function for the main program control section. It acquires information about the board's LEDs via the BSP function R_BSP_LedsGet.

The next step is to check the reset source. If the device was previously in Deep Software Standby state, then when woken from this power saving state, a Deep Software Standby Reset occurs. Applications may need to know this and do appropriate processing, different from a Pin Reset or Power On Reset. In this application the reset source is determined as soon as the device comes out of reset in the function void R_BSP_WarmStart (`bsp_warm_start_event_t event`) the file `user_warm_start.c`. If a Deep Software Reset is detected then the global flag g_deep_sw_reset_flag will have been set and if so, the board LEDs will blink indicating this. If the application is being executed via the OCD, then a text string will be sent to the debug console.

Following this area, the other peripherals that are required by the application are opened. These are the AGT, IRQ11, IRQ10, and RTC. When the RTC is opened, the time should be set. This setting only occurs if the device has not come out of Deep Software Standby mode. The assumption is the RTC time should be maintained through a power saving mode. The RTC should only be set when the application starts for the first time, or when a Power On reset or Pin Reset is detected.

Next, the Power Profile Framework is opened.

Next, is the application processing while(1) loop.

The application waits via a `while()` statement for one of three software flags to be set - `g_run_profile_change_flag`, `g_low_power_profile_change_flag`, or `g_low_power_enter_request_flag`.

If `g_run_profile_change_flag` is set, the Power Profile mode is changed and via a switch statement, the appropriate power profile is applied using the API call `g_sf_power_profiles_v2_common.p_api->runApply()`

If an error in applying the run profile occurs, a message is sent to the console and execution of the application stops. If there is no error, a message identifying the new mode is sent to the console and the green LED blinks to show the mode. The LED blinks one to seven times, depending on the mode.

In this application project, each of the run profiles has an individual pin configuration and this pin configuration is applied when the profile changes. The pin configuration changes the level of the LCD data lines, changing the displayed LCD color. This change gives the user a visual indication of the different pin configurations being applied.

If `g_low_power_profile_change_flag` is set, the Low Power Profile that can be entered is changed; through a switch statement, the chosen state is indicated via a text string to the console, as well as the red and amber LEDs illuminated. There are no power profile APIs associated to the part of the application processing.

If `g_low_power_enter_request_flag` is set, the selected Low Power Profile is entered via an API call, `g_sf_power_profiles_v2_common.p_api->lowPowerApply()`.

If there is an error in applying the low power profile, an error message is sent to the console and execution of the application stops. If there is no error, a message identifying the new low power mode is sent to the console and the green LED blinks to show the mode.

Following the main program control section are the callback functions. The application uses five callbacks, three of these are called in response to peripheral interrupts from the AGT, IRQ10, and IRQ11 interrupts. These callbacks set the software flags, which the application is waiting on, as described previously.

The other two callbacks are associated with Low Power Profiles:
`g_sf_power_profiles_v2_low_power_sleep` and
`g_sf_power_profiles_v2_low_power_sw_stby`.

If specified, the callback is called before and after Low Power Mode. In these two callbacks, the LEDs are toggled to show that the S7G2 board is going into and coming out of a Low Power Mode.

In addition to `power_profiles_v2_module_guide_project.c`, the project imports `user_debug_console.c`, `user_rtc.c`, `user_warm_start.c`, `user_lcd.c` as well as the accompanying header files. As the names suggest, user_debug_console.c contains functions that write test to the ISDE debug console and user_rtc.c contains functions associated with the Real Time Clock. The RTC is used to wake the device from Deep Software Standby mode, via the RTC Alarm interrupt. When the application requests the device should enter Deep Software Standby mode, it configures the RTC Alarm to "current RTC time + 10 seconds". The device is in Deep Software Standby mode for 10 seconds. User_warm_start.c contains a user-defined version of void R_BSP_WarmStart (bsp_warm_start_event_t event) that overrides the default function declaration called by the BSP when the device comes out of reset. User_lcd.c contains functions that enable the LCD TFT on the board. No text or images are written to the display and it is only used to show the change in I/O ports because of the run profile change.

The LCD TFT shows red, green, yellow, blue, purple, cyan and white, depending on the pin configuration in states.

The module application project uses several SSP modules to create the application. The following tables detail the settings for each module. Highlighted text indicates where changes from the default value were made.

Table 17. AGT Module Configuration Settings for the Application Project

ISDE Property	Value
Name	g_agt1
Parameter Checking	Default (BSP)
Channel	1
Mode	Periodic
Period Value	280864 (*1)
Period Unit	Raw Counts
Auto Start	True
Count Source	LOCO
AGTO Output Enabled	False
AGTIO Output Enabled	False
Output Inverted	False
Enable comparator A Output pin	False
Enable comparator B Output pin	False
Callback	cb_agt1
Underflow Interrupt Priority	Priority 8

Note: (1) It is required to generate an interrupt with a period of 8.57s. As it is not possible to enter fractional values in the Period value field, we use a RAW count value. Even though the AGT is a 16-bit timer, if a value > 65535 is entered the driver will use the AGT clock dividers to calculate the register values.

Table 18. Run Profile Configuration Settings for the Application Project

ISDE Property	Value
Name	g_sf_power_profiles_v2_run_PLL_240
Pin Configuration table	g_pin_cfg_pll_240(*1)

Note: (1):For how to create pin configuration tables, see To generate a new pin configuration.

Table 19. Run Profile CGC Configuration Instance Settings for the Application Project

ISDE Property	Value
Name	g_cgc_cfg_pll_240
System Clock	PLL
LOCO State Change	Start
MOCO State Change	Stop
HOCO State Change	Stop

ISDE Property	Value
Sub-Clock State Change	Stop
Main Clock State Change	Start
PLL State Change	Start
PLL Clock Source	Main Oscillator
PLL Divisor	2
PLL Multiplier	20.0
PCLKA Divisor	2
PCLKB Divisor	4
PCLKC Divisor	4
PCLKD Divisor	2
BCLK Divisor	2
FCLK Divisor	4
ICLK Divisor	1

Table 20. Run Profile Configuration Settings for the Application Project

ISDE Property	Value
Name	g_sf_power_profiles_v2_run_PLL_240_DIV_4
Pin Configuration table	g_pin_cfg_pll_240_div_4

Table 21. Run Profile CGC Configuration Instance Settings for the Application Project

ISDE Property	Value
Name	g_cgc_cfg_pll_240_div_4
System Clock	PLL
LOCO State Change	None
MOCO State Change	None
HOCO State Change	None
Sub-Clock State Change	None
Main Clock State Change	None
PLL State Change	None
PLL Clock Source	Main Oscillator
PLL Divisor	2
PLL Multiplier	20.0
PCLKA Divisor	4
PCLKB Divisor	4
PCLKC Divisor	4
PCLKD Divisor	4
BCLK Divisor	4
FCLK Divisor	4
ICLK Divisor	4

Table 22. Run Profile Configuration Settings for the Application Project

ISDE Property	Value
Name	g_sf_power_profiles_v2_run_HOCO_20
Pin Configuration table	g_pin_cfg_hoco_20

Table 23. Run Profile CGC Configuration Instance Settings for the Application Project

ISDE Property	Value
Name	g_cgc_cfg_hoco_20

ISDE Property	Value
System Clock	HOCO
LOCO State Change	None
MOCO State Change	Stop
HOCO State Change	Start
Sub-Clock State Change	Stop
Main Clock State Change	Start
PLL State Change	None
PLL Clock Source	Main Oscillator
PLL Divisor	2
PLL Multiplier	20.0
PCLKA Divisor	1
PCLKB Divisor	1
PCLKC Divisor	1
PCLKD Divisor	1
BCLK Divisor	1
FCLK Divisor	1
ICLK Divisor	1

Table 24. Run Profile Configuration Settings for the Application Project

ISDE Property	Value
Name	g_sf_power_profiles_v2_run_HOCO_20_DIV_2
Pin Configuration table	g_pin_cfg_hoco_20_div_2

Table 25. Run Profile CGC Configuration Instance Settings for the Application Project

ISDE Property	Value
Name	g_cgc_cfg_hoco_20_div_2
System Clock	HOCO
LOCO State Change	None
MOCO State Change	None
HOCO State Change	None
Sub-Clock State Change	None
Main Clock State Change	None
PLL State Change	None
PLL Clock Source	Main Oscillator
PLL Divisor	2
PLL Multiplier	20.0
PCLKA Divisor	2
PCLKB Divisor	2
PCLKC Divisor	2
PCLKD Divisor	2
BCLK Divisor	2
FCLK Divisor	2

Table 26. Run Profile Configuration Settings for the Application Project

ISDE Property	Value
Name	g_sf_power_profiles_v2_run_MOCO_8
Pin Configuration table	g_pin_cfg_moco_8

Table 27. Run Profile CGC Configuration Instance Settings for the Application Project

ISDE Property	Value
Name	g_cgc_cfg_moco_8
System Clock	MOCO
LOCO State Change	None
MOCO State Change	Start
HOCO State Change	Stop
Sub-Clock State Change	Stop
Main Clock State Change	Start
PLL State Change	None
PLL Clock Source	Main Oscillator
PLL Divisor	2
PLL Multiplier	20.0
PCLKA Divisor	1
PCLKB Divisor	1
PCLKC Divisor	1
PCLKD Divisor	1
BCLK Divisor	1
FCLK Divisor	1

Table 28. Run Profile Configuration Settings for the Application Project

ISDE Property	Value
Name	g_sf_power_profiles_v2_run_MOCO_8_DIV_8
Pin Configuration table	g_pin_cfg_moco_8_div_8

Table 29. Run Profile CGC Configuration Instance Settings for the Application Project

ISDE Property	Value
Name	g_cgc_cfg_moco_8_div_8
System Clock	MOCO
LOCO State Change	None
MOCO State Change	None
HOCO State Change	None
Sub-Clock State Change	None
Main Clock State Change	None
PLL State Change	None
PLL Clock Source	Main Oscillator
PLL Divisor	2
PLL Multiplier	20.0
PCLKA Divisor	8
PCLKB Divisor	8
PCLKC Divisor	8
PCLKD Divisor	8
BCLK Divisor	8
FCLK Divisor	8

Table 30. Run Mode LOCO_32k Configuration Settings for Application Project

ISDE Property	Value
Name	g_sf_power_profiles_v2_run_LOCO_32k
Pin Configuration table	g_pin_cfg_loco_32

Table 31. Run Profile CGC Configuration Instance Settings for Application Project

ISDE Property	Value
Name	g_cgc_cfg_loco_32k
System Clock	LOCO
LOCO State Change	Start
MOCO State Change	Stop
HOCO State Change	Stop
Sub-Clock State Change	Stop
Main Clock State Change	Start
PLL State Change	None
PLL Clock Source	Main Oscillator
PLL Divisor	2
PLL Multiplier	20.0
PCLKA Divisor	1
PCLKB Divisor	1
PCLKC Divisor	1
PCLKD Divisor	1
BCLK Divisor	1
FCLK Divisor	1

Table 32. External IRQ 11 Configuration Settings for Application Project

ISDE Property	Value
Parameter Checking	Default (BSP)
Name	g_external_irq11
Channel	11
Trigger	Falling
Digital Filtering	Enabled
Digital Filtering Sample Clock	PLCK/64
Interrupt enabled after initialization	True
Callback	cb_external_irq11
Pin Interrupt Priority	Priority 8

Table 33. External IRQ 10 Configuration Settings for Application Project

ISDE Property	Value
Name	g_external_irq10
Parameter Checking	Default (BSP)
Channel	10
Trigger	Falling
Digital Filtering	Enabled
Digital Filtering Sample Clock	PLCK/64
Interrupt enabled after initialization	True
Callback	cb_external_irq10
Pin Interrupt Priority	Priority 8

Table 34. Low Power Profile – Sleep – Configuration Settings for the Application Project

ISDE Property	Value
Name	g_sf_power_profiles_v2_low_power_sleep
Callback	cb_low_power_sleep
Low power entry pin configuration table	NULL
Low power exit pin configuration table	NULL

Table 35. Low Power LPM V2 Driver – Low Power Mode Sleep on r_lpmv2 – Configuration Settings for the Application Project

ISDE Property	Value
Name	g_lpmv2_sleep0

Table 36. Low Power Profile – Software Standby – Configuration Settings for the Application Project

ISDE Property	Value
Name	g_sf_power_profiles_v2_low_power_sw_stby
Callback	cb_low_power_sw_stby
Low power entry pin configuration table	NULL
Low power exit pin configuration table	NULL

Table 37. Low Power LPM V2 Driver – Low Power Mode Standby on r_lpmv2 – Configuration Settings for the Application Project

ISDE Property	Value
Name	g_lpmv2_standby0
Choose the low power mode	Standby
Output port state in standby and deep standby	No change
Select Standby Exit Sources	
IRQ0	Disabled
IRQ1	Disabled
IRQ2	Disabled
IRQ3	Disabled
IRQ4	Disabled
IRQ5	Disabled
IRQ6	Disabled
IRQ7	Disabled
IRQ8	Disabled
IRQ9	Disabled
IRQ10	Disabled
IRQ11	Enabled
IRQ12	Disabled
IRQ13	Disabled
IRQ14	Disabled
IRQ15	Disabled
IWDT	Disabled
Key Interrupt	Disabled
LVD1 Interrupt	Disabled
LVD2 Interrupt	Disabled
Analog Comparator High-speed 0 Interrupt	Disabled
RTC Alarm	Disabled
RTC Period	Disabled

ISDE Property	Value
USB High Speed	Disabled
USB Full Speed	Disabled
AGT1 Underflow	Disabled
AGT1 Compare Match A	Disabled
AG1 Compare Match B	Disabled
I2C0	Disabled

Table 38. Low Power Profile – Deep Software Standby – Configuration Settings for the Application Project

ISDE Property	Value
Name	g_sf_power_profiles_v2_low_power_deep_sw_stby
Callback	NULL
Low power entry pin configuration table	NULL
Low power exit pin configuration table	NULL

Table 39. Low Power LPM V2 Driver – Low Power Mode Standby on r_lpmv2 – Configuration Settings for the Application Project

ISDE Property	Value
Name	g_lpmv2_deep_standby0
Output port state in standby and deep standby	No change
Maintain or reset IO port states on exit from deep standby mode	Maintain the IO port states
Internal power supply control in deep standby mode	Maintain the internal power supply
Deep Standby Cancel Sources/Edges	
IRQ0	Disabled
IRQ0 edge	Disabled
IRQ1	Disabled
IRQ1 edge	Disabled
IRQ2	Disabled
IRQ2 edge	Disabled
IRQ3	Disabled
IRQ3 edge	Disabled
IRQ4	Disabled
IRQ4 edge	Disabled
IRQ5	Disabled
IRQ5 edge	Disabled
IRQ6	Disabled
IRQ6 edge	Disabled
IRQ7	Disabled
IRQ7 edge	Disabled
IRQ8	Disabled
IRQ8 edge	Disabled
IRQ9	Disabled
IRQ9 edge	Disabled
IRQ10	Disabled
IRQ10 edge	Disabled
IRQ11	Disabled
IRQ11 edge	Disabled
IRQ12	Disabled

ISDE Property	Value
IRQ12 edge	Disabled
IRQ13	Disabled
IRQ13 edge	Disabled
IRQ14	Disabled
IRQ14 edge	Disabled
IRQ15	Disabled
IRQ15 edge	Disabled
LVD1	Disabled
LVD1 edge	Disabled
LVD2	Disabled
LVD2 edge	Disabled
RTC Interval	Disabled
RTC Alarm	Enabled
NMI	Disabled
NMI edge	Disabled
USB High Speed	Disabled
USB Full Speed	Disabled
AGT1	Disabled

Table 40. RTC Configuration Settings for the Application Project

ISDE Property	Value
Parameter Checking	Default (BSP)
Name	g_rtc0
Clock Source	LOCO
Configure RTC hardware in open() call	No
Error Adjustment Value [DEPRECATED]	0
Error Adjustment Type [DEPRECATED]	None
Callback	NULL
Alarm Interrupt Priority	Priority 8
Period Interrupt Priority	Disabled
Carry Interrupt Priority	Priority 12

Table 41. GLCDC Configuration Settings for the Application Project

ISDE Property	Value
Parameter Checking	Default: BSP
Name	g_display
Name of display callback function to be defined by user	NULL
Input – Panel clock source select	Internal clock(GLCDCLK)
Input – Graphics screen1	Used
Input – Graphics screen1 frame buffer name	fb_background
Input – Number of Graphics screen1 frame buffer	2
Input – section where Graphics screen1 frame buffer allocated	bss
Input – Graphics screen1 input horizontal size	240
Input – Graphics screen1 vertical size	320
Input – Graphics screen1 input horizontal stride (not bytes but pixels)	256
Input – Graphics screen1 input format	16 bits RGB565

ISDE Property	Value
Input – Graphics screen1 input line descending	Not used
Input – Graphics screen1 input line repeat	Off
Input – Graphics screen1 input line repeat times	0
Input – Graphics screen1 layer coordinate X	0
Input – Graphics screen1 layer coordinate Y	0
Input – Graphics screen1 layer background color alpha	255
Input – Graphics screen1 layer background color Red	255
Input – Graphics screen1 layer background color Green	255
Input – Graphics screen1 layer background color Blue	255
Input – Graphics screen1 layer fading control	None
Input – Graphics screen1 layer fade speed	0
Input – Graphics screen2	Not used
Input – Graphics screen2 frame buffer name	fb_foreground
Input – Number of Graphics screen2 frame buffer	2
Input – section where Graphics screen2 frame buffer allocated	sdram
Input – Graphics screen2 input horizontal size	800
Input – Graphics screen2 vertical size	480
Input – Graphics screen2 input horizontal stride (not bytes but pixels)	800
Input – Graphics screen2 input format	16 bits RGB565
Input – Graphics screen2 input line descending	Off
Input – Graphics screen2 input line repeat	Off
Input – Graphics screen2 input line repeat times	0
Input – Graphics screen2 layer coordinate X	0
Input – Graphics screen2 layer coordinate Y	0
Input – Graphics screen2 layer background color alpha	255
Input – Graphics screen2 layer background color Red	255
Input – Graphics screen2 layer background color Green	255
Input – Graphics screen2 layer background color Blue	255
Input – Graphics screen2 layer fading control	None
Input – Graphics screen2 layer fade speed	0
Output – Horizontal total cycles	320
Output – Horizontal active video cycles	240
Output – Horizontal back porch cycles	6
Output – Horizontal sync signal cycles	4
Output – Horizontal sync signal polarity	Low active
Output – Vertical total lines	328
Output – Vertical active video lines	320
Output – Vertical back porch lines	4
Output – Vertical sync signal lines	4
Output – Vertical sync signal polarity	Low active
Output – Format	16 bits RGB565
Output – Endian	Little endian
Output – Color order	RGB
Output – Data Enable Signal Polarity	High active

ISDE Property	Value
Output – Sync edge	Rising Edge
Output – Background color alpha channel	255
Output – Background color R channel	0
Output – Background color G channel	0
Output – Background color B channel	0
CLUT	Not used
CLUT - CLUT buffer size	256
TCON – Hsync pin select	LCD_TCON2
TCON – Vsync pin select	LCD_TCON1
TCON – DataEnable pin select	LCD_TCON0
TCON – Panel clock division ratio	1/32
Color correction – Brightness	Off
Color correction – Brightness R channel	128
Color correction – Brightness G channel	128
Color correction – Brightness B channel	128
Color correction – Contrast	Off
Color correction – Contrast(gain) R channel	128
Color correction – Contrast(gain) G channel	128
Color correction – Contrast(gain) B channel	128
Color correction – Gamma correction(Red)	Off
Color correction – Gamma gain R[0-15]	0
Color correction – Gamma threshold R[0-15]	0
Color correction – Gamma correction(Green)	Off
Color correction – Gamma gain G[0-15]	0
Color correction – Gamma threshold G[0-15]	0
Color correction – Gamma correction(Blue)	Off
Color correction – Gamma gain B[0-15]	0
Color correction – Gamma threshold B[0-15]	0
Dithering	Off
Dithering – Mode	Truncate
Dithering – Pattern A	Pattern 11
Dithering – Pattern B	Pattern 11
Dithering – Pattern C	Pattern 11
Dithering – Pattern D	Pattern 11
Misc – Correction Process Order	Brightness and Contrast then Gamma
Line Detect Interrupt Priority	Disabled
Underflow 1 Interrupt Priority	Disabled
Underflow 2 Interrupt Priority	Disabled

Table 42. SCI_SPI Configuration Settings for the Application Project

ISDE Property	Value
Parameter Checking	Default (BSP)
Name	g_lcd_spi
Channel	0
Operating Mode	Master
Clock Phase	Data sampling on odd edge, data variation on even edge
Clock Polarity	Low when idle
Mode Fault Error	Disable
Bit Order	MSB First
Bitrate	100000
Bit Rate Modulation Enable	Enable
Callback	NULL
Receive Interrupt Priority	Priority 12
Transmit Interrupt Priority	Priority 12
Transmit End Interrupt Priority	Priority 12
Error Interrupt Priority	Priority 12

To generate a new pin configuration, use the following steps.

1. Make a copy of the default .pincfg file in the project directory. The default .pincfg file can be determined from the Pins Configuration tab.

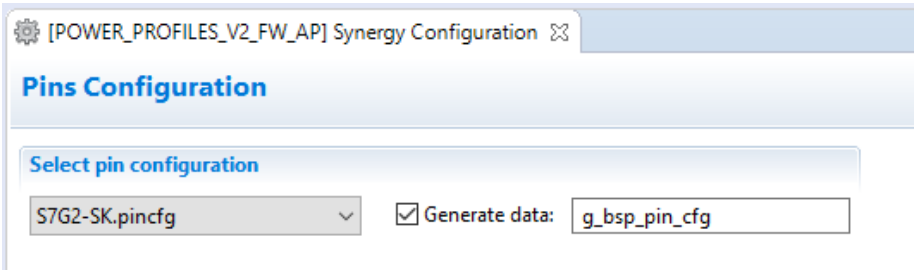

Figure 13. Default .pincfg file on Pins Configuration tab

In this case, S7G2-SK.pincfg is the default.

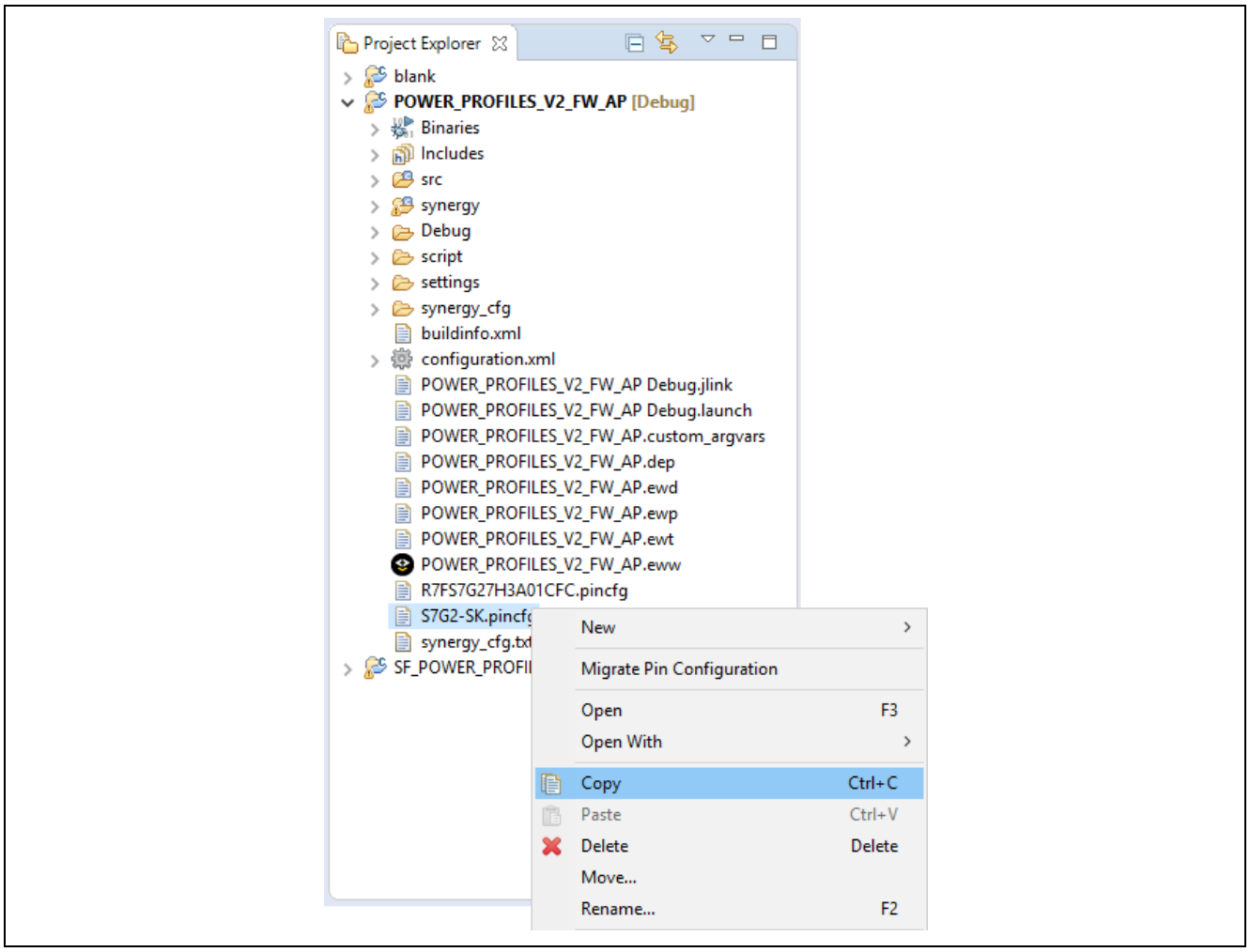

Figure 14. Copying the .pincfg file

When prompted, create a new file name after you paste the **.pincfg** file into the directory.

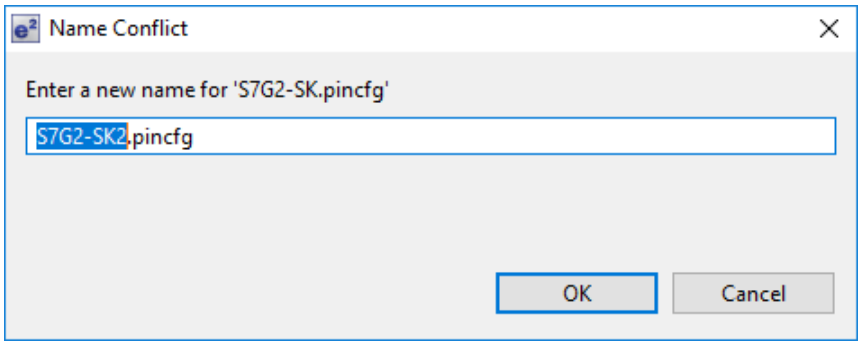

Figure 15. Creating a new .pincfg file name

For example, S7G2-SK_PLL_240.pincfg

In the Pins Configuration window, select the pin configuration, then select **Generate data** and give the pin configuration a unique name, as the following example shows. Make sure the names in the Generate data field match the ones in the Pin configuration table of Power Profiles V2 Run Profile modules.

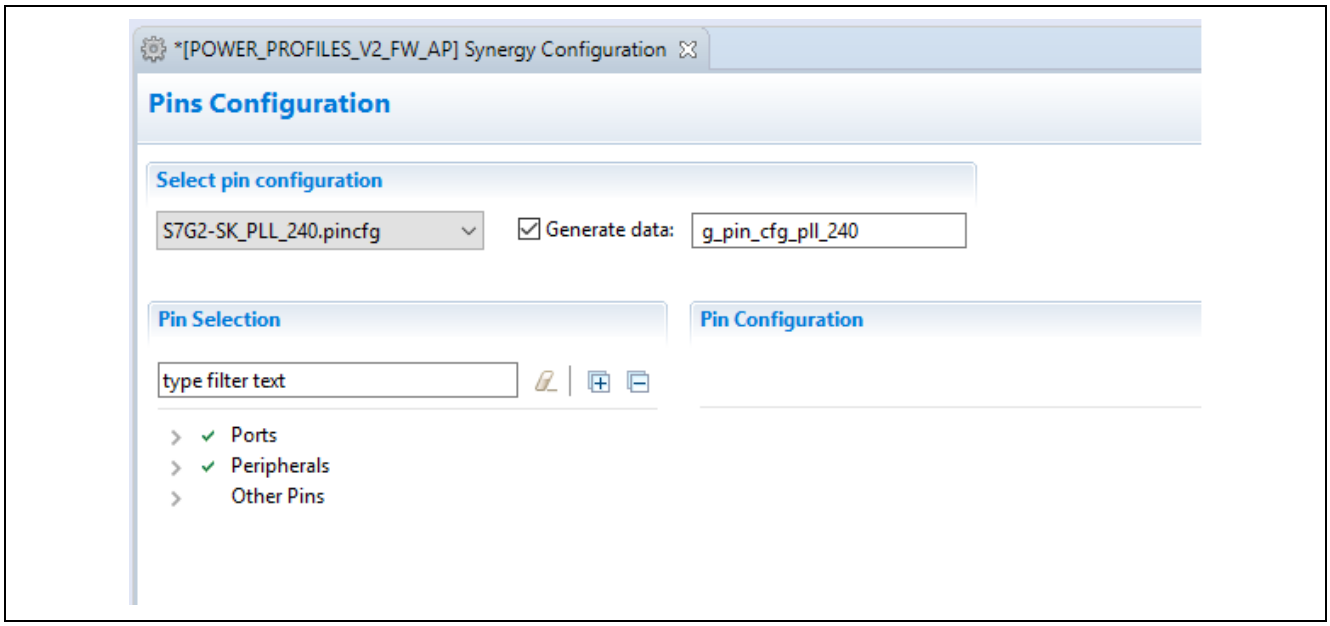

Figure 16. Configuring the new pin

Configure the pins, as required. This newly created pin configuration and label can be used in the Power Profile V2 configurations to change the pin configuration for each mode.

In this application project, there are seven pin configurations used, one for each Power Profile Run mode. The following table lists the output pin variations from the default pin configuration for each mode.

Changing a pin configuration changes the display color on the LCD TFT; the color coding in the table indicates the result for each pin output.

Table 43. Pin Variation from the default for each pin configuration

	g_bsp_pin_cfg_data (Pin configuration after BSP init which is called on Reset)	g_pin_cfg_pll_240_data	g_pin_cfg_pll_240_div_4_data	g_pin_cfg_hoco_20_data	g_pin_cfg_hoco_20_div_2_data	g_pin_cfg_moco_8_data	g_pin_cfg_moco_8_div_8_data	g_pin_cfg_loco_32_data
PORT_06_PIN_00 (LED1)	Low	disabled	disabled	disabled	disabled	disabled	disabled	disabled
PORT_06_PIN_01 (LED2)	High	disabled	disabled	disabled	disabled	disabled	disabled	disabled
PORT_06_PIN_02 (LED3)	High	disabled	disabled	disabled	disabled	disabled	disabled	disabled
PORT_06_PIN_06	Low	Low	Low	Low	High	High	High	High
PORT_06_PIN_07	Low	Low	Low	Low	High	High	High	High
PORT_06_PIN_10 (TFT LCD Reset)	Low	High	High	High	High	High	High	High
PORT_06_PIN_15	Low	Low	High	High	Low	Low	High	High
PORT_08_PIN_02	Low	Low	Low	Low	High	High	High	High
PORT_08_PIN_03	Low	Low	Low	Low	High	High	High	High
PORT_08_PIN_04	Low	Low	Low	Low	High	High	High	High
PORT_09_PIN_01	Low	High	Low	High	Low	High	Low	High

	g_bsp_pin_cfg_data (Pin configuration after BSP init which is called on Reset)	g_pin_cfg_pll_240_data	g_pin_cfg_pll_240_div_4_data	g_pin_cfg_hoco_20_data	g_pin_cfg_hoco_20_div_2_data	g_pin_cfg_moco_8_data	g_pin_cfg_moco_8_div_8_data	g_pin_cfg_loco_32_data
PORT_09_PIN_05	Low	High	Low	High	Low	High	Low	High
PORT_09_PIN_06	Low	High	Low	High	Low	High	Low	High
PORT_09_PIN_07	Low	High	Low	High	Low	High	Low	High
PORT_09_PIN_08	Low	High	Low	High	Low	High	Low	High
PORT_10_PIN_00	Low	Low	High	High	Low	Low	High	High
PORT_10_PIN_01	Low	Low	High	High	Low	Low	High	High
PORT_10_PIN_08	Low	Low	High	High	Low	Low	High	High
PORT_10_PIN_09	Low	Low	High	High	Low	Low	High	High
PORT_10_PIN_10	Low	Low	High	High	Low	Low	High	High

8. Customizing the Power Profiles V2 Framework Module for a Target Application

It can be seen from the configuration tables in section 7 that the Power Profile V2 Framework provides multiple configuration settings to match your application requirements. The Run Profile allows for multiple clock configurations to be set and switching between these is easily achieved with a simple API call. Similarly, the Low Power Profiles allow for multiple wake up sources to be configured. This module application project demonstrates only a fraction of the total possible combinations.

9. Running the Power Profiles V2 Framework Module Application Project

To run the Power Profiles V2 Framework module application project and to see it executing on a target kit, import it into your ISDE, compile and run debug. Refer to the *Renesas Synergy Project Import Guide* (r11an0023eu0121-synergy-ssp-import-guide.pdf), included with this application project, for instructions on importing the project into e^2 studio or IAR EW for Synergy, and building/running the application.

The application project can place the S7G2 device into Software Standby mode and Deep Software Standby mode. By default, when being debugged, these two modes of operation are not available.

In e² studio ISDE, enable these modes, in the project debug configurations to enable Low Power Handling and specify the script file **CM_low_power_debug.JlinkScript**.
This file is located on (your e² studio installation folder) \DebugComp\Synergy\ARM\Scripts.

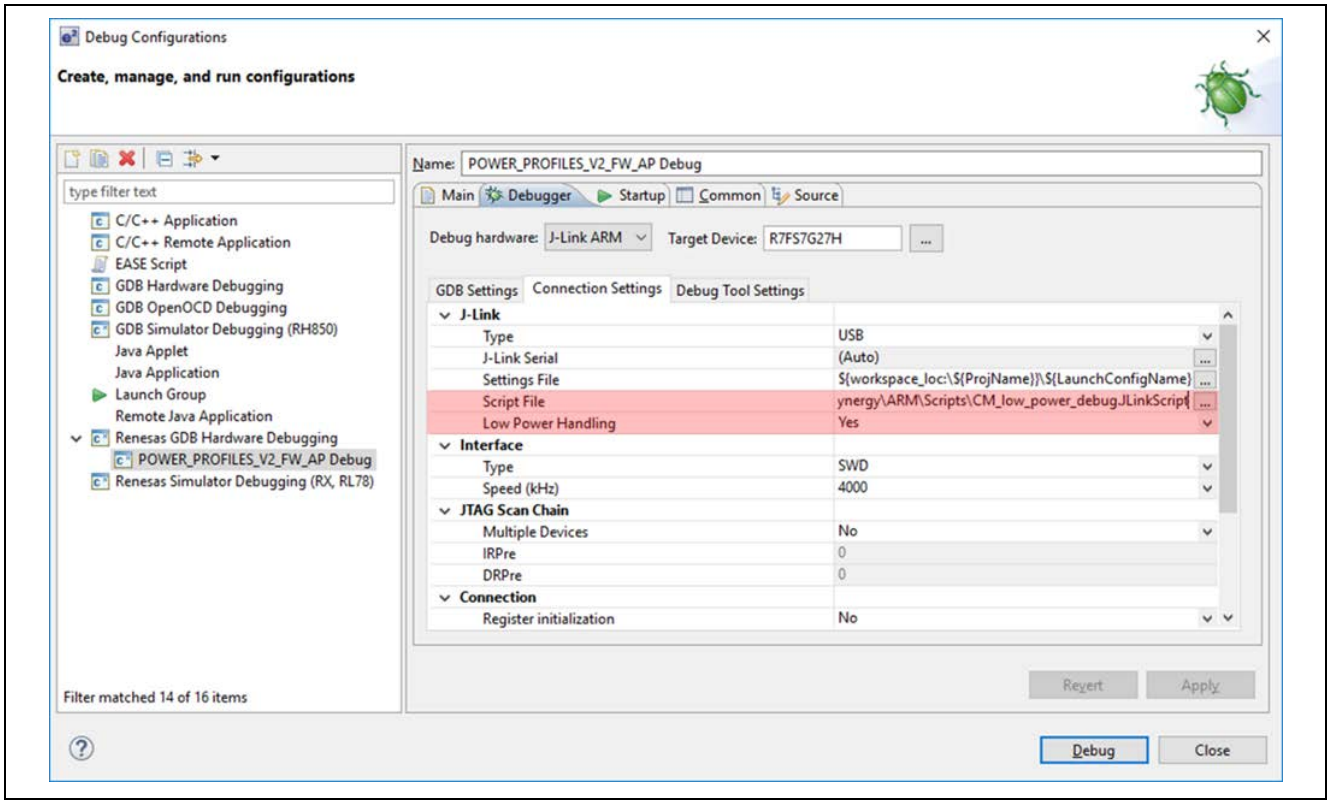

Figure 17. Enabling Low Power Handling and CM_low_power_debug.JLinkScript

In the IAR EW for Synergy, use the following steps to enable low power mode debug:

1. Enable these modes via the macro file, **low_power_debug.mac**.

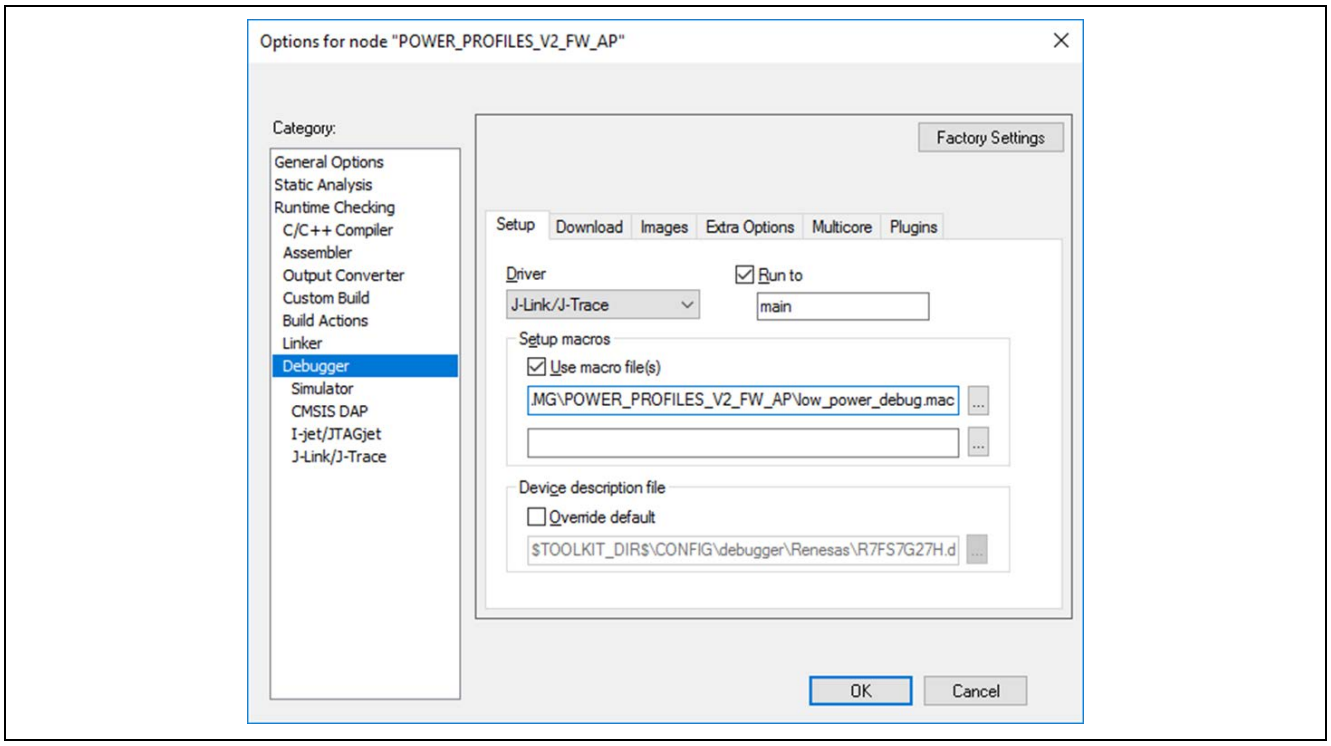

Figure 18. Enabling modes via low_power_debug.mac file

2. Open the file: settings/**LPM_V2_HAL_MG_AP_Debug.jlink** shown in the following figure.
   If the file does not exist, attach the debugger to the device and disconnect. The file is created when attaching to the device for the first time.

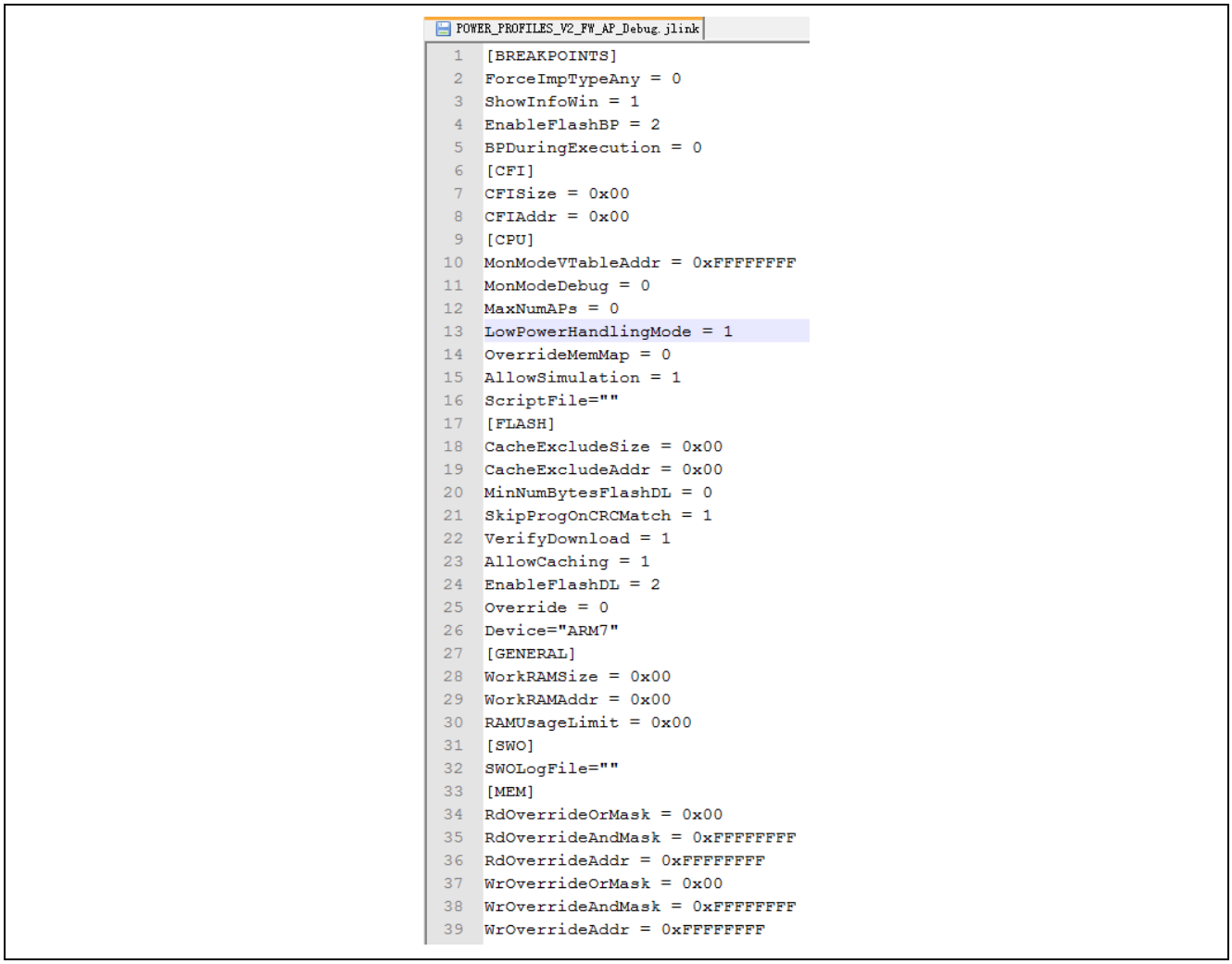

Figure 19. Enabling Low Power Handling

3. Change the following line:
   LowPowerHandling = 0 to **: LowPowerHandling = 1**
4. Save the file and debug.

Running the application project with Low Power Handling enabled allows the application to be debugged. However, the current consumption numbers are higher than normal execution, due to OCD being active. Realistic current consumption numbers can be achieved by running the device with the OCD disconnected.

Note that in Deep Software Standby mode, due to other peripheral devices on the SK-S7G2 board, a true number cannot be achieved.

For reference, the following table shows current values of Power Profiles, which are measured on J31 (removed R113) on SK-S7G2 V3.3:

Table 44. Current Values of Power Profiles

Run Profile \ Current value	Normal mode	Enter sleep mode	Enter software standby mode	Enter deep standby mode
g_sf_power_profiles_v2_run_PLL_240	49.8mA	39.1mA	2.1mA	80uA
g_sf_power_profiles_v2_run_PLL_240_DIV_4	25.1mA	22.0mA		
g_sf_power_profiles_v2_run_HOCO_20	15.4mA	14.1mA		
g_sf_power_profiles_v2_run_HOCO_20_DIV_2	13.0mA	12.4mA		
g_sf_power_profiles_v2_run_MOCO_8	12.4mA	11.8mA		
g_sf_power_profiles_v2_run_MOCO_8_DIV_8	10.8mA	10.7mA		
g_sf_power_profiles_v2_run_LOCO_32k	10.5mA	10.4mA		

10. Power Profiles V2 Framework Module Conclusion

This module guide has provided all the background information needed to select, add, configure, and use the module in an example project. Many of these steps were time consuming and error-prone activities in previous generations of embedded systems. The Renesas Synergy Platform makes these steps much less time consuming and removes the common errors like conflicting configuration settings or incorrect selection of low-level modules. The use of high-level APIs (in the application project) demonstrate the development-time savings in allowing work to begin at a high level and avoiding the time required in older development environments to use, or, in some cases, create, low level drivers.

11. Power Profiles V2 Framework Module Next Steps

For ease of demonstration, this module application project was created as a HAL project. The Power Profile V2 Framework can be used in a Threaded RTOS application.

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

12. Power Profiles V2 Framework Module Reference Information

SSP User Manual: Available in HTML format at www.renesas.com/us/en/products/synergy/software/ssp.html as a SSP distribution package, and also as a pdf from the Synergy Gallery.

Links to all the most up-to-date **sf_power_profiles_v2** module reference materials and resources are available on the Synergy Knowledge Base: https://en-support.renesas.com/knowledgeBase/16977551.

Website and Support

Visit the following vanity URLs to learn about key elements of the Synergy Platform, download components and related documentation, and get support.

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

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Jun.12.18	—	Initial Release
1.01	Feb.22.19	—	Updated to SSP 1.5.0

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
6. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

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

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

(Rev.4.0-1 November 2017)

Corporate Headquarters

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

Trademarks

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

Contact information

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

© 2019 Renesas Electronics Corporation. All rights reserved.