Renesas Synergy™ Platform

Power Profiles v2 (PPV2) Framework for DK-S3A7

R20AN0498EU0100
Rev.1.00
Mar 7, 2018

Introduction

This application project enables you to effectively use the Power Profiles v2 (PPv2) Framework module in your own design, and reduces your effort in using the power control modes and low power modes (LPM) provided in Synergy MCUs. In this document you are first introduced to basic knowledge about Synergy MCU power control modes and LPMs, which are abstracted into power profile configurations and APIs in the PPv2. Then the document describes the PPv2 framework operational flow, and a basic implementation when using the PPv2 in applications. And lastly, this document provides an example of the PPv2 framework that uses the power control mode and LPM mode transitions.

The PPv2 Framework improves control for users to set a power control mode and the LPM mode in the Synergy MCU. The PPv2 Framework supports all the features of the Power Profiles v1 available in the current release of the SSP framework using the LPM v1 driver. However, Power Profiles v1 and PPv2 Framework are not compatible. Power Profiles v1 and PPv2 Framework cannot be used in the same project. For all new projects, it is recommended that applications use PPv2 Framework.

This document guides you as you add the described module to your own design, configure it correctly for the target application, and write code using the included application example code as a reference and efficient starting point. References to API descriptions that better illustrate advanced uses of the module are available in the SSP User Manual version 1.3.0 or later, which is a valuable resource for creating increasingly complex designs.

Required Resources

To perform the procedures in this application notes, you need:

- Renesas DK-S3A7(v2.0) kit
- Micro USB cables
- e^2 studio ISDE 5.4.0.023 or later (Renesas Electronics America Inc.)
- IAR Embedded Workbench® for Renesas Synergy™ 7.71.3 or later (IAR Systems)
- Synergy Software Package (SSP) v1.3.0 (Renesas Electronics America Inc) or later.
- Synergy Standalone Configurator (SSC) v5.4.0.023 or later.

You can download the required Renesas software from the Renesas Synergy Gallery (https://synergygallery.renesas.com).

Prerequisites

As the user of this application note, it is assumed that you have some experience with the Renesas e^2 studio ISDE and SSP. For example, before you perform the procedure in this application note, you should follow the procedure in the *SSP User's Manual* to build and run the Blinky project. By doing so, you will become familiar with e^2 studio and the SSP. This also ensures that the debug connection to your board is functioning properly.

The intended audience are users who want to develop applications using PPv2 Framework with S1/S3/S5/S7 Synergy MCU series.

Contents

1. Synergy Power Profiles Overview

1.1 Power Profile v2 Framework Overview

The Power Profiles v2 (PPv2) Framework is a component of the Synergy Software Package (SSP), and is intended to help SSP users in configuring and utilizing sophisticated power saving features of the Synergy MCU Groups. The PPv2 modules can be instantiated into ThreadX RTOS aware multithreads, and configured with the SSP configurator to the expected power control modes and LPM modes, and then inserted into application. However, users are required to understand those power saving features of the Synergy MCU Groups to properly set up the PPv2 modules.

1.1.1 PPv2 Framework Features

Some of the supported features of the PPv2 are:

- Configurable options to set different MCU power control modes with customizable clock domains
- Configurable options to set different MCU low power modes with different IO port or pin configurations
- Supports both threaded and non-threaded operations

1.2 Synergy MCU Power Profile Overview

Synergy MCU Groups utilize two sets of control registers, Power Control Modes and Low Power Modes (LPMs), to support different power or performance requirements. This section describes basic concepts and usage of these modes. For more details, refer to the specific *Synergy Microcontroller Group User's Manual* for the configuration of the control register, and the *Synergy SSP User's Manual* for APIs.

1.2.1 Synergy MCU Power Control Modes

Power consumption can be reduced in Normal or running mode by selecting an appropriate operating power control mode with different clock sources and the operating frequencies. This capability enables flexible operation and optimization of the devices across several power and performance points. The system clock can be provided externally (from a single-ended clock source) or it can be generated internally using different on-chip oscillators such as the main clock oscillator, or sub-clock oscillator. Moreover, you can adjust the System Clock (ICLK), Peripheral Module Clocks (PCLKB, PCLKD), External Bus Clock (BCLK), etc. by setting different division ratios in Clock Generation Circuit (CGC) registers. A block diagram of the clock sources for the Synergy MCU S3 series is illustrated below.

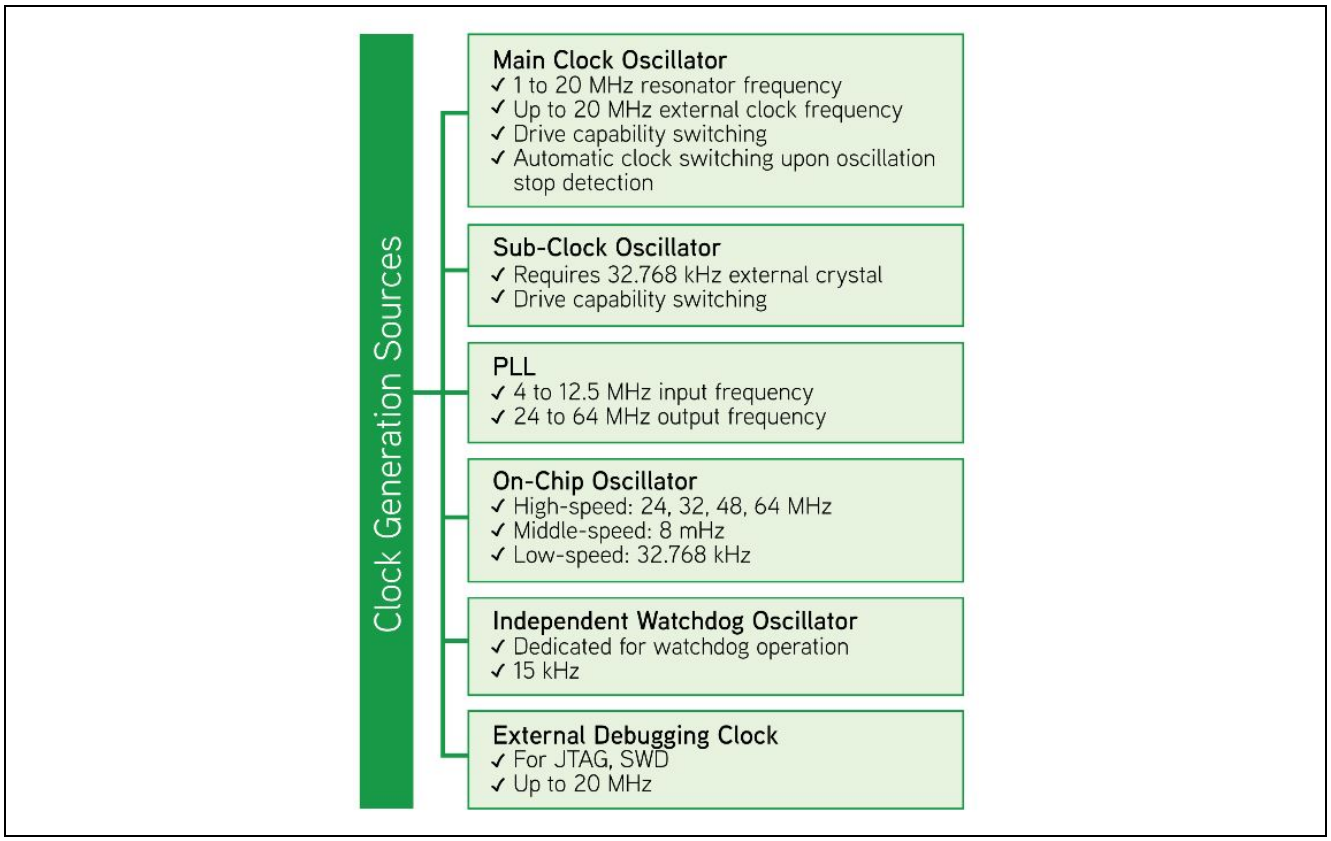

Figure 1 Clock generation source for Synergy MCU S3 Series

The Synergy MCU Groups define some power control modes, such as High-speed mode, Middle-speed mode, Low-voltage mode, Low-speed mode, and Subosc-speed mode with different clock generation sources and frequency ranges. The corresponding power consumption levels are decreased gradually. However, selecting a power control mode is a design tradeoff, where factors such as the processing time, (static and dynamic) power consumption, leakage current, and reliability should be considered.

The power control modes currently associated with the Synergy MCU Groups are listed in the following table.

Table 1 Power control modes in the following Synergy MCU Groups

	Power Control Modes				
Synergy MCU Group	**High-speed mode**	**Middle-speed mode**	**Low-voltage mode**	**Low-speed mode**	**Subosc-speed mode**
S124	x	x	x	x	x
S128	x	x	x	x	x
S3A3	x	x	x	x	x
S3A6	x	x	x	x	x
S3A7	x	x	x	x	x
S5D5	x			x	x
S5D9	x			x	x
S7G2	x			x	x

Note: Each Synergy MCU may have different power control modes definitions. The operating frequency and voltage of each power control mode is specified in the specific *Synergy Microcontroller Group User's Manual*. For example, the S3A7 MCU has 5 predefined power control modes as described in the *Synergy S3A7 Microcontroller Group User's Manual*, and the power consumption levels are shown in the following graphic.

Mode	Oscillator						
	PLL*1	High-speed on-chip oscillator	Middle-speed on-chip oscillator	Low-speed on-chip oscillator	Main clock oscillator	Sub-clock oscillator	IWDT-dedicated on-chip oscillator
High-speed	Available	Available	Available	Available	Available	Available	Available
Middle-speed	Available	Available	Available	Available	Available	Available	Available
Low-voltage	N/A	Available	Available	Available	Available	Available	Available
Low-speed	N/A	Available	Available	Available	Available	Available	Available
Subosc-speed	N/A	N/A	N/A	Available	N/A	Available	Available

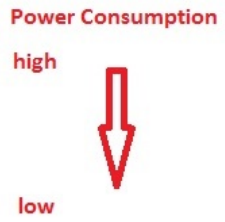

Figure 2 Power control modes and their power consumption of the S3A7 MCU

The power control modes are defined in the Synergy MCU user's manuals.

1.2.2 Synergy MCU Low Power Modes

To further reduce the power consumption, the Synergy MCUs provide users Low Power Modes (LPM) by allowing operation of a peripheral while keeping the CPU and other peripherals clock gated or powered down. There are 4 possible different LPM modes: Sleep mode, Software Standby mode, Deep Software Standby mode, and Snooze mode. As an example, the core is clock gated and peripheral are available in sleep mode. The core and most peripherals are clock gated but data retained in Software Standby mode. In addition, Renessa Synergy™ S7 Series and S5 Series MCUs have Deep Software Standby mode, where the core and most peripherals are powered off. Their effects on the Synergy MCUs power consumption and throughput are illustrated as below

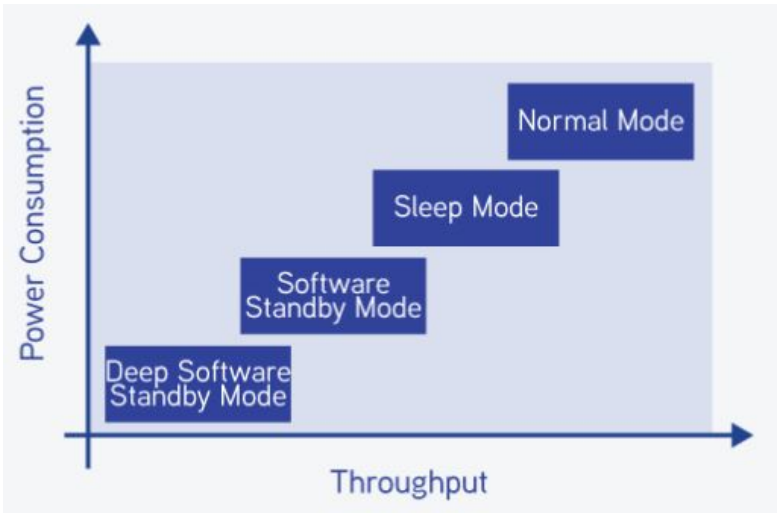

Figure 3 Power consumption and throughput of the LPM modes

Table 2 Low power modes supported in the Synergy MCU Groups

Synergy MCU Group	Low Power Modes			
	Sleep Mode	Software Standby Mode	Deep Software Standby Mode	Snooze Mode
S124	x	x		x
S128	x	x		x
S3A3	x	x		x
S3A6	x	x		x
S3A7	x	x		x
S5D5	x	x	x	x
S5D9	x	x	x	x
S7G2	x	x	x	x

1.2.2.1 Sleep Mode

In Sleep mode, the CPU stops operating but the contents of its internal registers are retained. Other peripheral functions do not stop. The CPU can be woken up by any interrupt, RES pin reset, a power-on reset, a voltage monitor reset, an SRAM parity error reset, or a reset caused by an IWDT or a WDT underflow.

1.2.2.2 Software Standby Mode (SBY)

In SBY the CPU and most of the on-chip peripheral functions and oscillators are clock gated. However, the contents of the CPU internal registers and SRAM data, the states of on-chip peripheral functions, and the I/O ports are retained. Software Standby mode allows a significant reduction in power consumption because most of the oscillators stop in this mode. Only those interrupts specified by the Wake-UP Interrupt Enable Register (WUPEN) can cancel the Software Standby mode.

1.2.2.3 Deep-Software Standby Mode (DSBY)

In DSBY more power consumption reduction compared to SBY is achieved by stopping more components such as oscillators, SRAM and flash. In this mode the contents of CPU internal registers and SRAM data, the states of on-chip peripheral functions and the I/O ports can be retained.

1.2.2.4 Snooze (SNZ)

Snooze feature provides operational flexibility to dramatically reduce current consumptions. Snooze is an extension of the Software Standby mode where some peripheral modules can operate without waking up the CPU. The Snooze mode can be entered through the Software Standby mode via some interrupt sources and woken up by those available interrupts in the Software Standby Mode.

A general description of those LPM modes can be summarized in the following table, but their detailed configurations are defined in the corresponding *Synergy Microcontroller Group User's Manual*.

Table 3 LPM modes and their basic configurations

Mode of Operation	Core	Flash	SRAM	RTC, AGT, Vbatt, LVD	Other Peripherals	IO Pins	Snooze
Normal	Operating	Selectable	Selectable	Selectable	Selectable	Selectable	N/A
Sleep	Clock Gated	Selectable	Selectable	Selectable	Selectable	Selectable	N/A
Software Standby	Clock Gated	Data Retained	Data Retained	Selectable	Clock Gated	State Retained	Available
Deep Software Standby	Powered Off	Powered Off	Partially Powered Off	Selectable	Powered Off	State Retained	N/A

Note: Selectable means that operating or disable can be selected by control registers. Some modules are also controlled by module-stop bit.

1.2.2.5 Transitions between the Low Power Modes

The transitions between the Normal mode and the LPM modes can be abstracted as shown in the following figure.

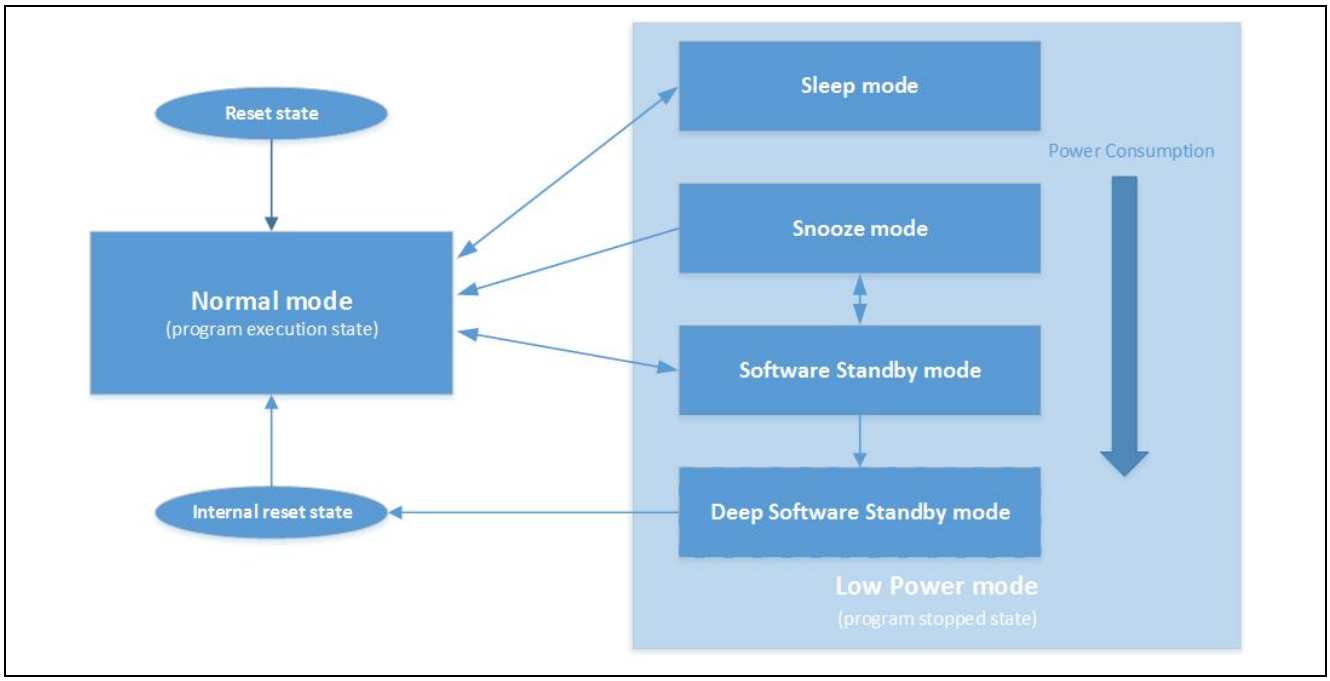

Figure 4 Transitions between the Normal mode and 4 LPM modes

Note: This graph only shows possible LPM modes but does not include the transition conditions of each mode, since different Synergy MCU Groups have different number of the LPM modes, and different transition conditions. In addition, their actual power consumption is also dependent on the LPM mode configurations. The users should refer to the specific *Synergy Microcontroller Group User's Manual*.

2. Power Profile v2 Framework Operational Overview

The PPv2 Framework provides a generic API for supporting low level power profiles in the Synergy MCU, when used with the LPM v2 Driver, CGC Driver, and IO Port Driver. It can be considered as an advanced control interface over the power consumption of the MCU, and can be used both in an application with or without ThreadX RTOS. Internally, it relies on the LPM v2, IOPORT, and CGC Drivers of the SSP, and provides an easy-to-use software interface to control the power modes of the MCU.

2.1 Power Profile v2 Framework

The PPv2 Framework provides two main profiles to control the MCU power consumption, the **Run Profile** and **Low Power Profile**. The module structure can be illustrated in the following diagram.

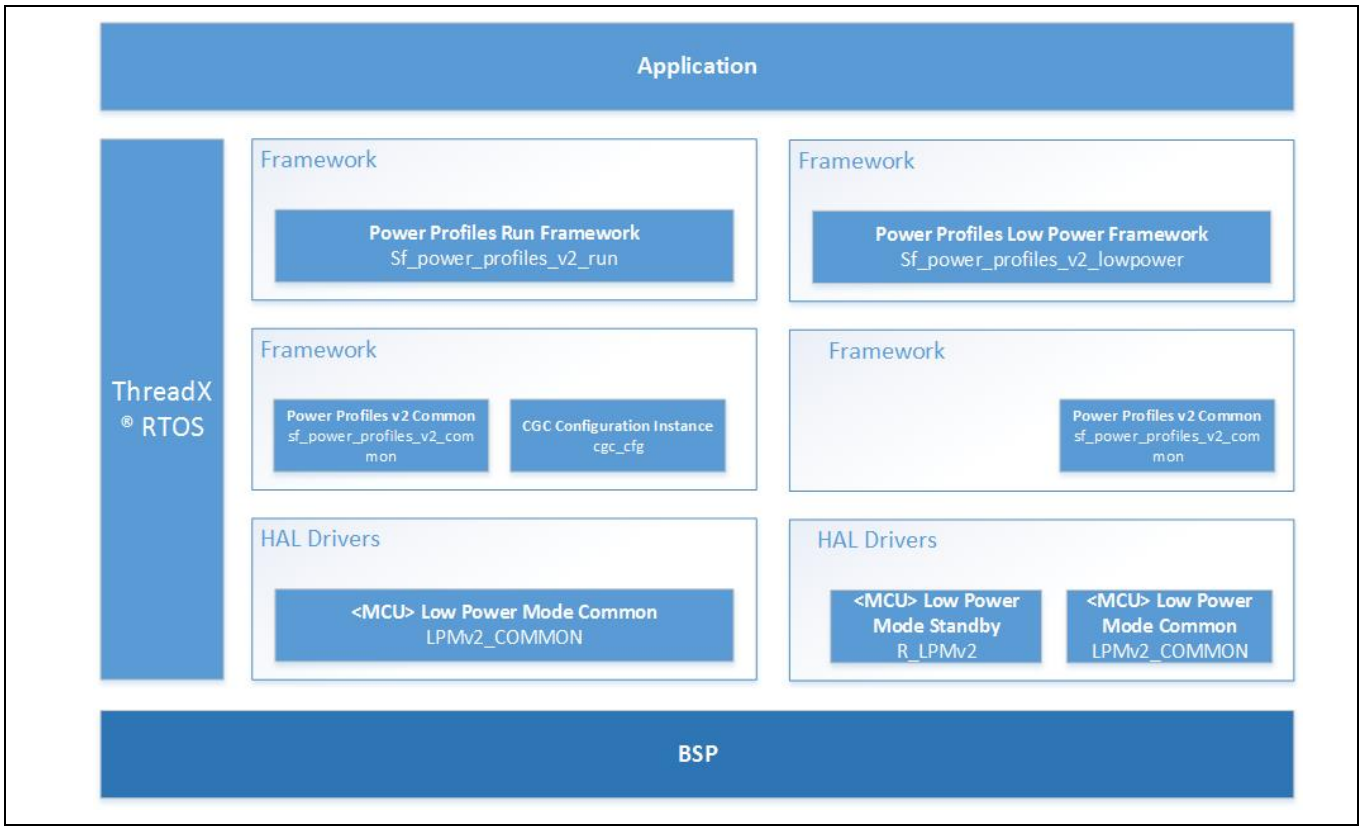

Figure 5 PPv2 Framework Stack

2.1.1 PPv2 Framework Run Profile

The **Run Profile** uses a CGC Clocks configuration and an IO Port pin configuration to set the system clocks and IO Port pins of the MCU in the normal running modes. Its function is implemented in the **`RunApply()`** API to perform the following tasks in the specified order.

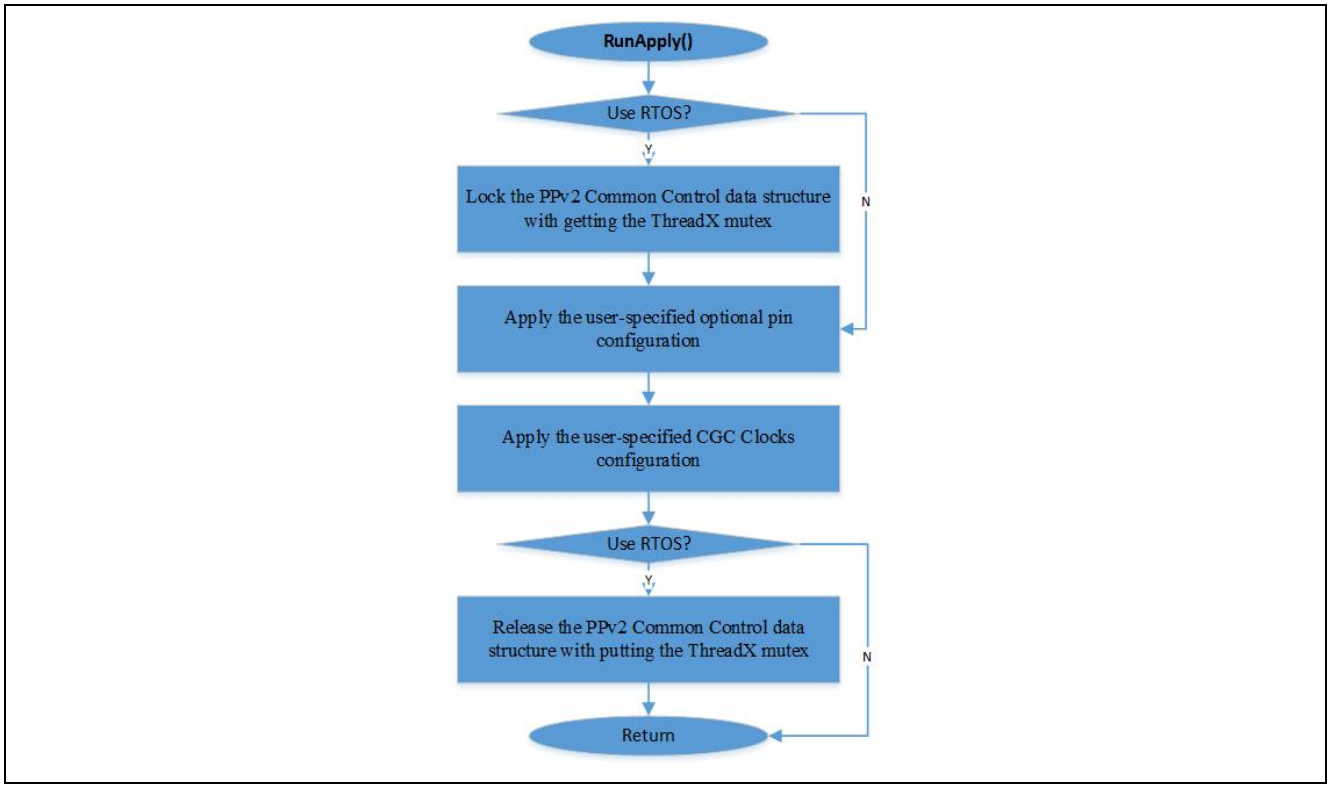

Figure 6 PPv2 `RunApply()` process

2.1.2 PPv2 Framework Low Power Profile

The **Low Power Profile** uses an LPM v2 configuration and both a pre-LPM and a post-LPM IO port configuration to set the low power mode and IO port pins before entering the configured low power mode and after waking up from the low power mode. See the SSP User's Manual and the particular *Synergy Microcontroller Group User's Manual* for details on the available low power modes.

The internal function **`LowPowerApply()`** API performs the following tasks in order.

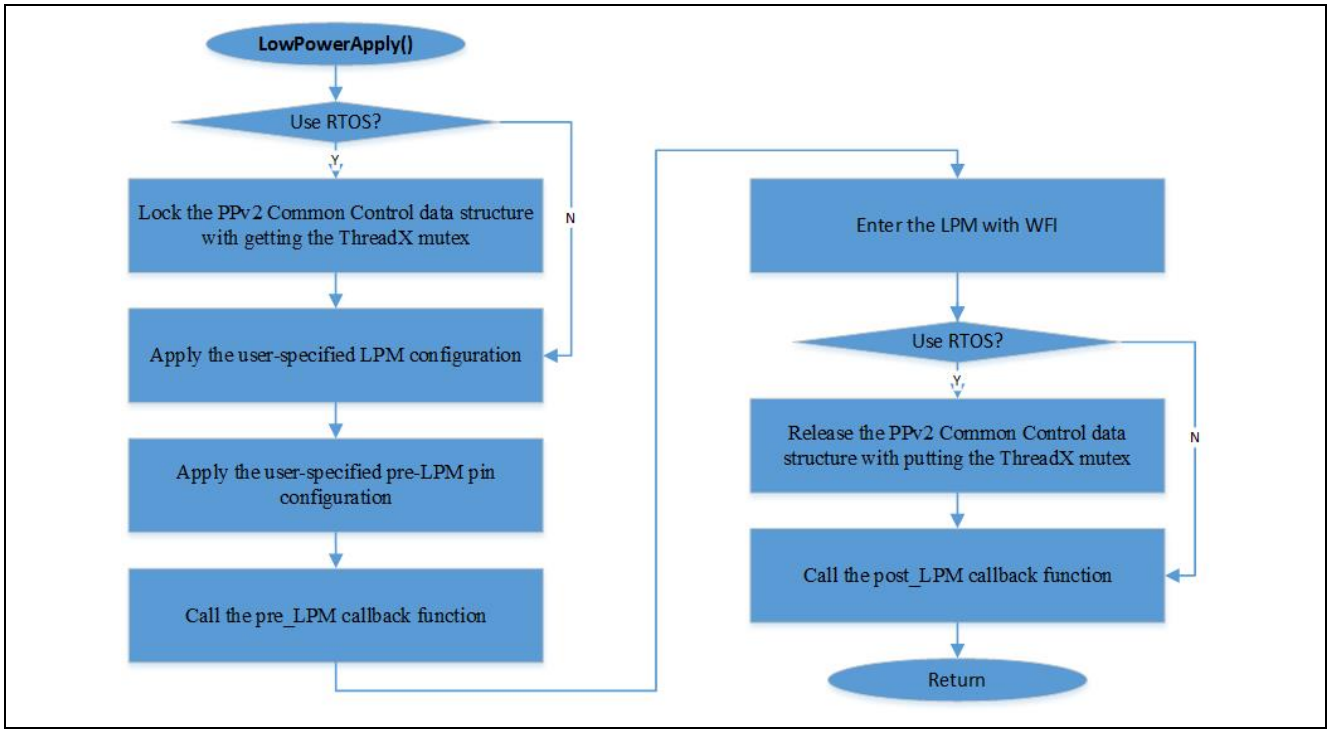

Figure 7 The PPv2 `LowPowerApply()` process

2.2 PPv2 Low Power Modes Operational Notes

The following operational notes are based on observations of the current release of the PPv2 framework and driver, and will be updated as more user feedback is received, and new versions of the power profiles package are released.

- **Power Profile v1 and PPv2**

  The Power Profiles v1 and PPv2 Framework are not compatible, so do not use PPv1 and PPv2 Framework in the same project. For all new projects, it is recommended that applications use PPv2 Framework.

- **An LPM v2 driver instance is added to PPv2 applications by default**

  This MCU specific LPM v2 driver defines configurations and APIs for configuring, enabling and disabling LPM operations in structures of `lpmv2_mcu_cfg_t` and `lpmv2_api_t`, then will be instantiated in the automatically generated file `common_data.c`. So, PPv2 Framework instance is based on an LPM v2 Driver instance.

- **Additional CGC driver instance for using PPv2 runApply() function**

  A default CGC driver is included in all Synergy projects, and instantiated in the file `common_data.c`, which will not increase the code size of a project. Another instance of the CGC Driver is required for a CGC Clocks configuration of a PPv2 `runApply()` function.

- **I/O port driver instance for different pin configurations**

  Different pin configuration tables can be defined in the PPv2 power profiles, but only one I/O port driver instance `ioport_instance_t` is instantiated in the automatically generated file `common_data.c`.

- **Operation with ThreadX**

  As shown in the above `RunApply()` and `LowPowerApply()` processes, the PPv2 Framework APIs uses a ThreadX intrinsic objects like a mutex for multithread applications when used with ThreadX.

- **Special Consideration on Multithreads Applications**

  In following LPM modes: Software Standby, Deep Software Standby, or Snooze modes; the source clock for `Systick` may be configured to be disabled. So special consideration is needed when implementing a multithread RTOS project with the PPv2 LPM modes.

- **Debugger Usage in LPM Modes**

  The MCU may not enter or stay in Software Standby and Deep Software Standby modes with the debugger attached, since the MCU may be woken up from Software Standby and Deep Software Standby modes by the debugger.

2.3 PPv2 Module Limitations

- The Power Profiles V2 Framework open function will not be called automatically prior to the main if the project does not use ThreadX. The initialization must be done explicitly by calling `g_common_init()` or by explicitly calling the API function `open()`. This is not a PPv2 limitation but a result of any Framework module that supports being used without an RTOS.
- The PPv2 Framework does not handle starting or stopping MCU peripherals, since no property view is available to select which peripherals to be stopped in a LPM mode, so users must stop them manually.
- Current version of PPv2 Framework only support following Synergy MCUs: S124, S128, S3A7, S5D9, and S7G2.
- Current version of PPv2 Framework doesn't support the transition from the Snooze to the Normal in a LPM Standby mode.

3. Power Profile v2 Module APIs Overview

PPv2 API functions at the framework layer and this section provides an operational overview of their usage.

3.1 PPv2 Framework API Functions

Assume that one of the PPv2 profiles is already added from the framework pulldown menu, and its configuration data structures corresponding to your instance are declared in the `sf_power_profile_v2_api.h.` This has three parts, `p_ctrl, p_cfg,` and `p_api`.

```
/* This structure encompasses everything that is needed to use an instance of this interface. */
typedef struct st_sf_power_profiles_v2_instance
{
    sf_power_profiles_v2_ctrl_t      * p_ctrl;    ///< Pointer to the control structure for this
instance
    sf_power_profiles_v2_cfg_t const * p_cfg;     ///< Pointer to the configuration structure for
this instance
    sf_power_profiles_v2_api_t const * p_api;     ///< Pointer to the API structure for this
instance
} sf_power_profiles_v2_instance_t;
```

The control structure `sf_power_profiles_v2_ctrl_t` is defined as below.

```
typedef struct st_sf_power_profiles_v2_ctrl
{
    uint32_t            open;        ///< Used by driver to check if pointer to control block is
valid
#if (1 == BSP_CFG_RTOS)
    TX_MUTEX            mutex;       ///< Mutex used to protect access to lower level driver
hardware registers
#endif /* (1 == BSP_CFG_RTOS) */
} sf_power_profiles_v2_ctrl_t;
```

The configuration structure sf_power_profiles_v2_cfg_t is defined as below.

```
typedef struct st_sf_power_profiles_v2_cfg
{
    /** Pointer to additional settings (not currently in use) */
    void                                 const * p_extend;
} sf_power_profiles_v2_cfg_t;
```

The **Run Profile** configuration has a structure:

```
typedef struct st_sf_power_profiles_v2_run_cfg
{
    /** Pointer to IOPORT settings */
    ioport_cfg_t                         const * p_ioport_pin_tbl;
    /** Pointer to a CGC configuration */
    cgc_clocks_cfg_t                     const * p_clock_cfg;
    /** Pointer to additional settings */
    void                                 const * p_extend;
} sf_power_profiles_v2_run_cfg_t;
```

The **Low Power Profiles** configuration has a structure:

```
typedef struct st_sf_power_profiles_v2_low_power_cfg
{
    /** Pointer to IOPORT settings to apply after exiting the low power mode   */
    ioport_cfg_t                         const * p_ioport_pin_tbl_exit;
    /** Pointer to IOPORT settings to apply before entering low power mode */
    ioport_cfg_t                         const * p_ioport_pin_tbl_enter;
    /** Pointer to an LPMv2 instance */
    lpmv2_instance_t                     const * p_lower_lvl_lpm;
    /** Callback function */
    void                                      (* p_callback)(sf_power_profiles_v2_callback_args_t
* p_args);
    /** Placeholder for user data */
    void                                       * p_context;
    /** Pointer to additional settings  */
    void                                 const * p_extend;
} sf_power_profiles_v2_low_power_cfg_t;
```

The PPv2 framework APIs (sf_power_profiles_v2_api_t) are summarized as follows.

Table 4 PPv2 framework API summary

Function Name	Example API Call and Description
.open	g_sf_power_profiles_v2_common.p_api->**open**(g_sf_power_profiles_v2_common.p_ctrl, g_sf_power_profiles_v2_common.p_cfg); Initialized the PPv2 framework. [in,out] p_ctrl Pointer to a structure allocated by user. Elements initialized here.(See definition of sf_power_profiles_v2_ctrl_t) [in] p_cfg Pointer to configuration structure. Elements of the structure must be set by user. (See definition of sf_power_profiles_v2_cfg_t)
.runApply	g_sf_power_profiles_v2_common.p_api->**runApply**(g_sf_power_profiles_v2_common.p_ctrl, &p_cfg); Apply a **Run profile**. [in] p_ctrl Pointer to control block set in the **open()** API above. [in] p_cfg Pointer to the run configuration structure. Elements of the structure must be set by user. (See definition of sf_power_profiles_v2_run_cfg_t)

.lowPowerApply	`g_sf_power_profiles_v2_common.p_api->`
	`lowPowerApply(g_sf_power_profiles_v2_common.p_ctrl,&p_cfg);`
	
	Apply a **Low Power profile**.
	
	`[in] p_ctrl Pointer to control block set in the open() API above.`
	`[in] p_cfg Pointer to the low power configuration structure. Elements of the structure must be set by user. (See definition of sf_power_profiles_v2_low_power_cfg_t)`

.versionGet	`g_sf_power_profiles_v2_common.p_api->versionGet(&version);`
	
	Get the version and place it at the pointer version, `p_version`.
	
	`[out] p_version Code and API version used.`

. close	`g_sf_power_profiles_v2_common.p_api->`
	`close(g_sf_power_profiles_v2_common.p_ctrl);`
	
	Close the framework.
	
	`[in] p_ctrl Pointer to control block set in the open() API above.`

Note: For more detailed 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.

The return values of the PPv2 framework APIs are defined as below.

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. See the SSP *User's Manual* API References for the associated module for a definition of all relevant status return values.

4. Including the PPv2 Framework in an Application

There are two possible ways to include the PPv2 Framework modules into an application:

- With using ThreadX
- Without using ThreadX

4.1 Including PPv2 Framework using ThreadX

The typical steps in using the PPv2 Framework in an application are:

Step 1: Add the **PPv2 Run Profile** modules into a thread.
Step 2: Configure power control modes, and set the pin configurations (Run Profile).
Step 3: Add the **PPv2 Low Power Profile** modules into a thread.
Step 4: Configure the LPM modes to set the transition conditions, and the pin configurations (Low Power Profile) for pre- and post- LPM modes.

This section focuses on Step 2 & 3, to add PPv2 Framework profile modules into a thread. The profile configurations for Steps 2 & 4 are covered in the section 5.

4.1.1 Adding PPv2 Run Profile Module into a Thread

Assume that a Synergy C project is already created for a selected MCU device by following the e^2 studio ISDE User Guide, and a new thread is also created by clicking a **New Thread** in the **Thread** tab, then click on the **Name** and **Symbol** entries in the property view, and enter a distinctive name and symbol for the new thread.

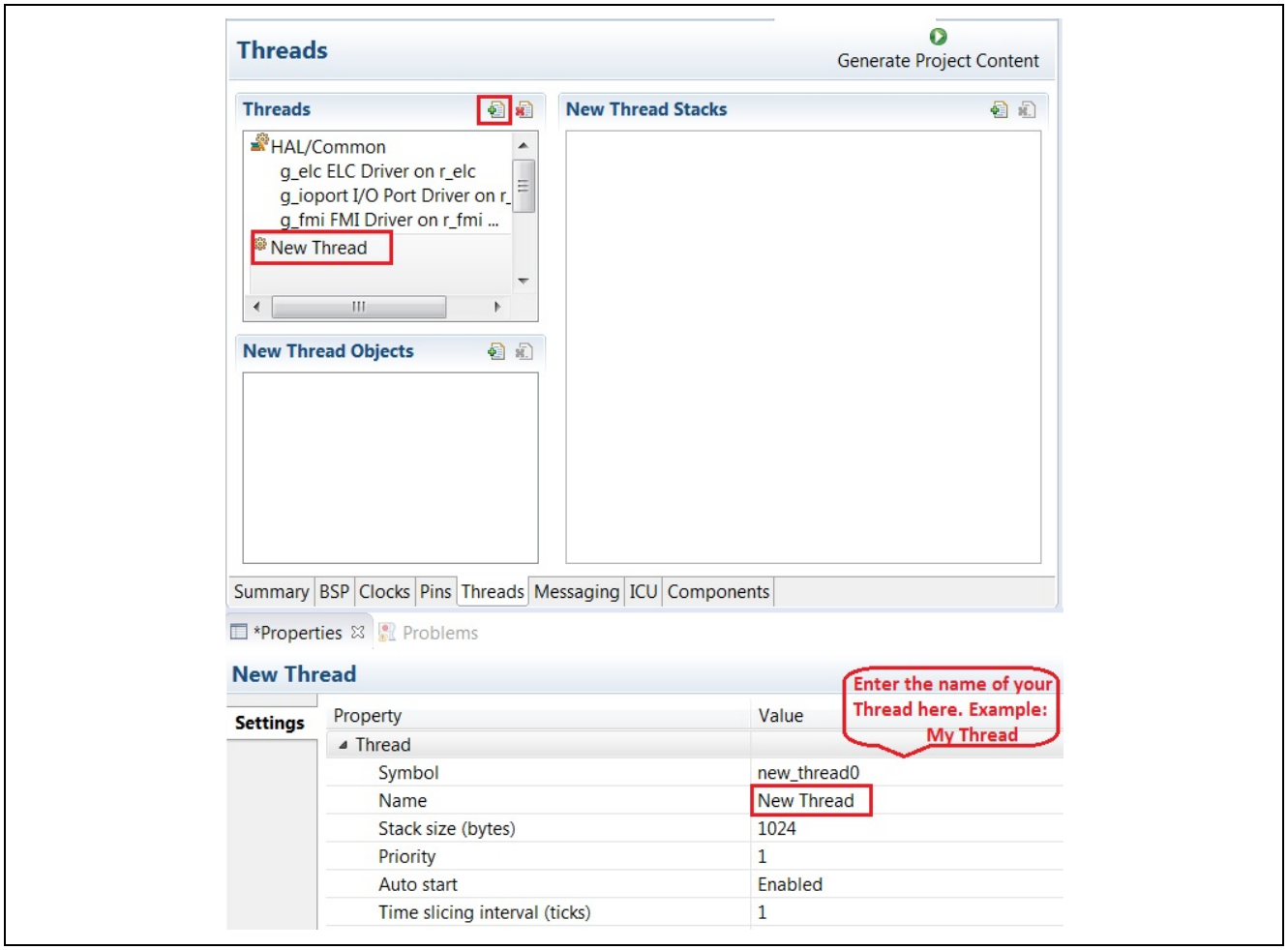

Figure 8 Adding a new thread on the Thread tab

The On the selected thread, click on **New** to select the Run profile module from the pull-down menu.

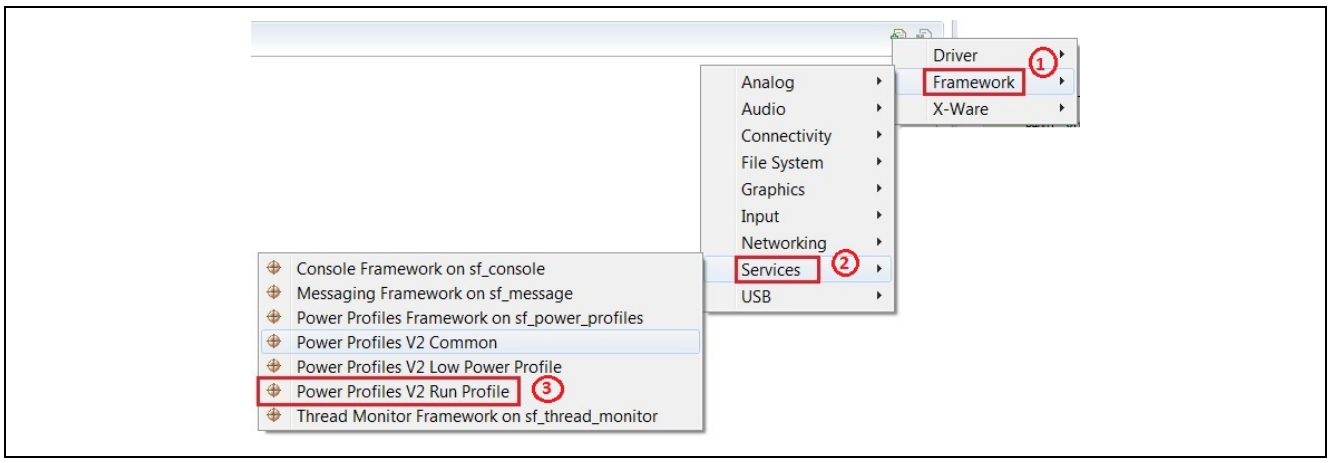

Figure 9 Adding a PPv2 Run Profile module

Then change the name of this Run Profile module to a meaningful name, such as `g_sf_power_profiles_v2_run_high_speed_mode`.

Note: You should give each instance of the PPv2 module a unique name in your project.

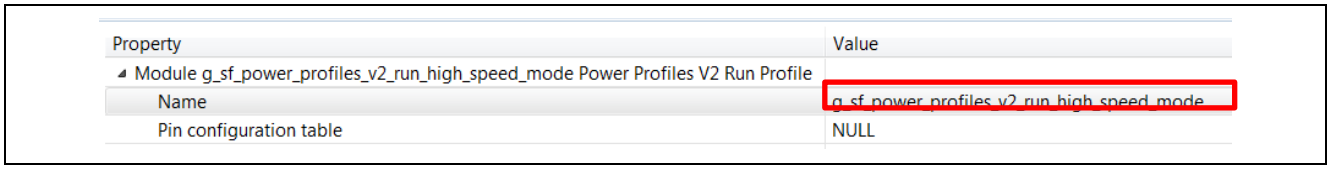

Figure 10 Naming the Run profile module

Figure 11 shows the created power control modes in the SSP configurator. Define its operating frequencies by switching clocks on and off, changing clock dividers, and selecting the system clocks in the Properties view of **the CGC Configuration Instance** module, which is covered in section 5.1.2.

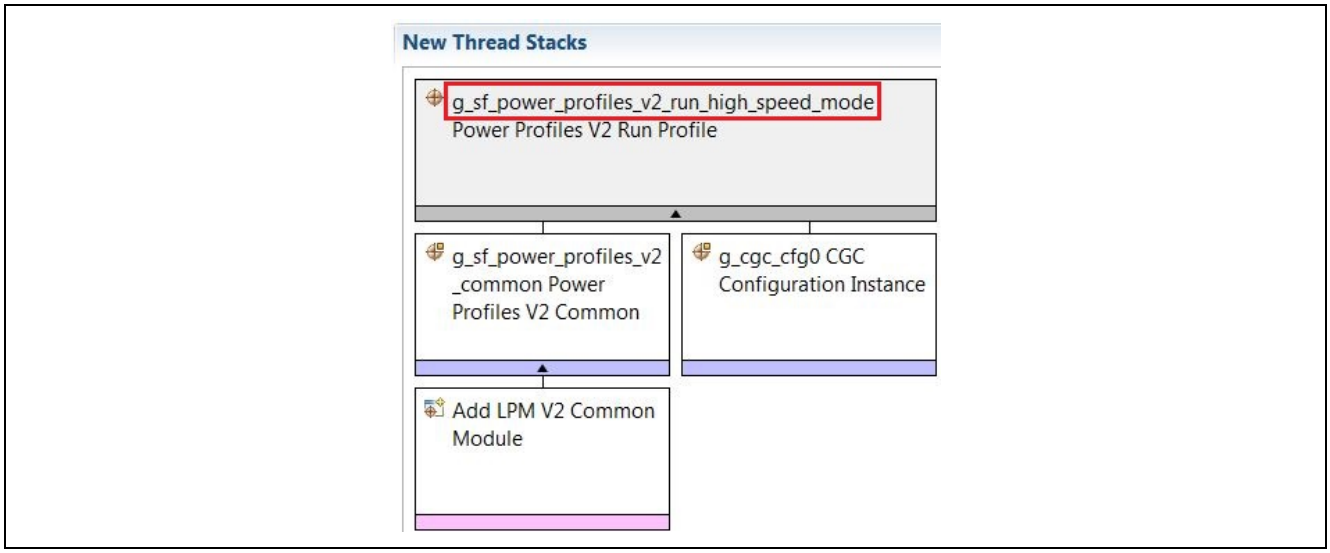

Figure 11 A power control mode defined in the PPv2 Run profile

Figure 12 shows selecting a LPM common module from the **New** pull-down menu, which contains specific PPv2 framework settings for a selected Synergy MCU Group, such as S3A7.

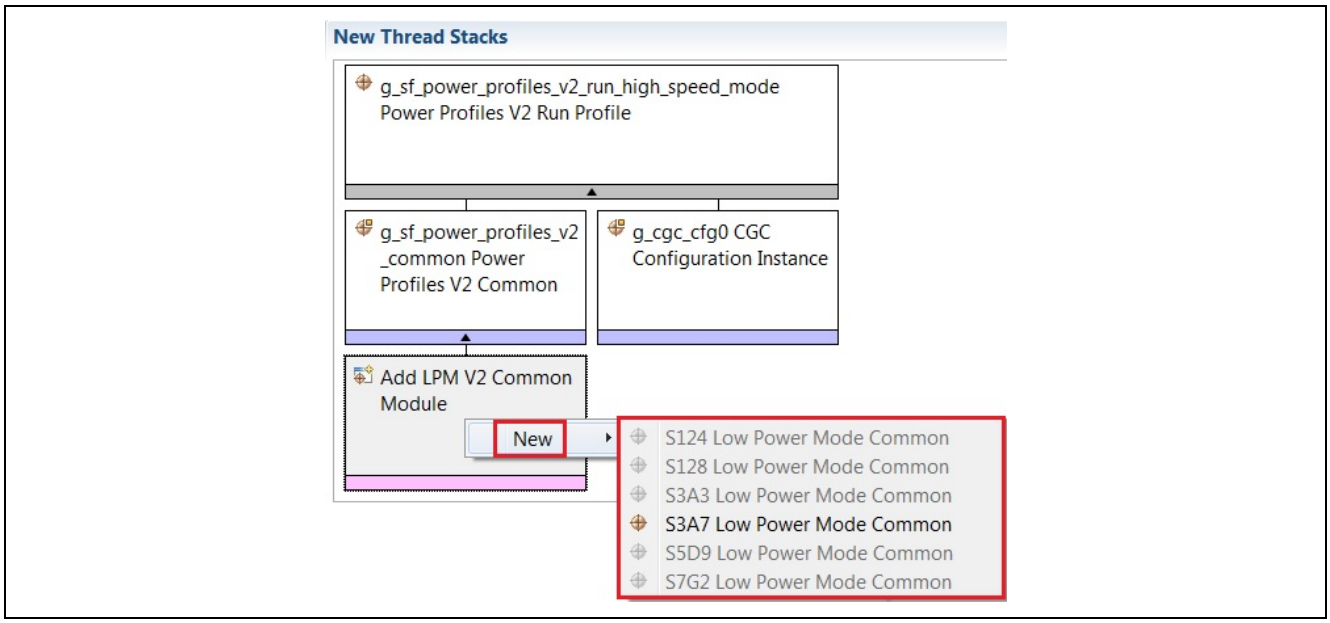

Figure 12 Adding the LPM v2 common module

4.1.2 Adding the PPv2 Low Power Profile Module into a Thread

On the selected thread, click **New** to select the Low Power profile module from the pull-down menu.

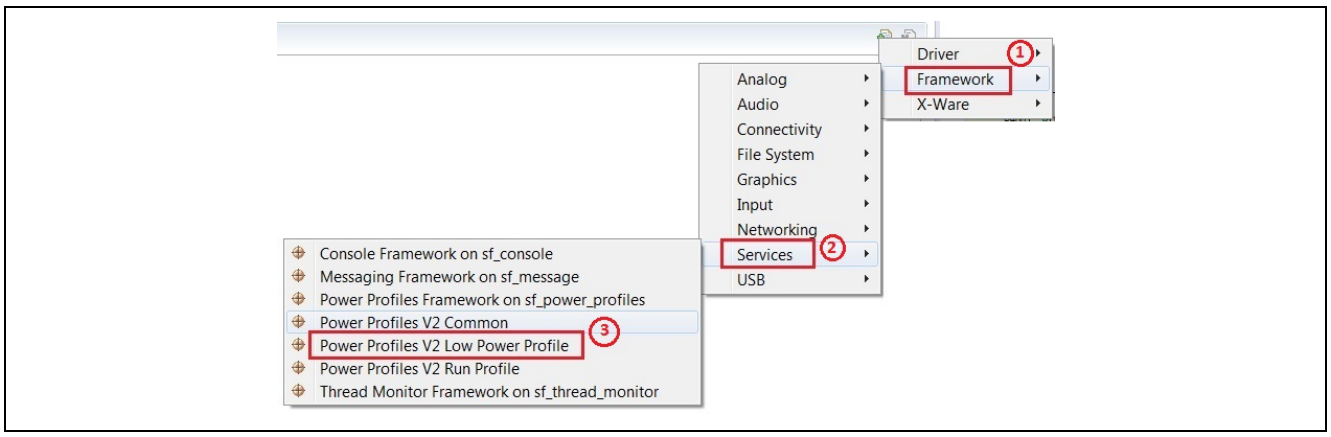

Figure 13 Adding a PPv2 Low Power profile module

A LPM Low Power profile module is generated in the SSP configurator. Available LPM modes for selected the MCU will be enabled.

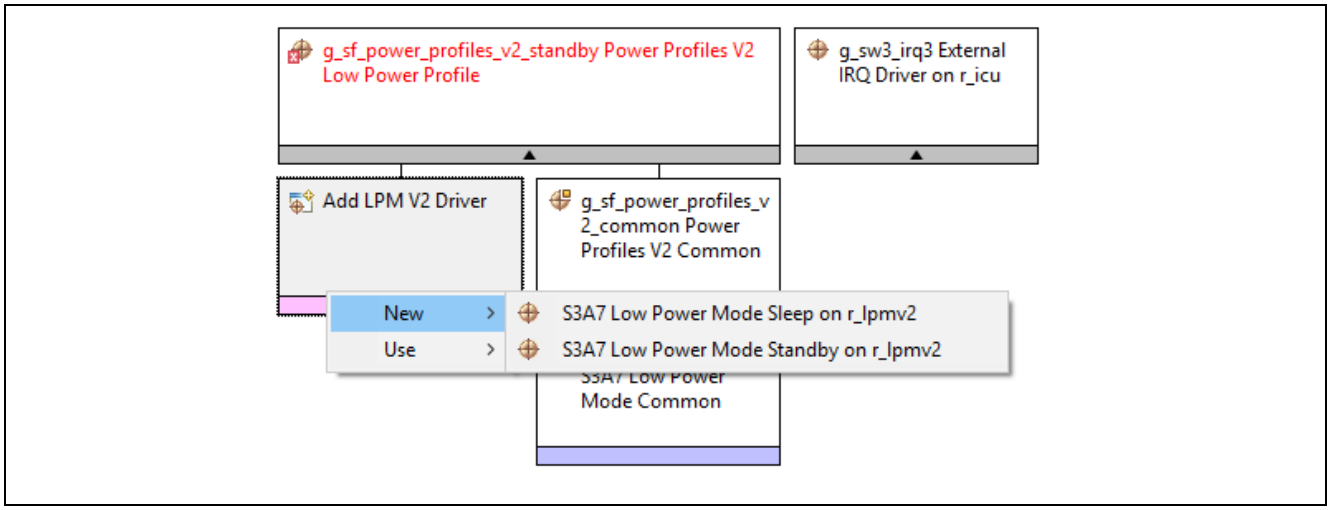

Figure 14 PPv2 Low Power Profile Module

In their applications users can add a PPv2 framework module to represent Sleep, Standby, or Deep Standby mode.

4.2 Including PPv2 Framework without using ThreadX

Similar to the steps of including PPv2 into a ThreadX based application, the PPv2 Framework profile modules can also be added into a non-thread project:

Step 1: Add the PPv2 Run Profile modules into the HAL/Common Stacks.
Step 2: Configure power control modes, and set the pin configurations (Run Profile).
Step 3: Add the PPv2 Low Power Profile modules into the HAL/Common Stacks.
Step 4: Configure the LPM modes to set the transition conditions, and the pin configurations (Low Power Profile) for pre- and post- LPM modes.

Step 2 & 4 profile configurations are covered in the section 5.

4.2.1 Adding PPv2 Run Profile Module into the HAL/Common Stacks

Assume that a Synergy C project is already created for a selected MCU device by following the e^2 studio ISDE User Guide, then click the **Add** button to add the following PPv2 Framework Run profile module.

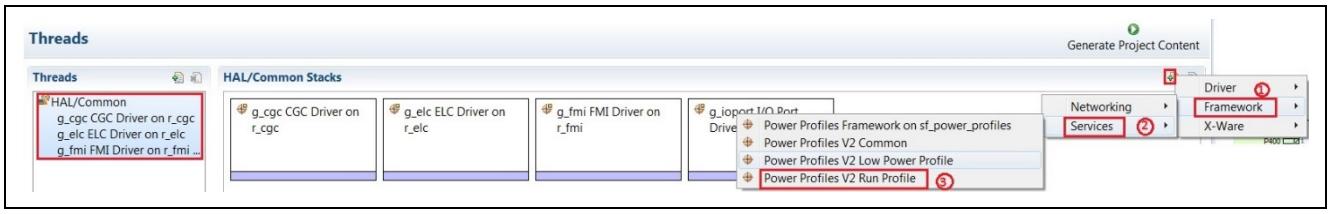

Figure 15 Adding a PPv2 Run Profile module into a HAL/Common stack

Assign a unique name for this Run Profile module in the property view, such as **`g_sf_power_profiles_v2_run_high_speed_mode`**. Now, click **New** on the LPM common module to select a PPv2 framework configuration for a Synergy MCU Group, like Figure 9 to Figure 13 show.

4.2.2 AddingPPv2 Low Power Profile Module into the HAL/Common Stacks

Similar to the operations adding PPv2 Low Power profile modules into a thread in the section 4.1.2, there are different PPv2 LPM profile modules that can be added by clicking the **Add** button, and then selecting the Low Power profile module from the pull-down menu.

5. Configuring PPv2 Framework Modules

Up to now sections described different ways to include the PPv2 Framework modules into applications, the next step is to show how to configure the PPv2 Framework profile modules, provide detailed configuration parameters, and give recommended values so that the user can apply them as applicable in their project applications.

Before discussing configurations of PPv2 Framework Profile modules, the property of a new thread is explained, the thread could be used with PPv2 Framework profiles shown in the Figure 8 (Section 4.2.2).

Table 5 Configurations for a new thread

ISDE Property	Value	Description
Symbol	New_thread0(default)	User can specify different name
Name	New Thread (default)	User can specify different name
Stack size (bytes)	1024 (default)	Application dependent
Priority	1 (default)	User can adjust this priority based on specific application
Auto start	Enabled (default)	User can adjust this setting based on the application implementation
Time slicing interval (ticks)	1 (default)	User can adjust this interval based on specific application

5.1 Configuration of PPv2 Framework Run Profile

Assumed that a PPv2 Framework Run profile module has been included into a project by performing operations in the Section 4, with module configurations discussed in the following section.

5.1.1 I/O Configuration of PPv2 Framework Run Profile

Figure 16 shows a Run profile, such as the following high-speed mode, and its configuration settings

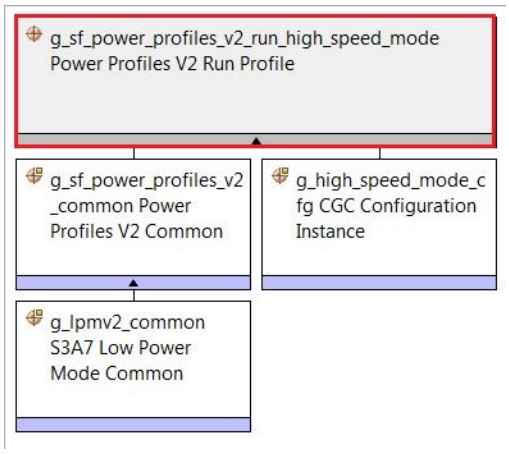

Figure 16 PPv2 Run Profile module

Table 6 Configuration settings on a Run profile module

ISDE Property	Value	Description
Name	g_sf_power_profiles_v2_run_high_speed_mode	Module name
Pin configuration table	NULL (default)	A pin configuration table

A pin configuration table can be created for a power mode, and linked into the Run profile property. Otherwise, the power mode uses the default pin assignment given by `g_bsp_pin_cfg`.

Basic steps to create a custom pin configuration table are given as follows:

- Copy a given board pin configuration file, such as `S3A7-DK.pincfg`, by right-clicking the file with the mouse button, selecting **Copy** from the menu, and then **Paste** to copy it into the same project.
- Rename this new pin configuration file, e.g., `S3A7-DK2_RUN.pincfg`.

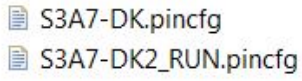

Figure 17 Create a new pin configuration file with Copy and Pates

- Figure 18 shows selecting the new pin configuration from the pull-down menu on the Pins tab of SSP Configurator, then specifying a pin configuration name to be generated.

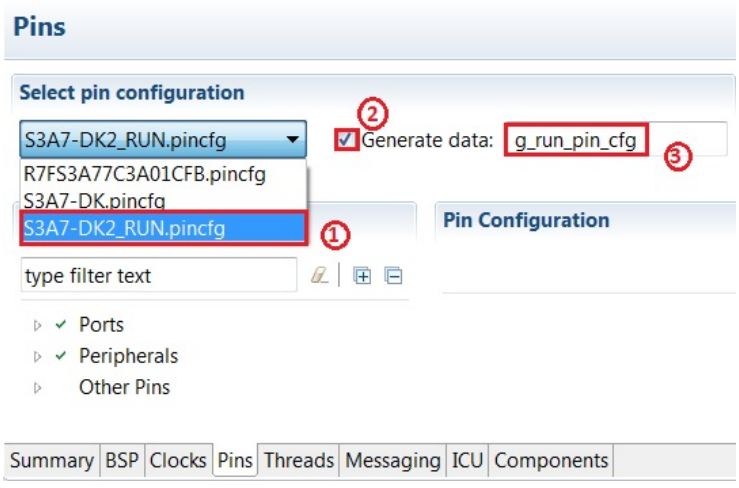

Figure 18 Select a new pin configuration file to be assigned

- Configure I/O functions on each pin, and then generate a new pin configuration by pressing the **Generate Project Content** button on the SSP Configurator as specified in the *Synergy SSP User's Manual*.

5.1.2 CGC Configuration of PPv2 Framework Run Profile

A power control mode of this Run Profile can be defined using the property view of a CGC Configuration Instance.

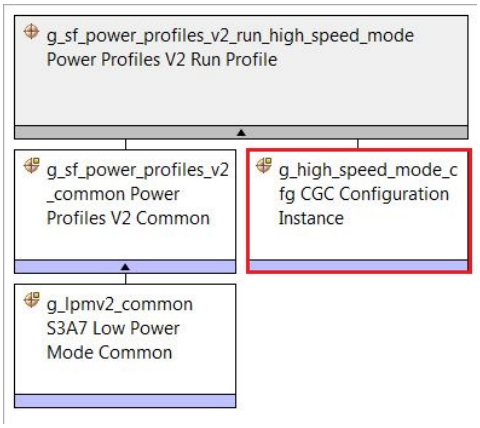

Figure 19 CGC Configuration Instance in the Run Profile module

Table 7 lists parameters of the CGC configuration property view.

Table 7 CGC configuration settings on a Run profile module

ISDE Property	Value	Description
Name	g_cgc_cfg	Module name
System Clock	HOCO (default), MOCO, LOCO, Main Oscillator, Sub-Clock, PLL	Set the system clock source
LOCO State Change	None (default), Stop, Start	LOCO state change selection
MOCO State Change	None (default), Stop, Start	MOCO state change selection
HOCO State Change	None (default), Stop, Start	HOCO state change selection
Sub-Clock State Change	None (default), Stop, Start	Sub-clock state change selection
Main Clock State Change	None (default), Stop, Start	Main clock state change selection
PLL State Change	None (default), Stop, Start	PLL source clock selection
PLL Source Clock	HOCO (default), MOCO, LOCO, Main Oscillator, Sub-Clock, PLL	Set the PLL source
PLL Divisor	1 (default), 2, 3, 4	PLL Output Frequency Division
PLL Multiplier	10.0(default), 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, 31.0	PLL Output Frequency Multiplication
PCLKA Divisor	1(default), 2, 4, 8, 16, 64	Peripheral Clock A Division
PCLKB Divisor	1(default), 2, 4, 8, 16, 64	Peripheral Clock B Division
PCLKC Divisor	1(default), 2, 4, 8, 16, 64	Peripheral Clock C Division
PCLKD Divisor	1(default), 2, 4, 8, 16, 64	Peripheral Clock D Division
BCLK Divisor	1(default), 2, 4, 8, 16, 64	External Bus Clock Division
FCLK Divisor	1(default), 2, 4, 8, 16, 64	Flash Clock Division
ICLK Divisor	1(default), 2, 4, 8, 16, 64	System Clock Division

Note: Assignments on these CGC parameters must be satisfied given the oscillator availability for each power control mode, and the expected frequencies of each operating clock. Specific relationships are shown in a CGC block diagram in the applicable Synergy MCU User's Manual.

The maximum operating frequency range for these clocks should not be outside the given ranges in the MCU User's Manual. Figure 20 shows a table from the *S3A7 Microcontroller Group User's Manual* listing clock specifications.

Table 9.2 Clock Generation Circuit Specifications (internal clock)

Item	Clock source	Clock supply	Specification
System clock (ICLK)	MOSC/SOSC/HOCO/MOCO/ LOCO/PLL	CPU, DTC, DMAC, Flash, SRAM	Up to 48 MHz Division ratio: 1/2/4/8/16/32/64
Peripheral module clock A (PCLKA)	MOSC/SOSC/HOCO/MOCO/ LOCO/PLL	Peripheral module (QSPI, SPI, SCI, SCE5, SDHI, CRC, IrDA, GPT bus-clock)	Up to 48 MHz Division ratio: 1/2/4/8/16/32/64
Peripheral module clock B (PCLKB)	MOSC/SOSC/HOCO/MOCO/ LOCO/PLL	Peripheral module (DAC12, IIC, SSI, DOC, CAC, CAN, AGT, POEG, CTSU)	Up to 32 MHz Division ratio: 1/2/4/8/16/32/64
Peripheral module clock C (PCLKC)	MOSC/SOSC/HOCO/MOCO/ LOCO/PLL	Peripheral module (ADC14 conversion clock)	Up to 64 MHz Division ratio: 1/2/4/8/16/32/64
Peripheral module clock D (PCLKD)	MOSC/SOSC/HOCO/MOCO/ LOCO/PLL	Peripheral module (GPT count clock)	Up to 64 MHz Division ratio: 1/2/4/8/16/32/64
FlashIF clock (FCLK)	MOSC/SOSC/HOCO/MOCO/ LOCO/PLL	FlashIF	1 MHz to 32 MHz (P/E) Up to 32 MHz (Read) Division ratio: 1/2/4/8/16/32/64
External bus clock (BCLK)	MOSC/SOSC/HOCO/MOCO/ LOCO/PLL	External bus	Up to 24 MHz Division ratio: 1/2/4/8/16/32/64
EBCLK pin output (EBCLK)	BCLK or 1/2 BCLK	EBCLK pin	Up to 12 MHz Division ratio: 1 or 2
USB clock (UCLK)	PLL	USBFS	48 MHz

Figure 20 Maximum operating frequency range for Synergy S3A7 MCU Group internal clocks

Since the current property view does not validate assignments automatically, use the CGC panel of the SSP Configurator to check your assignments first. Figure 21 shows the CGC configuration screen with a high-speed mode example.

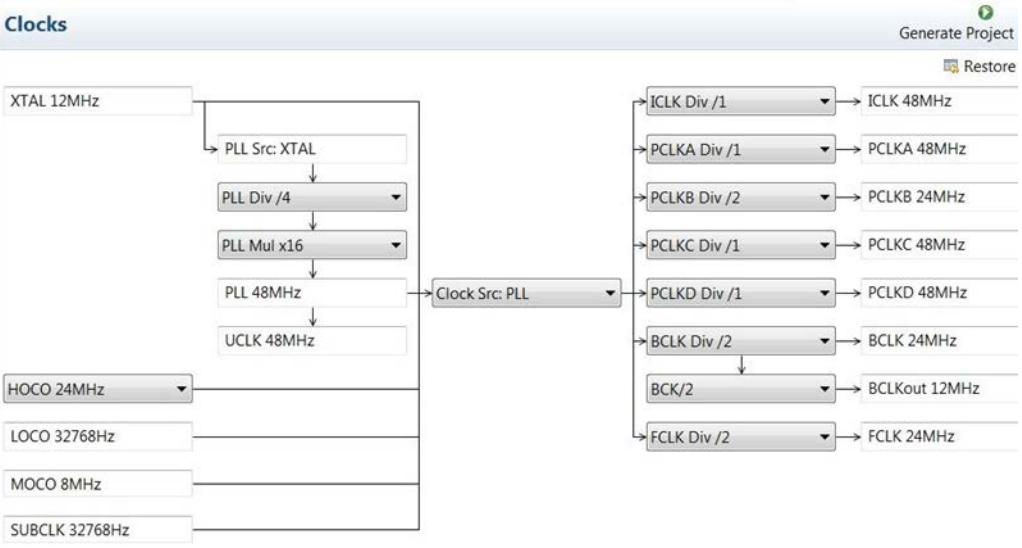

Figure 21 Checking the clock setting for a power control mode

Similarly, you can define the CGC configuration for other power control modes.

5.2 Configuration of PPv2 Framework Low Power Profile

Depending on the selected Synergy MCU Group, you may have three LPM modes available in the pull-down menu of the Low Power Profile module: Sleep, Software Standby, and Deep Software Standby.

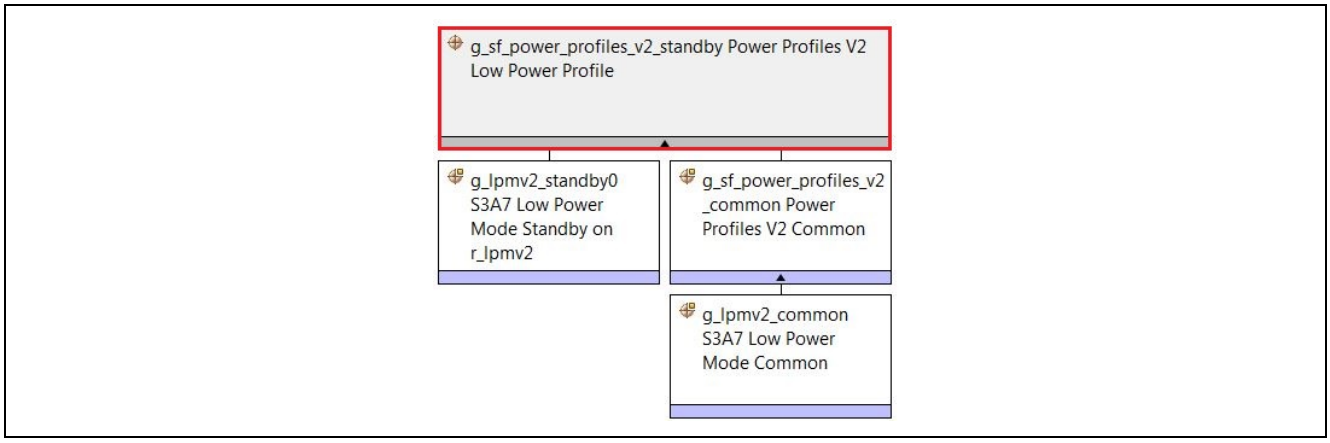

Figure 22 PPv2 Low Power Profile Standby module

Assume that a Low Power Profile module is already added into a thread, it will have the following configurations:

Table 8 Configuration settings on the PPv2 Low Power Profile module

ISDE Property	Value	Description
Name	g_sf_power_profiles_v2_low_power_0	Module name
Callback (Low Power Exit Event N/A when using Deep Software Standby)	NULL (default)	Callback function to handle the pre-entering LPM event and the post-exiting LPM event
Low power entry pin configuration table	NULL (default)	Pin configuration table for pre-entering LPM
Low power exit pin configuration table	NULL (default)	Pin configuration table for post-exiting LPM

The callback function can be used for handling the following events:

- `SF_POWER_PROFILES_V2_EVENT_PRE_LOW_POWER`
- `SF_POWER_PROFILES_V2_EVENT_POST_LOW_POWER`

You can change the IO port functionality with the following two pin configuration tables:

- Low power entry pin configuration table
- Low power exit pin configuration table

These tables are used internally in the PPv2 API function `LowPowerApply()`.

5.2.1 Configuration of the LPM Sleep Mode

The configuration of the LPM Sleep mode is simple, since any interrupt wakes the MCU from the Sleep mode. So only the module name can be renamed.

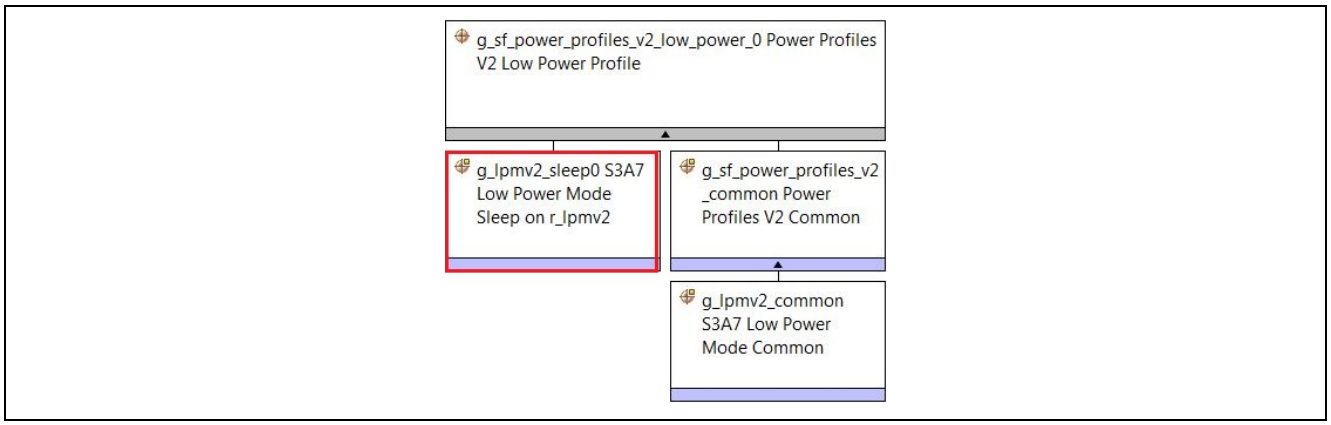

Figure 23 PPv2 Low Power Profile Sleep driver r_lpmv2

Table 9 Configuration Settings on the PPv2 Profile Sleep module

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

5.2.2 Configuration of the LPM Standby Mode

The Standby mode configuration sets up the exit triggers, and some transition conditions between the Standby and the Snooze mode, which are treated as a special case of the Standby mode in the current release of the PPv2. See the next section for more about the Snooze configuration.

Figure 24 shows an example of a S3A7 MCU Standby module.

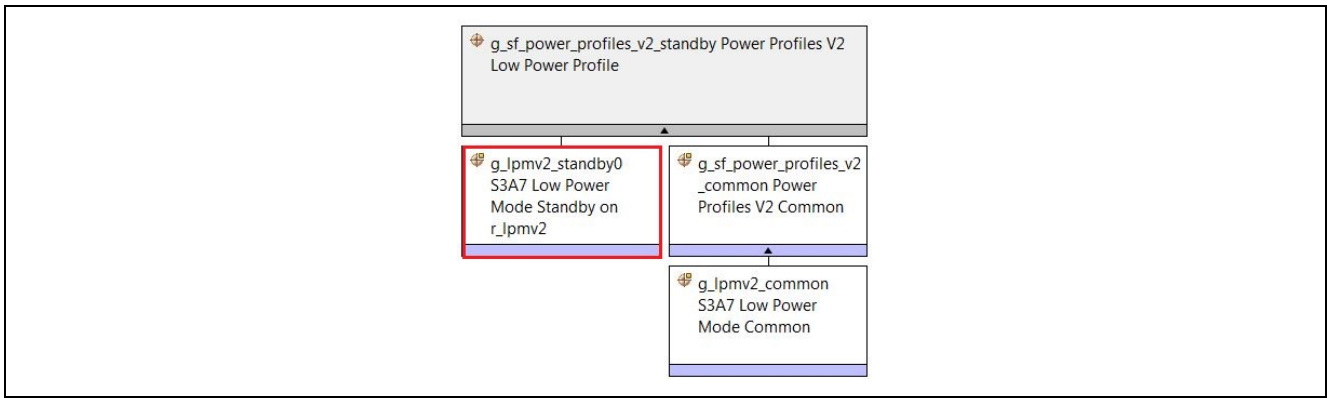

Figure 24 PPv2 Low Power Profile Standby driver r_lpmv2

Table 10 Configuration settings on the PPv2 Profile Standby module `r_lpmv2`

ISDE Property	Value	Description
Parameter Checking	BSP (default), Enabled, Disabled	Enables or disables the parameter checking
Name	g_lpmv2_standby0	Module name
Choose the low power mode	Standby (default), Standby with Snooze enabled	Low power mode selection
Output port state in standby and Deep Software Standby, applies to address output, data output, and other bus control pins	No Change (default), High Impedance state	Output port state selection
Select Standby Exit Sources		**Select Fields below**
IRQ[0:15]	Enabled, Disabled (default)	Select an external IRQ0 to IRQ15
IWDT	Enabled, Disabled (default)	IWDT selection

ISDE Property	Value	Description
Key Interrupt	Enabled, Disabled (default)	Key Interrupt selection
LVD1 Interrupt	Enabled, Disabled (default) (Default: Disabled)	LVD1 selection
LVD2 Interrupt	Enabled, Disabled (default)	LVD2 selection
Analog Comparator High-speed 0 Interrupt	Enabled, Disabled (default)	Analog Comparator selection
RTC Period	Enabled, Disabled (default)	RTC Period selection
RTC Alarm	Enabled, Disabled (default)	RTC Alarm selection
USBFS	Enabled, Disabled (default)	USBFS selection
AGT1 underflow	Enabled, Disabled (default)	AGT1 underflow selection
AGT1 Compare Match A	Enabled, Disabled (default)	AGT1 CMA selection
AGT1 Compare Match B	Enabled, Disabled (default)	AGT1 CMB selection
I2C 0	Enabled, Disabled (default)	I2C 0 selection
Snooze Mode Settings		
Snooze Entry Source	RXD0 falling edge (default), IRQ0:15, KINT (Key Interrupt), ACMPLP (Low-speed Analog Comparator), RTC Alarm, RTC Period, AGT1 Underflow, AGT1 Compare Match A, AGT1 Compare Match B	Source of entering the Snooze mode
Snooze Exit Sources		Select fields below
AGT1 Underflow	Enabled, Disabled (default)	AGT1 Underflow selection
DTC Transfer Completion	Enabled, Disabled (default)	DTC Transfer Completion selection
DTC Transfer Completion Negated signal	Enabled, Disabled (default)	DTC Transfer Completion Negated signal selection
ADC0 Compare Match	Enabled, Disabled (default)	ADC0 Compare Match selection
ADC0 Compare Mismatch	Enabled, Disabled (default)	ADC0 Compare Mismatch selection
SCI0 Address Match	Enabled, Disabled (default)	SCI0 Address Match selection
DTC state in Snooze Mode	Enabled, Disabled (default)	DTC state in Snooze Mode selection

5.2.3 Configuration of the LPM Snooze mode

Currently, as seen in Figure 25, the Snooze mode is enabled in the property view of the PPv2 Software Standby mode by selecting "**Standby with Snooze Enabled**" in the field of the "**Choose the low power mode**."

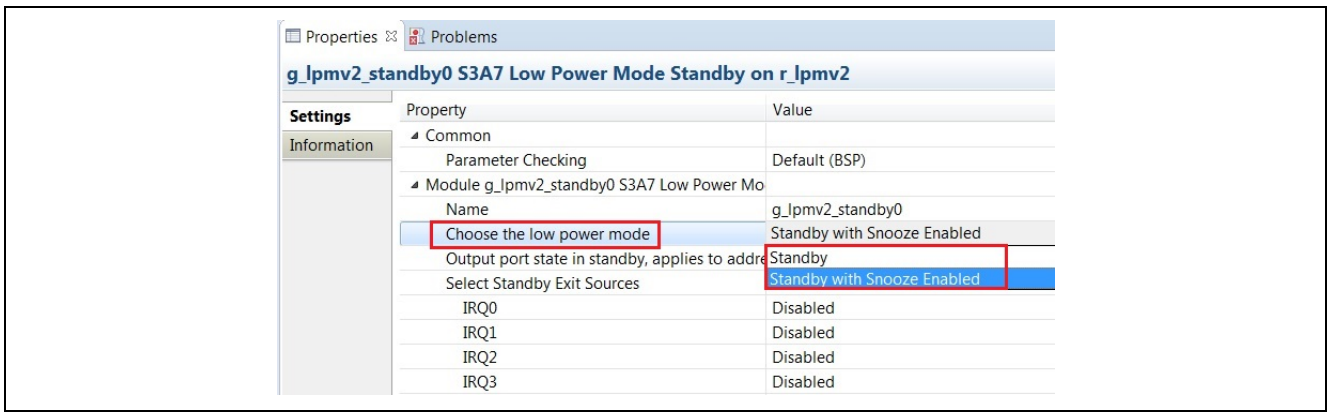

Figure 25 Create a Snooze mode in the Software Standby mode

5.2.4 Configuration of the LPM Deep Software Standby Mode

The configuration on the Deep Software Standby mode is to set the exit triggers of the Deep Software Standby mode, and the internal power supply options. Use the PPv2 Framework on the S5D9 as an example to show the possible configurations as follows:

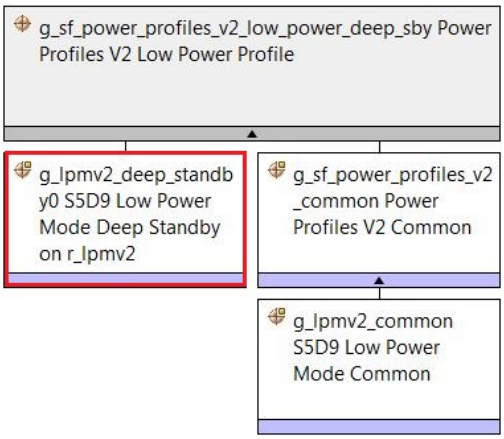

Figure 26 PPv2 Low Power Profile Deep Software Standby driver r_lpmv2

Table 11 Configuration settings on the PPv2 Profile Deep Software Standby module

ISDE Property	Value	Description
Parameter Checking	BSP (default), Enabled, Disabled	Enables or disables the parameter checking
Name	g_lpmv2_deep_standby0	Module name
Output port state in standby and Deep Software Standby, applies to address output, data output, and other bus control output pins	High impedance state, No change (default)	Output port state setting in Standby and Deep Software Standby
Maintain or reset the IO port states on exit from Deep Software Standby mode	Maintain the IO port states(default), Reset the IO port states	Output port state setting exit
Internal power supply control in Deep Software Standby mode	Maintain the internal power supply (default), Cut the power supply to standby RAM, Low-speed on-chip oscillator, AGTn, and USBFS/HS resume detecting unit, Cut the power supply to LVDn, standby RAM, Low-speed on-chip oscillator, AGTn, and USBFS/HS resume detecting unit	Internal power supply control in Deep Software Standby mode setting
Deep Software Standby Cancel Sources/Edges:		**Select Fields Below**
IRQ0	Enabled, Disabled (default)	IRQ0 selection
IRQ0 Edge	Disabled (default), Rising Edge, Falling Edge	IRQ0 Edge selection
IRQ1	Enabled, Disabled (default)	IRQ1 selection
IRQ1 Edge	Disabled(default), Rising Edge, Falling Edge	IRQ1 Edge selection
IRQ2	Enabled, Disabled (default)	IRQ2 selection
IRQ2 Edge	Disabled, Rising Edge, Falling Edge (default)	IRQ2 Edge selection
IRQ3	Enabled, Disabled (default)	IRQ3 selection
IRQ3 Edge	Disabled(default), Rising Edge, Falling Edge	IRQ3 Edge selection
IRQ4	Enabled, Disabled (default)	IRQ4 selection
IRQ4 Edge	Disabled(default), Rising Edge, Falling Edge	IRQ4 Edge selection
IRQ5	Enabled, Disabled (default)	IRQ5 selection

ISDE Property	Value	Description
IRQ5 Edge	Disabled, Rising Edge, Falling Edge (default)	IRQ5 Edge selection
IRQ6	Enabled, Disabled (default)	IRQ6 selection
IRQ6 Edge	Disabled(default), Rising Edge, Falling Edge	IRQ6 Edge selection
IRQ7	Enabled, Disabled (default)	IRQ7 selection
IRQ7 Edge	Disabled(default), Rising Edge, Falling Edge	IRQ7 Edge selection
IRQ8	Enabled, Disabled (default)	IRQ8 selection
IRQ8 Edge	Disabled(default), Rising Edge, Falling Edge	IRQ8 Edge selection
IRQ9	Enabled, Disabled (default)	IRQ9 selection
IRQ9 Edge	Disabled(default), Rising Edge, Falling Edge	IRQ9 Edge selection
IRQ10	Enabled, Disabled (default)	IRQ10 selection
IRQ10 Edge	Disabled(default), Rising Edge, Falling Edge	IRQ10 Edge selection
IRQ11	Enabled, Disabled (default)	IRQ11 selection
IRQ11 Edge	Disabled(default), Rising Edge, Falling Edge	IRQ11 Edge selection
IRQ12	Enabled, Disabled (Default: Disabled)	IRQ12 selection
IRQ12 Edge	Disabled(default), Rising Edge, Falling Edge	IRQ12 Edge selection
IRQ13	Enabled, Disabled (default)	IRQ13 selection
IRQ13 Edge	Disabled(default), Rising Edge, Falling Edge	IRQ13 Edge selection
IRQ14	Enabled, Disabled (default)	IRQ14 selection
IRQ14 Edge	Disabled, Rising Edge, Falling Edge (Default: Disabled)	IRQ14 Edge selection
IRQ15	Enabled, Disabled (Default: Disabled)	IRQ15 selection
IRQ15 Edge	Disabled(default), Rising Edge, Falling Edge	IRQ15 Edge selection
LVD1	Enabled, Disabled (default)	LVD1 selection
LVD1 Edge	Disabled(default), Rising Edge, Falling Edge	LVD1 Edge selection
LVD2	Enabled, Disabled (default)	LVD2 selection
LVD2 Edge	Disabled(default), Rising Edge, Falling Edge	LVD2 Edge selection
RTC Interval	Enabled, Disabled (default)	RTC Interval selection
RTC Alarm	Enabled, Disabled (default)	RTC Alarm selection
NMI	Enabled, Disabled (default: Disabled)	NMI selection
NMI Edge	Disabled(default), Rising Edge, Falling Edge	NMI Edge selection
USBFS	Enabled, Disabled (default)	USBFS selection
UBSHS	Enabled, Disabled (default)	UBSHS selection
AGT1	Enabled, Disabled (default)	AGT1 selection

Note: The property dialog of the PPv2 Framework provides two predefined configurations for internal power supply control in Deep Software Standby mode. More selections for stopping power supply to internal components are listed in the *Synergy Microcontroller Group User's Manual*, such as in Table 11.2 of the *Synergy S5D9 Microcontroller Group User's Manual*.

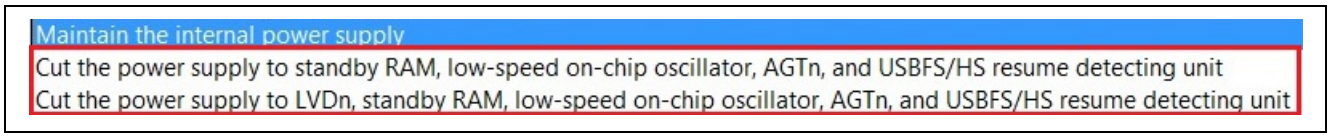

Figure 27 **Predefined internal power supply options in the PPv2 Framework Deep Software Standby mode**

5.3 Configuration of PPv2 Framework Common Modules

There are two Common modules shared among PPv2 Framework Run/Low Power profile modules:

- Automatically Generated Power Profile V2 Common Module

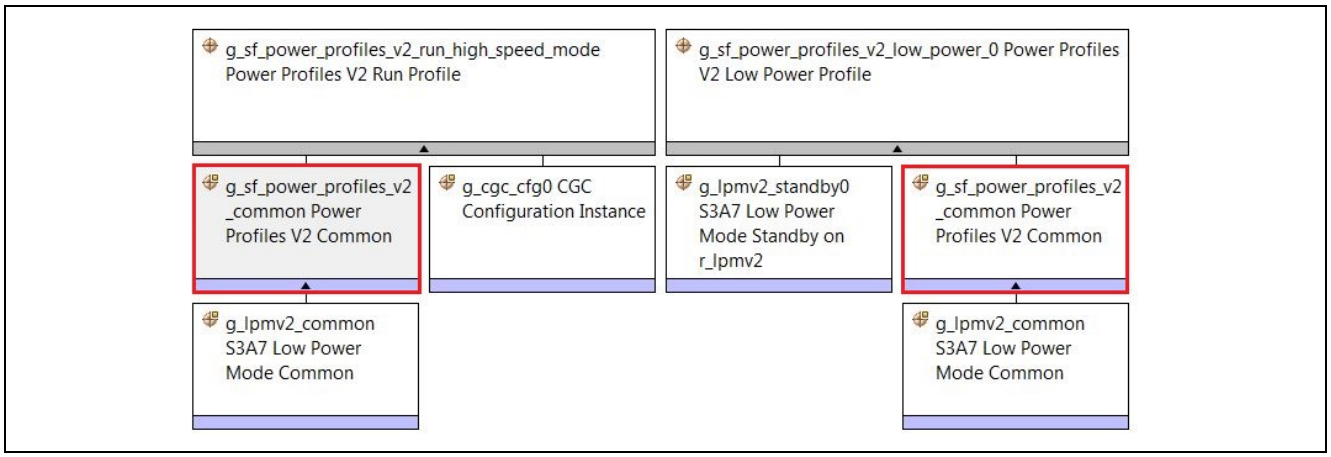

Figure 28 PPv2 Common module of PPv2 Run/Low Power profile modules

Table 12 lists their configuration settings.

Table 12 Configuration Settings for the Power Profiles V2 Common

ISDE Property	Value	Description
Parameter Checking	BSP(default), Enabled, Disabled	Enables or disables the parameter checking
Name	g_sf_power_profiles_v2_common	Module name

- Users Added MCU Specific Low Power Mode Common Module

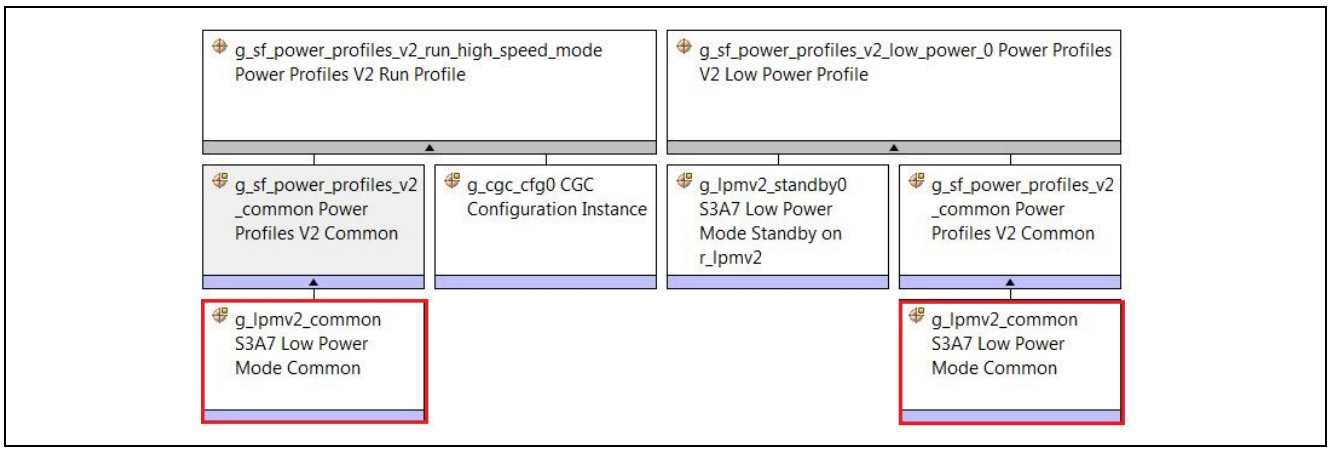

Figure 29 LPM Common module of PPv2 Run/Low Power profile modules

Table 13 lists their configuration settings.

Table 13 Configuration Settings for the Low Power Mode Common

ISDE Property	Value	Description
Parameter Checking	BSP(default), Enabled, Disabled	Enables or disables the parameter checking
Name	g_lpmv2_common	Module name

Note: The Synergy MCU S3A7 LPM module is already added in the Figure 12 when adding a PPv2 Framework Run profile module.

6. PPv2 Framework Application Example

The goal of this application project is to show how to create an RTOS-aware application with the PPv2 framework modules. It provides a general testing platform to enumerate different combinations of the power control modes and the LPM modes, as well as transitions among the modes. However, here only a simplified version is demonstrated, with two power control modes and two LPM modes.

The DK-S3A7 is selected as the development board for this project since it has more user-controlled features, such as three press buttons, a potentiometer, and several LEDs to visualize the MCU state change.

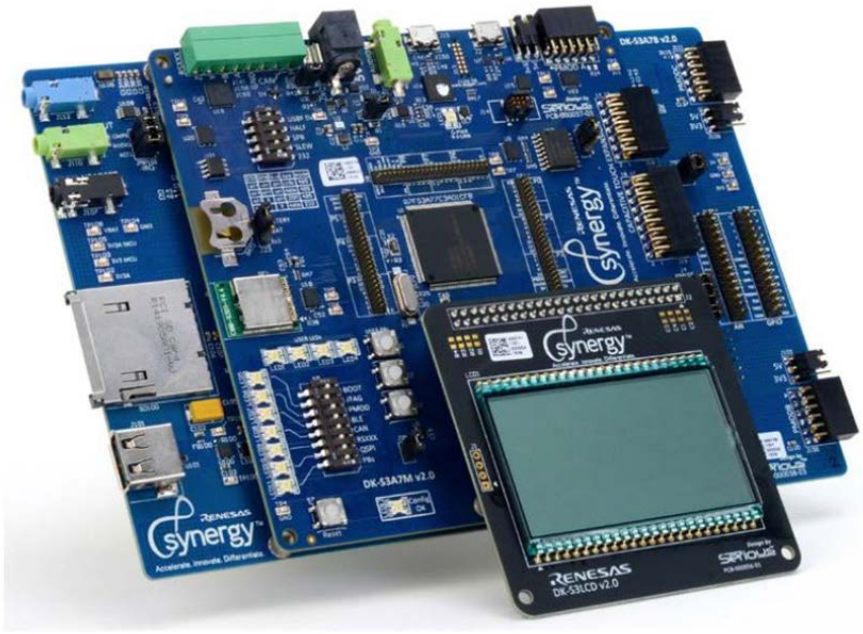

Figure 30 DK-S3A7 v2.0 is demonstrated as a PPv2 project

The Synergy S3A7 MCU Group has five operating power control modes:

- High-speed mode
- Middle-speed mode
- Low-speed mode
- Low-voltage mode
- Subosc-speed mode

The Synergy S3A7 MCU Group has three LPM modes:

- Sleep mode
- Software Standby mode
- Snooze mode

Figure 31 shows mode transitions and their triggering conditions. For details, see *Synergy S3A7 Microcontroller Group User's Manual* (Renesas, Feb 2016).

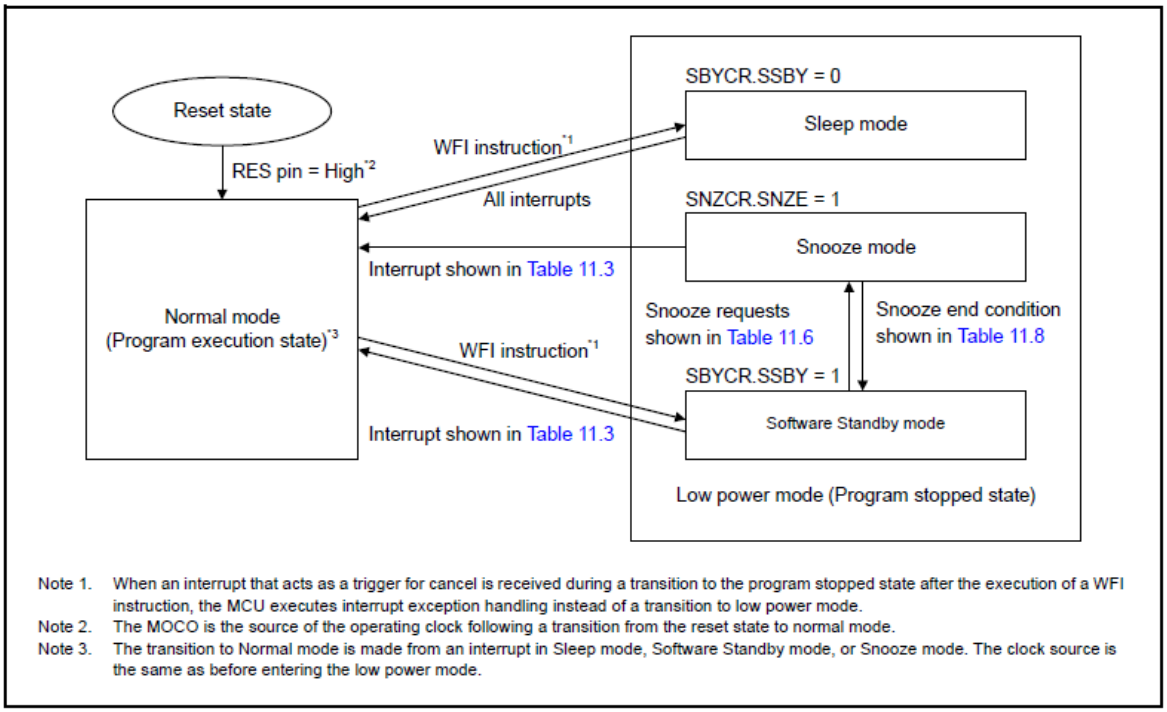

Figure 31 LPM mode transitions for the Synergy S3A7 Group MCU

6.1 Project Overview

The project operations can be summarized as follows:

- Press the S1 button on the DK-S3A7 v2.0 board to select a power control mode, then hold the S1 button to enter the selected power control mode. Currently, only High-Speed and Middle-Speed modes are implemented, their transitions are marked with the solid red lines.

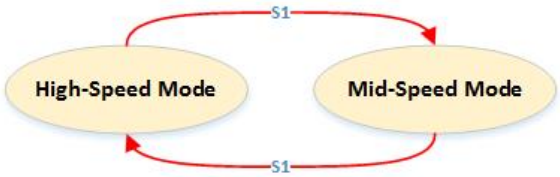

Figure 32 Press the button S1 for selecting a power control mode

- The LED1_G and LED2_G on the DK-S3A7 v2.0 board are used to represent the power control modes as follows:

Power Control Modes	LED1_G	LED2_G
High-speed Mode (default)	On	On
Middle-speed Mode	Off	On

- Press the S2 button to select a LPM mode and the Normal mode, and then hold S2 to enter the selected LPM mode. Currently only Standby mode is implemented, so its transition with Normal mode is marked with a red solid line.

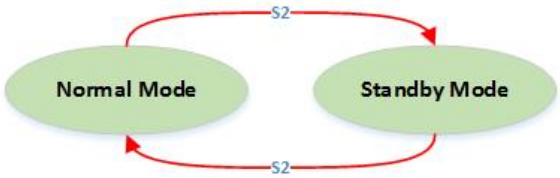

Figure 33 Press the button S2 to select a LPM mode

- Use the LED1_R and LED2_R on the DK-S3A7 v2.0 to represent the power control modes as follows:

LPM Modes	LED1_R	LED2_R
Normal mode (default)	On	On
Software Standby mode	Off	On

- The Software Standby mode is entered, where the CPU, most of the on-chip peripheral functions and oscillators, stop.
  - Note: The LED1_R and LED1_G are a pair of physically adjacent LEDs, so it will look like an orange color LED when both of Red and Green are on. The pair of LED2_R and LED2_G also has similar observation.

Figure 34 shows a state diagram of these mode transitions.

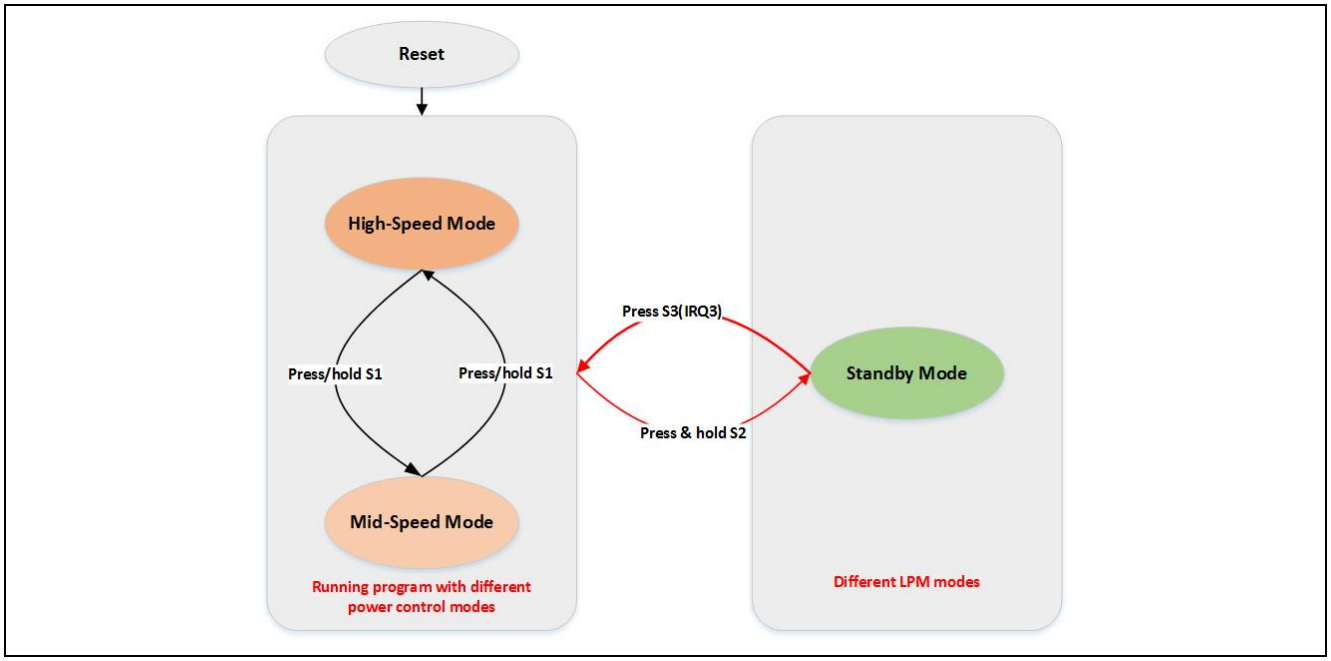

Figure 34 State transitions in the PPv2 project with DK-S3A7

6.2 Project Software Architecture

The software architecture can be partitioned into two parts and implemented with four threads:

- The **frontend** to handle the events from the user interface:
  - **SW1_thread:** Create a timer for checking the S1 button condition, then decide whether to select a power control mode or send a message to change the power control mode based on the events from the S1 timers
  - **SW2_thread:** Create a timer for checking the S2 button condition, then decide whether to select a LPM mode or send a message to change the LPM mode based on the events from the S2 timers.
- The **backend** to enter the user selected power control mode and the LPM modes by applying the PPv2 framework APIs:
  - Power_control_mode_thread: Suspended when application starts, then resumed by the thread SW1, and set the MCU into a power control mode specified in a message from the SW1 thread.
  - LPM_mode_thread: Suspended when application starts, then resumed by the thread SW2, and enter a LPM mode specified in a message from the SW2 thread.

Figure 35 shows these partitions and threads.

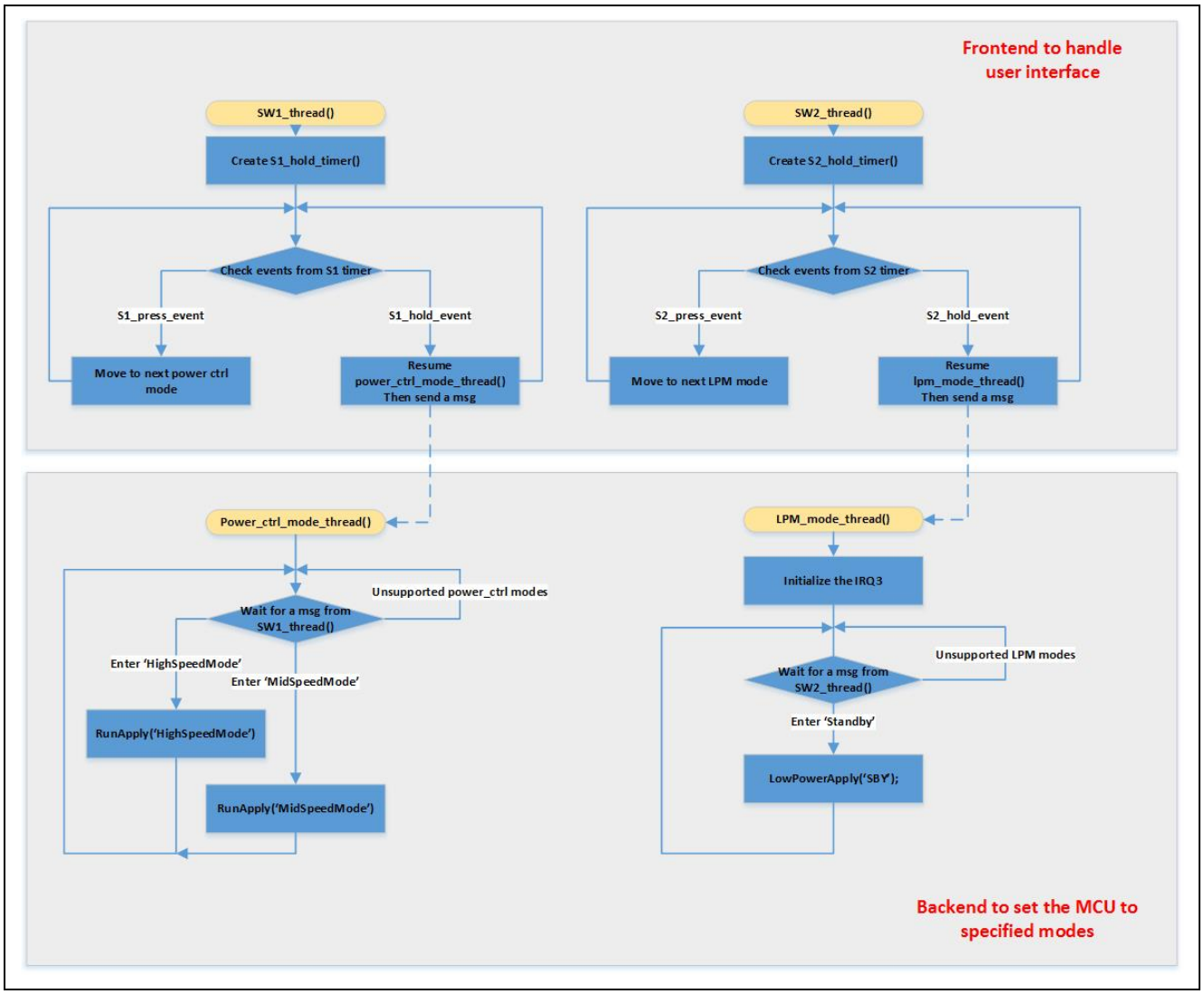

Figure 35 Software architecture of the PPv2 project

6.3 Project Configuration

Following the instructions given in the section 4, Figure 36 shows four threads associated ThreadX event and the message queues that are created with the ISDE e^2 studio and the SSP Configurator.

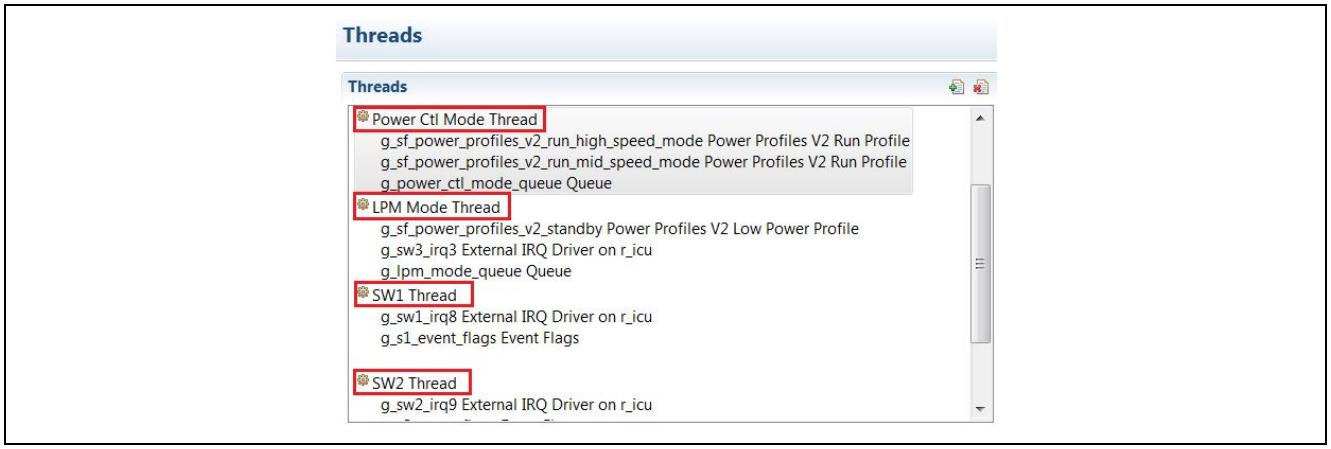

Figure 36 Four threads created in the PPv2 application project

6.3.1 Configuration of the Power_Ctl_Mode_Thread

Currently, two different power control modes `high_speed_mode` and `mid_speed_mode` are added to the `power_ctl_thread`. This thread can be expanded by adding more power modes.

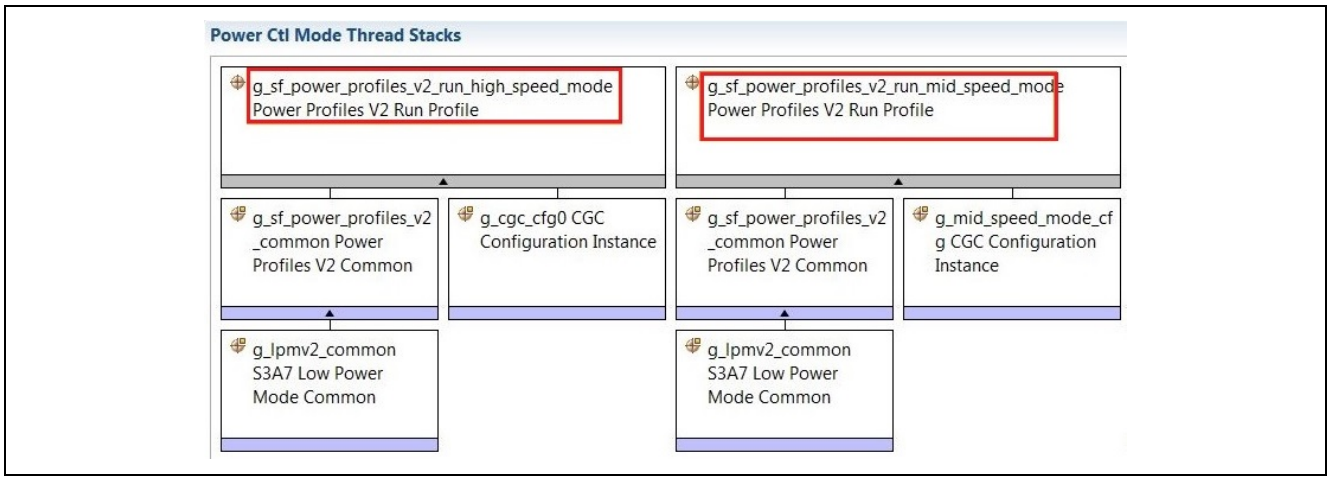

Figure 37 High-speed and middle speed modes of the `Power_Ctl_Mode_Thread` in the project

The Power Ctl Mode thread is suspended when the application starts, and is resumed by the thread SW1 when users holds the S1 for selecting a power control mode. Figure 38 show how its property view is set.

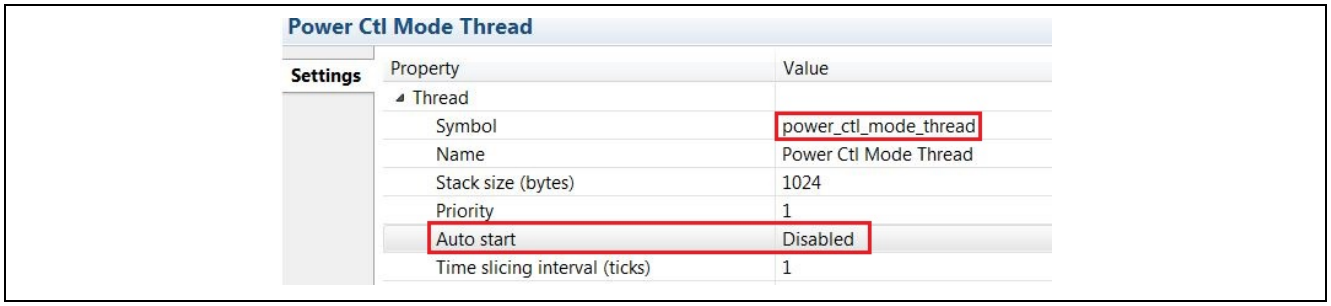

Figure 38 The power_ctl_mode_thread will be suspended when the application starts

Both power control modes use the default pin configuration table `g_bsp_pin_cfg` as their IO port configurations.

- The CGC configuration of the `g_sf_power_profiles_v2_run_high_speed_mode` uses a 24 MHz HOCO as the system clock. Other clocks are generated as follows:

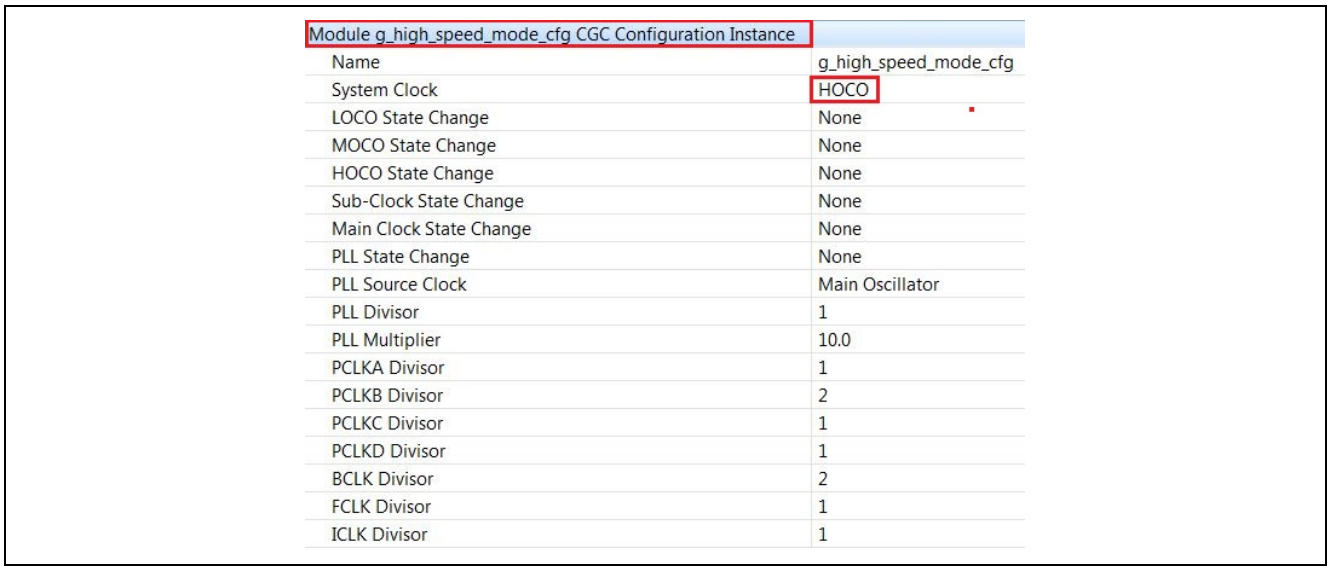

Figure 39 CGC configuration of a 24 MHz High-speed Mode in the project

- The CGC configuration of the `g_sf_power_profiles_v2_run_mid_speed_mode` uses a 8 MHz MOCO as the system clock. Other clocks are generated as follows.

Module g_mid_speed_mode_cfg CGC Configuration Instance	
Name	g_mid_speed_mode_cfg
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 Source Clock	Main Oscillator
PLL Divisor	1
PLL Multiplier	10.0
PCLKA Divisor	1
PCLKB Divisor	2
PCLKC Divisor	1
PCLKD Divisor	1
BCLK Divisor	2
FCLK Divisor	1
ICLK Divisor	1

Figure 40 CGC configuration of an 8 MHz Middle-speed Mode in the project

6.3.2 Configuration of the LPM_Mode_Thread

Currently, only one PPv2 framework module, the Standby module, and the External IRQ HAL driver are added to this thread. However, it can easily be extended to add more PPv2 Low Power modules as shown in Figure 33.

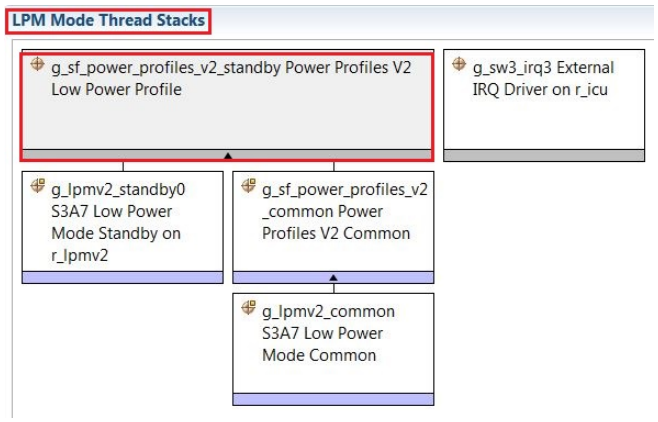

Figure 41 Modules of the LPM_thread_mode in the PPv2 application project

- Similar to the Power Ctl Mode thread, the LPM Mode thread will be suspended till the thread SW2 resumes it.
- The IO functionality of pre- and post- Software Standby mode are defined with two pin configuration tables, `g_run_pin_cfg` and `g_sby_pin_cfg`.

Module g_sf_power_profiles_v2_standby Power Profiles V2 Low Power Profile	
Name	g_sf_power_profiles_v2_standby
Callback (Low Power Exit Event N/A when using Deep Software Standby)	cb_pre_post_sby_fun
Low power entry pin configuration table	g_sby_pin_cfg
Low power exit pin configuration table	g_run_pin_cfg

Figure 42 Two pin configuration tables used in the PPv2 application project

Figure 43 shows the difference between these two pin configurations.

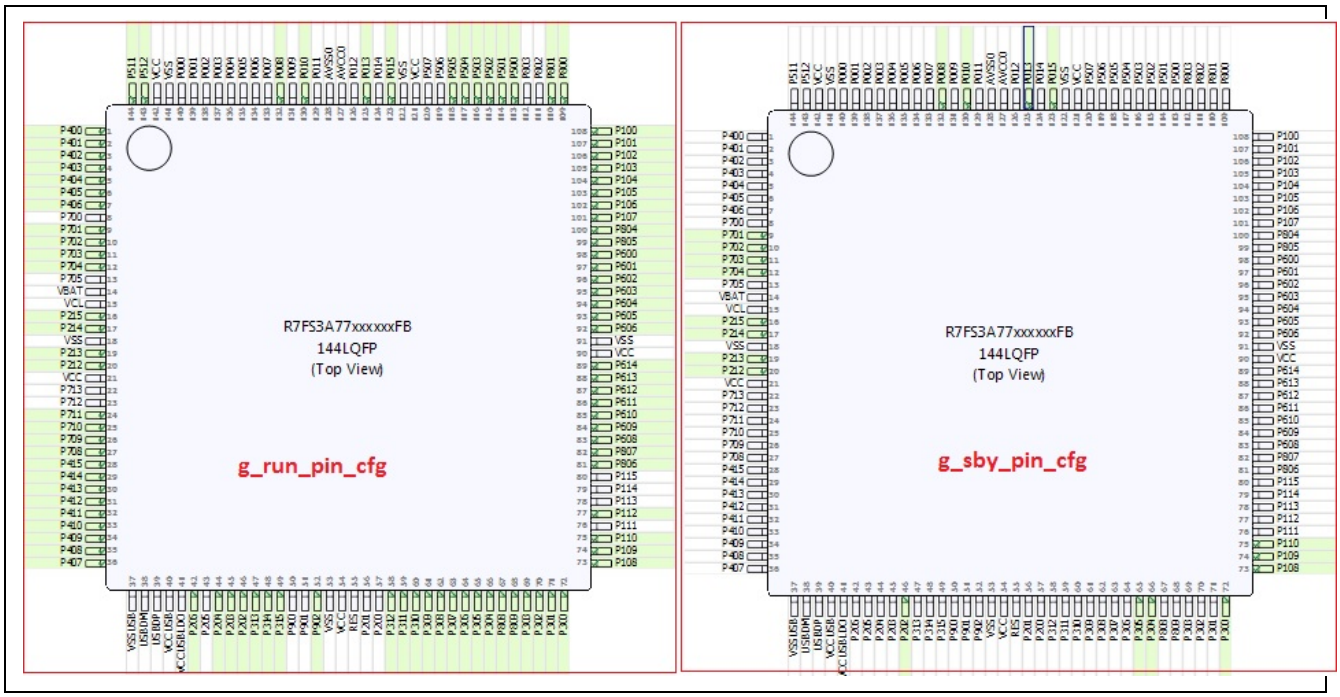

Figure 43 Differences between two pin configurations in the project

As seen in the preceding graphic, those assigned pins are marked with green, and so `g_run_pin_cfg` allows more IO operations than `g_sby_pin_cfg`.

- The IRQ3 is enabled as a condition for transitioning the Software Standby mode to the Normal mode.

Select Standby Exit Sources	Select fields below:
IRQ0	Disabled
IRQ1	Disabled
IRQ2	Disabled
IRQ3	Enabled
IRQ4	Disabled

Figure 44 Configuration of the transition between the Standby and Normal modes

7. Running the PPv2 Framework Module Application Example

Next, how to program and run the project on the DK-S3A7 board.

7.1 Powering up the Board

- To connect to the board:
  - Turn on the JTAG and PBs with the switch S5
  - Connect the Micro USB end of the supplied USB cable to the DK-S3A7 v2.0 board J15 connector (DEBUG_USB).

Note: The kit contains a SEGGER J-Link® On-board (OB). The J-Link provides full debug and programming capabilities for the DK-S3A7 board.

  - Connect the other end of the USB cable to the USB port on your workstation.
- To program the board:
  For instructions on importing the project into e^2 studio and building/running the project, see the *Renesas Synergy Project Import Guide* (Renesas Electronics America, Inc., July, 2017) included with the application project.

7.2 Verifying the Demo

The user interface for testing this project is shown as follows.

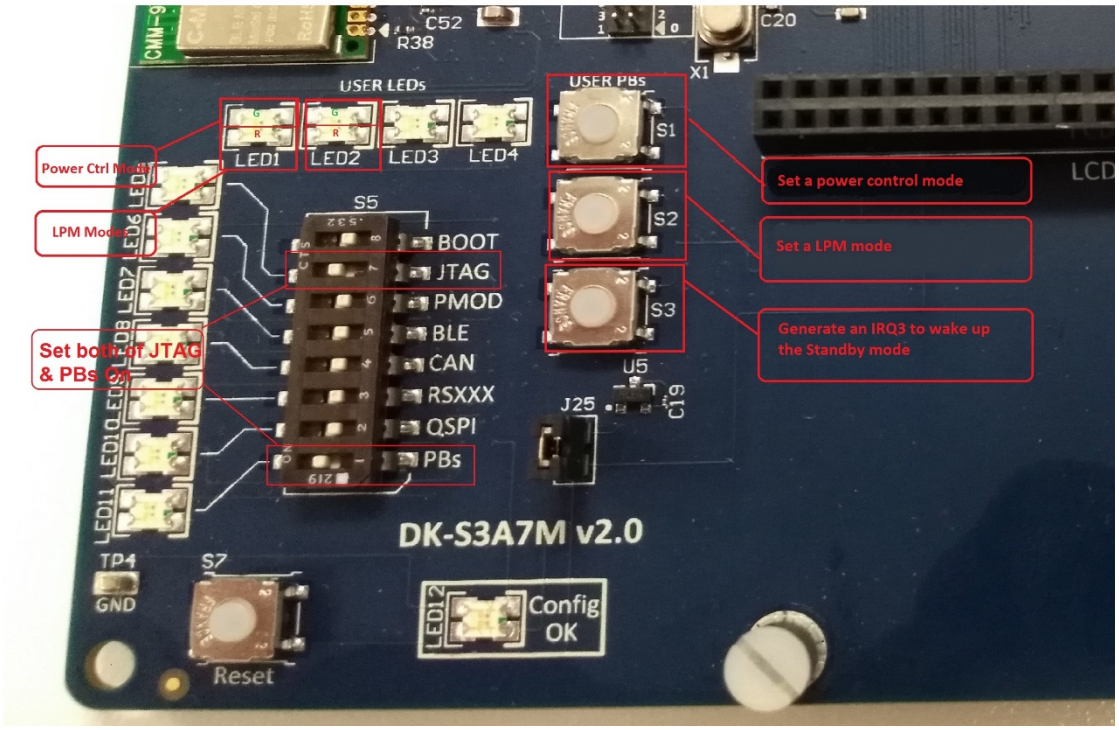

Figure 45 Demonstration setup on the DK-S3A7

The debugger may block the MCU to enter the Standby/Deep Standby modes, so you need to reset the board by pressing the button S7; or, perform a full power-on cycle by unplugging and plugging the USB connection from the J15 connector.

Buttons S1, S2 and S3 are used to enter selected power control modes, which the LEDs also indicate:

- Press the S1 button to enumerate different power control modes, the High-speed mode (default) and the Middle-speed mode
- Press and hold the S1 button to set the selected power control mode. The LED1_G blinks 4 times.
- Press the S2 button to switch the LPM mode between default Normal mode and the Software Standby mode.
- Press and hold S2 to set the selected LPM modes. The LED1_R blinks 4 times, following which all LEDs go OFF.
- Press the S3 to generate IRQ3 for waking up the MCU from the Software Standby mode.
- Repeat operations above to make transitions between different power control modes and the LPM modes.

8. Customizing the PPv2 Framework Module for a Target Application

- Users can easily substitute the potentiometer with other analog sources, such as light sensors, motion sensors, and re-define the ADC window functions for different wakeup conditions to create more complex applications.
- Users can add other power control modes and LPM modes with different CGC configuration and different exit sources for Snooze, Software Standby, or Deep Software Standby modes.
- Users can also measure power consumption of different modes on the J22.

Note: A proper board modification and testing environment must be set to obtain standby currents compatible with the specifications in the particular *Synergy Microcontroller Group User's Manual*.

9. References

IAR Systems. (n.d.). *IAR Embedded Workbench, Getting Started with IAR Embedded Workbench for Renesas Synergy.*

Keil. (2016). *CMSIS-Pack (v 1.4.1).* Retrieved from Usage and Description: http://www.keil.com/pack/doc/CMSIS/Pack/html/pages.html

Renesas Electronics America Inc. (n.d.). *Renesas Synergy™ Software Package, User's Manual.*

Renesas Electronics America Inc. (2015). *USBX CDC ACM (Host).*

Renesas Electronics America Inc. (2015). *USBX Device for CDCACM .*

Renesas Electronics America Inc. (n.d.). *The Renessas e^2 studio ISDE 5.4.0.023.*

Renesas Electronics America, Inc. (Feb, 2017). *Release Notes v1.2.0.*

Renesas Electronics America, Inc. (Feb, 2018). *Renesas Synergy™ Project Import Guide*

Renesas. (Feb 2016). *S3A7 User's Manual: Microcontrollers.* Renesas Electronics.

USB 3.0 Promoter Group. (2014). *USB 3.1 Specification.*

Website and Support

Support: https://synergygallery.renesas.com/support

Technical Contact Details:

- America: https://www.renesas.com/en-us/support/contact.html
- Europe: https://www.renesas.com/en-eu/support/contact.html
- Japan: https://www.renesas.com/ja-jp/support/contact.html

Revision History

		Description	
Rev.	**Date**	**Page**	**Summary**
1.00	Mar 7, 2018	—	Initial version

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

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

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

RENESAS

Renesas Electronics Corporation

SALES OFFICES http://www.renesas.com

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

Renesas Electronics America Inc.
1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A.
Tel: +1-408-432-8888, Fax: +1-408-434-5351

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700, Fax: +44-1628-651-804

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709 Quantum Plaza, No.27 ZhichunLu, Haidian District, Beijing, 100191 P. R. China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, 200333 P. R. China
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HAL 2nd Stage, Indiranagar, Bangalore 560 038, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
17F, KAMCO Yangjae Tower, 262, Gangnam-daero, Gangnam-gu, Seoul, 06265 Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5338

Colophon 7.0