Renesas Synergy™ Platform

Power Profiles v2 (PPv2) Framework for PK-S5D9

R20AN0474EU0103
Rev.1.03
Mar 7, 2018

Introduction

This Application Note will enable you to effectively use the Power Profiles v2 (PPv2) Framework module in your own design, and reduce your effort in using the power control modes and Low Power Modes (LPMs) provided in the Renesas Synergy™ MCU Group. This document introduces some basic knowledge about the Synergy MCU power control modes and LPMs, which will be abstracted into power profile configurations and APIs in the PPv2. It also describes the PPv2 framework operational flow, and a basic implementation procedure for using the PPv2 in applications. This document provides an example of using the PPv2 Framework to use the power control mode and LPM mode transitions.

The PPv2 Framework provides more control for you 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 Synergy Software Program (SSP) Framework using the LPM v1 driver. The Power Profiles v1 (PPv1) and PPv2 Framework are not compatible. PPv1 and PPv2 Frameworks cannot be used in the same project. For all new projects, it is recommended that applications use the PPv2 Framework.

Using this document, you will be able to 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 an efficient starting point. References to more detailed API descriptions that illustrate more advanced uses of the module are available in the *SSP User's Manual 1.4.0* or later, which is a valuable resource for creating more complex designs.

Required Resources

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

- PK-S5D9 (v1.0) Synergy MCU Kit
- Micro USB cables
- Synergy e^2 studio ISDE 5.4.0.023 or later
- IAR Embedded Workbench® for Renesas Synergy™ 7.71.3 or later
- Synergy Software Package (SSP) v1.4.0 or later
- Synergy Standalone Configurator (SSC) 5.4.0.023 or later

You can download the required software from the Renesas Synergy™ Solutions Gallery.

Prerequisites

As the user of this application note, it is assumed that you have some experience with the Renesas Synergy 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 Groups Series.

Contents

1. Synergy Power Profiles Overview

1.1 Power Profile v2 Framework

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 using sophisticated power saving features of the Synergy MCU Series. The PPv2 modules can be instantiated into ThreadX® RTOS aware multithreads, configured with the SSP configurator to the expected power control modes and LPM modes, and then inserted into your application. You are required to understand the power saving features of the Synergy MCU Groups to properly set up the PPv2 modules.

1.2 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

1.3 Synergy MCU Power Profile Overview

Synergy MCU Groups use two sets of control registers, Power Control Modes (PCMs) and Low Power Modes (LPMs), to support different power or performance requirements. This section describes the 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.

1.3.1 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 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. You can adjust the System Clock (ICLK), Peripheral Module Clocks (PCLKB, PCLKD), External Bus Clock (BCLK), and others 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 shown as follows.

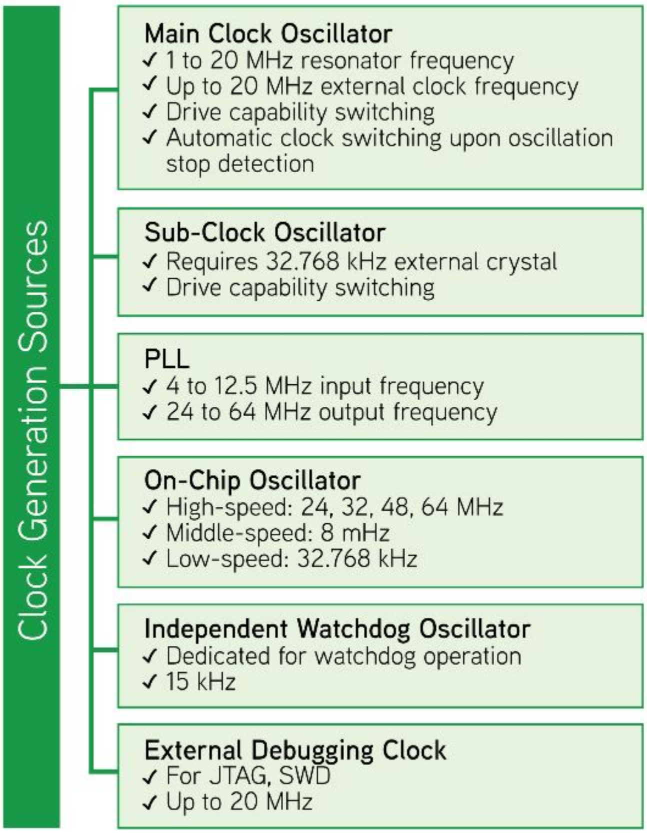


Figure 1 Clock Generation source for Synergy MCU S3 Series

The Synergy MCU Series define some power control modes such as High-speed mode, Middle-speed mode, Low speed mode, Low-voltage mode, and Subosc-speed mode with different clock generation sources and frequency ranges. The corresponding power consumption levels are decreased gradually. 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 Synergy MCU Groups

Synergy MCU Groups	Power Control Modes				
	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 five 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

Power Consumption

high

low

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

The power control modes are defined in the *Synergy MCU User's Manuals.*

1.3.2 Synergy MCU Low Power Modes

To further reduce the power consumption, the Synergy MCUs provide Low Power Modes (LPMs) by allowing operation of a peripheral while keeping the CPU and other peripherals clock gated or powered down. There are four possible LPM modes: Sleep mode, Software Standby mode, Deep Software Standby mode, and Snooze mode. As an example, the core is clock gated and peripherals are available in Sleep mode. The core and most peripherals are clock gated but data is 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 MCU power consumption and throughput are illustrated in the following figure.

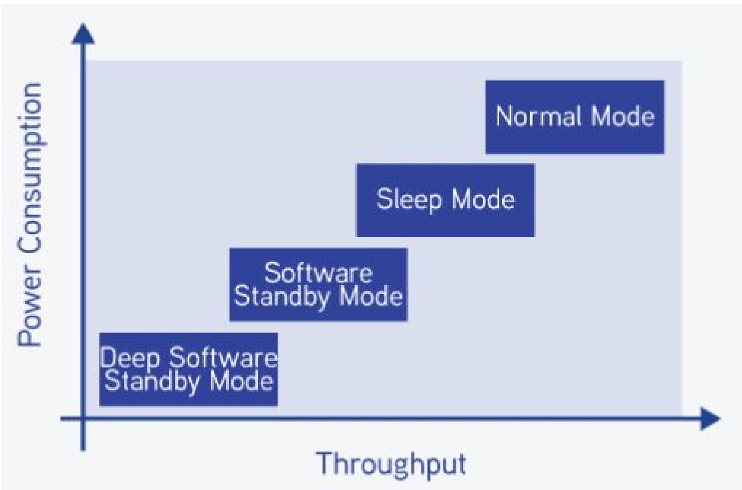


Figure 3 Power consumption and throughput of the LPM modes

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

(a) **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, a SRAM parity error reset, or a reset caused by an IWDT or a WDT underflow.

(b) **Software Standby Mode (SBY)**

In SBY the CPU and most of the on-chip peripheral functions and oscillators are clock gated. The contents of the CPU internal registers and SRAM data, the states of on-chip peripheral functions, and the I/O ports are retained. SBY 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 SBY.

(c) **Deep-Software Standby Mode (DSBY)**

In DSBY more power consumption reduction compared to SBY is achieved by stopping more components such as the 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.

(d) **Snooze (SNZ)**

SNZ feature provides operational flexibility to dramatically reduce current consumptions. SNZ is an extension of the Software Standby mode where some peripheral modules can operate without waking up the CPU. SNZ can be entered through the Software Standby mode through some interrupt sources and woken up by those available interrupts in the Software Standby mode. A general description of those LPM modes is summarized in the following table, but their detailed configurations are defined in the corresponding Synergy Microcontroller Groups User's Manuals.

Table 3 LPM modes and their basic configurations

Mode of Operation	Core	Flash	SRAM	RTC, AGT, VBATT, LVD	Other Peripherals	I/O 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 operate or disable options can be selected by control registers. Some modules are also controlled by the module-stop bit.

(e) **Transitions between the Low Power Modes**

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

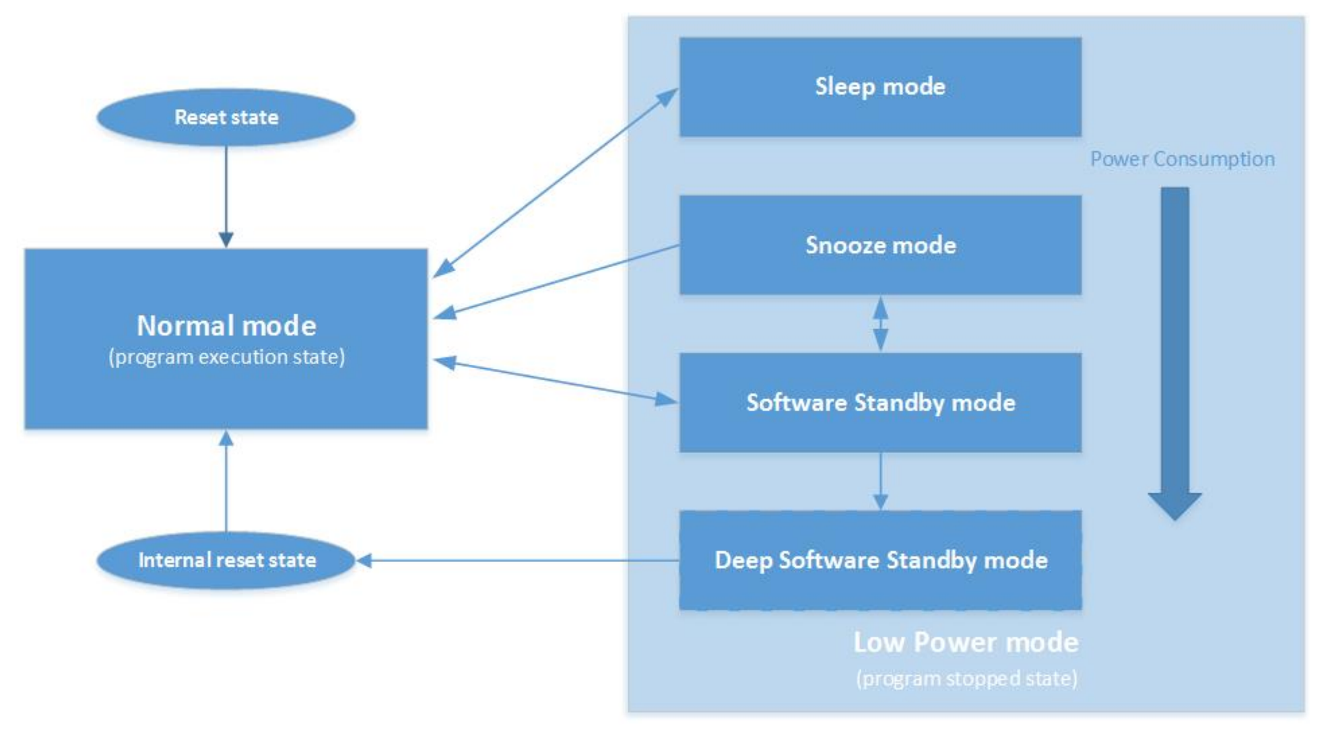


Figure 4 Transitions between the Normal mode and 4 LPM modes

Note: The preceding graphic only shows possible LPM modes. It does not include the transition conditions of each mode, as different Synergy MCU Groups have a different number of the LPM modes and different transition conditions. In addition, their actual power consumption is also dependent on the LPM mode configurations. Detailed information is provided in the corresponding Synergy *Microcontroller Groups User's Manuals*.

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, IO Port, 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 is illustrated in the following diagram.

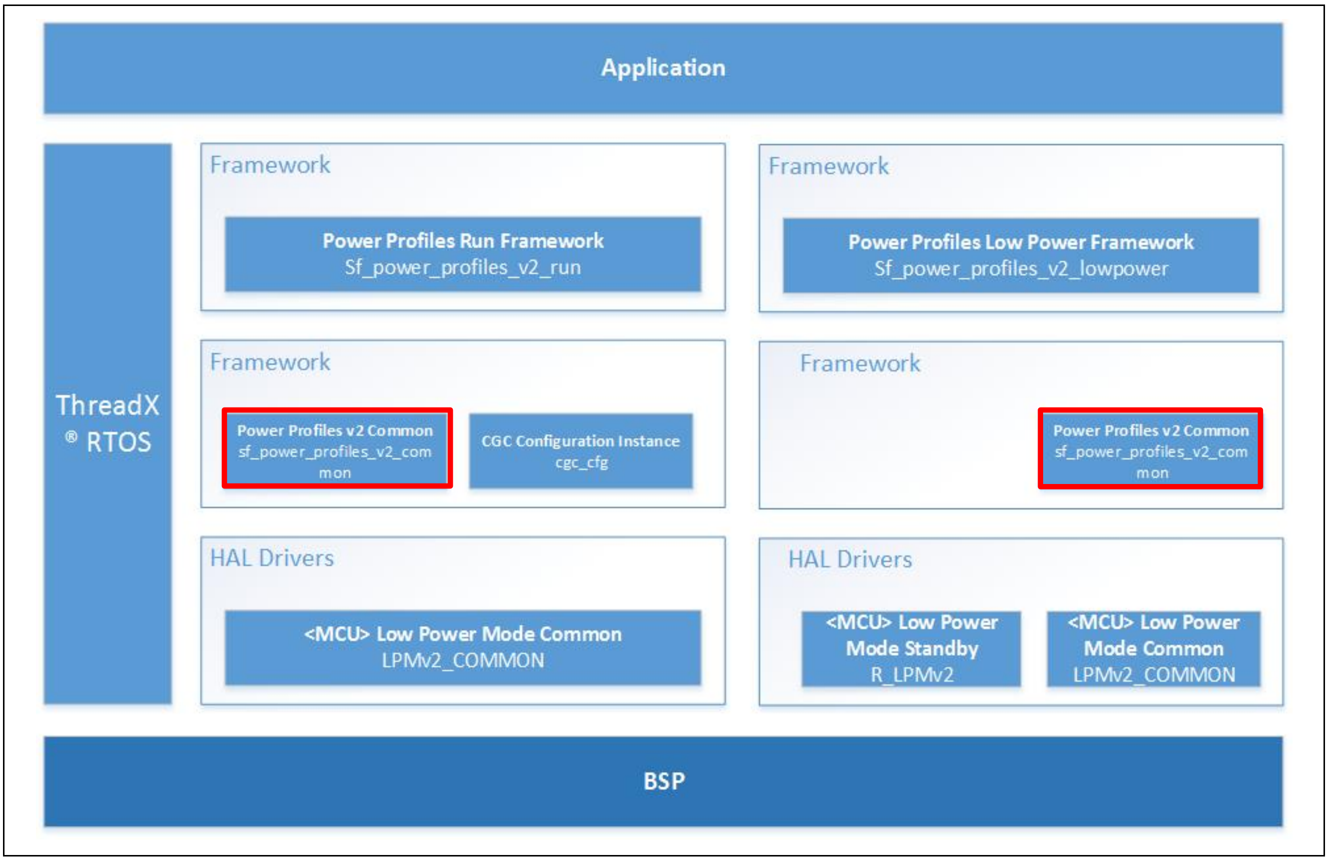


Figure 5 PPv2 Framework Stack

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.

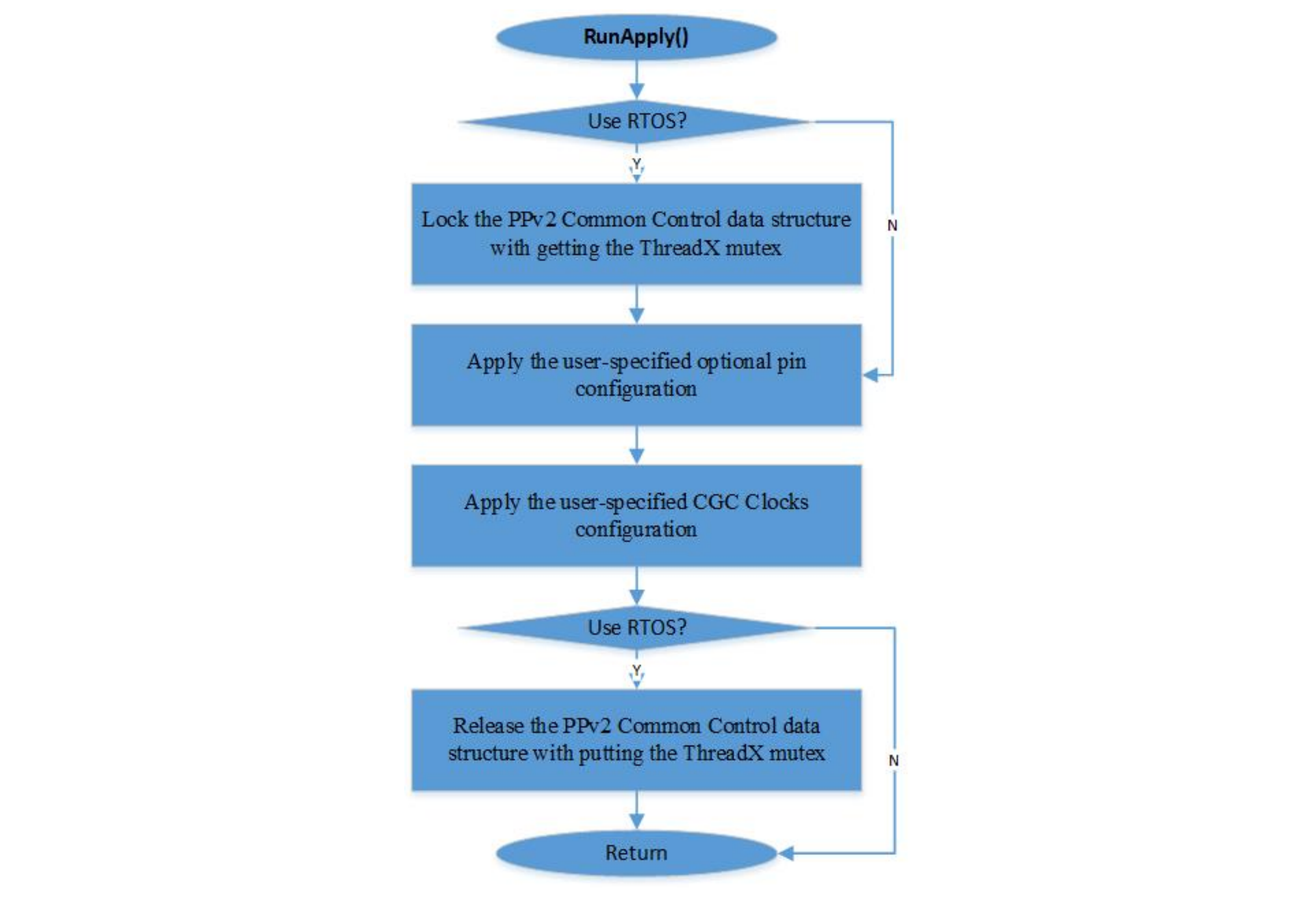


Figure 6 PPv2 `RunApply()` process

The **Low Power Profile** uses an LPM v2 configuration and 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 corresponding Synergy *Microcontroller Group User's Manuals* for details on the available Low Power modes.

The internal function `LowPowerApply()`API performs the tasks shown in the following figure, in order.

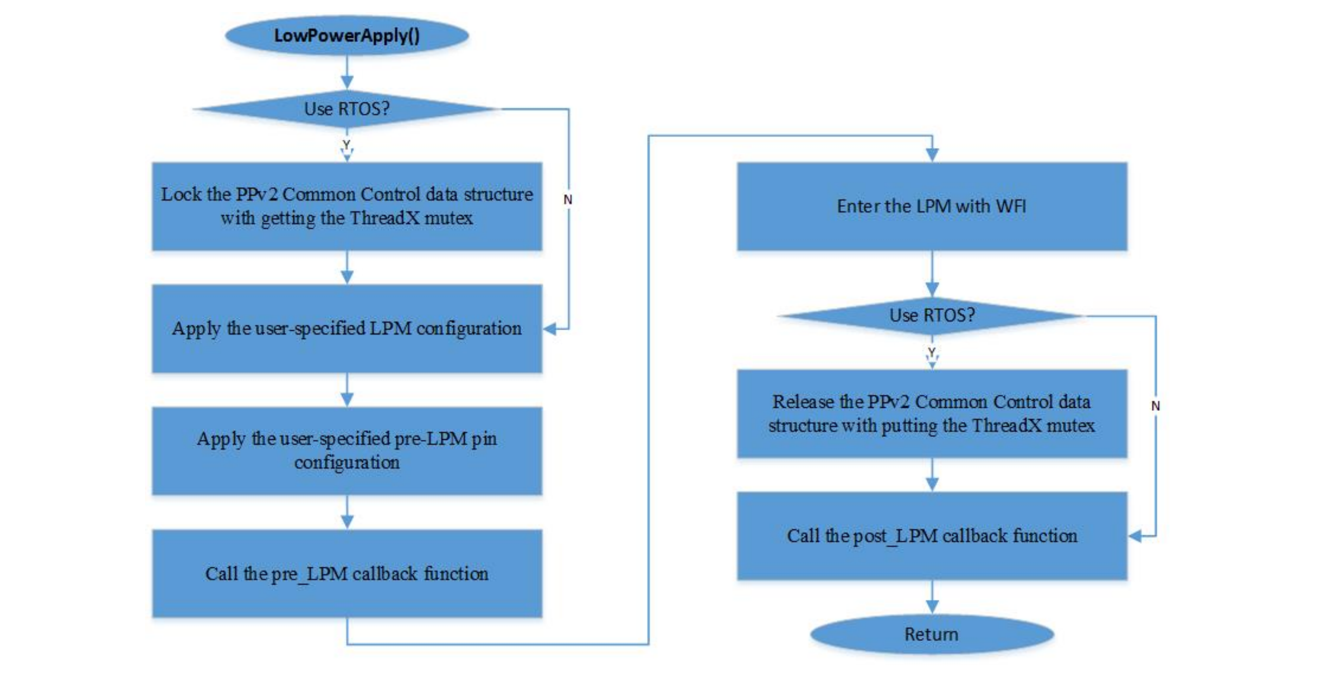


Figure 7 The PPv2 LowPowerApply() process

2.2 PPv2 Low Power Modes Operational Notes

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

Power Profiles v1 and Power Profiles v2

The PPv1 and PPv2 Frameworks are not compatible, so do not use both frameworks in the same project. For all new projects, it is recommended that applications use the 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`. It will then be instantiated in the automatically generated file `common_data.c`. So, the 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 application projects, and instantiated in the file `common_data.c` that will not increase the code size of a project. Another instance of the CGC Driver is required for the 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 use ThreadX intrinsic objects like a mutex for multithread applications when used with ThreadX.

Special Consideration on Multithreads Applications

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

Debugger Usage in LPM Modes

By default, debuggers will prevent MCUs from entering LPM modes since Arm® core is accessed by CoreSight™ debugging elements. There are two approaches to disconnect their attachment as follows:

- Perform a full power-on reset cycle by unplugging and plugging the USB connection for JTAG.
- Enable Low Power Handling in the Synergy e² studio debugger and utilize a provided J-LINK script in **Debugger tab** > **Connection Settings**, as shown in the following graphic.

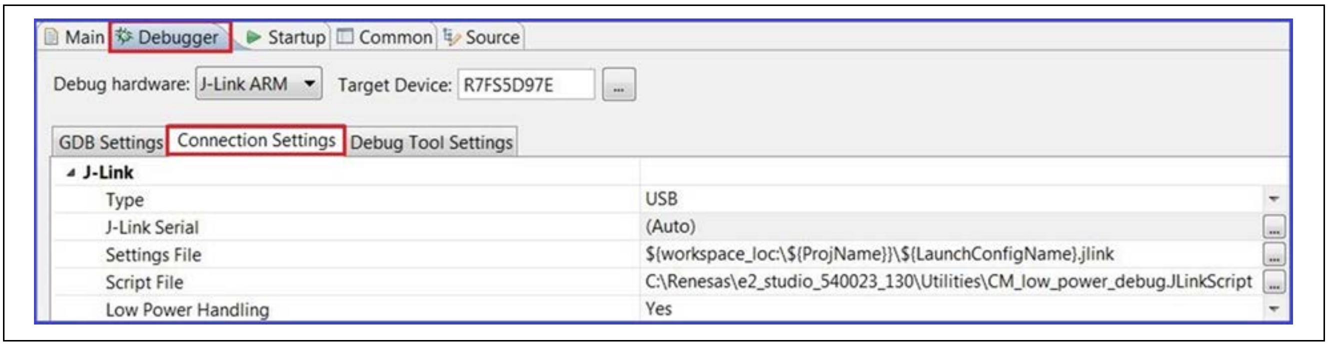


Figure 8 Enabling Low Power Handling in Synergy e² studio Debugger

The `CM_low_power_debug.JLinkScript` script will be provided on request.

2.3 PPv2 Module Limitations

- The PPv2 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 the API function `open()`. This is not a PPv2 limitation, but a general approach to initializing a Framework module without using ThreadX RTOS.
- The PPv2 Framework does not handle starting or stopping MCU peripherals. Since no property view is available to select which peripherals are to be stopped in a LPM mode, you must stop them manually.
- Current version of PPv2 Framework only supports the S124, S128, S3A7, S5D9, and S7G2 Synergy MCUs.

- Current version of PPv2 Framework does not support the transition from Snooze to Normal in a LPM Standby mode.

3. Power Profile v2 Module APIs Overview

This section describes the PPv2 API functions at the framework layer, and 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 follows”

```
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 is as follows:

```
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 is as follows:

```
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 in the following table.

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);`

Function Name	Example API Call and Description
	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.

Table 5 The return values of the PPv2 Framework APIs are defined as shown 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:

- 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 concentrates on the Step 2 and Step 3 to add PPv2 Framework profile modules into a thread. The profile configurations on the Step 2 and Step 4 will be discussed in 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 *e² studio ISDE User's Manual*, 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.

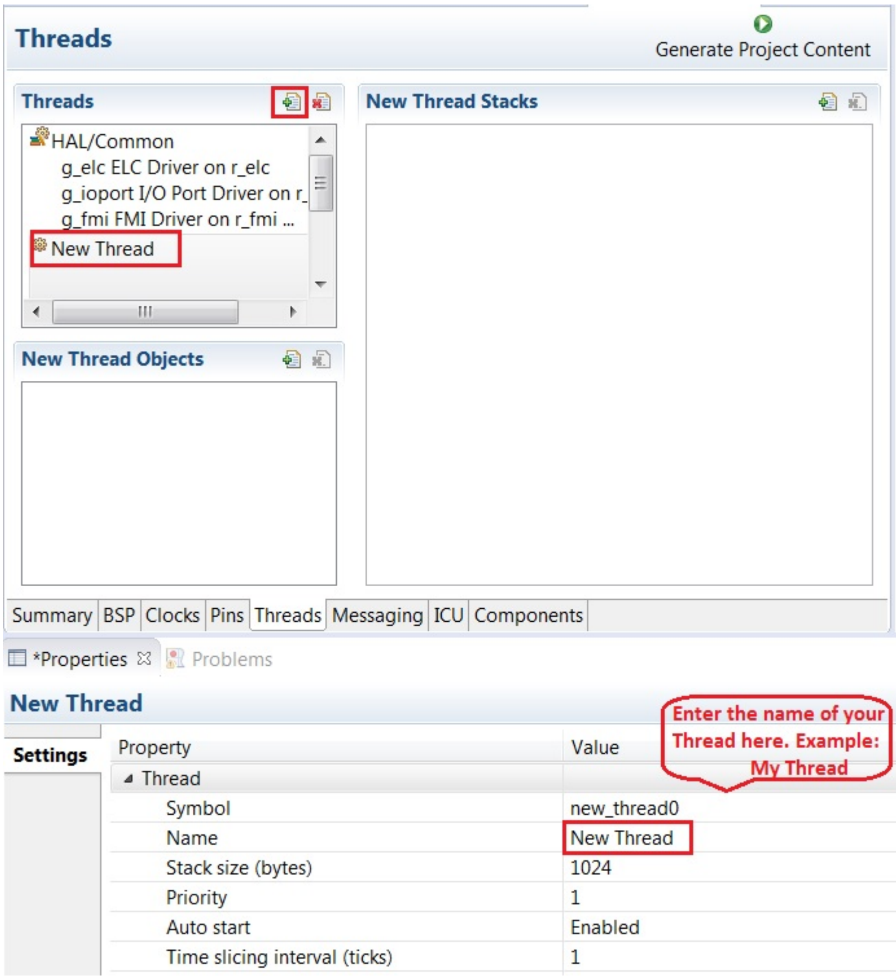


Figure 9 Adding a new thread on the Thread tab

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

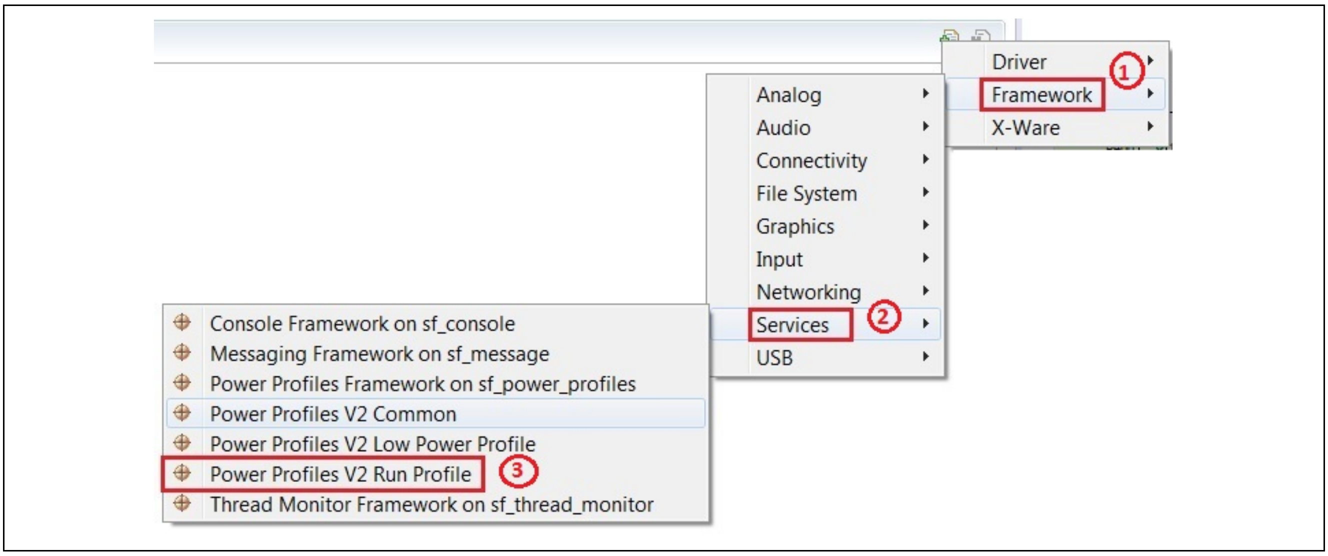


Figure 10 Adding a PPv2 Run Profile module

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.

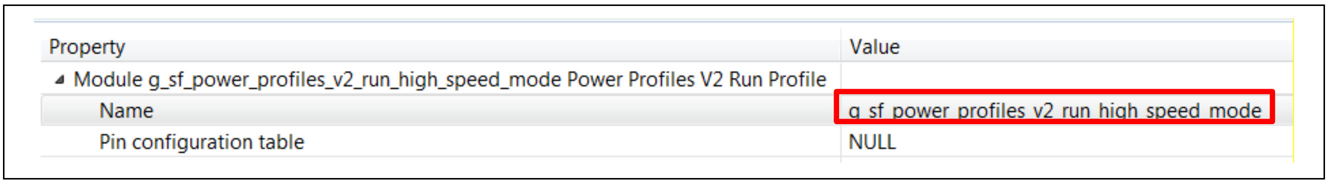


Figure 11 Naming the Run profile module

The created power control modes are shown in the SSP configurator in the following figure. The operating frequencies can be defined by switching clocks on and off, changing clock dividers, and selecting the system clocks in the Properties view of **the CGC Configuration Instance** module, that will be presented in the Section 5.2.

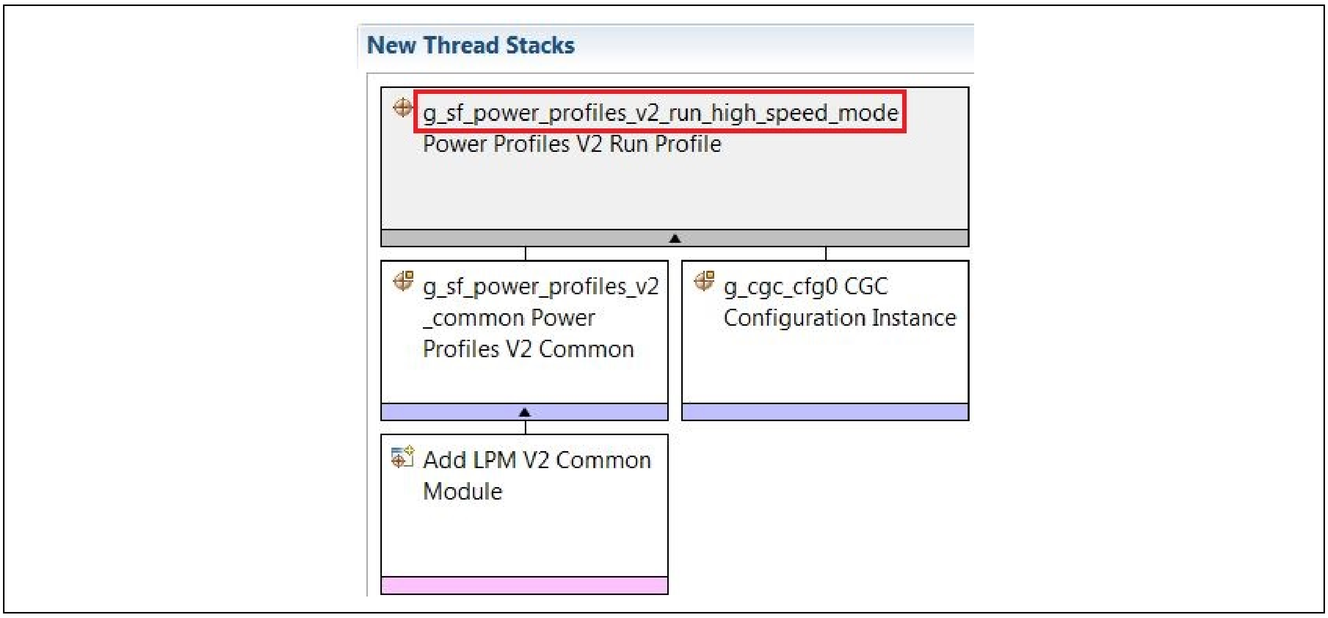


Figure 12 A power control mode defined in the PPv2 Run profile

Then, select a LPM common module from a pull-down menu of the New Thread Stacks containing specific settings of the PPv2 Framework configuration on a selected Synergy MCU Group, such as S5D9 shown in the following figure.

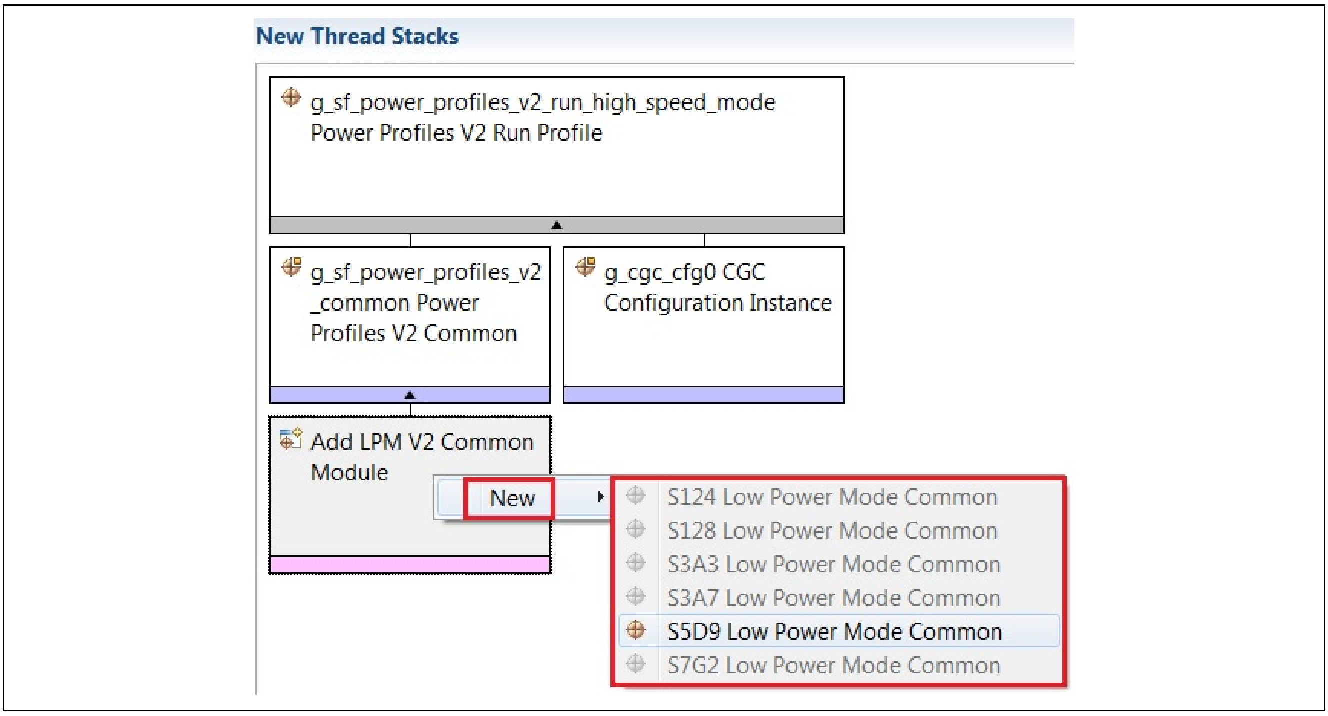


Figure 13 Adding the LPM v2 common module

4.1.2 Adding the PPv2 Low Power Profile Module into a Thread

On the selected thread, click on **New** and select the Low Power Profile module from the pull-down menu.

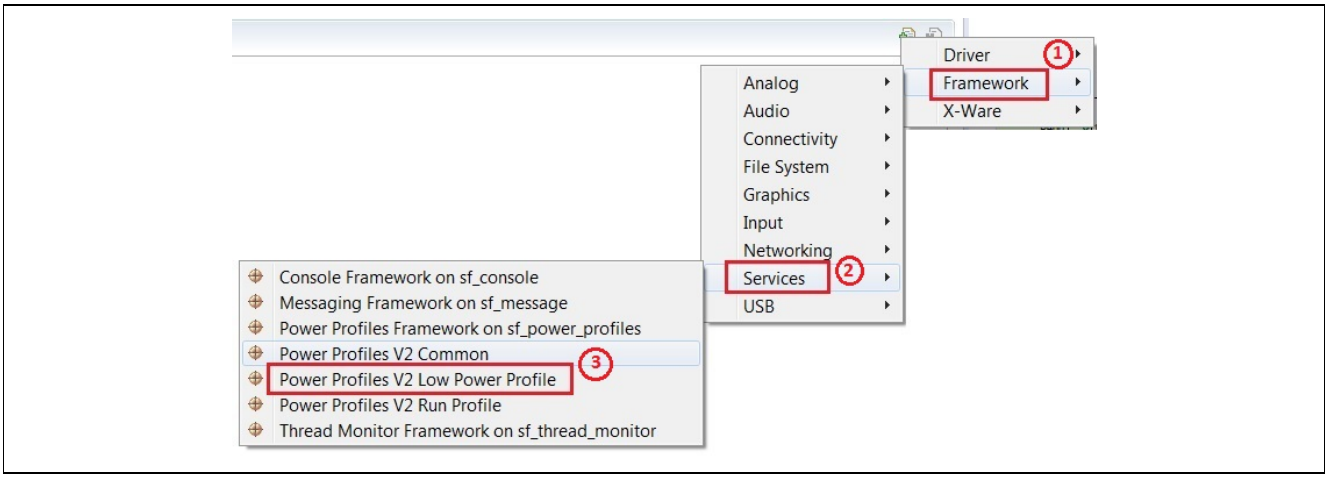


Figure 14 Adding a PPv2 Low Power Profile module

Then, a LPM Low Power Profile module is generated in the SSP configurator. Available LPM modes for the selected MCU will be enabled as shown in the following figure.

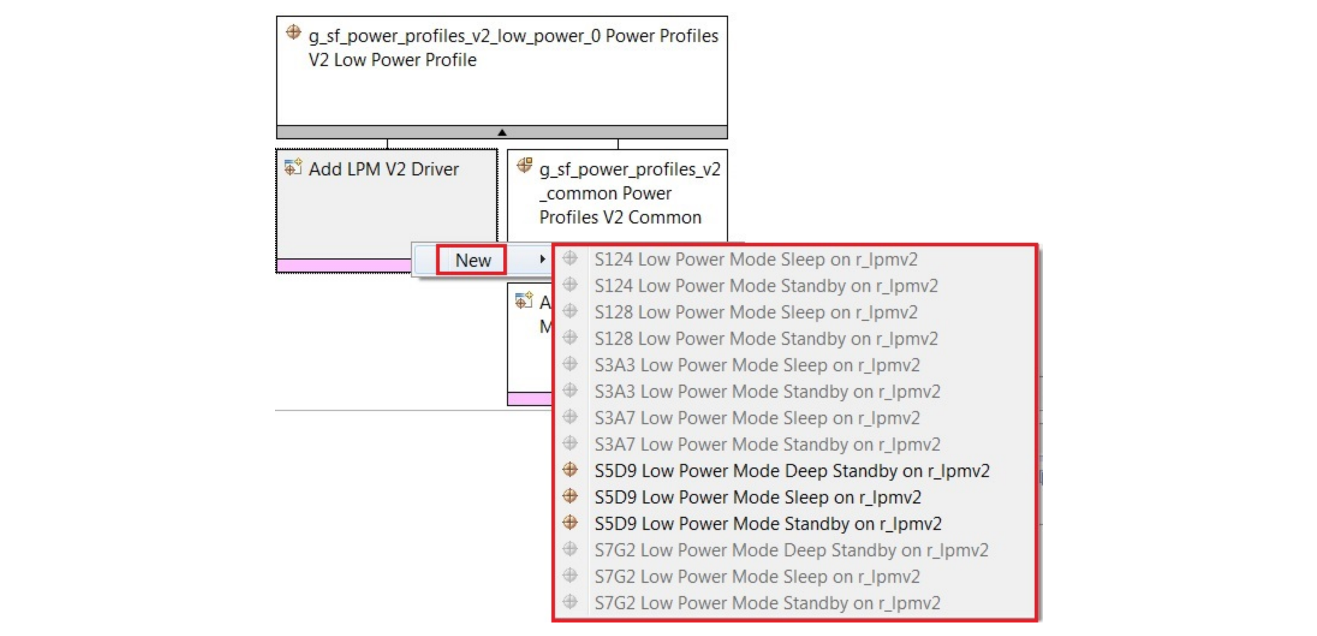


Figure 15 PPv2 Low Power Profile Module

You can add a PPv2 Framework module for representing Sleep mode, Standby mode, or Deep Standby mode in their applications.

4.2 Including PPv2 Framework without using ThreadX

Like 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.

The profile configurations on Step 2 and 4 will be discussed 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 *Synergy e^2 studio ISDE User's Manual*. Then, click on the **Add** button to add the PPv2 Framework Run profile module as shown in the following figure.

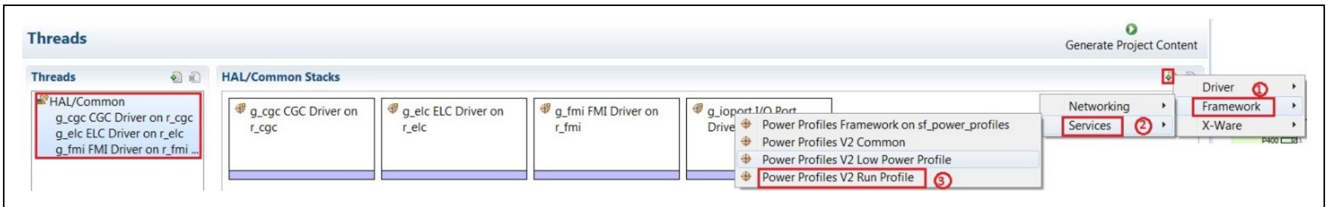


Figure 16 Adding a PPv2 Run Profile module into a HAL/Common stack

Assign a unique name for this Run Profile module, such as `g_sf_power_profiles_v2_run_high_speed_mode`, in the property view. Then click **New** on the LPM common module to select a PPv2 framework configuration for a Synergy MCU Group, as illustrated in Figure 10 to Figure 13.

4.2.2 Adding PPv2 Low Power Profile Module into the HAL/Common Stacks

Like the operations of adding PPv2 Low Power Profile modules into a thread in the Section 4.1.2, different PPv2 LPM profile modules can be added by clicking the **Add** button, then selecting the **Low Power Profile** module from the pull-down menu.

5. Configuring PPv2 Framework Modules

The previous sections describe different ways to include the PPv2 Framework modules into applications. This section will show how to configure the PPv2 Framework profile modules, and provide detailed configuration parameters and recommended values that you can use as applicable in your applications.
Before discussing configurations of the PPv2 Framework Profile modules, the following is a description of a property of a new thread that could be used with PPv2 Framework profiles as shown in the Figure 9 (Section 4.2.2).

Table 6 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

Assuming that a PPv2 Framework Run profile module has been included into a project by performing operations in Section 4, the configuration of that module is discussed in the following section.

5.1.1 I/O Configuration of PPv2 Framework Run profile

A Run profile, such as the following High-speed mode, has the configurations shown in the following figure.

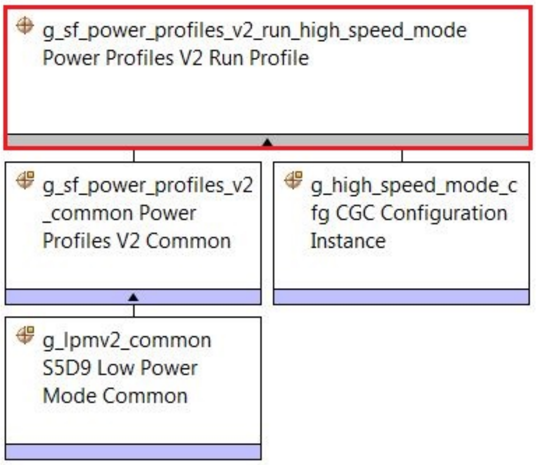


Figure 17 PPv2 Run profile module

Table 7 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 `S5D9-PK.pincfg` by clicking the right mouse button and selecting the **Copy** operation, then **Paste** it into the same project.
- Rename this new pin configuration file as `S5D9-PK_RUN.pincfg.`

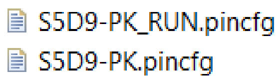


Figure 18 Create a new pin configuration file with Copy and Paste

- Select the new pin configuration from the pull-down menu on the Pins tab of SSP Configurator, then specify a pin configuration name to be generated as shown in the following figure.

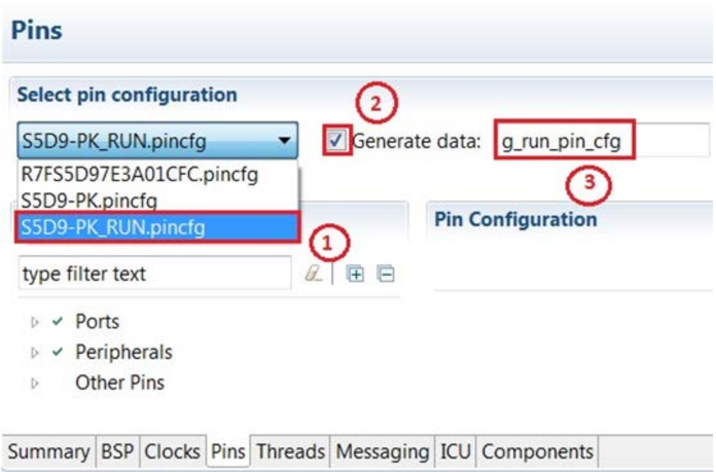


Figure 19 Select a new pin configuration file to be assigned

- Configure the I/O functions on each pin, then generate a new pin configuration when you press the button **Generate Project Content** 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.

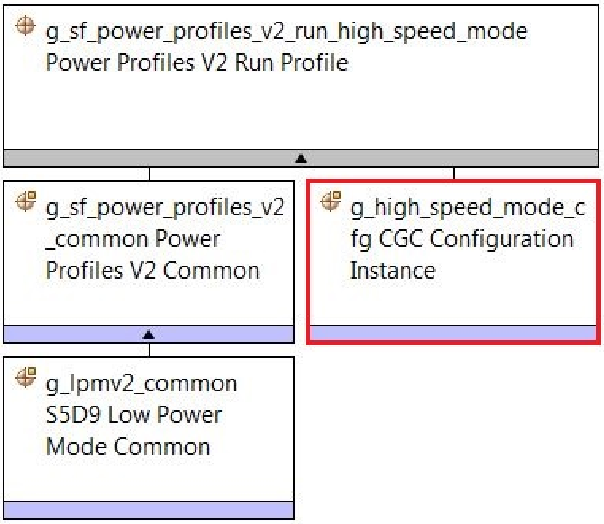


Figure 20 CGC Configuration Instance in the Run profile module

The parameters of the CGC configuration property view are shown as follows.

Table 8 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: The assignments on these CGC parameters must be satisfied with oscillator availability for each power control mode, and the expected frequencies of each operating clock. Their relationships are illustrated in a CGC Block Diagram in the corresponding Synergy MCU User’s Manual.

The maximum operating frequency range for these clocks should not be outside the given ranges in the corresponding *MCU User’s Manual*. The *S5D9 Microcontroller Group User’s Manual* lists this specification in Table 9, as follows.

Table 9 Maximum operating frequency range for Synergy S5D9 MCU Group internal clocks

Parameter	Clock sources	Clock supply	Specifications
Peripheral module clock A (PCLKA)	MOSC, SOSC, HOCO, MOCO, LOCO, PLL	Peripheral modules (ETHERC, EDMAC, USBHS, QSPI, SPI, SCI, SCE7, GLCDC, SDHI, CRC, JPEG engine, DRW, irDA, GPY bus-clock)	Up to 120 MHz*2 Division ratios: 1, 2, 4, 8, 16, 32, 64
Peripheral module clock B (PCLKB)	MOSC, SOSC, HOCO, MOCO, LOCO, PLL	Peripheral modules (IIC, SSIE, SRC, DOC, CAC, CAN, DAC12, POEG, CTSU, AGT, Standby SCRAM, ELC, I/O Ports, RTC, WDT, IWDT, ADC12, KINT, USBFS, ACMPHS, TSN, PDC)	Up to 60 MHz Division ratios: 1, 2, 4, 8, 16, 32, 64
Peripheral module clock C (PCLKC)	MOSC, SOSC, HOCO, MOCO, LOCO, PLL	Peripheral module (ADC12 conversion clock)	Up to 60 MHz Division ratios: 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 120 MHz Division ratios: 1, 2, 4, 8, 16, 32, 64
Flash interface clock (FCLK)	MOSC, SOSC, HOCO, MOCO, LOCO, PLL	Flash interface	Up to 60 MHz (P/E) Up to 60 MHz (read) *1 Division ratios: 1, 2, 4, 8, 16, 32, 64
External bus clock (BCLK)	MOSC, SOSC, HOCO, MOCO, LOCO, PLL	External bus	Up to 120 MHz Division ratios: 1, 2, 4, 8, 16, 32, 64
EBCLK pin output (EBCLK)	BCLK or ½ BCLK	EBCLK pin	Up to 60 MHz Division ratios: 1, 2
SDCLK pin output (SDCLK)	BCLK	SDCLK pin	Up to 120 MHz
USB clock (CABMCLK)	PLL	USB	48 MHz Division ratios: 3, 4, 5,
USB-PHY clock (USBMCLCK)	MOSC	USB-PHY	12, 20, 25 MHz
Can CLOCK (CANMCLK)	MOSC	CAN	8 to 24 MHz
LCD_CLCK pin output (LCD_CLK) and graphic LCD pixel clock (PXCLK)	LCD_EXTCLK, PLL output	LCD_CLK pin, peripheral module (Graphics LLCD Controller)	Up to 54 MHz (parallel RGB) Up to 60 MHz (serial RGB) LCD CLK : PXCLK = 1.1 (parallel RGB)

As the current property view does not validate your assignments automatically, you can use the CGC panel of the SSP Configurator to check your assignments first. The following screen shot of the CGC configuration is an example of the High-Speed mode.

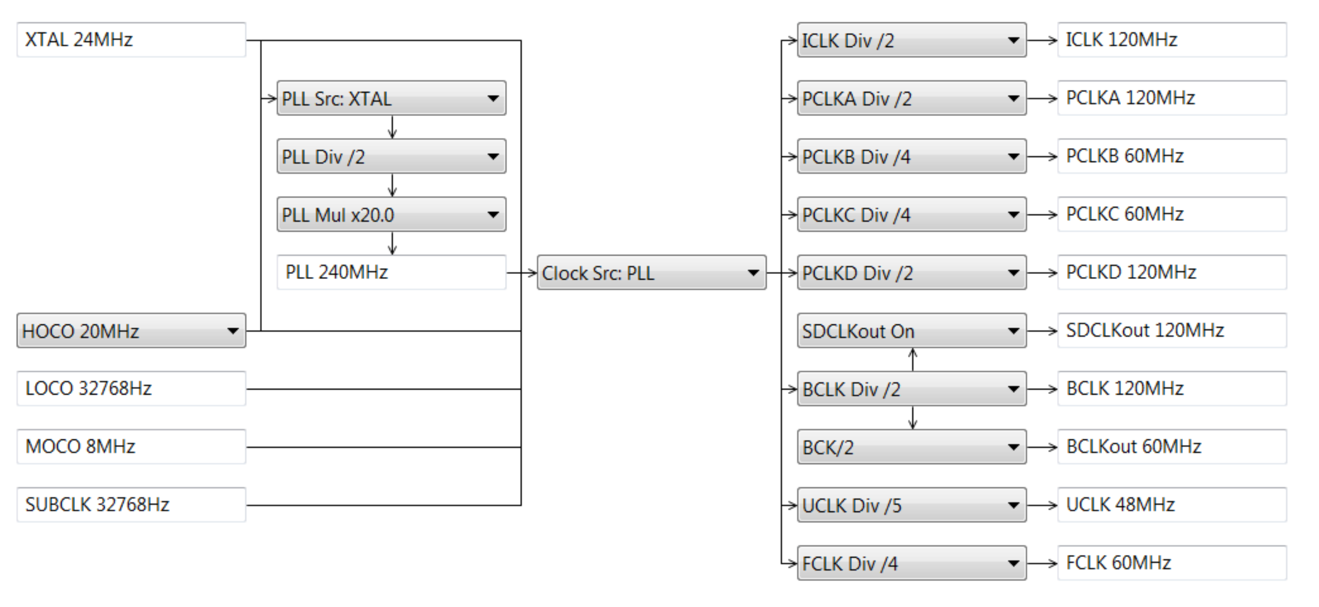


Figure 21 Checking the clock setting for a power control mode

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 3 LPM modes: Sleep, Software Standby, and Deep Software Standby modes, available in the pull-down menu of the Low Power Profile module.

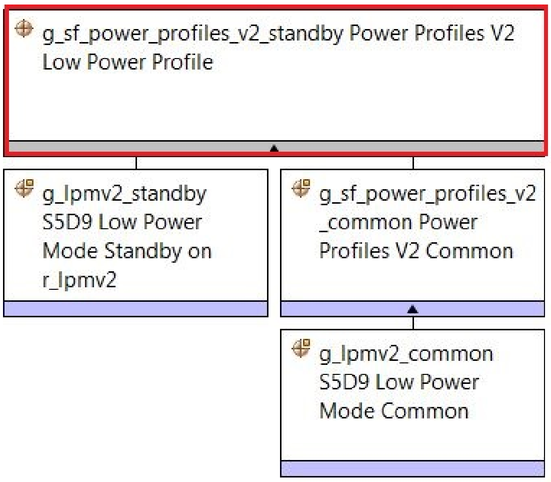


Figure 22 PPv2 Low Power Profile Standby module

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

Table 10 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

They will be 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. Only the module name can be changed.

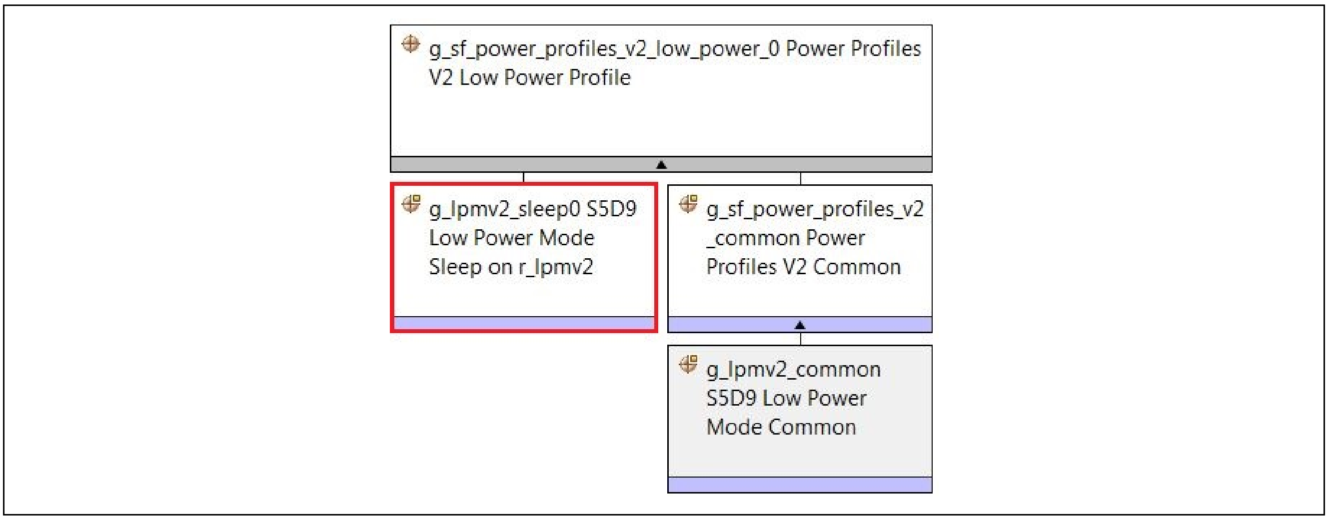


Figure 23 PPv2 Low Power Profile Sleep driver r_lpmv2

Table 11 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. More discussion on the Snooze configuration is provided in the next section.

As an example, a S5D9 MCU Standby module is shown in the following figure.

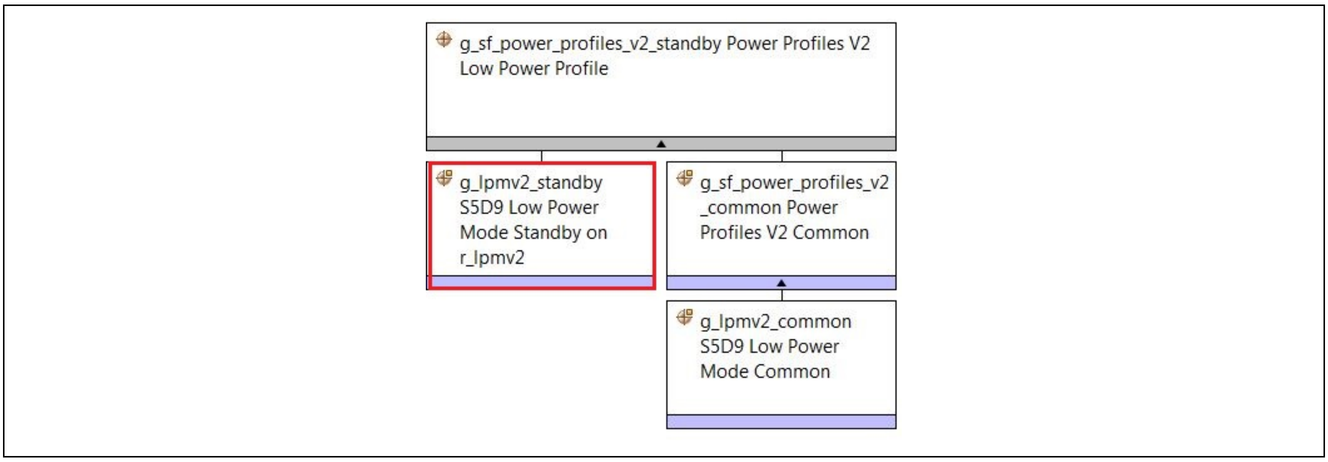


Figure 24 PPv2 Low Power Profile Standby driver r_lpmv2

Table 12 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
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, 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** as shown in the following figure.

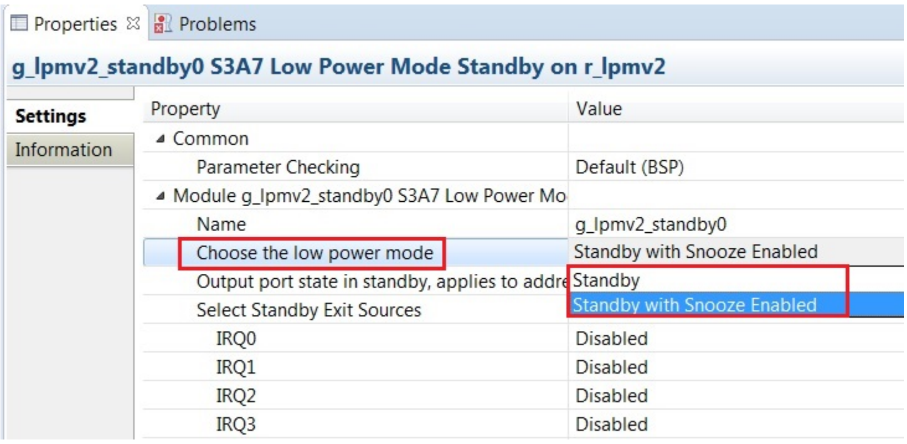


Figure 25 Create a Snooze mode in the Software Standby mode

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. The PPv2 Framework on the S5D9 is used as an example to show the possible configurations in the following figure.

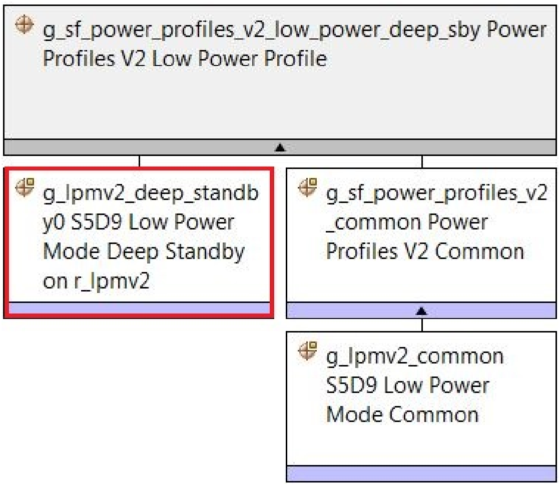


Figure 26 PPv2 Low Power Profile Deep Software Standby driver r_lpmv2

Table 13 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	Maintain the IO port states(default), Reset the IO port states	Output port state setting exit

ISDE Property	Value	Description
Standby mode		
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
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

ISDE Property	Value	Description
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*.

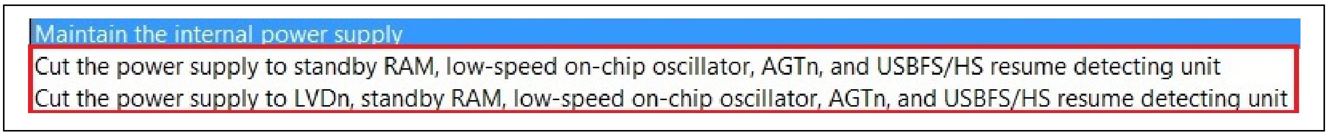


Figure 27 Predefined internal power supply options in PPv2 Framework Deep Software Standby mode

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

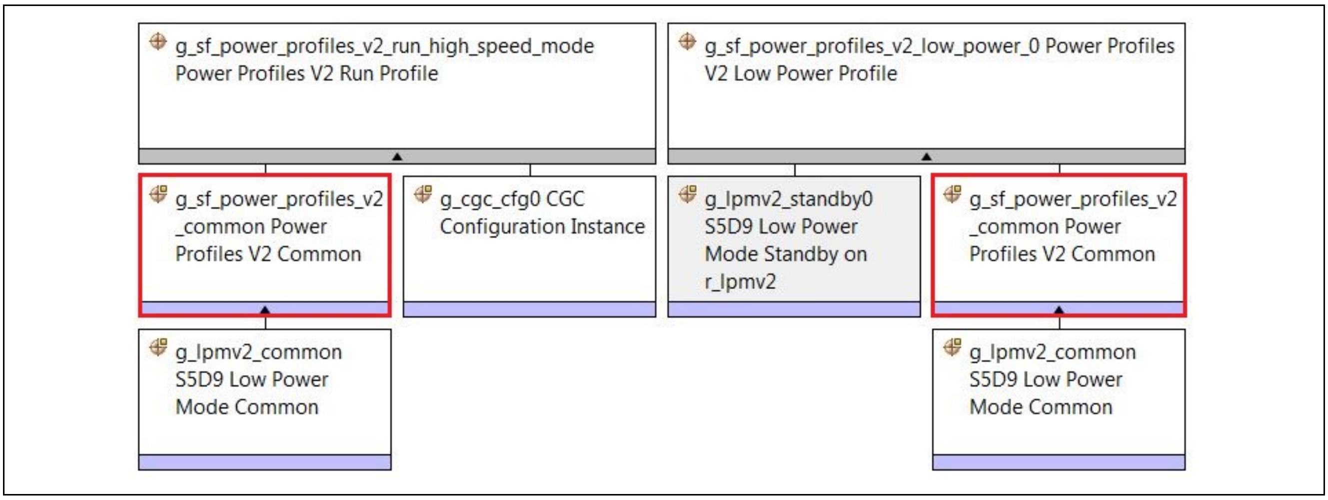


Figure 28 PPv2 Common module of PPv2 Run/Low Power Profile modules

The configuration settings are described as follows.

Table 14 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

- Added MCU Specific Low Power Mode Common Module

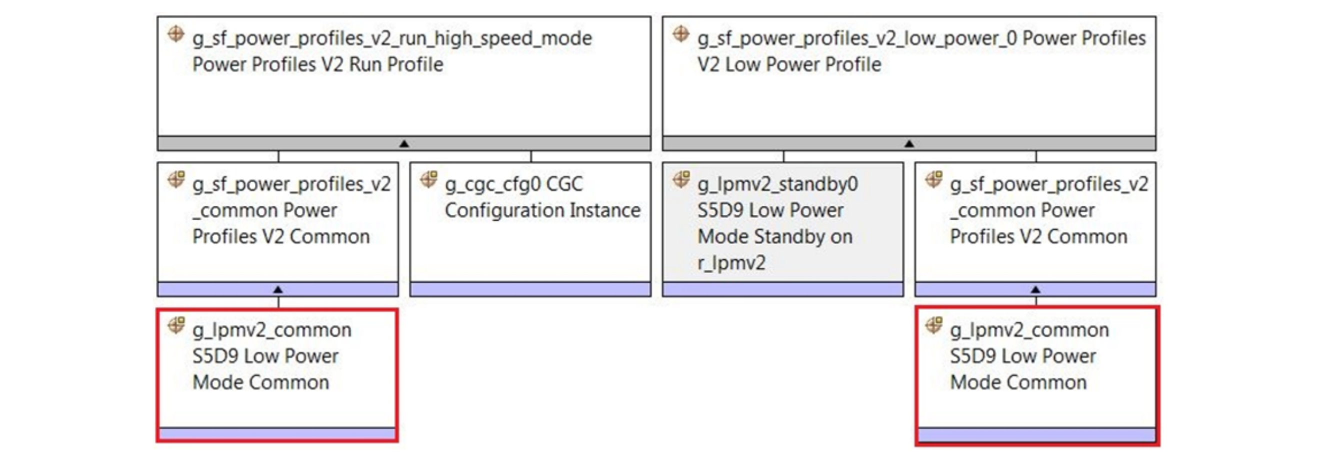


Figure 29 LPM Common module of PPv2 Run/Low Power Profile modules

The configuration settings are described as follows.

Table 15 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 S5D9 LPM module is already added in Figure 13 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 modes. However, only a simplified version is demonstrated here with two power control modes and two LPM modes.

6.1 Power Control Modes and LPM Mode supported by S5D9 MCU

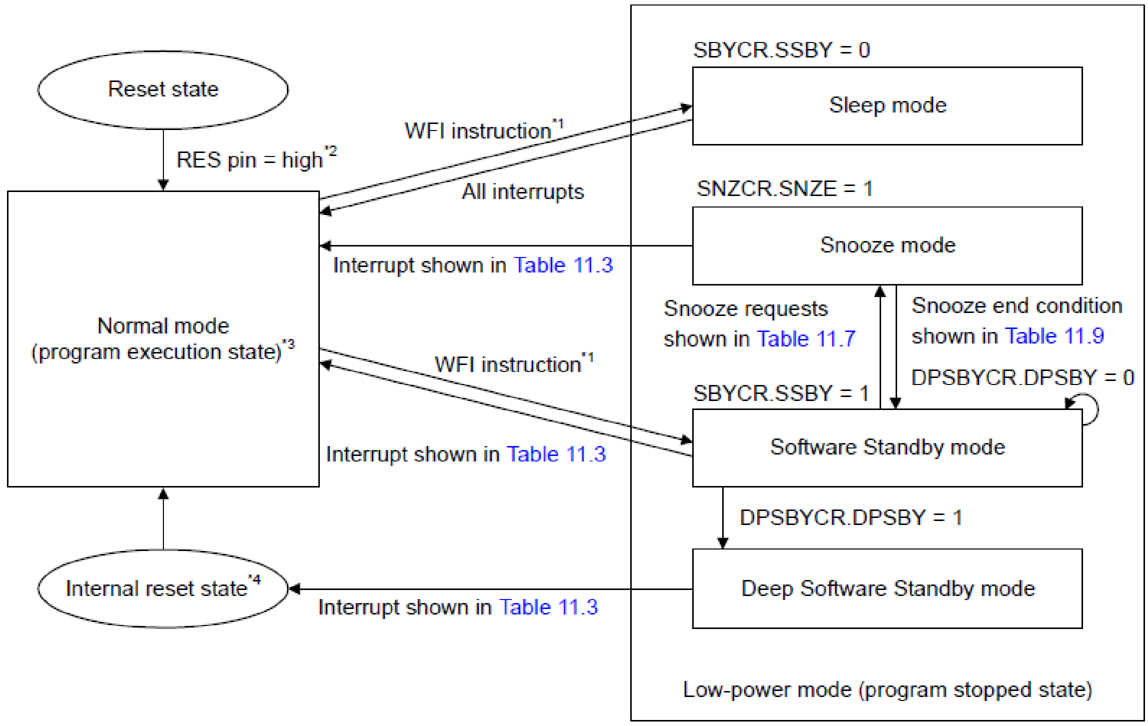
The Synergy S5D9 MCU Group has three operating power control modes:

- High-speed mode
- Low-speed mode
- Subosc-speed mode

The Synergy S5D9 MCU Group has four LPM modes:

- Sleep mode
- Software Standby mode
- Snooze mode
- Deep Software Standby mode

The mode transitions and their triggering conditions are illustrated in the following figure. For more details, see *Synergy S5D9 Microcontroller Group User's Manual*.



Note 1. When an interrupt that acts as a trigger for cancel is received during a transition to the program stopped state after the execution of a WFI instruction, the MCU executes interrupt exception handling instead of transitioning to low-power mode.

Note 2. The MOCO clock is the source of the operating clock following a transition from the reset state to Normal mode.

Note 3. The transition to Normal mode is made because of an interrupt from Sleep, Snooze, or Software Standby mode. The clock source is the same as before entering the low-power mode.

Note 4. When an available interrupt request is generated, an internal reset (Deep Software Standby reset) is generated over a fixed period. Canceling of Deep Software Standby mode accompanies release from the internal reset state, and then the MCU transitions to Normal mode and execute a reset exception processing with the MOCO clock as the source of the operating clock.

Figure 30 LPM mode transitions for the S5D9 Synergy MCU Group

6.2 Project Overview

PK-S5D9 v1.0 is selected as the development board for this project since it has user-controlled features such as two push buttons, and several LEDs to visualize the MCU state change.
The project operations can be summarized as follows:

- Press the S4 button on the PK-S5D9 v1.0 board to select a power control mode, then hold the S4 button to enter the selected power control mode. Currently, only High-Speed and Low-Speed modes are implemented, and their transitions are marked with the solid red lines.

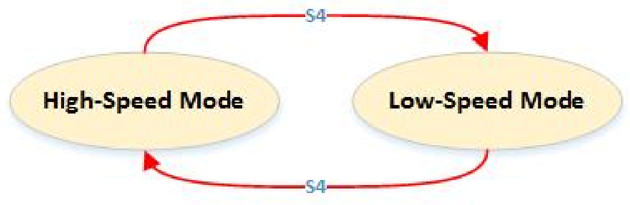


Figure 31 Press the button S4 for selecting a power control mode

- The LED1_GRN on the PK-S5D9 v1.0 board is used to represent the power control modes as follows:

Power Control Modes	LED1_GRN
High-speed Mode (default)	On
Low-speed Mode	Off

- Press the S5 button to select a LPM mode and the Normal mode, then hold the S5 to enter selected LPM mode. Currently only Standby mode is implemented, so its transition between the Normal mode is marked with a red solid line.

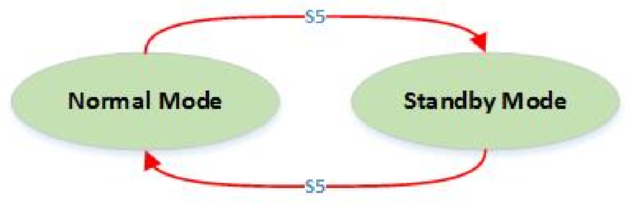


Figure 32 Press the button S5 for selecting a LPM mode

- Use the LED2_RED and LED3_YLW on the PK-S5D9 v1.0 to represent the power control modes as follows.

LPM Modes	LED2_RED	LED3_YLW
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.

A state diagram of these mode transitions is shown as follows.

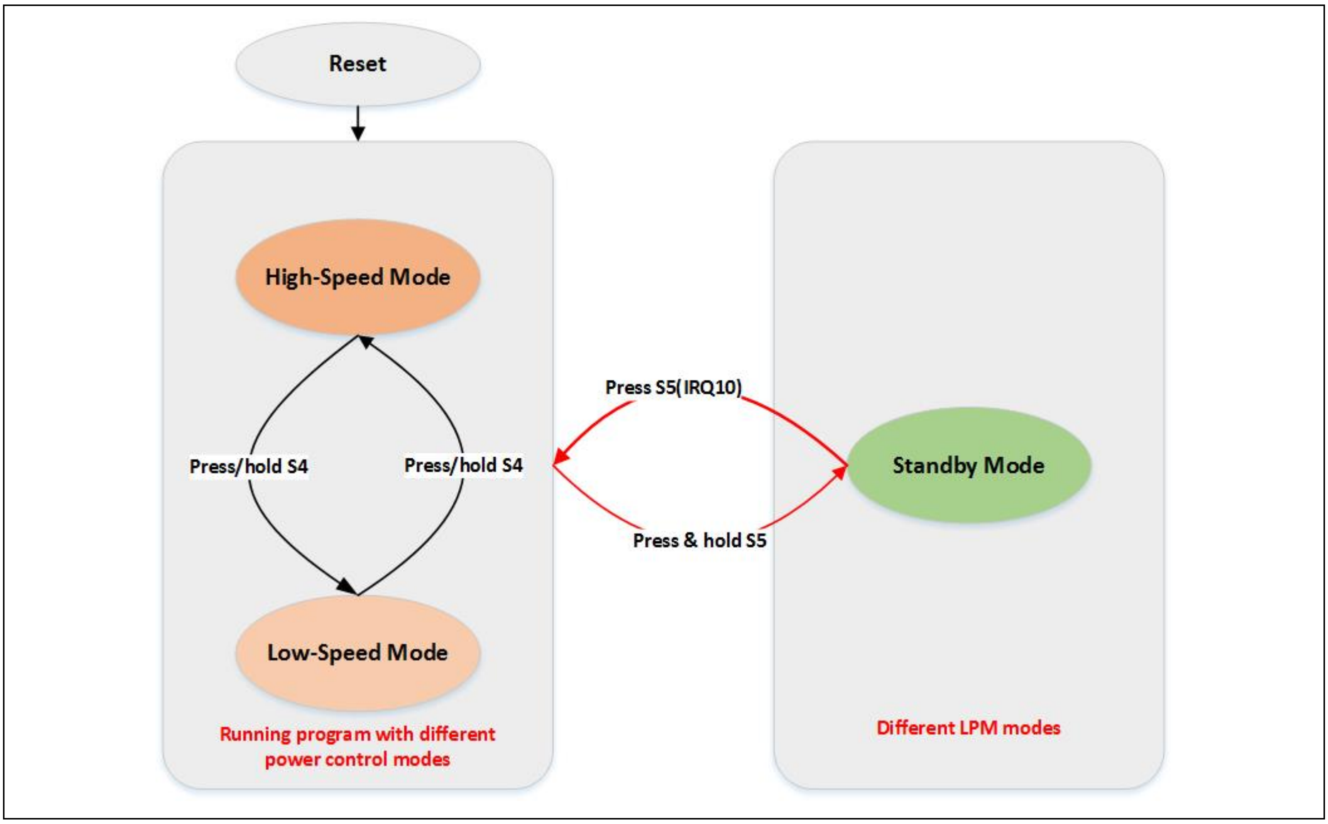


Figure 33 State transitions in the PPv2 project with PK-S5D9

6.3 Project Configuration

The project is implemented with one thread **PPv2 Thread**, and following modules:

- Two external IRQ drivers for SW4 and SW5:
  - `g_sw4_irq11`
  - `g_sw5_irq10`
- Two PPv2 Run profiles:
  - `g_sf_power_profiles_v2_run_high_speed_mode`
  - `g_sf_power_profiles_v2_run_high_speed_mode`
- One PPv2 Low Power Profile:
  - `g_sf_power_profiles_v2_standby`

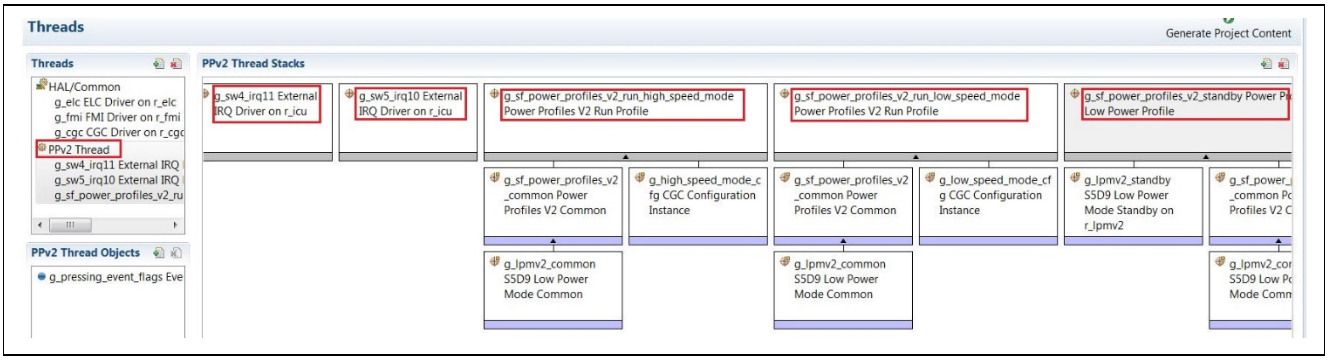


Figure 34 PPv2 Thread created in the PPv2 application project

6.3.1 Configuration of the Run profiles

Currently, two different power control modes `high_speed_mode` and `low_speed_mode` are added to the `ppv2_thread`. More **Run** profiles can be added to define more power control modes.

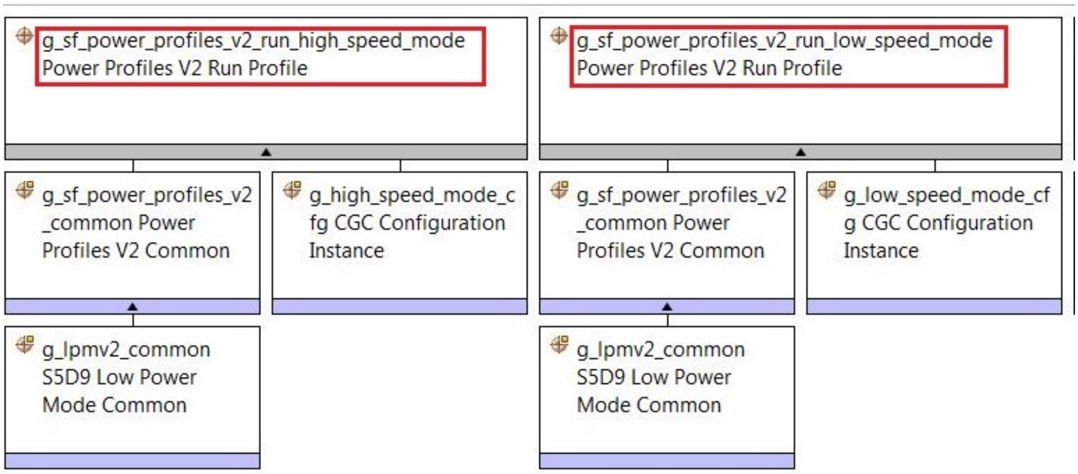


Figure 35 High-speed and Low-speed modes created in the project

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 240 MHz of PLL as a clock source to generate a 120 MHz of System Clock. Other clocks are generated with the following divisions:

Module g_high_speed_mode_cfg CGC Configur	
Name	g_high_speed_mode_cfg
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 Source Clock	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	2

Figure 36 CGC configuration of a 120 MHz High-speed Mode in the project

- The CGC configuration of the `g_sf_power_profiles_v2_run_low_speed_mode` uses a 32.768 kHz of LOCO as a clock source to generate a 32 kHz of the System Clock. Other clocks are generated as follows:

Module g_low_speed_mode_cfg CGC Configura	
Name	g_low_speed_mode_cfg
System Clock	LOCO
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	LOCO
PLL Divisor	1
PLL Multiplier	10.0
PCLKA Divisor	1
PCLKB Divisor	1
PCLKC Divisor	1
PCLKD Divisor	1
BCLK Divisor	1
FCLK Divisor	1
ICLK Divisor	1

Figure 37 CGC configuration of a 32 kHz Low-speed Mode in the project

6.3.2 Configuration of the Low Power Profile

Currently, only a Standby module is added to this thread. It can easily be extended to add more PPv2 Low Power modules.

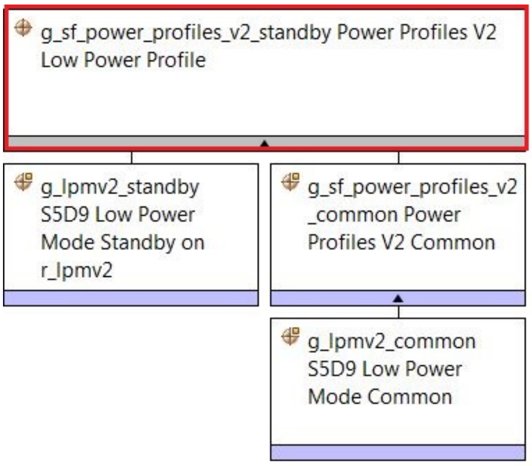


Figure 38 The LPM Standby profile module added in the PPv2 application project

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 39 Two pin configuration tables used in the PPv2 application project

The difference between these two pin configurations can be illustrated in the following figure.

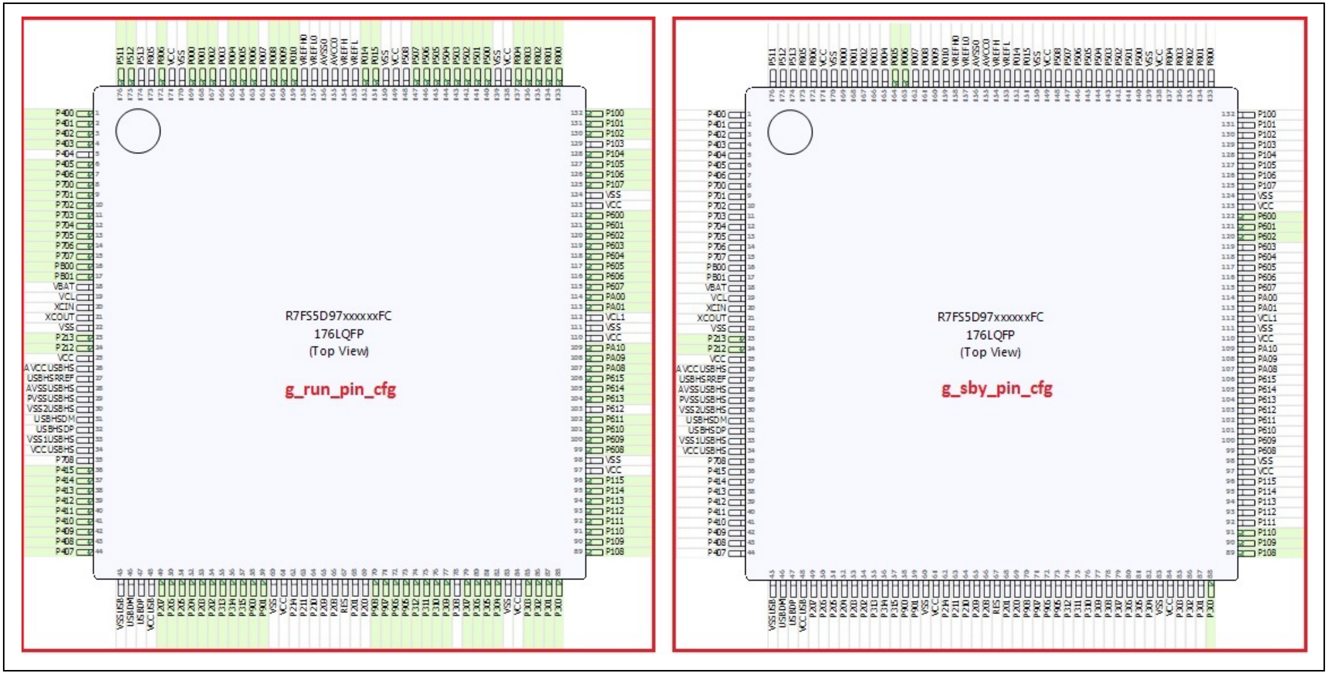


Figure 40 Differences between two pin configurations in the project

As seen in the preceding figure, the assigned pins are painted with green color, and `g_run_pin_cfg` allows more IO operations than `g_sby_pin_cfg`.

- The IRQ10 generated by pressing **S5** is **enabled** as a condition for transitioning the **Software Standby** mode to the **Normal** mode.

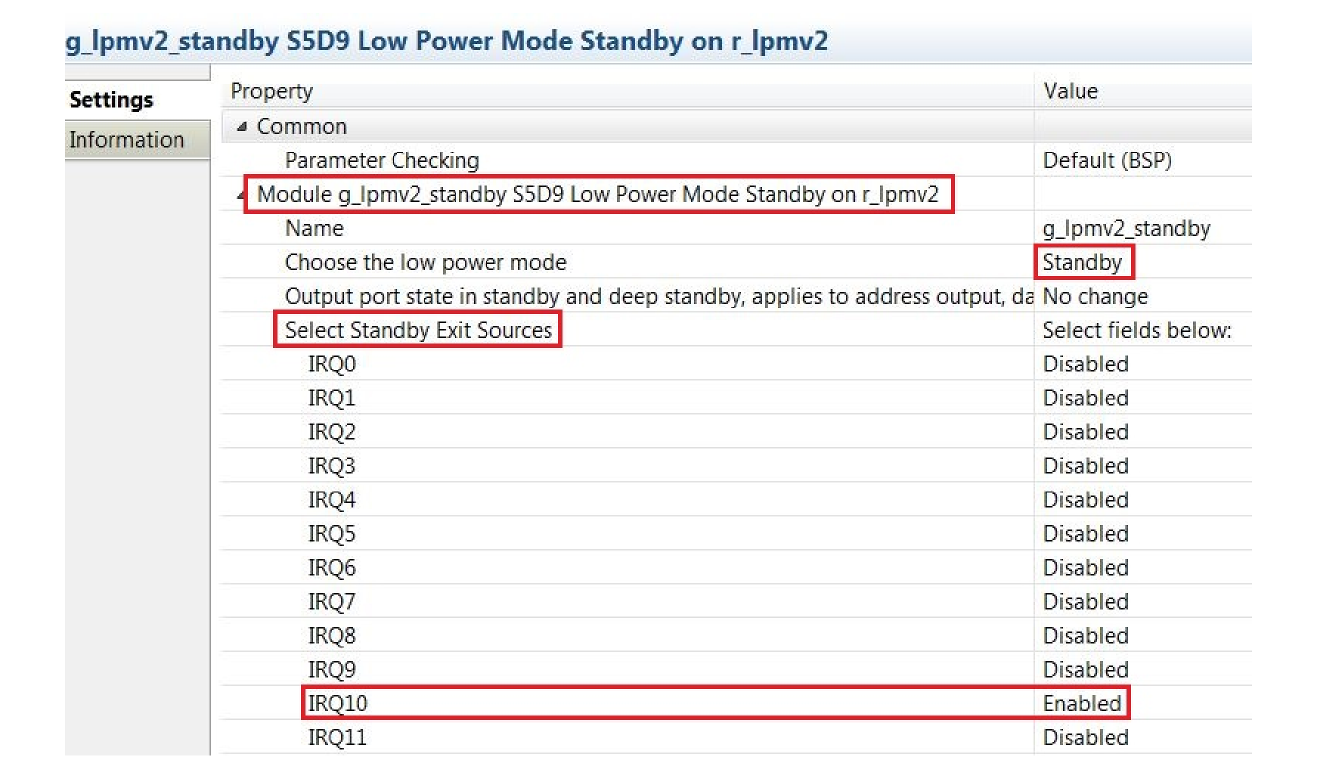


Figure 41 Configuration of the transition between the Standby and Normal modes

7. Running the PPv2 Framework Module Application Example

This section describes how to program and run the project on the PK-S5D9 Synergy MCU board.

7.1 Powering up the Board

- To connect to the board:
  - Connect the Micro USB end of the supplied USB cable to the PK-S5D9 v1.0 board J19 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 PK-S5D9 board.
  - Connect the other end of the USB cable to the USB port on your workstation.
- To program the board:
  - Refer to the *Synergy Project Import Guide* (r11an0023eu0119-synergy-ssp-import-guide.pdf) or instructions on importing the project into e^2 studio and building/running the project.

7.2 Verifying the Demo

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

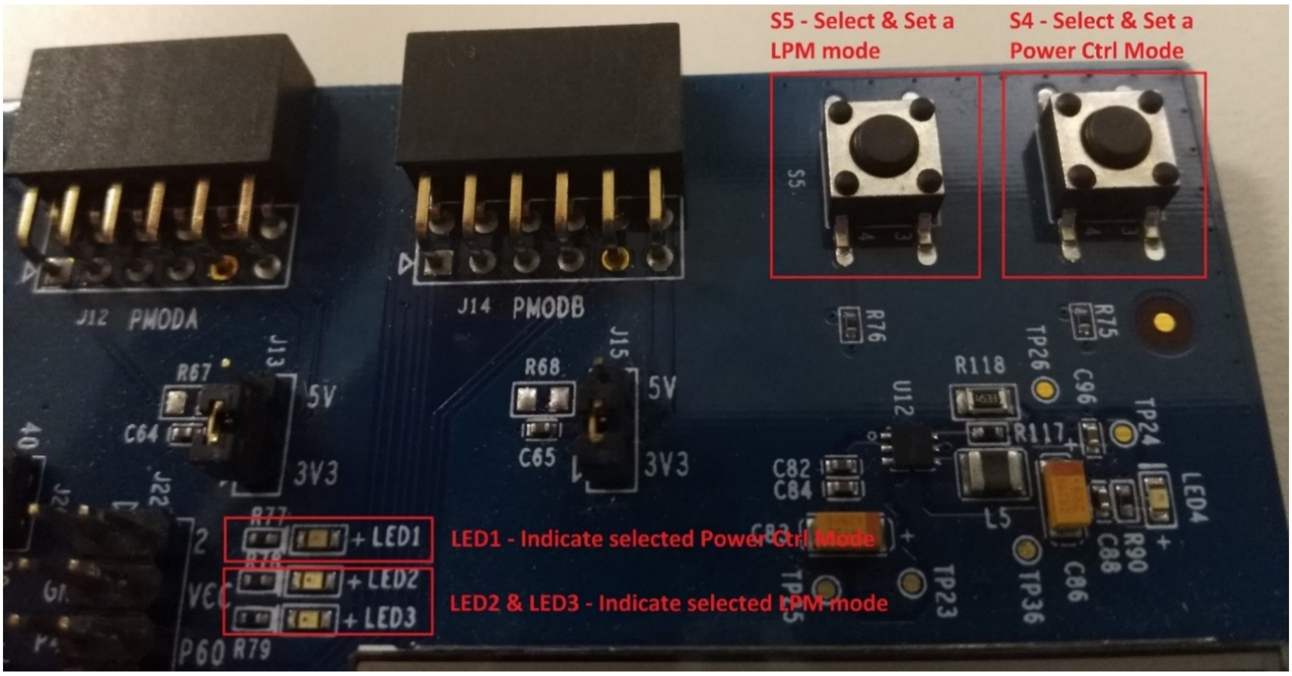


Figure 42 Demonstration setup on the PK-S5D9

As mentioned in the section 2.2.4, the debugger may block the MCU from entering the Standby/Deep Standby modes, so you need to reset the board by performing a full power-on reset cycle or by enabling Low Power Handling in the e^2 studio debugger. S4 and S5 buttons are used to enter the selected power control modes and LPM modes, that are also indicated with the LEDs:

- Press the **S4** button to select a power control mode, the High-speed mode (default) and the Low-speed mode.
- Press and hold the **S4** button to set the selected power control mode. The `LED1_GRN` blinks 4 times.
- Press the **S5** button to switch the LPM mode between default Normal mode and the Software Standby mode.
- Press and hold **S5** to set the selected LPM modes. All LEDs will go OFF after the `LED2_RED` blinks 4 times.
- Press the **S5** to generate IRQ10 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

- You 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.

  Note: Specification of a transition condition between Snooze and Normal mode is not fully supported with the currently released PPv2 Framework, so some control register manipulation is required.

- You can also measure power consumption of different modes on the J31.

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

9. References

- IAR Embedded Workbench® for Renesas Synergy™, *Getting Started with IAR Embedded Workbench for Renesas Synergy*.
- *Synergy Software Project (SSP) Import Guide*
- *S3A7 User's Manual: Microcontrollers*
- Renesas Synergy™ Software Package, *SSP User's Manual*.
- Keil. (2016). CMSIS-Pack (v 1.4.1). Retrieved from Usage and Description: http://www.keil.com/pack/doc/CMSIS/Pack/html/pages.html
- USBX CDC ACM (Host)
- USBX Device for CDCACM
- Renesas Synergy, e^2 studio ISDE 5.4.0.023
- 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

Rev.	Date	Description	
		Page	Summary
1.00	Nov 6, 2017	-	Initial release
1.01	Jan 18, 2018	-	Updated for SSP v1.3.3
1.02	Mar 6, 2018	-	Updated for SSP v1.4.0
1.03	Mar 7, 2018	-	Added PK-S5D9 to title

SALES OFFICES Renesas Electronics Corporation 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