Renesas Synergy™ Platform

I/O Port HAL Module Guide

R11AN0113EU0100
Rev.1.00
Aug 1, 2017

Introduction

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

The I/O Port HAL module is a high-level API for controlling I/O pins and is implemented on `r_ioport`. The I/O Port HAL module configures the board's pins and provides functions for manipulating them. The operating state of the I/O pins can be set via the Synergy configurator. When the Synergy project is built, a pin configuration file is created, and when the application runs, the BSP will configure the IO port accordingly, using the same APIs detailed in this document.

Contents

1. I/O Port HAL Module Features

This module configures one or more I/O pins. The direction of the pin or pins can be configured along with a number of other options provided as follows:

- Pull-up
- NMOS/PMOS
- Drive strength
- Event edge trigger (falling, rising or both)
- Whether the pin is to be used as an IRQ pin
- Whether the pin is to be used as an analog pin
- Whether the pin is to be used as a peripheral pin and which peripheral

The module also provides the following functionality:

- Changes the direction of one or more pins on a port
- Writes to one or more pins on a port
- Reads from one or more pins on a port
- Sets event output data
- Reads event input data

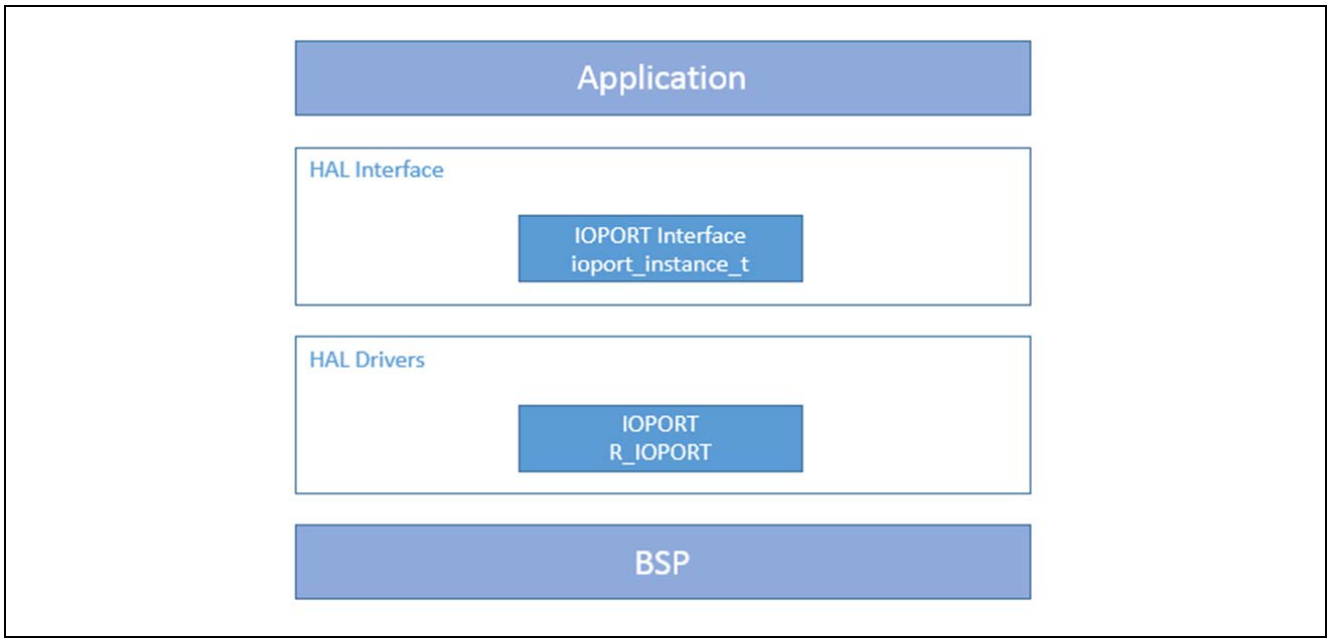


Figure 1 I/O Port HAL Module Block Diagram

2. I/O Port HAL Module APIs Overview

The I/O Port HAL module defines APIs for reading and writing from particular pins and ports. A complete list of the available APIs, an example API call, and a short description of each can be found in the following table. A table of return status values follows the API summary table.

Table 1 I/O Port HAL Module API Summary

Function Name	Example API Call and Description
.init	`g_ioport.p_api->init(g_ioport.p_cfg);` Initialize configuration of multiple pins.
.pinCfg	`g_ioport.p_api->pinCfg(IOPORT_PORT_00_PIN_00, IOPORT_CFG_IRQ_ENABLE | IOPORT_CFG_PORT_DIRECTION_INPUT);` Configure settings for an individual pin.
.pinDirectionSet	`g_ioport.p_api->pinDirectionSet(IOPORT_PORT_00_PIN_00, IOPORT_DIRECTION_INPUT);` Set the pin direction of a pin.
.pinEventInputRead	`g_ioport.p_api->pinEventInputRead(IOPORT_PORT_00_PIN_00, &pin_level);` Read the event (ELC) input data of the specified pin and return the level.
.pinEventOutputWrite	`g_ioport.p_api->pinEventOutputWrite(IOPORT_PORT_00_PIN_00, IOPORT_PIN_LEVEL_HIGH);` Write pin event (ELC) data.
.pinEthernetModeCfg	`g_ioport.p_api->pinEthernetModeCfg(IOPORT_ETHERNET_CHANNEL_0, IOPORT_ETHERNET_MODE_MII);` Configure the PHY mode of the Ethernet channels.
.pinRead	`g_ioport.p_api->pinRead(IOPORT_PORT_00_PIN_00, &pin_level);` Read level of a pin.
.pinWrite	`g_ioport.p_api->pinWrite(IOPORT_PORT_00_PIN_00, IOPORT_PIN_LEVEL_HIGH);` Write specified level to a pin.
.portDirectionSet	`g_ioport.p_api->portDirectionSet(IOPORT_PORT_00, direction_values, mask);` Set the direction of one or more pins on a port.
.portEventInputRead	`g_ioport.p_api->portEventInputRead(IOPORT_PORT_00, &pin_levels);` Read captured event (ELC) data for a port.
.portEventOutputWrite	`g_ioport.p_api->portEventOutputWrite(IOPORT_PORT_00, pin_levels, mask);` Write event (ELC) output data for a port.
.portRead	`g_ioport.p_api->portRead(IOPORT_PORT_00, &pin_levels);` Read states of pins on the specified port.
.portWrite	`g_ioport.p_api->portWrite(IOPORT_PORT_00, pin_levels, mask);` Write to multiple pins on a port.
.versionGet	`g_ioport.p_api->versionGet(&version);` Retrieve version information using the version pointer.

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

Table 2 Status Return Values

Name	Description
SSP_SUCCESS	API Call Successful.
SSP_ERR_INVALID_ARGUMENT	The port/pin/mask/direction/level (etc.) not valid.
SSP_ERR_ASSERTION	Unexpected null pointer.
SSP_ERR_UNSUPPORTED	Feature not supported; for instance, the Ethernet configuration is not supported on the device.

Note: Lower-level drivers may return common error codes. See *SSP User's Manual* **API References** for the associated module for a definition of all relevant status return values.

3. I/O Port HAL Module Operational Overview

The I/O Port HAL module is able to access the I/O ports of a device at the bit and port level. Both the port and the pin direction can be changed. Also, there are a number of configuration APIs provided to change the functionality of individual pins.

The I/O Port HAL module provides the following operations for configuring pins:

- Initializes the driver – performed by calling the `init` API:
  - Performs parameter checking and processes error conditions.
  - Handles VBATT domain pin configuration.
  - Writes PFS registers for pins.
- Configures pin – performed by calling the `pinCfg` API:
  - Performs parameter checking and processes error conditions (checks pin number pin, VBATT support).
  - Writes PFS register for the pin.
- Reads pin level – performed by calling the pinRead API:
  - Performs parameter checking and processes error conditions (checks pin number pin).
  - Reads PFS register for the pin.
- Reads all pin levels on a port – performed by calling the portRead API:
  - Performs parameter checking and processes error conditions (checks port number port).
  - Reads current value of PCNTR register value for the specified port.
- Writes pin level – performed by calling the pinWrite API:
  - Performs parameter checking and processes error conditions (check pin number pin and written level level).
  - Write to PFS register for the pin.
- Write multiple pin levels on a port – performed by calling the portWrite v:
  - Performs parameter checking and processes error conditions (checks port number port and pin mask mask).
  - Reads current configuration from the PCNTR register for the specified port.
  - Writes the pin levels to the PCNTR register for the specified port accordingly to the mask, preserving pin levels out of the scope.
- Sets the direction of multiple pins on a port – performed by calling portDirectionSet API:
  - Performs parameter checking and processes error conditions (checks port number port and pin mask mask).
  - Reads current configuration from the PCNTR register for the specified port.
  - Writes the pin levels to the PCNTR register for the specified port accordingly to the mask, preserving pin directions out of the scope.
- Writes pin direction – performed by calling the pinDirectionSet API:
  - Performs parameter checking and processes error conditions (checks pin number pin and written direction direction).
  - Writes to the PFS register for the pin.
- Reads event (ELC) port input – performed by calling the portEventInputRead API:
  - Performs parameter checking and processes error conditions (checks port number port).
  - Reads current value of the PCNTR register value for the specified port.
- Reads event (ELC) pin input – performed by calling the pinEventInputRead API:
  - Performs parameter checking and processes error conditions (checks pin number pin).
  - Reads current value of the PCNTR register value for the specified pin's port.
  - Gets the pin level by applying a pin mask to the PCNTR register value.
- Writes event (ELC) port output – performed by calling the portEventOutputWrite API:
  - Performs parameter checking and processes error conditions (checks port number port and pin mask mask_value).
  - Reads current configuration from the PCNTR register for the specified port.
  - Writes the pin levels to the PCNTR register for the specified port accordingly to the mask preserving pin levels out of the event scope.
- Writes event (ELC) pin output – performed by calling the pinEventOutputWrite API:
  - Performs parameter checking and processes error conditions (checks pin number pin and written level pin_value).
  - Writes the pin level to the PCNTR register for the specified pin's port accordingly to the mask that is preserving pin levels out of the event scope.

- Configures Ethernet channel PHY mode – performed by calling the ethernetModeCfg API:
  - Performs parameter checking and processes error conditions (checks Ethernet channel and mode).
  - Updates the Ethernet Control Register (PFENET).

3.1 I/O Port HAL Module Important Operational Notes and Limitations

3.1.1 I/O Port HAL Module Operational Notes

- A bit mask of 16 bits needs to be applied in order to read and write to a specific pin on a port; ports are numbered from 0 (LSB) to 15 (MSB).
- Ethernet configuration may not be supported on some devices.

3.1.2 I/O Port HAL Module Limitations

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

4. Including the I/O Port HAL Module in an Application

The e^2 studio Integrated Solution Developer Environment (ISDE) automatically adds the necessary I/O Port stack to the HAL/Common thread in the project and the stack is ready by default to operate. If the I/O Port Driver was inadvertently removed, the following process is available to add the driver to the HAL/Common Thread.

Note: It is assumed you are familiar with creating a project, adding threads, adding a stack to a thread,and configuring a block within the stack. If you are unfamiliar with any of these items, refer to the first few chapters of the *SSP User's Manual* to learn how to manage each of these important steps in creating SSP-based applications.

To add the I/O Port HAL Driver to your application, simply add it to a project thread using the stacks selection sequence given in the following table. (The default name for the I/O Port is g_ioport.)

Table 3 I/O Port HAL Module Selection Sequence

Resource	ISDE Tab	Stacks Selection Sequence
g_ioport I/O Port driver on r_ioport	Threads	Highlight HAL/Common and select New > Driver > System > I/O Port Driver on r_ioport

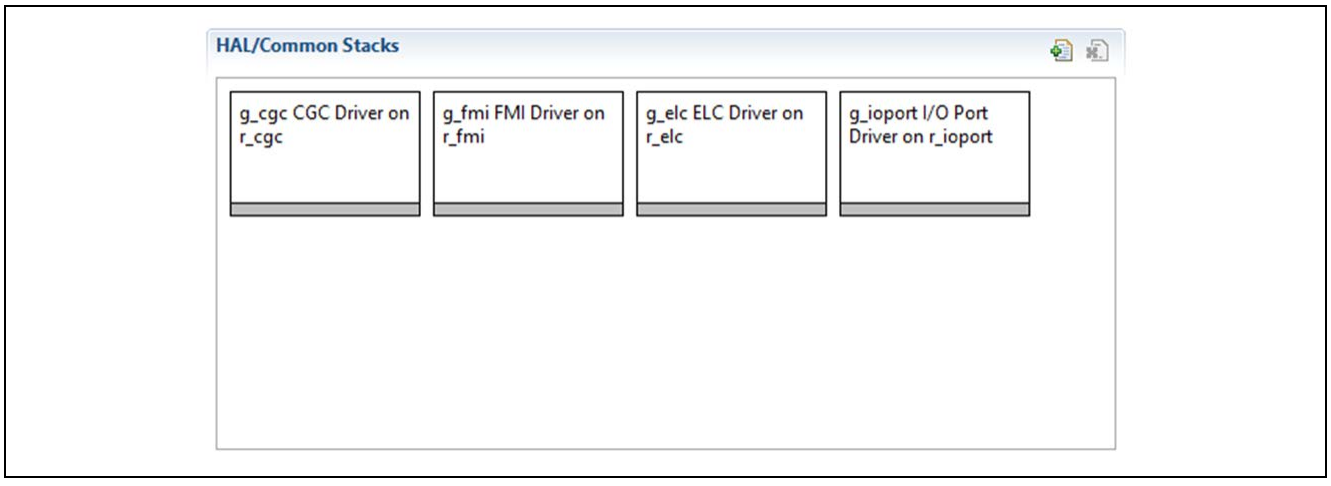


Figure 2 I/O Port HAL Module Stack

No additional lower-level modules need to be added or configured.

5. Configuring the I/O Port HAL Module

The module does not need any additional configuration. All the pin configuration is generated in the src/synergy_gen/pin_data.c file according to the BSP.

6. Using the I/O Port HAL Module in an Application

The typical steps in using the I/O Port HAL module in an application are:

1. Initialize the driver using the `init` API.

2. Configure the pins using the `pinCfg` API.
3. Read from specified pins and ports using the `pinRead` or `portRead` API.
4. Write to specified pins and ports using the `pinWrite` or `portWrite` API.

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

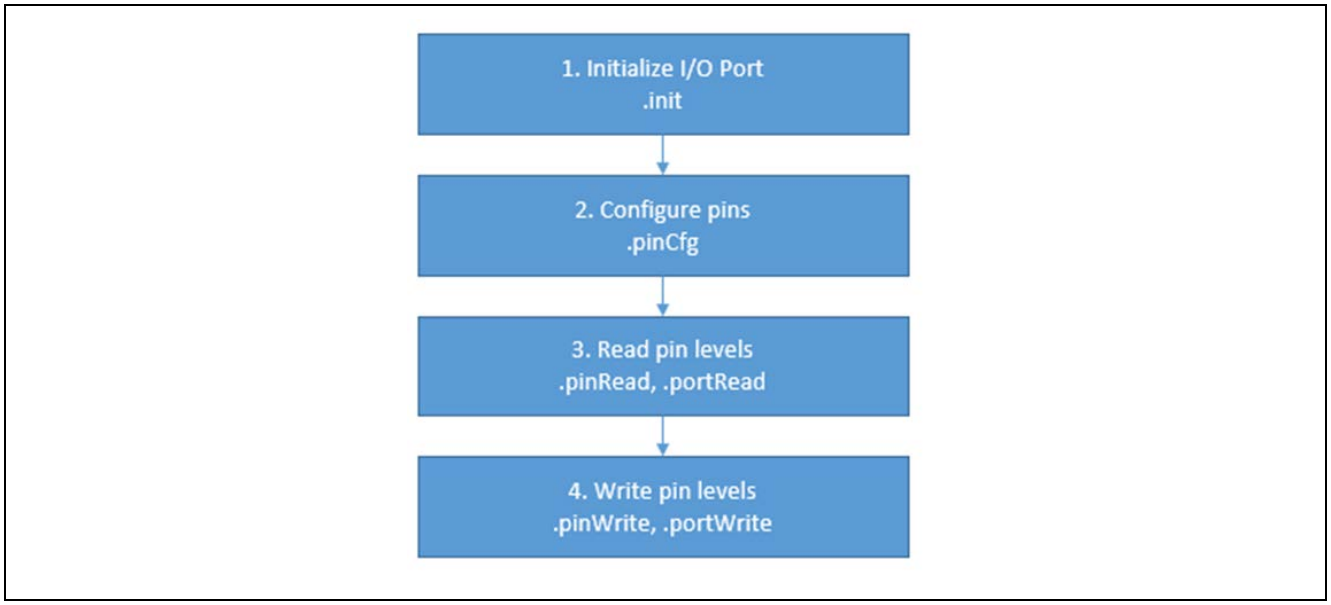


Figure 3 Flow Diagram of a Typical I/O Port HAL Module Application

7. The I/O Port HAL Module Application Project

The application project associated with this module guide demonstrates the aforementioned steps in a full design. You may want to import and open the application project within the ISDE and view the configuration settings for the I/O Port HAL module. You can also read over the code (in `hal_entry.c`) which is used to illustrate the I/O Port module APIs in a complete design.

The application project demonstrates the typical use of the I/O Port APIs. The application project main thread entry initializes the I/O Port HAL Framework, periodically reads S4 and S5 user buttons' pins and writes to LED1, LED2 and LED3 pins. The following table identifies the target versions for the associated software and hardware used by the application project:

Table 4 Software and Hardware Resources Used by the Application Project

Resource	Revision	Description
e² studio	5.3.1 or later	Integrated Solution Development Environment
SSP	1.2.0 or later	Synergy Software Platform
IAR EW for Synergy	7.71.2 or later	IAR Embedded Workbench® for Renesas Synergy™
SSC	5.3.1 or later	Synergy Standalone Configurator
SK-S7G2	v3.0 to v3.1 or later	Starter Kit

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

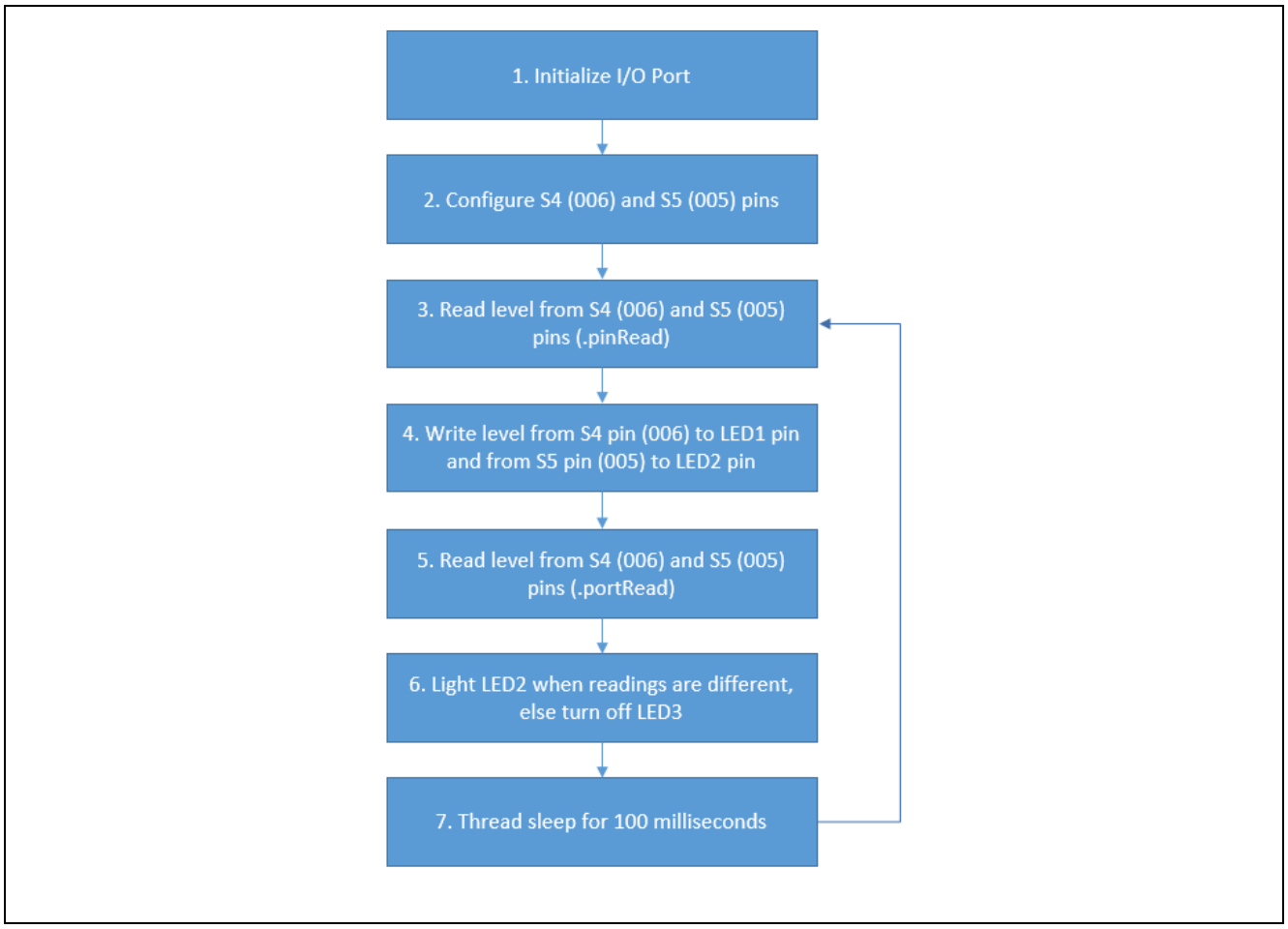


Figure 4 I/O Port HAL Module Application Project Flow Diagram

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

The code file contains the API calls for pin configuration, reading (individual pins and whole ports) and writing pin levels.

LEDs (1 and 2) are illuminated based on "S" user buttons. S4 controls LED1 and S5 controls LED2 in such a way that pressing a button lights a corresponding LED. These pins are illuminated based on the output of the `pinRead` function. Additionally, S4 and S5 pin levels are using the `portRead()` functions. The output of this function illuminates LED3. If the values of S4 & S5 match, then LED3 is illuminated; if there is a mismatch, LED3 remains off. The operation is repeated every 100ms.

8. Customizing the I/O Port HAL Module for a Target Application

A target application has I/O Port requirements specific to the application. The Synergy configurator allows the I/O ports of the target device to be configured via a graphical interface. The board support package (BSP) configures the I/O ports as specified as the target application starts. The I/O Port HAL module provides all the functionality to customize the I/O ports during run time, if required.

9. Running the I/O Port HAL Module Application Project

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

To implement the I/O Port HAL application in a new project, follow the steps for defining, configuring, auto-generating files, adding code, compiling and debugging on the target kit. Following these steps is a hands-on approach that can help make the development process with SSP more practical, while just reading over this guide will tend to be more theoretical.

Note: The following steps are described in sufficient detail for someone experienced with the basic flow through the Synergy development process. If these steps are not familiar, refer to the first few chapters of the SSP User's Manual for a description of how to accomplish these steps.

Use the following steps to create and run the I/O Port HAL module application project:

1. Create a new Renesas Synergy project for the SK-S7G2 board (S7G2-BSP) called "IO_PORT_HAL_MG_AP."
2. Select the S7G2-SK with no RTOS project template and create project.
3. Open Configuration.xml from the generated project and select the **Threads** tab.
4. Configure/check the I/O Port driver instance name.
5. Click on the **Generate Project Content** button.
6. Add the code from the supplied project files `ioport_hal.c` & "hal_entry.c".
7. Connect to the host PC via a micro USB cable to J19 on SK-S7G2.
8. Start to debug the application.
9. Press and release buttons S4 and S5 and watch the LEDs being lit.

10. I/O Port HAL Module Conclusion

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

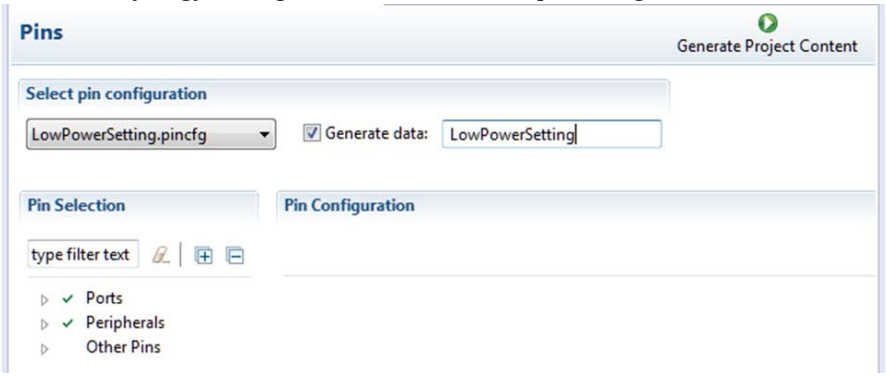
11. I/O Port HAL Module Next Steps

The Synergy Configurator sets pins automatically by the BSP initialization process. The configurator allows for multiple pin configurations to be set and these configurations can be set by the following API:

```
g_ioport.p_api->pinsCfg( & <My Pin Configuration Name> );
```

The pin configuration feature can be useful if a large number of pin states need to be changed. For example, when switching in and out of a power-saving mode. The following steps detail how to create an additional pin configuration and make calls:

1. In the root directory of your project, there is a `.pincfg` file (for example: S7G2-SK.pincfg).
   Make a copy of the `pincfg` file in the route directory and give it a new name, such as LowPowerSetting.pincfg).
2. In the Pins tab of the Synergy Configuration tool, select this pin configuration.



3. Click the **Generate** data box and specify a symbolic name.
4. Make the new I/O port settings specific to this profile, and then click **Generate Project Content**.
5. If you open the source file `\src\synergy_gen\pin_data.c`, you see that the new pin configuration structure has been created.
6. To set this new pin configuration, use the API call with the symbolic name given in step 3: `g_ioport.p_api->pinsCfg(&LowPowerSetting);`
   It is likely you will need to include the header file `include "bsp_pin_cfg.h"` in the file you use for the API, so that the compiler can resolve the structure name.

12. I/O Port HAL Module Reference Information

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

Links to all the most up-to-date r_ioport module reference materials and resources are available on the Synergy Knowledge Base: https://en-us.knowledgebase.renesas.com/English_Content/Renesas_Synergy%E2%84%A2_Platform/Renesas_Synergy_Knowledge_Base/r_ioport_Module_Guide_Resources.

Website and Support

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

Technical Contact Details:

- America: https://renesas.zendesk.com/anonymous_requests/new
- 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	Aug 1, 2017	—	Initial Release

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 disputes 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, drawing, chart, program, algorithm, 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 otherwise misappropriate 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, copy or otherwise misappropriation of Renesas Electronics products.
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; and 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.

   Renesas Electronics products are neither intended nor 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 damages (space and 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 third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. When using the 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 radiation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions or failure or accident arising out of the use of Renesas Electronics products beyond 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. Further, Renesas Electronics products are not subject to radiation resistance design. Please ensure to implement safety measures to guard them against the possibility of bodily injury, injury or damage caused by fire, and social damage in the event of 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 by your own responsibility as warranty for your products/system. Because the evaluation of microcomputer software alone is very difficult and not practical, please evaluate 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. Please investigate applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive carefully and sufficiently and use 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 not use Renesas Electronics products or technologies for (1) any purpose relating to the development, design, manufacture, use, stockpiling, etc., of weapons of mass destruction, such as nuclear weapons, chemical weapons, or biological weapons, or missiles (including unmanned aerial vehicles (UAVs)) for delivering such weapons, (2) any purpose relating to the development, design, manufacture, or use of conventional weapons, or (3) any other purpose of disturbing international peace and security, and you shall not sell, export, lease, transfer, or release Renesas Electronics products or technologies to any third party whether directly or indirectly with knowledge or reason to know that the third party or any other party will engage in the activities described above. When exporting, selling, transferring, etc., Renesas Electronics products or technologies, you shall comply with any applicable export control laws and regulations promulgated and administered by the governments of the countries asserting jurisdiction over the parties or transactions.
10. Please acknowledge and agree that you shall bear all the losses and damages which are incurred from the misuse or violation of the terms and conditions described in this document, including this notice, and hold Renesas Electronics harmless, if such misuse or violation results from your resale or making Renesas Electronics products available any third party.
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 majority-owned subsidiaries.

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

(Rev.3.0-1 November 2016)

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.
2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

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: +44-1628-585-100, Fax: +44-1628-585-900

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, P. R. China 200333
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 II Stage, Indiranagar, Bangalore, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 6.0