Renesas Synergy™ Platform

FMI HAL Module Guide

R11AN0129EU0101
Rev.1.01
Aug 23, 2017

Introduction

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

The FMI HAL module is a high-level API for applications that read records from the Factory MCU Information Flash Table. The FMI HAL module is implemented on r_fmi. The FMI HAL module uses the Flash Interface on the Synergy MCU.

Contents

1. FMI HAL Module Features

The FMI HAL module reads the FMIFRT (Factory MCU Information Flash Root Table) on a Synergy microcontroller, looking up the address of the start of the table in flash. The module sets the caller's pointer to the product information record from the table. This information may be used to determine the capabilities of features specific to this MCU package. Information available from the FMI HAL module includes:

- Product information: product name, package, pin count and temperature range
- Product features: version major, version minor and variant data
- Event information such as interrupts and events

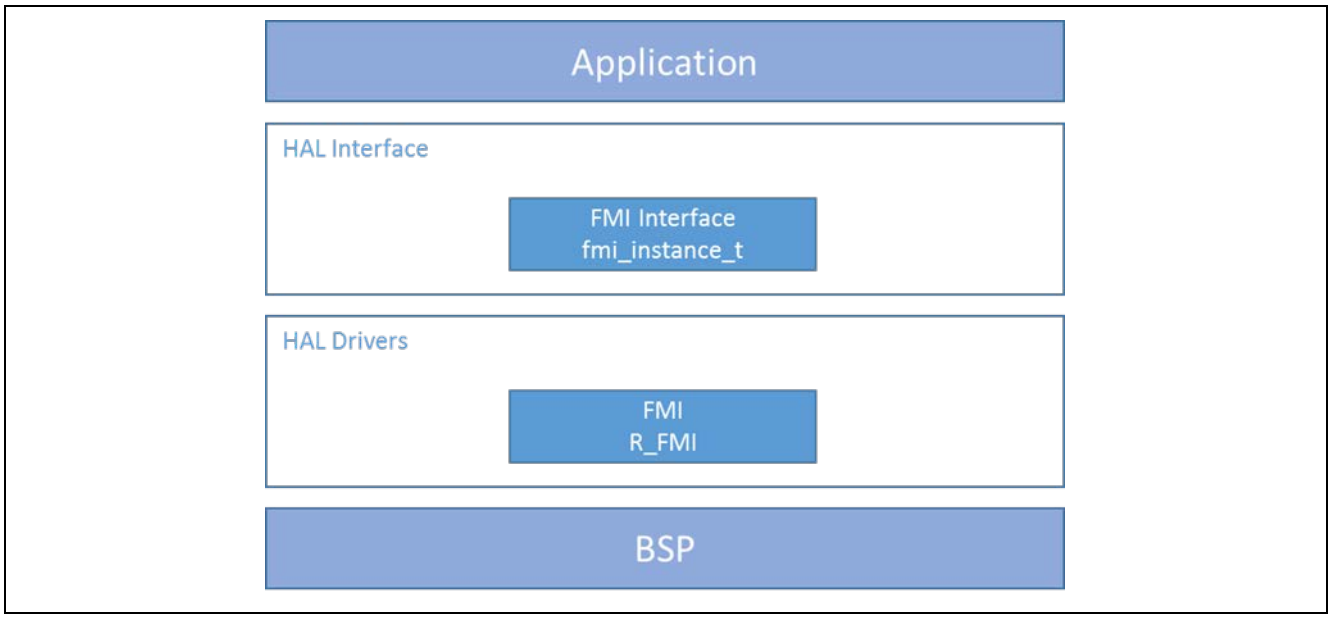

Figure 1 FMI HAL Module Block Diagram

2. FMI HAL Module APIs Overview

The FMI HAL module defines an API for accessing the FMIFRT. The following table has a complete list of the available APIs, an API call example and a brief description of each API. A table of status return values follows the API summary table.

Table 1 FMI HAL Module API Summary

Function Name	Example API Call and Description
.init	`g_fmi.p_api->init();` Initialize the FMI base pointer.
.productInfoGet	`g_fmi.p_api->productInfoGet(&g_pp_product_info);` Get product information record address into g_pp_product_info pointer.
.uniqueIdGet	`g_fmi.p_api->uniqueIdGet(&g_p_unique_id);` Copy the unique ID into the g_p_unique_id pointer.
.productFeatureGet	`g_fmi.p_api->productFeatureGet(&g_ssp_feature, &g_feature_info);` Get feature information and store it in g_feature_info pointer.
.eventInfoGet	`g_fmi.p_api->peventInfoGet(&g_ssp_feature,SSP_SIGNAL_GPT_COUNTER_OVERFLOW, &g_event_info);` Get event information and store it in g_event_info pointer.

Function Name	Example API Call and Description
.versionGet	`g_fmi.p_api->versionGet(&g_p_version);` Get the driver version based on compile time macros.

Note: Review the SSP User's Manual API References for the associated module to learn more about operations and definitions for the function data structures, typedefs, defines, API data, API structures and function variables.

Table 2 Status Return Values

Name	Description
SSP_SUCCESS	API Call Successful.
SSP_ERR_INVALID_FMI_DATA	The FMI data table provided is not valid
SSP_ERR_IP_CHANNEL_NOT_PRESENT	Requested channel does not exist on this MCU
SSP_ERR_IP_UNIT_NOT_PRESENT	Requested unit does not exist on this MCU
SSP_ERR_INTERNAL	Requested feature is in a format not supported at this time
SSP_ERR_IRQ_BSP_DISABLED	Event information could not be found
SSP_ERR_ASSERTION	Caller's pointer is null
SSP_ERR_INVALID_FACTORY_FLASH	Factory flash is not valid

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

3. FMI HAL Module Operational Overview

The FMI HAL module retrieves the product information record address and populates the `fmi_product_info_t` structure using the `productInfoGet` API.

The FMI HAL module copies unique ID and populates the `fmi_unique_id_t` structure using the `uniqueIdGet` API.

The FMI HAL module gets feature information and populates the `fmi_feature_info_t` structure using the `productFeatureGet` API.

The FMI HAL module fetches event information and populates the `fmi_event_info_t` structure using the `eventInfoGet` API.

The FMI HAL module gets code version and API version in `ssp_version_t` structure using the `versionGet` API.

3.1 FMI HAL Module Important Operational Notes and Limitations

3.1.1 FMI HAL Module Operational Notes

The `fmi_product_info_t::unique_id` is deprecated. It does not contain a unique ID if the factory MCU information is linked in by the application code. Use `uniqueIdGet` for the unique ID.

3.1.2 FMI HAL Module Limitations

- For limitations of the FMI HAL Interface and its implementation, see the latest SSP release notes.
- The FMI Driver has been tested on the S7G2 Synergy microcontroller family using the FMIFRT peripheral register. It is the only Synergy MCU that is currently programmed with data in the Factory MCU Information Table.
- For the S5D9 MCU, the `uniqueIdGet` API of the r_fmi module returns an error. There is no workaround.

4. Including the FMI HAL Module in an Application

This section describes how to include the FMI HAL module in an application using the SSP configurator.

Note: This process assumes that you are familiar with creating a project, adding threads, adding a stack to a thread and configuring a block within the stack.

To add the FMI to an application, simply add it to a thread using the stacks selection sequence given in the following table. (The default name for the FMI is r_fmi0. This name can be changed in the associated Properties window.)

Table 3 Flash Driver Selection Sequence

Resource	ISDE Tab	Stacks Selection Sequence
r_fmi FMI Driver on r_fmi	**Threads > HAL/Common**	**New Stack> Driver> System> FMI Driver on r_fmi**

The following figure shows that the FMI Driver on r_fmi is automatically added to the HAL/Common Stack when a new project is created, since every design requires the message to be supplied by the FMI HAL module.

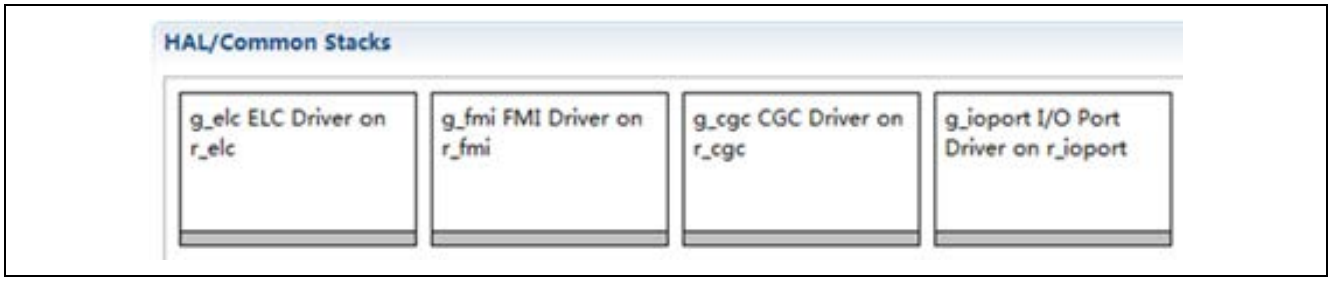

Figure 2 FMI HAL Module Stack

5. Configuring the FMI HAL Module

The FMI HAL module must be configured by the user for the desired operation. The available configuration settings and defaults for all the user-accessible properties are given in the Properties tab within the SSP configurator and are shown in the following tables for easy reference.

Note: You may want to open your ISDE, create the module and explore the property settings in parallel with looking over the Configuration Table Settings given below. This can help orient you and provide a useful hands-on approach to learning the ins and outs of developing with SSP.

Table 4 Configuration Settings for the FMI HAL Module on r_fmi

ISDE Property	Value	Description
Parameter Checking	BSP, Enabled, Disabled (Default: BSP)	Controls whether to include code for API parameter checking.
SSP MCU Information Symbol Name	g_fmi_data (Default)	This symbol maps to the base address where the factory flash table information is found. It should not be modified.
Part Number Mask	0xFE00 (Default)	Each bit represents one character in the Synergy part number, where the MSB is the first character in the part number ('R'). Set bits to ensure the part number in the MCU factory flash matches the part number in the SSP MCU Information. The default mask checks everything except operating temperature, software ID, and quality ID.
Name	g_fmi (Default)	Module instance name.

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

5.1 FMI HAL Module Clock Configuration

No specific clock configurations are required for the FMI HAL Driver.

5.2 FMI HAL Module Pin Configuration

No specific pin configurations are required for the FMI HAL Driver.

6. Using the FMI HAL Module in an Application

The typical steps in using the FMI in an application are:

1. Initialize the FMI using the `init` API, it is automatically initialized after Reset.
2. Use the `productInfoGet` API to get product information.
3. Use the `uniqueIdGet` API to get unique ID.
4. Use the `productFeatureGet` API to get feature information.
5. Use the `eventInfoGet` API to get event information.
6. Use the `versionGet` API to get driver version information.

The following figure illustrates these common steps in a typical operational flow diagram:

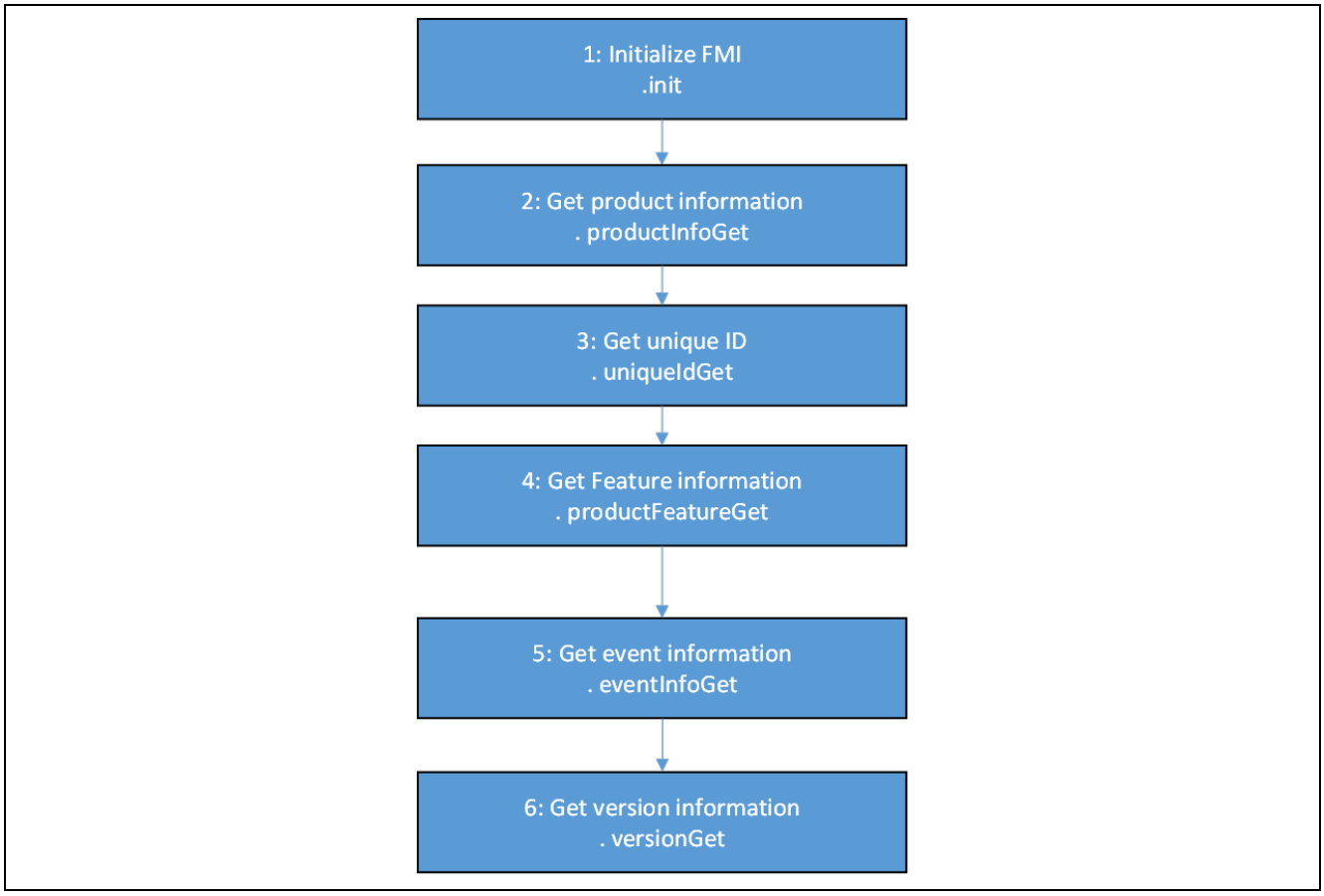

Figure 3 Flow Diagram of a Typical FMI HAL Module Application

7. The FMI HAL Module Application Project

The application project associated with this module guide demonstrates the aforementioned steps to using the FMI HAL module in a full design. The project can be found using the link provided in the References section at the end of this document. You may want to import and open the application project within the ISDE and view the configuration settings for the FMI HAL module. You can also read over the code (in <fmi_hal_mg.c>) which illustrates the FMI APIs in a complete design.

The application project demonstrates the typical use of the FMI APIs. The application project initializes the FMI module before main thread entry. From the `fmi_hal_mg.c` file for this application, the product information, unique ID, product feature information, event information and driver version are read and output to debug the virtual console.

Table 5 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 Renesas 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	Starter Kit

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

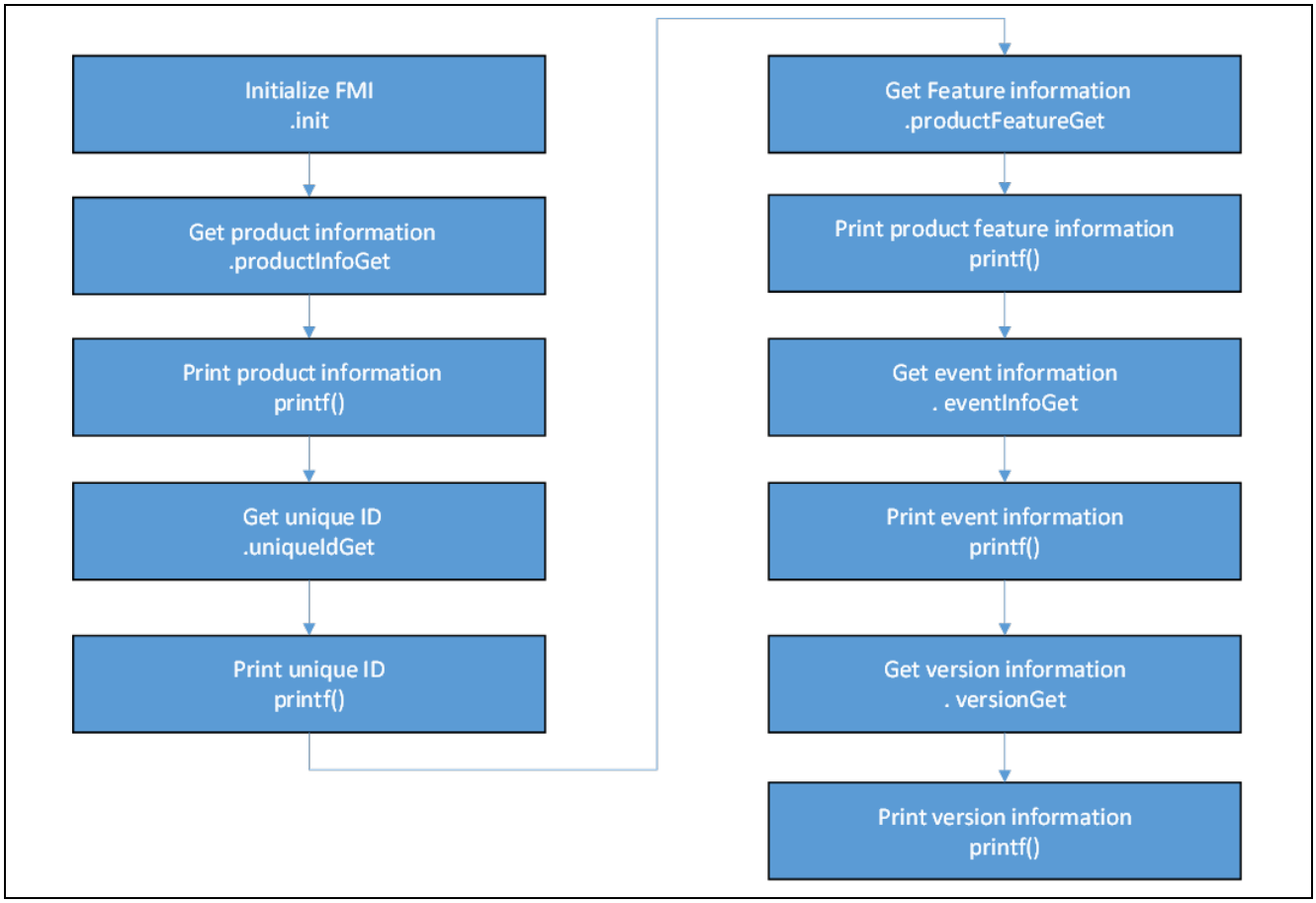

Figure 4 FMI HAL Module Application Project Flow Diagram

The complete application project can be found using the link provided in the FMI HAL Module Reference Information section. The `fmi_hal_mg.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 the FMI APIs.

In the c file of the system, FMI is first initialized by the `g_fmi_on_fmi.init` API. There is no need in `fmi_hal_mg.c` to initialize the FMI again.

Calling `productInfoGet` API retrieves the product information for output with the printf() routine. The example product information retrieves the MCU name, product marking, pin number, max frequency, and so on. Different messages can be output to meet project requirements.

Calling the `uniqueIdGet` API retrieves the unique 4-byte product ID for output with the printf() routine.

Before calling `productFeatureGet` API, specify the particular feature ID, the channel number and the unit number. In the sample program, the unit 0 and channel 0 are specified. Calling `productFeatureGet` API retrieves the channel count and the register start address of channel 0 and unit 0 for output with the printf() routine.

Before calling the `eventInfoGet` API, specify a particular feature ID and enable the interrupt event. Calling `eventInfoGet` API retrieves the event number for output with the printf() routine. Two product modules are set in

the FMI application project. First, the GPT channel 1 is set and the GPT1interrupt is enabled. Second, a 12 bit A/D converter (ADC12) unit0 is set and an ADC scan end interrupt is enabled. If the module event information is desired, add this module to the configuration and enable the interrupt in the property box.

Calling the `versionGet` API retrieves the driver versions set in `r_fmi.h` and `r_fmi_api.h` for output with the printf() routine.

Note: The descriptions provided assume that you are familiar with using printf() and the debug console in the Synergy Software Package. If you need to review, see the Knowledge Base article, "How do I Use Printf() with the Debug Console in the Synergy Software Package." A link to the Knowledge Base is provided in the FMI HAL Module References section. Alternatively, the user can see the results via the watch variables in the debug mode.

Since the FMI HAL Module is in the HAL/Common Thread stack by default, typically there is no need to change settings in the configuration. However, there are a few key properties configured in this application project to support the required operations and the physical properties of the target board and MCU. The following tables lists these properties with the values set for this specific project. You can also open the application project and view these settings in the property window as a hands-on exercise.

Table 6 FMI Configuration Settings for the Application Project

ISDE Property	Value Set
Name	g_fmi
SSP MCU Information Symbol Name	g_fmi_data
Part Number Mask	0xFE00

Table 7 GPT Configuration Settings for the Application Project

ISDE Property	Value Set
Name	g_timer0
Channel	1
Interrupt Priority	Priority 2

Table 8 ADC12 Configuration Settings for the Application Project

ISDE Property	Value Set
Name	g_adc0
Unit	0
Scan End Interrupt Priority	Priority 2

8. Customizing the FMI HAL Module for a Target Application

Since the FMI does not require any specific configuration besides the default settings, there is no need to revise the configuration.

9. Running the FMI HAL Module Application Project

To run the FMI Application project and see it executed on a target kit, simply import it into your ISDE, compile and run debug.

To implement the FMI application in a new project, use the following steps to define, configuring, auto-generating files, adding code, compiling and debugging on the target kit. Following these steps provides a hands-on approach that helps make the development process with SSP more practical, while just reading over this guide tends to be more theoretical.

Note: The steps provided have 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.

To create and run the FMI Application Project, use the following steps:

1. Create a new Renesas Synergy project for the S7G2-SK called **FMI_HAL_MG_AP**.
2. Select the **Threads** tab.
3. Confirm FMI, GPT, and ADC are added to the HAL/Common thread stack. If they are not, add them.
4. Click **Generate Project Content**.
5. Add code from the supplied project file `fmi_hal_mg.c` or copy over the generated `fmi_hal_mg.c` file.
6. Connect the USB cable to the host PC (use connector J19 DEBUG_USB).
7. Start to debug the application.
8. The output can be viewed in the Renesas Debug Virtual Console.

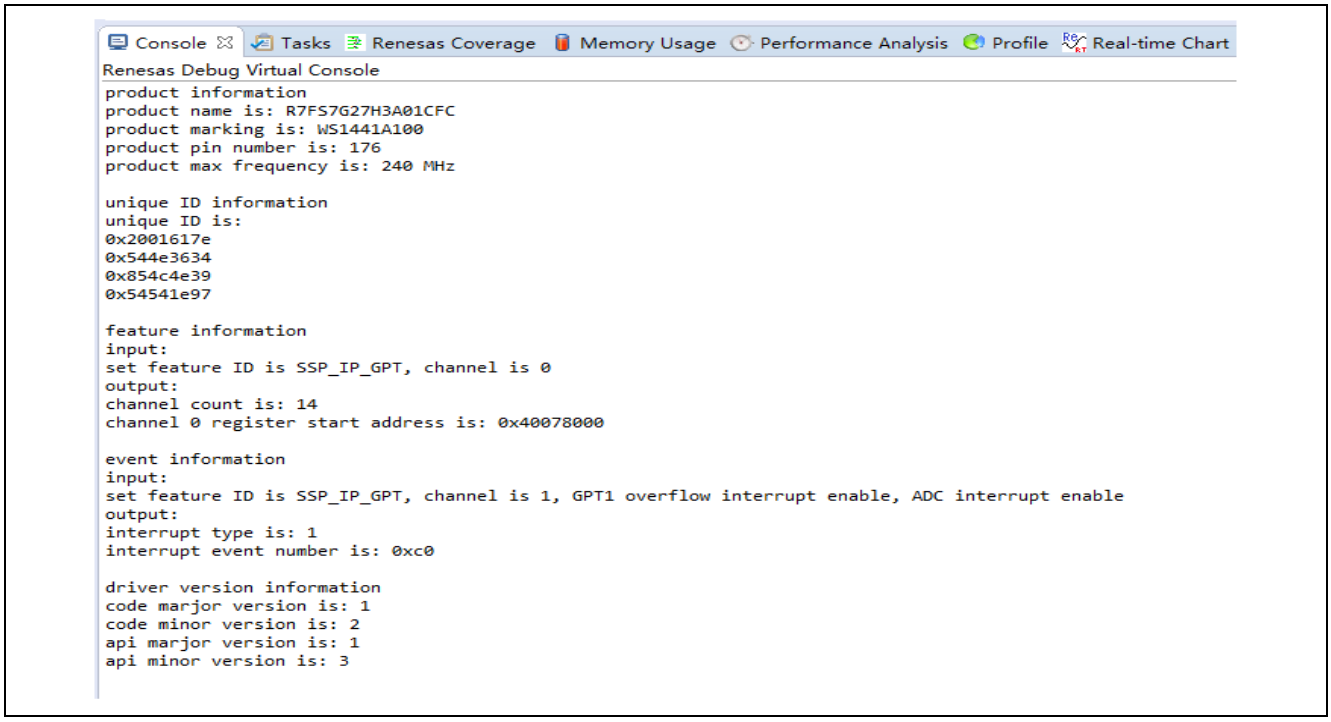

Figure 5 Example Output from FMI HAL Module Application Project

10. FMI HAL Module Conclusion

This module guide has provided all the background information needed to select, add, configure and use the module in an example project. Many of these steps were time consuming and error-prone activities in previous generations of embedded systems. The Renesas Synergy Platform makes these steps much less time consuming and removes the common errors, like conflicting configuration settings or the incorrect selection of low-level drivers. The use of high-level APIs (as demonstrated in the application project) 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. FMI HAL Module Next Steps

After you have mastered a simple FMI HAL Module project, you may want to review a more complex example. In particular, it is useful to the board support package (BSP) driver to get product feature and event information for every module in the initialization or open routine.

12. FMI HAL Module Reference Information

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

Links to all the most up-to-date r_fmi 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_FMI_Module_Guide_References.

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	May 24, 2017		Initial version
1.01	Aug 23, 2017		Update the Hardware and Software Resources Table

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)

RENESAS

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